

# Vorhersage von Aktienkursen mithilfe rekurrenter Neuronaler Netze

Sandra Litz



# Impressum

## Inhaltlich verantwortlich

Autor/-in der Abschlussarbeit

## Institution

Der Fachbereich Automatisierung und Informatik ist ein Fachbereich der Hochschule Harz. Die Hochschule Harz ist eine Körperschaft des öffentlichen Rechts. Sie wird durch den Rektor Prof. Dr. Folker Roland gesetzlich vertreten: info@hs-harz.de.

## Umsatzsteuer-Identifikationsnummer

DE231052095

## Adresse

Hochschule Harz  
Fachbereich Automatisierung und Informatik  
Friedrichstraße 57-59  
38855 Wernigerode

## Kontakt

Dekanin des Fachbereiches Automatisierung und Informatik  
Prof. Dr. Andrea Heilmann  
**Tel.:** +49 3943 659 300  
**Fax:** +49 3943 659 300  
**E-Mail:** dekanin-ai@hs-harz.de

## Aufsichtsbehörde

Das Ministerium für Wirtschaft, Wissenschaft und Digitalisierung des Landes Sachsen-Anhalt (MW), Hasselbachstraße 4, 39104 Magdeburg, ist die zuständige Aufsichtsbehörde.

## ISSN 2702-2293

## Haftungsausschluss

Die Hochschule Harz weist auf Folgendes hin:

Die Hochschule Harz ist lediglich für die Veröffentlichung der einzelnen Werke zuständig, sie übernimmt keinerlei Haftung. Vielmehr gilt Folgendes:

- für den Inhalt der Publikation ist der/die Autor/-in verantwortlich
- mit der Erfassung in der Schriftenreihe Wernigeröder Automatisierungs- und Informatik-Texte verbleiben die Urheberrechte beim Autor/bei der Autorin
- die Einhaltung von Urheber- und Verwertungsrechten Dritter liegt in der Verantwortung des Autors/der Autorin

Vor Veröffentlichung bestätigte der/die Autor/-in,

- dass mit der Bereitstellung der Publikation und jedes Bestandteils (z.B. Abbildungen) nicht gegen gesetzliche Vorschriften verstoßen wird und Rechte Dritter nicht verletzt werden
- dass im Falle der Beteiligung mehrerer Autoren am Werk der/die unterzeichnende Autor/-in stellvertretend im Namen der übrigen Miturheber/-innen handelt
- im Falle der Verwendung personenbezogener Daten den Datenschutz (durch Einholen einer Einwilligung des Dritten zur Veröffentlichung und Verbreitung des Werks) zu beachten
- dass im Falle einer bereits erfolgten Veröffentlichung (z.B. bei einem Verlag) eine Zweitveröffentlichung dem Verlagsvertrag nicht entgegensteht
- dass die Hochschule Harz von etwaigen Ansprüchen Dritter (z.B. Mitautor/-in, Miturheber/-in, Verlage) freigestellt ist

Hochschule Harz – Hochschule für angewandte Wissenschaften

Fachbereich Automatisierung und Informatik



Masterarbeit

# **Vorhersage von Aktienkursen mithilfe rekurrenter Neuronaler Netze**

Autor:

Sandra Litz

29. August 2019

Betreuer:

Prof. Dr. Frieder Stolzenburg

Prof. Dr.-Ing. Thomas Leich

Fachbereich Automatisierung und Informatik

**Sandra Litz:**

*Vorhersage von Aktienkurse mithilfe rekurrenter neuronaler Netze* Masterthesis,  
Hochschule Harz, 2019.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>viii</b>
<b>Tabellenverzeichnis</b>	<b>x</b>
<b>Quelltextverzeichnis</b>	<b>xi</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Problemstellung . . . . .	1
1.2 Zielstellung der Arbeit . . . . .	2
1.3 Verwandte Arbeiten . . . . .	3
1.4 Aufbau der Arbeit . . . . .	4
<b>2 Grundlagen</b>	<b>7</b>
2.1 Die Aktienanalyse . . . . .	7
2.1.1 Begriffe des Finanzmarktes . . . . .	8
2.1.2 Die Kursbildung an der Börse . . . . .	9
2.1.3 Die Fundamentalanalyse . . . . .	10
2.2 Statistische Zeitreihenanalyse . . . . .	12
2.2.1 Zeitreihen und Zeitreihenanalyse . . . . .	12
2.2.2 Stochastische Prozesse . . . . .	13
2.2.3 Komponentenmodell der Zeitreihe . . . . .	15
2.2.4 Statistische Analysemethoden . . . . .	16
2.2.4.1 Lineare Regression . . . . .	16
2.2.4.2 Autoregressive Modelle . . . . .	19
2.3 Künstliche Neuronale Netze . . . . .	25
2.3.1 Aufbau Neuronaler Netze . . . . .	25
2.3.2 Mathematische Darstellung eines Neuronalen Netzes . . . . .	26
2.3.3 Lernen Neuronaler Netze . . . . .	27
2.3.4 Rekurrente Neuronale Netze . . . . .	28
2.3.4.1 Echo State Network . . . . .	30
2.3.4.2 Predictive Neural Network . . . . .	32
<b>3 Die Vorhersagemethoden</b>	<b>37</b>

3.1	Die Datenerhebung . . . . .	37
3.2	Die Modellbildung der Linearen Regression . . . . .	38
3.2.1	Die Einzelkursvorhersage durch die Linearen Regression . . . . .	38
3.2.2	Die modifizierten Vorhersagen mithilfe der Linearen Regression	41
3.3	Die Modellbildung eines ARIMA-Modells . . . . .	41
3.3.1	Die Einzelkursvorhersage mithilfe eines ARIMA- Modells . . . . .	42
3.3.2	Die Einzelkursvorhersage mithilfe eines SARIMA- Modells . . . . .	47
3.3.3	Die modifizierten Vorhersagen mithilfe des ARIMA-/SARIMA- Modells . . . . .	49
3.4	Die Modellbildung des Predictive Neural Network . . . . .	50
3.4.1	Die Einzelkursvorhersage mithilfe des PrNN . . . . .	50
3.4.1.1	Die Initialisierung des PrNN . . . . .	51
3.4.1.2	Das Lernen der Ausgabegewichte . . . . .	52
3.4.1.3	Die Schätzung des Aktienkurses . . . . .	52
3.4.2	Vorhersage aller Aktienkurse . . . . .	53
3.4.3	Vorhersage eines einzelnen Aktienkurses aus allen . . . . .	55
<b>4</b>	<b>Evaluierung</b>	<b>59</b>
4.1	Die Auswertung der Einzelkursvorhersage . . . . .	59
4.1.1	Die Lineare Regression . . . . .	60
4.1.2	Das ARIMA-Modell . . . . .	64
4.1.3	Das SARIMA-Modell . . . . .	68
4.1.4	Das Predictive Neural Network . . . . .	71
4.2	Die Auswertung der Vorhersage aller Aktienkurse gleichzeitig . . . . .	74
4.3	Die Auswertung der Vorhersage eines einzelnen Aktienkurses aus allen	77
4.4	Gesamtauswertung der Ergebnisse . . . . .	80
4.4.1	Die Zuverlässigkeit der Prognosemethoden . . . . .	81
4.4.2	Berücksichtigung der Fundamentaldaten . . . . .	84
4.4.3	Der Vergleich der Methoden hinsichtlich der Anwendung . . . . .	85
<b>5</b>	<b>Zusammenfassung der Ergebnisse</b>	<b>87</b>
5.1	Zusammenfassung der Arbeit . . . . .	87
5.2	Weitere Forschungsziele . . . . .	88
<b>Anhang A Anhang</b>		<b>89</b>
A.1	Liste der Aktienkurse . . . . .	89
<b>Anhang B Skripte der Vorhersagemethoden</b>		<b>91</b>
B.1	Skript zur Linearen Regression . . . . .	91
B.2	Skript zum ARIMA-Modell . . . . .	92
B.3	Skript zum SARIMA-Modell . . . . .	93
B.4	Skript zum PrNN . . . . .	95

---

<b>Anhang C</b>	<b>Auswertung der Vorhersagen</b>	<b>105</b>
C.1	Die Vorhersage mit der Linearen Regression . . . . .	105
C.1.1	Die Abbildungen zur Vorhersage mit der Linearen Regression .	105
C.1.2	RMSE zur Vorhersage mit der Linearen Regression . . . . .	112
C.2	Die Vorhersage mit dem ARIMA-Modell . . . . .	115
C.2.1	Die Abbildungen zur Vorhersage mit dem ARIMA-Modell . .	115
C.2.2	Der RMSE zum ARIMA-Modell . . . . .	121
C.3	Die Abbildungen zur Vorhersage mit dem SARIMA-Modell . . . . .	124
C.3.1	Tabelle der Abweichungen zum SARIMA- Modell . . . . .	128
C.3.2	Tabelle der Abweichungen zum ARIMA/SARIMA - Modell . .	131
C.4	Die Vorhersage mit dem Predictive Neural Network . . . . .	133
C.4.1	Die Abbildungen zur Vorhersage mit dem PrNN . . . . .	133
C.4.2	Tabelle der Abweichungen zum PrNN (Einzelkursvorhersage) .	139
C.4.3	Tabelle der Abweichungen zum PrNN (Multiple Vorhersage) .	142
C.4.4	Tabelle der Abweichungen zum PrNN (Einer aus allen) . . . .	145
C.4.5	Tabelle der Abweichungen zur Vorhersage mit den Fundamen- taldaten . . . . .	148
	<b>Literaturverzeichnis</b>	<b>151</b>





# Abbildungsverzeichnis

2.1	Kursverlauf der IBM-Aktie über ein Jahr [1] . . . . .	10
2.2	Vorwärtsgerichtetes KNN, Quelle: Eigene Darstellung . . . . .	25
2.3	Rekurrente Verbindungen, Quelle: Eigene Darstellung . . . . .	29
2.4	Echo state network [2] . . . . .	30
3.1	Test- und Trainingsdaten des Aktienkurses von Adidas . . . . .	39
3.2	Lineares Regressionsmodell . . . . .	40
3.3	Die transformierte Zeitreihe . . . . .	43
3.4	Automatischer Suchlauf für das ARIMA(p,d,q)-Modell . . . . .	45
3.5	Analyse der Residuenanalyse . . . . .	46
3.6	Geschätzter Aktienkurs mithilfe des ARIMA-Modells . . . . .	47
3.7	Geschätzter Aktienkurs mithilfe des SARIMA-Modells . . . . .	49
3.8	Geschätzter Aktienkurs mithilfe des PrNN . . . . .	53
3.9	Geschätzter Aktienkurs mithilfe des Multi-PrNN . . . . .	55
3.10	Geschätzter Aktienkurs mithilfe des PrNN (Einer aus allen) . . . . .	56
4.1	Vorhersage des Adidas-Aktienkurses mithilfe der Linearen Regression (762 Daten) . . . . .	61
4.2	Vorhersage des Adidas-Aktienkurses mithilfe der Linearen Regression (500 Daten) . . . . .	62
4.3	Vorhersage des Adidas-Aktienkurses mithilfe der Linearen Regression (300 Daten) . . . . .	63
4.4	Vorhersage des Adidas-Aktienkurses mit dem ARIMA-Modell (762 Daten) . . . . .	64

4.5	Vorhersage des RWE-Aktienkurses mit dem ARIMA-Modell (762 Daten)	65
4.6	Vorhersage des Adidas-Aktienkurses mit dem ARIMA-Modell (500 Daten)	66
4.7	Vorhersage des Adidas-Aktienkurses mit dem ARIMA-Modell (300 Daten)	67
4.8	Vorhersage des Adidas-Aktienkurses mit dem SARIMA-Modell (762 Daten)	68
4.9	Vorhersage des Adidas-Aktienkurses mit dem PrNN (762 Daten - 200 Neuronen)	72
4.10	Vorhersage des Adidas-Aktienkurses mit dem PrNN (762 Daten - 400 Neuronen)	72
4.11	Vorhersage des Adidas-Aktienkurses mit dem PrNN (762 Daten - 600 Neuronen)	73
4.12	Vorhersage des Adidas-Aktienkurses mit dem Multi-PrNN (762 Daten - 200 Neuronen)	75
4.13	Vorhersage des Adidas-Aktienkurses mit dem Multi-PrNN (762 Daten - 400 Neuronen)	75
4.14	Vorhersage des Adidas-Aktienkurses mit dem Multi-PrNN (762 Daten - 600 Neuronen)	76
4.15	Vorhersage des Adidas-Aktienkurses mit dem PrNN (Einer aus allen, 762 Daten - 200 Neuronen)	78
4.16	Vorhersage des Adidas-Aktienkurses mit dem PrNN (Einer aus allen, 762 Daten - 400 Neuronen)	79
4.17	Vorhersage des Adidas-Aktienkurses mit dem PrNN (Einer aus allen, 762 Daten - 600 Neuronen)	79
4.18	Vorhersage des Adidas-Aktienkurses mit dem PrNN und dem KGV	84
C.1	Vorhersage mithilfe der Linearen Regression	110
C.2	Vorhersage der Aktienkurse mithilfe des ARIMA-Modells	119
C.3	Vorhersage der Aktienkurse mithilfe des ARIMA-Modells	126

# Tabellenverzeichnis

3.1	Die Koeffizienten des ARIMA(2,1,2)-Modells . . . . .	45
3.2	Optionen der Vorhersage . . . . .	51
4.1	Auswertung der Linearen Regressionsmodelle nach Anzahl der Daten	63
4.2	Auswertung der ARIMA-Modelle nach Anzahl der Daten . . . . .	67
4.3	Die Zuordnung der Modellformen . . . . .	69
4.4	Anzahl verbesserter ARIMA-Modelle . . . . .	70
4.5	Auswertung der Vorhersagen der Modelle ARIMA/SARIMA . . . . .	71
4.6	Auswertung der Einzelkursvorhersage (PrNN) nach Anzahl der Daten und Neuronen . . . . .	74
4.7	Auswertung der Multi-Vorhersage (PrNN) nach Anzahl der Daten und Neuronen . . . . .	77
4.8	Auswertung der Vorhersage einer aus allen (PrNN) nach Anzahl der Daten und Neuronen . . . . .	80
4.9	Gesamtauswertung der Einzelkursvorhersage . . . . .	81
4.10	Gesamtauswertung des PrNN (Einzel - Multi) . . . . .	83
4.11	Gesamtauswertung des PrNN (Einzel - Einer aus allen) . . . . .	83
C.1	RMSE zu den Linearen Regressionsvorhersagen . . . . .	112
C.4	RMSE der ARIMA-Modell Vorhersagen . . . . .	121
C.7	RMSE der SARIMA-Modell Vorhersagen . . . . .	128
C.10	RMSE der kombinierten SARIMA-Modell Vorhersagen . . . . .	131
C.13	RMSE zur Einzelkursvorhersage . . . . .	140

C.14 RMSE zur Multi-Vorhersage . . . . .	143
C.15 RMSE zur Vorhersage einer aus allen . . . . .	146
C.16 RMSE der Vorhersagen mit den Fundamentaldaten . . . . .	148

# Quelltextverzeichnis

B.1	Skript zum Linearen Regressionsverfahren . . . . .	91
B.2	Skript zum ARIMA-Modell . . . . .	92
B.3	Skript zum SARIMA-Modell . . . . .	94
B.4	prep_prediction.m . . . . .	95
B.5	compute_predictions.m . . . . .	95
B.6	read_in_data.m . . . . .	96
B.7	single_stock.m . . . . .	97
B.8	predict.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg) . . . . .	98
B.9	reservoir.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg) . . . . .	100
B.10	compute.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg) . . . . .	100
B.11	nrmse.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg) . . . . .	100
B.12	multiple_stock.m . . . . .	101
B.13	single_from_multiple.m . . . . .	102
B.14	fundamental_stock.m . . . . .	103



**Thema und Aufgabenstellung der Masterarbeit  
MA AI 01/2019**

**Für Sandra Litz**

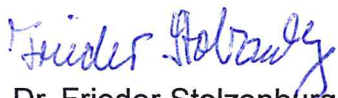
**Vorhersage von Aktienkursen mithilfe von rekurrenten neuronalen Netzen**

Die Vorhersage der zukünftigen Entwicklung von Aktienkursen ist eines der gefragtesten, gleichzeitig aber auch komplexesten und schwierigsten Themen des Aktienmarktes. Zuverlässige Prognosen sollen Investoren und Unternehmen dabei unterstützen, den Verlauf von Aktienkursen vorherzusehen, um z.B. das Risiko in falsche Aktien zu investieren zu minimieren. Viele unterschiedliche Methoden und Techniken, wie z.B. Regressionsanalysen oder Neuronale Netze, werden zunehmend eingesetzt, um die Aktienkursentwicklung vorherzusagen zu können.

In dieser Masterarbeit werden ausgewählte Methoden der Regressionsanalyse, autoregressive Modelle und rekurrente neuronale Netze zur Vorhersage von Aktienkursen eingesetzt. Das Ziel ist es, zu identifizieren, inwiefern Aktienkurse mithilfe von rekurrenten neuronalen Netzen vorhergesagt werden können und wie hoch dessen Zuverlässigkeit im Vergleich zu den anderen Methoden ist. Vor allem soll in diesem Zusammenhang eine einfache Form der rekurrenten neuronalen Netze, das Predictive Neural Network (PrNN), näher untersucht und eine entsprechende Anwendung erstellt werden. Dabei wird ebenfalls die Zuverlässigkeit im Vergleich zu anderen Methoden geprüft. Genutzt werden historische Aktiendaten von mindestens 30 Unternehmen, welche parallel als Eingabewerte (Input) für das PrNN verwendet werden. Der Verlauf dieser Kurse soll für einen bestimmten Zeitraum vorhergesagt werden.

**Die Masterarbeit beinhaltet folgende Aufgaben:**

- Recherche und Analyse ausgewählter Vorhersagemethoden
- Entwicklung einer Anwendung zu PrNN mit Octave
  - Verwendung von mindestens 30 Aktienkursen als Input
  - Vorhersage eines Aktienkurses, ggf. aus dem Verlauf der anderen
- Vergleich und Auswertung der Ergebnisse
- Analyse von möglichen Einflüssen auf Aktienkurse



Prof. Dr. Frieder Stolzenburg  
1. Prüfer



Prof. Dr.-Ing. Thomas Leich  
2. Prüfer





# Danksagung

An dieser Stelle möchte ich mich bei allen, die mich bei der Entstehung dieser Masterarbeit unterstützt haben, herzlich bedanken.

Einen Besonderen Dank möchte ich meinem Betreuer Herrn Prof. Dr. Frieder Stolzenburg aussprechen. Ich bedanke mich herzlich für die Unterstützung bei der Themenfindung und der sehr guten Betreuung bei der Bearbeitung meiner Masterarbeit.

Zudem Bedanke ich mich bei Herrn Prof. Dr.-Ing. Thomas Leich für die Betreuung meiner Arbeit und die Bereitschaft, die Rolle des Zweitprüfers zu übernehmen.

Besonders möchte ich mich bei meiner Familie und meinen Freunden für das Verständnis, die Geduld und die Unterstützung bei der Bearbeitung meiner Masterarbeit bedanken.



# 1. Einführung

Dieser Abschnitt der Ausarbeitung stellt zunächst die Problemstellung vor, die mithilfe dieser Arbeit untersucht werden soll. In diesem Zusammenhang wird dementsprechend das Ziel dieser Ausarbeitung definiert. Zusätzlich werden einige verwandte Arbeiten vorgestellt, die den aktuellen Forschungsstand hinsichtlich der Problemstellung beschreiben. Abschließend wird in diesem Kapitel der Aufbau der Arbeit erläutert.

## 1.1 Problemstellung

Seit Jahren ist der Handel mit Aktien eine beliebte Methode viel Geld in kurzer Zeit oder aber auch auf lange Sicht zu erwirtschaften. Wird zur richtigen Zeit in die 'richtige' Aktie investiert, d.h. eine Aktie gewählt, die einen zukünftigen Wertanstieg verspricht, ermöglicht dies einem potenziellen Käufer hohe Gewinne zu erzielen. Die richtige Aktie zu identifizieren, stellt jedoch viele Marktteilnehmer an der Börse vor eine große Herausforderung. Da der zukünftige Verlauf des Aktienkurses unklar ist, bringt der Handel auch das hohe Risiko mit sich, das gesetzte Vermögen bei einem Wertabstieg innerhalb von kurzer Zeit zu verlieren. Spätestens nach den großen Finanzkrisen wird deutlich, welche Ausmaße die Spekulation an der Börse auf die Investoren haben kann. Der Zusammenbruch der Börse im Jahr 1929 in den USA, auch bekannt als 'schwarzer Donnerstag' oder die Finanzkrise im Jahr 2008, welche sich auch auf die Wirtschaft in Deutschland auswirkte, führten zur Insolvenz vieler Unternehmen und Banken und hatte eine hohe Massenarbeitslosigkeit zur Folge [3, S.315]. Das Ergebnis waren hohe finanzielle Verluste der Aktionäre, aber auch hohe Staatsverschuldungen. Neben plötzlichen Börsenzusammenbrüchen führen häufige Trendwechsel der Aktienkurse, d.h. der häufige Preisan- und abstieg dieser Kurse, ebenfalls zu finanziellen Schäden für die beteiligten Marktteilnehmer.

Schon vor diesen Wirtschaftskrisen, aber spätestens zu diesem Zeitpunkt, werden zunehmend Methoden entwickelt, die es ermöglichen sollen, einen Aktienkursverlauf vorherzusagen. Damit sollen die bestehenden Unsicherheiten in Aktien zu investieren beseitigt, hohe finanzielle Verluste verhindert und große Gewinne erzielt werden. Dementsprechend bestehend heute viele unterschiedliche Verfahren und Prognosemethoden, welche unter anderem auf Basis des historischen Aktienkursverlaufs zuverlässige Vorhersagen ermöglichen sollen. Neben klassischen Autoregressiven Methoden, erfolgt dabei der vermehrte Einsatz Künstlicher Neuronaler Netze zur Analyse und Vorhersage von Aktienkursen. Erweitert wird die Auswahl mit kombinierten Prognoseverfahren. Damit steigt die Auswahl an Prognosemethoden, die bei Investitionsentscheidungen als Entscheidungshilfe hinzugezogen werden kann, an.

Die unterschiedlichen Prognosemethoden unterscheiden sich dabei hinsichtlich der Vorgehensweise, die vorliegenden Aktienkurse zu analysieren und zu prognostizieren. Entsprechend der Vorgehensweisen variiert ebenfalls die Zuverlässigkeit der Prognosen. Diese ist bei Investitionen jedoch ausschlaggebend, da damit Fehlentscheidungen in Bezug auf möglicher Aktienkäufe oder -verkäufe verhindert werden können. Entsprechend der ermittelten Prognosen können die Investitionsentscheidungen frühzeitig getroffen und ausgeführt werden.

Aus der Vielfalt an Auswahlmöglichkeiten ist für viele Marktteilnehmer unklar, welche Methode die besten Prognosen ermöglicht, sodass die bestmöglichen Investitionsentscheidungen am Aktienmarkt getroffen werden können. Um das beste Verfahren jedoch auswählen zu können, ist ein direkter Vergleich der Prognosemethoden notwendig.

## 1.2 Zielstellung der Arbeit

Das Ziel dieser Arbeit ist es, aus ausgewählten bekannten Vorhersagemethoden zu identifizieren, welches Verfahren im Vergleich die besten Vorhersagen ermöglicht. Bei den ausgewählten Methoden handelt es sich zum einen um klassische Modelle, die schon seit vielen Jahren zur Prognose von Aktienkursen eingesetzt werden. Hierzu wird das Lineare Regressionsmodell, sowie die autoregressiven Modelle "autoregressive integrated moving average" (ARIMA) und das "saisonal autoregressive integrated moving average" (SARIMA) eingesetzt. Zum anderen soll neben den aufgeführten klassischen Modellen ein innovativer Ansatz im Bereich der künstlichen neuronalen Netze getestet werden. Dabei handelt es sich um das Predictive Neural Network, welches die Prognose mithilfe einer Form eines rekurrenten Neuronalen Netzes ermöglicht. Die Prognose aller Aktienkurse erfolgt dabei auf Basis des historischen Verlaufs der Kurse.

Zur Bewertung und zum Vergleich der Modelle zueinander, sollen dabei unter anderem für ein repräsentatives Ergebnis 30 Aktienkurse mit jeder Methode einzeln vorhergesagt werden. Da in der Regel bei Investitionsentscheidungen nicht nur ein

Aktienkurs sondern parallel mehrere Aktienkurse untersucht werden, gilt es zudem innerhalb dieser Ausarbeitung zu untersuchen, welche Modelle das Einlesen und die Prognose aller Aktienkurse gleichzeitig ermöglicht und wie sich der Einfluss der Aktienkurse dabei auf das Prognoseergebnis auswirkt. Mittels dieser Form der Prognose, ermöglicht dies bei Investitionsentscheidungen an der Börse, mehrere Aktienkurse gleichzeitig zu untersuchen und im direkten Vergleich gegenüberzustellen. Zudem soll in diesem Zusammenhang untersucht werden, inwiefern die Prognose optimiert werden kann, sollte ein Aktienkurs aus allen anderen Aktienkursen vorhergesagt werden. Die Annahme bei dieser Prognoseform besteht hierbei darin, dass ein Zusammenhang zwischen einzelnen Aktienkursen vorliegt. Dabei soll identifiziert werden, ob ein Trendaufstieg bzw. -abstieg eines Aktienkurses ein Indiz für einen Aufstieg- bzw. Abstieg eines anderen Aktienkurses ist und damit der Verlauf eines Aktienkurses auf Basis des Verlaufs eines anderen prognostiziert werden kann.

Da im Rahmen der Vorhersage der Aktienkurse unklar ist, welche Anzahl einzulesender Daten für die Prognose der Aktienkurse die zuverlässigsten Vorhersagen ermöglicht, soll ebenfalls der Einfluss der Anzahl der vergangenen Werte einer Zeitreihe untersucht werden. Das Ziel hierbei ist es zudem zu identifizieren, welche Prognosemethoden tendenziell für langfristige oder kurzfristige Prognosen besser geeignet ist.

Zudem soll eine Kombination der historischen Daten eines Aktienkurses mit unterschiedlichen Fundamentalkennzahlen des jeweiligen Unternehmens prognostiziert werden. Diese Kennzahlen beschreiben den wirtschaftlichen Erfolg eines Unternehmens. Dabei soll im Rahmen dieser Ausarbeitung abschließend die beste Prognosemethode ermittelt und anschließend getestet werden, inwiefern die Prognose durch den Einfluss von Fundamentaldaten verbessert werden kann.

Als Ergebnis soll damit bestimmt werden, welche Methode in welcher Form die im Vergleich besten Prognosen ermöglicht. Dabei sollen ebenfalls die Vorteile, Nachteile und Grenzen der einzelnen Methoden erkannt werden. Vor allem soll dabei die Eignung des Predictive Neural Network als neues Verfahren zur Prognose von Aktienkursen untersucht werden.

### 1.3 Verwandte Arbeiten

Da wie schon einleitend erfasst wurde, die Vorhersage von Aktienkursen eine herausfordernde Problemstellung und damit ein beliebtes Forschungsziel darstellt, bestehen viele unterschiedliche Ansätze und Methoden zur Prognose. Neben der Weiterentwicklung der klassischen Prognosemethoden und der Methoden der künstlichen neuronalen Netze, wird alternativ häufig eine Kombination dieser Modelle und Methoden untersucht. Der Einsatz kombinierter Modelle erfolgt vor allem im Rahmen der Prognose mit dem ARIMA-Modell.

Zur Prognose von Zeitreihen verwenden Mei et al. [4] dabei eine Kombination aus einem ARIMA-Modell und einer “support vector machine” (SVM, dt.: Stützbekformmaschine) vor, die eine zuverlässige Prognose der Aktienkurse ermöglichen soll. Dabei erfolgt die erste Vorhersage mit dem ARIMA-Modell. Im Anschluss werden die Residuen, d.h. die Abweichungswerte zwischen den Prognosen und dem originalen Aktienkurs weiter mittels eines SVM untersucht. Die wichtigsten Informationen des Aktienkurses, die dabei nicht mithilfe des ARIMA-Modells erfasst werden konnten, werden mit der SVM analysiert. Damit sollen die wichtigsten Muster des Aktienkurses erkannt und diese prognostiziert werden.

Im Rahmen der Zeitreihenprognose erfolgt ebenfalls zunehmend der Einsatz neuronaler Netze. Rebane et al. [5] stellen hierzu in ihrer Arbeit den Einsatz rekurrenter neuronaler Netze vor, die im Vergleich zum klassischen ARIMA-Modell, die Vorhersage der Kryptowährung ermöglichen soll. Mithilfe eines überwachten Lernalgorithmus, dem “Sequence-to-Sequence”-Verfahren, welches zur Umwandlung bzw. Vorhersage von Sprachsequenzen zu Textsequenzen genutzt wird, sollen ebenfalls die Vorhersage der Zeitreihe der Kryptowährung erfolgen. Mit einem Prognosefenster von 40 Tagen konnten mithilfe des RNN besser Vorhersagen als mit dem ARIMA-Modell erzielt werden. Zudem war es mit dem RNN nicht möglich, starke Preisabfälle zu prognostizieren.

Ein weiteres bekanntes künstliches neuronales Netz, welches zur Prognose von Zeitreihen untersucht wird, ist das “Long short-term memory”-Netz (LSTM). Dieses wird als eigenes Modell zur Prognose eingesetzt [6] oder aber auch in Kombination anderer Modelle und Methoden zur Optimierung bestehender LSTM-Netze verwendet.

Viele eingesetzte Methoden werden dabei jedoch einzeln untersucht und stehen dabei nur selten im Vergleich zu anderen Modellen. Aufgrund dessen, wird meist nicht deutlich, welche Methoden im Vergleich dabei die besten Vorhersagen ermöglichen. In diesem Zusammenhang erfolgt in dieser Ausarbeitung der Vergleich zwischen drei Modellen zur Prognose von Aktienkursen. Als neuer Ansatz wird hierbei das Predictive Neural Network (PrNN) nach Stolzenburg et al. [7] analysiert, welches zur Vorhersage von Aktienkursen noch nicht untersucht wurde.

## 1.4 Aufbau der Arbeit

Die Ausarbeitung beginnt mit der Beschreibung der Grundlagen in Kapitel 2, die für das Verständnis der ausgewählten Methoden essentiell sind. Zunächst werden dabei die Grundlagen des Lineare Regressionsmodells, anschließend des ARIMA-Modells bzw. als Erweiterung das SARIMA-Modell erläutert. Abschließend erfolgt in diesem Kapitel die Beschreibung der Grundlagen der neuronalen Netze sowie die der rekurrenten neuronalen Netze, die ausschlaggebend für den Aufbau des PrNNs sind.

---

Im dritten Kapitel der Ausarbeitung wird der Aufbau der ausgewählten Methoden aufgeführt. Da jede Methode eine unterschiedliche Vorgehensweise zur Bestimmung der Vorhersagen aufweist, wird in diesem Abschnitt der Aufbau der jeweiligen Quellcodes, sowie die wichtigsten Schritte und Berechnungen zur Modellbildung und Prognose aufgeführt. Beginnend wird dabei die Modellbildung des Linearen Regressionsverfahrens, sowie des ARIMA bzw. SARIMA-Modells erläutert. Abschließend wird der Aufbau des Predictive Neural Networks näher beschrieben. Für jede Methode wird dabei gegebenenfalls der entsprechende Quellcode und die Vorgehensweise zur Vorhersage aller Aktienkurse gleichzeitig, als auch zur Prognose eines Aktienkurses auf Grundlage aller vorgestellt.

Kapitel 4 beinhaltet im Anschluss die Ergebnisse der Vorhersagen, sowie die Evaluation der Ergebnisse. Für jedes Modell wird dabei zunächst die Prognose eines einzelnen Aktienkurses in Kombination einer variierenden Anzahl an Beobachtungen vorgestellt. Dies wird daraufhin für die Vorhersage aller 30 Aktienkurse gleichzeitig, als auch für die Vorhersage eines Aktienkurses aus allen für alle Prognosemethoden aufgeführt. Abschließend erfolgt der direkte Vergleich der einzelnen Methoden und eine Gesamtbeurteilung der Ergebnisse. Dabei wird ebenfalls mit der ermittelten besten Vorhersagemethode die Fundamentaldaten als zusätzlicher Faktor eingelesen und identifiziert, inwiefern mithilfe dieser Daten die Prognose optimiert werden kann.

Im abschließenden Kapitel 5 werden die wichtigsten Erkenntnisse der Arbeit zusammengefasst.





## 2. Grundlagen

Dieses Kapitel beinhaltet die wichtigsten Grundlagen, die für die Vorhersage von Aktienkursen notwendig sind. Eingeleitet wird das Kapitel mit der Aktienanalyse als Definition und damit als Einleitung in die Thematik der Ausarbeitung. Für ein besseres Verständnis des Themas werden zunächst die wichtigsten Begriffe des Aktien- und Finanzmarktes definiert und des weiteren erläutert, wie ein Aktienkurs zustande kommt. Im Rahmen der Aktienanalyse wird die Fundamentalanalyse als eines der wichtigsten Analyse- und bewertungsmethoden vorgestellt. Einige wichtige Fundamentaldaten sollen anschließend für die Vorhersage der Aktienkurse genutzt werden. Eine weitere wichtige Analyseform die erläutert wird, ist die Zeitreihenanalyse, wobei zwei bekannte Analysemethoden näher beschrieben werden. Einen weiteren Abschnitt der Grundlagen bilden die Neuronalen Netze, insbesondere die rekurrenten Neuronalen Netze. In diesem Zusammenhang wird das Predictive Neural Network, welches als neuer Ansatz zur Vorhersage von Zeitreihen genutzt werden soll, abschließend erläutert.

### 2.1 Die Aktienanalyse

Die Aktienanalyse wird von vielen Investoren und Analysten genutzt, um historische und aktuelle Daten, wie z.B. den vergangenen Schlusskurs einer Aktie zu analysieren, sowie verschiedene Einflüsse auf einen Aktienkurs näher zu untersuchen. Das Ziel der Aktienanalyse ist es, eine Aktie zu bewerten. Das bedeutet, dass dabei beurteilt wird, welches Potenzial die Aktie aufweist, um dabei dessen zukünftige Entwicklung zu schätzen. Die Vorhersage und die richtige Einschätzung und Bewertung der Aktie stellt jedoch viele Analysten und Investoren vor eine große Herausforderung. Zum einen erschwert die hohe Dynamik durch den häufigen Trendwechsel der Aktienkurse an der Börse die Vorhersage der Aktienkurse und zum anderen bestehen wie schon

einleitend festgestellt unterschiedliche Börsentheorien, die verschiedene Analyseansätze verfolgen, sodass eine hohe Auswahl an Analysemethoden vorliegt. Diese Methoden unterscheiden sich hinsichtlich der Daten und Kennzahlen, die zur Bewertung und Vorhersage untersucht werden. Dementsprechend ergeben sich unterschiedliche Lösungen. Die Bestimmung der "richtigen" Lösung ist jedoch ausschlaggebend für den Investor, da die Investitionsentscheidungen deutlich von den Ergebnissen der Analysen abhängen. Bekannte Analysemethoden sind dabei die Technische Analyse und die Fundamentalanalyse. Für sind jedoch lediglich die Fundamentaldaten aus der Fundamentalanalyse von Bedeutung.

### **2.1.1 Begriffe des Finanzmarktes**

Im Folgenden sind für diese Ausarbeitung die wichtigsten Begriffe des Finanzmarktes definiert.

#### **Anleihen**

Eine Anleihe ist, wie eine Aktie, ein Wertpapier. Dabei leiht ein Investor dem Staat oder einem Unternehmen Geld, wofür Zinsen an den Geldgeber gezahlt werden.

#### **Cashflow**

Der Cashflow ist eine Kennzahl, die den Zahlungsfluss bzw. die Zahlungsfähigkeit eines Unternehmens beschreibt [8, S.91]. Dabei werden die Einnahmen und die Ausgaben des Unternehmens verrechnet. Die Differenz zwischen den Einnahmen und den Ausgaben des Unternehmens bilden den Cashflow. Diese Kennzahl zeigt auf, wie die finanzielle Lage eines Unternehmens im vergangenen Geschäftsjahr oder der betrachteten Periode war bzw. aktuell ist.

#### **Dividende**

Die Dividende meint einen festgelegten Teil eines Gewinns, den ein Unternehmen innerhalb einer bestimmten Periode erwirtschaftet hat und den ein Aktionär erhält [9, S.89]. Diese Ausschüttung erfolgt jährlich oder quartalsweise. Die Höhe der Ausschüttung wird dabei vom Unternehmen bzw. der Aktiengesellschaft bestimmt.

#### **Rendite**

Die Rendite meint den tatsächlichen Ertrag einer Geldanlage, der innerhalb eines bestimmten Zeitraumes erwirtschaftet wird. Mit der Rendite wird demnach der Ertrag angegeben, der bei dem Verkauf einer Aktie im Vergleich zum Kaufpreis eingebracht wird. Dabei hängt die Höhe der Rendite vom wirtschaftlichen Erfolg eines Unternehmens ab [10, S.89].

#### **Sekundärmarkt**

Der Sekundärmarkt meint einen Markt an dem z.B. sich im Umlauf befindende

Wertpapiere gehandelt werden.

### **Gewinn**

Der Gewinn eines Unternehmens meint den finanziellen Überschuss, der innerhalb eines bestimmten Zeitraums bzw. einer Periode erwirtschaftet wird. Ein Gewinn besteht demnach, wenn entgegen aller Aufwendungen und Kosten des Unternehmens ein Ertrag erzielt werden konnte [11].

### **2.1.2 Die Kursbildung an der Börse**

Die Börse stellt einen organisierten Markt dar, an dem unter anderem Wertpapiere, Devisen aber auch Rohstoffe gehandelt werden [12, S.60]. Eines der bekanntesten Handelsformen ist der Wertpapierhandel, da dieser neben dem Kauf und Verkauf von unter anderem Anleihen, Fondsanteilen und Zertifikaten ebenfalls den Handel von Aktien beinhaltet. Der Aktienhandel findet auf unterschiedlichen Wegen statt. Zum einen ist der Handel entweder elektronisch über die Computerbörse, wie z.B. über den in Deutschland bekannten Handelsplatz "exchange electronic trading" (Xetra) oder physisch an der Präsenzbörse möglich. Die Wertpapierbörse bildet dabei einen Sekundärmarkt, bei welchem Käufer und Verkäufer zusammengeführt werden und der Handel ermöglicht wird [13, S.7].

Sobald ein Unternehmen Bedarf an einer großen Menge Eigenkapital hat, hat dieses die Möglichkeit sich das Kapital durch den Verkauf von Aktien an der Börse zu beschaffen. Voraussetzung hierfür ist die Gründung einer Aktiengesellschaft (AG), die nach eingehender Prüfung berechtigt ist, an der Börse mit Aktien zu handeln. Durch den Kauf der Aktien durch die Aktionäre, erhält die Aktiengesellschaft Kapital für ihre geplanten Investitionen. Die Aktionäre selbst wiederum werden zu Anteilseignern dieser AG und tragen mit dem Kauf zur wirtschaftlichen Entwicklung des Unternehmens bei [12, S.60]. Ob ein Marktteilnehmer in das Unternehmen Geld investiert, hängt von der persönlichen Erwartung an dem zukünftigen wirtschaftlichen Erfolg des Unternehmens ab. Entsprechend der Erwartungen der Marktteilnehmer werden Aktien verkauft bzw. gekauft.

Der Erwartungswert der Marktteilnehmer wird stetig durch unterschiedliche Faktoren beeinflusst. Diese Einflüsse können unter anderem von politischer Natur oder aber auch unterschiedliche wirtschaftliche Kennzahlen des Unternehmens, wie z.B. der Gewinn sein, die ausschlaggebend für die Kauf- bzw. Verkaufsentscheidung eines Aktionärs sind. Viele potenzielle Investoren berücksichtigen zur Bewertung einer Aktie bzw. eines Unternehmens wiederum nur den historischen Verlauf des Aktienpreises einer Aktie. Entsprechend der für den Investor oder Analysten ausschlaggebenden Kennzahlen wird die jeweilige Analyseform gewählt, die diese Kennzahlen analysiert und auf Basis dessen Prognosen erstellt. Sobald nach den Analyseergebnissen hohe Gewinne oder Verluste prognostiziert werden, wird empfohlen in die



Abbildung 2.1: Kursverlauf der IBM-Aktie über ein Jahr [1]

Aktie zu investiert oder diese gegebenenfalls zu verkaufen. Dadurch entwickelt sich das Angebot und die Nachfrage nach einer Aktie, wodurch der Aktienpreis reguliert wird. Dieser bewegt sich dabei in Abwärts-, Aufwärts- oder Seitwärtstrends, wobei ein Aktienkurs oftmals Schwankungen in alle Trendrichtungen aufweist. Aufgrund dessen, wird die Prognose eines Aktienkurses oftmals erschwert. Der Preisverlauf wird dabei in einem Diagramm, eher bekannt aus dem englischen als Chart visualisiert. Dabei wird die Höhe des Preises in Abhängigkeit zur Zeit dargestellt, wie z.B. in Abbildung 2.1 am Beispiel Aktie der International Business Machines Corporation (IBM) aufgeführt.

### 2.1.3 Die Fundamentalanalyse

Eine weit verbreitete Methode der Aktienanalyse und -bewertung ist die Fundamentalanalyse. Das Ziel dieser Methode ist es, eine Aktie bzw. ein Unternehmen nach ihrem "inneren Wert" zu beurteilen und diesen Wert mit dem aktuellen Kurswert zu vergleichen und die Aktie bzw. das Unternehmen dadurch zu bewerten [12, S.43]. Der "innere Wert" zeigt den objektiven Wert einer Aktie. Zur Bewertung und Berechnung des "inneren Wertes" werden unterschiedliche kursbeeinflussende Faktoren untersucht. Dabei wird eine Aktie als unterbewertet eingestuft, sobald der berechnete objektive Wert der Aktie größer ist als der Preis, zu dem die Aktie zu dem aktuellen Zeitpunkt an der Börse gehandelt wird. Trifft dies zu, wird der Kauf der Aktie empfohlen, da der Fundamentalist von einer zukünftigen Korrektur des Preises und damit einem Preisanstieg in der Zukunft ausgeht. Sollte der bestimmte "innere Wert" einer Aktie jedoch unter dem aktuellen Verkaufswert der Aktie liegen, wird dies als Verkaufssignal wahrgenommen, da dabei von einer überbewerteten Aktie ausgegangen und ein Preisabfall vermutet wird [14, S.205].

Die Herausforderung der Fundamentalanalyse liegt jedoch in der Bestimmung des “inneren Wertes” und der Bewertung einer Aktie bzw. eines Unternehmens. Hinzugezogen werden dabei unterschiedliche gesamtwirtschaftliche, branchenspezifische und Unternehmenskennzahlen, die auch Fundamentaldaten genannt werden [12, S.44]. Sobald diese identifiziert wurden, werden Schätzungen in Bezug auf die zukünftige Entwicklung erstellt.

Zur Vorhersage der Aktienkurse wird in dieser Ausarbeitung unter anderem der Ansatz der Fundamentalanalyse verfolgt. Das Ziel der Ausarbeitung ist hierbei nicht, den “inneren Wert” eines Aktienkurses zu bestimmen. Der Fokus wird hierbei daher lediglich auf die historischen und aktuellen Unternehmenskennzahlen gelegt, die Hinweise auf die wirtschaftliche Leistung und die zukünftige wirtschaftliche Entwicklung des Unternehmens geben sollen. Zur Bestimmung des “inneren Wertes” eines Aktienkurses und einer detaillierten Erläuterung der Fundamentalanalyse wird auf die Literatur [15, 9] verwiesen

Die für diese Ausarbeitung wichtigsten Daten belaufen sich auf die klassischen Kennzahlen der Fundamentalanalyse. Diese werden in der Praxis unter anderem zum Vergleich mit anderen Unternehmen oder im Vergleich zum eigenen historischen Verlauf genutzt, um festzustellen, wie die Aktie bewertet und ob es im Vergleich zu anderen richtig bewertet ist [14, S.271].

**Kurs-Gewinn Verhältnis (KGV)** Eines der wichtigsten Bewertungskennzahlen der Fundamentalanalyse ist das Kurs-Gewinn Verhältnis

$$\text{KGV} = \frac{\text{Kurs der Aktie}}{\text{jährlicher Gewinn je Aktie}}. \quad (2.1)$$

Das KGV berechnet dabei das Verhältnis zwischen dem aktuellen Aktienpreis und dem zukünftigen erwarteten Gewinn der Gesellschaft. Das Ergebnis gibt dabei an, nach wieviel Jahren bei dem betrachteten konstanten Gewinn der Preis an der Börse angemessen wäre bzw. wird dabei das Vielfache des Gewinns angegeben, mit dem an der Börse gehandelt wird [15, S.443]. Bei der Fundamentalanalyse wird daher die Aktie gewählt, die im Vergleich zu anderen Aktien einen niedrigeren KGV aufweist bzw. die im Verlauf der Jahre einen sinkenden KGV aufzeigt.

**Kurs-Cashflow Verhältnis (KCV)** Das Verhältnis zwischen dem Cashflow und dem aktuellen Kurswert ist eine weitere Kennzahl, die häufig bei der Aktienbewertung eingesetzt wird

$$\text{KCV} = \frac{\text{Kurs der Aktie}}{\text{Cashflow je Aktie}}. \quad (2.2)$$

Der aufgeführte Kurs-Cashflow Indikator zeigt, wie das Verhältnis zwischen der Differenz der Einnahmen und Ausgaben eines Unternehmens und dem zu dem Zeitpunkt betrachteten Kurspreis ist [15, S.510]. Im Vergleich zum durchschnittlichen

KCV der jeweiligen Branche der betrachteten Aktie bzw. zum eigenen historischen Verlauf des KCV der Aktie, ist eine Aktie mit einem niedrigeren KGV attraktiv für den Fundamentalisten [15, S.517].

**Dividendenrendite** Die Dividendenrendite ist eine wichtige Kennzahl, die für viele Investoren einen hohen Einfluss hinsichtlich ihrer Anlageentscheidungen hat. Dabei wird das Verhältnis der Dividende zum Aktienkurs in Prozent berechnet

$$\text{Dividendenrendite} = \frac{\text{Dividende}}{\text{Kurs der Akte}} \cdot 100\%. \quad (2.3)$$

Generell zielt der Investor auf eine Aktie mit einer konstanten oder einer steigenden Dividendenrendite ab. Denn eine höhere bzw. eine stabile Dividendenrendite weist auf ein stabiles Unternehmen mit einer konstanten Ausschüttung hin. Eine steigende Dividende kann ein Indiz für einen steigenden wirtschaftlichen Erfolg des Unternehmens sein.

## 2.2 Statistische Zeitreihenanalyse

Die statistische Zeitreihenanalyse ist eine weit verbreitete Methode zur Analyse und Prognose von Zeitreihen. Dabei stellen Aktienkurse finanzielle Zeitreihen dar, die mithilfe unterschiedlicher Methoden der Zeitreihenanalyse untersucht werden, sodass auf Grundlage der Analyseergebnisse der weitere Verlauf prognostiziert werden kann. Um Aktienkurse vorhersagen zu können, ist es zunächst wichtig zu verstehen wie eine Zeitreihe aufgebaut ist und welche Eigenschaften diese aufweist, sodass geeignete Prognosemethoden eingesetzt werden können.

### 2.2.1 Zeitreihen und Zeitreihenanalyse

Eine Zeitreihe ist eine endliche, meist diskrete zeitlich geordnet Folge von Messwerten bzw. Beobachtungen  $Y = y_1, y_2, \dots, y_n$  zum Zeitpunkt  $n \in \mathbb{N}$ , die zu einer Menge einer Variable  $Y$  zusammengefasst werden [16]. Die Menge der Beobachtungszeitpunkte können z.B. Klimadaten, die Absatzmenge eines Unternehmens oder Aktienkurse sein. Die Besonderheit bei Zeitreihen ist, dass nicht nur die Informationen der einzelnen Beobachtungen wichtig für die Auswertungen sind, sondern zusätzlich die zeitliche Anordnung der Daten berücksichtigt werden muss [17, S.2]. Die Beobachtungswerte können dabei in festen aber auch unterschiedlichen Zeitabständen auftreten. In der Wirtschaft liegen Zeitreihen jedoch oftmals mit unregelmäßigen Abständen zwischen den einzelnen Werten vor. Beispielsweise treten einzelne Werte der Schlusskurse der Aktien nur unregelmäßig auf, da diese nicht täglich erfasst werden. Der Grund dafür ist, dass an den meisten Handelsplätzen an Feiertagen und Wochenenden nicht gehandelt wird. Eine Beobachtung einer Zeitreihe kann dabei genau einen Wert (skalärer Wert) aufweisen, genannt univariate Zeitreihe oder aber

auch mehrere Variablen beinhalten. Diese Form der Zeitreihen werden als multivariate Zeitreihen bezeichnet.

Diese Zeitreihen können mithilfe der Zeitreihenanalyse analysiert werden. Da einige Zeitreihen Muster aufweisen, die sich im Laufe der Zeit wiederholen, gilt es bei der Zeitreihenanalyse zum einen, diese Regelmäßigkeiten zu identifizieren. Zum anderen sollen dabei die Relation der Beobachtungswerte einer Zeitreihe zu verschiedenen Zeitpunkten analysiert und die kausalen Zusammenhänge verstanden werden, um damit die Entwicklung nachvollziehen zu können [17, S.1]. Sobald diese Zusammenhänge und die Muster der Zeitreihe erkannt werden, können Prognosen in Bezug auf die weitere Entwicklung der Zeitreihe erstellt werden. Dies erfolgt durch die Extrapolation, d.h. der Schätzung von Prognosen auf Basis der erkannten Eigenschaften der vorliegenden Zeitreihe. Damit dies erfolgen kann ist es notwendig, ein geeignetes mathematisches Modell zu finden, welches die Struktur und die Dynamik der Zeitreihe am besten abbildet und beschreibt [18, S.1053].

Die Identifizierung eines für die entsprechende vorliegende Zeitreihe geeigneten Modells ist der erste Schritt der Zeitreihenanalyse. Der zweite Schritt besteht darin, dass die Parameter der Modelle bestmöglich geschätzt werden müssen. Je besser diese Parameter geschätzt werden, umso besser ist das Modell an die Zeitreihe angepasst, sodass zuverlässigere Vorhersagen getroffen werden können. Anschließend erfolgt die Prüfung der Güte des Modells, woraufhin die Prognose der Zeitreihen erfolgen kann. Das Verfahren der Zeitreihe wird in im Rahmen der Grundlagen der entsprechenden Prognosemethoden näher erläutert.

## 2.2.2 Stochastische Prozesse

In der statistischen Zeitreihenanalyse besteht die Annahme, dass das mathematische Modell einer Zeitreihe einem stochastischen Prozess folgt [19]. Bei einem stochastischen Prozess handelt es sich um eine zeitlich geordnete Folge von Zufallsvariablen  $\{Y_t\}$ . Allgemein werden die Zufallsvariablen des stochastischen Prozesses durch  $\{Y_t, t \in T\}$  angegeben, wobei  $T$  den Beobachtungszeitraum und  $t$  einen bestimmten Zeitpunkt angibt [20, S.24]. Ein beobachteter Wert  $y_t$  der Zeitreihe unterliegt dabei einer Zufallsvariablen  $Y_t$ . Folglich handelt es sich bei einer Zeitreihe um die Realisierung der Zufallsvariablen. Die vorliegende Zeitreihe stellt dabei nur die Realisierung einer Stichprobe mit einem begrenzten Zeitfenster des stochastischen Prozesses dar [19, S.22].

Mithilfe der Zeitreihenanalyse sollen die Eigenschaften des zugrundeliegenden stochastischen Prozesses anhand dieser Stichprobe bestmöglich bestimmt werden. Hinzugezogen werden dabei bestimmte Parameter, die diesen Prozess beschreiben. Diese Parameter werden auch Momente genannt.



$$\text{Mittelwert (Erwartungswert):} \quad \mu(t) = E(Y_t) \quad (2.4)$$

$$\text{Varianz:} \quad \sigma(t) = \text{Var}(Y_t) = E[(Y_t - \mu(t))^2] \quad (2.5)$$

$$\begin{aligned} \text{Autokovarianzfunktion:} \quad \gamma(t_1, t_2) &= \text{Cov}(Y_{t_1}, Y_{t_2}) & (2.6) \\ &= E[(Y_{t_1} - \mu(t_1)) * (Y_{t_2} - \mu(t_2))] \end{aligned}$$

$$\begin{aligned} \text{Autokorrelationsfunktion:} \quad p(\tau) &= \frac{\gamma(\tau)}{\gamma(0)} & (2.7) \\ &= \frac{\text{Cov}(Y_t, Y_{t+\tau})}{\text{Var}(Y_t)} \quad (\tau \in T) \end{aligned}$$

Der Mittelwert bzw. Erwartungswert nach Formel 2.4 gibt die Niveaulage der Zufallsvariablen des stochastischen Prozesses an. Die betrachteten Zufallsvariablen  $Y = Y_1, Y_2, \dots, Y_T$  können dabei unterschiedlich verteilt sein. Die Mittelwerte hängen dabei von den Zeitpunkten  $t$  ab und werden mit der Mittelwertfunktion  $\mu_1, \mu_2, \dots, \mu_n$  erfasst. Mithilfe der Varianz wird bestimmt, wie stark eine Zufallsvariable  $Y_t$  von dem ermittelten Mittelwert abweicht. Ein weiteres Moment ist die Autokovarianzfunktion, um zu bestimmen ob und wie zwei Zufallsgrößen voneinander abhängen bzw. kann mithilfe der Autokorrelationsfunktion die serielle Abhängigkeit der Variablen beschrieben werden [18, S.1057]. Bei den genannten Momenten reicht es meist schon aus, die ersten beiden Momente der vorliegenden Zeitreihe zu schätzen, um Aussagen über den zugrunde liegenden stochastischen Prozess einer Zeitreihe machen zu können. Problematisch bei der Erfassung des Prozesses ist es, dass dieser, wie schon in der Einführung beschrieben, durch unterschiedliche Faktoren beeinflusst wird, welche nicht durch den Verlauf der Zeitreihe identifiziert werden können. Dabei ist es demnach unmöglich alle Einflüsse zu erfassen. Dem geschätzten grundlegenden Prozess wird daher eine Zufallsabweichung  $e_t$  hinzugefügt, die mit einem Erwartungswert von Null, einer konstanten Varianz und einer annähernd normalen Verteilung einhergeht [21, S.16].

### Stationäre Prozesse

Eine wichtige Klasse der stochastischen Prozesse sind die stationären Prozesse, welche ausschlaggebend für die Modellbildung der zu untersuchenden autoregressiven Modelle sind. Stationäre Prozesse werden zwischen einem streng stationären und einem schwach stationären Prozess unterschieden. Ein schwach stationärer Prozess liegt vor, wenn die folgenden drei Anforderungen bzw. Momente erfüllt sind:

$$\mu(t) = \mu, \quad (2.8)$$

$$\sigma^2(t) = \sigma^2, \quad (2.9)$$

$$\gamma(t_i, t_j) = \gamma(t_j - t_i). \quad (2.10)$$

Der Mittelwert und die Varianz sind dabei von der Zeit unabhängig. Ein stationärer Prozess weist keine Änderungen in seinem Niveau und weist eine Varianz  $\sigma$  auf [18, S.1053]. Die Autokorrelation ist dabei abhängig von dem zeitlichen Abstand zwischen  $j$  und  $i$ , ist jedoch unabhängig von der Zeit. Da der Mittelwert und die Varianz bei einem stationären Prozess von der Zeit unabhängig sind und einen konstanten Wert annehmen, wird die Modellbildung erleichtert. Da diese Momente für die gesamte Zeitreihe gelten, kann angenommen werden, dass dieser Verlauf auch auf den zukünftigen Verlauf übertragen werden kann. Die Wahrscheinlichkeitsverteilung ändert sich mit einer zeitlichen Verschiebung nicht [19, S.26]. Damit können die Vorhersagen der Zeitreihe erheblich verbessert werden.

Bei einem streng stationären stochastischen Prozess gilt, dass sich die Verteilung der Folge bei einer Verschiebung in der Zeit nicht ändert. Ein Beispiel für diesen Prozess ist das weiße Rauschen (engl. white noise), welches eines der wichtigsten Modellklassen eines stationären Prozesses bildet. Die Zufallsvariablen sind unabhängig und identisch voneinander verteilt und die Varianz ist zu jedem Zeitpunkt  $t$  konstant. In der Regel liegt der Mittelwert  $\mu$  bei Null. Für die meisten Modelle der Zeitreihenanalyse ist es schon ausreichend, wenn eine Zeitreihe mit einer schwachen Stationarität vorliegt.

In der Regel liegen in der Wirtschaft meist jedoch nichtstationäre Zeitreihen vor. Dabei ändert sich durch den Einfluss unterschiedlicher wirtschaftlicher Faktoren der Verlauf des Aktienkurses. Dabei entstehen langfristige Veränderungen im Niveau der Zeitreihe, wodurch die Modellbildung erschwert und Prognosen damit nur unzureichend sind [18, S.1053]. Eine nichtstationäre Zeitreihe kann dabei durch z.B. einen linearen Trendverlauf hervorgerufen werden. Da die meisten Methoden der Zeitreihenanalyse jedoch auf die Modellbildung von stationären Zeitreihen ausgelegt sind, ist es notwendig, nichtstationäre Zeitreihen in stationäre Zeitreihen zu transformieren. Dies wird in Abschnitt 2.2.4.2 näher erläutert.

### 2.2.3 Komponentenmodell der Zeitreihe

Neben einem linearen Trendverlauf kann der Verlauf einer Zeitreihe unter anderem durch eine Saisonale Komponente beeinflusst werden. Der Einfluss dieser Komponenten erschwert zusätzlich die Modellbildung für die Bestimmung von Prognosen. Die Zeitreihenanalyse hat dabei zum Inhalt, diese Komponenten zu identifizieren. Mögliche Komponenten werden mithilfe eines Komponentenmodells abgebildet, welches die Grundlage für die Analyse mit unterschiedlichen Methoden, wie z.B. dem

ARIMA-Modell bildet. Bei der Zeitreihenanalyse geht man davon aus, dass eine Zeitreihe in der Regel in vier Komponenten zerlegt werden kann, die den Verlauf der Zeitreihe bestimmen. Dabei wird entsprechend der genutzten Prognosemethode jede einzelne Komponente gesondert betrachtet, analysiert und entsprechende Prognosen erstellt. Zum Schluss werden diese wieder zusammengefügt. Das Ergebnis ist eine Prognose für die zukünftige Entwicklung der Zeitreihe.

$$\text{Additives Modell:} \quad y_t = m_t + k_t + s_t + u_t \quad (2.11)$$

Die standardmäßigen Komponenten sind dabei zum einen die Trendkomponente  $m_t$ , welche die langfristige Entwicklung des Prozesses beschreibt. Dabei handelt es sich um einen positiven oder negativen deterministischen Trend. Die Komponenten  $k_t$  und  $s_t$  sind zyklische Modellkomponenten, wobei die erste zyklische Schwankungen, wie z.B. eine Konjunkturschwankung beschreibt. Diese Komponente ist auch als Konjunkturkomponente bekannt. Die Trendkomponente und die Konjunkturkomponente werden meist zu einer "glatten Komponente"  $g_t$  zusammengefasst. Die zweite Komponente  $s_t$  hingegen beschreibt periodische Schwankungen, die jährlich auftreten. Dabei handelt es sich um die Saisonkomponente. Zum Schluss wird noch die Restkomponente  $u_t$  identifiziert, die den Rest beschreibt, der nicht durch andere Komponenten erklärt werden kann und zufällig ist [19, S.41 f.]. Zu beachten ist, dass eine Zeitreihe nicht alle Komponenten aufweist. So gibt es einige Zeitreihen, die nicht durch Konjunkturschwankungen oder saisonale Zyklen beeinflusst werden. Die einzelnen Komponenten werden durch unterschiedliche Verfahren bestimmt und analysiert.

## 2.2.4 Statistische Analysemethoden

Zur Abbildung einer Zeitreihe und Analyse des zugrunde liegenden stochastischen Prozesses, können unterschiedliche Modelle genutzt werden. Dabei werden in diesem Abschnitt die Lineare Regression und die autoregressiven Verfahren "autoregressive integrated moving average" (ARIMA) und das "saisonal autoregressive moving average" (SARIMA) vorgestellt.

### 2.2.4.1 Lineare Regression

Eine Möglichkeit, eine Zeitreihe mithilfe eines Modells abzubilden und damit entsprechende Prognosen zu treffen, ist das Lineare Regressionsmodell. Mithilfe der Regressionsanalyse wird dabei die Beziehung zweier Merkmale untersucht. Bei einer Zeitreihe wird dabei die Abhängigkeit einer abhängigen Variable (Regressand), z.B. der Schlusskurs einer Aktie, zur Zeit  $t$ , welches die unabhängige Variable (Regressor) darstellt, untersucht [16, S.60]. Dabei soll analysiert werden, wie sich der Regressand im Laufe der Zeit verändert bzw. soll damit auch geschätzt werden, wie sich der Regressand in der Zukunft verändern wird [16, S.67]. Weist eine Zeitreihe

einen linearen bzw. deterministischen Trend auf, kann dieser mithilfe der linearen Regression abgebildet werden [16, S.132]. Bei diesem Trend bleiben die Zuwächse bei jeder Zeiteinheit konstant.

Um den zugrunde liegenden Trend einer Zeitreihe abzubilden und damit die zukünftige Entwicklung prognostizieren zu können, werden mittels einer Regressionsgerade die beobachteten Werte der Zeitreihe bestmöglich approximiert [16, S.58]. Die Regressionsfunktion wird durch die lineare Funktion

$$\hat{y} = \alpha + \beta \cdot t \quad (t = 1, 2, \dots, n) \quad (2.12)$$

beschrieben, wobei die unbekannt Parameter bzw. Regressionskoeffizienten  $\alpha$  und  $\beta$  auf Basis der vorliegenden Zeitreihe optimal geschätzt werden, sodass die Abweichung der geschätzten Werte  $\hat{y}$  zu den Beobachtungen der Zeitreihe  $y$  möglichst gering sind [22, S.64]. Die Konstante  $\alpha$  gibt dabei als konstantes Glied den Schnittpunkt der Gerade mit der Ordinate an, wohingegen der Koeffizient  $\beta$  die Steigung bzw. die Neigung der Geraden angibt. Diese ist im linearen Trendmodell konstant.

Eine weit verbreitete Methode zur Schätzung der Parameter ist die Methode der kleinsten Quadrate. In der Regel bestehen Abweichungen, genannt Residuen  $e$ , vom geschätzten Wert der Geraden  $\hat{y}$  zum dazugehörigen beobachteten Wert der abhängigen Variable  $y$ . Die Abweichung wird durch  $e_t = y_t - \hat{y}_t$  bestimmt. Um ein geeignetes Trendmodell zu erstellen, müssen dabei die Abweichung der Geraden zu den Beobachtungswerten minimal sein. Dies kann mit der Methode der kleinsten Quadrate realisiert werden. Dabei sollen die Koeffizienten so geschätzt werden, dass die Summe der quadrierten Residuen minimal ist [16, S.70]. Damit erfolgt die Berechnung der Parameter von  $\alpha$  und  $\beta$  nach den folgenden Gleichungen nach [23, S.61-64]. Dabei wird nach Formel 2.14 die Bedingung der Methode der kleinsten Quadrate aufgestellt, die quadrierten Residuen zu minimieren.

$$Q = \sum_{t=1}^n e_t^2. \quad (2.13)$$

$$Q = \sum_{t=1}^n [y_t - (\alpha + \beta \cdot x_t)]^2 \rightarrow \min! \quad (2.14)$$

Für  $\alpha$  und  $\beta$  erhält man nach partieller Differentiation aus Formel 2.14 folgende Bedingungen:

$$\frac{\partial Q}{\partial \alpha} = \frac{\partial \sum (y_t - \alpha - \beta x_t)^2}{\partial \alpha} = 0 \quad (2.15)$$

$$\frac{\partial Q}{\partial \beta} = \frac{\partial \sum (y_t - \alpha - \beta x_t)^2}{\partial \beta} = 0. \quad (2.16)$$

Die Gleichungen 2.15 und 2.16 werden zu 2.17 und 2.18 umgestellt.

$$\begin{aligned}
 -\sum_{t=1}^n 2(y_i - \alpha - \beta x_i) &= 0 \\
 \alpha \sum_{t=1}^n 1 + \beta \sum_{t=1}^n x_i &= \sum_{t=1}^n y_i \\
 \alpha &= \frac{1}{n} \sum_{t=1}^n y_i - \beta \frac{1}{n} \sum_{t=1}^n x_i \\
 \alpha &= \bar{y} - \beta \bar{x} \\
 \bar{y} &= \alpha + \beta \bar{x}
 \end{aligned} \tag{2.17}$$

$$\begin{aligned}
 -\sum_{t=1}^n 2(y_i - \alpha - \beta x_i)x_i &= 0 \\
 \sum_{t=1}^n x_i y_i - \sum_{t=1}^n x_i \alpha - \sum_{t=1}^n \beta x_i^2 &= 0 \\
 \alpha \sum_{t=1}^n x_i + \beta \sum_{t=1}^n x_i^2 &= \sum_{t=1}^n x_i y_i
 \end{aligned} \tag{2.18}$$

Die Variablen  $\bar{y}$  und  $\bar{x}$  bilden dabei die Mittelwerte von  $y$  und  $x$ . Aus 2.17 und 2.18 entsteht ein Gleichungssystem mit den unbekanntem  $\alpha$  und  $\beta$ . Durch Umformung der beiden Gleichungen ergibt sich die Gleichungen zur Berechnung der Unbekannten mit

$$\beta = \frac{\sum_{t=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{t=1}^n (x_i - \bar{x})^2} \tag{2.19}$$

und durch Einsetzen von  $\beta$  ergibt sich für  $\alpha$ , wie schon in 2.18 aufgestellt

$$\alpha = \bar{y} - \beta \cdot \bar{x}. \tag{2.20}$$

Mithilfe der berechneten Parameter und der geschätzten Funktion können die Prognosen der Zeitreihe bestimmt werden. Die Prognose für einen bestimmten Zeitpunkt wird dabei durch die Angabe des gewünschten Zeitpunktes für  $x_t$  realisiert. Damit kann die Vorhersage eines Zeitpunktes durch  $\hat{y}(n+k) = \alpha + \beta \cdot (n+k)$  bestimmt werden.

### 2.2.4.2 Autoregressive Modelle

Eine weiterer Modellklasse der Regression sind die autoregressiven Modelle. Diese sind lineare Modelle, die zur Beschreibung von stochastischen Prozessen geeignet sind und vor allem zur Prognose von Zeitreihen genutzt werden. Ein bekanntes Modell zur Analyse und Vorhersage von Zeitreihen ist das Modell des autoregressiven gleitenden Durchschnitts, abgekürzt von der englischen Bezeichnung autoregressive moving average, auch bekannt als ARMA Modell. Dieser setzt sich aus einem autoregressiven AR(p)-Prozess und einem Prozess des gleitenden Durchschnitts, bekannt aus dem englischen als Moving Average, zusammen. Dieser ist als MA(q)-Prozess definiert. Beide Prozesse werden zunächst separat betrachtet.

#### AR(p) - Prozess

Bei einem AR(p)-Prozess besteht die Annahme, dass der aktuelle Wert  $y_t$  anhand der vergangenen  $p$  Werte einer Zeitreihe bestimmt werden kann, wobei  $p$  die Anzahl der betrachteten vergangenen Zeitschritte festlegt [24, S.84]. Damit basiert ein Wert  $y_t$  auf einer Linearkombination vergangener Werte desselben Prozesses. Ein autoregressiver Prozess (AR) der Ordnung  $p$  liegt vor, wenn für die Zeitreihe

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2.21)$$

gilt. Die Koeffizienten  $\phi_1, \dots, \phi_p$  stellen dabei die Gewichtungen der vergangenen Daten der Zeitreihe bzw. des stochastischen Prozesses dar. Dadurch wird festgelegt, welchen Einfluss die vergangenen Werte auf den geschätzten Wert von  $y_t$  haben, welcher als  $\hat{y}_t$  dargestellt wird. Der Fehlerterm  $\varepsilon_t$  meint die Abweichung zwischen dem prognostizierten Wert und dem realen Wert der Zeitreihe. Bei einem für die Abbildung der Zeitreihe geeigneten AR(p)-Modell handelt es sich bei dem Fehlertermen um den in Abschnitt 2.2.2 beschriebenen Prozess des weißen Rauschens mit einem Mittelwert von Null und einer konstanten Varianz  $\sigma_\varepsilon^2$ . Schwankt der AR(p)-Prozess jedoch um einen von Null verschiedenen Erwartungswert  $\mu$ , so gilt:

$$y_t - \mu = \phi_1 (y_{t-1} - \mu) + \phi_2 (y_{t-2} - \mu) + \dots + \phi_p (y_{t-p} - \mu) + \varepsilon_t \quad (2.22)$$

bzw.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2.23)$$

$$\text{mit} \quad c = \mu(1 - \phi_1 - \dots - \phi_p) \quad (2.24)$$

Meist wird ein AR(p) Prozess auch durch Einsetzen eines Verschiebungs-Operators oder auch Backshift- oder Lag-Operators  $B$  abgekürzt. Dieser beinhaltet die Verschiebung der Zeitreihe um eine Zeiteinheit in die Vergangenheit mit

$$B y_t = y_{t-1}. \quad (2.25)$$

Dabei ist auch eine mehrfache Anwendung des Verschiebungsoperators möglich. Dabei ergibt sich durch Einsetzen von  $B$  für das AR(p) Modell:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) y_t = \varepsilon_t \quad (2.26)$$

bzw.

$$\phi(B) y_t = \varepsilon_t. \quad (2.27)$$

Wie schon in Abschnitt 2.2.2 erläutert, basiert ein AR(p)-Modell auf der Annahme, dass es sich bei der vorliegenden Zeitreihe um eine stationäre Zeitreihe handelt. Um die Stationarität jedoch zu gewährleisten, müssen die Koeffizienten bzw. das charakteristische Polynom der linearen Abbildung einigen Bedingungen genügen. Das charakteristische Polynom ergibt sich dabei aus den Koeffizienten  $\phi_1, \dots, \phi_p$  mit  $\Phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$ . Mittels des Polynoms können die Eigenwerte der linearen Abbildung bestimmt werden, die die Eigenschaften des stochastischen Prozesses wiedergeben. Durch die errechneten Eigenwerte werden die Nullstellen des Gleichungssystems bestimmt. Damit die Stationarität des Prozesses gewährleistet werden kann, darf das charakteristische Polynom dabei keine Einheitswurzel aufweisen. Die Nullstellen, die durch die Koeffizienten ermittelt werden, dürfen damit nicht bei eins, d.h. auf dem Einheitskreis liegen [19, S.80 ff.]. Sollte dies jedoch zutreffen, handelt es sich um eine nichtstationäre Zeitreihe mit einer Einheitswurzel. Ein weit verbreiteter nichtstationärer Prozess, ist der Prozess des zufälligen Wanderns, auch Random Walk genannt. Dieser stellt eine besondere Form eines autoregressiven Prozesses erster Ordnung dar. Dieser AR(1)-Prozess ist in Formel 2.28 aufgestellt.

$$y_t = \phi y_{t-1} + \varepsilon_t \quad (2.28)$$

Bei dieser Form eines AR(1)-Prozesses ist der Wert  $y_t$  nur von dem vergangenen Wert  $y_{t-1}$  und der entsprechenden Gewichtung  $\phi$ , sowie dem Rauschterm  $\varepsilon_t$  abhängig. Bei einem Random Walk gilt dabei für  $\phi = 1$ . Der vorliegende Prozess enthält damit eine Nullstelle bei eins und ist damit nichtstationär. Dieser Prozess wird auch stochastischer Trend genannt. Anders kann der Random Walk auch als

$$y_t = y_0 + \sum_{t=1}^n \varepsilon_t \quad (2.29)$$

dargestellt werden. Das bedeutet, dass jeder weitere Wert von Zufallswerten abhängt. Eine Erweiterung dieses Prozesses ist der Random Walk mit Drift. Der Drift beschreibt dabei eine Konstante  $c$ , um die der Prozess erweitert wird. Die Konstante bildet damit einen grundlegenden Trend, um den sich die Zeitreihe entwickelt.

$$Y_t = c + \Phi Y_{t-1} + \varepsilon_t. \quad (2.30)$$

**MA(q)-Prozesse**

Wie schon zuvor beschrieben, besteht ein ARMA-Modell jedoch, neben einem autoregressiven Prozess, ebenfalls aus einem Prozess des gleitenden Durchschnitts. Dieser wird hierbei als MA(q)-Prozess bezeichnet. Dieser ist nach Formel 2.31 definiert.

$$Y_t = \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q}. \quad (2.31)$$

Bei  $\varepsilon$  handelt es sich wieder um ein weißes Rauschen mit einem Mittelwert von Null und einer Varianz  $\sigma_t^2$  [20, S.29]. Der Fehlerterm  $\varepsilon$  wird dabei durch  $\varepsilon_t = y_{t-1} - \hat{y}_{t-1}$  berechnet. Bei einem MA(1)-Prozess wird damit die Abweichung der Schätzung  $\hat{y}$  zu  $y$  zum Zeitpunkt  $t - 1$  berücksichtigt. Durch  $\theta$  wird dabei festgelegt, mit welcher Gewichtung die vergangene Abweichung für die Berechnung von  $\hat{y}_t$  bei einem ARMA(p,q)-Prozess berücksichtigt wird.

Durch Einsetzen des Verschiebungsoperators ergibt sich für das Modell:

$$Y_t = \theta(B)\varepsilon_t. \quad (2.32)$$

Ein MA(q)-Prozess ist immer schwach stationär [24, S.90].

**ARMA(p,q) - Modell**

Ein aus einem AR(p) und MA(q)-Prozess zusammengesetzter ARMA(p,q) Prozess liegt vor, wenn gilt:

$$y_t = \phi_1y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q} \quad (2.33)$$

Wie schon zuvor erläutert, definieren die Ordnungen p des autoregressiven und q des Prozesses des gleitenden Durchschnitts, welche vergangenen Werte zur Schätzung von  $y_t$  berücksichtigt werden müssen. Mit dem Verschiebungsoperator ergibt sich hierfür

$$\phi(B)y_t = \theta(B)\varepsilon_t. \quad (2.34)$$

Damit ein für die zu prognostizierende Zeitreihe geeignetes ARMA(p,q)-Modell identifiziert werden kann, sollte diese Zeitreihe die Bedingung der Stationarität erfüllen. In der Wirtschaft ist dies meist jedoch nicht der Fall. Die Zeitreihen werden, wie schon zuvor erläutert, durch unterschiedliche Komponenten, wie z.B. einem Trend oder Saisonalitäten beeinflusst. Damit ist eine Transformation für diese Form der Zeitreihen zunächst notwendig, bevor ein geeignetes ARMA(p,q)-Modell zur Abbildung der Zeitreihe ermittelt werden kann. Diesen Fall haben auch Box- und Jenkins in ihrem "autoregressive integrated moving average" (ARIMA(p,d,q)) Modell berücksichtigt.

**ARIMA(p,d,q)-Modell**

Aufbauend auf dem ARMA(p,q)-Modell, beinhaltet ein ARIMA(p,d,q)-Modell die



Transformation der nichtstationären Zeitreihen, bei der das charakteristische Polynom  $\phi$  eine Nullstelle bei  $(1) = 0$  enthält, in eine stationäre Zeitreihe. Diese Transformation erfolgt mithilfe der Differenzbildung der Zeitreihe. Sollte ein ARIMA(p,d,q)-Modell Anwendung zur Abbildung einer Zeitreihe finden, wird deutlich, dass diese einen integrierten Prozess enthält. Dieser Prozess meint den zuvor beschriebenen stochastischen Trend, welcher innerhalb der Zeitreihe vorliegt und nur durch die Differenzbildung aus der Zeitreihe eliminiert werden kann, sodass ein für die Zeitreihe geeignetes ARMA(p,q)-Modell identifiziert werden kann.

Der Grad der Differenzbildung einer Zeitreihe wird durch die Ordnung  $d$  bestimmt [20]. Eine Differenzbildung der ersten Ordnung meint dabei die Berechnung der Differenz benachbarter Werte. Dies kann beschrieben werden durch

$$\Delta y_t = y_t - y_{t-1} \quad (t = 1, 2, \dots, T) \quad (2.35)$$

bzw. ergibt sich durch einsetzen des Verschiebungsoperators die Gleichung

$$y_t - y_{t-1} = (1 - B)y_t. \quad (2.36)$$

Dabei wird ein Prozess gebildet, der schon nach der ersten Differenzbildung stationär sein kann. Es ist jedoch möglich, dass eine mehrfache Differenzbildung notwendig ist. Daraus ergibt sich mit Einsetzen des Verschiebungsoperators nach Formel 2.36 der Prozess

$$\Delta^d y_t = (1 - B)^d y_t. \quad (2.37)$$

Durch die Zusammensetzung der Differenzierung, des AR(p)- und MA(q)-Prozesses ergibt sich daraus ein ARIMA(p,d,q) Modell für Prozesse mit einem stochastischen Trend. Dieser ist nach [20] definiert als:

$$\Phi(B)(1 - B)^d y_t = \theta(B)\varepsilon_t \quad (2.38)$$

Nachdem die Differenzbildung und damit die Eliminierung des stochastischen Trends der Zeitreihe durchgeführt wurde, kann die Bestimmung eines geeigneten ARMA(p,q)-Modells nach Formel 2.33 erfolgen.

### Parameterschätzung

Damit ein für die vorliegende Zeitreihe geeignetes ARIMA(p,d,q)-Modell identifiziert werden kann, müssen dabei zunächst die Ordnungen von  $p$ ,  $q$  und  $d$  des AR(p), MA(q) und der Grad der Differenzbildung geschätzt werden. Diese werden nach George E.P. Box und Gwilym M. Jenkins benannten Box-Jenkins-Verfahren ermittelt.

Im ersten Schritt wird dabei die Ordnung von  $d$  ermittelt bzw. muss in diesem Zusammenhang zunächst identifiziert werden, ob die vorliegende Zeitreihe stationär ist oder zunächst transformiert werden muss. Hierzu wird ein Hypothesentest hinzugezogen, welcher auf diese Fälle prüft. Die bekanntesten Testverfahren

sind dabei das Kwiatkowski–Phillips–Schmidt–Shin (KPSS) Testverfahren [25], der Augmented-Dickey Fuller Test (ADF) [26] oder der Phillips-Perron Test [27]. Für diese Ausarbeitung wird dabei der KPSS Test hinzugezogen. Bei den Testverfahren handelt es sich um Hypothesentests, die jeweils die Zeitreihe hinsichtlich einer vorliegenden Einheitswurzel prüfen. Die Nullhypothese  $H_0$  des KPSS-Tests besagt dabei, dass es sich bei der vorliegenden Zeitreihe um eine Trendstationäre Zeitreihe handelt, wohingegen die Alternativhypothese  $H_1$  annimmt, dass eine Einheitswurzel in dem stochastischen Prozess vorliegt. Bei einem KPSS-Test besteht die Annahme, dass die vorliegende Zeitreihe als Summe eines deterministischen Trends, eines zufälligen Prozesses und einem stationären Fehler nach  $y(t) = r_t(\alpha) + \beta(t) + \varepsilon_t$  aufgefasst werden kann. Sollte die Hypothese  $H_0$  gelten, so wird angenommen, dass für die Varianz des zufälligen Wanderns  $\sigma = 0$  gilt. Damit würde eine stationäre Zeitreihe vorliegen. Diese Aussage wird unter anderem mithilfe unterschiedlicher Teststatistiken überprüft. Für eine ausführliche Beschreibung des Hypothesentests wird auf die Literatur [25] verwiesen.

Sobald die Ordnung von  $d$  bestimmt wurde, erfolgt die Identifizierung der Ordnungen für  $p$  und  $q$  des  $AR(p)$  bzw. des  $MA(q)$ -Prozesses, sowie die Schätzung der entsprechenden Koeffizienten  $\phi_1, \dots, \phi_p$  und  $\theta_1, \dots, \theta_q$ . Bekannte Methoden, die zur Schätzung dieser Parameter eingesetzt werden, ist zum einen die Methode der kleinsten Quadrate, welche schon im Rahmen der Linearen Regression beschrieben wurde. Zum anderen wird hierfür die Maximum-Likelihood-Methode eingesetzt [19, S.102 ff.]. Mithilfe dieser Methoden werden die unbekannt Parameter so gewählt, dass die vorliegende Zeitreihe bestmöglich abgebildet werden kann.

### **SARIMA-Modell**

In der Regel ist es zusätzlich möglich, dass Zeitreihen ein saisonales Muster aufweisen und damit von einer saisonalen Komponente beeinflusst werden. Wie in Abschnitt 2.2.3 aufgeführt, ist es sinnvoll, die vorliegende Zeitreihe in seine Komponenten aufzuteilen und diese separat zu analysieren. Damit kann auf Grundlage der Ergebnisse die vorliegende Zeitreihe bestmöglich abgebildet werden. Dies ist mittels eines erweiterten  $ARIMA(p,d,q)$ -Modells, dem  $ARIMA(p,d,q)(P,D,Q)_m$ -Modell möglich. Dieses wird auch als saisonales  $ARIMA$  bzw. ( $SARIMA$ )-Modell bezeichnet. Dabei repräsentieren die Parameter  $p,d$  und  $q$ , wieder die Ordnungen des  $ARIMA(p,d,q)$ -Modells der von der saison bereinigten Zeitreihe. Die Parameter  $P,D$  und  $Q$  legen die Ordnungen des  $ARIMA(P,D,Q)$ -Modells für die Zeitreihe der saisonalen Komponente fest. Alle Parameter werden jeweils so gewählt, dass die saisonale Zeitreihe als auch die von der saisonalen Komponente bereinigte Zeitreihe bestmöglich abgebildet werden kann. Der Parameter  $m$  legt dabei die Frequenz bzw. die Periode fest. Damit wird das Muster innerhalb der Periode identifiziert, welches sich über mehrer Perioden wiederholt. Ein  $SARIMA$ -Prozess liegt vor, wenn unter Verwendung des Verschiebungsoperators

$$\phi_p(B)\Phi_p(B^s)(1-B)^d(1-B^s)^D y_t = \theta_q(B)\Theta_Q(B^l)\varepsilon_t \quad (2.39)$$

gilt [19, S.136 ff]. Bei einem SARIMA-Modell wird zur Bestimmung der Parameter zunächst die vorliegende Zeitreihe in diese beiden Komponenten aufgeteilt bzw. wird dabei die vorliegende Zeitreihe von der saisonalen Komponente bzw. von dem saisonalen Einfluss zunächst bereinigt. Das Ergebnis sind die bereinigte Zeitreihe sowie eine Zeitreihe der saisonalen Komponente. Diese Aufteilung erfolgt meist mithilfe gleitender Durchschnitte. Dabei wird jeder Wert  $y_t$  aus einem Durchschnittswert der vergangenen  $m$  Werte bestimmt. Beide Zeitreihen werden dabei mit unterschiedlichen Methoden analysiert. Die Modellbildung eines SARIMA-Modells erfolgt mithilfe unterschiedlicher Methoden und Verfahren. Sobald beide Zeitreihen analysiert wurden, erfolgt die Prognose auf Grundlage des wie in Abschnitt 2.2.3 beschriebene Zusammenfügens beider Komponenten. Die Vorgehensweise zur Datenerhebung in dieser Ausarbeitung wird im weiteren Verlauf der Arbeit am Beispiel eines Aktienkurses näher erläutert.

## 2.3 Künstliche Neuronale Netze

Durch den Einsatz der künstlichen neuronalen Netze ist es möglich, komplexe Modelle und Prozesse abzubilden und zu simulieren. Dabei wird in diesem Abschnitt zunächst der Aufbau des künstlichen neuronalen Netzes und dessen mathematische Abbildung aufgestellt. Auf Grundlage dessen, wird die für die Ausarbeitung verwendete komplexere Form der neuronalen Netze, das rekurrenten neuronalen Netz, näher erläutert. Abschließend wird das Predictive Neural Network als ausschlaggebendes Verfahren für diese Arbeit beschrieben.

### 2.3.1 Aufbau Neuronaler Netze

Nach dem Abbild der Informationsverarbeitung im Gehirn mittels der Neuronen sind künstliche neuronale Netze (KNN) aufgebaut. Für eine Erläuterung zu dem Aufbau des neuronalen Netzes des Gehirns wird in diesem Zusammenhang auf die Literatur [28] verwiesen.

Die KNN werden durch einen gerichteten Graphen dargestellt, der aus einer Menge von Knoten und Kanten bestehen. Die Knoten repräsentieren dabei die Neuronen eines neuronalen Netzes, wohingegen die Kanten die Verbindungen der Neuronen zueinander abbilden. Ein Beispiel für ein künstliches neuronales Netz ist in vereinfachter Form in Abbildung 2.2 dargestellt.

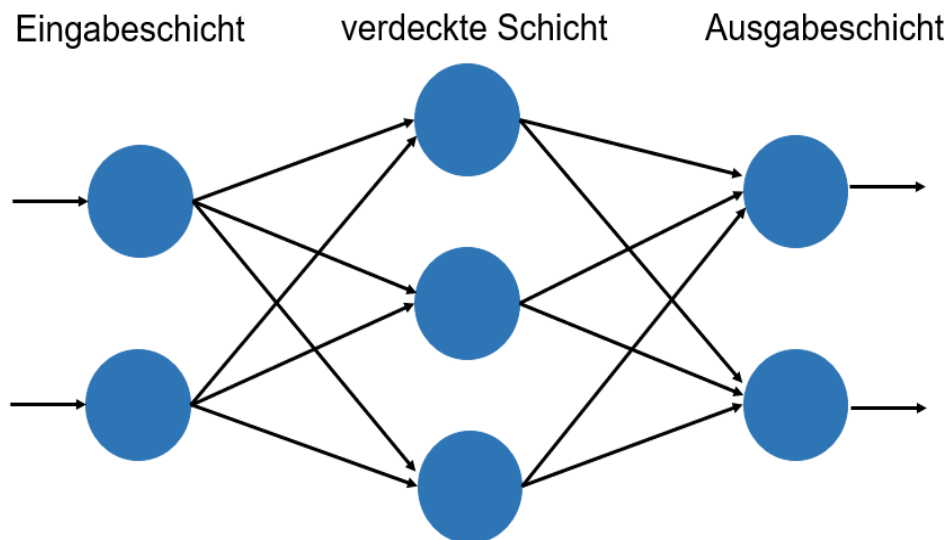


Abbildung 2.2: Vorwärtsgerichtetes KNN, Quelle: Eigene Darstellung

Bei einem künstlichen neuronalen Netz können die Neuronen in drei unterschiedliche Klassen unterteilt werden. Zum einen werden bestimmte Neuronen als Eingabeneuronen definiert. Die Eingabeneuronen werden dabei als Eingabeschicht zusammengefasst. Bei einem künstlichen neuronalen Netz nehmen diese einen bestimmten

Wert an. Zum anderen können die Neuronen neben der Eingabeschicht einer verdeckten Schicht zugeordnet werden. Diese sind für die Informationsverarbeitung der Eingaben zuständig. Diese Schicht ist nicht mit der Umgebung verbunden, sondern befindet sich zwischen der Eingabe- und der Ausgabeschicht eines Netzes. Die Neuronen der verdeckten Schicht sammeln dabei die entsprechenden Eingangssignale und werten diese durch eine Aktivierungsfunktion aus. Der Aktivierungswert wird anschließend an die entsprechenden Ausgabeneuronen übertragen. Die Verbindungen zwischen den Neuronen der einzelnen Schichten werden durch Kanten im künstlichen neuronalen Netz dargestellt. Jede Verbindung ist dabei gewichtet, wodurch eine Bindung zwischen zwei Neuronen entsprechend der Gewichte gestärkt oder geschwächt werden kann.

Das neuronale Netz, welches in Abbildung 2.2 aufgeführt ist, zeigt ein mehrschichtiges neuronales Netz mit einem vorwärtsgerichteten Informationsfluss. Dieser wird daher aus dem englischen auch als Feed-Forward Netzwerk bezeichnet, bei dem die Eingangssignale über die Neuronen der verdeckten Schicht hin zur Ausgabeschicht verarbeitet werden. Der Aufbau der Neuronalen Netze variiert jedoch nach Problemstellung, wie zum Beispiel bei der Vorhersage von Zeitreihen bzw. Aktienkursen, wobei die vergangenen Werte der Reihe gefordert sind. Hierzu eignen sich vor allem neuronale Netze, die vergangene Werte bei der Berechnung neuer Ausgaben berücksichtigen. Ein bekanntes neuronales Netz ist hierbei das rekurrente neuronale Netz, welches durch die Berücksichtigung vergangener Werte einem Netz mit einem Gedächtnis entspricht.

### 2.3.2 Mathematische Darstellung eines Neuronalen Netzes

Das mathematische Modell eines künstlichen neuronalen Netzes wird für die Modellierung des Netzes mithilfe des Computer benötigt, sodass komplexe Prozesse dadurch simuliert werden können. Dabei entspricht ein Neuron in der mathematischen Darstellung einer Variable, die einen bestimmten Wert annehmen kann. Die Neuronen einer bestimmten Schicht werden meist zu einem Vektor zusammengefasst. Damit ergibt sich beispielsweise für die Eingangsneuronen  $x_1, \dots, x_n$  der Vektor  $X = (x_1, \dots, x_n)$  der Eingabeschicht womit dessen Zustände beschrieben werden [29, S.141]. Jede Verbindung von einem Neuron zu einem anderen wird dabei mit einem Gewicht  $w$  multipliziert. Diese Gewichte werden dabei in einer Adjazenzmatrix dargestellt, die die topologische Struktur des Neuronalen Netzes beschreibt. Dabei ergibt sich für ein künstliches neuronales Netz eine Gewichtsmatrix der Form

$$W = \begin{pmatrix} w_{1,1} & \dots & w_{1,m} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \dots & w_{n,m} \end{pmatrix}. \quad (2.40)$$

Die Indizes  $i = 1, \dots, n$  und  $j = 1, \dots, m$  geben dabei die Neuronen an, zwischen denen die jeweilige gewichtete Verbindung besteht. In der Gewichtsmatrix werden die Ver-

bindungsgewichte von Neuron  $j$  nach Neuron  $i$  gelesen. Hierbei wird angenommen, dass die Neuronen Schichtenweise, d.h. von der Eingabeschicht über die verdeckte Schicht hin zur Ausgabeschicht, nummeriert sind. Besteht zwischen zwei Neuronen keine Verbindung, so wird diese mit Null gewichtet [29, S.141]. Die Gewichte der direkten Rückkopplungen der Neuronen befinden sich dabei auf der Hauptdiagonalen.

Neben dem Zustandsvektor der Eingaben  $\mathbf{x} = (x_1, \dots, x_n)$  werden mithilfe von Transformationsfunktionen die Zustände der Neuronen der verdeckten Schicht und die der Ausgabeneuronen berechnet. Unterschieden wird dabei zwischen der Propagierungsfunktion, der Aktivierungsfunktion und der Ausgabefunktion. Die erste Funktion berechnet dabei die Netzeingabe  $net_j$  des künstlichen neuronalen Netzes. Meist handelt es sich um eine Aufsummierungsfunktion bei der die Eingangswerte  $x_j$  mit den jeweiligen Gewichten  $w_{i,j}$  multipliziert und aufsummiert werden [16, S.581]. Durch die berechnete Netzeingabe und einer Aktivierungsfunktion wird der Aktivierungszustand eines Neurons  $x_i$  durch

$$x_i = f\left(\sum_{j=1}^n w_{ij}x_j\right) \quad (2.41)$$

bestimmt [30, S.269]. Damit wird festgelegt, wie stark die Werte der Vorgängerneuronen auf den Aktivierungszustand eines Neurons  $x_i$  wirken. Eine mögliche Aktivierungsfunktion sind die sigmoide Funktionen, wie z.B. die logistische Funktion (Formel 2.42)

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.42)$$

[29, S.151]. Eine weitere Möglichkeit ist die Verwendung der Identitätsfunktion für die  $f(x) = x$  gilt. Dabei handelt es sich um eine lineare Aktivierungsfunktion, wobei die berechneten Aktivierungswerte immer weiter anwachsen können, da keine Beschränkungen vorliegen. Daher wird diese Funktion eher selten als Aktivierungsfunktion, dabei aber meist neben der Schwellwertfunktion als Ausgabefunktion eingesetzt [30, S.269].

### 2.3.3 Lernen Neuronaler Netze

Das Lernen innerhalb des künstlichen neuronalen Netzes erfolgt mittels der Verbindungen der einzelnen Neuronen. Durch die entsprechende Gewichtung der Verbindung zwischen zwei Neuronen innerhalb des künstlichen neuronalen Netzes, können diese damit gestärkt oder aber auch geschwächt werden. Aufgrund dessen, wird der Einfluss, den der jeweilige Neuronenwert auf die Ausgabe des neuronalen Netzes

haben soll, festgelegt. Die Gewichte werden dabei mittels unterschiedlicher Lernverfahren bestimmt.

Ein weit verbreitetes Lernverfahren, welches auf Grundlage des Lernen im menschlichen Gehirn definiert wurde, ist das durch D. Hebb (1949) definierte Lernverfahren und bildet die Grundlage des Lernvorganges der künstlichen Neuronalen Netze. Mithilfe der Hebb'schen Lernregel, werden die Gewichte einer Verbindung durch eine Lernrate angepasst. Häufig werden neben den Gewichten auch weitere Parameter, wie z.B. diese Lernrate selbst erlernt und verändert. Mathematisch ausgedrückt wird nach der Hebb'schen Regel bei künstlichen neuronalen Netzen die Veränderung des Gewichtes  $w_{ij}$  durch

$$\Delta w_{ij} = \eta x_i x_j \quad (2.43)$$

berechnet. Die Veränderung  $\Delta w_{ij}$  ergibt sich dabei aus der konstanten Lernrate  $\eta$ , dem aktivierten Wert  $x_i$  und dem Aktivierungswert des Neurons  $x_j$ . Für die Lernrate gilt dabei meist  $0 < \eta < 1$ , wobei  $\eta$  entsprechend der Problemstellung gewählt wird. Dieses Lernverfahren bildet die Basis für viele weitere modifizierte Lernmethoden. Generell wird zwischen dem überwachten und dem unüberwachten Lernen unterschieden mit dem ein künstliches neuronales Netz trainiert werden kann.

Der Unterschied zwischen beiden Lernformen ist, dass bei dem überwachten Lernen eine gewünschte Ausgabe von Anfang an bekannt ist. Diese ist in Form eines Zielvektors gegeben. Dem vorliegenden neuronalen Netz werden zu Beginn Eingabedaten über die Eingabeneuronen übergeben. Diese werden mithilfe der einzelnen Transformationsfunktionen und der Schichten wie in Abschnitt 1.3.3 beschrieben verarbeitet und eine Ausgabe berechnet. Diese Ausgabe wird mit dem gewünschten Ausgabe verglichen. Dabei soll die Differenz bzw. die Abweichung zwischen dem gewünschten Output und der von dem neuronalen Netz errechneten Ausgabe minimiert werden [31, S.128].

Im Gegensatz zum überwachten Lernen bestehen beim unüberwachten Lernen keine vorliegenden Zielvorgaben bzw. Zielwerte. Dabei geht es um die Selbstorganisation des Netzes ohne eine Rückmeldung durch Zielvorgaben zu erhalten [32, S.122]. Eine mögliche Form der Selbstorganisation ist das Gruppieren von Daten. Diese Form wird auch nach dem Englischen Begriff als Cluster bezeichnet. Eingaben, die sich ähneln, werden zu Gruppen zusammengefasst. Damit werden für die Daten einer Gruppe die gleichen Ausgabewerte geliefert. Die Ähnlichkeit wird dabei meist mit einer Abstandsfunktion, wie z.B. der euklidischen Abstandsfunktion bestimmt [33, S.40f.].

### 2.3.4 Rekurrente Neuronale Netze

Die rekurrenten neuronalen Netze unterscheiden sich zu den vorwärtsgerichteten neuronalen Netzen vor allem durch die Verknüpfungen der Neuronen innerhalb des

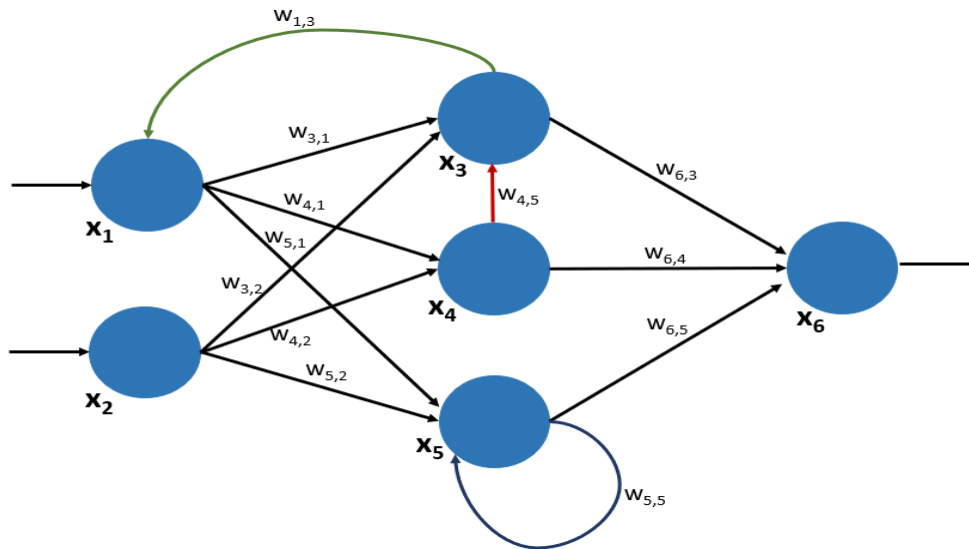


Abbildung 2.3: Rekurrente Verbindungen, Quelle: Eigene Darstellung

Netzwerks. So besteht bei den vorwärtsgerichteten Neuronalen Netzen lediglich Verbindungen von der Eingabeschicht über gegebenenfalls die verdeckte Schicht, hin zur Ausgabeschicht. Rekurrente Netze hingegen enthalten Zyklen, wodurch prinzipiell Verbindungen zwischen allen Neuronen der Schichten bestehen können [31, S.140]. Die Rückkopplung kann dabei über verschiedene Schichten laufen. Wie in Abbildung 2.3 dargestellt, bestehen drei mögliche Formen der rekurrenten Verbindungen. Zum einen ist eine direkte Rückkopplung möglich, bei der der Ausgang eines Neurons wieder als Eingang impliziert wird. Indirekte Rückkopplungen verlaufen über eine Schicht, wobei der Ausgang eines Neurons einer Schicht als Eingangswert eines Neurons einer vorherigen Schicht wirkt. Ergänzend ist die seitliche Rückführung möglich, die über Neuronen derselben Schicht verlaufen.

Rekurrente Neuronale Netze eignen sich vor allem zur Modellierung von Sequenzen mit einer zeitlichen geordneten Folge von Werten. Durch den rekurrenten Aufbau eines Netzes kann der zeitliche Aspekt damit berücksichtigt werden. Die Ausgabe eines Neurons zu einem bestimmten Zeitpunkt wird als Input eines anderen Neurons zum nachfolgenden Zeitpunkt zugeführt, sodass wichtige Informationen eines früheren Zustands einer Sequenz nicht verloren gehen. Aufgrund dieser Architektur wird dem Neuronalen Netz somit ein Gedächtnis hinzugefügt.

Dabei bestehen unterschiedliche Ausprägungen und Ansätze zur Modellierung eines rekurrenten Neuronalen Netzes. Bekannte Netze sind unter anderem das Hopfield Netz, welches ein einschichtiges rekurrentes neuronales Netz darstellt, bei der die Neuronen als Eingabe- und als Ausgabeneuronen genutzt werden. Es bestehen Verbindungen zwischen allen Neuronen, wobei keine direkten Rückkopplungen möglich sind [34]. Eine weitere Möglichkeit, rekurrente Neuronale Netze zu modellieren, sind



mithilfe von Elman Netzen, bei denen neben den Neuronen der Eingabe-, der Ausgabe und der verdeckten Schicht sogenannte Kontextneuronen in der Eingabeschicht ergänzt werden. Die Ausgaben der verdeckten Schicht werden durch die Rückführungen zu den Kontextneuronen in der Kontextschicht gespeichert. Somit stehen dem neuronalen Netz dies zuvor berechneten Werte neben den neuen Eingabewerten ebenfalls als Eingabe zur Verfügung. Die Kontextneuronen bilden damit das Gedächtnis des Netzes [35]. Moderne Ansätze der rekurrenten Neuronalen Netze implizieren die Darstellung der verdeckten Schichten als Reservoir.

### 2.3.4.1 Echo State Network

Ein neuer Ansatz für rekurrente neuronale Netze stellt H. Jäger mit dem Echo State Network vor (ESN) [2]. Wie bei einem klassischen mehrschichtigen neuronalen Netz, besteht das ESN wie in Abbildung 2.4 dargestellt, aus einer Eingabeschicht, einer verdeckten Schicht und einer Ausgabeschicht. Die verdeckte Schicht wird nach Jäger jedoch als dynamisches Reservoir verstanden.

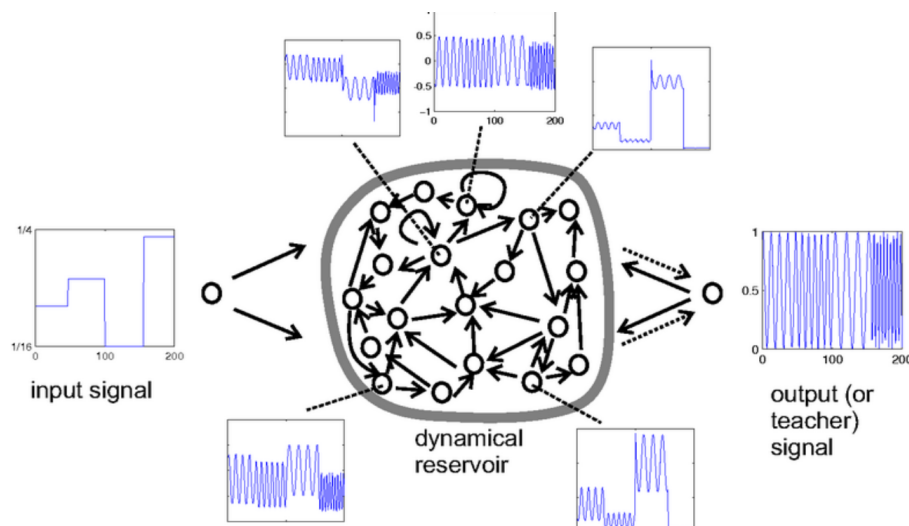


Abbildung 2.4: Echo state network [2]

Die Größe des Reservoirs ist dabei aufgabenabhängig und beinhaltet  $N$  nicht linear aktivierte Neuronen, die alle miteinander verbunden sind und mit zufällig gewählten Werten initialisiert werden. Diese werden in einem  $N$ -dimensionalen Zustandsvektor  $x(n)$  zusammengefasst. Die Gewichte zwischen den Neuronen werden zufällig festgelegt und nicht mehr verändert. Das Reservoir wird durch  $K$  Eingangsneuronen bzw. Eingangssignale angesteuert, die ebenfalls mit allen Neuronen des Reservoirs durch zufällig gewählte Verbindungsgewichte verbunden sind. In einem  $K$ -dimensionalen Vektor  $u(n)$  wird der Zustand der Eingabeneuronen zum Zeitpunkt  $n$  erfasst. Aus dem Reservoir gehen Signale an die Neuronen der Ausgabeschicht hervor. Für die

Neuronen der Ausgabeschicht ergibt sich ein Vektor  $y(n)$  mit  $L$  Dimensionen. Möglich ist eine Rückführung eines Signals der Ausgabeschicht hin zum Reservoir, die jedoch optional ist. Die Verbindungsgewichte zwischen den Neuronen werden in Gewichtsmatrizen dargestellt. Damit ergibt sich für die Eingabeschicht eine  $K \times N$  Gewichtsmatrix  $W^{in}$ . Die Gewichte innerhalb des Reservoirs werden durch die Matrix  $W^{res}$  der Größe  $N \times N$  erfasst. Für die Ausgabe wiederum ergibt sich die Gewichtsmatrix  $W^{out}$  mit  $L \times (K + N)$  Gewichten. Bei einem ESN werden nur die Ausgabegewichte gelernt. Sollte eine Rückkopplung der Ausgabeschicht zum Reservoir vorliegen, werden die bestehenden Gewichtsmatrizen durch die Gewichtsmatrix  $W^{fb}$  der Größe  $N \times L$  ergänzt. Sollte das neuronale Netz keine Verbindung von der Ausgabeschicht hin zum Reservoir aufweisen, so gilt  $W^{fb} = 0$ .

Sobald das neuronale Netz initialisiert wird, kann mithilfe der Gewichtsmatrizen und der Eingabewerte zum Zeitpunkt  $n + 1$ , sowie dem aktuellen Zustand des Reservoirs der entsprechende Reservoirzustand  $n + 1$  nach

$$x(n + 1) = f(W^{res}x(n) + W^{in}u(n + 1) + W^{fb}y(n)) \quad (2.44)$$

bestimmt werden. Diese Zustandsaktualisierung wird durch eine sigmoidale Funktion  $f$ , wie zum Beispiel der logistischen Funktion nach Formel 2.42 berechnet. Um eine Ausgabe für den Zeitpunkt  $n$  des ESN berechnen zu können, wird zunächst der erweiterte Systemstatus  $z(n)$  nach

$$z(n) = \begin{pmatrix} x(n) \\ u(n) \end{pmatrix} \quad (2.45)$$

durch die Verkettung der Eingangs- und Reservoirzustände erstellt.

Unter der Verwendung einer Aktivierungsfunktion, wie der Identitätsfunktion oder einer sigmoiden Funktion, der Ausgabegewichte  $W^{out}$  und des erweiterten Zustands  $z_n$  wird nach

$$y(n) = g(W^{out} \cdot z(n)) \quad (2.46)$$

die Ausgabe  $y(n)$  des Netzes berechnet. Für die Berechnung der Ausgabe müssen jedoch zunächst die Ausgabegewichte  $W^{out}$  gelernt werden.

Soll dem rekurrenten neuronalen Netz eine Eingabe mit bestimmte Daten zugeführt werde, ergibt für das Echo State Network die entsprechende Eingabesequenz  $u(1), \dots, u(n_{max})$  zu den Zeitpunkten  $n = 1, \dots, n_{max}$ . Nach Formel 2.45 führt dies zu  $z(1), \dots, z(n_{max})$  Systemzuständen des neuronalen Netzes, die zeilenweise in einer

Matrix  $Z$  mit einer Größe von  $n_{max} \times (N + K)$  zusammengefasst werden. Dabei werden asymptotisch die zufällig bestimmten Zustandswerte bei der Initialisierung des Netzes eliminiert, da diese für die Berechnung der Ausgaben nicht erwünscht sind. Die Reservoirzustände sollen dabei lediglich aus den Eingaben  $u(1), \dots, u(n_{max})$  bestimmt werden. Dies wird als Echo State Eigenschaft (Echo state property) bezeichnet. Um das ESP zu garantieren, muss die Gewichtsmatrix des Reservoirs bestimmten Bedingungen genügen. Diese Bedingung folgt empirischen Untersuchungen, mit der belegt werden kann, dass für jede Eingabe das ESP erfüllt ist, sobald die Gewichtsmatrix  $W^{res}$  ein Spektralradius kleiner als eins aufweist.

Wie auch für die Systemzustände werden die Ausgaben  $d(n)$  zeilenweise in einer Ausgabematrix  $D$  der Größe  $n_{max} \times L$  erfasst. Sollten rückgekoppelte Verbindungen von der Ausgabeschicht zum Reservoir bestehen, so wird ein Teil der Daten  $d(n)$  in die Ausgabeneuronen eingefügt, die entsprechend der Eingaben die gewünschten Ausgaben bilden (teacher forcing). Sollte keine Rückkopplung bestehen, werden die zugehörigen gewünschten Ausgabewerte (teacher values)  $d(n)$  der Eingabewerte  $u(n)$  zeilenweise in einer Lehrerausgabematrix  $D$  der Größe  $n_{max} \times L$  sortiert. Damit können die Ausgabegewichte nach

$$W_{out} = (Z^{-1}D)^T \quad (2.47)$$

gelernt werden. Mit der durch die Ausgabegewichte  $W^{out}$  gelernte Linearkombination der Reservoirneuronen können somit die Ausgaben des ESN berechnet werden.

#### 2.3.4.2 Predictive Neural Network

Ein weiteres rekurrentes neuronales Netz wird von Stolzenburg et al.[7] vorgestellt, bei dem es sich um das vorhersagende neuronale Netz, das Predictive Neural Network handelt. Dieses setzt sich aus Eingabeneuronen, verdeckten Neuronen, Ausgabeneuronen und den entsprechenden Verbindungen zusammen. Die Besonderheit des Netzes ist dabei, dass dieses als großes Reservoir beschrieben werden kann (vgl. Abbildung 2.4), welches neben den wie bei den in Abschnitt 2.3.4.1 beschriebenen Reservoirneuronen, die Eingabe- und Ausgabeneuronen erfasst. Die Eingabe- und Ausgabeneuronen werden dabei jedoch nicht unterschieden. Die Ausgabewerte des Netzes zu einem bestimmten Zeitpunkt  $t$  entsprechen den Eingabewerten zum nächsten Zeitpunkt  $t + 1$ , wodurch der Eingabewert nach jedem Zeitschritt durch den berechneten Ausgabewert lediglich überschrieben wird. Mithilfe des PrNN können Zeitreihen untersucht und entsprechende Vorhersagen getroffen werden. Wie in dem Kapitel der Zeitreihenanalyse erläutert gilt auch hier, die Entwicklung der Zeitreihe innerhalb eines bestimmten Zeitraumes bestmöglich zu verstehen und das neuronale Netz nach dem grundlegenden Muster entsprechend zu trainieren, sodass zuverlässige Vorhersagen getroffen werden können. Das Trainieren bzw. der Lernprozess betrifft dabei zum einen wie auch schon bei den ESN die Ausgabegewichte. Zum anderen jedoch ebenfalls die gesamte Netzwerkarchitektur. Dabei gilt es, die

Architektur auf die für die Prognosen tatsächlich benötigten Komponenten zu beschränken.

Um die Ausgaben des PrNN bestimmen zu können, werden zunächst die Verbindungsgewichte der einzelnen Neuronen zueinander erfasst. Für die Verbindungen von den Eingabe- bzw. Ausgabeneuronen zu den Reservoirneuronen ergibt sich die Gewichtsmatrix  $W^{in}$ . Die Gewichtsmatrix  $W^{res}$  fasst die Verbindungen der Reservoirneuronen  $N^{res}$  zueinander in einer  $N^{res} \times N^{res}$  Matrix zusammen. Die Gewichte der Verbindungen der Reservoirneuronen hin zu den Ausgabeneuronen werden in der Gewichtsmatrix  $W^{out}$  notiert. Die Gewichtsmatrizen ergeben zusammen nach Formel 2.48 die sogenannten Transitionsmatrix.

$$W = \begin{bmatrix} & W^{out} \\ W^{in} & W^{res} \end{bmatrix} \quad (2.48)$$

Die Gewichte der Eingabeneuronen in  $W^{in}$  und der Reservoirneuronen in  $W^{res}$  werden zufällig und identisch von der Standardnormalverteilung verteilt gewählt. Die Normen der Gewichtsmatrizen sollte ausgeglichen sein. Eine weitere Bedingung bei der Bestimmung von  $W^{res}$  besteht darin, dass der Spektralradius, d.h. der größte absolute Eigenwert dieser Matrix, einen Wert von Eins annehmen muss. Erst dann ist ein stabileres Neuronales Netz gewährleistet.

Der Zustände der Eingabe-/Ausgabe- und der Reservoirneuronen zu einem bestimmten Zeitpunkt werden durch den Vektor  $x(n)$  dargestellt. Der Initialzustand entspricht dabei dem Vektor  $x(0)$  mit  $N$  Komponenten und entspricht dem Startvektor des PrNN. Durch die Gewichtsmatrix  $W$  und die Zustände des Reservoirs  $x(n)$  kann nach Formel 2.49 die Ausgabe im Ausgabemodus zum Zeitschritt  $(t + \tau)$  berechnet werden, wobei  $\tau = 1$  entspricht.

$$y(t + \tau) = g(w_1 x_1(t) + \dots + w_n x_n(t)) \quad (2.49)$$

Dabei erfolgt die Aktivierung der Neuronen für alle Neuronen gleichzeitig über eine lineare Aktivierung  $g$ , die bei dem PrNN die Identitätsfunktion ist. Der damit errechnete Wert von  $y(t + \tau)$  steht dem Netz wiederum als Eingabe im nächsten Zeitschritt zur Verfügung.

Um diese Ausgaben jedoch so berechnen zu können, dass die Zeitreihe bestmöglich approximiert wird und somit zuverlässige Vorhersagen bestimmt werden können, müssen die Ausgabegewichte gelernt werden. Das Lernen der Gewichte erfolgt jedoch im Empfangsmodus des PrNN. Das bedeutet, dass dabei keine Ausgaben wie nach Formel 2.49 definiert berechnet werden. Da die gewünschten Ausgaben durch die zu betrachtende Zeitreihe mit der Sequenz  $S(0), \dots, S(n)$  vorliegen, werden diese

lediglich in die Ausgabeneuronen implementiert und müssen nicht noch bestimmt werden. Dabei handelt es sich um die im Abschnitt des ESN beschriebenen “teacher forced values”. Durch die initialisierten Verbindungsgewichte der Reservoir- und der Eingangsneuronen, sowie die durch  $S(0), \dots, S(n)$  festgelegten Eingabewerte gilt es zunächst, den Reservoirzustand nach Formel 2.51 zu berechnen. Hierzu wird zunächst der Startvektor  $x_0$  festgelegt, welcher sich aus dem Startvektor des Reservoirs mit  $R(t_0) = r$  und den Eingabewerten von  $S(0)$  zusammensetzt. Für eine bessere Stabilität des Netzes muss der Startvektor  $r$  jedoch nach Formel 2.50 normiert werden [7, S.4]. Auf Grundlage des damit errechneten Startvektors  $x_0$  wird der gesamte Reservoirzustand nach Formel 2.51 ermittelt.

$$r = \frac{1}{\sqrt{N_{res}}} \cdot [1 \dots 1]^T \quad (2.50)$$

$$R(t + \tau) = [W^{in} \quad W^{res}] \begin{bmatrix} S(t) \\ R(t) \end{bmatrix} \quad (2.51)$$

Damit der nächste Eingabewert  $S(t+\tau)$  bestimmt werden kann, werden die Zustände der Eingabeneuronen und die berechneten Reservoirzustände in einer Matrix  $X$  mit

$$X = \begin{bmatrix} S(0), \dots, S(n-1) \\ R(0), \dots, R(n-1) \end{bmatrix} \quad (2.52)$$

zusammengefasst. Da jeder Ausgabewert dem Eingabewert zum nächsten Zeitpunkt entspricht, wird eine weitere Matrix benötigt, die die Ausgaben in einer Ausgabematrix mit  $Y^{out} = [S(1), \dots, S(n)]$  durch  $Y^{out} = S(t + \tau)$  aus  $X(t)$  erfasst, wobei  $S(0)$  nicht vorhergesagt werden kann. Damit sind die Eingabezustände, die korrespondierenden Ausgabezustände und die Zustände der Reservoirneuronen bekannt, sodass die entsprechenden Ausgabegewichte durch Lösen einer Lineargleichung nach Formel 2.53 berechnet werden können. Somit kann die Berechnung der Ausgaben durch die gelernten Gewichte nach Formel 2.54 realisiert werden.

$$W^{out} = Y^{out} / X \quad (2.53)$$

$$Y^{out} = W^{out} \cdot X \quad (2.54)$$

Sobald die Ausgabegewichte bekannt sind, können die Vorhersagen unbekannter Werte einer Zeitreihe in dem Ausgabemodus durch Formel 2.55 berechnet werden.

$$\begin{bmatrix} S(t + \tau) \\ R(t + \tau) \end{bmatrix} = W \cdot \begin{bmatrix} S(t) \\ R(t) \end{bmatrix} \quad (2.55)$$

Zu dem Lernen der Gewichte, wird die Netzwerkarchitektur gelernt bzw. wird dieses nur auf die wichtigsten Komponenten reduziert. Diese Komponenten können mithilfe der Eigenwertzerlegung der Transitionsmatrix nach Formel 2.56 bestimmt werden.

$$W = V \cdot D \cdot V^{-1} \quad (2.56)$$

Dabei werden die Eigenvektoren  $v_1, \dots, v_N$  der Transitionsmatrix in der Matrix  $V$  zusammengefasst. Die Diagonalmatrix  $D$  beinhaltet die Eigenwerte  $\lambda_1, \dots, \lambda_N$ , die der Größe nach absteigend sortiert sind. Durch die Linearkombination der Eigenvektoren ergibt sich damit der Startvektor des PrNN  $x_0$  aus  $x_1v_1 + \dots + x_Nv_N$ . Vereinfacht kann dies durch  $V \cdot x$  mit  $x = [x_1, \dots, x_N]^T$  ausgedrückt werden. Daraus ergibt sich für  $x = V^{-1} \cdot x_0$ . Da die Gewichtsmatrix  $W$  eine lineare Abbildung jedes Eigenvektors  $v_k$  mit den Eigenwerten  $\lambda_k$  ist, wobei  $1 \leq k \leq N$  gilt, folgt  $W \cdot v_k = \lambda_k v_k$  [7, S.3]. Auf Grundlage der erläuterten Linearkombination ergibt sich mit Formel 2.57 die Ausgabe durch die Funktion  $f(t)$  zum Zeitpunkt  $t$  aus Formel 2.58.

$$W \cdot x_0 = W \cdot (x_1v_1 + \dots + x_Nv_N) = x_1\lambda_1v_1 + \dots + x_N\lambda_Nv_N \quad (2.57)$$

$$f(t) = W^t \cdot x_0 = x_1\lambda_1^t v_1 + \dots + x_N\lambda_N^t v_N = V \cdot D^t \cdot x \quad (2.58)$$

Nach Formel 2.58 und der Annahme, dass die Dimension  $d = 1$  vorliegt, gilt

$$f(t) = \alpha_1\lambda_1^t + \dots + \alpha_N\lambda_N^t, \quad (2.59)$$

wobei  $\alpha_1, \dots, \alpha_N$  aus Formel 2.58 bestimmt werden kann. Die Eigenwerte der Transitionsmatrix  $W$  werden wieder durch  $\lambda_1, \dots, \lambda_N$  dargestellt. Um identifizieren zu können, ob die entsprechenden Netzwerkkomponenten von  $\alpha_k$  und  $\lambda_k$  relevant für die Berechnung der Ausgabe ist, wird eine Fehlersumme nach Formel 2.60 berechnet. Dabei handelt es sich um den "Root mean square error" (RMSE, dt.: Wurzel der mittleren quadratischen Abweichung), welche nach

$$E_k = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (2.60)$$

definiert wird. Damit kann die Beurteilung der Modellgüte beschrieben werden. Der RMSE erfasst damit die durchschnittliche Abweichung zwischen den geschätzten Werten und den Werten der originalen Zeitreihe. Dabei werden die Komponenten  $k$  mit den größten berechneten Werten von  $E_k$  gewählt. Es gilt jedoch die Bedingung, dass die kumulierte Summe von  $E_k$  einen zuvor festgelegten Schwellenwert  $\theta \leq 1$  nicht überschreiten darf. Entsprechend der ausgewählten Komponenten folgen aus den Matrizen  $V$ ,  $D$  und  $x$  die reduzierten Matrizen  $\overset{*}{V}$ ,  $\overset{*}{D}$  und  $\overset{*}{x}$  [7, S.7]. Aus der Matrix  $V$  der Eigenvektoren werden die Zeilen gewählt, die mit den Eingabe- bzw. Ausgabekomponenten korrespondieren, sowie die Spalten mit den relevanten Netzwerkkomponenten. Aus  $D$  werden die Zeilen und Spalten entsprechend der relevanten Komponenten selektiert, wohingegen aus  $x$  die Zeilen der relevanten Komponenten gewählt werden.

## 3. Die Vorhersagemethoden

In diesem Abschnitt der Ausarbeitung wird der Aufbau der in den Grundlagen vorgestellten Vorhersagemethoden beschrieben. Die vorgestellten Modelle werden dabei mithilfe unterschiedlicher Software umgesetzt. Um die Funktionsweise und den Aufbau der einzelnen Methoden zu erläutern, werden diese am Beispiel des Aktienkurses von Adidas beschrieben. Im Rahmen der Modellbildung soll in diesem Abschnitt zudem identifiziert werden, welche Prognosemethoden die in der Einleitung vorgestellten Modifikationen der Vorhersagen ermöglichen. Das bedeutet, dass dabei zum einen ermittelt werden soll, welche Methoden das Einlesen, sowie die Vorhersage aller 30 Aktienkurse gleichzeitig ermöglichen. Zum anderen sollen die Methoden hinsichtlich der Möglichkeit untersucht werden, dass zunächst alle Aktienkurse gleichzeitig eingelesen werden, woraufhin die Prognose dabei jedoch nur für einen Aktienkurs erfolgt. Das Ziel dabei ist es zu identifizieren, ob Zusammenhänge zwischen den einzelnen Aktienkursen vorliegen und somit die Prognose eines einzelnen Aktienkurses durch den Einfluss unterschiedlicher branchenabhängiger und -unabhängiger Aktienkurse möglich ist.

### 3.1 Die Datenerhebung

Die Aktienkurse, die in dieser Ausarbeitung vorhergesagt werden sollen, werden über die Internetseite Yahoo Finance <sup>1</sup> bezogen. Die Finanzseite stellt den historischen Verlauf der Aktienkurse von unterschiedlichen Handelsplätzen zur Verfügung und ermöglicht zusätzlich, diese herunterzuladen. Aus Gründen der Einheitlichkeit werden die für diese Ausarbeitung verwendeten Aktienkurse von dem in Deutschland bedeutenden Handelsplatz "exchange electronic trading" (Xetra) bezogen. Über die Kursanfrage von Yahoo Finance werden 30 Aktienkurse willkürlich und branchenunabhängig gewählt. Dabei wird der Verlauf der Aktienkurse vom 01.01.2016 bis zum

---

<sup>1</sup><https://de.finance.yahoo.com> (zuletzt Aufgerufen am 20.06.2019)



01.01.2019, also einem Zeitraum von drei Jahren, untersucht. Über Yahoo Finance werden die Daten eines Aktienkurses in einer Datei im Format "Comma seperated value" (CSV) zum Herunterladen bereitgestellt. Das Dateiformat bietet den Vorteil, dass die einzelnen Daten in Textform durch Kommata getrennt vereinfacht durch die genutzten Programme eingelesen und verarbeitet werden können. Für die Vorhersage der Aktienkurse ist weiter zu berücksichtigen, dass sich für den Zeitraum von drei Jahren insgesamt 762 Handelstage ergeben. Dies hat den Grund, dass wie bei den meisten öffentlich-rechtlichen Handelsplätzen an Feiertagen und am Wochenende nicht gehandelt wird. Im Durchschnitt ergeben sich pro Jahr demnach 254 Handelstage an der Börse. Die jeweiligen CSV-Dateien enthalten das Datum, zu dem die relevanten Werte erfasst wurden, den entsprechenden Eröffnungskurs, den höchsten und den niedrigsten Handelspreis des jeweiligen Tages, den tatsächlichen sowie den bereinigten Schlusskurs und das Handelsvolumen. Das Handelsvolumen meint die Anzahl der an dem Tag gehandelten Aktien. Der bereinigte Schlusskurs beinhaltet den Neuberechneten Schlusspreis des jeweiligen Tages ohne Ausschüttungen bzw. Dividenden. Für diese Ausarbeitung gilt es jedoch, die Schlusskurse bzw. die Schlusspreise der Aktien vorherzusagen. Die zu untersuchenden Aktienkurse können aus Abschnitt A.1 entnommen werden.

Die Erhebung der Fundamentaldaten, die als Einflussgröße zur Vorhersage der Aktienkurse genutzt werden, erfolgt über die Internetseite Ariva <sup>2</sup>. Die Fundamentaldaten KGV, KCV und der Dividendenrendite werden dabei für jeden Aktienkurs für die Jahre 2016 bis 2018 erfasst. Für den KGV wird dabei der Gewinn je Aktie, für den KCV der Cashflow je Aktie und für die Dividendenrendite die Dividende je Aktie von der Finanzseite entnommen. Dabei wird für jedes Jahr der jeweilige Durchschnittswert erfasst. Damit ergeben sich beispielsweise für den Gewinn je Aktie für einen Zeitraum von drei Jahren drei verschiedene Werte, die berücksichtigt werden. Damit können entsprechend der in Abschnitt 2.1.3 aufgestellten Formeln die Berechnung des KGV, des KCV und der Dividendenrendite für jeden einzelnen Tag manuell erfolgen.

## 3.2 Die Modellbildung der Linearen Regression

Die Modellbildung und die Vorhersage der Aktienkurse mithilfe der Linearen Regression wird entsprechend der Prognoseform bzw. Modifikation der Vorhersage näher beschrieben.

### 3.2.1 Die Einzelkursvorhersage durch die Linearen Regression

Die erste Methode, die zur Vorhersage der Aktienkurse genutzt wird, ist die Lineare Regression. Hierzu wird die Statistik Software R unter der Version 3.5.2 <sup>3</sup> eingesetzt

---

<sup>2</sup><https://www.ariva.de> (zuletzt Aufgerufen am 18.07.2019)

<sup>3</sup><https://cran.r-project.org/bin/windows/base/old/3.5.2/>

[36]. Die Vorhersage der Aktienkurse und die Evaluierung der Modellgüte erfolgt dabei in zwei Schritten (Quelltext: B.1). Zunächst ist es notwendig, ein für den Aktienkurs geeignetes Lineares Regressionsmodell zu identifizieren und zu entwickeln. Dies erfolgt auf Grundlage eines bestimmten Anteils der eingelesenen Beobachtungen der 762 Daten. Dieser Datensatz wird als Trainingsdatensatz bezeichnet. Die restlichen Daten bilden den Testdatensatz. Auf Grundlage der Trainingsdaten wird ein Modell identifiziert, welches diese bestmöglich abbildet. Sobald ein geeignetes Modell erstellt wird, erfolgt die Prognose der Zeitreihe. Der Zeitraum der Prognose entspricht dabei dem des Testdatensatzes. Zur Identifizierung der Modellgüte wird abschließend der prognostizierte Testdatensatz und der reale Verlauf des Aktienkurses innerhalb der Testdaten miteinander verglichen.

Die Aufteilung der Daten für das Lineare Regressionsmodell erfolgt dabei zunächst zu einem Verhältnis von 80 zu 20. Damit enthält der Trainingsdatensatz 80% der gesamten Beobachtungen der drei Jahre. Somit werden die Werte vom 04.01.2016 bis zum 24.05.2018, also 610 Beobachtungen zur Modellbildung genutzt. Die Daten vom 25.05.2018 bis zum 28.12.2019 repräsentieren den Testdatensatz. Daraus ergeben sich für die Testdaten 152 Beobachtungen. Diese Aufteilung ist für den Aktienkurs von Adidas in Abbildung 3.1 dargestellt.

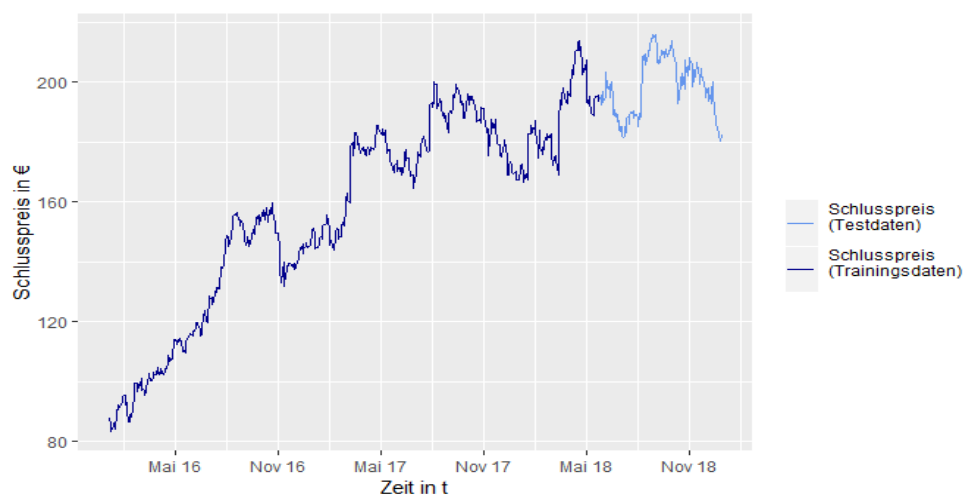


Abbildung 3.1: Test- und Trainingsdaten des Aktienkurses von Adidas

Im nächsten Schritt wird eine lineare Regressionsgerade ermittelt, die den Schlusspreis der Aktie in Abhängigkeit zur Zeit abbilden soll. Mithilfe der Gerade wird der Verlauf des Aktienkurses über den gesamten Zeitraum beschrieben, sodass auf Basis dessen der weitere Verlauf des Kurses prognostiziert werden kann. Im Rahmen des Regressionsmodells repräsentieren die Schlusspreise der Aktie den Regressand  $y$ , der in Abhängigkeit zur Zeit untersucht werden soll. Diese bildet den Regressor. Die Zeit wird dabei als einzelne Zeitschritte mit  $t = 0, 1, 2, \dots, n$  erfasst. Ein Zeitschritt

$t$  entspricht dabei einem Handelstag. Auf Basis der Beobachtungswerte erfolgt die Bestimmung des linearen Regressionsmodells nach Formel 2.12. Dabei gilt es zunächst jedoch die Unbekannten  $\alpha$  und  $\beta$  der Regressionsgerade zu schätzen. Nach dem Optimalitätskriterium der Methode der kleinsten Quadrate nach Formel 2.14 werden diese so bestimmt, dass die Regressionsgerade die Werte des Aktienkurses bestmöglich approximiert. Nach diesem Kriterium wird  $\beta$  dabei nach Formel 2.19 ermittelt. Daraus ergibt sich  $\alpha$  mit Formel 2.20.

Für den Schlusskurs von Adidas werden damit die Parameter  $\alpha = 104.6698$  und  $\beta = 0.1701$  ermittelt. In Bezug auf den vorliegenden Aktienkurs bedeutet dies, dass der Preis der Aktie zum Zeitpunkt  $t_0 = 104.6698\text{€}$  beträgt. Da der Regressionskoeffizient  $\beta$  den Zuwachs der Trendgeraden beschreibt, steigt der geschätzte Schlusspreis mit jedem Zeitschritt  $t$  um  $\beta$  an. Damit ergibt sich für den Aktienkurs von Adidas das Lineare Regressionsmodell nach Formel 3.1.

$$\hat{y}(t) = 104.6698 + 0.1701 \cdot t \quad (3.1)$$

Die Schätzung der Trainingsdaten durch das Modell sind in Abbildung 3.2 dargestellt.



Abbildung 3.2: Lineares Regressionsmodell

Im letzten Schritt wird das Regressionsmodell hinsichtlich seiner Prognosegüte bzw. der Zuverlässigkeit geprüft. Zur Evaluierung wird damit der Testdatensatz hinzugezogen. Nach dem aufgestellten Regressionsmodell in Formel 3.1 werden zunächst Prognosen in Bezug auf die Aktienkursentwicklung erstellt. Die Anzahl der Prognosewerte entspricht dabei dem Umfang der Testdaten. Die Prognose eines Wertes

zu einem gewünschten Zeitpunkt erfolgt durch Einsetzen des zu prognostizierenden Handelstags für  $t$  in Formel 3.1. Da der Trainingsdatensatz 610 Zeitschritte enthält, kann z.B. der erste Beobachtungswert des Testdatensatzes durch Einfügen des Zeitpunktes  $t = 611$  bestimmt werden. Durch Einsetzen des Zeitpunktes in Formel 3.1 ergibt sich die Formel  $\hat{y}(611) = 104.6698 + 0.1701 \cdot 611$ . Für diesen Tag wird damit ein Schlusspreis von  $\hat{y}(611) = 208.5712$  ermittelt. Dies wird für alle zu prognostizierenden Zeitschritte durchgeführt. Die Auswertung der Prognosen und die Evaluierung der Modellgüte erfolgt in Kapitel 4.

### 3.2.2 Die modifizierten Vorhersagen mithilfe der Linearen Regression

Das Lineare Regressionsmodell ermöglicht das Einlesen eines einzelnen Aktienkurses zu einem Zeitpunkt. Das Einlesen aller 30 Aktienkurse gleichzeitig mithilfe dieses Modells ist jedoch nicht möglich. Eine mögliche Maßnahme ist dabei die Aktienkurse einzeln mithilfe einer Schleife einzulesen und nacheinander das beschriebene Vorgehen der Modellbildung mithilfe der Linearen Regression durchzuführen. Jeder Aktienkurs wird damit einzeln betrachtet und hat keinen Einfluss auf die Vorhersage der weiteren Aktienkurse, wodurch die Ergebnisse sich nicht zu den zuvor ermittelten Ergebnisse der Einzelkursvorhersage unterscheiden.

Da nicht alle Aktienkurse gleichzeitig eingelesen werden können, ist es demnach ebenfalls nicht möglich mithilfe der Linearen Regression ein Aktienkurs auf Basis von allen Aktienkursen zu prognostizieren. Damit wird im Rahmen der Evaluierung die Lineare Regression für diese beiden Modifikationen zur Prognose der Aktienkurse ausgeschlossen und nicht weiter untersucht.

### 3.3 Die Modellbildung eines ARIMA-Modells

Wie schon in Abschnitt 2.2.2 beschrieben, gilt es bei der Zeitreihenanalyse ein Modell zu identifizieren, welches den grundlegenden stochastischen Prozess der Zeitreihe bestmöglich abbildet. Das ARIMA Modell ist eine Möglichkeit, den Aktienkurs zu modellieren und die Eigenschaften der Zeitreihe zu schätzen, sodass Vorhersagen getroffen werden können. In diesem Zusammenhang wird neben dem klassischen ARIMA-Modell das erweiterte saisonale ARIMA-Modell, das SARIMA-Modell untersucht und zur Prognose eingesetzt. Damit sollen gegebenenfalls vorliegende Saisonalitäten der Zeitreihe zur Prognose berücksichtigt werden. Zudem gilt es, wie schon bei dem Linearen Regressionsmodell, das ARIMA- bzw. SARIMA-Modell hinsichtlich der möglichen Prognosemodifikationen zu untersuchen. Die Vorgehensweisen und Unterschiede werden im folgenden näher erläutert. Alle Verfahren werden ebenfalls mithilfe der Statistik-Software R ausgeführt.

Dabei bleibt die zuvor festgelegte Aufteilung der Beobachtungen der Zeitreihe bestehen. Die Modellbildung erfolgt damit wieder auf Grundlage der 610 Trainingsdaten, wohingegen die 152 Testdaten zur Evaluierung der Modellgüte hinzugezogen wird.

### 3.3.1 Die Einzelkursvorhersage mithilfe eines ARIMA- Modells

Zur Bestimmung eines geeigneten ARIMA-Modells wird als Beispiel ebenfalls der Aktienkurs von Adidas, welcher in Abbildung 3.1 dargestellt ist, untersucht. Um ein geeignetes ARIMA(p,d,q)-Modell erstellen zu können, müssen bestehende Regelmäßigkeiten der Zeitreihe identifiziert werden, sodass die Wahrscheinlichkeitsverteilung geschätzt und somit Vorhersagen getroffen werden können. Ausschlaggebend für das Modell ist die Bestimmung geeigneter Parameter für p,d und q, d.h. die Ordnungen für den AR(p)-Prozess, den MA(q)-Prozess und den Grad der Differenzbildung. Damit wird bestimmt, welche vergangenen Zeitschritte bzw. welche vergangenen Fehler zwischen dem prognostizierten und tatsächlichen Kursverlauf einen Einfluss auf die Schätzung von  $\hat{y}$  haben. Zudem wird mit dem Parameter d festgelegt ob eine Transformation der Zeitreihe durch die Differenzbildung notwendig ist, sollte eine nichtstationäre Zeitreihe mit einer Einheitswurzel vorliegen. Wie schon in Abschnitt 2.2.4.2 erläutert, erfolgt die Schätzung dieser Parameter mithilfe unterschiedlicher Tests und Methoden. Zudem ist es notwendig, die entsprechenden Koeffizienten bzw. Gewichtungen  $\phi$  für das AR(p)-Modell und  $\theta$  für das MA(q)-Modell zu schätzen, sodass die vorliegende Zeitreihe bestmöglich abgebildet werden kann. Die Identifizierung dieser Parameter gestaltet sich in der Praxis als aufwendig. Aufgrund dessen, wurden im Laufe der Jahre unterschiedliche Vorgehensweisen entwickelt, die das bestmögliche Modell bestimmen sollen. In dieser Ausarbeitung wird das Vorgehen nach Rob J. Hyndman und Yeasmin Khandakar eingesetzt und vorgestellt [37, S.9-19]. Hyndman und Khandakar entwickelten einen Algorithmus auf Basis des von Box und Jenkins vorgestellten Ansatz zur Bestimmung eines ARIMA(p,d,q)-Modells [38]. Dieser Algorithmus, welcher mittels der Funktion “auto.arima(...)” in R ausgeführt wird, ermöglicht die automatische Bestimmung eines für den Aktienkurs geeigneten ARIMA(p,d,q)-Modells [37]. Die Bestimmung des Modells erfolgt dabei auf Basis des Trainingsdatensatzes, welcher wie schon zuvor beschrieben 80% der gesamten Beobachtungen des jeweiligen Aktienkurses bzw. in diesem Fall des Aktienkurses von Adidas enthält. Die Prognosen werden dabei mit dem entsprechenden Skript berechnet (Quelltext: B.2).

Zur Identifizierung geeigneter Parameter gilt es zunächst zu prüfen, ob es sich bei der vorliegenden Zeitreihe um eine stationäre Zeitreihe handelt oder ob diese gegebenenfalls von einer nichtstationären Zeitreihe mit einer Einheitswurzel in eine stationäre Zeitreihe transformiert werden muss, sodass anschließend ein geeignetes ARMA(p,q) Modell ermittelt werden kann. Die Funktion “auto.arima(...)” bietet dabei die Möglichkeit mithilfe eines Einheitswurzeltests die Zeitreihe auf Stationarität zu prüfen. Genutzt wird dabei der in Abschnitt 2.2.4.2 beschriebene KPSS-Hypothesentest. Mittels der “kpss.test(...)”-Funktion in R wird auf die Nullhypothesen geprüft, dass die vorliegende Zeitreihe stationär bzw. trendstationär ist. Das Ergebnis des KPSS-Tests wird dabei mithilfe eines Signifikanztests ausgewertet. Ausschlaggebend ist dabei das Signifikanzniveau  $\kappa$ , sowie die mittels der Funktion berechnete Kennzahl  $p$ , die entgültig festlegt, ob die Nullhypothese angenommen oder verworfen wird. Das

Signifikanzniveau wird dabei auf  $\kappa = 0.01$  gesetzt. Dies beschreibt, dass die Wahrscheinlichkeit, dass die Nullhypothese fälschlicherweise verworfen wird, maximal 1% betragen darf. Da das Ergebnis des KPSS-Tests auf Basis der vorliegenden Stichprobe bzw. der vorliegenden Zeitreihe ermittelt wurde, besteht die Wahrscheinlichkeit, dass die Nullhypothese angenommen bzw. verworfen wird, wobei diese Aussage nicht für den gesamten stochastischen Prozess gilt. Das Ergebnis ist dabei lediglich auf die Stichprobe zurückzuführen, sodass die Nullhypothese fälschlicherweise angenommen bzw. verworfen wird. Sollte damit mittels des Signifikanztests ein Wert von  $p > 0.01$  ermittelt werden, so kann die Nullhypothese angenommen werden. Damit handelt es sich bei der zu untersuchenden Zeitreihe um eine stationäre Zeitreihe. Bei  $p \leq 0.01$  wird die Nullhypothese abgelehnt und die Alternativhypothese angenommen, sodass eine nichtstationäre Zeitreihe mit einer Einheitswurzel vorliegt [39].

Für den Aktienkurs von Adidas ergibt sich dabei ein Wert von  $p \leq 0.01$ . Damit kann angenommen werden, dass die Nullhypothese abgelehnt und die Alternativhypothese angenommen werden kann. Demnach enthält die Zeitreihe eine Einheitswurzel bzw. einen integrierten Prozess. Somit ist die Transformation der Zeitreihe durch die Differenzbildung nach Formel 2.35 notwendig. Nach dem KPSS-Test der transformierten Zeitreihe ergibt sich ein Wert mit  $p > 0.01$ , sodass die Nullhypothese angenommen werden kann. Damit handelt es sich bei der differenzierten Zeitreihe um eine stationäre Zeitreihe bzw. gegebenenfalls um eine trendstationäre Zeitreihe. Die transformierte neue Zeitreihe ist in Abbildung 3.3 dargestellt. Für das ARIMA Modell ergibt sich damit zunächst ein ARIMA(p,1,q) Modell.

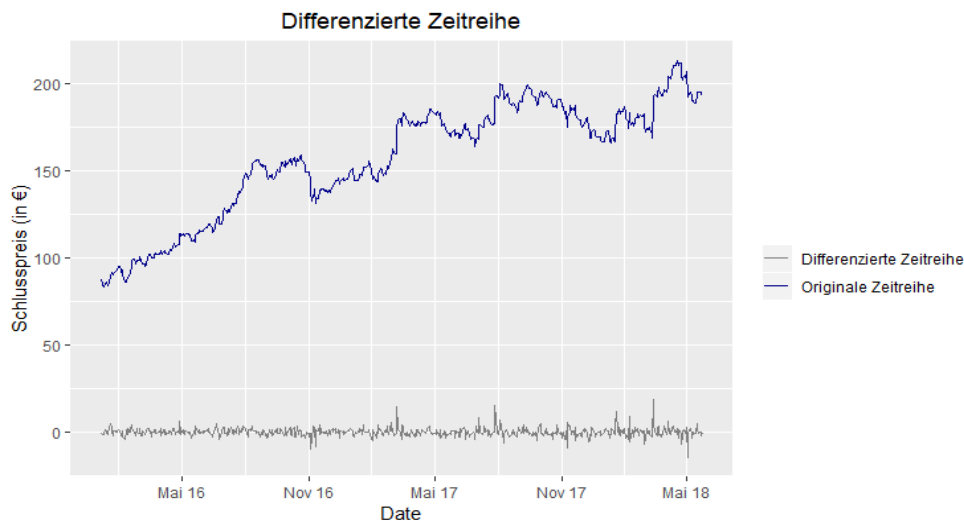


Abbildung 3.3: Die transformierte Zeitreihe

Im nächsten Schritt werden die Parameter  $p$  und  $q$  des bisher ermittelten ARIMA( $p,1,q$ ) Modells bestimmt. Mittels der “auto.arima(...)” Funktion, werden zunächst vier mögliche Modelle getestet. Diese Modelle sind im folgenden aufgestellt.

ARIMA(2,d,2)

ARIMA(0,d,0)

ARIMA(1,d,0)

ARIMA(0,d,1)

Für den Aktienkurs von Adidas wird, wie zuvor ermittelt, für  $d=1$  eingesetzt. Das Ziel dabei ist es, das Modell aus den vier Möglichkeiten zu ermitteln, welches die Zeitreihe im Vergleich zu allen anderen Modellen am besten approximiert. Das beste Modell wird nach dem “Akaike Informationskriterium (AIC)” gewählt [37]. Das AIC nach Formel 3.2 vergleicht die jeweiligen Modelle hinsichtlich festgelegter Kriterien. Dabei werden neben der Anpassungsgüte des Modells mittels einer Maximum-Likelihood Funktion ebenfalls die Anzahl der Parameter für  $p$  und  $q$ , die dabei genutzt werden, berücksichtigt. Das Ziel dabei ist es, die Komplexität des Modells so gering wie möglich zu halten. Um dies zu beachten, wird das Modell mit steigender Anzahl an Parametern schlechter bewertet als die Modelle mit einer geringeren Anzahl.

$$AIC = -2\log(L) + 2(p + q + k) \quad (3.2)$$

Mithilfe der “auto.arima(...)” Funktion wird ebenfalls berücksichtigt, dass bei der Zeitreihe eine Konstante, also ein Drift vorliegen kann. Sollte angenommen werden, dass für den Drift  $c \neq 0$  gilt, so wird dieser mit  $k = 1$  in Formel 3.2 mit einberechnet. Andernfalls wird für  $c = 0$  der Wert  $k = 0$  eingesetzt. Die Variable  $L$  meint das Maximum-Likelihood Schätzverfahren, mithilfe dessen die Koeffizienten  $\phi$  und  $\theta$  für den AR( $p$ ) bzw. den MA( $q$ ) Prozess bestimmt werden [40]. Durch das Schätzverfahren werden die Parameter gewählt, die die Zeitreihe bestmöglich abbilden. Es wird anschließend das Modell mit dem geringsten Wert für den AIC als aktuell bestes Modell ausgewählt. Auf Basis dessen werden Schrittweise die Ordnungen für  $p$  und  $q$  erhöht bzw. auch verringert und der AIC nach Formel 3.2 für jedes neue Modell berechnet. Ist der AIC bei einem veränderten Modell geringer, so wird dieses als aktuell bestes Modell gewählt. Da die Funktion nicht unendlich viele Testversuche durchführen soll, werden für  $p$  und  $q$  Begrenzungen hinsichtlich möglicher Ordnungen von  $0 \leq p \leq 5$  bzw.  $0 \leq q \leq 5$  eingesetzt [37].

Mithilfe der “auto.arima(...)” Funktion wird nun ein automatischer Testlauf gestartet, welcher nach dem AIC das bestmögliche ARIMA( $p,d,q$ ) Modell für die Zeitreihe des Adidas Aktienkurses identifiziert. Das Ergebnis ist in Abbildung 3.4 dargestellt.

```

Fitting models using approximations to speed things up...

ARIMA(2,1,2) with drift      : 2841.482
ARIMA(0,1,0) with drift     : 2851.506
ARIMA(1,1,0) with drift     : 2853.049
ARIMA(0,1,1) with drift     : 2852.023
ARIMA(0,1,0)                : 2852.403
ARIMA(1,1,2) with drift     : 2855.921
ARIMA(3,1,2) with drift     : 2845.412
ARIMA(2,1,1) with drift     : 2856.284
ARIMA(2,1,3) with drift     : 2843.501
ARIMA(1,1,1) with drift     : 2854.991
ARIMA(3,1,3) with drift     : 2846.186
ARIMA(2,1,2)                : 2842.82

Now re-fitting the best model(s) without approximations...

ARIMA(2,1,2) with drift      : 2842.304

Best model: ARIMA(2,1,2) with drift

```

Abbildung 3.4: Automatischer Suchlauf für das ARIMA(p,d,q)-Modell

Damit weist das ARIMA(2,1,2)-Modell mit Drift den niedrigsten AIC mit einem Wert von 2841.482 auf. Mittels der Maximum-Likelihood Funktion in Formel 3.2 werden zusätzlich die Koeffizienten  $\phi$  und  $\theta$  bestimmt, die den Verlauf der Trainingsdaten bestmöglich abbilden. Der Drift wird hierbei als  $c$  bezeichnet. Die entsprechenden Werte sind in Tabelle 3.1 aufgeführt.

Tabelle 3.1: Die Koeffizienten des ARIMA(2,1,2)-Modells

Koeffizient	Wert
$\phi_1$	0.9764
$\phi_2$	-0.9366
$\theta_1$	-0.9535
$\theta_2$	0.8747
$c$	0.1741

Die Annahme, dass eine Transformation der Zeitreihe zunächst durchgeführt wird, wird auch durch die identifizierte Ordnung der Differenzbildung von  $d=1$  bestätigt. Zudem werden zur Prognose die vergangenen zwei Beobachtungen durch den AR(2)-Modell sowie die ermittelten Fehler der vergangenen zwei Zeitschritte zwischen der prognostizierten Zeitreihe und den tatsächlichen Verlauf des Aktienkurses in Bezug auf die Trainingsdaten durch das MA(2)-Modell als für die Prognose relevanten Werte identifiziert.

Im letzten Schritt gilt es zu überprüfen, ob das errechnete Modell die wichtigsten Informationen und Muster der Zeitreihe berücksichtigt, die mithilfe des AR(2) bzw. des MA(2) Prozesses erklärt werden können. Im bestmöglichen Fall handelt es sich



bei den Residuen, d.h. bei den Abweichungen zwischen den Werten des geschätzten und des realen Aktienkurses, die nicht durch die Prozesse beschrieben werden können, um ein weißes Rauschen. Sollte dies nicht zutreffen, muss überprüft werden, ob das Modell manuell angepasst und optimiert werden kann. Dies kann mithilfe des Korrelogramm in R untersucht werden, welches die Autokorrelation der Residuen untersucht. Diese sollten bei einem weißen Rauschen nicht signifikant sein. Für den Aktienkurs von Adidas mit dem aufgestellten ARIMA(2,1,2)-Modell ist die Beziehung der Residuen in Abbildung 3.5 dargestellt.

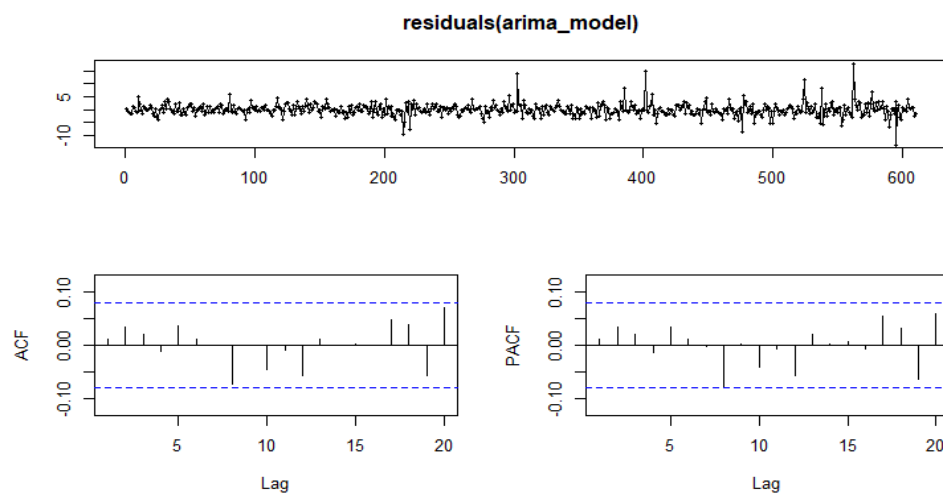


Abbildung 3.5: Analyse der Residuenanalyse

Aufgeführt ist dabei die Korrelation der Residuen zu der Verschiebung um eine Zeiteinheit  $t$ . Die Begrenzungen durch die blaue Linie weisen dabei auf das Signifikanzniveau hin, mit der die Annahme, dass Korrelationen bestehen angenommen bzw. verworfen werden kann. Da dies zu keinem Zeitpunkt überschritten wird, kann angenommen werden, dass keine signifikanten Korrelationen zwischen den einzelnen Residuen bestehen und es sich damit um ein weißes Rauschen handelt. Somit wird das Modell als geeignetes Modell festgelegt. Nach dem aufgestellten Modell ergibt sich der geschätzte Aktienkursverlauf von Adidas in Abbildung 3.6.

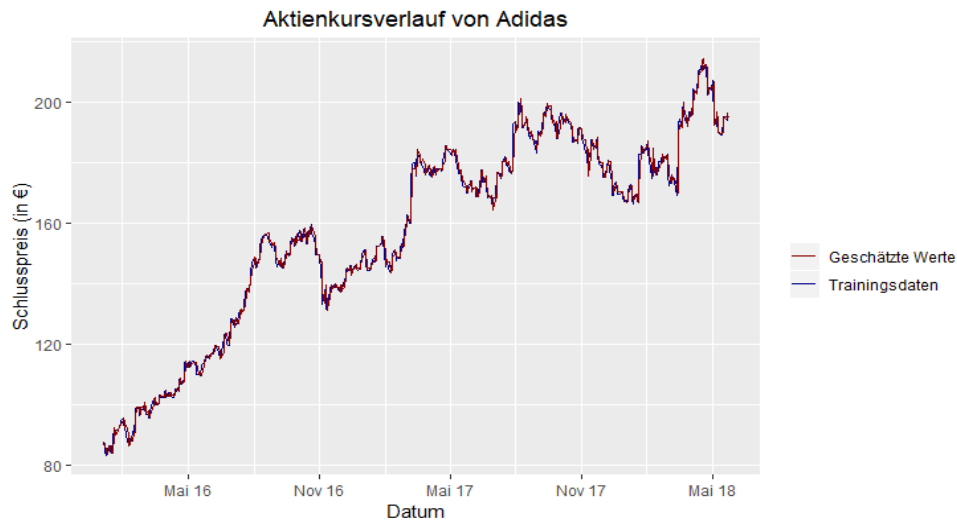


Abbildung 3.6: Geschätzter Aktienkurs mithilfe des ARIMA-Modells

Die weitere Vorhersage mit dem ermittelten  $ARIMA(2,1,2)$  Modell mit Drift sowie den in Tabelle 3.1 aufgeführten Koeffizienten erfolgt mithilfe der Funktion “forecast(...)”. Dabei wird zunächst der Prognosezeitraum von 152 Zeitschritte, also 152 Handelstagen festgelegt und auf Basis des errechneten  $ARIMA(2,1,2)$ -Modells die entsprechenden Vorhersagen getroffen.

### 3.3.2 Die Einzelkursvorhersage mithilfe eines SARIMA- Modells

Das klassische  $ARIMA(p,d,q)$ -Modell bietet die Möglichkeit Zeitreihen vorherzusagen, wobei mögliche saisonale Zyklen nicht berücksichtigt werden. Das erweiterte  $ARIMA$ -Modell, das saisonale  $ARIMA$  bzw. ( $SARIMA$ )-Modell, berücksichtigt solche Periodizitäten. Ein für die Zeitreihe geeignetes  $SARIMA$ -Modell kann ebenfalls mithilfe der Funktion “auto.arima(...)” bestimmt werden [37]. Die Bestimmung eines geeigneten Modells für eine Zeitreihe mit einem saisonalen Muster erfolgt dabei auf Grundlage des in Abschnitt 2.2.3 beschriebenen Komponentenmodells. Das Ziel hierbei ist es, die vorliegende Zeitreihe von der saisonalen Komponente zu bereinigen. Das Ergebnis sind damit zwei Zeitreihen. Eine Zeitreihe enthält dabei die Werte der saisonalen Komponente, wohingegen die zweite Zeitreihe Werte aufweist, die von der saisonalen Komponente bereinigt sind. Damit wird ermöglicht, ein geeignetes  $ARIMA(P,D,Q)$ -Modell für die erfasste saisonale Zeitreihe und ein geeignetes  $ARIMA(p,d,q)$ -Modell für die bereinigte Zeitreihe zu identifizieren. Die erfassten Modelle werden damit zu einem  $ARIMA(p,d,q)(P,D,Q)_m$ -Modell zusammengesetzt, sodass auf Grundlage dessen, die Prognose der Zeitreihe erfolgen kann. Dieses wird in dieser Ausarbeitung mit dem Skript nach (Quelltext: B.3) ausgeführt.

Damit unter Verwendung der “auto.arima(...)”-Funktion eine saisonale Komponente ermittelt werden kann, muss eine Frequenz  $m$  manuell festgelegt werden. Diese Fre-

quenz legt den Zeitraum einer Periode fest, innerhalb welcher ein saisonales Muster identifiziert werden soll. Dabei kann angenommen werden, dass das für diese Periode ermittelte ARIMA(P,D,Q)-Modell für jede Periode eingesetzt werden kann. Für diese Ausarbeitung wird die Frequenz dabei auf 254 festgelegt. Wie zuvor beschrieben, entsprechen 254 Zeitschritte im Durchschnitt einem Handelsjahr an der Börse. Damit wird angenommen, dass sich das Muster innerhalb eines Handelsjahres jedes Jahr wiederholt. Im nächsten Schritt erfolgt die Aufteilung der Zeitreihe in seine Komponenten. Dies erfolgt mithilfe der “`stl(...)`”-Funktion in R. “STL” steht dabei für “Seasonal and Trend decomposition using Loess”. Damit ist die Zersetzung der Zeitreihe in Saisonalitäten und einem Trend mithilfe des Loess Glättungsverfahrens gemeint. Loess steht für “Locally weighted scatterplot smoothing”. Mithilfe einer gewichteten lokalen Regression werden hierbei die Werte der saisonalen Komponente geschätzt. Für eine detailliertere Beschreibung des Verfahrens und der “`stl(...)`”-Funktion wird auf die Literatur [41] verwiesen.

Die Bestimmung der Parameter erfolgt im nächsten Schritt ähnlich zu der bei dem klassischen ARIMA-Modell beschriebenen Vorgehensweise. Zunächst wird ermittelt, inwiefern beide Komponenten in stationäre Zeitreihen transformiert werden müssen. Dabei wird im ersten Schritt die Ordnung von  $D$ , d.h. der saisonalen Zeitreihe, mittels einer Erweiterung des Canova-Hansen Tests identifiziert [42]. Dabei wird mittels einer Hypothesentest geprüft, ob die Komponente einen integrierten Prozess, also eine Einheitswurzel aufweist und somit zunächst transformiert werden muss. Die Nullhypothese sagt dabei aus, dass stationäre saisonale Zyklen bei der saisonalen Komponente vorliegen, wohingegen die Alternativhypothese auf eine vorliegende Einheitswurzel hinweist. Nachdem die Ordnung für  $D$  bestimmt wurde, wird im nächsten Schritt  $d$  für die von der Saison bereinigte Zeitreihe ermittelt [37, S.10]. Hinzugezogen wird dabei wieder das KPSS-Testverfahren. Sollten die Ordnungen für  $d$  und  $D$  bestimmt sein, so werden im nächsten Schritt die Ordnungen für  $p$ ,  $q$ ,  $P$  und  $Q$  bestimmt. Wie zuvor beschrieben, werden zunächst vier Modelle getestet. Sollte  $m > 1$  gelten, d.h. sollten Saisonalitäten berücksichtigt werden, so ergeben sich die folgenden angepassten ersten Testmodelle.

$$\text{ARIMA}(2,d,2)(1,D,1)$$

$$\text{ARIMA}(0,d,0)(0,D,0)$$

$$\text{ARIMA}(1,d,0)(1,D,0)$$

$$\text{ARIMA}(0,d,1)(0,D,1)$$

Dabei wird ebenfalls berücksichtigt, dass ein integrierter Prozess mit Drift, also einer Konstante  $c$  vorliegen kann. Dies wird ebenfalls getestet und berücksichtigt. Mit dem ersten nach dem AIC nach Formel 3.3 ermittelten bestmöglichen Modell, werden die Ordnungen weiter erhöht bzw. verringert.

$$AIC = -2\log(L) + 2(p + q + P + Q + k) \quad (3.3)$$

Dabei gilt für  $p$  und  $q$  wieder  $0 \leq p \leq 5$  bzw.  $0 \leq q \leq 5$ , wohingegen  $P$  und  $Q$  Werte von  $0 \leq P \leq 1$  bzw.  $0 \leq Q \leq 1$  annehmen können [37]. Mittels des automatischen Suchlaufs wird das  $ARIMA(p,d,q)(P,D,Q)_m$  Modell für den Aktienkurs von Adidas ausgewählt, welches den geringsten AIC aufweist. Durch die Funktion “auto.arima(...)” wird für den Aktienkurs mit einem AIC von 1895.21 ein  $ARIMA(0,1,0)(0,1,0)_m$ -Modell als geeignetes Modell identifiziert. Die Schätzung für den Trainingsdatensatz aus dem ermittelten Modell ist Abbildung 3.7 aufgeführt.

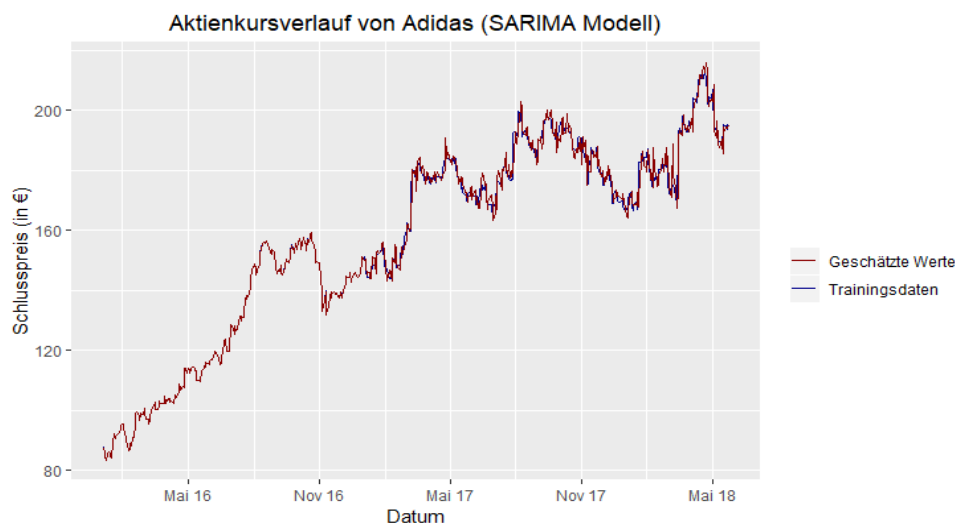


Abbildung 3.7: Geschätzter Aktienkurs mithilfe des SARIMA-Modells

Damit wurden für die saisonale Komponente, als auch für die von dem saisonalen Einfluss bereinigten Zeitreihe um Zeitreihen, die jeweils eine Einheitswurzel aufweisen. Da für beide Zeitreihen keine  $AR(p)$ - bzw.  $AR(P)$ - und  $MA(q)$ - bzw.  $MA(Q)$ -Modelle erfasst wurden, handelt es sich um ein saisonales Random-Walk-Modell.

Die Prognose des Aktienkurses von Adidas erfolgt ebenfalls mittels der “forecast(...)”-Funktion, bei dem mittels des ermittelten Modells Vorhersagen getroffen werden. Diese werden im Abschnitt der Evaluierung in Kapitel 4 vorgestellt und ausgewertet.

### 3.3.3 Die modifizierten Vorhersagen mithilfe des ARIMA-/SARIMA-Modells

Wie auch mit dem Linearen Regressionsverfahren, ist es mit einem ARIMA-Modell nicht möglich, alle Aktienkurse gleichzeitig einzulesen und zu prognostizieren. Dies ist wieder nur über das Einlesen der Aktienkurse mithilfe einer Schleife möglich,

wobei sich das entsprechende Modell und die Prognosen nicht zu dem der Einzelkursvorhersage ändern würde. Da diese Form der Prognose keine Auswirkung auf die Vorhersage hat, wird die Möglichkeit alle Aktienkurse gleichzeitig einzulesen in der Evaluierung nicht betrachtet.

Damit ist auch die Prognose eines einzelnen Aktienkurses aus allen nicht möglich und wird in dieser Ausarbeitung nicht weiter analysiert.

## 3.4 Die Modellbildung des Predictive Neural Network

Ein weiteres Modell, welches zur Vorhersage der Aktienkurse untersucht wird, ist das Predictive Neural Network nach Stolzenburg et al. [7]. Dabei sollen mithilfe eines Reservoirs nach dem Vorbild des Echo State Networks, die historischen Daten der Aktienkurse eingelesen und entsprechend der Eingaben die Ausgabegewichte des Netzes erlernt werden. Sobald das Netz trainiert wurde, können weitere Vorhersagen der Zeitreihe erfolgen. Der Aufbau und die Vorgehensweise werden in diesem Kapitel vorgestellt. Dabei ist das PrNN vollständig in Octave (Version 4.4.1) implementiert<sup>4</sup> [43]. Die Dimensionsreduktion des Neuronalen Netzes wird in dieser Ausarbeitung nicht untersucht.

### 3.4.1 Die Einzelkursvorhersage mithilfe des PrNN

Im ersten Schritt der Ausarbeitung wird mittels des Predictive Neural Network ein einzelner Aktienkurs vorhergesagt. Anhand des Skripts “prep\_prediction.m” (Quelltext B.4) werden zunächst bestimmte Voreinstellungen bearbeitet. Unter anderem wird innerhalb der Umgebung die Anzahl der Neuronen, aus denen das Reservoir gebildet werden soll, manuell festgelegt. Diese wird zunächst zur Beschreibung des Modells auf  $N = 200$  gesetzt und im Rahmen der Evaluierung im weiteren Verlauf der Ausarbeitung verändert. Die Anzahl von 200 Neuronen stellt hierbei die geringste Anzahl an Neuronen dar, die in dieser Ausarbeitung untersucht wird. Zusätzlich kann ausgewählt werden, wieviele Beobachtungen des Aktienkurses der letzten drei Jahren zur Entwicklung des PrNN eingelesen und berücksichtigt werden sollen. Zur Beschreibung des Modells werden zunächst alle erfassten Daten der drei Jahre eingelesen. Dies entspricht 762 Beobachtungen. Die Anzahl wird ebenfalls im Rahmen der Evaluierung verändert. Durch das Skript “prep\_prediction.m” soll ebenfalls festgelegt werden, welche Form der Vorhersage durchgeführt werden soll. Dies wird mithilfe des “switch”-Befehls und der Variable “select” ermöglicht. Für die Variable “select” wird festgelegt, welche Form der Vorhersage erfolgen soll. Mögliche Optionen sind dabei in Tabelle 3.2 aufgeführt.

---

<sup>4</sup><https://www.gnu.org/software/octave/>

Tabelle 3.2: Optionen der Vorhersage

Option	Prognoseform
select = 0	Einzelkursvorhersage
select = 1	Vorhersage aller Aktienkurse gleichzeitig
select = 2	Vorhersag eines Aktienkurses aus allen

Zur Vorhersage eines einzelnen Kurses zu einem bestimmten Zeitpunkt wird “select=0” gewählt. Damit wird für alle weiteren Skripte festgelegt, welche entsprechenden Funktionen ausgeführt werden müssen.

Das Skript, welches die einzelnen Funktionen zur Berechnung des PrNN enthält und die Vorhersagen ausführt, ist das Skript “compute\_predictions.m” (Quelltext B.5). Durch das Ausführen des Skriptes, werden alle für die Prognose notwendigen Funktionen aufgerufen. Nachdem hierbei “prep\_predictions.m” ausgeführt wurde, werden die Daten eines ausgewählten Aktienkurses mittels des Skripts “read\_in\_data.m” (Quelltext B.6) in Octave eingelesen und entsprechend der zuvor beschriebenen Aufteilung mit einem Verhältnis von 80 zu 20 in die Trainings- und Testdaten aufgeteilt. Für die Einzelkursvorhersage muss dabei manuell festgelegt werden, welcher Kurs eingelesen und prognostiziert werden soll. Jeder Aktienkurs wird dabei durch eine Nummer ausgewählt. Diese ist in Abschnitt A.1 entsprechend der Reihenfolge zu wählen. Für den Aktienkurs von Adidas wird damit die Zahl zwei eingesetzt. Damit liegen für die Prognose des Aktienkurses die Beobachtungen in den Trainings- und Testdaten vor. Die Trainingsdaten umfassen dabei 610 Beobachtungen, wohingegen die Testdaten 152 Daten umfasst.

#### 3.4.1.1 Die Initialisierung des PrNN

Je nach Prognoseform wird innerhalb des Skripts “compute\_predictions.m” die jeweilige Funktion zur Prognose des Aktienkurses durchgeführt. Für die Einzelkursvorhersage wird dabei das Skript “single\_stock.m” (Quelltext B.7) ausgeführt. Dieses beinhaltet die Funktion “predict.m” (Quelltext B.8). Hierbei erfolgt die Entwicklung des PrNN. Die Anzahl der Daten, die dabei prognostiziert werden sollen, entspricht der Anzahl des Testdatensatzes. Damit ergibt sich für  $M$ , welches die zu prognostizierenden Zeitschritte angibt,  $M = 152$ . Entsprechend der Anzahl der Beobachtungen des Trainingsdatensatzes, welches zunächst 610 Beobachtungen umfasst und der festgelegten Anzahl an Reservoirneuronen mit  $N = 200$ , werden dabei zunächst das Reservoir des PrNN sowie die entsprechende Gewichtsmatrix initialisiert. Die Gewichtsmatrix  $W$  bzw. die Transitionsmatrix, die die gewichteten Verbindungen zwischen den einzelnen Neuronen beschreibt, wird nach Formel 2.48 aufgestellt. Diese entspricht dabei einer Matrix von  $W((N + d) \times (N + d))$ , wobei  $d$  der Anzahl der Eingabeneuronen bzw. der Anzahl der einzulesenden Aktienkurse repräsentiert. Da zunächst nur ein Aktienkurs eingelesen wird, ergibt sich  $d = 1$ , sodass die

Gewichtsmatrix für die Eingabegewichte  $W^{in}$ , die Reservoirgewichte  $W^{res}$  und die Ausgabegewichte  $W^{out}$  mit einer Größe von  $W(201 \times 201)$  initialisiert wird. Wie in Abschnitt 2.3.4.2 erläutert, werden die Werte von  $W^{in}$  und  $W^{res}$  zufällig gewählt. Mittels des Skripts “reservoir.m” (Quelltext B.9) werden dabei die Werte von  $W^{res}$  mit einem Spektralradius von  $\lambda = 1$  bestimmt.

Im nächsten Schritt werden die Zustandswerte der Neuronen innerhalb des Reservoirs in der Zustandsmatrix  $X$  initialisiert. Die Größe der Matrix  $X$  wird mit  $X((N + d) \times (n + 1))$  bestimmt, wobei  $n$  der Anzahl der Trainingsdaten entspricht. Daraus ergibt sich eine Matrix der Größe  $X(201 \times 610)$ . Um die gesamte Zustandsmatrix des Reservoirs nach Formel 2.52 berechnen zu können, wird ein Startvektor  $R(t_0 = r)$  bestimmt, der die initialen Zustandswerte des Reservoirs beinhaltet. Zur Berechnung weiterer Zustände, werden mittels des Skripts “compute.m” (Quelltext B.10) die Trainingsdaten im Empfangsmodus in die Zustandsmatrix eingelesen, sodass sich für den Zustand zum Zeitpunkt  $t = t_0$  ein Startvektor  $x_0$  aus  $S(t_0)$  und  $R(t_0)$  ergibt. Daraus ergibt sich vereinfacht dargestellt eine Matrix der Form

$$X = \begin{pmatrix} S_{1,1} & \dots & S_{1,610} \\ R_{2,1} & \dots & R_{2,610} \\ \vdots & \ddots & \vdots \\ R_{230,1} & \dots & R_{230,610} \end{pmatrix}. \quad (3.4)$$

Damit entspricht die erste Zeile der Matrix den Werten der Trainingsdaten. Die restlichen Zeilen repräsentieren die Reservoirzustände des PrNN. Mittels dem zuvor ermittelten Startvektor  $x_0$  und der zuvor berechneten Gewichtsmatrix werden alle weiteren Zustände der Matrix nach Formel 2.51 berechnet.

### 3.4.1.2 Das Lernen der Ausgabegewichte

Sobald der Reservoirzustand und die einzelnen Gewichte bestimmt wurden, können die Ausgabegewichte durch das PrNN gelernt werden. Dies wird im Ausgabemodus mittels des Skripts “compute.m” ausgeführt. Da die Ausgabe der PrNN zum Zeitpunkt  $t$  der Eingabe zum Zeitpunkt  $t + 1$  entspricht und die Beobachtungen des Aktienkurses zum gesamten betrachteten Zeitraum vorliegen, sind alle Ausgabewerte bekannt. In einem Ausgabevektor  $Y^{out}$  werden aus diesem Grund die Werte der Zeitreihe ab dem Zeitpunkt  $t_1$  eingelesen. Mittels der Eingabewerte und den korrespondierenden Ausgabewerte können die Ausgabegewichte  $W^{out}$  nach Formel 2.53 berechnet werden. Durch die Zustandsmatrix und der ermittelten Gewichtsmatrix wird damit ein an die vorliegende Zeitreihe angepasstes PrNN erstellt.

### 3.4.1.3 Die Schätzung des Aktienkurses

Da das Prognoseergebniss durch die für  $W^{in}$  und  $W^{res}$  ermittelten Zufallszahlen beeinflusst wird, ist es möglich, dass entsprechend dieser Zahlen im ersten Testdurchlauf

schlechtere Prognosen erzielt werden, als in einem zweiten Testdurchlauf. Demnach wird für die Modellbildung und die Vorhersage der Aktienkurse die Funktion “predict.m” 20 mal durchlaufen und daraus das Modell gewählt, welches den geringsten RMSE zwischen den prognostizierten Testdaten und den entsprechenden Werten des Aktienkurses ermöglicht. Die durchschnittlichen Abweichung bzw. der RMSE wird innerhalb des Skripts “predict.m” mithilfe des Skripts “nrmse.m” ermittelt. Dies wird mittels einer Schleife realisiert, bei der das erste Modell zunächst als bestes Modell gewählt wird. Ist die genannte Abweichung im nächsten Testdurchlauf geringer, wird das bestehende Modell von dem neu ermittelten Modell ersetzt und als neues bestes Modell festgelegt. Der geschätzte Aktienkursverlauf der Trainingsdaten des besten Modells ist in Abbildung 3.8 abgebildet.



Abbildung 3.8: Geschätzter Aktienkurs mithilfe des PrNN

Im nächsten Schritt werden mittels der erlernten Ausgabegewichte, Vorhersagen für  $M = 152$  weitere Zeitschritte getroffen. Dies wird durch die Ausführung der Funktion “compute.m” im Ausgabemodus ermöglicht. Auf Grundlage der Zustandsmatrix  $X$  und der Gewichtsmatrix  $W$  werden dabei die Vorhersagen nach Formel 2.49 berechnet.

### 3.4.2 Vorhersage aller Aktienkurse

Im Gegensatz zu der Linearen Regression und dem ARIMA-Modell ist das Einlesen und die Vorhersage aller 30 Aktienkurse gleichzeitig mit dem PrNN möglich. Zunächst müssen dabei die benötigten Konfigurationen des PrNN in “prep\_prediction.m” dementsprechend geändert werden. Die Anzahl der Neuronen wird zunächst wieder auf  $N = 200$  gesetzt und es werden wie zuvor 762 Beobachtungen zur Analyse der Prognosemethode verwendet. Manuell muss jedoch die Form der Vorhersage



festgelegt werden. Das bedeutet, dass sich der Wert von “select” auf “select=1” ändert. Sobald das Skript “compute\_predictions.m” ausgeführt wird, werden durch das Skript “read\_in\_data.m” alle CSV-Dateien zu den Aktienkursen eingelesen und in einer Matrix gespeichert. Die Daten der Aktienkurse werden anschließend wieder in die Trainingsdaten und Testdaten, entsprechend der zuvor beschriebenen Aufteilung separiert.

Zur Berechnung der Vorhersagen wird anschließend das Skript “multiple\_stock.m” (Quelltext B.12), welches das Skript “predict.m” zur Modellbildung und Prognose beinhaltet, ausgeführt. Die Variable M wird hierbei wieder auf M=152 gesetzt, sodass nach der Modellbildung im Rahmen der Prognose 152 Zeitschritte vorhergesagt werden. Durch “predict.m” werden die Gewichtsmatrix und die Zustandsmatrix zunächst initialisiert. Aufgrund der höheren Anzahl der Zeitreihen, die es zu untersuchen gilt, vergrößert sich das Reservoir, sowie die entsprechende Gewichtsmatrix. Für 30 Aktienkurse und einer Anzahl von 200 Neuronen für das Reservoir, ergibt sich eine Gewichtsmatrix mit  $W(230 \times 230)$  Werten, sowie  $X(230 \times 610)$  für die Zustandsmatrix. Die Werte werden wie zuvor in Abschnitt 3.4.1.1 und Abschnitt 3.4.1.2 beschrieben bestimmt. Nachdem hierbei die Gewichtsmatrix und die Zustandsmatrix initialisiert wird, erfolgt die Bestimmung der Zufallszahlen der Eingabegewichte  $W^{in}$  und die Reservoirgewichte  $W^{res}$ . Zudem wird dabei wieder ein Startvektor  $r$  der Zustandsmatrix  $X$  ermittelt. Dieser beinhaltet die Startwerte für das Reservoir des PrNN zum Zeitpunkt  $t_0$ . Die Modellbildung erfolgt anschließend auf Basis von “compute.m”, innerhalb welcher alle weiteren Reservoirzustände bis zum Zeitpunkt  $t_{610}$  ermittelt werden. Dieser Zeitpunkt entspricht dabei dem letzten Zeitpunkt der Trainingsdaten. Die Zustandsmatrix, die berechnet werden soll, ist in Formel 3.5 aufgestellt.

$$X = \begin{pmatrix} S_{1,1} & \dots & S_{1,610} \\ \vdots & \ddots & \vdots \\ S_{30,1} & \dots & S_{30,610} \\ R_{31,1} & \dots & R_{31,610} \\ \vdots & \ddots & \vdots \\ R_{230,1} & \dots & R_{230,610} \end{pmatrix}. \quad (3.5)$$

Die erste Zeile bis hin zur Zeile 30, beinhalten die Werte der Aktienkurse, die eingelesen werden. Die Werte ab Zeile 31 entsprechen dabei den Zuständen des Reservoirs. Demnach werden zur Berechnung zunächst die ersten 30 Zeilen mit den entsprechenden Trainingsdaten der Aktienkurse befüllt. Aufgrund der Reihenfolge, in der die Aktienkurse eingelesen werden, werden die Werte für den Aktienkurs von Adidas in die zweite Zeile eingesetzt. Die Werte des zuvor ermittelten Startvektors  $r$  werden in die erste Spalte entsprechend der Zeilen des Reservoirs eingefügt. Mittels der Gewichtsmatrix und dem Startvektor werden, wie auch schon bei der Einzelkursvorhersage, im nächsten Schritt die Reservoirzustände zum Zeitpunkt  $t + 1$  berechnet

und in der Matrix gespeichert. Damit werden nacheinander alle Reservoirzustände ermittelt.

Im letzten Schritt werden die Ausgabegewichte der Gewichtsmatrix  $W$  ermittelt. Die Ausgabegewichte werden dabei für jeden Aktienkurs und jeden Zeitschritt berechnet. Mittels der Transitionsmatrix werden im nächsten Schritt die Trainingsdaten der Zeitreihe geschätzt. Die Zuverlässigkeit der Schätzungen ist in Abbildung 3.9 dargestellt.

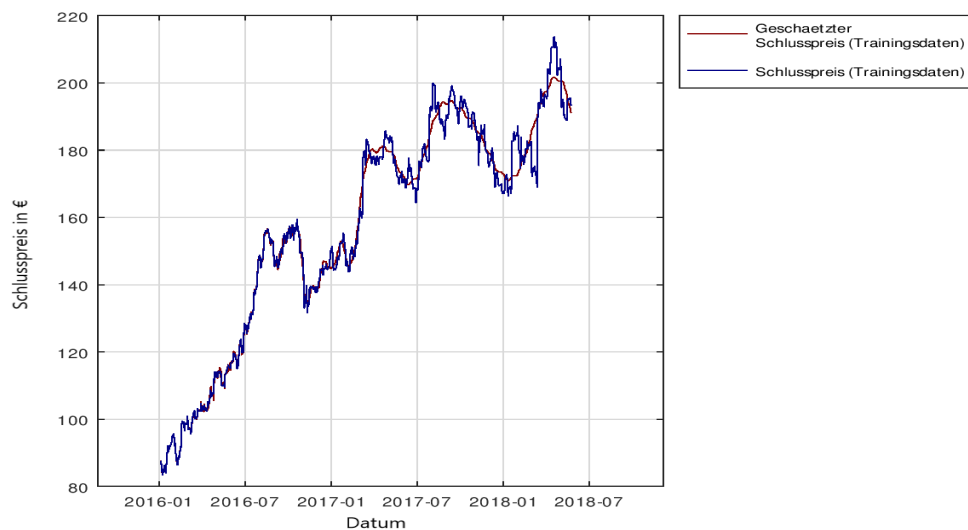


Abbildung 3.9: Geschätzter Aktienkurs mithilfe des Multi-PrNN

Zur Prognose der 152 Testdaten werden im nächsten Schritt die letzten geschätzten Werte der Aktienkurse zum Zeitpunkt  $t_{610}$  als neue Startvektor festgelegt. Mittels der Transitionsmatrix und des Startvektors werden im Ausgabemodus durch “compute.m” die nächsten 152 Werte für alle 30 Aktienkurse berechnet.

Die Berücksichtigung der Fundamentaldaten in dieser Ausarbeitung erfolgt analog, wie bei diesem Modell des PrNN. Dabei werden alle Fundamentalwerte gleichzeitig als einzelne Zeitreihen eingelesen und auf Basis dessen, Prognosen der Zeitreihe durchgeführt. Die Vorhersage mit den Fundamentaldaten ist in einer separaten Funktion aufgeführt. Hierzu ist es lediglich notwendig, innerhalb des Skripts “prep\_prediction” den Ordner mit den Aktienkursen zu ändern. Die Kurse mit den Fundamentaldaten sind hierbei aufgeführt. Anschließend wird zur Berechnung der Prognosen lediglich das Skript “fundamental\_stock.m” (Quelltext: B.14) ausgeführt.

### 3.4.3 Vorhersage eines einzelnen Aktienkurses aus allen

Auch die Vorhersage eines einzelnen Aktienkurses aus allen ist mittels des PrNNs möglich. Das Einlesen der Daten erfolgt, wie bei der Vorhersage aller Aktienkurse

gleichzeitig, mittels “reand\_in\_data.m”. Für die Prognose des ausgewählten Aktienkurses gilt es jedoch hierbei, den jeweiligen Aktienkurs als letzten Aktienkurs in der Matrix aufzulisten. Zur Berechnung des PrNN wird das entsprechende Skript “single\_from\_multiple.m”(Quelltext B.13) ausgeführt. Die Modellbildung erfolgt dabei auf Basis aller Aktienkurse. Mittels des Skripts “predict.m” wird dabei das entsprechende Modell berechnet. Zu beachten ist dabei, dass die Anzahl der Zeitschritte  $M$ , die es zu prognostizieren gilt, dabei zunächst auf  $M=0$  gesetzt werden müssen. Die Modellbildung und Schätzung der Trainingsdaten erfolgt dabei analog zum der Vorgehensweise bei dem Einlesen und der Vorhersage aller Aktienkurse nach Formel 3.5. Demnach werden dabei die Trainingsdaten mithilfe aller 30 Aktienkurse geschätzt, jedoch zunächst nicht weiter Vorhergesagt.

Die Schätzungen der Trainingsdaten für den Aktienkurs von Adidas sind in Abbildung 3.10 abgebildet.



Abbildung 3.10: Geschätzter Aktienkurs mithilfe des PrNN (Einer aus allen)

Die Prognose der Testdaten erfolgt dabei mithilfe von “compute.m”. Jedoch geschieht dies nicht wie zuvor im Ausgabemodus sondern im Eingabemodus. Zur Prognose wird dabei eine neue Zustandsmatrix  $X^*$  für 152 Zeitschritte berechnet. Als Startvektor werden dabei die Werte aus der zuvor bestimmten Zustandsmatrix  $X$  zum Zeitpunkt  $t_{610}$  festgelegt. Die neue zu berechnende Matrix ist in Formel 3.6 dargestellt.

$$X^* = \begin{pmatrix} S_{1,610} & \dots & S_{1,762} \\ \vdots & \ddots & \vdots \\ S_{29,610} & \dots & S_{29,762} \\ R_{30,610} & \dots & R_{30,762} \\ \vdots & \ddots & \vdots \\ R_{230,610} & \dots & R_{230,762} \end{pmatrix}. \quad (3.6)$$

Dabei werden die ersten 29 Aktienkurse aus der Matrix aller eingelesenen Aktienkurse in die Zustandsmatrix  $X^*$  eingelesen. Die Zeile 30 wird dabei für die Prognose des Aktienkurses von Adidas nicht befüllt. Der Aktienkurs von Adidas wird damit als Teil des Reservoirs festgelegt und dementsprechend berechnet. Aus dem Startvektor  $x_0$  und der Gewichtsmatrix, werden die Reservoirzustände und damit auch der Aktienkurs von Adidas ermittelt und damit vorhergesagt.



## 4. Evaluierung

In diesem Abschnitt der Ausarbeitung erfolgt die Beurteilung der Prognosen mithilfe der zuvor vorgestellten Vorhersagemethoden. Das Ziel dabei ist es zu identifizieren, welches Modell in welcher Form im Vergleich die zuverlässigsten Vorhersagen der Aktienkurse ermöglicht. In diesem Zusammenhang wird in diesem Abschnitt der Einfluss der Anzahl der eingelesenen Beobachtungen der Zeitreihe, als auch speziell für das PrNN die Anzahl der Neuronen für das Reservoir, untersucht. Im ersten Schritt erfolgt dabei die Auswertung der Einzelkursvorhersage. Anschließend wird die Vorhersage aller Aktienkurs gleichzeitig mittels des PrNN analysiert. Das Ziel hierbei ist es, das Modell zu identifizieren, welches aus der Kombination der Anzahl der Daten und Neuronen die bestmöglichen Vorhersagen ermöglicht. Die gleiche Auswertung wird dabei für die Einzelkursvorhersage auf Basis aller anderen Aktienkurse durchgeführt.

Abschließend erfolgt die Gesamtauswertung der Vorhersagen. Dabei wird identifiziert, welches Modell mittels welcher Prognoseform die zuverlässigsten Vorhersagen ermöglicht. Das ermittelte beste Modell aus allen, wird im letzten Evaluationsschritt unter Berücksichtigung der Fundamentaldaten untersucht. Damit soll identifiziert werden, ob die entsprechende Vorhersage der Aktienkurse dadurch verbessert werden.

### 4.1 Die Auswertung der Einzelkursvorhersage

Die Evaluierung der Modelle für die Prognose eines einzelnen Aktienkurses wird am Beispiel des Aktienkurses von Adidas, welcher auch schon in Kapitel 3 zur Modellbildung hinzugezogen wurde, zunächst vorgestellt. Für die Auswertung der Vorhersagen werden jedoch die Prognosen aller in Abschnitt A.1 gelisteten Aktienkurse berücksichtigt. Mithilfe jedes Modells werden 152 Zeitschritte vorhergesagt, da dies

dem Umfang des Testdatensatzes entspricht. Damit wird der Vergleich zwischen der prognostizierten Zeitreihe und dem originalen Aktienkurs ermöglicht. Bewertet wird die Prognosegüte anhand des RMSE nach Formel 2.60, welcher die durchschnittliche Abweichung zwischen dem originalen Aktienkurs und dem entsprechend des Modells vorhergesagten Kurses errechnet.

Im ersten Schritt der Auswertung der Einzelkursvorhersage werden zunächst 762 Daten eingelesen, woraufhin die Anzahl der Daten auf 500 und anschließend auf 300 Daten reduziert wird und auf Basis dessen die Vorhersagen getroffen werden. Das Ziel dabei ist es zu bestimmen, welche Anzahl an Daten im Vergleich den Aktienkurs bestmöglich approximiert. Dabei wurde die Anzahl der Daten unabhängig von dem Startzeitpunkt innerhalb eines Handelsjahrs gewählt. Zudem wird die Anzahl der Beobachtungen mit einem Abstand von 200 Tagen verringert, um bestimmen zu können, ob tendenziell eine geringere Anzahl an Daten bessere Prognosen ermöglicht. Für alle Modelle bleibt dabei die Aufteilung der Trainings- und Testdaten zu einem Verhältnis von 80% für die Trainingsdaten und 20% für die Testdaten bestehen. Für das PrNN wird zudem die Anzahl der Neuronen für das Reservoir verändert, um eine optimale Reservoirgröße zu ermitteln.

#### 4.1.1 Die Lineare Regression

Die erste Beurteilung der Modellgüte mit insgesamt 762 eingelesenen Beobachtungen kann bereits schon graphisch durch die Abbildung der Vorhersagen im Vergleich zum realen Aktienkursverlauf erfolgen. Für das Modell der Linearen Regression ergibt sich nach dem in Abschnitt 3.2 aufgestellten Regressionsmodell  $\hat{y}(t) = 104.6698 + 0.1701 \cdot t$  für 610 Trainingsdaten ein prognostizierter Aktienkurs von Adidas, der in Abbildung 4.1 dargestellt ist.

Da das ermittelte Modell einen positiven y-Achsenabschnitt, als auch eine konstante Steigung aufweist, erhöht sich der prognostizierte Wert des Aktienkurses mit jedem Zeitschritt. Die Vorhersage für die 152 Handelstage zeigt, dass auf Basis der Regressionsfunktion der vorhergesagte Aktienkurs linear ansteigt. Mittels der Regressionsgerade wird ein aufsteigender Aktienkurs prognostiziert, wohingegen der reale Verlauf des Kurses von Adidas einen wechselnden Trend aufweist. Dabei liegt sogar zum Ende des Jahres 2018 ein absteigender Trend des Aktienkurses vor.

Für die Evaluierung wird hierbei der RMSE ermittelt, welcher die durchschnittliche Abweichung des prognostizierten Aktienkurses zum realen Kursverlauf beschreibt. Die Abweichung der Schätzungen der Trainingsdaten zum realen Verlauf des Aktienkurses beträgt im Durchschnitt einen Wert von 13.34 €. Die Vorhersage der Testdaten weist, wie schon in der Abbildung 4.1 deutlich wurde, sogar eine höhere durchschnittliche Abweichung zum Aktienkurs von Adidas auf. Diese beträgt dabei einen Wert von 24.98 €. Da die Regressionsfunktion auf Basis der Trainingsdaten errechnet wurde, sind die Abweichungen dementsprechend geringer.

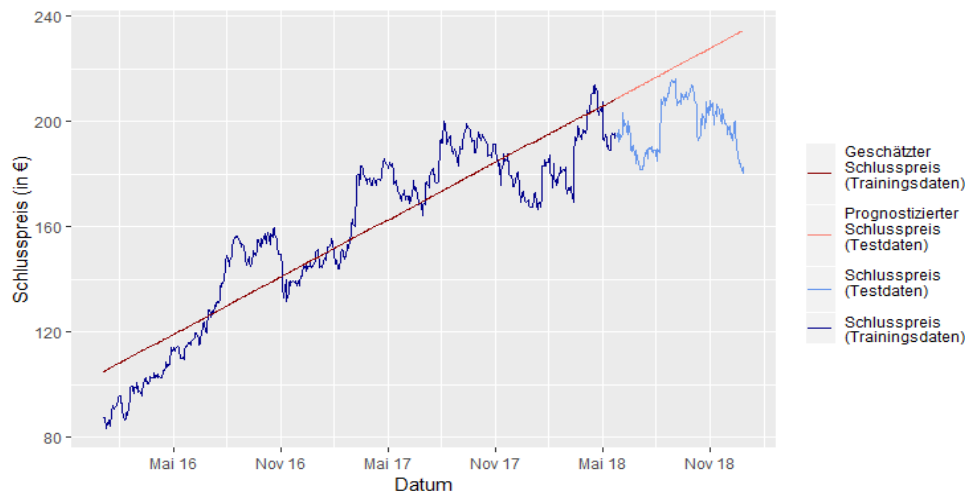


Abbildung 4.1: Vorhersage des Adidas-Aktienkurses mithilfe der Linearen Regression (762 Daten)

Sobald jedoch weitere Werte prognostiziert werden sollen, weisen die Vorhersagen höhere Abweichungen auf. Der bestimmte lineare Anstieg des Aktienkurses, welcher auf Grundlage der Trainingsdaten bestimmt wurde, liegt innerhalb des Testdatensatzes nicht vor.

Im Rahmen der Evaluation werden alle weiteren Aktienkurse mithilfe der Linearen Regression vorhergesagt. Damit kann im weiteren Verlauf der Ausarbeitung ein direkter graphischer Vergleich der Vorhersagen mit dem ARIMA- und SARIMA-Modell, sowie des PrNN erfolgen. Die Vorhersagen entsprechen den Prognosen mittels der erfassten Beobachtungen von drei Jahren. Die Abbildungen zur Vorhersage mit dem Linearen Regressionsverfahren sind im Anhang in Abschnitt C.1.1 aufgeführt. Dabei sind die entsprechenden Regressionsfunktionen bzw. Regressionsmodelle von  $\hat{y}$  innerhalb der Abbildungen aufgeführt.

Die Abbildung 4.1 zeigt, dass der Aktienkurs vor allem im ersten Jahr des Beobachtungszeitraumes ansteigt, jedoch im Laufe der Zeit abflacht. Eine mögliche Maßnahme zur Optimierung der Prognosen ist es damit, die Anzahl der Beobachtungen der Aktienkurse zu reduzieren, sodass nur die relevanten letzten Zeitschritte für die Modellbildung genutzt werden. Sollten die ersten Beobachtungen demnach nicht berücksichtigt werden, so verändert sich dementsprechend das Regressionsmodell. Zunächst sollen daher lediglich die letzten 500 Beobachtungen zur Erstellung des Regressionsmodells und Evaluation berücksichtigt werden. Dies entspricht den Beobachtungen der Zeitreihe mit einem Zeitraum vom 11.01.2017 bis zum 06.08.2017, d.h 400 Daten für den Trainingsdatensatz und vom 07.08.2018 bis zum 28.12.2018, also 100 Daten für den Testdatensatz. Damit ergibt sich eine Regressionsfunktion mit  $\hat{y} = 166.02155 + 0.07648t$ . Die Vorhersage des Aktienkurses ist in Abbildung 4.2



dargestellt. Die durchschnittlichen Abweichungen betragen bei dem Trainingsdatensatz  $RMSE_{Train} = 10.90$  und den Testdatensatz  $RMSE_{Test} = 10.06$ .

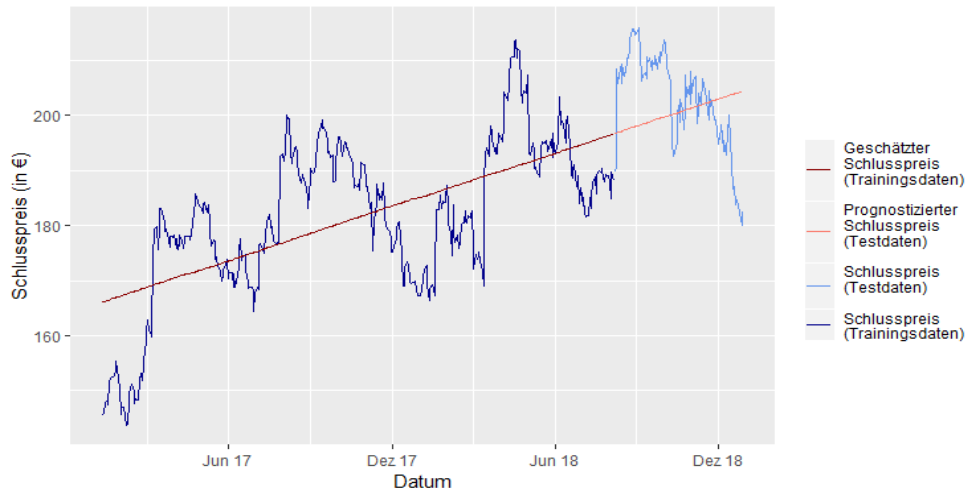


Abbildung 4.2: Vorhersage des Adidas-Aktienkurses mithilfe der Linearen Regression (500 Daten)

Im zweiten Testdurchlauf wird die Anzahl der Beobachtungen weiter auf 300 Daten des Aktienkurses reduziert. Der Trainingsdatensatz beinhaltet nun 240 Beobachtungen mit den Daten vom 23.10.2017 bis zum 01.10.2018. Der Testdatensatz beträgt 60 Daten vom 02.10.2018 bis zum 28.12.2018. Damit ergibt sich eine Regressionsfunktion mit  $\hat{y} = 174.4546 + 0.1312t$ . Auf Basis dessen ergibt sich der geschätzte und prognostizierte Aktienkurs in Abbildung 4.3.

Der in Abbildung 4.3 geschätzte Aktienkursverlauf zeigt die durchschnittlichen Abweichungen mit dem  $RMSE_{Train} = 9.57$  für den Trainings- und der  $RMSE_{Test} = 13.78$  für den Testdatensatz.

Bei dem Vergleich aller Modelle wird deutlich, dass hierbei der Trainingsdatensatz mit mit 300 Daten, also 240 Trainingsdaten im Vergleich bestmöglich abgebildet werden kann. Dennoch können auf Basis von 400 Trainingsdaten die zuverlässigsten Prognosen erzielt werden. Da jedoch ausgeschlossen werden muss, dass dieses Ergebnis nicht nur aufgrund des Aktienkursverlaufs ermittelt sondern für alle Aktienkurse gilt, werden für die in Abschnitt A.1 aufgeführten 30 Aktienkurse die Vorhersagen mithilfe der Linearen Regression durchgeführt. Die jeweiligen Abweichungen zwischen den geschätzten und prognostizierten Aktienkursverlauf zum realen Kursverlauf sind in der Tabelle im Anhang in Abschnitt C.1.2 aufgeführt. Dabei wird für jeden Aktienkurs der RMSE für jeweils den Trainings- und den Testdatensatz entsprechend der Anzahl der Beobachtungen erfasst. Zu beachten ist dabei, dass der RMSE nur für die Werte des jeweiligen Aktienkurses miteinander verglichen

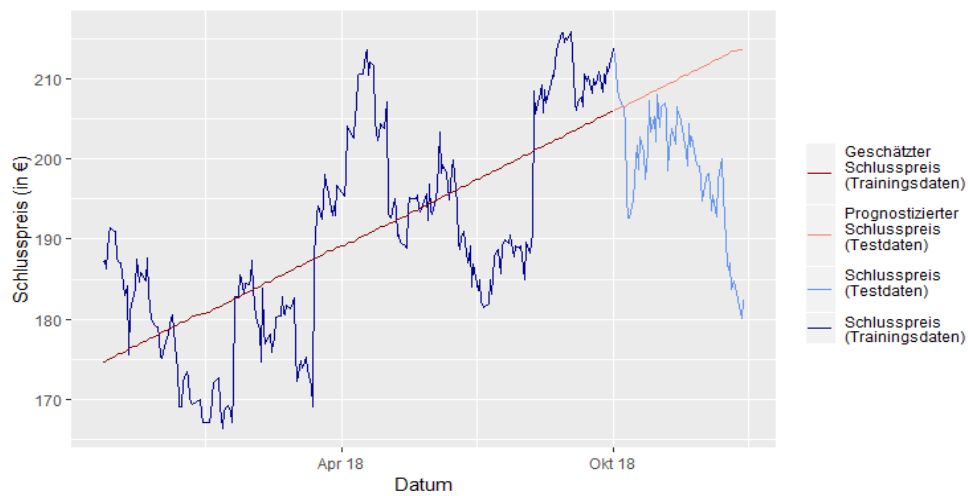


Abbildung 4.3: Vorhersage des Adidas-Aktienkurses mithilfe der Linearen Regression (300 Daten)

werden kann, jedoch nicht im Vergleich zu allen anderen Aktienkursen steht. Der Grund hierfür ist, dass ein Modell, welches einen RMSE von 10 für einen Aktienkurs mit einer Preisspanne zwischen 1000€ und 1500€ erzielt, als geeignetes Modell erachtet werden kann. Für einen Aktienkurs mit einer Preisspanne von z.B. 50€ bis 100€ wiederum, kann die Performance des Modells bzw. dessen Prognosen damit als unzureichend eingestuft werden.

Werden die errechneten Abweichungen der jeweiligen Aktienkurse zwischen den Schätzungen der Trainingsdaten und den Prognosen der Testdaten mit dem realen Aktienkursverlauf nach Abschnitt C.1.2 miteinander verglichen, wird deutlich, dass die Vorhersage mithilfe der Linearen Regression bei 300 Beobachtungen die bestmögliche Abbildung der Zeitreihe erzielt. Die Auswertung ist in Tabelle 4.1 zusammengefasst.

Tabelle 4.1: Auswertung der Linearen Regressionsmodelle nach Anzahl der Daten

Datensatz	Anzahl eingelesener Daten (D)		
	300 D	500 D	762 D
$E_{Train}$	19/30	6/30	5/30
$E_{Test}$	18/30	7/30	5/30

Die Tabelle zeigt dabei die Häufigkeit, entsprechend der Anzahl der eingelesenen Beobachtungen, mit der der geringste RMSE erzielt wurde. Das bedeutet für die erste Zeile und erste Spalte, dass bei 300 eingelesenen Daten, d.h. 240 Trainingsdaten, bei 19 von 30 Aktienkursen der geringste RMSE im Vergleich zu der Schätzung

der Trainingsdaten bei 500 oder 762 eingelesenen Daten, erzielt werden konnte. Für 19 von 30 Aktienkurse konnte damit bei 300 eingelesenen Daten der Trainingsdatensatz am besten geschätzt werden. Gleiches gilt für die Prognose der Testdaten. Mit 300 Beobachtungen ist der RMSE bei 18 von 30 Aktienkursen im Vergleich der RMSE mit dem Modell von 500 und 762 Daten geringer und ermöglicht damit die bestmöglichen Vorhersagen. Die im Vergleich schlechtesten Abbildungen der Trainingsdaten, als auch der Vorhersagen, werden auf Basis von 762 Beobachtungen, d.h. 610 Trainingsdaten getroffen.

Dementsprechend ergibt das Lineare Regressionsmodell auf Basis von 240 Trainingsdaten und 60 Testdaten hierbei die zuverlässigsten Vorhersagen der Aktienkurse.

### 4.1.2 Das ARIMA-Modell

Im Gegensatz zu dem Linearen Regressionsmodell, gestaltet sich die Modellbildung, wie in Abschnitt 3.3 beschrieben, bei dem ARIMA-Modell als aufwändiger. Aus dem errechneten ARIMA(2,1,2)-Modell und den in Tabelle 3.1 aufgeführten Koeffizienten, welche auf Basis von zunächst 762 eingelesenen Beobachtungen bzw. 610 Trainingsdaten ermittelt wurden, ergibt sich die Vorhersage des Aktienkurses von Adidas, welche in Abbildung 4.4 abgebildet ist.



Abbildung 4.4: Vorhersage des Adidas-Aktienkurses mit dem ARIMA-Modell (762 Daten)

Deutlich wird ein linearer Trendverlauf des prognostizierten Aktienkurses, welcher durch den Drift  $\mu = 0.1741$  bestimmt wird. Dieser bildet eine Konstante, welche für den gesamten Verlauf des prognostizierten Aktienkurses besteht und die Prognosen beeinflusst. Im Gegensatz zu der Schätzung der Trainingsdaten, werden die Komponenten des Moving Average-Prozesses bei der Prognose nicht berücksichtigt.

Der Grund hierfür ist, dass der reale Kursverlauf bei Prognosen nicht vorliegt. Damit können die Abweichungen zwischen den vergangenen geschätzten Werten der Testdaten und den Werten des Aktienkurses von Adidas mit  $\varepsilon_{t-1}$  und  $\varepsilon_{t-2}$  nicht bestimmt und damit der MA(2)-Prozess nicht berücksichtigt werden. Die Prognosen basieren demnach auf den letzten zwei prognostizierten Werte der Zeitreihe  $y_{t-1}$  und  $y_{t-2}$ , den errechneten Koeffizienten  $\phi_1, \phi_2$  und dem Drift. Damit ergibt sich für den Trainingsdatensatz der RMSE von  $RMSE_{train} = 2.47$  und für den Testdatensatz ein Wert von  $RMSE_{test} = 13.14$ .

Für einige Aktienkurse werden jedoch ARIMA(p,d,q)-Modelle identifiziert, bei denen ein ARIMA(0,1,0)-Prozess ermittelt und damit keine Korrelationen zwischen den einzelnen Beobachtungen sowie den Fehlerkomponenten identifiziert wurden. Damit ergibt sich aus dem AR(p)= 0, dem MA(q)= 0 und einer Differenzbildung der Zeitreihe der Ordnung d= 1 das zusammengesetzte ARIMA(0,1,0)-Modell für beispielsweise den Aktienkurs von RWE. Dieser ist in Abbildung 4.5 dargestellt.

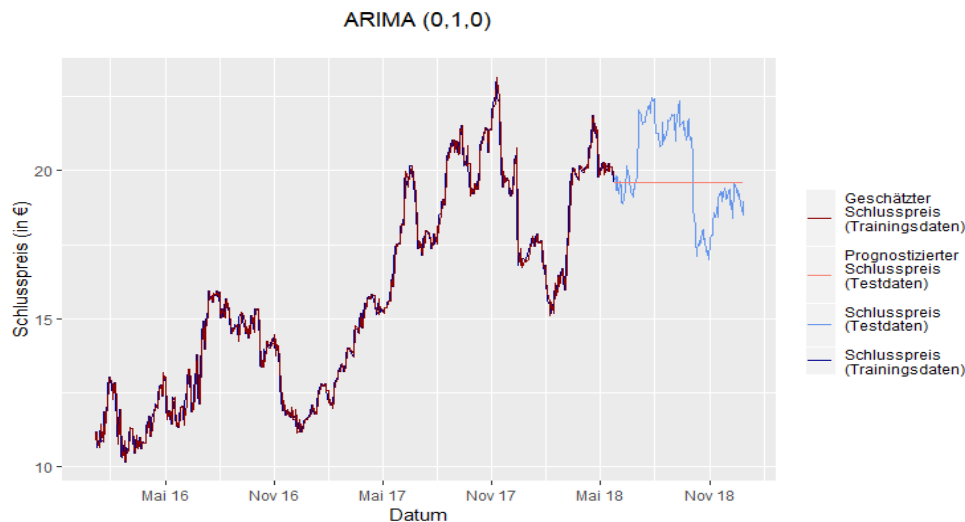


Abbildung 4.5: Vorhersage des RWE-Aktienkurses mit dem ARIMA-Modell (762 Daten)

Bei dem Aktienkurs handelt es sich demnach um einen Zufallsweg (Random Walk), welcher nichtstationär ist und von Zufallswerten beeinflusst wird. Dementsprechend wird für das ARIMA(0,1,0)-Modell der letzte Wert des Trainingsdatensatzes als konstanter Wert für die Prognose verwendet. Damit ergibt sich für alle 152 zu prognostizierenden Zeitschritte ein prognostizierter Schlusspreis von 19.605€, wodurch sich der Prognosefehler  $RMSE_{Train} = 0.35$  bzw.  $RMSE_{Test} = 1.58$  für die Vorhersage ergibt.

Die Abbildungen der Vorhersagen aller Aktienkurse mithilfe eines ARIMA(p,d,q)-Modells sind in Abschnitt C.2.1 abgebildet. Dabei sind die entsprechenden Ordnungen der Modelle sowie die errechneten Koeffizienten aufgeführt. Insgesamt ergeben

sich von insgesamt 30 Aktienkursen 14 Random Walk-Modelle, die einen konstanten Wert für die Prognose der Zeitreihe ergeben. Dabei wird wie zuvor beschrieben der jeweils letzte Wert des Trainingsdatensatzes für die Vorhersage der 152 Zeitschritte eingesetzt.

Wird die Anzahl der Beobachtungen auf 500 Daten und damit 400 Trainingsdaten und 100 Testdaten reduziert, so ergibt sich ein prognostizierter Aktienkurs für Adidas, welcher in Abbildung 4.6 dargestellt ist. Auch hierfür ergibt sich ein ARIMA(2,1,2)-Modell, wobei kein Drift identifiziert und veränderte Parameter für  $\phi$  und  $\theta$  errechnet werden.

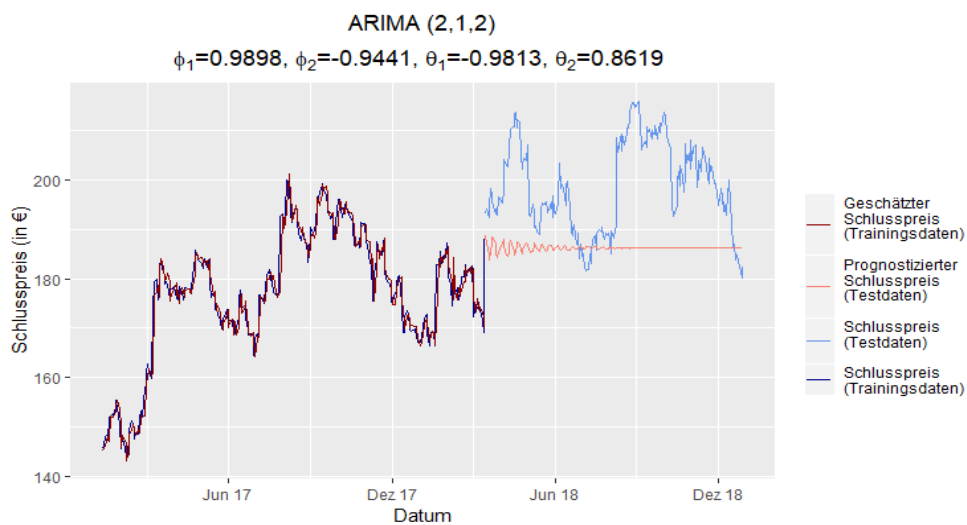


Abbildung 4.6: Vorhersage des Adidas-Aktienkurses mit dem ARIMA-Modell (500 Daten)

Der RMSE für den Trainingsdatensatz beträgt dabei eine durchschnittliche Abweichung von 2.80, wobei der RMSE für den Testdatensatz auf einen Wert von 17.32 ansteigt. Nach der Prognose aller 30 Aktienkurse ergibt sich dabei für 17 von 30 Aktienkursen ein Random Walk Modell. Für das Modell der auf 300 Beobachtungen reduzierten Zeitreihe, d.h. mit 240 Trainingsdaten und 60 Testdaten, ergibt sich ein ARIMA(p,d,q)-Modell, welches in Abbildung 4.7 aufgeführt ist.

Dabei handelt es sich wie zuvor beschrieben um ein Random Walk-Modell. Der prognostizierte Aktienkurs weist damit einen konstanten Wert von 181.85 € als Prognose auf. Damit ergibt sich für das Modell ein RMSE von 3.21 für den Trainingsdatensatz und 16.19 für den Testdatensatz. Für 19 von 30 Aktienkurse wurde dabei ein Random Walk-Modell ermittelt.

Damit ist das Modell für den Aktienkurs von Adidas mit 762 Daten für die Abbildung der Trainingsdaten, als auch für die Testdaten am besten geeignet. Zur Gesamtauswertung gilt es jedoch, alle 30 Aktienkurse bei der Prognose mit einem

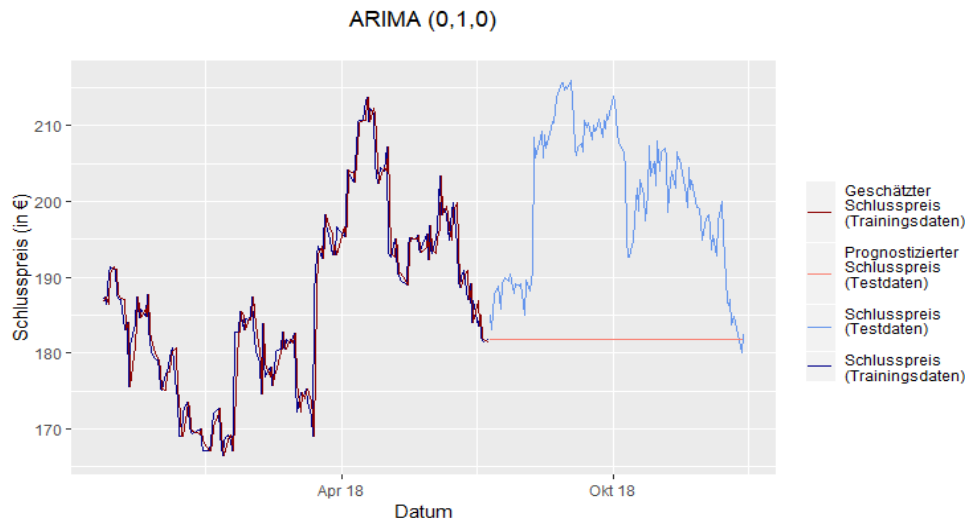


Abbildung 4.7: Vorhersage des Adidas-Aktienkurses mit dem ARIMA-Modell (300 Daten)

ARIMA(p,d,q)-Modell zu berücksichtigen. Für jeden Aktienkurs wird zur Auswertung dabei jeder errechnete RMSE, entsprechend der Anzahl der eingelesenen Beobachtungen, miteinander verglichen. Für den Aktienkurs von Adidas weist beispielsweise das ARIMA(2,1,2)-Modell den geringsten RMSE zu den Trainings- und Testdaten mit 300 eingelesenen Beobachtungen im Vergleich zu den Modellen auf Basis der 500 und 762 Daten auf. Damit kann dieser Aktienkurs auf Basis von 300 Daten bestmöglich geschätzt werden. Dieser Vergleich erfolgt für jeden Aktienkurs. Das Ergebnis des Vergleichs ist in Tabelle 4.2 aufgeführt.

Tabelle 4.2: Auswertung der ARIMA-Modelle nach Anzahl der Daten

Datensatz	Anzahl eingelesener Daten (D)		
	300 D	500 D	762 D
$E_{Train}$	3/30	11/30	16/30
$E_{Test}$	12/30	10/30	8/30

Bei der Analyse der Daten kann festgelegt werden, dass der Trainingsdatensatz tendenziell mit einer hohen Anzahl an Beobachtungen eine bessere Abbildung der Zeitreihe ermöglicht. Mithilfe von 762 Beobachtungen, d.h. 610 Trainingsdaten, konnten für 16 von 30 Aktienkurse im Vergleich zu den anderen Modellen jeweils die besten Schätzungen erzielt werden. Sobald auf Basis der festgelegten ARIMA(p,d,q)-Modelle jedoch die Vorhersage erfolgt, ermöglichen diese bei 762 Beobachtungen lediglich für 8 der 30 Aktienkurse im Vergleich die beste Prognose. Damit werden mithilfe von 610 Trainingsdaten die schlechtesten Prognosen erzielt. Die zuverlässigsten Vorhersagen ermöglicht das Modell mit 300 Beobachtungen und damit auf

Basis von 240 Trainingsdaten. Damit wird deutlich, dass gute Schätzungen in Bezug auf die Trainingsdaten kein Indiz für die bestmöglichen Prognosen ist.

### 4.1.3 Das SARIMA-Modell

Sollte eine Zeitreihe ein saisonales Muster aufweisen, welches sich innerhalb bestimmter Perioden wiederholt, so kann eine Vorhersage der Zeitreihen mithilfe des saisonalen  $ARIMA(p,d,q)(P,D,Q)_m$ -Modells erfolgen. Für die Berechnung der Saisonalität wird dabei ein sich jährlich wiederholendes saisonales Muster angenommen. Für den Aktienkurs ergibt sich damit ein geschätzter bzw. prognostizierter Kursverlauf, welcher in Abbildung 4.8 dargestellt ist. Für diesen Aktienkurs wird ein  $ARIMA(0,1,0)(0,1,0)_{254}$ -Modell ermittelt.

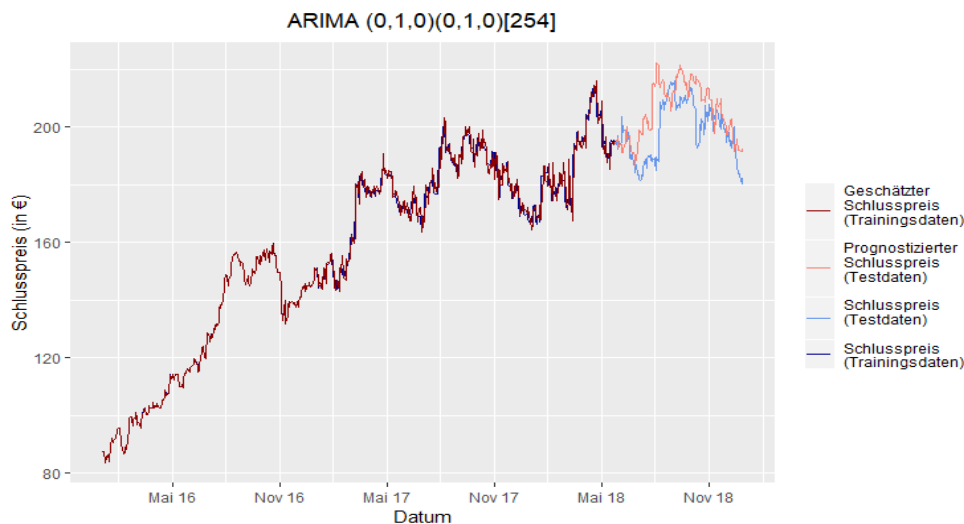


Abbildung 4.8: Vorhersage des Adidas-Aktienkurses mit dem SARIMA-Modell (762 Daten)

Unter dem Einsatz der Funktion “auto.arima” wird für die ermittelte Zeitreihe der saisonalen Komponente, als auch für die von der saisonalen Komponente bereinigten Zeitreihe, eine nichtstationäre Zeitreihen identifiziert. Dabei handelt es sich demnach um Zeitreihen mit einer Einheitswurzel, die in stationäre Zeitreihen transformiert werden. Für den  $AR(p)$ - und den  $MA(q)$ -Prozess ergeben sich dabei die Modelle  $AR(0)$  und  $MA(0)$ , da keinerlei Korrelationen zwischen den Werten der Zeitreihe erkannt wurden. Bei dem Aktienkurs von Adidas handelt es sich damit um ein saisonales Random-Walk-Modell. Zur Prognose der Zeitreihe wird dabei nicht nur, wie zuvor bei dem  $ARIMA(p,d,q)$ -Modell ermittelt, der letzte Wert des Trainingsatzes hinzugezogen. Bei einem saisonalen Random-Walk-Modell wird als Prognose hierzu das saisonale Muster berücksichtigt. Wie in Abbildung 4.8 wird als Prognose das saisonale Muster wiederholt.

Da nicht alle Zeitreihen jährlich-saisonale Regelmäßigkeiten aufweisen, kann nicht für jede Zeitreihe im Rahmen der Testdurchläufe ein entsprechendes SARIMA-Modell auf Basis von 610 Beobachtungen ermittelt werden. Für einige Modelle wird anstelle einer saisonalen Komponente ein verändertes ARIMA(p,d,q)-Modell ermittelt, wohingegen für einige die zuvor errechneten ARIMA(p,d,q)-Modelle als am besten geeignete Modelle bestehen bleiben. Die entsprechende Modellform für die jeweiligen Aktienkurse mit einem neu ermittelten Modell werden in Tabelle 4.3 durch ein  $\times$  zugeordnet.

Tabelle 4.3: Die Zuordnung der Modellformen

Unternehmen	Modell ohne saisonaler Komponente	Modell mit saisonaler Komponente
Adidas		$\times$
Airbus		$\times$
Alphabet	$\times$	
Audi	$\times$	
Beiersdorf	$\times$	
Bertrandt	$\times$	
BMW	$\times$	
Continental	$\times$	
Daimler		$\times$
ExxonMobil	$\times$	
Infineon	$\times$	
Lindt & Sprüngli		$\times$
Microsoft	$\times$	
Nike		$\times$
RWE		$\times$
SAP	$\times$	
Volkswagen	$\times$	
Wirecard	$\times$	

Aus den 30 vorliegenden Aktienkursen wird mittels der festgelegten Frequenz für 18 Aktienkurse ein neues Modell ermittelt. Davon wird für sieben der 19 Aktienkurse eine saisonale Komponente ermittelt, wohingegen für elf Aktienkurse das ARIMA(p,d,q)-Modell aufgrund der festgelegten Frequenz neu bestimmt wurde. Die entsprechenden Kursverläufe bei 762 Daten sind als Beispiel in Abschnitt C.3 aufgeführt.

Sobald die Anzahl der Trainingsdaten auf 400 aus insgesamt 500 Daten sowie auf 240 bei 300 eingelesenen Daten reduziert wird, erhält man jeweils weniger als zwei Perioden, die zur Modellbildung analysiert und verglichen werden können. Demnach kann bei einer verringerten Anzahl keine saisonale Komponente identifiziert werden, sodass die in Abschnitt 3.3.2 beschriebene “stl” Funktion zur Identifizierung der



saisonalen Komponente nach der Funktion “auto.arima” nicht eingesetzt wird. Dennoch ergeben sich für einige Aktienkurse aufgrund der festgelegten Frequenz von 254 Daten veränderte ARIMA( $p,d,q$ )-Modelle. Die errechneten Abweichungen zwischen den mittels der ermittelten Modelle geschätzten Trainingsdaten, den prognostizierten Testdaten und den jeweiligen originalen Aktienkursen sind in Abschnitt C.3.1 aufgeführt.

Bei dem Vergleich der durchschnittlichen Abweichungen  $RMSE_{train}$  und  $RMSE_{test}$  der SARIMA- und ARIMA-Modelle wird deutlich, dass für die meisten Aktienkurse die zuvor bestimmten ARIMA-Modelle dennoch bessere Vorhersagen ermöglichen. Der Vergleich zwischen den beiden Modellen ist in Tabelle 4.4 aufgeführt. Die erste Zeile zeigt dabei die Anzahl der neu bestimmten SARIMA bzw. ARIMA-Modelle aus den 30 Aktienkursen entsprechend der Anzahl der eingelesenen Beobachtungen auf. Demnach werden beispielsweise für 18 von 30 Aktienkursen bei 610 Trainingsdaten veränderte Modelle ermittelt. Die zweite Zeile der Tabelle zeigt dabei die Anzahl der SARIMA-Modelle, die zuverlässigere Vorhersagen im Vergleich zu dem zuvor errechneten ARIMA-Modell erzielen. Die ermittelten SARIMA-Modelle ermöglichen dabei für fünf der 18 SARIMA-Modelle eine bessere Approximation der Testdaten, als die zuvor errechneten ARIMA-Modelle.

Tabelle 4.4: Anzahl verbesserter ARIMA-Modelle

	Anzahl eingelesener Daten (D)		
	300 D	500 D	762 D
Anzahl veränderter Modelle	8/30	11/30	18/30
Verbessert durch SARIMA-Modell	1/8	0/11	5/18

Bei der Auswertung der Tabelle wird deutlich, dass sobald mehr als eine Periode der Zeitreihe vorliegt und damit eine saisonale Komponente identifiziert werden kann, einige SARIMA-Modelle eine Prognose ermöglichen, die die vorliegenden Zeitreihen besser approximiert, als die zuvor errechnete ARIMA-Modelle. Bei einer Anzahl von 400 Trainingsdaten und damit 500 Gesamtdaten werden 11 neue ARIMA-Modelle ermittelt, wobei keines der neuen Modelle zu einer Verbesserung der Prognose beiträgt. Bei 240 Trainingsdaten kann jedoch für ein Aktienkurs eine zuverlässigere Prognose erstellt werden.

Für den weiteren Verlauf der Ausarbeitung und den Vergleich mit den anderen Modellen, werden die zuvor bestehenden ARIMA-Modelle mit den SARIMA-Modellen ersetzt, die eine zuverlässigere Vorhersage der jeweiligen Aktienkurse ermöglichen. Damit wird zur Auswertung der Methoden und Modelle eine Kombination der ARIMA- und SARIMA-Modelle verwendet. Hierzu wird die aus Abschnitt C.2.2 ermittelte Tabelle der durchschnittlichen Abweichungen zwischen den Trainings- und Testdaten und dem realen Kursverlauf, welche mittels der ARIMA-Modelle ermittelt

wurden, mit den entsprechenden durchschnittlichen Abweichungen der SARIMA-Modelle korrigiert. Damit ergibt sich als finale Tabelle der ARIMA/SARIMA Vorhersagemethode die Tabelle in Abschnitt C.3.2. Zudem verändert sich die für die ARIMA-Modelle ermittelte Tabelle 4.2. Korrigiert wird diese mit der Anzahl der besten Prognosen und Schätzungen hinsichtlich der Anzahl der Beobachtungen der SARIMA-Modelle. Es ergibt sich die Tabelle 4.5.

Tabelle 4.5: Auswertung der Vorhersagen der Modelle ARIMA/SARIMA

Datensatz	Anzahl eingelesener Daten (D)		
	300 D	500 D	762 D
$E_{Train}$	1/30	12/30	17/30
$E_{Test}$	12/30	9/30	9/30

Die Tabelle verändert sich damit insofern, dass mittels eines SARIMA-Modells auf Basis von 610 Trainingsdaten für ein Aktienkurs zuverlässigere Vorhersagen als auf Basis von zuvor 400 Trainingsdaten mit dem ARIMA-Modell. Dennoch werden die besten Prognosen mit den Modellen auf Basis von 300 Daten erzielt.

#### 4.1.4 Das Predictive Neural Network

Für das PrNN soll im Rahmen der Evaluation in Kombination der einzulesenden Daten ebenfalls untersucht werden, inwiefern sich die Anzahl der Reservoirneuronen auf die Vorhersage auswirkt. Dabei wird jeweils die Anzahl der Neuronen auf 200, 400 und 600 Neuronen manuell festgelegt. Wie schon bei der Anzahl der Beobachtungen soll identifiziert werden, welche Anzahl an Neuronen tendenziell besser für die Prognosen geeignet ist. Untersucht werden soll dabei, welche Auswirkungen der Einsatz beispielsweise von 600 Neuronen auf die Modellbildung bei 610 Daten und die entsprechende Prognose von 152 hat. Die Auswirkung soll jedoch ebenfalls auf eine stark verringerte Anzahl an Beobachtungen untersucht werden.

Bei einer Anzahl von 762 Beobachtungen, die eingelesen werden, ergibt sich bei 200 Neuronen ein vorhergesagter Aktienkurs, welcher in Abbildung 4.9 abgebildet ist. Die Vorhersage ist dabei das Ergebnis mit dem geringsten RMSE für den Testdatensatz aus 20 Testdurchläufen. Dabei ergeben sich die Schätzungen bzw. Prognosen mit den durchschnittlichen Abweichungen zu der Zeitreihe mit  $RMSE_{Training} = 9.61$  und  $RMSE_{Testing} = 9.81$ . Sobald die Anzahl der Neuronen auf 400 erhöht wird, ergibt sich der Aktienkurs in Abbildung 4.10, für den der  $RMSE_{Training} = 7.12$  für den Trainingsdatensatz und der  $RMSE_{Testing} = 9.11$  für den Testdatensatz ermittelt werden.

Mithilfe von 600 Reservoirneuronen wird die Prognose in Abbildung 4.11 bestimmt, bei der die durchschnittliche Abweichung der geschätzten Trainingsdaten zu dem



Abbildung 4.9: Vorhersage des Adidas-Aktienkurses mit dem PrNN (762 Daten - 200 Neuronen)



Abbildung 4.10: Vorhersage des Adidas-Aktienkurses mit dem PrNN (762 Daten - 400 Neuronen)

realen Verlauf einen Wert von 6.59 und für die Prognose eine durchschnittliche Abweichung von 9.37 annimmt.

Deutlich wird dabei, dass mit einer steigenden Anzahl der Neuronen die Approximation der geschätzten Zeitreihe an die Trainingsdaten besser wird. Die im Vergleich schlechteste Schätzung der Trainingsdaten wird mit einem RMSE von 9.61 durch das PrNN mit lediglich 200 Neuronen errechnet. Die Schätzungen des PrNN mit 400 Neuronen erzielt mit einem RMSE von 7.12 ebenfalls eine im Vergleich zu dem PrNN



Abbildung 4.11: Vorhersage des Adidas-Aktienkurses mit dem PrNN (762 Daten - 600 Neuronen)

mit 200 Neuronen gute Approximation an die Zeitreihe mit den Trainingsdaten. Vergleicht man die Abweichung der errechneten Vorhersagen für den Testdatensatz mit dem des Trainingsdatensatz, so ergibt sich eine veränderte Reihenfolge in Bezug auf die Zuverlässigkeit. Das beste Ergebnis ermöglicht dabei das PrNN mit 400 Neuronen und einem RMSE von 9.11, woraufhin das PrNN mit 600 Neuronen mit einem RMSE von 9.37 folgt. Die im Vergleich schlechteste Vorhersage wird durch das PrNN mit 200 Neuronen errechnet. Hierbei liegt der RMSE bei 9.81. Da die Ergebnisse von den zufällig gewählten Gewichtswerten für  $W_{in}$  und  $W_{res}$  beeinflusst werden, muss jedoch in diesem Zusammenhang ausgeschlossen werden, dass das PrNN mit den 400 Neuronen zufällig die beste Vorhersage ermöglicht. Hierzu werden die Schätzungen des Trainingsdatensatzes und die Vorhersagen des Testdatensatzes für alle weiteren Aktienkurse bestimmt. Zudem wird in diesen Testdurchläufen die Anzahl der einzulesenden Beobachtungen verändert. Dabei wird wie schon zuvor, zwischen 762, 500 und 300 Beobachtungen unterschieden. Die Abbildungen bei der Vorhersage auf Grundlage von 762 Daten sind zum Vergleich aller anderen Prognosemethoden in Abschnitt C.4.1 aufgeführt.

Die Gesamtübersicht der durchschnittlichen Abweichungen ist für jeden Aktienkurs in Abschnitt C.4.2 aufgeführt. Dabei werden der RMSE für die Trainingsdaten mit  $E_{Train}$  bzw. für die Testdaten mit  $E_{Test}$  erfasst. Für jede Anzahl der Neuronen werden die durchschnittlichen Abweichungen entsprechend der Anzahl der eingelesenen Daten berechnet. Das Ziel der Auswertung des Modells ist es, das PrNN mit einer Kombination aus Neuronen und Beobachtungen zu wählen, welches jeweils für den Trainings- und Testdatensatz den geringsten RMSE erzielt. Berücksichtigt werden

dabei die Ergebnisse aller Aktienkurse. Das Ergebnis ist in Tabelle 4.6 zusammengefasst.

Tabelle 4.6: Auswertung der Einzelkursvorhersage (PrNN) nach Anzahl der Daten und Neuronen

Datensatz	Anzahl Neuronen (N) zur Anzahl eingelesener Daten (D)								
	200 N			400 N			600 N		
	300D	500D	762D	300D	500D	762D	300D	500D	762D
$E_{Train}$	0/30	0/30	0/30	0/30	0/30	0/30	30/30	0/30	0/30
$E_{Test}$	11/30	5/30	6/30	1/30	2/30	2/30	0/30	2/30	1/30

Bei der Auswertung der Tabelle in Abschnitt C.4.2 nach Tabelle 4.6 wird deutlich, dass das PrNN mit  $N = 600$  und auf Grundlage von 300 eingelesenen Daten im Durchschnitt den niedrigsten RMSE für die Abbildung der Trainingsdaten aufweist. Auf Basis dieser Kombination können damit die Trainingsdaten aller 30 Aktienkurse besser abgebildet werden, als mit jeder anderen Kombination aus der Anzahl der Neuronen und Beobachtungen. Bei der Abbildung der Testdatensätze und damit der Vorhersagen, erzielt jedoch das PrNN mit einer Kombination aus 200 Neuronen und 300 Daten für den Großteil der Aktienkurse die bestmöglichen Annäherungen des Kursverlaufs. Für 11 von insgesamt 30 Aktienkursen kann mittels des gebildeten Modells der geringste RMSE der Testdaten erzielt werden.

Für den Vergleich der einzelnen Prognosemethoden, werden in der Gesamtauswertung im weiteren Verlauf der Ausarbeitung die Ergebnisse des PrNN mit 200 Neuronen und 300 Daten berücksichtigt.

## 4.2 Die Auswertung der Vorhersage aller Aktienkurse gleichzeitig

Wie schon in Kapitel 3 zuvor erläutert, ist das Einlesen aller Aktienkurse sowie die Vorhersage dieser zum gleichen Zeitpunkt lediglich mit dem PrNN möglich. Entsprechend wie bei der Vorhersage eines einzelnen Aktienkurses, soll in diesem Teil der Ausarbeitung untersucht werden, inwiefern sich die Anzahl der Beobachtungen und die Anzahl der Neuronen auf die Vorhersage aller 30 Aktienkurse auswirkt.

Zunächst wird hierbei, wie schon bei der Einzelkursvorhersage, der Einfluss der Anzahl der Neuronen anhand des Aktienkurses von Adidas vorgestellt. Damit ist hierzu schon ein erster direkter Vergleich zu den anderen Vorhersagemethoden möglich. Die Vorhersage erfolgt dabei auf Grundlage von 762 Daten. Lediglich die Anzahl der Neuronen wird hierbei verändert. Die erste Prognose für den Aktienkurs von Adidas erfolgt dabei mit einem PrNN aus einer Kombination von 762 Daten und

200 Neuronen. Die Schätzungen der Trainingsdaten, welche auch schon in Kapitel 3 vorgestellt wurde, und die Prognose für den Aktienkurs von Adidas sind in Abbildung 4.12 dargestellt.

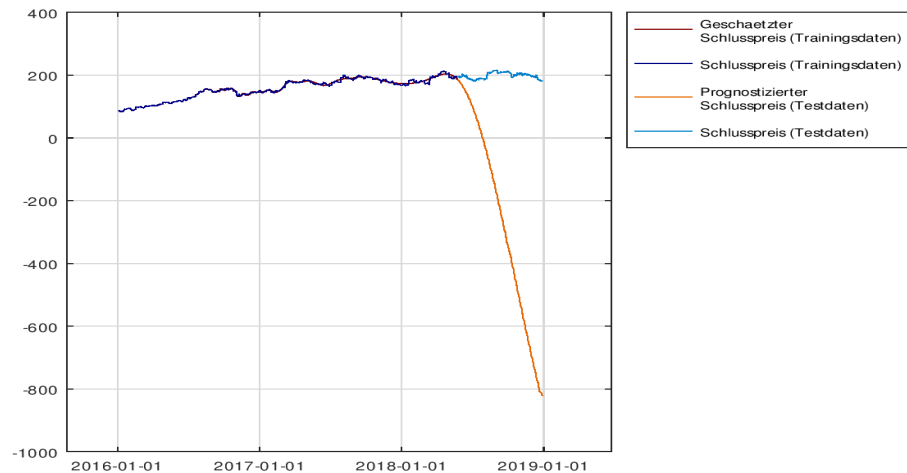


Abbildung 4.12: Vorhersage des Adidas-Aktienkurses mit dem Multi-PrNN (762 Daten - 200 Neuronen)

Bei dem geschätzten Aktienkurs nach Abbildung 4.12 wird ein RMSE für die Trainingsdaten mit  $\text{RMSE}_{train} = 3.85$  ermittelt. Für den Testdatensatz ergibt sich dabei ein RMSE von  $\text{RMSE}_{test} = 301.70$ . Deutlich wird dabei, dass die prognostizierten Werte der Vorhersage rasant abnehmen.

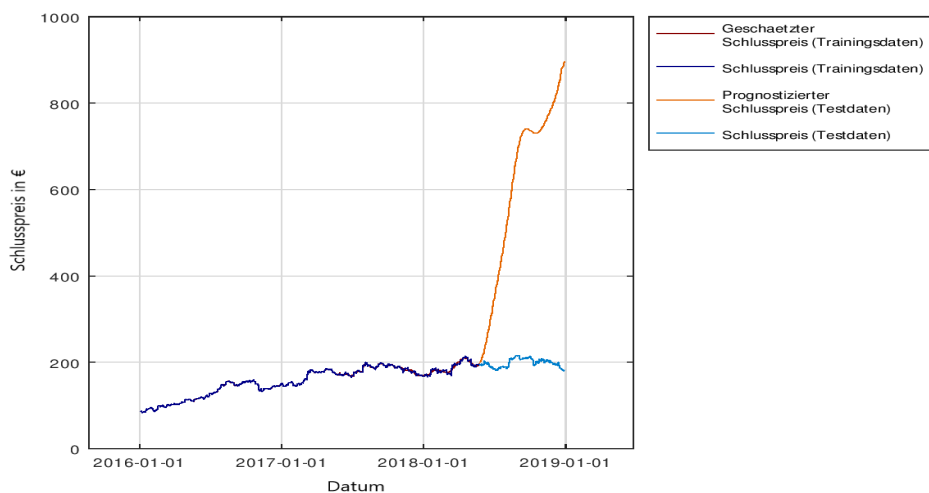


Abbildung 4.13: Vorhersage des Adidas-Aktienkurses mit dem Multi-PrNN (762 Daten - 400 Neuronen)

Mit einer steigenden Neuronenzahl auf 400 Neuronen und 762 Daten ergibt sich ein prognostizierter Aktienkurs, welcher in Abbildung 4.13 aufgeführt ist. Die Schätzungen der Trainingsdaten wird hierbei verbessert, wohingegen die Zuverlässigkeit der Abbildung der Testdaten hierbei mit einer steigenden Anzahl an Neuronen abnimmt. Dabei werden die durchschnittlichen Abweichungen zwischen dem prognostizierten Aktienkurs und dem originalen Aktienkurse mit  $RMSE_{train} = 2.02$  und  $RMSE_{test} = 453.14$ . Die Zuverlässigkeit der Vorhersagen ändert sich ebenfalls bei einem PrNN mit 600 Neuronen. Die Abbildung der Trainings- und Testdaten kann mittels des Modells verbessert werden. Hierzu ergibt sich für die Trainingsdaten ein RMSE mit  $RMSE_{train} = 2.5 * 10^{-7}$  und für die Testdaten der  $RMSE_{test} = 283.60$ . Damit können mithilfe dieses Modells die beste Approximation vor allem der Trainingsdaten, als auch der Testdaten im Vergleich zu den anderen Kombinationsmöglichkeiten bei der Vorhersage aller Kurse gleichzeitig erzielt werden. Die Vorhersage des Aktienkurses von Adidas mit einem PrNN aus 600 Neuronen und 762 Daten ist in Abbildung 4.14 aufgeführt.

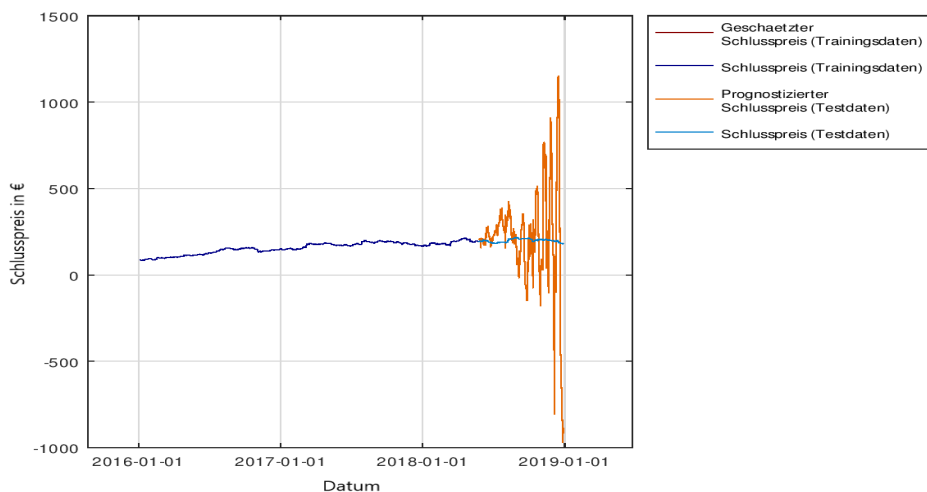


Abbildung 4.14: Vorhersage des Adidas-Aktienkurses mit dem Multi-PrNN (762 Daten - 600 Neuronen)

Um jedoch beurteilen zu können, welche Anzahl an Neuronen tendenziell besser für die multiple Vorhersage ist, werden die Vorhersagen für alle anderen 29 Aktienkurse durchgeführt. Getestet werden dabei für das PrNN alle genannten Kombinationen aus der Anzahl der Daten und Neuronen. Berücksichtigt werden dabei 762, 500 und 300 Daten, sowie 200, 400 und 600 Neuronen. Die entsprechenden durchschnittlichen Abweichungen durch den RMSE sind in Abschnitt C.4.3 aufgeführt. Wie auch schon bei den anderen Methoden, werden diese Abweichungen miteinander verglichen und für jeden Aktienkurs das beste PrNN entsprechend der Kombination aus Neuronen und Daten ermittelt. Der Vergleich ist in Tabelle 4.7 zusammengefasst.

Tabelle 4.7: Auswertung der Multi-Vorhersage (PrNN) nach Anzahl der Daten und Neuronen

Datensatz	Anzahl Neuronen (N) zur Anzahl eingelesener Daten (D)								
	200 N			400 N			600 N		
	300D	500D	762D	300D	500D	762D	300D	500D	762D
$E_{Train}$	0	0	0	2	0	0	28	0	0
$E_{Test}$	6	0	2	4	0	1	17	0	0

Die Tabelle zeigt dabei die Anzahl der Reservoirneuronen im Zusammenhang mit 300, 500 und 762 Beobachtungen. Die erste Zeile gibt dabei für jede Kombination die Anzahl der Aktienkurse aus, die dabei im Vergleich zu allen anderen Kombinationen die besten Schätzungen berechnen. Die zweite Zeile enthält die entsprechende Anzahl für die im Vergleich besten Prognosen. Beispielsweise kann aus der Tabelle abgelesen werden, dass für 2 von 30 Aktienkursen mit einem PrNN von 400 Neuronen und 300 Daten, im Vergleich zu den anderen Kombinationen, Schätzung des Trainingsdatensatz mit dem geringsten RMSE erzielt werden konnten.

Im Vergleich aller Testdurchläufe wird deutlich, dass der Trainingsdatensatz, als auch der Testdatensatz im Durchschnitt bestmöglich mit einer Kombination aus einem PrNN mit 600 Neuronen und 300 einzulesende Beobachtungen abgebildet werden kann. Dabei konnten bei 28 von 30 Aktienkursen der Trainingsdatensatz und bei 17 von 30 Aktienkursen der Testdatensatz am besten approximiert werden. Da die Anzahl von 300 Beobachtungen und 600 Neuronen zu den bestmöglichen Prognosen führen, wird dieses Modell bei der Gesamtauswertung im weiteren Verlauf der Arbeit berücksichtigt.

### 4.3 Die Auswertung der Vorhersage eines einzelnen Aktienkurses aus allen

Die Vorhersage eines einzelnen Aktienkurses auf Grundlage von allen Kursen ist wie schon in Kapitel 3 ermittelt, lediglich mittels des Predictive Neural Network möglich. Nach dem in Kapitel 3 beschriebenen Aufbau, basiert die Modellbildung des PrNN dabei auf allen Aktienkursen, wohingegen für die Prognose bis auf den zu prognostizierenden Aktienkurs alle Aktienkurse berücksichtigt werden. Wie schon bei den zuvor ermittelten Vorhersagemethoden, wird hierbei die Zuverlässigkeit der Prognosen nach der Anzahl der einzulesenden Beobachtungen untersucht. Dabei wird die Vorhersage mit insgesamt 300 Daten, 500 Daten und 762 Daten durchgeführt. Neben der Anzahl der Beobachtungen der Zeitreihe, wird ebenfalls der Einfluss der Neuronenanzahl des Reservoirs von 200 Neuronen, 400 Neuronen und 600 Neuronen untersucht. Damit wird in diesem Abschnitt der Ausarbeitung für jeden zu prognostizierenden Aktienkurs der RMSE der Trainings- und Testdaten für jede Kombination der Neuronen- und Datenanzahl ermittelt. Das Ergebnis wird hierbei zunächst



für einen ersten direkten Vergleich der Vorhersage mit allen anderen Prognosemethoden am Beispiel von Adidas aufgezeigt. Die Anzahl wird dabei, wie schon bei den anderen Prognosemethoden, auf 762 Beobachtungen, d.h. 610 Trainingsdaten festgelegt. Für jeden Aktienkurs wird die Prognose 20 Mal durchgeführt und daraus das Modell mit der besten Vorhersage für den jeweils zu untersuchenden Aktienkurs ausgewählt.

Die Prognose des Aktienkurses von Adidas durch alle anderen 29 Aktienkurse, erfolgt zunächst bei 610 Trainingsdaten und 152 Testdaten mit einem Reservoir des PrNNs von 200 Neuronen. Dieses ist in Abbildung 4.15 dargestellt.

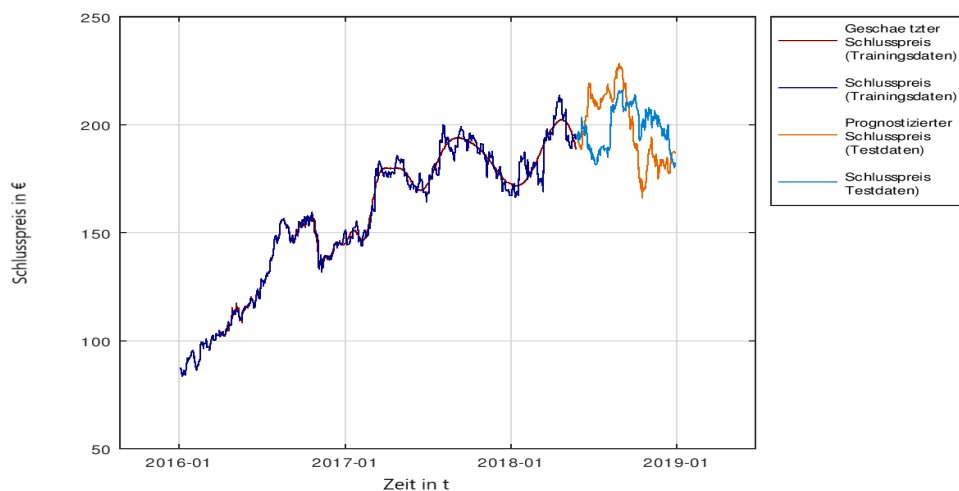


Abbildung 4.15: Vorhersage des Adidas-Aktienkurses mit dem PrNN (Einer aus allen, 762 Daten - 200 Neuronen)

Der RMSE für den Trainingsdatensatz beträgt dabei eine durchschnittliche Abweichung von  $RMSE_{train} = 3.85$  und für den Testdatensatz  $RMSE_{test} = 18.67$ . Sobald die Anzahl an Neuronen auf 400 Neuronen erhöht wird, wird deutlich, dass die Approximation der geschätzten Werte durch das PrNN für den Trainingsdatensatz verbessert wird. Hierbei verringert sich die durchschnittliche Abweichung auf einen Wert von 2.17. Jedoch wird die Prognose auf Basis des ermittelten Modells schlechter. Der RMSE für den Testdatensatz beträgt dabei einen Wert von  $RMSE_{test} = 20.33$ . Dieser geschätzte Aktienkurs von Adidas bei 400 Neuronen ist in Abbildung 4.16 aufgeführt.



Abbildung 4.16: Vorhersage des Adidas-Aktienkurses mit dem PrNN (Einer aus allen, 762 Daten - 400 Neuronen)

Wird die Anzahl der Neuronen auf 600 erhöht, erfolgt die im Vergleich schlechteste Prognose bei der Einzelkursvorhersage aus allen Aktienkursen mit den durchschnittlichen Abweichungen von  $RMSE_{train} = 0.32$  und  $RMSE_{test} = 184.47$ . Die Prognose ist in Abbildung 4.17 zu sehen.

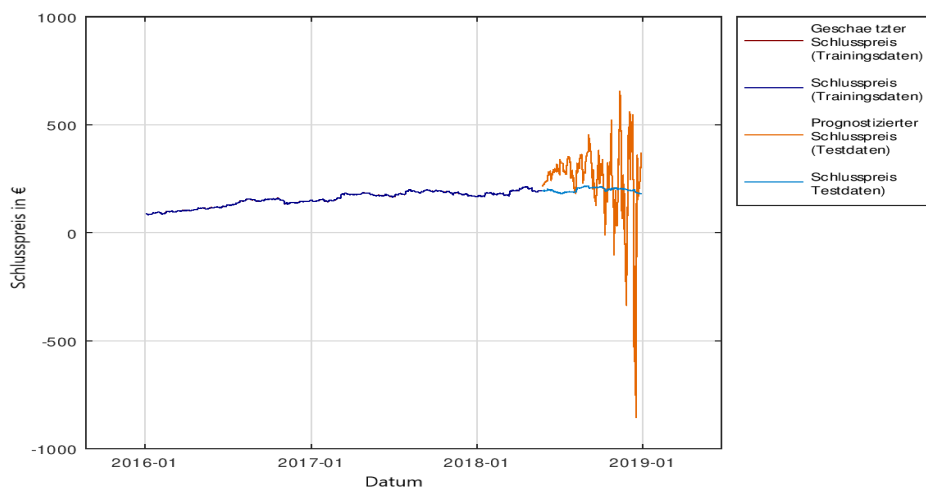


Abbildung 4.17: Vorhersage des Adidas-Aktienkurses mit dem PrNN (Einer aus allen, 762 Daten - 600 Neuronen)

Damit werden die zuverlässigsten Prognosen für Adidas mit einem PrNN von 200 Neuronen für das Reservoir ermöglicht. Da dieses Ergebnis jedoch von den Zufalls-

zahlen und dem Verlauf des Aktienkurses beeinflusst werden, werden für ein repräsentatives Ergebnis und die Gesamtauswertung, die Vorhersagen für alle weiteren 29 Aktienkurse vorgenommen. Für jeden Aktienkurs wird dabei die Prognose 20 mal durchgeführt und dabei das Modell mit dem geringsten RMSE des Testdatensatzes ausgewählt. Die durchschnittlichen Abweichungen für jeden Aktienkurs sind in Abschnitt C.4.4 aufgeführt.

Entsprechend der in Abschnitt C.4.4 erfassten Abweichungen für jede möglichen Kombination aus der Anzahl an Reservoirneuronen und der Anzahl der Beobachtungen des jeweiligen Aktienkurses ergibt sich zusammengefasst eine Auswertung, die in Tabelle 4.8 gelistet ist. Die erste Zeile weist dabei auf die dabei festgelegte Anzahl der Neuronen hin, wobei die die zweite Zeile aufzeigt, wieviele Beobachtungen dabei eingesetzt werden. Die Anzahl der Aktienkurse, die mit der jeweiligen Kombination aus Daten und Neuronen den im Vergleich geringsten RMSE erzielt haben, ist in der entsprechenden Spalte aufgeführt.

Tabelle 4.8: Auswertung der Vorhersage einer aus allen (PrNN) nach Anzahl der Daten und Neuronen

Datensatz	Anzahl Neuronen (N) zur Anzahl eingelesener Daten (D)								
	200 N			400 N			600 N		
	300D	500D	762D	300D	500D	762D	300D	500D	762D
$E_{Train}$	0/30	0/30	0/30	8/30	0/30	0/30	17/30	5/30	0/30
$E_{Test}$	0/30	3/30	0/30	5/30	0/30	0/30	22/30	0/30	0/30

Wie schon bei dem Aktienkurs von Adidas deutlich wurde, können im Durchschnitt die zuverlässigsten Vorhersagen aus einer Kombination von 300 Beobachtungen, d.h. 240 Trainingsdaten und 60 Testdaten, und einem Reservoir mit 600 Neuronen erzielt werden. Bei der Vorhersage der einzelnen Kurse durch den Einfluss aller anderen 29 Aktienkurse, können für 17 von 30 Aktienkursen im Vergleich zu allen anderen Kombinationen die bestmögliche Approximation der Trainingsdaten ermöglicht werden. Für den Testdatensatz konnten durch dieses Modell 22 von 30 Aktienkurse die zuverlässigsten Prognosen getroffen werden. Dabei ist für 22 Aktienkurse der RMSE der Testdaten im Vergleich am geringsten.

## 4.4 Gesamtauswertung der Ergebnisse

Damit eindeutig festgelegt werden kann, welche Vorhersagemethode für welche Form der Vorhersage die beste Prognose der Aktienkurse ermöglicht, wird in diesem Abschnitt der Ausarbeitung ein Vergleich aller Modelle durchgeführt. Dabei werden zunächst die Ergebnisse der Linearen Regression, des ARIMA/SARIMA-Modells und des PrNN für die Einzelkursvorhersage miteinander verglichen. Hierbei wird das Modell identifiziert, welches die bestmöglichen Vorhersagen ermöglicht. Für die

Gesamtauswertung wird das beste Modell der Einzelkursvorhersage zudem mit den Vorhersagen aller Aktienkurse gleichzeitig und der Prognose der Einzelkurse durch den Einfluss aller weiteren Kurse gegenübergestellt. Das Ziel ist es, auch hieraus die Form der Vorhersage zu wählen, die die zuverlässigsten Prognosen ermöglicht. Neben der Zuverlässigkeit wird in diesem Abschnitt ebenfalls die Anwendung und die Komplexität der Methoden selbst beurteilt.

#### 4.4.1 Die Zuverlässigkeit der Prognosemethoden

Bei dem Vergleich der Ergebnisse zu der Vorhersage eines einzelnen Aktienkurses wird zunächst deutlich, dass für die Lineare Regression, die ARIMA und SARIMA-Modelle und das PrNN eine Vorhersage mit 300 eingelesenen Daten bzw. mit 240 Trainingsdaten und 60 Testdaten die zuverlässigsten Prognosen ermöglicht werden.

Um im nächsten Schritt zu Identifizieren, welche Methode die im Vergleich zuverlässigsten Prognosen erzielt, werden die zuvor ermittelten durchschnittlichen Abweichungen der Trainings- und Testdaten zu dem realen Kursverlauf für die Methoden der Linearen Regression, der Vorhersage aus der Kombination der ARIMA und SARIMA-Modellen und dem Predictive Neural Network mit 200 Neuronen miteinander verglichen. Für jeden Aktienkurs wird dabei entsprechend der Modelle der erfasste RMSE für den Trainings- und den Testdatensatz miteinander verglichen. In diesem Zusammenhang wird dabei die Methode als zuverlässigste Methode, die für den jeweiligen Aktienkurs den geringsten RMSE erzielen konnte. Die Anzahl der Aktienkurse, die dabei am besten mit der jeweiligen Methode geschätzt werden konnte, ist in Tabelle 4.9 zusammengefasst.

Tabelle 4.9: Gesamtauswertung der Einzelkursvorhersage

Datensatz	Vorhersagemethode		
	Lineare Regression	ARIMA/SARIMA	PrNN
$E_{Train}$	0/30	24/30	6/30
$E_{Test}$	1/30	6/30	23/30

Die Spalte der Tabelle 4.9 zeigt dabei die jeweilige Methode und die Zeile den untersuchten Datensatz an. Bei der Auswertung wird dabei deutlich, dass die Trainingsdaten mithilfe der ARIMA und SARIMA-Modelle im Vergleich zu den anderen Methoden am besten abgebildet werden können. Für 24 von 30 Aktienkursen werden mithilfe dieser Modelle für die Trainingsdaten, im Vergleich zu den anderen Methoden, die besten Schätzungen erzielt. Das PrNN ermöglicht für sechs der 30 Aktienkurse im Vergleich die beste Approximation, wohingegen die Lineare Regression die Trainingsdaten am schlechtesten abbildet. Hierbei kann kein Aktienkurs besser geschätzt werden als von den anderen Methoden.

Dennoch ist für das Ergebnis der Ausarbeitung vor allem die Prognosegüte der Modelle ausschlaggebend. Dabei erzielt die Methode der Linearen Regression auch hier die schlechtesten Ergebnisse. Lediglich für einen Aktienkurs kann die im Vergleich beste Prognose ermöglicht werden. Da die Aktienkurse häufig starken Schwankungen und Trendwechsel unterliegen, die einem nicht linearen Trend folgen, können diese durch das Lineare Regressionsverfahren nur ungenügend abgebildet werden. Diese Methode ist lediglich für Aktienkurse mit einem linearen Trendverlauf geeignet. Auch die ARIMA- und SARIMA-Modelle auf Basis der "auto.arima" Funktion in R können den Verlauf der Aktienkurse nur bedingt prognostizieren. Im Gegensatz zu dem Trainingsdatensatz ermöglichen die ARIMA/SARIMA-Modelle nur für sechs Aktienkurse die zuverlässigsten Prognosen. Für die Prognose mit dem ARIMA(p,d,q)-Modell, welches sich aus einem AR(p)-Modell, dem MA(q)-Modell und dem Grad der Differenzbildung d der Zeitreihe zusammensetzt, können die Informationen des MA(q)-Prozesses nicht berücksichtigt werden. Der Grund hierfür ist, dass der MA(q)-Prozess aus den vergangenen Abweichungswerten zwischen den zuvor prognostizierten Werten der Zeitreihe und den tatsächlichen Beobachtungen der Zeitreihe ermittelt wird. Da bei der Prognose der reale Verlauf der Zeitreihe noch nicht vorliegt, ist es nicht möglich, diese Abweichung zu bestimmen. Damit können diese Informationen für die Prognose nicht berücksichtigt werden. Die Vorhersage basiert damit auf der gegebenenfalls differenzierten Zeitreihe, den vergangenen Werten des AR(p)-Modells und gegebenenfalls einem Drift. Entgegen dieser Modelle, können die zuverlässigsten Prognosen für 23 von 30 Aktienkursen mithilfe des Predictive Neural Networks mit einem Reservoir von 200 Neuronen und 300 Daten erzielt werden. Auf Grundlage der Trainingsdaten können die Ausgabegewichte des neuronalen Netzes so an die vorliegenden Daten angepasst werden, dass im Vergleich zu den anderen Methoden gute Prognosen ermittelt werden können.

Zusammengefasst wird als Ergebnis deutlich, dass im Vergleich der drei Prognosemethoden, die Trainingsdaten mit den ARIMA und SARIMA-Modellen bestmöglich bei einer Anzahl von 300 Daten abgebildet werden können. Die zuverlässigsten Vorhersagen werden jedoch mittels des Predictive Neural Network mit einem Reservoir von 200 Neuronen und 300 Daten erzielt.

Ein weiterer Vorteil des Predictive Neural Network, im Gegensatz zu der Linearen Regression und dem Einsatz von ARIMA/SARIMA-Modellen, ist die Möglichkeit, mehrere Aktienkurse gleichzeitig einzulesen und auszugeben. Damit kann bei Investitionsentscheidungen ein Vergleich mehrerer Aktienkurse gleichzeitig erfolgen. Im Rahmen der Evaluierung und Gesamtauswertung gilt es im nächsten Schritt dennoch zu identifizieren, welche Prognoseform die Testdaten bestmöglich approximiert. Das bedeutet, dass hierbei das PrNN bei der Einzelkursvorhersage mit der Prognose des jeweiligen Aktienkurses aus der Prognose aller Aktienkurse gleichzeitig verglichen wird. Hinzugezogen wird dabei für die Einzelkursvorhersage das Modell aus der Kombination aus 200 Neuronen und 300 Daten, da hierbei die zuverlässigsten Vorhersagen getroffen werden konnten. Bei der Vorhersage aller Aktienkurse gleichzeitig,

werden ebenfalls die Ergebnisse aus der Vorhersage mit 300 Daten zum Vergleich einbezogen. Allerdings unterscheidet sich dieses Modell in Bezug auf die Anzahl der Neuronen. Dabei wird das Ergebnis aus der Kombination von 600 Neuronen und 240 Trainings- und 60 Testdaten zur Auswertung eingesetzt. Die zuvor erfassten RMSE in Abschnitt C.4.3 für das multidimensionale PrNN und die in Abschnitt C.4.2 aufgeführten RMSE der Einzelvorhersage werden zur Auswertung miteinander verglichen, um zu identifizieren, welches Modell die zuverlässigsten Vorhersagen ermöglicht. Das Ergebnis ist in Tabelle 4.10 abgebildet.

Tabelle 4.10: Gesamtauswertung des PrNN (Einzel - Multi)

	Einzelkurs (200N-300D)	Multi-PrNN (600N-300D)
$E_{Train}$	0/30	30/30
$E_{Test}$	24/30	6/30

Nach der Tabelle 4.10 wird deutlich, dass die Schätzungen der Trainingsdaten mithilfe des multiplen PrNN für alle Aktienkurse bessere Annäherungen an den realen Kursverlauf aufweisen. Die zuverlässigen Prognosen jedoch, können mithilfe der Einzelkursvorhersage erzielt werden. Der RMSE für die Testdaten ist dabei bei 24 von 30 Aktienkursen mit der Einzelkursvorhersage geringer als bei der Vorhersage mit dem multidimensionalen PrNN.

Damit werden im nächsten Schritt die Ergebnisse der Einzelkursvorhersage ebenfalls mit denen der Vorhersage der einzelnen Kurse durch alle weiteren Kurse verglichen. Für das PrNN der Einzelkursvorhersage bleibt dabei als Auswertungsmodell das Modell mit 300 Daten und 200 Neuronen bestehen. Für den Vergleich werden dabei die Ergebnisse der Einzelkursvorhersage aus allen mittels der Kombination aus 300 Daten und 600 Neuronen berücksichtigt, da damit die für das Modell besten Prognosen erzielt werden konnten. Dabei werden die entsprechenden in Abschnitt C.4.2 und Abschnitt C.4.4 aufgeführten RMSE für jeden Aktienkurs zum Vergleich herangezogen. Das Ergebnis ist in Tabelle 4.11 aufgeführt.

Tabelle 4.11: Gesamtauswertung des PrNN (Einzel - Einer aus allen)

	Einzelkurs (200N-300D)	Einer aus allen (600N-300D)
$E_{Train}$	1/30	29/30
$E_{Test}$	17/30	13/30

Aus der Tabelle 4.11 wird ersichtlich, dass die besseren Schätzungen der Trainingsdaten mit dem multidimensionalen Modell erzielt werden können. Diese können dabei für 29 der 30 Aktienkurse im Vergleich zu der Einzelkursvorhersage besser

approximiert werden. Bei der Prognose werden die Testdaten jedoch mit der Einzelkursvorhersage besser abgebildet. Dabei können mittels dieses Modells für 17 der 30 Aktienkurse zuverlässigere Prognosen erzielt werden.

Insgesamt wird damit deutlich, dass die zuverlässigsten Vorhersagen im Vergleich mit allen anderen untersuchten Methoden mit der Einzelkursvorhersage des Predictive Neural Network ermöglicht werden.

#### 4.4.2 Berücksichtigung der Fundamentaldaten

Im letzten Schritt der Identifikation des bestmöglichen Prognosemodells werden die Fundamentaldaten bei der Prognose berücksichtigt. Damit soll untersucht werden, ob eine Kombination aus der Zeitreihenanalyse und der Aktienanalyse zu einer verbesserten Prognose der Aktienkurse führt. Dies soll mit dem besten Modell aus allen durchgeführt werden. Da das PrNN bei der Einzelkursvorhersage mit 200 Neuronen und 300 Daten, im Vergleich zu allen anderen untersuchten Methoden, im Durchschnitt die zuverlässigsten Prognosen erzielt, wird mit diesem Modell die Vorhersage unter Berücksichtigung der Fundamentaldaten durchgeführt.

Während der Testdurchläufe wurde deutlich, dass das Predictive Neural Network keine Prognosen mit mehr als einer Zeitreihe der Fundamentaldaten ermöglicht. Demnach wird lediglich der Einfluss des Kurs-Gewinn-Verhältnis in diesem Abschnitt untersucht. Für das PrNN mit 300 Daten und 200 Neuronen ergibt sich unter Berücksichtigung des KGV der geschätzte Aktienkurs von Adidas, welcher in Abbildung 4.18 abgebildet ist.

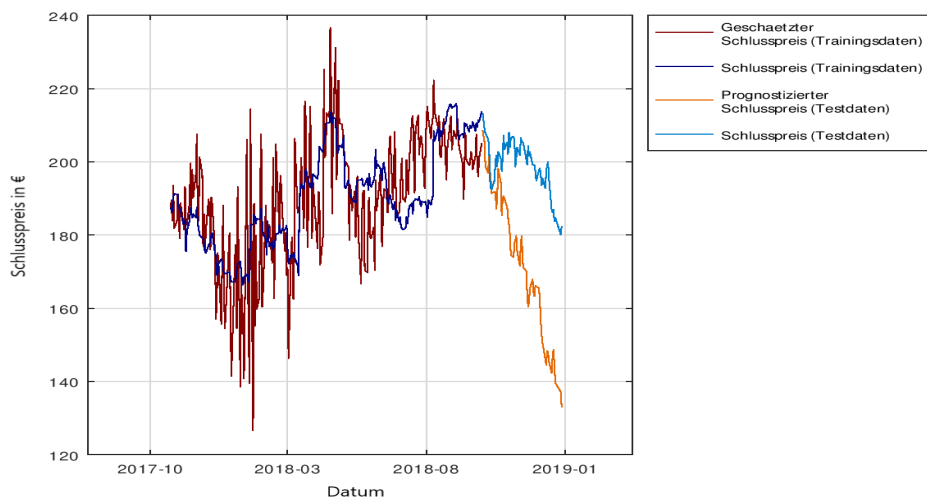


Abbildung 4.18: Vorhersage des Adidas-Aktienkurses mit dem PrNN und dem KGV

Die Prognose ergibt dabei ein RMSE für den Trainingsdatensatz von  $\text{RMSE}_{\text{train}} = 31.09$  und für den Testdatensatz  $\text{RMSE}_{\text{test}} = 149.82$ . Damit kann unter Berücksichti-

gung des KGV keine Optimierung der Vorhersage erzielt werden, da sich für den RMSE der Einzelkursvorhersage die Werte von  $\text{RMSE}_{\text{train}} = 6.54$  und  $\text{RMSE}_{\text{test}} = 8.64$  ergeben. Dennoch wird die Vorhersage für alle weiteren Aktienkurse durchgeführt. Das Ergebnis ist in Abschnitt C.4.5 aufgeführt.

Bei dem Vergleich der RMSE der Einzelkursvorhersage und der Vorhersage mit den Fundamentaldaten wird deutlich, dass unter Berücksichtigung des KGV für vier Aktienkurse eine bessere Approximation der Trainingsdaten ermöglicht wird. Die Vorhersagen der Testdaten fällt jedoch deutlich schlechter aus. Dabei konnten für alle 30 Aktienkurse die zuverlässigsten Prognosen mit der Einzelkursvorhersage ohne Fundamentaldaten erzielt werden.

Damit bleibt das PrNN mit 300 Daten und 200 Neuronen hinsichtlich der Zuverlässigkeit die beste Prognosemethode im Vergleich zu allen untersuchten Methoden.

### 4.4.3 Der Vergleich der Methoden hinsichtlich der Anwendung

Ein weiterer Aspekt, welcher die Vorteile bzw. auch Nachteile der jeweiligen Prognosemethode verdeutlicht, ist die Anwendung der Methoden und deren Komplexität.

Mithilfe der Linearen Regression können in kurzer Zeit Prognosen in Bezug des eingeleseenen Aktienkurses erstellt werden. Der Einsatz der Linearen Regression gestaltet sich dabei als simpel und kann mithilfe der aufgestellten Funktion ausgeführt werden. Die Prognose erfolgt anschließend automatisch durch das Programm. Hierzu werden keine speziellen Vorkenntnisse zur Modellbildung benötigt bzw. müssen hierfür manuell keine Vorkehrungen durch den Anwender zur Prognose der Aktienkurse durchgeführt werden. Zur Prognose muss dabei lediglich die CSV-Datei manuell ausgewählt werden, die die vergangenen Schlusskurse des jeweiligen Aktienkurses enthält. Wie schon zuvor ermittelt, weist diese Methode im Vergleich zu allen weiteren vorgestellten Prognosemethoden jedoch deutliche Schwächen hinsichtlich der Zuverlässigkeit auf.

Mithilfe eines ARIMA bzw. SARIMA-Modells können, im Gegensatz zum Linearen Regressionsmodell, zuverlässigere Prognosen erzielt werden. Der Einsatz von ARIMA bzw. SARIMA-Modellen jedoch, erweist sich als komplexer als alle anderen untersuchten Prognosemethoden. Sollte die Analyse der Zeitreihen und die Prognose dieser manuell erfolgen, sind einige Grundkenntnisse in Bezug auf die Modellierung von Zeitreihen mithilfe von  $\text{ARIMA}(p,d,q)$ -Modellen notwendig. Dabei gestaltet sich, wie im Laufe der Ausarbeitung deutlich wurde, die Bestimmung der Ordnungen von  $p, d$  und  $q$  als auch die Parameterbestimmung von  $\phi$  und  $\theta$  als komplex und erfordert eine detailliertere Analyse der vorliegenden Zeitreihe. Erste Ergebnisse können dabei mittels der Funktion "auto.arima" durch einen automatischen Suchlauf erzielt werden. Durch die Anpassung bestimmter Parameter, zum Beispiel durch die Festlegung einer bestimmten Frequenz oder Saison innerhalb der Zeitreihe, können dabei die Modelle gegebenenfalls optimiert werden. Diese Parameter müssen jedoch



manuell festgelegt werden. Damit diese jedoch bestimmt werden können, muss eine erste Analyse der Zeitreihe durch den Anwender erfolgen und manuell angegeben werden, welche Parameter, wie z.B. die Frequenz, zur Modellbildung berücksichtigt werden soll. Durch eine steigende Anzahl an Parameter steigt jedoch ebenfalls die Komplexität des Modells an und damit auch die Rechenzeit. Schon bei der Verwendung von SARIMA-Modellen für Zeitreihen mit einer saisonalen Komponente, erhöht sich die Rechenzeit um das drei- bis vierfache als bei der Berechnung ohne saisonaler Komponente.

Im Vergleich hierzu ermöglicht das PrNN eine schnelle Verarbeitung der Daten und zuverlässige Prognosen. Der Nachteil des PrNNs jedoch ist es, dass für gute Prognosen teilweise mehr als zehn Durchläufe der Berechnungen benötigt werden. Die Rechenzeit beträgt dabei jedoch deutlich weniger als bei einem Durchlauf eines SARIMA-Modells für eine Zeitreihe mit einer saisonalen Komponente. Die Dauer der Rechenzeit kann dabei mit der doppelten Rechenzeit eines klassischen ARIMA-Modells gleichgesetzt werden.

# 5. Zusammenfassung der Ergebnisse

Dieses Kapitel fasst abschließend die Ergebnisse der Ausarbeitung zusammen. Damit werden die in dieser Arbeit erfassten wichtigsten Erkenntnisse hervorgehoben und vorgestellt.

## 5.1 Zusammenfassung der Arbeit

Mit dem Ziel dieser Ausarbeitung, mögliche bestehende Modelle zur Vorhersage von Aktienkursen vorzustellen und zu vergleichen, wurden die klassischen Prognosemethoden der Linearen Regression, des ARIMA- bzw. des SARIMA-Modells und dem Predictive Neural Network als innovativer Ansatz gewählt. Zwischen diesen Modellen sollte daraus die Methode gewählt werden, die die zuverlässigsten Prognosen ermöglicht und damit bei Investitionsentscheidungen an der Börse als Entscheidungshilfe berücksichtigt werden kann. Mit der Berechnung der durchschnittlichen Abweichungen zwischen dem prognostizierten und realen Kursverlauf mit dem RMSE, konnte dabei die beste Methode ermittelt werden.

Mit den mittels unterschiedlicher Software erstellten Prognosemethoden wurde deutlich, dass für das Ergebnis und die Zuverlässigkeit der Vorhersagen unter anderem die Anzahl der eingelesenen Werte des Aktienkurses ausschlaggebend ist. Aus dem Vergleich zwischen 300, 500 und 762 Werten wurde deutlich, dass in der Regel im Durchschnitt die besten Prognosen mit 300 Beobachtungen erzielt werden können. Mit 240 Trainingsdaten, mithilfe welcher die Modellbildung der einzelnen Methoden erfolgt, und der Vorhersage von 60 Testdaten können hierbei aus 30 Aktienkursen für die meisten Kurse die zuverlässigsten Vorhersagen getroffen werden. Demnach eignen sich diese Modelle vor allem bei kurzfristigen Vorhersagen.

Im Bezug auf das Predictive Neural Network konnte ebenfalls der Einfluss der Reservoirgröße auf die Vorhersagen ermittelt werden. Mit einer Auswahl von einer Reser-

voirgröße mit 200, 400 und 600 Neuronen, wurde deutlich, dass bei der Analyse eines einzelnen Aktienkurses eine geringere Anzahl an Neuronen für die Berechnungen besser geeignet ist. Hierzu konnten für die meisten Aktienkurse die besten Approximationen mit einem PrNN auf Basis von 200 Neuronen und 300 Daten ermittelt werden. Sobald jedoch mehrere Aktienkurse eingelesen werden, wird ein größeres Reservoir für die Vorhersagen benötigt. Hierbei konnten die besten Vorhersagen mit einem PrNN mit 600 Neuronen erzielt werden.

Zudem konnte mithilfe dieser Ausarbeitung identifiziert werden, dass lediglich das Predictive Neural Network die Möglichkeit aufzeigt, mehrere Aktienkurse gleichzeitig einzulesen und vorherzusagen. Dies konnte mit dem ARIMA- und SARIMA-Modell als auch der Linearen Regression nicht eingesetzt werden. Zudem kann nur mit dem PrNN ein Aktienkurs auf Grundlage aller anderen 29 Aktienkurse vorhergesagt werden.

Im Vergleich aller Methoden und Prognoseformen stellt sich nach dieser Ausarbeitung heraus, dass das Predictive Neural Network im Durchschnitt bei der Einzelkursvorhersage die zuverlässigsten Vorhersagen für einen Aktienkurs ermöglicht und damit im Vergleich zu den anderen Modellen am besten bei Investitionsentscheidungen berücksichtigt werden kann. Mithilfe des Modells kann ein zukünftiger Trendanstieg oder -abstieg eines Aktienkurses am besten prognostiziert werden.

## 5.2 Weitere Forschungsziele

Im Rahmen der Aktienanalyse besteht die Möglichkeit, eine Verbesserung der Prognose mittels des PrNN auch unter Berücksichtigung einer Dimensionsreduktion durchzuführen. Dabei wird das neuronale Netz lediglich auf die zur Vorhersage wichtigsten Komponenten reduziert.

Im weiteren kann ein Vergleich weiterer Modelle erfolgen. Da vor allem der Einsatz von künstlichen neuronalen Netzen zunehmend zur Prognose von Aktienkurse eingesetzt wird, ist es sinnvoll, die Zuverlässigkeit dieser mit dem des PrNN zu vergleichen. Mögliche Modelle wären dabei zum Beispiel das erwähnte LSTM.

# A. Anhang

## A.1 Liste der Aktienkurse

Unternehmen		
1. Audi	11. Comdirect	21. Lindt & Sprüngli
2. Adidas	12. Continental	22. Lufthansa
3. Airbus	13. Daimler	23. Microsoft
4. Alphabet	14. Deutsche Telekom	24. Nike
5. BASF	15. Eon	25. RWE
6. BMW	16. Evotec	26. SAP
7. Bayer	17. Exxon Mobil	27. Siemens
8. Beiersdorf	18. FedEx	28. Thyssen Krupp
9. Bertrandt	19. Fielmann	29. Volkswagen
10. CocaCola	20. Infineon Technologies	30. Wirecard



# B. Skripte der Vorhersagemethoden

## B.1 Skript zur Linearen Regression

```
1 library('ggplot2')
2 library('tseries')
3 library('zoo')
4 library('hydroGOF') #for nrmse
5 library('NMOF')
6 library('forecast')
7 library('Metrics')
8
9 #read in data
10 sdata=read.csv('C:/Users/litz_/Desktop/stock_prediction/stock/Continental.csv',
11               header=TRUE, sep=",", stringsAsFactors=FALSE)
12 sdata$Date = as.Date(sdata$Date) #set Date in according format
13   'Date'
14 #seperate data into training- and testdata
15 sdata=sdata[1:nrow(sdata),]
16 sdata = cbind(sdata, "observation"=1:nrow(sdata)) #add timesteps t (t1=1,t2
17   =2,...,tn=n) for regression model
18   training_size = ceiling(0.80 * nrow(sdata)) #set limit for training and
19   testing data
20 #training data set
21 training_data=sdata[1:training_size,] #create new data frame for
22   training data
23 ts_data = ts(training_data[,5]) #declare closing price as
24   time series
25 training_data$timeseries = ts(ts_data) #add time series to training_
26   data
27 #testing data set
28 testing_data=sdata[(training_size+1):nrow(sdata),] #create new data frame and
29   fill with testing data
30 ts_data = ts(testing_data[,5]) #set closing price as time
31   series
32 testing_data$timeseries = ts(ts_data) #add time series to training_
33   data
```

```

29
30 #developpe regression model
31 regression_model=lm(timeseries~observation ,data=training_data)      #compute
32   regression from training_data
33 print(regression_model)                                             #show
34   intercept and slope of regression line
35
36 training_data$Regression=as.ts(predict(regression_model, training_data)) #
37   estimate y for each time step
38
39 #prediction for testing_values
40 prediction=predict(regression_model, testing_data)                  #
41   predict testing data
42 testing_data$Prediction=predict(regression_model, testing_data)
43
44 #plot testing and training and prediction
45 ggplot()+
46   labs(title="", x="Zeit in t", y="Schlusspreis in Euro", col="") +
47   theme(plot.title=element_text(hjust=0.5), legend.position="none") +
48   geom_line(data=training_data, aes(x=Date, y = Close, col = "Schlusspreis \n(
49     Trainingsdaten)")) +
50   geom_line(data=training_data, aes(x=Date, y = Regression, col = "Geschaetzter
51     \nSchlusspreis \n(Trainingsdaten)\n"))+
52   geom_line(data=testing_data, aes(x=Date, y = Close, col = "Schlusspreis \n(
53     Testdaten)\n")) +
54   geom_line(data=testing_data, aes(x=Date, y = Prediction, col = "
55     Prognostizierter \nSchlusspreis \n(Testdaten)\n")) +
56   scale_x_date(date_labels = "%b %y", date_breaks = "6 months")+
57   scale_color_manual(values = c("#8B0000", "#FA8072", "#6495ED", "#00008B"))
58   #annotate("text", x=as.Date("2016-05-03"), y=840, label=expression(paste(hat(y))*
59     = 594.1132 + 0.2775*t))
60   #annotate("text", x=as.Date("2016-06-03"), y=Inf, label=expression(paste(hat(y))*
61     = 174.4546 + 0.1312*t"), vjust=1, hjust=1)
62
63 #calculate RMSE for training and testing data
64 RMSE_train=rmse(training_data$timeseries , training_data$fit)
65 RMSE_train
66
67 RMSE_test=rmse(testing_data$timeseries , testing_data$Prediction)
68 RMSE_test

```

Quelltext B.1: Skript zum Linearen Regressionsverfahren

## B.2 Skript zum ARIMA-Modell

```

1 rm(list=ls(all=TRUE))
2 library('ggplot2')
3 library('forecast')
4 library('tseries')
5 library('zoo')
6 library('hydroGOF') #for nrmse
7 library('TSA')
8
9 #read in data (insert stock name)
10 sdata=read.csv('C:/Users/litz_/Desktop/stock_prediction/stock/Adidas.csv', header=
11   TRUE, sep=";", stringsAsFactors=FALSE)
12 sdata$Date = as.Date(sdata$Date) #set Date in according format '
13   Date'
14
15 sdata=sdata[1:nrow(sdata),]
16
17 #seperate data into training- and testdata

```

```

16 training_size = ceiling(0.80 * nrow(sdata))           #set limit for training
    data
17
18 #training data
19 training_data=sdata[1:training_size,]                #create new data frame and
    fill with training data
20 ts_data = ts(training_data[,5])                     #declare of closing price as
    a time series
21 training_data$timeseries = ts(ts_data)              #add time series to training_
    data
22
23 #testing data
24 testing_data=sdata[(training_size+1):nrow(sdata),] #create new data frame and
    fill with testing data
25 ts_data = ts(testing_data[,5])                     #set closing price as time
    series
26 testing_data$timeseries = ts(ts_data)              #add time series to training_
    data
27
28
29 #build ARIMA/SARIMA model
30
31 arima_model=auto.arima(training_data$timeseries, trace=TRUE, approximation=FALSE)
    #develop arima_model according to timeseries
32
33 tdisplay(residuals(arima_model),lag.max=15)          #
    check residuals
34 training_data$fit=arima_model$fitted               #
    save estimated training data
35
36 #predict testing data
37 predicted=forecast(arima_model,h=nrow(testing_data)) #
    predict testing data
38 predicted_tmp=as.numeric(predicted$mean)
39 testing_data$Prediction=predicted_tmp
40
41 ggplot()+
42   labs(title=(expression(atop("ARIMA (0,1,1)",paste(theta[1], '-0.2154 '))), x="
    Datum", y="Schlusspreis (in ???)", col="") +
43   theme(plot.title=element_text(hjust=0.5)) +
44   geom_line(data=training_data, aes(x=Date, y = Close, col = "Schlusspreis \n(
    Trainingsdaten)")) +
45   geom_line(data=training_data, aes(x=Date, y = fit, col = "Geschaetzter \
    nSchlusspreis \n(Trainingsdaten)\n"))+
46   geom_line(data=testing_data, aes(x=Date, y = Close, col = "Schlusspreis \n(
    Testdaten)\n")) +
47   geom_line(data=testing_data, aes(x=Date, y = Prediction, col = "
    Prognostizierter \nSchlusspreis \n(Testdaten)\n")) +
48   scale_x_date(date_labels = "%b %y", date_breaks = "3 months")+
49   scale_color_manual(values = c("#8B0000", "#FA8072", "#6495ED", "#00008B"))
50
51
52 #calculate RMSE for training and testing data
53 RMSE_train=rmse(training_data$timeseries, training_data$fit)
54 RMSE_train
55
56 RMSE_test=rmse(testing_data$timeseries, testing_data$Prediction)
57 RMSE_test

```

Quelltext B.2: Skript zum ARIMA-Modell

## B.3 Skript zum SARIMA-Modell



```

1  rm(list=ls(all=TRUE))
2  library('ggplot2')
3  library('forecast')
4  library('tseries')
5  library('zoo')
6  library('hydroGOF') #for nrmse
7
8  #read in data
9  sdata=read.csv('C:/Users/litz_/Desktop/stock_prediction/stock/RWE.csv',header=
10 TRUE, sep=";", stringsAsFactors=FALSE)
11 sdata$Date = as.Date(sdata$Date) #set Date in according format '
12 Date'
13 sdata=sdata[1:nrow(sdata),]
14
15 #seperate data into training- and testdata
16 training_size = ceiling(0.80 * nrow(sdata)) #set limit for training
17 data
18 #training data set
19 training_data=sdata[1:training_size,] #create new data frame and
20 fill with training data
21 ts_data = ts(training_data[,5]) #declare of closing price as
22 a time series
23 training_data$timeseries = ts(ts_data) #add time series to training-
24 data
25
26 #testing data set
27 testing_data=sdata[(training_size+1):nrow(sdata),] #create new data frame and
28 fill with testing data
29 ts_data = ts(testing_data[,5]) #set closing price as time
30 series
31 testing_data$timeseries = ts(ts_data) #add time series to training-
32 data
33
34 #Seasonal ARIMA model
35 training_data$timeseries=ts(training_data$timeseries, frequency=254)
36 sarima_model=auto.arima(ts(training_data$timeseries, frequency=254), trace=TRUE,
37 seasonal=TRUE) #develop sarima_model according to timeseries
38
39 tsdisplay(residuals(sarima_model), lag.max=15) #analyse
40 residuals
41 training_data$fit=sarima_model$fitted
42 training_data$fit=ts(training_data$fit, frequency=254)
43 #predict testing data
44 predicted=forecast(sarima_model, h=nrow(testing_data))
45 predicted_tmp=as.numeric(predicted$mean)
46 testing_data$Prediction=predicted_tmp
47
48 #plot stock
49 ggplot()+
50 labs(title=(expression(atop("ARIMA (2,1,2)", paste(phi[1], '-0.5822 ', phi[2], '
51 =-0.6042 ', theta[1], '=0.6008 ', theta[2], '=0.7502 '))))), x="Zeit in t", y="
52 Schlusspreis in Euro", col="") +
53 theme(plot.title=element_text(hjust=0.5)) +
54 geom_line(data=training_data, aes(x=Date, y = Close, col = "Schlusspreis \n(
55 Trainingsdaten)")) +
56 geom_line(data=training_data, aes(x=Date, y = fit, col = "Geschtzter \
57 nSchlusspreis \n(Trainingsdaten)\n"))+
58 geom_line(data=testing_data, aes(x=Date, y = Close, col = "Schlusspreis \n(
59 Testdaten)\n")) +
60 geom_line(data=testing_data, aes(x=Date, y = Prediction, col = "
61 Prognostizierter \nSchlusspreis \n(Testdaten)\n")) +
62 scale_x_date(date_labels = "%b %y", date_breaks = "3 months")+

```

```

48     scale_color_manual(values = c("#8B0000", "#FA8072", "#6495ED", "#00008B"))
49
50
51 #calculate RMSE for training and testing data
52 RMSE_train=rmse(training_data$timeseries , training_data$fit)
53 RMSE_train
54
55 RMSE_test=rmse(testing_data$timeseries , testing_data$Prediction)
56 RMSE_test

```

Quelltext B.3: Skript zum SARIMA-Modell

## B.4 Skript zum PrNN

```

1  addpath("library")
2
3  global N=200;
4  global limit=300;      #(limit=700 - take only last 700 values from data of 3
5  years)
6  select=2;             #0=take only one stock, 1=take every listed stock within
7  file
8  global makeplot= true;
9
10 %files
11 #change filepath according to selected prediction form
12 global pathstock = "stock/";          #original stock
13 pathstock = glob('stock/*.csv');
14
15 #global pathstock = "stockfund/";     #CSV files with fundamental values
16 #pathstock = glob('stockfund/*.csv');

```

Quelltext B.4: prep\_prediction.m

```

1  #run compute_prediction to calculate predictions
2
3  prep_prediction;
4
5  #read in stock data
6  [training_data, testing_data, data, stock_name]=read_in_data();
7  S=training_data(2:end, :);
8  T=testing_data(2:end, :);
9
10 #determine training and testing size
11 size_training=size(training_data);
12 size_training=size_training(:, 2);
13 size_testing=size(testing_data);
14
15
16 #predict m more steps
17 M=size_testing(:, 2);      #m corresponds to size of testing_data for evaluation
18
19
20
21 #develop model and predict testing_data
22 switch(select)
23     case 0 #single stock prediction
24         [Out, A, J, Y, W, X, RMSE_training_f, RMSE_testing_f]=single_stock(S, T, size_
25             training, M, testing_data, training_data);

```

```

26     case 1    #multiple stock prediction
27         [Out,A,J,Y,W,X,RMSE_training_f,RMSE_training_avg,RMSE_testing_f,RMSE_
           testing_avg,count]=multiple_stock(S,T,size_training,M,testing_data,
           training_data);
28
29     case 2    #predict one stock from others
30         [Out,Err,A,J,Y,W,X,RMSE_training_f,RMSE_testing_f,count,Out_f]=single_from_
           multiple(S,T,size_training,0,testing_data,training_data);
31     endswitch

```

### Quelltext B.5: compute\_predictions.m

```

1  function [training_data,testing_data,data,stock_name]=read_in_data()
2  prep_prediction;
3
4  #read in data
5  switch (select)
6  case 0    #read in single stock
7      stockdata=fopen(pathstock{1});
8      Dat= textscan(stockdata,'%s%f%f%f%f%f%f%d','Delimiter',' ',' ','headerlines
           ',1);
9      price=transpose(Dat{5});
10     data(2,:)=price;
11     name = dir(pathstock{1});
12     stock_name=name.name
13     fclose(pathstock{1});
14
15
16 case 1    #read in every stock within folder
17     stock_name=0;
18     numberofstock = numel(pathstock);
19     for i=1:numberofstock
20         stockdata=fopen([pathstock{i}]);
21         Dat= textscan(stockdata,'%s%f%f%f%f%f%f%d','Delimiter',' ',' ','
           headerlines',1);
22         price=transpose(Dat{5});
23         data(i+1,:)=price;
24         name = dir([pathstock{i}]);
25         stock_name=name.name
26         fclose([pathstock{i}]);
27     endfor
28
29
30 case 2    #read in every stock in folder
31     stock_name=0;
32     numberofstock = numel(pathstock);
33     for i=1:numberofstock
34         stockdata=fopen([pathstock{i}]);
35         Dat= textscan(stockdata,'%s%f%f%f%f%f%f%d','Delimiter',' ',' ','
           headerlines',1);
36         price=transpose(Dat{5});
37         data(i+1,:)=price;
38         name = dir([pathstock{i}]);
39         stock_name=name.name
40         fclose([pathstock{i}]);
41     endfor
42     data([31,2],:)=data([2,31],:);    #set position of needed stock to last
           position
43 endswitch
44
45 #set date
46 data(1,:)=datenum(Dat{:},1);
47 lim=(numel(data(2,:))+1)-limit
48 data=data(:,lim:end);

```

```

49
50 #splitting data into training and testing
51 limit_train=round(numel(data(2,:))*0.8);
52 training_data=data(:,1:limit_train);
53 testing_data=data(:,limit_train+1:end);
54
55 endfunction

```

Quelltext B.6: read\_in\_data.m

```

1 function [Out,A,J,Y,W,X,RMSE_training_f,RMSE_testing_f,predicted_f]=single_stock(S,T
   ,size_training,M,testing_data,training_data)
2 prep_prediction;
3
4 threshold=1000 #set threshold to find best model
5 count=0
6
7 for i=1:20
8     [Out,Err,A,J,Y,W,X]=predict(S,M,N,theta=0,epsilon=Inf,delta=0,Range=0); #
   build model and predict m more steps
9     predicted=Out(:,size_training+1:end);
10    RMSE_testing=normse(T,predicted,1,0);
11    if (RMSE_testing<threshold) #select best model
12        count=count+1
13        threshold=RMSE_testing;
14        Out_f=Out;
15        A_f=A;
16        J_f=J;
17        Y_f=Y;
18        W_f=W;
19        X_f=X;
20
21        #evaluate PrNN model
22        predicted_f=Out_f(:,size_training+1:end);
23        model= Out_f(:,1:size_training);
24
25        RMSE_training_f=Err;
26        RMSE_testing_f=normse(T,predicted_f,1,0);
27    endif
28 endfor
29
30
31 %make plots
32 dates_test=testing_data(1,:);
33 dates_train=training_data(1,:);
34
35
36 if (makeplot)
37     plot(dates_train,model,";Geschaetzter \nSchlusspreis \n(Trainingsdaten);",
   'linestyle','-', 'color',[0.53,0,0],
38         dates_train,S(:,:),";Schlusspreis \n(Trainingsdaten);", 'linestyle',
   '-', 'color',[0,0,0.53],
39         dates_test,predicted_f,";Prognostizierter \nSchlusspreis \n(Testdaten
   );", 'linestyle','-', 'color',[0.9,0.4,0],
40         dates_test,T(:,:),";Schlusspreis \nTestdaten);", 'linestyle','-', '
   color',[0,0.5,0.8]);
41     lgd=legend("location","northeastoutside");
42     set(lgd,'fontsize',9);
43     datetick("x","yyyy-mm","keeplimits")
44 endif
45 endfunction

```

Quelltext B.7: single\_stock.m

```

1 function [Out,Err,A,J,Y,W,X] = predict(S,M=0,N=-1,theta=0,epsilon=Inf,delta=0,Range
   =0)
2 % predict next values of a given input sequence
3 % S : input sequence (time series)
4 % M : prediction for M more steps
5 % N : reservoir size (number of neurons) (-1 = same number of equations and
   unknowns)
6 % theta : threshold for NRMSE (0 = no dimensionality reduction)
7 % epsilon : desired maximal deviation of eigenvalues from absolute value 1 (Inf =
   no cleaning)
8 % delta : minimal distance between eigenvalues (0 = no clustering)
9 % Range : indices of relevant input/output components (0 = take all components)
10
11 % Out : output sequence (predicted)
12 % Err : NRMSE ouf predicted output
13 % A : output matrix (for reduced number of dimensions)
14 % J : Jordan matrix (for dynamics of reduced version)
15 % Y : network dynamics state sequence
16 % W : overall original learned transition matrix [Wout; Win Wres]
17 % Wout : learned output weights [Wout]
18 % Wir : input and reservoir weight matrix [Win Wres]
19 % X : original input and reservoir state sequence [S; R]
20
21
22 % INITIALISATION
23 n = columns(S)-1; % sequence length
24 d = rows(S); % number of inputs, often = 1
25 if (N<0)
26     N = n-d;
27 endif
28
29
30 % LEARN OUTPUT WEIGHTS
31
32 % network initialization (randomly)
33 W = zeros(d+N);
34 Index = d+(1:N);
35 W(Index,1:d) = randn(N,d)/sqrt(d); % balanced input weights [Win]
36 X = zeros(d+N,n+1); % input and reservoir state sequence
37 W(Index,Index) = reservoir(N); % Wres
38 X(Index,1) = start(N);
39
40 % drive given input through reservoir (input receiving mode)
41
42
43 X(:,1:n+1) = compute(W,X(:,1),0,S);
44 % learn output weights
45 Yout = S(:,2:n+1); % predicted sequence
46 warning('off','Octave:singular-matrix');
47 W(1:d,:) = Yout/X(:,1:n); % output weights
48 warning('on','Octave:singular-matrix');
49
50
51 % REAL JORDAN DECOMPOSITION
52
53 % preparation
54 if (Range==0)
55     Range = 1:d;
56 endif
57 [Centr,Multi] = cluster(eig(W),delta); % clustering eigenvalues
58 [Lambda,Multi] = clean(Centr,Multi,epsilon); % omit small eigenvalues
59
60 % construct Jordan matrix and determine mapping matrix
61 [J,N,K] = jormat(Lambda,Multi);

```

```

62 Y = compute(J, start(N), n); % output generating mode
63 A = (X/Y)(Range, :);
64
65
66 % DIMENSION REDUCTION
67
68 % initialisation
69 In = X(Range, :);
70 Out = A*Y; % predicted sequence
71 Err = nrmse(In, Out); % original error
72
73 if (Err < theta)
74
75     % compute error for each component and sort them in ascending order
76     Err1 = zeros(K, 1);
77     pos = 1; % position in Jordan matrix
78     for (k=1:K)
79         m = Multi(k);
80         Index = [1:pos-1 pos+m:N];
81         J1 = J(Index, Index);
82         Y1 = compute(J1, start(N-m), n);
83         A1 = (X/Y1)(Range, :);
84         Out1 = A1*Y1;
85         Err1(k) = nrmse(In, Out1);
86         pos += m;
87     endfor
88
89     % re-order Jordan components
90     [Err1, Ord] = sort(Err1);
91     Lambda = Lambda(Ord);
92     Multi = Multi(Ord);
93     [J, N, K] = jormat(Lambda, Multi);
94
95     % omit components as long as error remains small enough
96     for (k=1:K)
97         m = Multi(k);
98         J2 = J(m+1:end, m+1:end);
99         N -= m;
100        Y2 = compute(J2, start(N), n);
101        A2 = (X/Y2)(Range, :);
102        Out2 = A2*Y2;
103        Err2 = nrmse(In, Out2);
104        if (Err2 < theta)
105            Out = Out2;
106            Err = Err2;
107            A = A2;
108            J = J2;
109            Y = Y2;
110        else
111            break;
112        endif
113    endfor
114
115 endif
116
117 % PREDICTION
118
119 if (M > 0)
120     Out = [Out(:, 1:end-1) A*compute(J, Y(:, end), M)];
121 endif
122
123 endfunction

```

Quelltext B.8: predict.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg)

```

1 function W = reservoir(N)
2 % create reservoir of size N with weight matrix W and unit spectral radius,
3 % corresponding start vector r with unit norm
4
5 % reservoir
6 if (N>1)
7     while (true)
8         try
9             W = randn(N); % Gaussian noise square matrix
10            W /= abs(eigs(W,1,'lm')); % set spectral radius
11            break;
12        catch
13            ;
14        end_try_catch
15    endwhile
16 else
17     W = ones(N);
18 endif
19
20 endfunction

```

Quelltext B.9: reservoir.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg)

```

1 function X = compute(W,x,m=10,S=x)
2 % compute  $X = W^t * x$  for  $t = 0, 1, \dots$ 
3 % input receiving mode for given input S
4 % output generating mode for m more steps
5
6 % W : recursive function definition (square matrix)
7 % x : initial column vector (start vector)
8
9 d = rows(S);
10 n = columns(S)-1;
11 N = rows(x);
12 X = zeros(N,n+1+m);
13 X(:,1) = x;
14
15 % input receiving mode
16 X(1:d,1:n+1)=S;
17 if (n>0)
18     I = (d+1):N;
19     M = W(I,:);
20     for (k=2:n+1)
21         X(I,k) = M*X(:,k-1);
22     endfor
23 endif
24
25 % output generating mode
26 if (m>0)
27     x = X(:,n+1);
28     for (k=n+1+(1:m))
29         x = W*x;
30         X(:,k) = x;
31     endfor
32 endif
33
34 endfunction

```

Quelltext B.10: compute.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg)

```

1 function result = nrmse(In,Out,Mode=1,Flag=0)
2 % normalised root mean square error

```

```

3
4 % In : given sequence
5 % Out : computed sequence
6 % Mode : kind of normalisation
7 % Flag : 0 = one averaged value or 1 = per row
8
9 result = (Out-In).^2;
10 switch (Mode)
11     case 0 % without normalisation
12         result = sum(result,2);
13     case 1 % wrt. number of elements
14         result = mean(result,2);
15     case 2 % divided by variance
16         result = mean(result,2) ./var(In,0,2);
17 endswitch
18
19 if (Flag)
20     result = mean(result);
21 endif
22
23 result = sqrt(result);
24 endfunction

```

Quelltext B.11: nrmse.m (Zur Verfügung gestellt von Prof. Dr. Stolzenburg)

```

1 function [Out,A,J,Y,W,X,RMSE_training_f,RMSE_training_avg,RMSE_testing_f,RMSE_
   testing_avg,count]=multiple_stock(S,T,size_training,M,testing_data,training_
   data)
2 prep_prediction;
3
4 threshold=50000;
5 count=0
6 for i=1:20
7     [Out,Err,A,J,Y,W,X]=predict(S,M,N,theta=0,epsilon=Inf,delta=0,Range=0);
8     RMSE_training=Err;
9     predicted=Out(:,size_training+1:end);
10    RMSE_testing=nrmse(T,predicted,1,1);
11    if(RMSE_testing<threshold)
12        count=count+1
13        threshold=RMSE_testing;
14        Out_f=Out;
15        A_f=A;
16        J_f=J;
17        Y_f=Y;
18        W_f=W;
19        X_f=X;
20
21        #evaluate PrNN model
22        predicted_f=Out_f(:,size_training+1:end);
23        model= Out_f(:,1:size_training);
24        RMSE_training_f=Err;
25        RMSE_training_avg=nrmse(S,model,1,1);
26
27        RMSE_testing_f=nrmse(T,predicted_f,1,0);
28        RMSE_testing_avg=nrmse(T,predicted_f,1,1);
29    endif
30 endfor
31
32 RMSE_training_f
33 RMSE_testing_f
34
35 #make plots
36 dates_test=testing_data(1,:);
37 dates_train=training_data(1,:);

```



```

38
39 #plot multiple stocks (training, testing and predicted values)
40 if(makeplot)
41     for i=1:3
42         figure(i)
43         plot(dates_train,model(i,:), ";Geschaetzter \nSchlusspreis (Trainingsdaten)
44             ";", 'linestyle', '-', 'color', [0.53,0,0],
45             dates_train,S(i,:), ";Schlusspreis (Trainingsdaten);", 'linestyle', '-
46             ', 'color', [0,0,0.53],
47             dates_test,predicted_f(i,:), ";Prognostizierter \nSchlusspreis (
48             Testdaten);", 'linestyle', '-', 'color', [0.9,0.4,0],
49             dates_test,T(i,:), ";Schlusspreis (Testdaten);", 'linestyle', '-', '
50             color', [0,0.5,0.8]);
51         legend("location", "northeastoutside")
52         datetick("x", "yyyy-mm", "keeplimits")
53     endfor
54 endif
55 endfunction

```

Quelltext B.12: multiple\_stock.m

```

1 function [Out, Err, A, J, Y, W, X, RMSE_training_f, RMSE_testing_f, count, Out_f]=single_from_
2     multiple(S,T, size_training, M, testing_data, training_data)
3     prep_prediction;
4     threshold=1500;      #set threshold to choose best model within loop
5     count=0
6     for i=1:20
7         [Out, Err, A, J, Y, W, X]=predict(S,0,N, theta=0, epsilon=Inf, delta=0, Range=0); #
8             build model
9
10        total_data=[S T];      #combine training and testing data
11        d=rows(T);
12        n=columns(T);
13        pred=zeros(d+N,n+1);      #develop matrix for predicted values
14
15        pred(:,1:n+1) = compute(W,X(:, size_training),0,total_data(1:29, size_training
16            :end)); #predict testing data
17        pred=pred(:,2:n+1);
18        RMSE_testing=normse(T(30,:), pred(30,:));
19
20        if(RMSE_testing<threshold)      #choose best model
21            count=count+1;
22            threshold=RMSE_testing;
23            pred_f=pred;
24            Out_f=[Out(30,:) pred_f(30,:)];
25            A_f=A;
26            J_f=J;
27            Y_f=Y;
28            W_f=W;
29            X_f=X;
30
31            #evaluate PrNN model
32            predicted_f=Out_f(:, size_training+1:end);
33            model= Out_f(:,1: size_training);
34
35            RMSE_training_f=Err(30,:);
36            RMSE_testing_f=RMSE_testing;
37        endif
38    endfor
39
40    RMSE_training_f

```

```

39 RMSE_testing_f
40
41 #plot data
42 dates_test=testing_data(1,:);
43 dates_train=training_data(1,:);
44
45 if(makeplot)
46     plot(dates_train,model,"Geschae tzter \nSchlusspreis \n(Trainingsdaten);", '
         linestyle', '-','color',[0.53,0,0],
47         dates_train,S(30,:), "Schlusspreis \n(Trainingsdaten);", 'linestyle', '-
         ', 'color',[0,0,0.53],
48         dates_test,predicted_f,"Prognostizierter \nSchlusspreis \n(Testdaten);"
         , 'linestyle', '-','color',[0.9,0.4,0],
49         dates_test,T(30,:), "Schlusspreis \nTestdaten);", 'linestyle', '-','
         color',[0,0.5,0.8]);
50     lgd=legend("location","northeastoutside");
51     set(lgd,'fontsize',9);
52     datetick("x","yyyy-mm","keeplimits")
53 endif
54 endfunction

```

Quelltext B.13: single\_from\_multiple.m

```

1 prep_prediction;
2
3 #read in data
4 stockdata=fopen(pathstock{28});
5     Dat= textscan(stockdata,'%s %s %f %f %f %f %d %f %f %f %f %f %f', '
         Delimiter',' ','headerlines',1);
6     name = dir(pathstock{28});
7     stock_name=name.name
8
9     price=transpose(Dat{6});
10    KGV=transpose(Dat{12});           #select fundamental value
11    %KCV=transpose(Dat{13});
12    %DIV=transpose(Dat{14});
13    data(2,1:762) = price;
14    data(3,1:762) = KGV;
15    %data(4,1:762)= KCV;
16    %data(5,1:762)= DIV;
17    fclose(pathstock{28});
18
19    data(1,1:762)=datenum(Dat{: ,2});   #if fundamental values need to be read in
20    lim=(numel(data(2,:))+1)-limit
21    data=data(:,lim:end);
22
23 #split data into training and testing
24    limit_train=round(numel(data(2,:))*0.8);
25    training_data=data(:,1:limit_train);
26    testing_data=data(:,limit_train+1:end);
27
28    S=training_data(2:end,:);
29    T=testing_data(2:end,:);
30
31 #determine training and testing size
32    size_training=size(training_data);
33    size_training=size_training(:,2);
34    size_testing=size(testing_data);
35
36 #set M for prediction for m more steps
37    M=size_testing(:,2);   #m corresponds to size of testing_data for evaluation
38
39
40

```

```

41
42 #predict stock price
43
44 threshold=50000000;      #set threshold to determine best model
45 count=0                  #count improved models
46
47 for i=1:20
48     [Out, Err, A, J, Y, W, X]=predict(S,M,N, theta=0, epsilon=Inf, delta=0, Range=1);
49     RMSE_training=Err;
50     predicted=Out(:, size_training+1:end);
51     RMSE_testing=normse(T, predicted, 1, 1);
52     if(RMSE_testing<threshold)
53         count=count+1
54         threshold=RMSE_testing;
55         Out_f=Out;
56         A_f=A;
57         J_f=J;
58         Y_f=Y;
59         W_f=W;
60         X_f=X;
61
62         #evaluate PrNN model
63         predicted_f=Out_f(:, size_training+1:end);
64         model= Out_f(:, 1:size_training);
65         RMSE_training_f=Err;          #RMSE for training data
66
67         RMSE_testing_f=normse(T(1,:), predicted_f, 1, 0); #RMSE for testing data
68     endif
69 endfor
70
71
72 RMSE_training_f
73 RMSE_testing_f
74
75
76 #make plots
77 dates_test=testing_data(1,:);
78 dates_train=training_data(1,:);
79
80 if(makeplot)
81     plot(dates_train, model(1,:), "Geschaetzter \nSchlusspreis (Trainingsdaten)
82         "; 'linestyle', '-', 'color', [0.53,0,0],
83         dates_train, S(1,:), "Schlusspreis (Trainingsdaten)"; 'linestyle', '-
84         ', 'color', [0,0,0.53],
85         dates_test, predicted_f(1,:), "Prognostizierter \nSchlusspreis (
86         Testdaten)"; 'linestyle', '-', 'color', [0.9,0.4,0],
87         dates_test, T(1,:), "Schlusspreis (Testdaten)"; 'linestyle', '-', '
88         color', [0,0.5,0.8]);
89     legend("location", "northeastoutside")
90     datetick("x", "yyyy-mm", "keeplimits")
91 endif

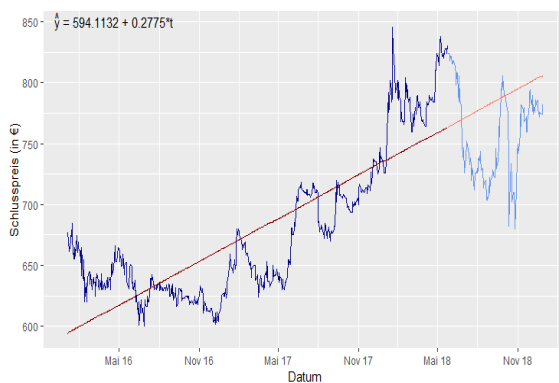
```

Quelltext B.14: fundamental\_stock.m

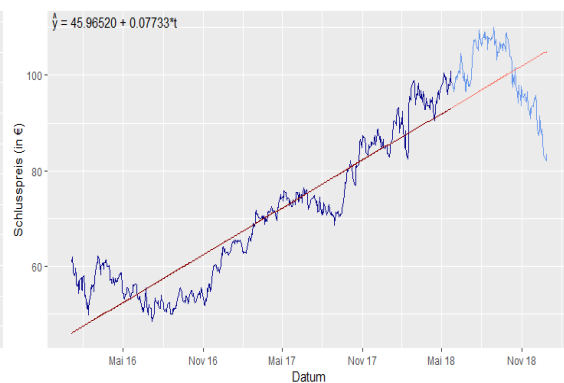
# C. Auswertung der Vorhersagen

## C.1 Die Vorhersage mit der Linearen Regression

### C.1.1 Die Abbildungen zur Vorhersage mit der Linearen Regression

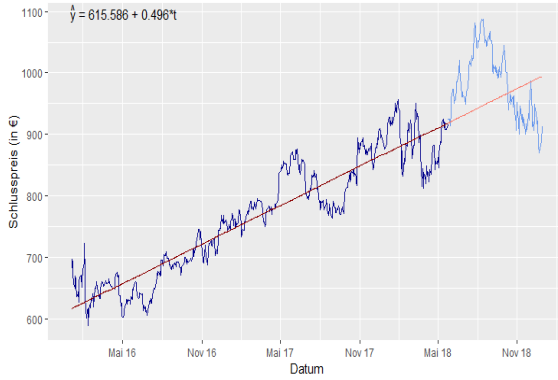


(1) Vorhersage des Aktienkurses von Audi

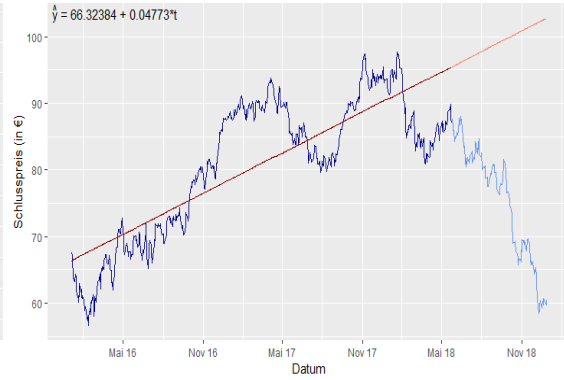


(2) Vorhersage des Aktienkurses von Airbus

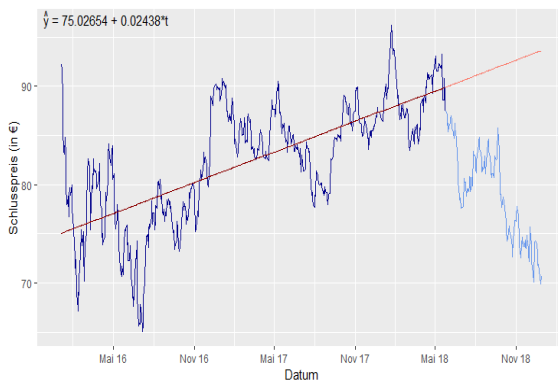
— Geschätzter Schlusspreis (Trainingsdaten) — Prognostizierter Schlusspreis (Testdaten) — Schlusspreis (Testdaten) — Schlusspreis (Trainingsdaten)



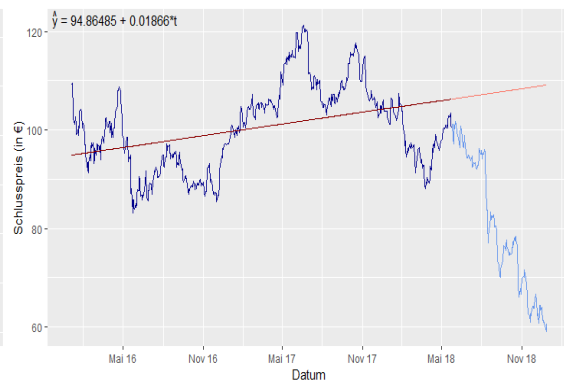
(3) Vorhersage des Aktienkurses von Alphabet



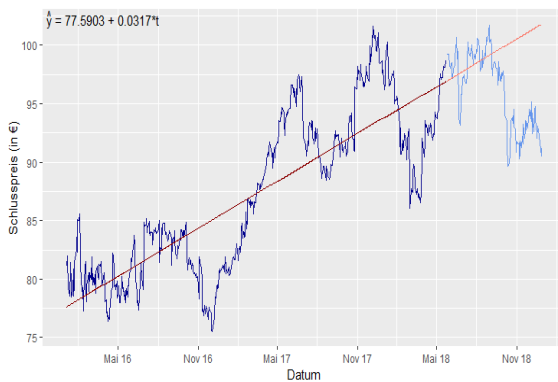
(4) Vorhersage des Aktienkurses von BASF



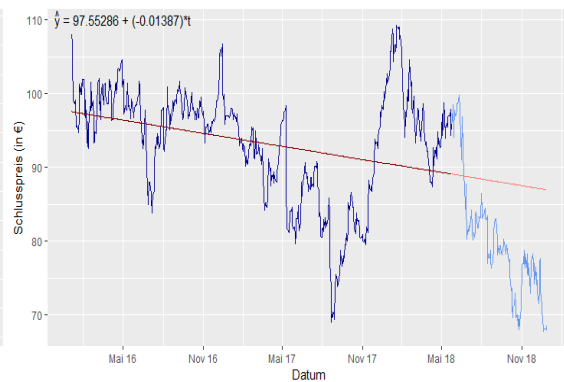
(5) Vorhersage des Aktienkurses von BMW



(6) Vorhersage des Aktienkurses von Bayer

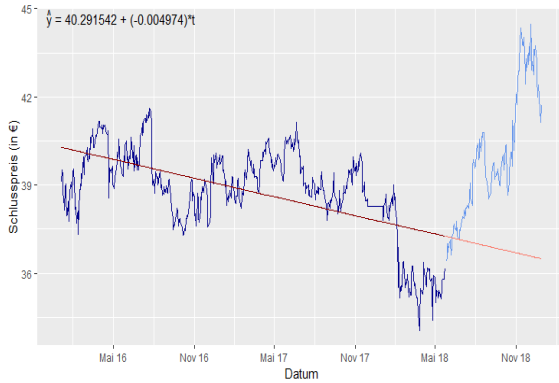


(7) Vorhersage des Aktienkurses von Beiersdorf

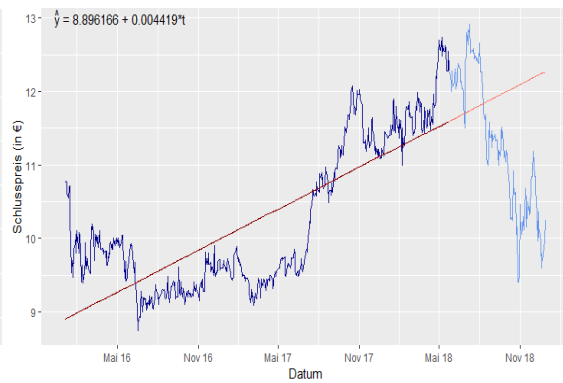


(8) Vorhersage des Aktienkurses von Bertrandt

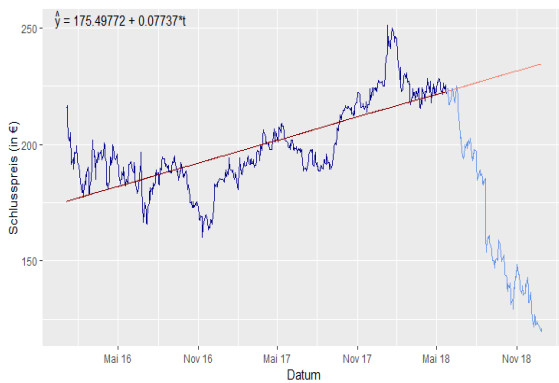
— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)



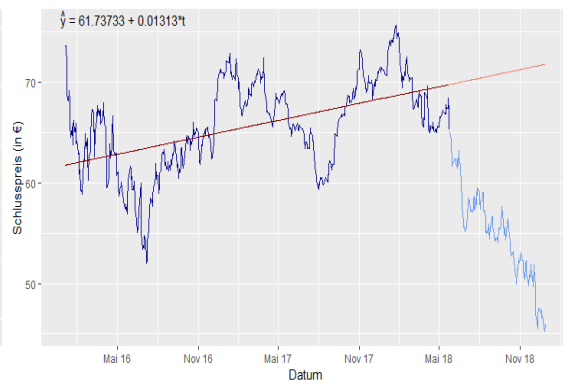
(9) Vorhersage des Aktienkurses von Coca Cola



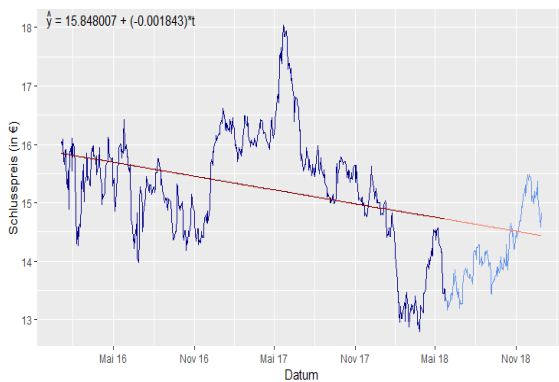
(10) Vorhersage des Aktienkurses von Comdirect



(11) Vorhersage des Aktienkurses von Continental



(11) Vorhersage des Aktienkurses von Daimler



(12) Vorhersage des Aktienkurses von Deutsche Telekom



(13) Vorhersage des Aktienkurses von Eon

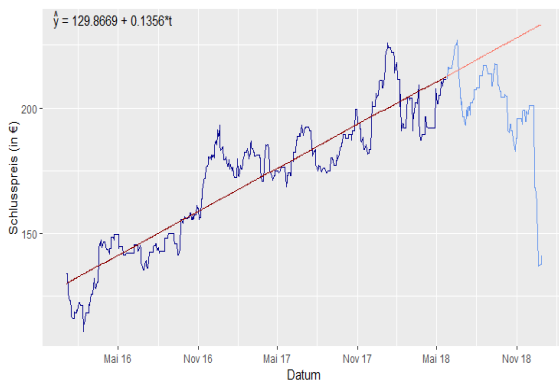
■ Geschätzter Schlusspreis (Trainingsdaten)   
 ■ Prognostizierter Schlusspreis (Testdaten)   
 ■ Schlusspreis (Testdaten)   
 ■ Schlusspreis (Trainingsdaten)



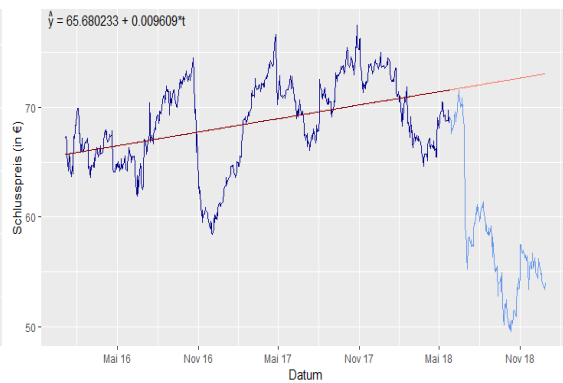
(14) Vorhersage des Aktienkurses von Evotec



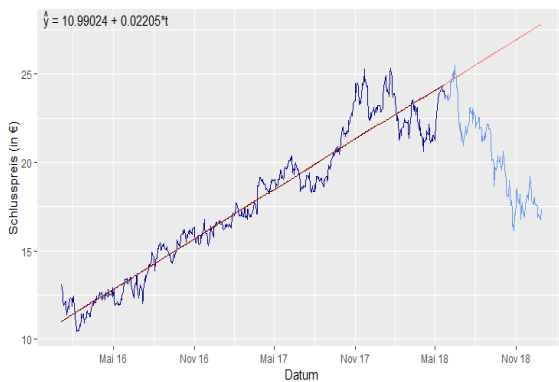
(15) Vorhersage des Aktienkurses von Exxon Mobil



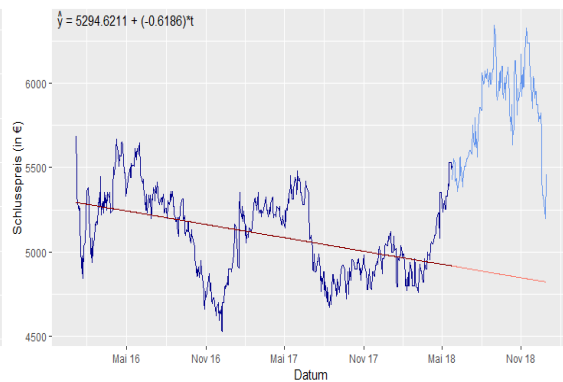
(16) Vorhersage des Aktienkurses von FedEx



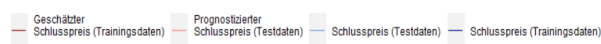
(17) Vorhersage des Aktienkurses von Fielmann

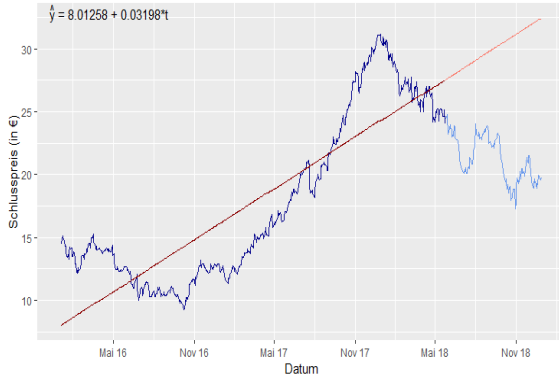


(18) Vorhersage des Aktienkurses von Infineon Technologies

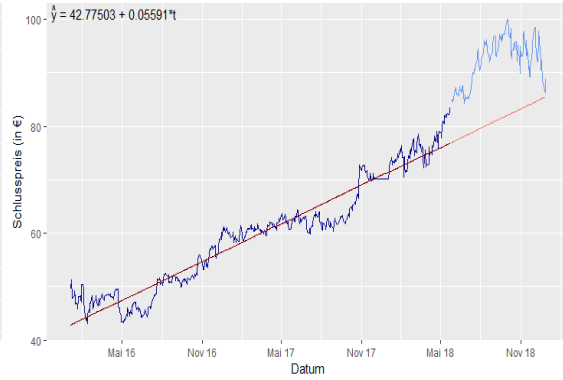


(19) Vorhersage des Aktienkurses von Lindt

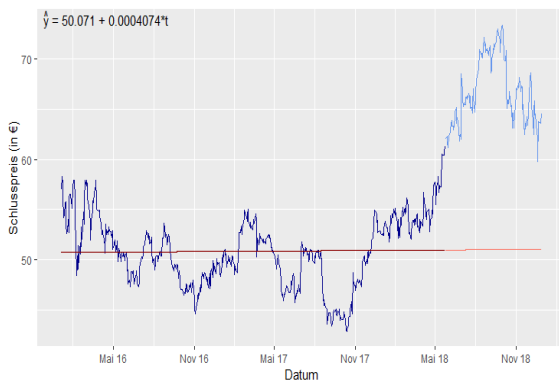




(20) Vorhersage des Aktienkurses von Lufthansa



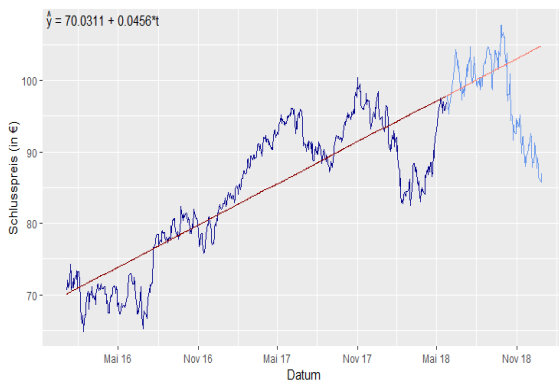
(21) Vorhersage des Aktienkurses von Microsoft



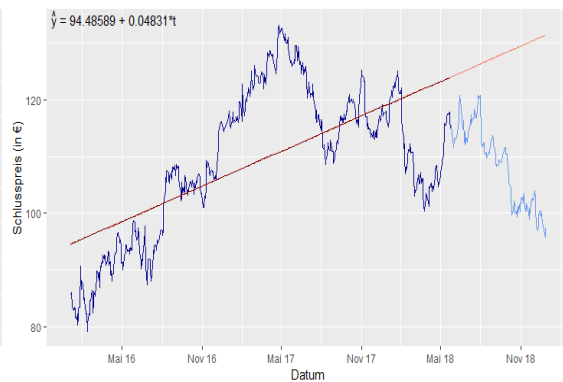
(22) Vorhersage des Aktienkurses von Nike



(23) Vorhersage des Aktienkurses von RWE



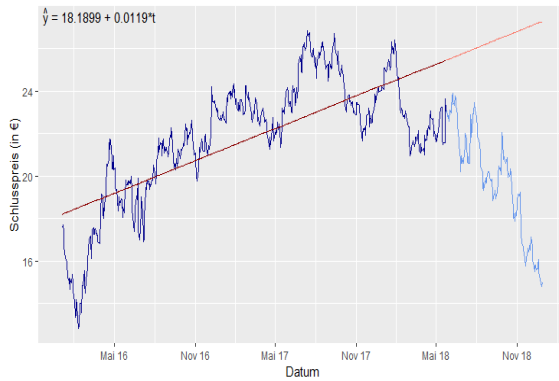
(24) Vorhersage des Aktienkurses von SAP



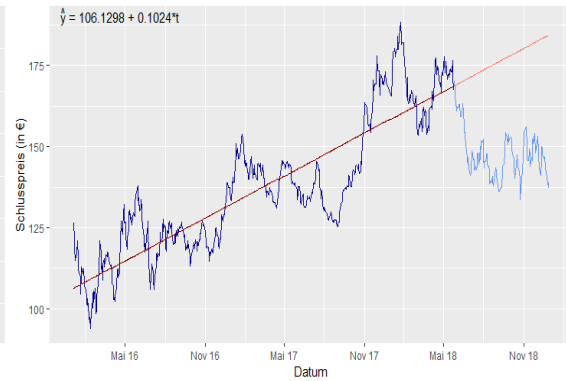
(25) Vorhersage des Aktienkurses von Siemens

— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)

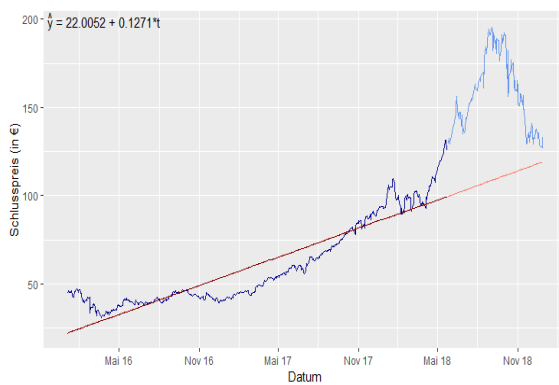




(26) Vorhersage des Aktienkurses von Thyssen Krupp



(27) Vorhersage des Aktienkurses von Volkswagen



(28) Vorhersage des Aktienkurses von Wirecard



Abbildung C.1: Vorhersage mithilfe der Linearen Regression



### C.1.2 RMSE zur Vorhersage mit der Linearen Regression

Tabelle C.1: RMSE zu den Linearen Regressionsvorhersagen

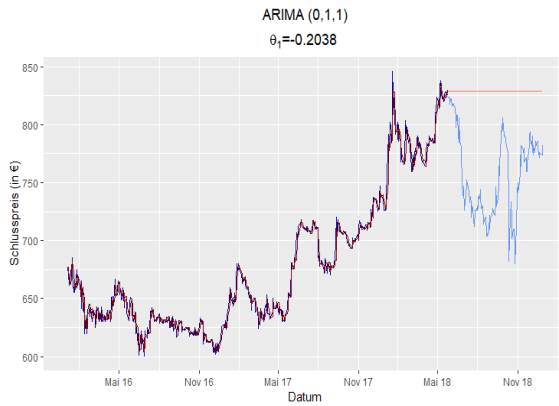
Unternehmen	Datensatz	Anzahl eingeleiteter Beobachtungen		
		300	500	762
Adidas	$E_{Train}$	9.57	10.90	13.34
	$E_{Test}$	13.78	10.06	24.98
Airbus	$E_{Train}$	2.74	3.53	5.44
	$E_{Test}$	19.44	13.79	9.07
Alphabet	$E_{Train}$	45.93	42.11	34.11
	$E_{Test}$	123.81	72.82	71.40
Audi	$E_{Train}$	38.69	30.45	34.00
	$E_{Test}$	30.48	79.14	47.00
BASF	$E_{Train}$	2.61	4.29	5.87
	$E_{Test}$	9.83	15.07	25.70
Bayer	$E_{Train}$	5.42	5.91	7.93
	$E_{Test}$	12.11	23.79	30.56
Beiersdorf	$E_{Train}$	3.63	3.72	3.80
	$E_{Test}$	5.56	5.92	5.32
Bertrandt	$E_{Train}$	7.65	7.88	7.18
	$E_{Test}$	8.59	17.23	10.65
BMW	$E_{Train}$	3.38	3.61	4.29
	$E_{Test}$	7.11	9.89	13.78
CocaCola	$E_{Train}$	1.65	1.19	1.19
	$E_{Test}$	4.16	5.29	3.92
Comdirect	$E_{Train}$	0.46	0.38	0.63
	$E_{Test}$	1.75	2.56	1.26
Continental	$E_{Train}$	16.59	12.13	0.37
	$E_{Test}$	34.85	35.02	71.45
Daimler	$E_{Train}$	2.60	4.17	3.83
	$E_{Test}$	2.23	11.31	16.53
Deutsche Telekom	$E_{Train}$	0.60	0.63	0.92
	$E_{Test}$	1.66	1.84	0.82
Eon	$E_{Train}$	0.52	0.70	0.87
	$E_{Test}$	0.37	1.26	0.85
Evotec	$E_{Train}$	2.08	2.64	2.30
	$E_{Test}$	1.76	2.34	2.63
Exxon Mobil	$E_{Train}$	3.79	3.71	4.83
	$E_{Test}$	3.46	7.33	5.68
FedEx	$E_{Train}$	10.29	9.44	10.21
	$E_{Test}$	31.34	27.02	29.99
Fielmann	$E_{Train}$	3.00	3.84	3.56
	$E_{Test}$	3.43	11.85	15.50
Infineon Technologies	$E_{Train}$	1.13	1.35	1.02
	$E_{Test}$	3.27	6.57	6.60
Lindt & Sprüngli	$E_{Train}$	185.50	269.73	220.97
	$E_{Test}$	380.90	745.67	1007.51
Lufthansa	$E_{Train}$	1.49	3.69	3.31
	$E_{Test}$	1.33	9.88	9.14
Microsoft	$E_{Train}$	2.66	3.40	2.68
	$E_{Test}$	7.06	6.93	11.63

Unternehmen	Datensatz	Anzahl eingelesener Beobachtungen		
		300	500	762
Nike	$E_{Train}$	2.21	4.46	3.52
	$E_{Test}$	9.72	7.93	16.06
RWE	$E_{Train}$	1.73	1.98	1.89
	$E_{Test}$	3.24	3.12	2.61
SAP	$E_{Train}$	5.45	4.50	5.03
	$E_{Test}$	11.08	6.72	7.66
Siemens	$E_{Train}$	5.31	5.67	9.82
	$E_{Test}$	8.97	5.57	20.94
Thyssen Krupp	$E_{Train}$	1.21	1.38	1.99
	$E_{Test}$	2.93	4.19	7.12
Volkswagen	$E_{Train}$	10.05	12.68	10.16
	$E_{Test}$	6.59	25.32	30.43
Wirecard	$E_{Train}$	11.74	8.92	10.17
	$E_{Test}$	50.75	30.89	53.12

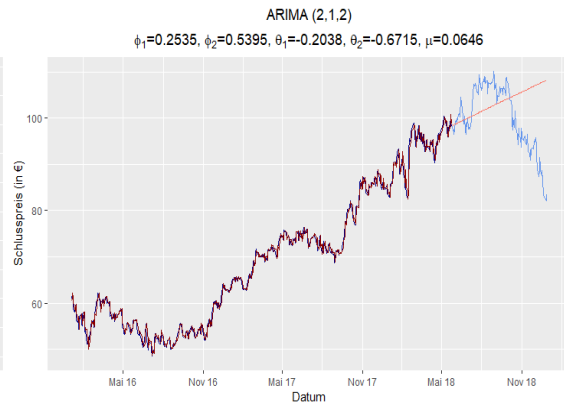


## C.2 Die Vorhersage mit dem ARIMA-Modell

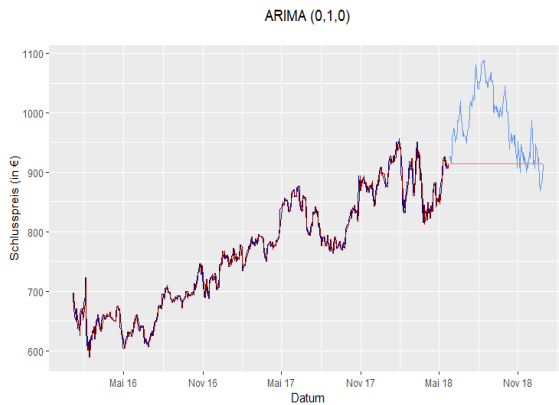
### C.2.1 Die Abbildungen zur Vorhersage mit dem ARIMA-Modell



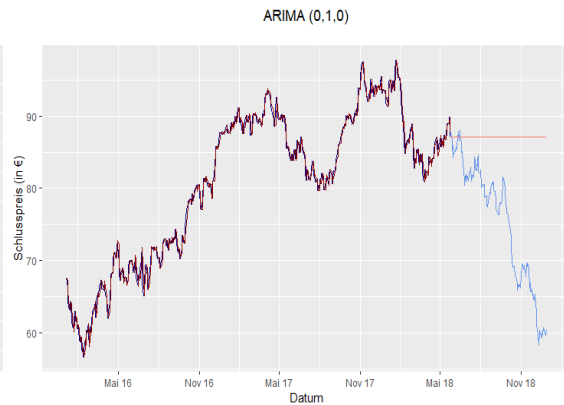
(1) Vorhersage des Aktienkurses von Audi



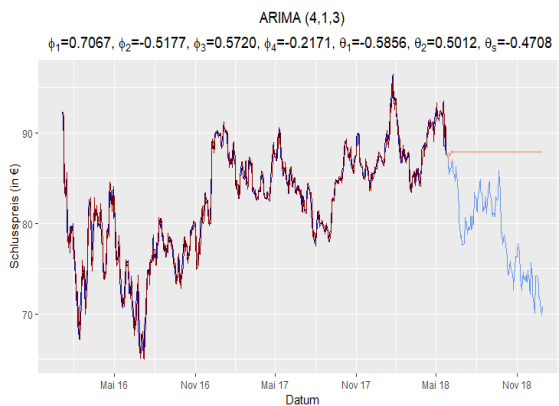
(2) Vorhersage des Aktienkurses von Airbus



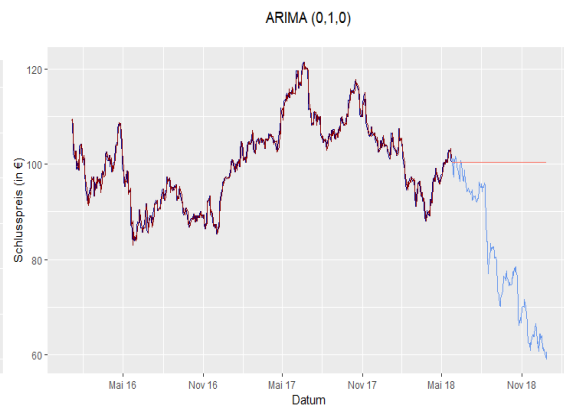
(3) Vorhersage des Aktienkurses von Alphabet



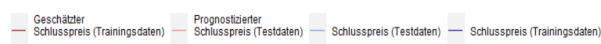
(4) Vorhersage des Aktienkurses von BASF

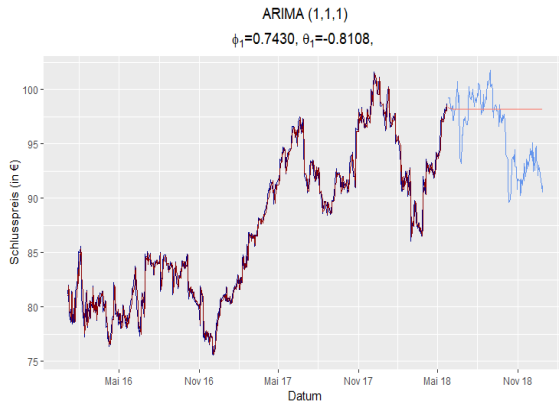


(5) Vorhersage des Aktienkurses von BMW

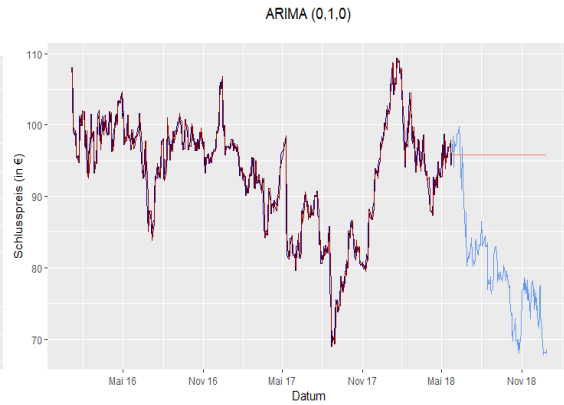


(6) Vorhersage des Aktienkurses von Bayer

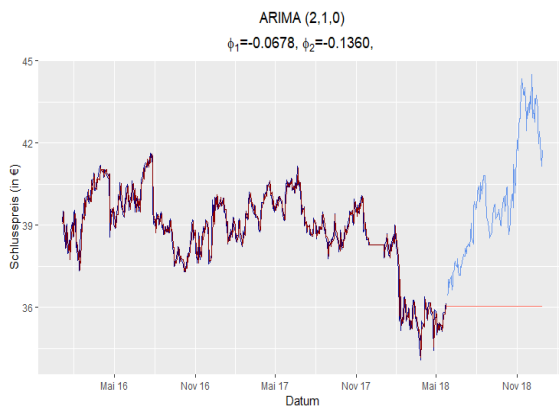




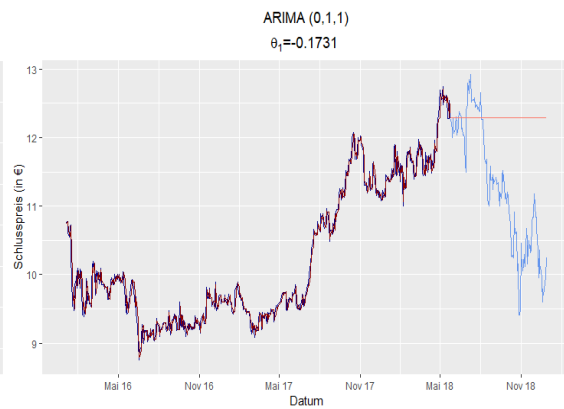
(7) Vorhersage des Aktienkurses von Beiersdorf



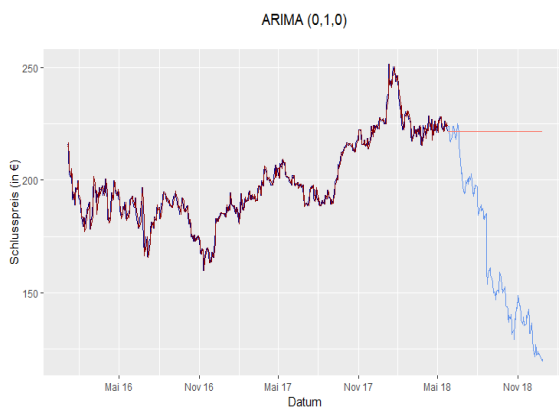
(8) Vorhersage des Aktienkurses von Bertrandt



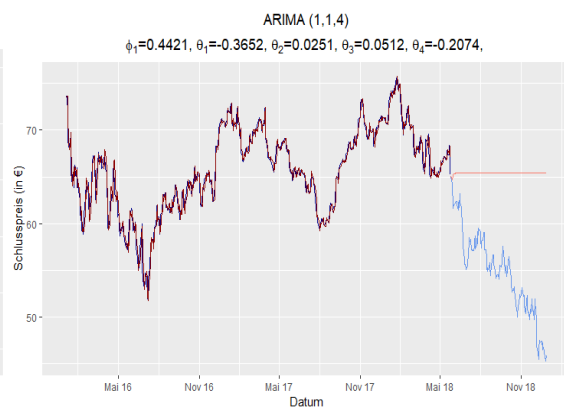
(9) Vorhersage des Aktienkurses von Comdirect



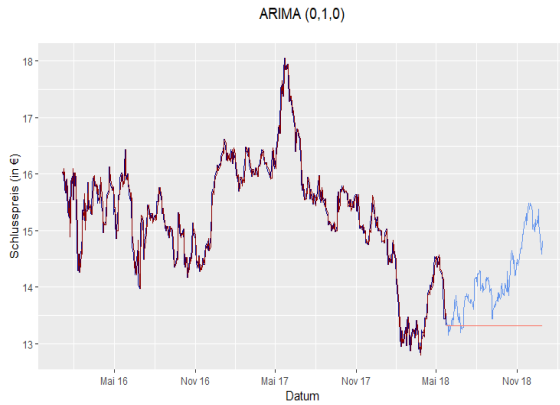
(10) Vorhersage des Aktienkurses von Comdirect



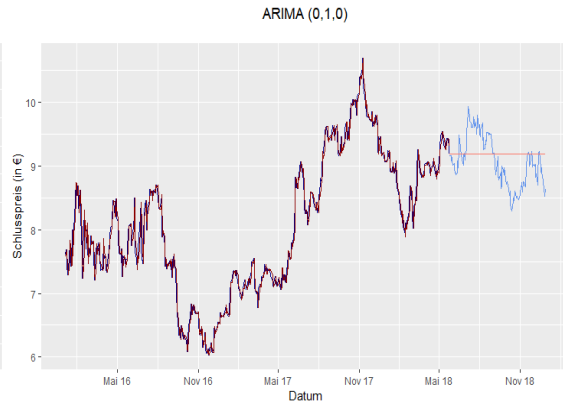
(11) Vorhersage des Aktienkurses von Continental



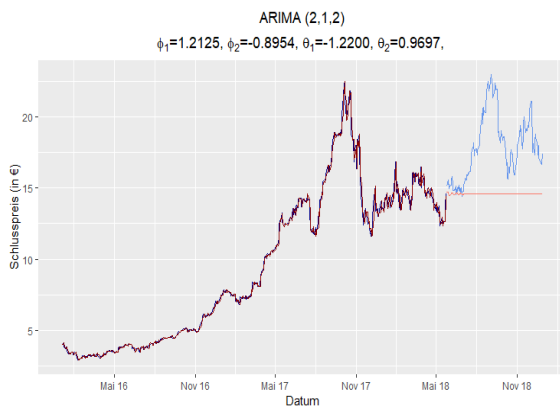
(12) Vorhersage des Aktienkurses von Daimler



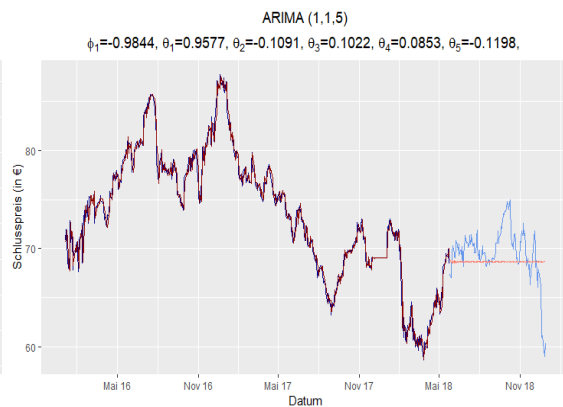
(13) Vorhersage des Aktienkurses von Deutsche Telekom



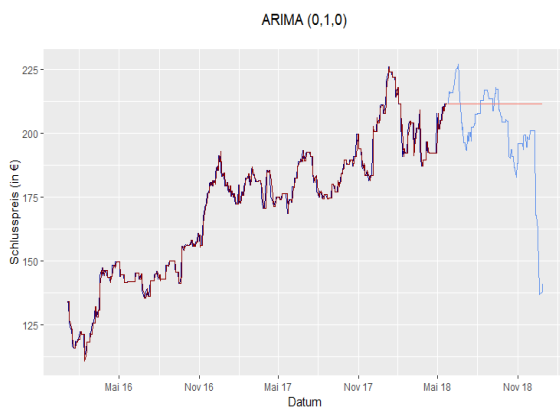
(14) Vorhersage des Aktienkurses von Eon



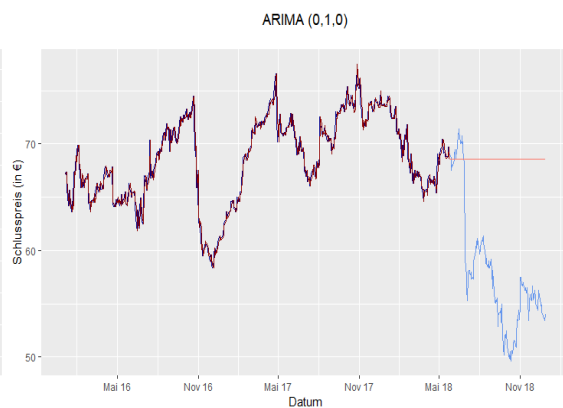
(15) Vorhersage des Aktienkurses von Evotec



(16) Vorhersage des Aktienkurses von Exxon Mobil



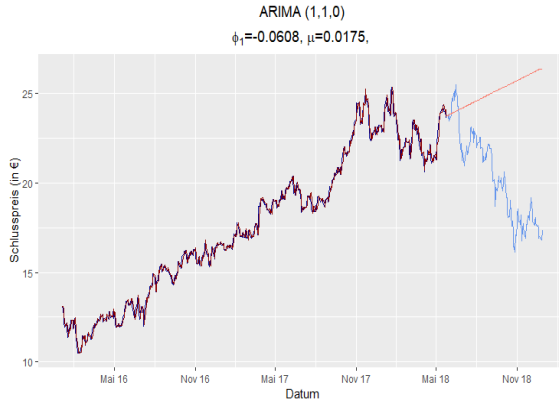
(17) Vorhersage des Aktienkurses von FedEx



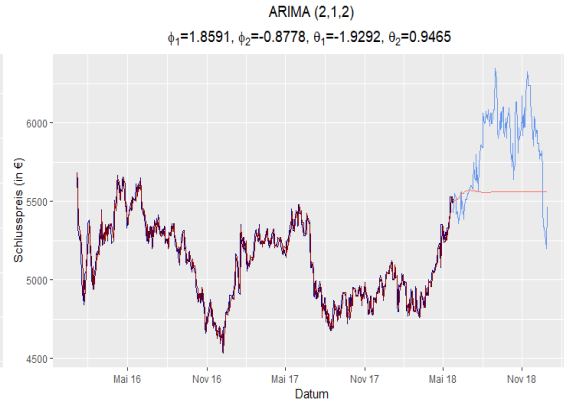
(18) Vorhersage des Aktienkurses von Fielmann

— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)

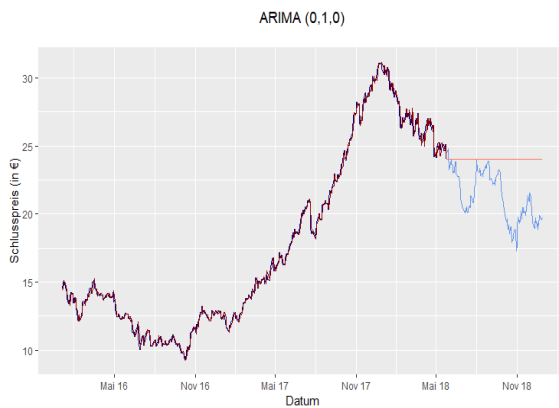




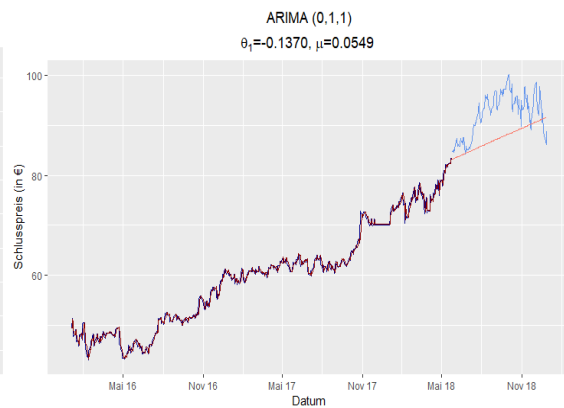
(19) Vorhersage des Aktienkurses von Infineon Technologies



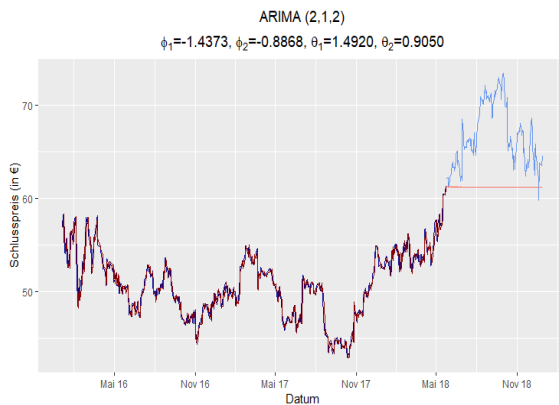
(20) Vorhersage des Aktienkurses von Lindt



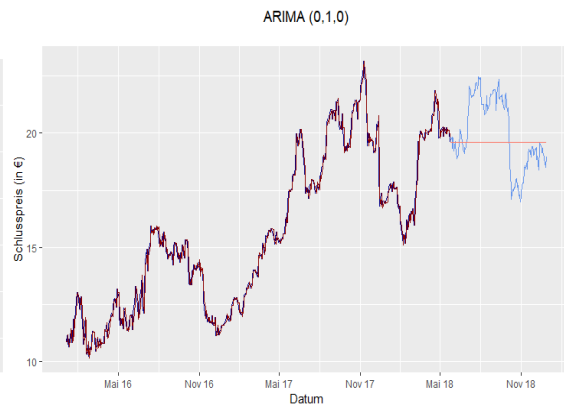
(21) Vorhersage des Aktienkurses von Lufthansa



(22) Vorhersage des Aktienkurses von Microsoft

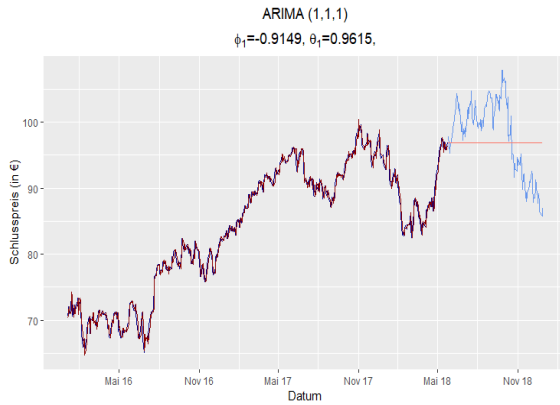


(23) Vorhersage des Aktienkurses von Nike

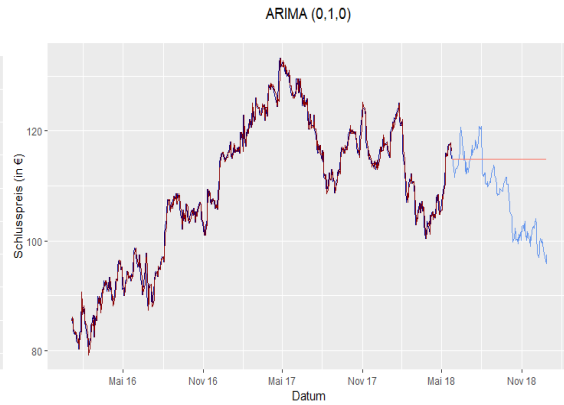


(24) Vorhersage des Aktienkurses von RWE

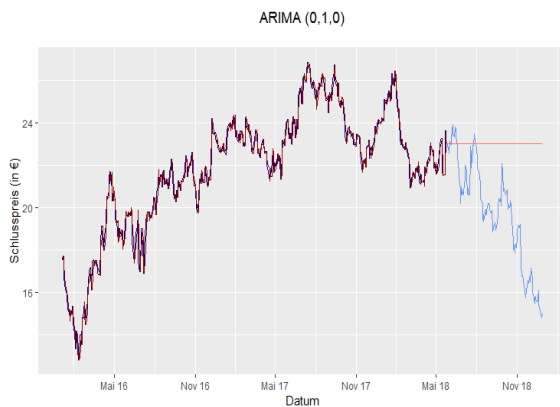
— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)



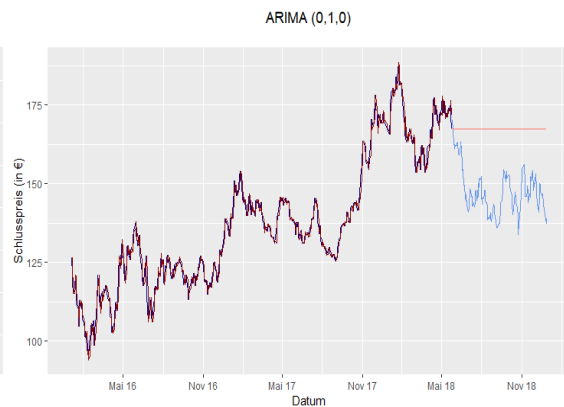
(25) Vorhersage des Aktienkurses von SAP



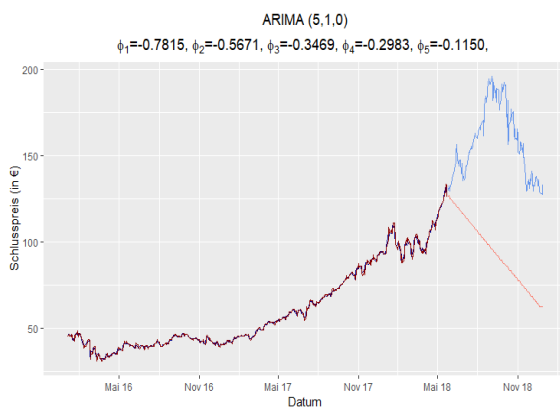
(26) Vorhersage des Aktienkurses von Siemens



(27) Vorhersage des Aktienkurses von Thyssen Krupp



(28) Vorhersage des Aktienkurses von Volkswagen



(29) Vorhersage des Aktienkurses von Wirecard

— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)

Abbildung C.2: Vorhersage der Aktienkurse mithilfe des ARIMA-Modells



## C.2.2 Der RMSE zum ARIMA-Modell

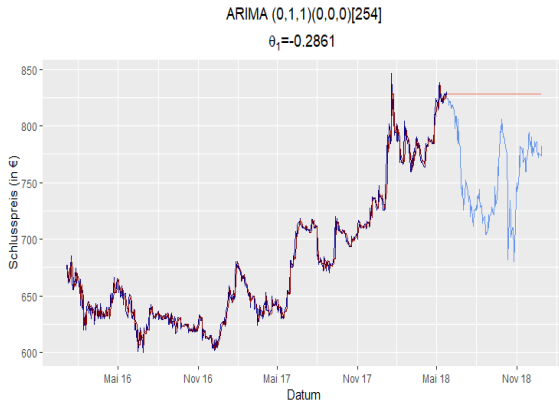
Tabelle C.4: RMSE der ARIMA-Modell Vorhersagen

Unternehmen	Datensatz	Anzahl eingeleiteter Beobachtungen		
		300	500	762
Adidas	$E_{Train}$	3.21	2.81	2.74
	$E_{Test}$	16.19	17.33	13.14
Airbus	$E_{Train}$	1.50	1.26	1.11
	$E_{Test}$	14.15	17.34	9.04
Alphabet	$E_{Train}$	14.47	12.07	11.2
	$E_{Test}$	106.06	96.28	86.29
Audi	$E_{Train}$	9.25	8.07	7.41
	$E_{Test}$	43.40	35.08	80.51
BASF	$E_{Train}$	1.01	0.93	0.93
	$E_{Test}$	11.18	11.56	14.23
Bayer	$E_{Train}$	1.40	1.22	1.34
	$E_{Test}$	4.53	24.15	23.83
Beiersdorf	$E_{Train}$	1.02	0.89	0.92
	$E_{Test}$	5.25	6.20	3.87
Bertrandt	$E_{Train}$	1.67	1.78	1.7
	$E_{Test}$	5.67	8.55	17.41
BMW	$E_{Train}$	1.00	0.91	1.12
	$E_{Test}$	4.26	7.41	9.87
CocaCola	$E_{Train}$	0.34	0.30	0.33
	$E_{Test}$	2.67	1.91	4.51
Comdirect	$E_{Train}$	0.16	0.13	0.12
	$E_{Test}$	0.92	1.63	1.35
Continental	$E_{Train}$	3.28	2.52	2.84
	$E_{Test}$	18.05	43.76	63.56
Daimler	$E_{Train}$	0.74	0.69	0.84
	$E_{Test}$	4.70	6.43	11.20
Deutsche Telekom	$E_{Train}$	0.14	0.15	0.18
	$E_{Test}$	0.99	0.60	1.04
Eon	$E_{Train}$	0.12	0.12	0.15
	$E_{Test}$	0.27	0.77	0.39
Evotec	$E_{Train}$	0.57	0.64	0.36
	$E_{Test}$	1.44	2.32	4.05
Exxon Mobil	$E_{Train}$	0.81	0.74	0.84
	$E_{Test}$	5.97	2.92	2.65
FedEx	$E_{Train}$	3.22	2.81	2.58
	$E_{Test}$	23.99	20.81	19.32
Fielmann	$E_{Train}$	1.00	0.86	0.78
	$E_{Test}$	2.89	6.64	11.99
Infineon Technologies	$E_{Train}$	0.39	0.34	0.30
	$E_{Test}$	2.22	3.65	5.61
Lindt & Sprüngli	$E_{Train}$	64.33	58.63	58.88
	$E_{Test}$	392.30	201.15	279.69
Lufthansa	$E_{Train}$	0.46	0.40	0.35
	$E_{Test}$	1.47	2.27	3.25
Microsoft	$E_{Train}$	1.00	0.85	0.81
	$E_{Test}$	9.42	4.80	5.97

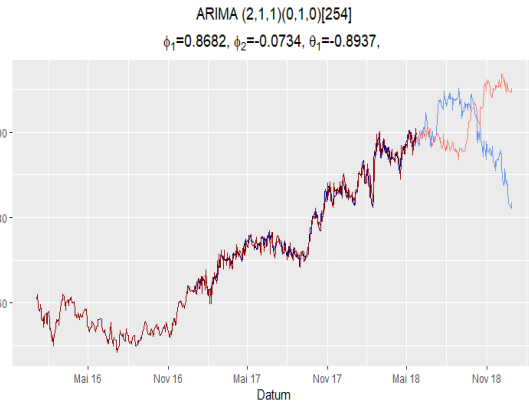
Unternehmen	Datensatz	Anzahl eingelesener Beobachtungen		
		300	500	762
Nike	$E_{Train}$	0.88	0.82	0.82
	$E_{Test}$	11.50	3.33	6.37
RWE	$E_{Train}$	0.38	0.34	0.35
	$E_{Test}$	2.57	2.22	1.58
SAP	$E_{Train}$	1.14	0.92	0.89
	$E_{Test}$	14.50	6.89	5.60
Siemens	$E_{Train}$	1.34	1.40	1.40
	$E_{Test}$	9.52	8.48	9.20
Thyssen Krupp	$E_{Train}$	0.42	0.39	0.43
	$E_{Test}$	3.61	3.61	3.97
Volkswagen	$E_{Train}$	2.70	2.31	2.50
	$E_{Test}$	7.56	4.48	21.07
Wirecard	$E_{Train}$	2.96	1.73	1.37
	$E_{Test}$	61.68	35.04	69.46



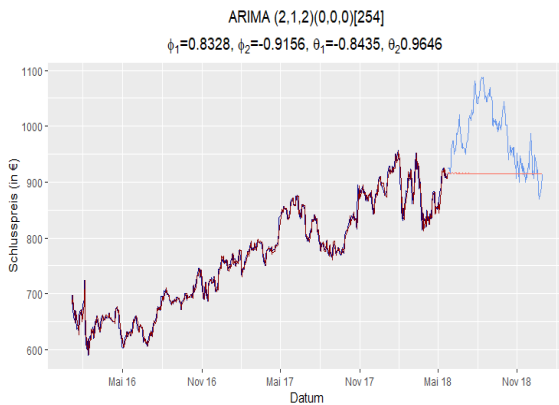
### C.3 Die Abbildungen zur Vorhersage mit dem SARIMA-Modell



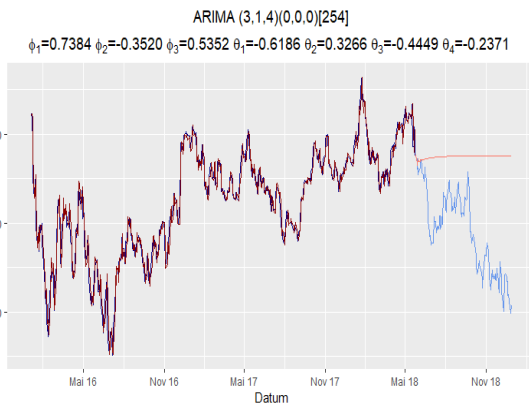
(1) Vorhersage des Aktienkurses von Audi



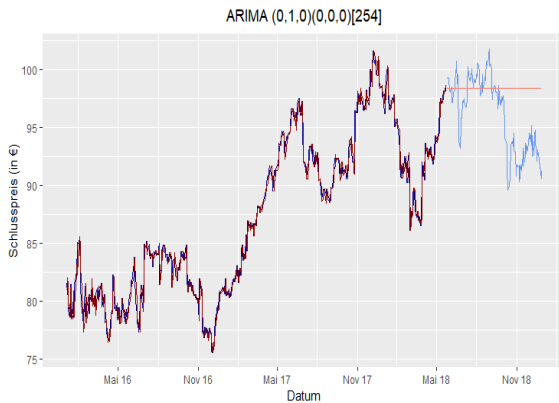
(2) Vorhersage des Aktienkurses von Airbus



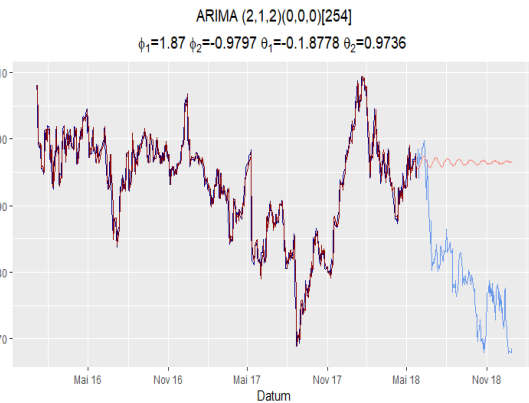
(3) Vorhersage des Aktienkurses von Alphabet



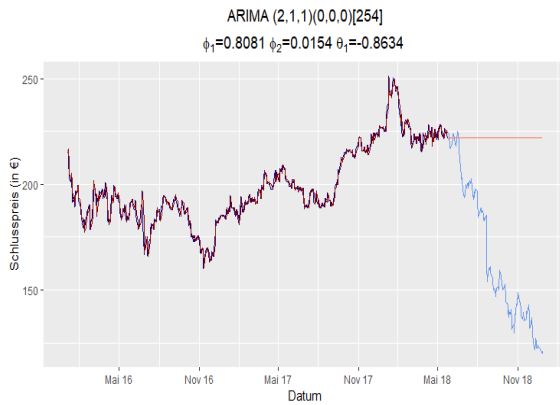
(4) Vorhersage des Aktienkurses von BMW



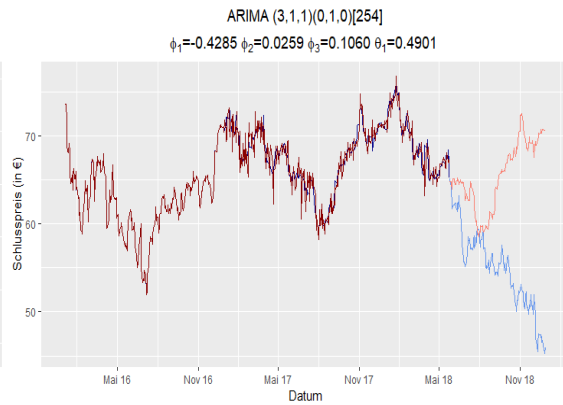
(5) Vorhersage des Aktienkurses von Beiersdorf



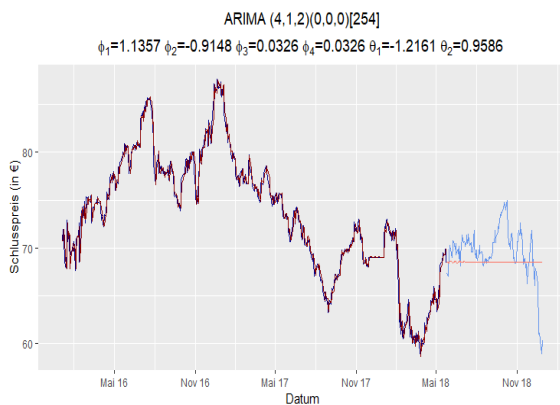
(6) Vorhersage des Aktienkurses von Bertrandt



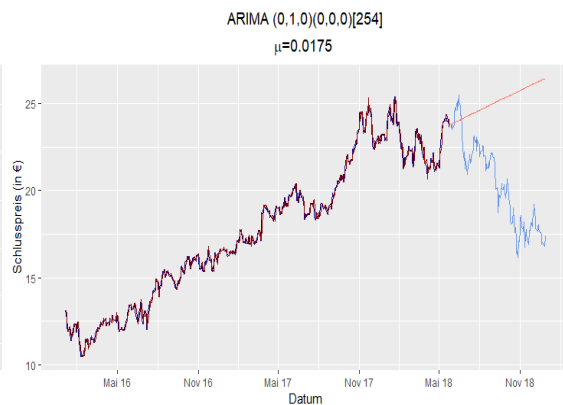
(7) Vorhersage des Aktienkurses von Continental



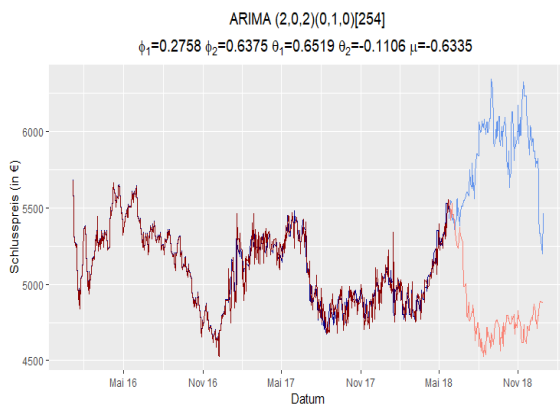
(8) Vorhersage des Aktienkurses von Daimler



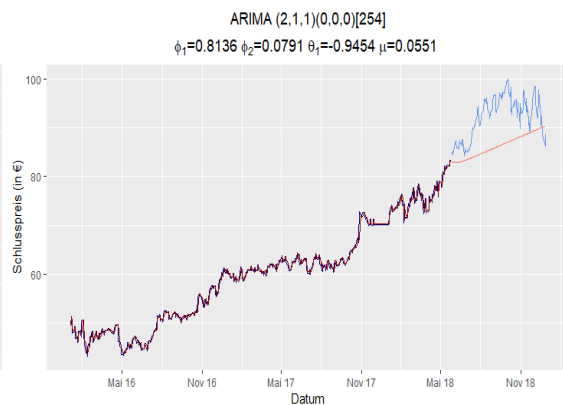
(9) Vorhersage des Aktienkurses von Exxon Mobil



(10) Vorhersage des Aktienkurses von Infineon Technologies



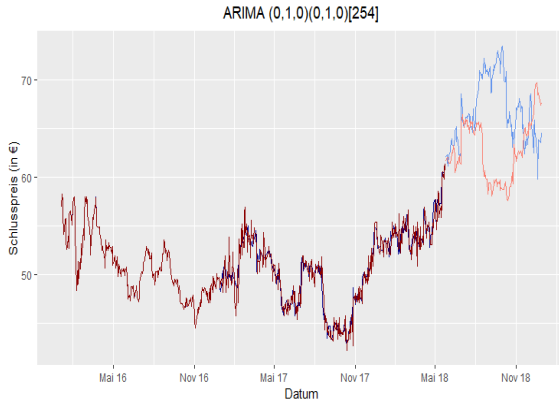
(11) Vorhersage des Aktienkurses von Lindt



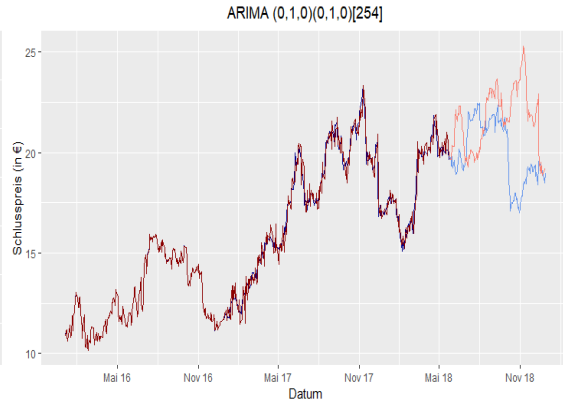
(v) Vorhersage des Aktienkurses von Microsoft

— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)

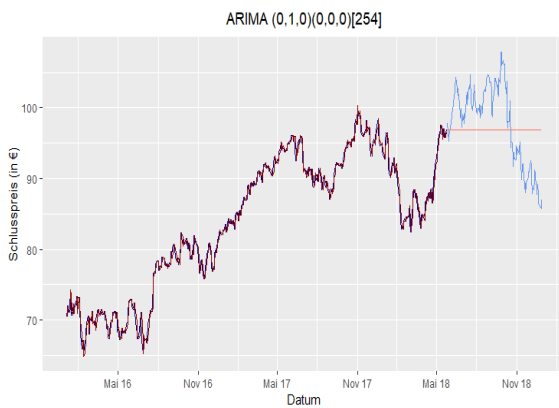




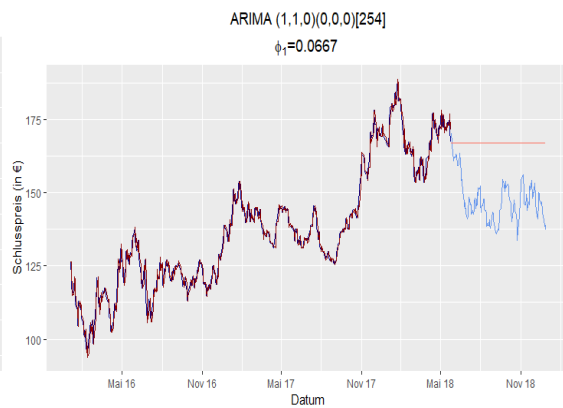
(12) Vorhersage des Aktienkurses von Nike



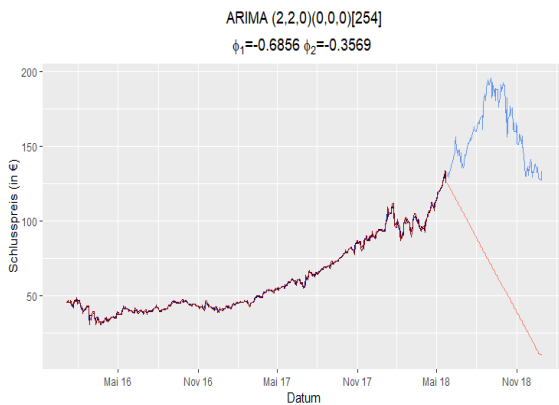
(13) Vorhersage des Aktienkurses von RWE



(14) Vorhersage des Aktienkurses von SAP



(15) Vorhersage des Aktienkurses von Volkswagen



(16) Vorhersage des Aktienkurses von Wirecard



Abbildung C.3: Vorhersage der Aktienkurse mithilfe des ARIMA-Modells



### C.3.1 Tabelle der Abweichungen zum SARIMA- Modell

Tabelle C.7: RMSE der SARIMA-Modell Vorhersagen

Unternehmen	Datensatz	Anzahl eingelesener Beobachtungen		
		300	500	762
Adidas	$E_{Train}$	3.21	2.81	2.63
	$E_{Test}$	16.19	17.33	10.94
Airbus	$E_{Train}$	1.50	1.26	1.18
	$E_{Test}$	14.15	17.34	12.10
Alphabet	$E_{Train}$	14.47	12.07	11.08
	$E_{Test}$	106.06	96.28	85.89
Audi	$E_{Train}$	9.47	8.07	7.41
	$E_{Test}$	43.73	35.08	80.50
BASF	$E_{Train}$	1.01	0.93	0.93
	$E_{Test}$	11.18	11.56	14.23
Bayer	$E_{Train}$	1.41	1.22	1.34
	$E_{Test}$	9.81	24.15	23.83
Beiersdorf	$E_{Train}$	1.02	0.89	0.92
	$E_{Test}$	5.25	6.20	3.88
Bertrandt	$E_{Train}$	1.70	1.78	1.71
	$E_{Test}$	5.60	8.55	18.04
BMW	$E_{Train}$	1.00	0.91	1.12
	$E_{Test}$	4.26	7.43	9.54
CocaCola	$E_{Train}$	0.34	0.30	0.33
	$E_{Test}$	2.67	1.93	4.51
Comdirect	$E_{Train}$	0.16	0.13	0.12
	$E_{Test}$	0.92	1.63	1.35
Continental	$E_{Train}$	3.28	2.52	2.83
	$E_{Test}$	18.05	43.76	64.11
Daimler	$E_{Train}$	0.77	0.69	0.89
	$E_{Test}$	4.91	6.43	12.81
Deutsche Telekom	$E_{Train}$	0.14	0.15	0.18
	$E_{Test}$	0.99	0.60	1.04
Eon	$E_{Train}$	0.12	0.12	0.15
	$E_{Test}$	0.78	0.77	0.39
Evotec	$E_{Train}$	0.57	0.64	0.36
	$E_{Test}$	1.44	2.32	4.05
Exxon Mobil	$E_{Train}$	0.81	0.74	0.86
	$E_{Test}$	5.97	2.92	2.71
FedEx	$E_{Train}$	3.22	2.81	2.58
	$E_{Test}$	23.99	20.81	19.32
Fielmann	$E_{Train}$	0.99	0.86	0.78
	$E_{Test}$	5.85	6.64	11.99
Infineon Technologies	$E_{Train}$	0.33	0.34	0.30
	$E_{Test}$	2.25	3.65	5.62
Lindt & Sprüngli	$E_{Train}$	64.33	58.48	66.07
	$E_{Test}$	392.30	201.98	1117.501
Lufthansa	$E_{Train}$	0.46	0.40	0.35
	$E_{Test}$	1.47	2.56	3.25
Microsoft	$E_{Train}$	1.01	0.85	0.81
	$E_{Test}$	10.45	4.86	6.59

Unternehmen	Datensatz	Anzahl eingelesener Beobachtungen		
		300	500	762
Nike	$E_{Train}$	0.88	0.83	0.86
	$E_{Test}$	11.50	3.39	6.71
RWE	$E_{Train}$	0.38	0.34	0.37
	$E_{Test}$	2.54	2.22	3.01
SAP	$E_{Train}$	1.14	0.93	0.89
	$E_{Test}$	14.50	6.91	5.60
Siemens	$E_{Train}$	1.34	1.40	1.40
	$E_{Test}$	9.52	8.48	9.20
Thyssen Krupp	$E_{Train}$	0.42	0.39	0.43
	$E_{Test}$	3.61	3.61	3.97
Volkswagen	$E_{Train}$	2.70	2.29	2.49
	$E_{Test}$	7.56	5.48	20.78
Wirecard	$E_{Train}$	2.96	1.73	1.41
	$E_{Test}$	61.68	35.04	98.43



### C.3.2 Tabelle der Abweichungen zum ARIMA/SARIMA - Modell

Tabelle C.10: RMSE der kombinierten SARIMA-Modell Vorhersagen

Unternehmen	Datensatz	Anzahl eingelesener Beobachtungen		
		300	500	762
Adidas	$E_{Train}$	3.21	2.81	<b>2.63</b>
	$E_{Test}$	16.19	17.33	<b>10.94</b>
Airbus	$E_{Train}$	1.50	1.26	1.11
	$E_{Test}$	14.15	17.34	9.04
Alphabet	$E_{Train}$	14.47	12.07	<b>11.08</b>
	$E_{Test}$	106.06	96.28	<b>85.89</b>
Audi	$E_{Train}$	9.25	8.07	<b>7.41</b>
	$E_{Test}$	43.40	35.08	<b>80.50</b>
BASF	$E_{Train}$	1.01	0.93	0.93
	$E_{Test}$	11.18	11.56	14.23
Bayer	$E_{Train}$	1.40	1.22	1.34
	$E_{Test}$	4.53	24.15	23.83
Beiersdorf	$E_{Train}$	1.02	0.89	0.92
	$E_{Test}$	5.25	6.20	3.87
Bertrandt	$E_{Train}$	<b>1.70</b>	1.78	1.7
	$E_{Test}$	<b>5.60</b>	8.55	17.41
BMW	$E_{Train}$	1.00	0.91	<b>1.12</b>
	$E_{Test}$	4.26	7.41	<b>9.54</b>
CocaCola	$E_{Train}$	0.34	0.30	0.33
	$E_{Test}$	2.67	1.91	4.51
Comdirect	$E_{Train}$	0.16	0.13	0.12
	$E_{Test}$	0.92	1.63	1.35
Continental	$E_{Train}$	3.28	2.52	2.84
	$E_{Test}$	18.05	43.76	63.56
Daimler	$E_{Train}$	0.74	0.69	0.84
	$E_{Test}$	4.70	6.43	11.20
Deutsche Telekom	$E_{Train}$	0.14	0.15	0.18
	$E_{Test}$	0.99	0.60	1.04
Eon	$E_{Train}$	0.12	0.12	0.15
	$E_{Test}$	0.27	0.77	0.39
Evotec	$E_{Train}$	0.57	0.64	0.36
	$E_{Test}$	1.44	2.32	4.05
Exxon Mobil	$E_{Train}$	0.81	0.74	0.84
	$E_{Test}$	5.97	2.92	2.65
FedEx	$E_{Train}$	3.22	2.81	2.58
	$E_{Test}$	23.99	20.81	19.32
Fielmann	$E_{Train}$	1.00	0.86	0.78
	$E_{Test}$	2.89	6.64	11.99
Infineon Technologies	$E_{Train}$	0.39	0.34	0.30
	$E_{Test}$	2.22	3.65	5.61
Lindt & Sprüngli	$E_{Train}$	64.33	58.63	58.88
	$E_{Test}$	392.30	201.15	279.69
Lufthansa	$E_{Train}$	0.46	0.40	0.35
	$E_{Test}$	1.47	2.27	3.25
Microsoft	$E_{Train}$	1.00	0.85	0.81
	$E_{Test}$	9.42	4.80	5.97

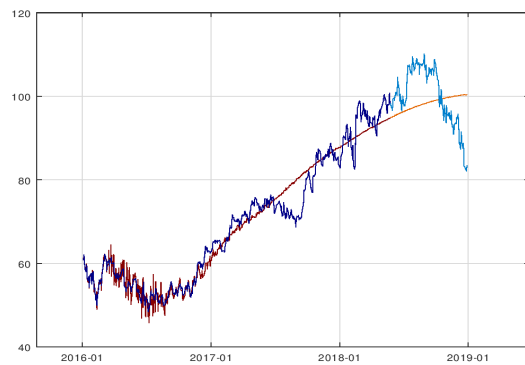
Unternehmen	Datensatz	Anzahl eingelesener Beobachtungen		
		300	500	762
Nike	$E_{Train}$	0.88	0.82	0.82
	$E_{Test}$	11.50	3.33	6.37
RWE	$E_{Train}$	0.38	0.34	0.35
	$E_{Test}$	2.57	2.22	1.58
SAP	$E_{Train}$	1.14	0.92	0.89
	$E_{Test}$	14.50	6.89	5.60
Siemens	$E_{Train}$	1.34	1.40	1.40
	$E_{Test}$	9.52	8.48	9.20
Thyssen Krupp	$E_{Train}$	0.42	0.39	0.43
	$E_{Test}$	3.61	3.61	3.97
Volkswagen	$E_{Train}$	2.70	2.31	<b>2.49</b>
	$E_{Test}$	7.56	4.48	<b>20.78</b>
Wirecard	$E_{Train}$	2.96	1.73	1.37
	$E_{Test}$	61.68	35.04	69.46

## C.4 Die Vorhersage mit dem Predictive Neural Network

### C.4.1 Die Abbildungen zur Vorhersage mit dem PrNN



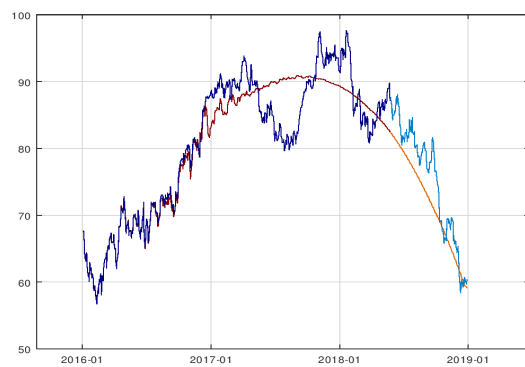
(1) Vorhersage des Aktienkurses von Audi



(2) Vorhersage des Aktienkurses von Airbus



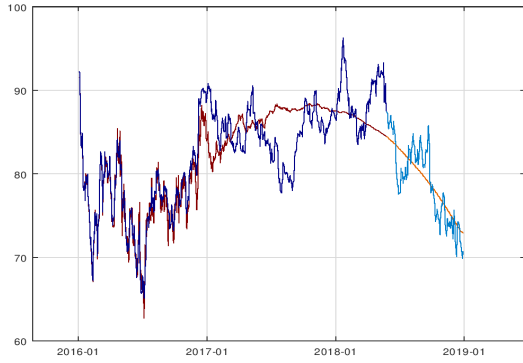
(3) Vorhersage des Aktienkurses von Alphabet



(4) Vorhersage des Aktienkurses von BASF

Geschätzter Schlusspreis (Trainingsdaten)
  Prognostizierter Schlusspreis (Testdaten)
  Schlusspreis (Testdaten)
  Schlusspreis (Trainingsdaten)

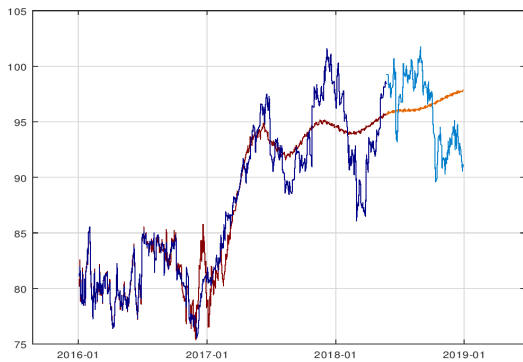




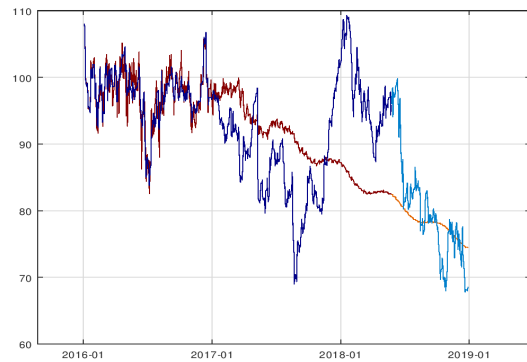
(5) Vorhersage des Aktienkurses von BMW



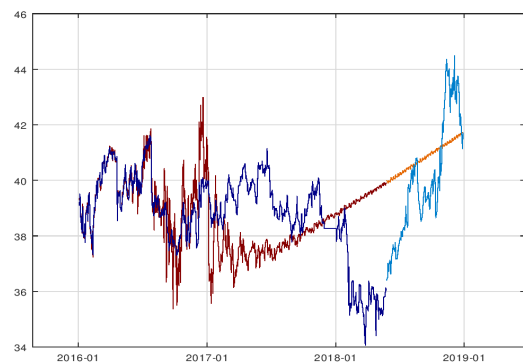
(6) Vorhersage des Aktienkurses von Bayer



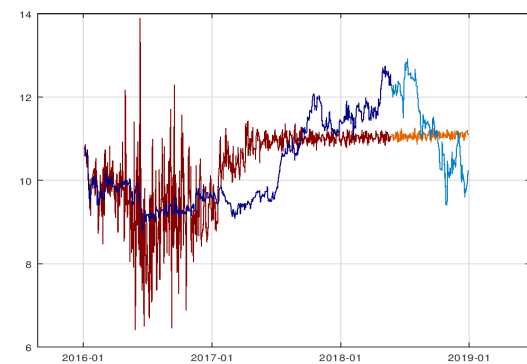
(7) Vorhersage des Aktienkurses von Beiersdorf



(8) Vorhersage des Aktienkurses von Bertrandt

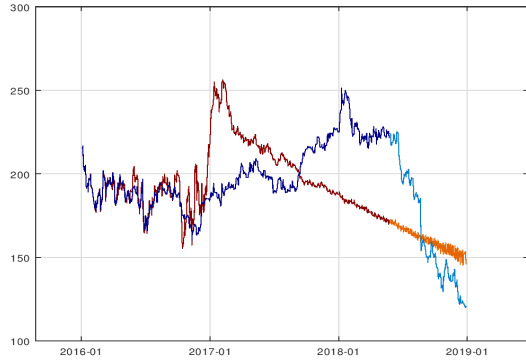


(9) Vorhersage des Aktienkurses von Coca Cola



(10) Vorhersage des Aktienkurses von Comdirect

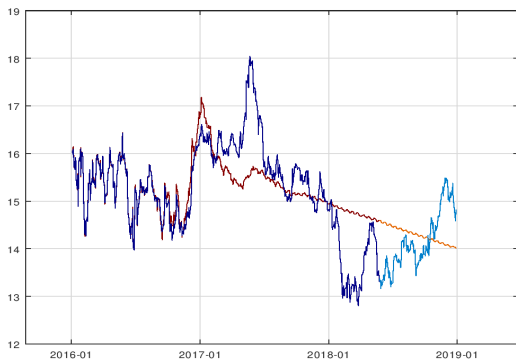
— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)



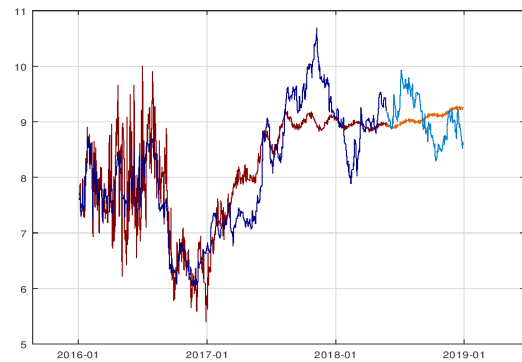
(11) Vorhersage des Aktienkurses von der Continental



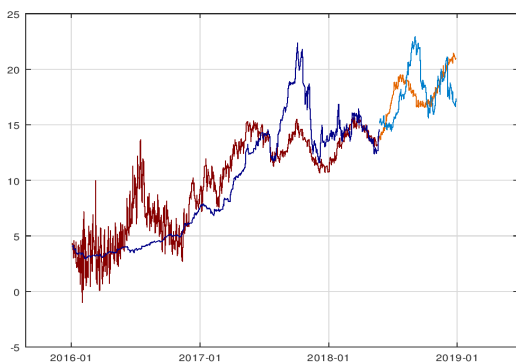
(12) Vorhersage des Aktienkurses von Daimler



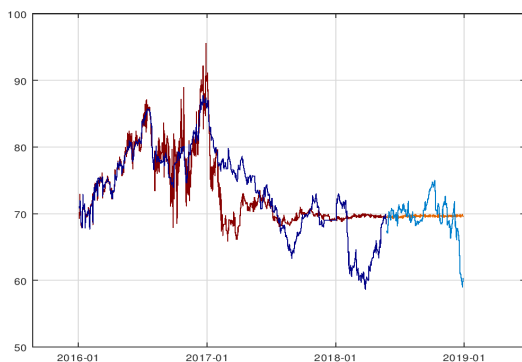
(13) Vorhersage des Aktienkurses von der Deutschen Telekom



(14) Vorhersage des Aktienkurses von Eon

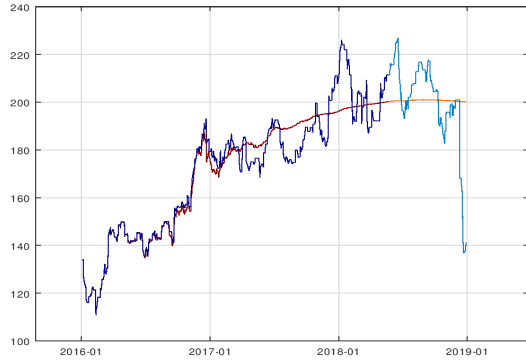


(15) Vorhersage des Aktienkurses von Evotec

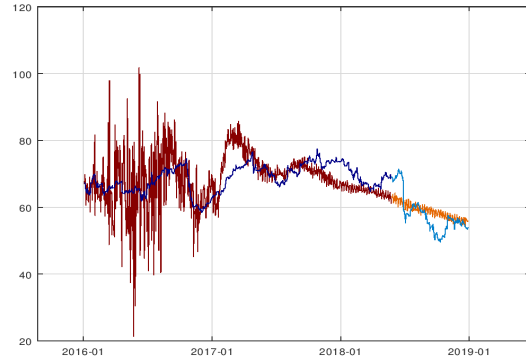


(16) Vorhersage des Aktienkurses von ExxonMobil

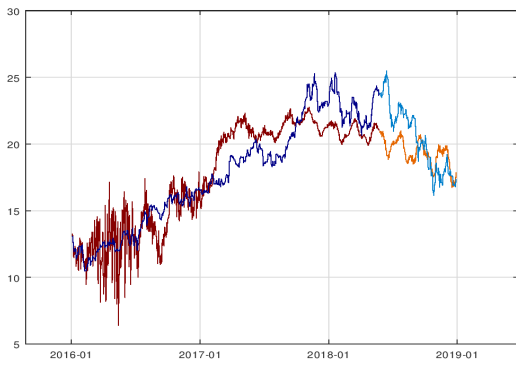
— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)



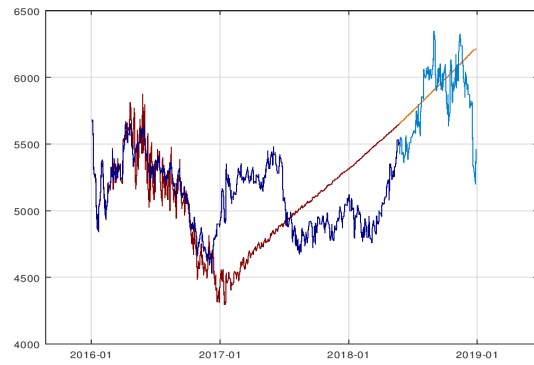
(17) Vorhersage des Aktienkurses von FedEx



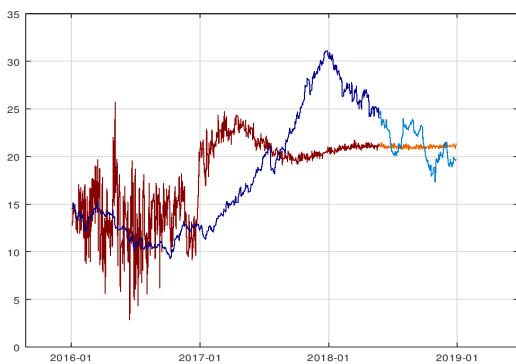
(18) Vorhersage des Aktienkurses von der Fielmann



(19) Vorhersage des Aktienkurses von Infineon Technologies



(20) Vorhersage des Aktienkurses von Lindt & Sprüngli



(21) Vorhersage des Aktienkurses von der Lufthansa

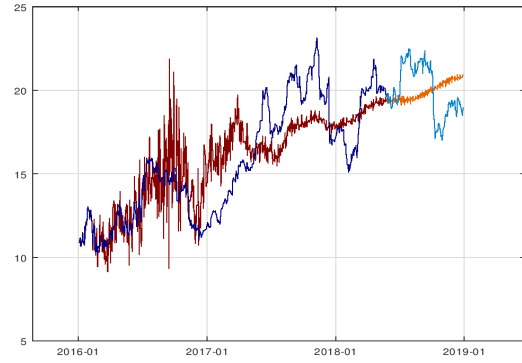


(22) Vorhersage des Aktienkurses von Microsoft

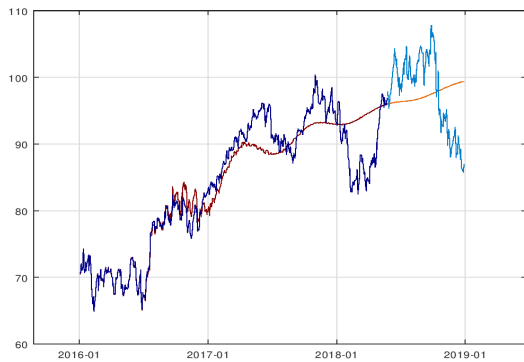
— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)



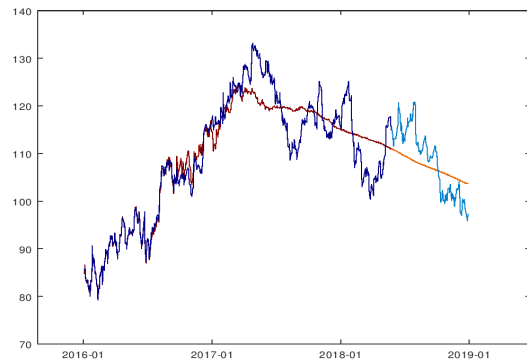
(23) Vorhersage des Aktienkurses von Nike



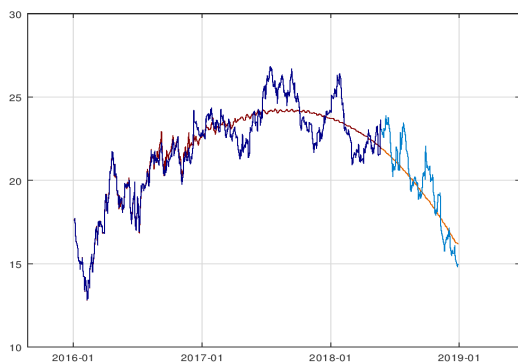
(24) Vorhersage des Aktienkurses von RWE



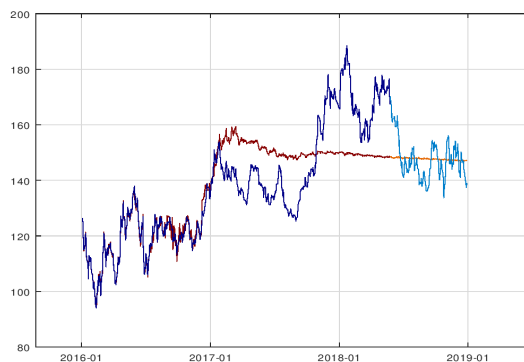
(25) Vorhersage des Aktienkurses von SAP



(26) Vorhersage des Aktienkurses von Siemens

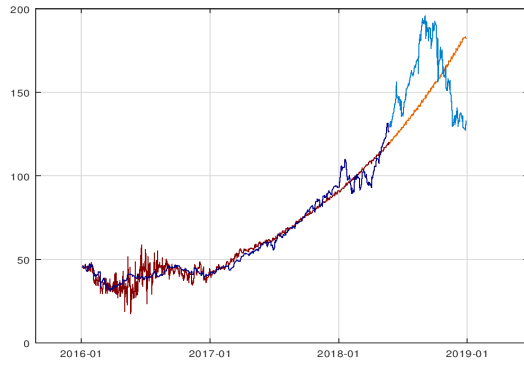


(27) Vorhersage des Aktienkurses von Thyssen Krupp



(28) Vorhersage des Aktienkurses von Volkswagen

— Geschätzter Schlusspreis (Trainingsdaten)   
 — Prognostizierter Schlusspreis (Testdaten)   
 — Schlusspreis (Testdaten)   
 — Schlusspreis (Trainingsdaten)



(29) Vorhersage des Aktienkurses von Wirecard

— Geschätzter Schlusspreis (Trainingsdaten) — Prognostizierter Schlusspreis (Testdaten) — Schlusspreis (Testdaten) — Schlusspreis (Trainingsdaten)

### C.4.2 Tabelle der Abweichungen zum PrNN (Einzelkursvorhersage)

Unternehmen	200N			400 N			600 N			
	300	500	762	300	500	762	300	500	762	
Adidas	$E_{Train}$	6.54	8.29	9.61	0.001	4.31	7.12	1.74e-13	1.13	6.59
	$E_{Test}$	8.64	10.04	9.81	11.91	10.31	9.11	26.54	9.38	9.37
Airbus	$E_{Train}$	2.21	5.74	4.44	0.00081	3.06	3.18	7.535e-13	1.27	3.49
	$E_{Test}$	13.06	8.73	9.74	19.75	9.90	97.77	9.75	11.81	9.30
Alphabet	$E_{Train}$	36.58	35.12	30.23	0.00047	33.03	31.15	1.8e-7	7.98	25.21
	$E_{Test}$	39.97	48.55	71.80	63.73	69.49	70.42	65.85	43.52	70.82
Audi	$E_{Train}$	15.68	33.830	34.33	0.00091	17.00	34.71	0.000000089	10.51	31.66
	$E_{Test}$	25.79	32.03	36.94	57.78	36.53	38.19	279.63	40.04	36.11
BASF	$E_{Train}$	2.41	3.17	4.07	0.002	1.70	3.59	1.3e-10	0.75	3.57
	$E_{Test}$	3.89	2.91	4.07	8.58	2.76	3.73	10.38	2.55	3.29
Bayer	$E_{Train}$	2.27	9.55	4.68	0.17	3.52	9.33	2.7e-13	2.76	3.53
	$E_{Test}$	3.24	6.87	5.21	8.98	8.30	12.24	18.27	11.53	5.79
Beiersdorf	$E_{Train}$	1.05	3.68	4.00	0.002	0.91	2.96	2.07e-14	0.78	4.01
	$E_{Test}$	4.22	3.26	3.99	6.83	4.10	2.99	3.54	3.55	4.52
Bertrandt	$E_{Train}$	2.02	17.19	6.99	0.00043	2.57	7.91	1.12e-10	1.79	7.75
	$E_{Test}$	3.80	3.88	8.69	5.11	13.38	6.35	10.07	5.00	6.27
BMW	$E_{Train}$	1.75	2.11	5.53	0.003	1.34	9.33	1.13e-7	0.98	3.53
	$E_{Test}$	1.44	2.14	4.50	4.64	2.03	3.90	42.98	2.59	3.57
CocaCola	$E_{Train}$	0.45	1.75	1.19	0.02	0.49	2.03	1.12e-9	0.18	1.50
	$E_{Test}$	1.23	1.25	2.99	5.64	1.14	1.85	5.18	2.46	1.94
Comdirect	$E_{Train}$	0.21	1.31	0.74	1.2e-7	0.22	0.59	4.06e-14	0.26	0.49
	$E_{Test}$	3.80	3.88	10.2	5.11	13.38	0.68	10.07	5.00	0.94
Continental	$E_{Train}$	06.37	25.62	20.86	0.14	3.58	23.68	1.5e-7	2.31	19.81
	$E_{Test}$	8.86	20.14	17.49	35.45	15.62	27.06	111.25	16.08	30.52
Daimler	$E_{Train}$	0.95	10.99	6.59	0.03	1.07	3.91	2.1e-11	1.01	4.75
	$E_{Test}$	1.56	2.15	5.48	3.75	1.60	1.78	4.78	1.72	2.16
Deutsche Telekom	$E_{Train}$	0.18	0.78	0.86	4.8e-5	0.29	0.72	8.7e-13	0.12	1.28
	$E_{Test}$	0.36	0.32	0.77	0.68	0.51	0.85	0.77	0.81	1.18
Eon	$E_{Train}$	0.19	0.43	0.82	0.003	0.15	0.69	1.1e-11	0.13	0.41
	$E_{Test}$	0.41	0.33	0.46	1.15	0.41	0.80	1.50	0.25	0.45
Evotec	$E_{Train}$	1.58	2.86	2.20	0.02	0.80	2.15	6.29e-12	0.41	2.24
	$E_{Test}$	1.63	2.34	2.20	10.27	2.54	2.56	9.88	2.35	4.28

Unternehmen	200N			400 N			600 N			
	300	500	762	300	500	762	300	500	762	
Exxon Mobil	$E_{Train}$	0.74	3.46	6.03	0.01	0.86	4.62	2.1e-11	0.53	3.53
	$E_{Test}$	4.75	3.05	2.11	4.40	3.00	2.56	5.98	3.58	4.28
FedEx	$E_{Train}$	2.58	8.39	12.10	0.01	5.32	8.11	1.8e-10	3.39	7.44
	$E_{Test}$	20.63	16.56	18.48	27.86	16.68	16.12	29.78	17.78	15.74
Fielmann	$E_{Train}$	2.31	2.49	5.01	0.02	1.81	5.76	14.04e-13	1.56	5.39
	$E_{Test}$	3.37	4.09	4.87	4.15	3.60	4.69	3.50	4.12	3.76
Infineon Technologies	$E_{Train}$	0.21	0.92	3.65	0.0001	0.84	1.89	3.8e-9	0.34	0.71
	$E_{Test}$	1.15	0.98	2.45	1.45	1.10	2.33	2.20	1.95	1.33
Lindt & Sprüngli	$E_{Train}$	126.69	137.48	379.53	0.006	93.34	300.34	0.00001	30.89	297.62
	$E_{Test}$	268.32	293.79	230.53	476.31	307.99	286.52	1000.6	231.00	274.82
Lufthansa	$E_{Train}$	0.50	1.84	5.50	0.001	1.43	5.44	4.1e-11	0.37	3.41
	$E_{Test}$	0.71	1.23	1.98	2.09	1.92	2.73	3.01	2.02	1.53
Microsoft	$E_{Train}$	2.67	2.02	3.06	0.008	0.53	6.922	1.8e-6	1.40	2.30
	$E_{Test}$	6.81	5.05	4.19	13.00	4.60	4.30	15.50	5.02	4.70
Nike	$E_{Train}$	4.81	1.60	5.41	0.0004	1.46	7.28	5.4e-9	0.73	2.62
	$E_{Test}$	8.68	5.83	4.38	11.70	5.54	6.07	12.45	5.45	4.72
RWE	$E_{Train}$	0.51	1.39	1.95	0.03	1.20	2.20	7.4e-12	0.54	2.40
	$E_{Test}$	3.40	1.62	1.85	3.47	1.52	1.82	1.89	2.18	1.93
SAP	$E_{Train}$	9.69	6.29	3.87	0.00008	1.54	3.70	3.4e-6	0.46	2.70
	$E_{Test}$	9.34	10.14	6.19	15.34	8.39	6.47	16.07	8.71	6.19
Siemens	$E_{Train}$	1.24	4.66	6.05	6.6e-6	3.36	4.67	5.1e-8	1.78	4.86
	$E_{Test}$	4.48	3.77	5.50	5.34	4.79	6.92	7.74	5.40	6.00
Thyssen Krupp	$E_{Train}$	0.58	1.54	0.97	0.0003	0.84	0.90	5.26e-12	0.33	0.83
	$E_{Test}$	1.18	1.04	1.20	3.83	1.26	1.21	3.20	2.45	1.32
Volkswagen	$E_{Train}$	2.43	7.09	13.57	0.0001	4.78	59.26	9.1e-10	2.75	19.35
	$E_{Test}$	6.73	9.69	8.24	6.44	9.48	8.81	14.88	12.16	7.04
Wirecard	$E_{Train}$	2.88	6.38	4.64	0.0017	9.07	4.23	2.5e-8	4.25	3.38
	$E_{Test}$	81.59	53.18	29.52	88.40	32.71	30.02	85.68	34.8	29.72

Tabelle C.13: RMSE zur Einzelkursvorhersage





## C.4.3 Tabelle der Abweichungen zum PrNN (Multiple Vorhersage)

Unternehmen	200N			400 N			600 N			
	300	500	762	300	500	762	300	500	762	
Adidas	$E_{Train}$	0.45	2.82	3.85	5.51e-11	2.68e-10	2.02	4.59e-12	5.23e-9	2.5e-7
	$E_{Test}$	37.21	548.14	301.70	59.71	121.91	453.14	24.63	57.40	283.60
Airbus	$E_{Train}$	0.23	1.32	1.53	2.39e-12	8.53e-11	1.21	2.23e-13	2.21e-9	1.6e-7
	$E_{Test}$	13.06	29.48	65.64	37.93	31.83	172.56	17.87	22.83	136.07
Alphabet	$E_{Train}$	1.16	12.02	15.09	4.02e-11	7.01e-12	13.30	2.3e-13	2.31e-8	2.09e-6
	$E_{Test}$	120.19	706.44	1126.07	211.44	366.70	2206.85	165.46	133.47	1522.93
Audi	$E_{Train}$	1.25	6.64	8.02	1.85e-11	9.24e-11	5.59	1.78e-12	2.83e-11	2.9e-7
	$E_{Test}$	192.60	707.81	208.04	103.04	574.32	577.68	109.15	137.06	1359.11
BASF	$E_{Train}$	0.12	0.85	1.10	3.41e-12	9.83e-10	0.73	2.28e-13	2.88e-9	1.41e-7
	$E_{Test}$	7.63	207.14	22.12	4.75	35.90	58.92	10.38	30.85	100.48
Bayer	$E_{Train}$	0.14	0.96	1.49	2.63e-12	1.08e-11	0.93	2.21e-12	3.42e-9	1.81e-8
	$E_{Test}$	10.64	175.98	16.12	12.59	60.62	93.07	14.82	26.22	168.09
Beiersdorf	$E_{Train}$	0.07	0.86	0.92	3.70e-12	1.07e-10	0.74	2.42e-13	2.78e-9	1.51e-8
	$E_{Test}$	8.08	43.06	12.59	9.98	41.49	104.15	5.2	16.33	152.07
Bertrandt	$E_{Train}$	0.22	2.37	2.08	2.81e-13	1.09e-10	1.19	2.1e-13	2.98e-9	2.01e-8
	$E_{Test}$	23.11	66.28	70.00	29.76	83.80	293.28	10.75	29.71	534.09
BMW	$E_{Train}$	0.09	0.79	1.21	2.01e-12	1.02e-12	0.64	2.01e-13	2.99e-9	2.31e-8
	$E_{Test}$	15.54	38.49	163.34	32.37	50.85	138.64	7.14	23.89	178.79
CocaCola	$E_{Train}$	0.02	0.25	0.39	1.24e-12	4.04e-11	0.23	8.71e-14	1.11e-8	7.71e-9
	$E_{Test}$	7.05	69.82	8.22	5.04	22.10	2.88	2.89	14.32	68.54
Comdirect	$E_{Train}$	0.01	0.12	0.13	3.82e-12	1.24e-11	0.10	2.71e-13	3.22e-10	2.11e-8
	$E_{Test}$	1.31	36.02	1.24	1.51	3.15	33.13	1.62	1.88	28.83
Continental	$E_{Train}$	0.55	2.45	2.96	9.29e-11	2.74e-11	1.83	5.21e-13	6.67e-9	3.91e-11
	$E_{Test}$	33.02	290.77	164.07	38.65	92.75	330.76	29.21	60.09	507.64
Daimler	$E_{Train}$	0.07	0.69	0.82	2.10e-11	7.6e-10	0.51	1.61e-13	2.31e-9	2.11e-8
	$E_{Test}$	11.26	325.23	31.07	14.69	38.41	84.12	4.78	28.62	162.08
Deutsche Telekom	$E_{Train}$	0.01	0.14	0.21	8.77e-13	1.55e-13	0.12	3.31e-14	5.11e-10	3.41e-9
	$E_{Test}$	3.03	47.20	25.93	2.55	9.88	22.67	2.14	5.92	20.24
Eon	$E_{Train}$	0.01	0.11	0.17	2.71e-13	7.95e-11	0.09	2.11e-14	2.44e-10	2.71e-9
	$E_{Test}$	0.78	40.76	9.08	0.98	4.49	5.03	0.61	4.35	11.62
Evotec	$E_{Train}$	0.07	0.37	0.57	4.57e-13	2.63e-12	0.37	3.71e-14	42.11e-10	3.22e-13
	$E_{Test}$	4.06	119.38	48.41	8.76	8.79	53.36	2.74	11.99	96.5

Unternehmen	200N			400 N			600 N			
	300	500	762	300	500	762	300	500	762	
	$E_{Train}$	0.11	0.56	1.04	1.75e - 12	1.98e-11	0.58	1.55e - 14	2.33e-9	1.34e-8
$E_{Test}$	8.27	113.14	30.91	6.87	26.44	98.35	4.07	17.82	263.41	
FedEx	$E_{Train}$	0.24	2.80	3.64	5.63e - 11	7.9e-11	2.21	4.77e - 13	6.12e-9	3.94e-8
	$E_{Test}$	28.61	193.61	49.91	57.64	114.30	339.75	23.28	38.08	368.93
Fielmann	$E_{Train}$	0.17	1.02	0.82	1.82e - 12	8.21e-10	0.97	1.61e - 13	2.12e-9	1.22e-8
	$E_{Test}$	18.21	309.57	52.89	33.44	65.04	89.84	8.51	34.05	158.93
Infineon Technologies	$E_{Train}$	0.04	0.35	0.43	5.73e - 13	2.33e-11	0.39	5.54e - 14	5.88e-10	3.12e-9
	$E_{Test}$	2.48	175.10	48.98	6.29	7.50	131.08	3.53	6.77	37.27
Lindt & Sprüngli	$E_{Train}$	9.63	42.22	51.90	1.26e - 10	6.77e-11	36.10	1.22e - 12	1.55e-8	8.54e-7
	$E_{Test}$	557.82	410.22	629.86	619.11	1534.21	1801.41	365.72	647.35	10829.99
Lufthansa	$E_{Train}$	0.08	0.36	0.41	6.26e - 12	1.54e-11	0.29	6.21e - 12	6.42e-0	4.54e-9
	$E_{Test}$	6.16	130.16	45.58	9.72	14.56	18.17	2.52	19.36	64.44
Microsoft	$E_{Train}$	0.07	0.71	0.94	2.72e - 12	6.78e-11	0.76	1.72e - 14	1.92e-8	1.11e-7
	$E_{Test}$	7.47	129.51	93.65	16.03	21.50	122.58	7.55	11.34	97.88
Nike	$E_{Train}$	0.10	0.65	0.76	1.68e - 12	4.41e-10	0.70	1.32e - 14	1.42e-8	1.43e-9
	$E_{Test}$	8.02	84.43	59.88	18.46	34.09	108.27	7.58	20.01	172.39
RWE	$E_{Train}$	0.03	0.31	0.52	5.29e - 13	3.33e-11	0.31	4.87e - 14	4.67e-10	5.12e-6
	$E_{Test}$	5.27	145.37	2.86	8.23	17.36	8.76	8.76	9.09	34.72
SAP	$E_{Train}$	0.12	1.03	1.06	4.23e - 11	7.99e-12	1.01	2.23e - 14	2.73e-9	1.62e-6
	$E_{Test}$	9.70	263.42	17.34	15.07	32.86	176.67	2.81	16.53	100.01
Siemens	$E_{Train}$	0.18	1.25	1.66	5.15e - 11	2.64e-12	1.25	2.94e - 14	3.71e-9	1.88e-7
	$E_{Test}$	8.29	481.18	50.10	6.03	55.52	136.02	12.99	24.00	266.46
Thyssen Krupp	$E_{Train}$	0.06	0.36	0.42	9.61e - 13	2.32e-11	0.25	6.16e - 13	7.91e-10	5.41e-8
	$E_{Test}$	4.82	88.47	30.72	4.45	17.31	7.11	12.67	5.71	67.09
Volkswagen	$E_{Train}$	0.25	2.04	2.95	4.41e - 12	7.62e-11	2.24	3.84e - 11	4.92e-8	3.02e-8
	$E_{Test}$	51.89	81.80	388.72	59.47	97.35	256.44	3.03	66.58	316.01
Wirecard	$E_{Train}$	0.40	2.23	1.93	3.01e - 12	4.94e-10	1.98	2.89e - 10	1.68e-8	9.31e-9
	$E_{Test}$	42.71	71.02	108.52	38.73	131.35	488.14	28.17	70.82	151.22

Tabelle C.14: RMSE zur Multi-Vorhersage



## C.4.4 Tabelle der Abweichungen zum PrNN (Einer aus allen)

Unternehmen	200N			400 N			600 N			
	300	500	762	300	500	762	300	500	762	
Adidas	$E_{Train}$	0.32	2.73	3.72	4.19e-12	1.24e-6	2.17	2.22e-11	6.18e-10	182.77
	$E_{Test}$	23.72	31.99	17.63	18.24	73.00	20.33	12.90	26.77	597.67
Airbus	$E_{Train}$	0.18	1.33	1.50	2.80e-11	8.05e-10	1.14	9.92e-13	1.04e-12	69.69
	$E_{Test}$	9.54	22.20	21.28	8.39	18.69	16.36	4.63	22.69	48.01
Alphabet	$E_{Train}$	1.45	14.21	26.67	8.71e-13	38.33	10.10	1.67e-13	8.46e-10	124.68
	$E_{Test}$	555.59	34.19	84.35	100.51	204.66	103.28	65.61	173.77	3.40e+18
Audi	$E_{Train}$	1.16	6.42	9.45	9.25	1.67	5.66	2.02	8.14e-9	657.79
	$E_{Test}$	149.27	84.05	118.20	51.62	550.40	74.05	41.70	216.20	594.31
BASF	$E_{Train}$	85.26	0.76	2.43	1.61e-12	1.05e-9	0.70	3.34e-14	6.41e-9	82.61
	$E_{Test}$	17.55	5.76	10.32	6.02	41.50	8.28	2.62	8.94	353.53
Bayer	$E_{Train}$	4.80	1.14	1.63	6.97	1.13	0.91	3.18e-14	0.086	89.74
	$E_{Test}$	14.50	19.06	9.93	7.17	23.39	13.65	6.45	16.68	8.92e+13
Beiersdorf	$E_{Train}$	0.08	1.57	0.95	1.31e-11	10.68	0.59	1.17e-13	2.54e-14	87.57
	$E_{Test}$	8.83	8.39	4.68	5.22	20.81	11.92	4.61	8.75	124.30
Bertrandt	$E_{Train}$	0.27	1.39	3.14	93.47e-14	89.23	1.09	1.52e-12	4.45e-12	1.29
	$E_{Test}$	15.52	9.08	10.39	8.59	30.96	16.58	7.40	15.69	605.77
BMW	$E_{Train}$	84.55	0.75	1.28	6.70e-12	6.40e-9	0.58	1.83e-14	1.48e-12	84.171
	$E_{Test}$	13.74	8.76	24.45	4.26	68.09	20.34	5.86	16.54	3.59e+41
CocaCola	$E_{Train}$	0.05	0.26	0.39	2.76e-13	3.47e-3	0.21	5.71e-14	1.75e-9	4.99
	$E_{Test}$	4.02	2.01	6.75	1.67	8.94	8.96	1.72	2.17	69.81
Comdirect	$E_{Train}$	0.02	0.12	0.15	7.81e-10	0.10	0.08	6.42e-14	2.15e-10	10.41
	$E_{Test}$	1.32	2.32	1.06	0.76	1.88	1.02	0.50	1.53	4.84
Continental	$E_{Train}$	215.88	2.54	3.22	2.30e-13	4.88e-7	1.78	1.21e-11	5.01e-14	206.06
	$E_{Test}$	166.28	18.87	80.23	21.10	126.06	68.97	12.42	39.80	2.01e+41
Daimler	$E_{Train}$	0.09	0.67	2.42	1.01e-14	1.68e-8	0.49	2.51e-14	1.75e-14	63.83
	$E_{Test}$	21.76	9.58	5.81	6.42	408.01	13.80	4.51	10.28	516.27
Deutsche Telekom	$E_{Train}$	0.01	0.14	0.20	3.06e-14	0.28	0.17	1.82	4.58e-14	16.57
	$E_{Test}$	1.18	1.05	1.51	1.16	3.82	1.30	0.97	1.10	100.21
Eon	$E_{Train}$	0.01	0.10	0.21	7.41e-14	9.15	0.08	4.45e-15	7.04e-12	7.31
	$E_{Test}$	0.57	0.67	1.36	0.55	1.44	1.93	0.45	1.30	9.31
Evotec	$E_{Train}$	0.07	0.38	0.67	8.42e-14	12.95	0.37	1.60e-14	2.08e-13	10.81
	$E_{Test}$	4.59	4.31	2.35	2.15	6.26	5.38	2.70	7.84	6.36e+28

Unternehmen	200N			400 N			600 N			
	300	500	762	300	500	762	300	500	762	
Fielmann	$E_{Train}$	0.12	1.07	0.85	1.91e - 13	3.17	0.51	5.26e - 14	0.61e-11	0.037
	$E_{Test}$	10.61	3.30	8.46	7.06	28.36	9.08	6.80	5.38	424.06
Infineon Technologies	$E_{Train}$	0.04	0.36	0.44	2.69e - 14	9.13e-13	0.22	2.13e - 15	2.82e-9	19.93
	$E_{Test}$	4.18	2.16	1.89	1.47	8.62	2.96	1.25	2.83	554.09
Lindt & Sprüngli	$E_{Train}$	11.89	41.51	53.10	9.11e - 11	346.14	29.06	1.88e - 11	1.05e-9	5228.51
	$E_{Test}$	1003.51	602.25	409.95	250.10	9.96e + 7	991.19	265.13	971.44	1.24e+8
Lufthansa	$E_{Train}$	25.65	0.34	1.21	2.40e - 13	28.29	0.29	1.04e - 12	5.51e-7	23.21
	$E_{Test}$	3.49	1.69	6.28	2.02	9.93	4.00	1.80	4.30	18.69
Microsoft	$E_{Train}$	0.13	0.93	0.99	1.74e - 13	9.08	0.61	1.7e - 13	1.29e-11	65.03
	$E_{Test}$	13.19	6.87	4.59	5.81	133.30	12.92	4.17	7.84	4.03e+8
Nike	$E_{Train}$	0.13	0.63	1.05	1.67e - 13	1.51e-3	1.54	4.37e - 14	2.74e-8	68.74
	$E_{Test}$	9.81	13.30	8.38	8.48	17.80	6.35	4.07	9.48	128.24
RWE	$E_{Train}$	0.04	0.29	0.61	1.34e - 13	16.81	0.48	1.88e - 14	1.30e-11	8.88e-7
	$E_{Test}$	5.37	2.19	2.93	1.89	6.41	3.46	1.53	3.01	37.30
SAP	$E_{Train}$	0.14	1.13	1.05	7.05e - 14	1.59	0.74	2.74e - 14	1.17e-11	85.59
	$E_{Test}$	14.41	12.16	6.88	7.07	49.26	10.23	6.34	22.70	6.05e+32
Siemens	$E_{Train}$	0.20	1.29	1.97	1.04	0.03	0.96	2.89e - 13	6.57e-11	110.77
	$E_{Test}$	11.44	9.40	28.05	6.04	17.20	10.84	4.18	17.06	2.76e+10
Thyssen Krupp	$E_{Train}$	0.06	0.38	0.49	1.64e - 18	22.31	0.24	1.28e - 13	7.63e-12	21.84
	$E_{Test}$	3.92	12.86	2.40	2.20	6.90	5.05	1.96	4.91	7.00e+43
Volkswagen	$E_{Train}$	0.30	2.09	3.60	1.44e - 12	162.80	1.80	1.32e - 12	1.86e-7	118.02
	$E_{Test}$	49.86	11.25	21.45	12.55	237.92	49.31	11.54	35.07	2.38e+11
Wirecard	$E_{Train}$	0.41	2.24	2.02	1.42e - 10	6.77	1.54	1.27e - 12	2.38e-8	59.8
	$E_{Test}$	44.61	34.15	21.81	15.45	304.44	32.17	27.23	22.04	37.56

Tabelle C.15: RMSE zur Vorhersage einer aus allen



### C.4.5 Tabelle der Abweichungen zur Vorhersage mit den Fundamentaldaten

Tabelle C.16: RMSE der Vorhersagen mit den Fundamentaldaten

Unternehmen	Datensatz	Ergebnis aus: 300 Daten-200 Neuronen
Adidas	$E_{Train}$	31.09
	$E_{Test}$	149.82
Airbus	$E_{Train}$	42.74
	$E_{Test}$	810.18
Alphabet	$E_{Train}$	83.79
	$E_{Test}$	935.04
Audi	$E_{Train}$	27.52
	$E_{Test}$	382.51
BASF	$E_{Train}$	15.04
	$E_{Test}$	1008.01
Bayer	$E_{Train}$	37.49
	$E_{Test}$	14869.03
Beiersdorf	$E_{Train}$	1.01
	$E_{Test}$	17.85
Bertrandt	$E_{Train}$	51.53
	$E_{Test}$	117938
BMW	$E_{Train}$	2.83
	$E_{Test}$	296.28
CocaCola	$E_{Train}$	23.54
	$E_{Test}$	113.01
Comdirect	$E_{Train}$	1.90
	$E_{Test}$	18.96
Continental	$E_{Train}$	4.18
	$E_{Test}$	49.97
Daimler	$E_{Train}$	42.74
	$E_{Test}$	1359.21
Deutsche Telekom	$E_{Train}$	2.51
	$E_{Test}$	14.54
Eon	$E_{Train}$	0.62
	$E_{Test}$	30.72
Evotec	$E_{Train}$	6.55
	$E_{Test}$	215154.22
Exxon Mobil	$E_{Train}$	0.61
	$E_{Test}$	64.54
FedEx	$E_{Train}$	134.84
	$E_{Test}$	739.27
Fielmann	$E_{Train}$	15.48
	$E_{Test}$	696.27
Infineon Technologies	$E_{Train}$	67.34
	$E_{Test}$	11065.09
Lindt & Sprüngli	$E_{Train}$	74.75
	$E_{Test}$	1874.22

---

Unternehmen	Datensatz	Ergebnis aus: 300 Daten-200 Neuronen
Lufthansa	$E_{Train}$	0.88
	$E_{Test}$	37.42
Microsoft	$E_{Train}$	8.93
	$E_{Test}$	1047.7
Nike	$E_{Train}$	1.20
	$E_{Test}$	83.58
RWE	$E_{Train}$	38.94
	$E_{Test}$	343232.90
SAP	$E_{Train}$	5.96
	$E_{Test}$	189.16
Siemens	$E_{Train}$	5.76
	$E_{Test}$	143.99
Thyssen Krupp	$E_{Train}$	22.20
	$E_{Test}$	3575983.06





# Literaturverzeichnis

- [1] Ibm - aktienkurs, April 2019. (zuletzt aufgerufen am 02.04.2019). URL: <https://www.ariva.de/ibm-aktie>. (zitiert auf Seite vii und 10)
- [2] Herbert Jaeger. Echo state network. *Scholarpedia*, 2(9):2330, 2007. revision #188245. doi:10.4249/scholarpedia.2330. (zitiert auf Seite vii und 30)
- [3] Rainer Elschen. *Der Werdegang der Krise - Von der Subprime- zur Systemkrise*. Wiesbaden: Gabler- GWV Fachverlag GmbH, 2009. (zitiert auf Seite 1)
- [4] Wenjuan Mei, Pan Xu, Ruochen Liu, and Jun Liu. Stock price prediction based on arima-svm model. *International Conference on Big Data and Artificial Intelligence*, 2018. (zitiert auf Seite 4)
- [5] Jonathan Rebane, Stojan Dennic, Isak Karlsson, and Panagiotis Papapetrou. Seq2seq rnns and arima models for cryptocurrency prediction: A comparative study. *SIGKDD Fintech Conference*, 2018. doi:[https://doi.org/10.475/123\\_4](https://doi.org/10.475/123_4). (zitiert auf Seite 4)
- [6] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654 – 669, 2018. URL: <http://www.sciencedirect.com/science/article/pii/S0377221717310652>, doi:<https://doi.org/10.1016/j.ejor.2017.11.054>. (zitiert auf Seite 4)
- [7] Frieder Stolzenburg, Olivia Michael, and Oliver Obst. Predictive neural networks. *CoRR*, abs/1802.03308, 2018. URL: <http://arxiv.org/abs/1802.03308>. (zitiert auf Seite 4, 32, 34, 35, 36 und 50)
- [8] Viktor Hesse and Christian Riedel. *Fundamentalanalyse versus Chartanalyse - Methoden der Aktienbewertung im Vergleich*. Wiesbaden: Gabler Verlag, 2016. (zitiert auf Seite 8)
- [9] Viktor Heese. *Aktienbewertung mit Kennzahlen - Kurschancen und -risiken fundiert beurteilen*. Wiesbaden: Gabler Verlag, 2011. (zitiert auf Seite 8 und 11)

- [10] Klaus Spremann and Pascal Gantenbein. *Finanzmärkte - Grundlagen, Instrumente, Zusammenhänge*. München: UVK Verlagsgesellschaft, 4. Auflage, 2017. (zitiert auf Seite 8)
- [11] o.V. Zentrale gröÙe der geschätzstätigkeit, 2013. (zuletzt aufgerufen am 27.04.2019). URL: [http://www.business-on.de/definition-gewinn-zentrale-groesse-der-geschaefsttaetigkeit-\\_id44310.html](http://www.business-on.de/definition-gewinn-zentrale-groesse-der-geschaefsttaetigkeit-_id44310.html). (zitiert auf Seite 9)
- [12] Thomas Schuster and Margarita Uskova. *Finanzierung: Anleihen, Aktien, Optionen*. Wiesbaden: Springer-Gabler Verlag, 2015. (zitiert auf Seite 9, 10 und 11)
- [13] Philipp Hornung Gernot Hinterleitner. *Marktqualität und Insiderhandel auf Kapitalmärkten - Eine experimentelle Untersuchung unterschiedlicher Handelsformen*. Wiesbaden: Gabler Verlag, 2016. doi:10.1007/978-3-658-12850-0. (zitiert auf Seite 9)
- [14] Enzo Mondello. *Finance: Angewandte Grundlagen*. Wiesbaden: Gabler Verlag, 2018. (zitiert auf Seite 10 und 11)
- [15] Enzo Mondello. *Aktienbewertung - Theorie und Anwendungsbeispiele*. Wiesbaden: Springer Verlag, 2. Auflage, 2017. (zitiert auf Seite 11 und 12)
- [16] Klaus Backhaus, Bernd Erichson, Wulff Plinke, and Rolf Weiber. *Multivariate Analysemethoden - Eine anwendungsorientierte Einführung*. Wiesbaden: Gabler Verlag, 14. Auflage, 2016. doi:10.1007/978-3-662-46076-4. (zitiert auf Seite 12, 16, 17 und 27)
- [17] Manfred Deistler and Wolfgang Scherrer. *Modelle der Zeitreihenanalyse*. Basel: Birkhäuser Basel, 2018. (zitiert auf Seite 12 und 13)
- [18] Christof Wolf and Henning Best, editors. *Handbuch der sozialwissenschaftlichen Datenanalyse*. Wiesbaden: VS Verlag für Sozialwissenschaften, Wiesbaden, 2010. (zitiert auf Seite 13, 14 und 15)
- [19] Jürgen Vogel. *Prognose von Zeitreihen - Eine Einführung für Wirtschaftswissenschaftler*. Wiesbaden: Springer Gabler Verlag, 2015. doi:10.1007/978-3-658-06837-0. (zitiert auf Seite 13, 15, 16, 20, 23 und 24)
- [20] Chris Chatfield. *TIME-SERIES FORECASTING*. London: Chapman&Hall/CRC, 2001. (zitiert auf Seite 13, 21 und 22)
- [21] Peter Mertens and Susanne Rässler. *Prognoserechnung*. Heidelberg: Physica-Verlag, 4. Auflage, 2012. (zitiert auf Seite 14)

- 
- [22] Bankhofer Udo and Vogel Jürgen. *Datenanalyse und Statistik - Eine Einführung für Ökonomen im Bachelor*. Wiesbaden: Gabler Verlag, 2008. (zitiert auf Seite 17)
- [23] John Komlos and Bernd Süßmuth. *Empirische Ökonomie - Eine Einführung in Methoden und Anwendungen*. Wiesbaden: Springer-Verlag, 2010. doi:10.1007/978-3-642-01705-6. (zitiert auf Seite 17)
- [24] Robert H. Shumway and Stoffer David S. *Time Series Analysis and Its Applications - With R Examples*. Springer Science + Business Media, 3. Auflage, 2011. (zitiert auf Seite 19 und 21)
- [25] Denis Kwiatkowski, Peter C. B. Phillips, Peter Schmidt, and Yongcheol Shin. Testing for the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54:1, 1992. doi:10.1016/0304-4076(92)90104-y. (zitiert auf Seite 23)
- [26] Said E. Said and David A. Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 12 1984. doi:10.1093/biomet/71.3.599. (zitiert auf Seite 23)
- [27] Pierre Perron Philipps, Peter C. B. Testing for a unit root in time series regression. *Biometrika*, 75(2):335–346, 06 1988. doi:10.1093/biomet/75.2.335. (zitiert auf Seite 23)
- [28] Christopher Mayer zu Reckendorf Henning Beck, Sofia Anastasiadou. *Faszinierendes Gehirn - Eine bebilderte Reise in die Welt der Nervenzelle*. Berlin, Heidelberg: Springer Spektrum, 2018. doi:10.1007/978-3-662-47092-3. (zitiert auf Seite 25)
- [29] Jörn Schmidt, Christina Klüver, and Jürgen Klüver. *Programmierung naturanaloger Verfahren - Soft Computing und verwandte Methoden*. Wiesbaden: Vieweg + Teubner Verlag, 2010. doi:10.1007/978-3-8348-9666-7. (zitiert auf Seite 26 und 27)
- [30] Ertel Wolfgang. *Grundkurs Künstliche Intelligenz - Eine praxisorientierte Einführung*. Wiesbaden: Springer Fachmedien, 2016. (zitiert auf Seite 27)
- [31] Christina Klüver, Jürgen Klüver, and Jörn Schmidt. *Modellierung komplexer Prozesse durch naturanaloge Verfahren - Soft Computing und verwandte Techniken*. Wiesbaden: Springer Vieweg, 2. Auflage, 01 2012. (zitiert auf Seite 28 und 29)
- [32] Oliver Kramer. *Computational Intelligence - Eine Einführung*. Berlin - Heidelberg: Springer-Verlag, 2009. (zitiert auf Seite 28)

- [33] Rudolf Kruse, Christian Borgelt, Christian Braune, Frank Klawonn, Christian Moewes, and Matthias Steinbrecher. *Computational Intelligence - Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Wiesbaden: Springer Fachmedien, 2. Auflage, 2015. (zitiert auf Seite 28)
- [34] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. URL: <https://www.pnas.org/content/79/8/2554>, arXiv:<https://www.pnas.org/content/79/8/2554.full.pdf>, doi: 10.1073/pnas.79.8.2554. (zitiert auf Seite 29)
- [35] Jeffrey L. Elman. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211, 1990. URL: <https://pdfs.semanticscholar.org/b71e/c700edfec2b969c9d27d33eac09188290294.pdf>. (zitiert auf Seite 30)
- [36] D. M. Smith W.N.Venables. *An introduction to R - Notes on R: A Programming Environment for Data Analysis and Graphics (Version 3.6.0 - 26.4.2016)*, 2019. URL: <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>. (zitiert auf Seite 39)
- [37] Rob J. Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(3):1–22, 3 2008. doi:10.18637/jss.v027.i03. (zitiert auf Seite 42, 44, 47, 48 und 49)
- [38] Gregory C. Reinsel George E. P. Box, Gwilym M. Jenkins. *TIME SERIES ANALYSIS - Forecasting and Control*. New Jersey: Prentice-Hall, 3. Auflage, 1994. (zitiert auf Seite 42)
- [39] R Bender and St Lange. Was ist der p-wert? *Deutsche Medizinische Wochenschrift - DEUT MED WOCHENSCHR*, 126:39–40, 01 2001. doi:10.1055/s-2001-12739. (zitiert auf Seite 43)
- [40] Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. New York: Springer Verlag, New York, NY, 1973. (zitiert auf Seite 44)
- [41] Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6:3–73, 1990. (zitiert auf Seite 48)
- [42] Fabio Canova and Bruce E. Hansen. Are seasonal patterns constant over time? a test for seasonal stability. *Journal of Business & Economic Statistics*, 13(3):237–252, 1995. URL: <https://www.tandfonline.com/doi/abs/10.1080/07350015.1995.10524598>, arXiv:<https://www.tandfonline.com/doi/abs/10.1080/07350015.1995.10524598>

---

`//www.tandfonline.com/doi/pdf/10.1080/07350015.1995.10524598`,  
doi:10.1080/07350015.1995.10524598. (zitiert auf Seite 48)

- [43] *GNU Octave (version 4.4.1) - Manual*, 2018. URL: <https://octave.org/doc/v4.4.1/index.html>. (zitiert auf Seite 50)



# **Eigenständigkeitserklärung**

Hiermit versichere ich, dass ich die vorliegende Arbeit bisher bei keiner anderen Prüfungsbehörde eingereicht, sie selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Wernigerode, 29.08.2019



Die Vorhersage der zukünftigen Entwicklung von Aktienkursen ist eines der gefragtesten, gleichzeitig aber auch komplexesten und schwierigsten Themen des Aktienmarktes. Zuverlässige Prognosen sollen Investoren und Unternehmen dabei unterstützen, den Verlauf von Aktienkursen vorherzusehen, um z. B. das Risiko in falsche Aktien zu investieren zu minimieren. Viele unterschiedliche Methoden und Techniken, wie z. B. Regressionsanalysen oder Neuronale Netze, werden zunehmend eingesetzt, um die Aktienkursentwicklung vorhersagen zu können.