

Selbination - Ein hybrides Meta-Lernverfahren
zur automatischen Selektion und Kombination
geeigneter Prognosemodelle für die Produktionsplanung

Vom Fachbereich Produktionstechnik
der
UNIVERSITÄT BREMEN

zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte

Dissertation
von
Dipl.-Math. Mirko Sebastian Kück

Gutachter:

Prof. Dr.-Ing. Michael Freitag, Universität Bremen

Prof. Dr.-Ing. Bernd Hellingrath, Westfälische Wilhelms-Universität Münster

Tag der mündlichen Prüfung: 11.10.2022

Danksagung

Diese Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am BIBA - Bremer Institut für Produktion und Logistik an der Universität Bremen. Ich möchte vielen Personen danken, die mich auf dem steinigen Weg zur Fertigstellung der Dissertation unterstützt haben.

Mein besonderer Dank gilt Prof. Dr.-Ing. Michael Freitag für die Betreuung meines Promotionsverfahrens und die durchgängige Förderung über die letzten Jahre. Viele kritische Diskussionen und sein kontinuierliches Mentoring prägten nachhaltig meinen bisherigen beruflichen Werdegang. Zudem danke ich Prof. Dr.-Ing. Bernd Hellingrath für die Übernahme der Zweitgutachterschaft. Prof. Dr.-Ing. Bernd Scholz-Reiter danke ich für das entgegengebrachte Vertrauen zum Antritt der Stelle eines wissenschaftlichen Mitarbeiters am BIBA.

Ich danke Prof. Dr.-Ing. Enzo Morosini Frazzon für die Gastfreundlichkeit, mit der er mich in Brasilien aufnahm und mir die Möglichkeit bot, mich fachlich und persönlich weiterzuentwickeln. Zudem bin ich dankbar dafür, in ihm einen guten Freund gewonnen zu haben. Des Weiteren möchte ich den vielen inspirierenden Wissenschaftlern danken, denen ich im Laufe der Jahre begegnet bin. Hervorheben möchte ich Prof. Dr. Sven Crone, dem ich für die produktive Zusammenarbeit und sein kritisches Feedback zu meinen Forschungsansätzen danken möchte. Ich danke meinen Kollegen am BIBA, die sich stets die Zeit genommen haben, innovative Forschungsideen zu diskutieren oder auch mit einer Debatte der letzten Fußballergebnisse für die notwendige Ablenkung zu sorgen. In diesem Zusammenhang möchte ich insbesondere Daniel Rippel, Hendrik Engbers und Benjamin Staar für ihre fachlichen Kommentare zu meinen Publikationen und meiner Dissertation danken. Marit Hoff-Hoffmeyer-Zlotnik und Dr.-Ing. Sebastian Grundstein danke ich für viele angenehme Gespräche während der Zeit im gemeinsamen Büro. Zudem danke ich Eike Broda und Jan Kebel, die mit ihren studentischen Arbeiten einen Beitrag zu meiner Dissertation leisteten.

Mein wichtigster Dank gilt meiner wundervollen Ehefrau Lili, die mich auf diesem langen Weg immer unterstützt hat, mir viel Verständnis für die arbeitsreichen Abende und Wochenenden entgegengebracht hat und mir Mut in den schwierigen Phasen der Arbeit zugesprochen hat. Ihre Liebe bildete das Fundament für die vorliegende Dissertation.

Zuletzt möchte ich einen speziellen Dank an meine liebevollen Eltern Hilde und Reinhold richten. In meinem gesamten Leben habe ich mich stets sicher gefühlt, denn ich wusste und weiß, dass sie mich in jeder möglichen Situation unterstützen würden. Meinem Bruder Malte danke ich dafür, dass er mein bester Freund ist und immer für mich da ist. Vielen Dank euch allen!

Zusammenfassung

Genauere Nachfrageprognosen sind von signifikanter Bedeutung für produzierende Unternehmen. Sie bilden die Basis der Produktionsplanung. Dies gilt insbesondere für Unternehmen, die Produkte auf Lager produzieren, wie beispielsweise häufig in der Konsumgüterindustrie. Solche Unternehmen müssen die Kundennachfrage für eine große Produktpalette in regelmäßigen Intervallen basierend auf Zeitreihen früherer Kundenbestellungen prognostizieren, weshalb automatische Prognoseverfahren notwendig sind. Nach dem No Free Lunch Theorem existiert kein Prognoseverfahren, welches für alle Datensätze besser prognostiziert als alle anderen Verfahren. Daher sollte entweder ein individuelles Prognosemodell pro Zeitreihe selektiert werden oder es sollten die Prognosen mehrerer Modelle kombiniert werden. Selektionsansätze bieten das Potenzial, das beste Prognosemodell für eine Zeitreihe zu selektieren, wenn ein dominantes Modell für diese existiert. Allerdings birgt eine Selektion das Risiko, ein schlechtes Modell zu selektieren, falls entweder das Selektionskriterium unpassend ist oder für mehrere Modelle eine ähnliche Prognosegüte zu erwarten ist. In diesem Fall haben Kombinationsansätze den Vorteil, dass sich die Stärken und Schwächen der verschiedenen Modelle zu einer robusteren Prognose ausgleichen. Alle bisher entwickelten Verfahren lassen sich entweder als Selektionsansätze für ein Prognosemodell oder als Kombinationsansätze für mehrere Modelle klassifizieren und es existieren bislang keine hybriden Verfahren zwischen Selektion und Kombination. Zur Nutzung der jeweiligen Vorteile der beiden Ansätze wird in der vorliegenden Arbeit ein hybrides Verfahren vorgestellt, das für jede Zeitreihe datengetrieben entweder selektiert oder kombiniert. Das sogenannte Selbinationsverfahren besteht aus einem Meta-Modell zur Schätzung der Eignungswahrscheinlichkeiten verschiedener Prognoseverfahrenskandidaten und einer Selektionsfunktion, die auf Basis der Eignungswahrscheinlichkeiten sowie einer sogenannten Hybriditätsgrenze ein oder mehrere Prognoseverfahren selektiert. Das neuartige Verfahren wird in einer umfassenden empirischen Studie anhand eines gängigen industriellen Datensatzes evaluiert und erzielt im Durchschnitt signifikant bessere Prognosen als gängige automatische Selektions-, Kombinations- und Pooling-Verfahren aus dem Stand der Forschung. Dabei erreicht das Selbinationsverfahren insbesondere eine höhere Prognosegüte als alle Originalteilnehmer und Benchmarks der NN3-Competition, einem wichtigen vergangenen Prognosewettbewerb, dessen Datensatz häufig zur objektiven Evaluation neuer Prognoseverfahren verwendet wird. In einer Vergleichsstudie unter den Originalbedingungen zeigt sich, dass das Selbinationsverfahren den Wettbewerb gewonnen hätte. In einer darauffol-

genden Lagerbestandssimulation erzielt das Selbinationsverfahren den besten Kompromiss aus erreichtem Servicegrad und mittlerem benötigten Sicherheitsbestand. Insgesamt bietet das neuartige Selbinationsverfahren dieser Arbeit großes Potenzial für produzierende Unternehmen zur Prognose der Kundennachfrage. Aufgrund der automatischen Funktionsweise besteht zusätzlich die Möglichkeit, das Verfahren in zukünftigen Studien auf andere industrielle Forschungsgebiete zu übertragen, wie beispielsweise für eine vorausschauende Instandhaltungsplanung oder für Produktqualitätsprognosen.

Abstract

Accurate demand forecasting is highly important for manufacturing companies since it provides a basis for production planning and inventory management. In particular, this holds for companies that manufacture to stock, which is often observed in the fast moving consumer goods industry. Such companies have to forecast customer demands regularly for a large number of products based on time series of past customer demands. Therefore, automatic forecasting methods have to be applied in order to avoid a manual fitting of forecasting models for each time series. Following the no free lunch theorem, there is no individual forecasting method that outperforms all others for all data conditions. Hence, a selection of the same aggregate model for all time series should be avoided. Instead, either an individual model per time series should be selected, or the forecasts of multiple models should be combined. Individual selection approaches offer potential if one dominant forecasting model can be expected for a time series. However, a selection of one model involves the risk to select the wrong model if either the selection criterion is inappropriate, or many models with a similar performance can be expected. In such cases, a combination of multiple models offers the advantage that the strengths and weaknesses of multiple methods compensate one another in a more robust forecast. In this context, all developed approaches in the state of the art can be classified as either a selection or a combination approach, while no hybrid approaches have been developed so far. To utilize the advantages of both approaches and to overcome the manual decision for either selection or combination, this thesis proposes a hybrid approach that selects a single or multiple methods depending on time series characteristics. The so-called selbination method consists of a meta-model to estimate method suitability probabilities and a selection function to select one or multiple forecasting methods based on the suitability probabilities and a so-called hybridity threshold. In a comprehensive empirical evaluation on an industrial dataset, the approach achieves the highest forecast accuracy compared to common benchmark methods comprising selection, combination and pooling methods. In particular, the selbination method obtains a higher forecast accuracy than all original participants and benchmarks of the NN3-Competition, an important past forecasting competition, whose data set is often used to evaluate new forecasting methods. In an empirical study subject to the original conditions of the competition, this thesis shows that the selbination method would have won the NN3-Competition. In addition to the evaluations regarding forecast accuracy, the performance of the new method is also evaluated in terms of an inventory simulation. Here, it obtains the best

trade-off between achieved service level and needed safety stock. All in all, the new selbination method offers great potential for manufacturing companies to forecast future customer demands. Because of the automatic functionality of the selbination method, it provides the opportunity to be transferred to other industrial applications, such as predictive maintenance or product quality predictions in future studies.

Inhaltsverzeichnis

Zusammenfassung	i
Abstract	iii
Inhaltsverzeichnis	v
Abbildungsverzeichnis	ix
Tabellenverzeichnis	xii
Abkürzungs- und Formelverzeichnis	xiv
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung und Forschungsthese	3
1.3 Wissenschaftlicher Beitrag der Arbeit	3
1.4 Aufbau der Arbeit	4
2 Grundlagen der Absatzplanung	6
2.1 Ziele produzierender Unternehmen	6
2.2 Bedeutung und Prozess der Absatzplanung	7
2.2.1 Übersicht	7
2.2.2 Bedeutung der Absatzplanung in verschiedenen Branchen	8
2.2.3 Prozess der Absatzplanung	10
2.3 Modellierung der Kundennachfrage	13
2.3.1 Grundlagen der Systemtheorie	13
2.3.2 Bedarfsplanung eines Kunden	13
2.3.3 Modellierung der Kundennachfrage in einem dynamischen System	16
2.4 Zusammenfassung	17
3 Grundlagen und Stand der Forschung zur Zeitreihenprognose	18
3.1 Überwachtes Lernen	18

3.1.1	Übersicht und Einordnung	18
3.1.2	Theoretische Grundlagen des überwachten Lernens	20
3.1.3	Klassifikation	25
3.1.4	Regression	27
3.2	Zeitreihenprognose	28
3.2.1	Zeitreihenprognoseprobleme	28
3.2.2	Evaluationskriterien für Zeitreihenprognosen für einzelne Zeitreihen . . .	29
3.2.3	Evaluationskriterien für Zeitreihenprognosen über mehrere Zeitreihen . .	34
3.3	Gängige Prognosemodellklassen und Benchmark-Datensätze	36
3.3.1	M3-Competition und NN3-Competition	36
3.3.2	Gängige Prognosemodellklassen	40
3.4	Verwendete Zeitreihenprognoseverfahren mit vorgegebener Modellklasse	41
3.4.1	Random-Walk-Modelle	41
3.4.2	ARIMA Modelle	44
3.4.3	Exponentielle Glättungsmodelle	46
3.4.4	Lokale k -nächste-Nachbarn-Modelle	49
3.4.5	Neuronale Netze	62
3.5	Zusammenfassung	69
4	Stand der Forschung und Taxonomie zur Selektion geeigneter Zeitreihenprognosemodelle	70
4.1	Prognosemodellselektionsproblem	70
4.2	Theoretische Grundlagen zur Selektion von Zeitreihenprognosemodellen	71
4.2.1	Übersicht	71
4.2.2	Ansätze zur Selektion von Prognosemodellen	73
4.2.3	Ansätze zur Kombination von Prognosemodellen	74
4.3	Stand der Forschung zur Selektion geeigneter Zeitreihenprognosemodelle	75
4.3.1	Übersicht	75
4.3.2	Vergleiche von Prognosefehlern	77
4.3.3	Regelbasierte Expertensysteme	80
4.3.4	Meta-Lernansätze	82
4.4	Taxonomie verschiedener Ansätze zur Selektion von Prognosemodellen	87
4.5	Limitierungen und Best Practices bestehender Modellselektionsansätze	90
5	Beschreibung und Konfiguration des Selbbinationsverfahrens	92
5.1	Übersicht	92
5.2	Training des Meta-Modells	95
5.2.1	Übersicht	95
5.2.2	Meta-Merkmale	97

5.2.3	Meta-Labels	104
5.2.4	Meta-Modell	106
5.2.5	Merkmalsselektion	110
5.3	Einsatz des Selbinationsverfahrens	112
5.3.1	Übersicht	112
5.3.2	Eignungswahrscheinlichkeiten von Prognoseverfahren	114
5.3.3	Hybriditätsgrenze	114
5.3.4	Einsatz von Prognoseverfahren und -modellen	117
5.4	Konfiguration der Hybriditätsgrenze	117
5.4.1	Experimentbeschreibung	117
5.4.2	Evaluationsergebnisse	119
5.5	Implementierung des Selbinationsverfahrens	119
5.6	Zusammenfassung	121
6	Evaluation des Selbinationsverfahrens hinsichtlich Prognosegüte und Rechen-	
	aufwand	123
6.1	Experimentbeschreibung	123
6.1.1	Prognoseverfahren	123
6.1.2	Datensatz	125
6.1.3	Evaluationskriterien	126
6.2	Evaluation hinsichtlich der Prognosegüte	128
6.2.1	Evaluation rollierender Prognosen für verschiedene Ursprünge und Horizonte	128
6.2.2	Evaluation unter den Bedingungen der NN3-Competition	132
6.3	Evaluation hinsichtlich des Rechenaufwands	135
6.4	Evaluation hinsichtlich der Robustheit	137
6.4.1	Strategien zur Konfiguration der Parameter des Selbinationsverfahrens . .	137
6.4.2	Vergleich der Prognosegüten verschiedener Parameterkonfigurationen . . .	139
6.5	Zusammenfassung	141
7	Evaluation des Selbinationsverfahrens mittels Lagerbestandssimulation	143
7.1	Simulationsmodell und Lagerhaltungspolitik	143
7.1.1	Zustandsvariablen und Ablauf der Simulation	143
7.1.2	Implementierung des Simulationsmodells	148
7.2	Experimentbeschreibung	149
7.2.1	Untersuchungsdimensionen	149
7.2.2	Bestimmung der Initialwerte für die Simulation	149
7.2.3	Evaluationskriterien	150
7.3	Ergebnisse	151
7.4	Zusammenfassung	154

8	Schlussbetrachtung	156
8.1	Zusammenfassung der Hauptergebnisse	156
8.2	Beantwortung der Forschungsthese	160
8.3	Ausblick	162
	Literaturverzeichnis	165
A	Eigene Publikationen und Würdigung studentischer Arbeiten	193
B	Ergebnisse der M3- und der NN3-Competition	195
B.1	Ergebnisse der M3-Competition	195
B.2	Ergebnisse der NN3-Competition	197
C	Vergleich verschiedener Selektionsmethoden für lokale k-nächste-Nachbarn-Modelle	200
C.1	Vergleich verschiedener Regularisierungsmethoden	200
C.2	Vergleich statischer und dynamischer Modellselektionsmethoden	204
D	Selektionen verschiedener Parameterkonfigurationen des Selbinationsverfahrens	210
E	Evaluation des Selbinationsverfahrens hinsichtlich des MASE	212

Abbildungsverzeichnis

1.1	Aufbau der Arbeit in Anlehnung an den Cross-industry standard process for data mining	5
2.1	Zielsystem der Produktionslogistik in Anlehnung an Wiendahl (2014)	7
2.2	Bevorratungsstrategien in Anlehnung an Wiendahl (2014)	9
2.3	Exemplarische Supply Chain in der Konsumgüterindustrie in Anlehnung an Syntetos et al. (2016)	10
2.4	Einflüsse der Absatzplanung auf andere Planungsprozesse in der Supply Chain Planning Matrix in Anlehnung an Rohde et al. (2000) und Meyr et al. (2015) . .	11
2.5	Typischer Prognoseprozess in Anlehnung an Kilger und Wagner (2015)	12
2.6	Systemdefinition in Anlehnung an Wiendahl (2014)	14
2.7	Materialbedarfsarten in Anlehnung an Hartmann (2002) und Wiendahl (2014) .	14
2.8	Kundennachfrage als Output der Bedarfsplanung des Kunden	15
3.1	Mengendiagramm zur Einordnung der Gebiete Data Science, künstliche Intelligenz, maschinelles Lernen und überwachtes Lernen	19
3.2	Training und Einsatz von Modellen beim überwachten Lernen	21
3.3	Bestimmung der optimalen Lernepoche für erklärende und prädiktive Modelle in Anlehnung an Hastie et al. (2009)	24
3.4	Unteranpassung, gute Anpassung und Überanpassung bei Regressions- und Klassifikationsmodellen	26
3.5	Unterteilung einer Zeitreihe in eine Trainings-, eine Validierungs- und eine Testmenge sowie Ermittlung des mittleren Validierungs- und des mittleren Testfehlers	30
3.6	Genutzte Werte bei einer Zeitreihenkreuzvalidierung mit fixiertem oder rollierendem Ursprung	32
3.7	Beispielvisualisierung der Ergebnisse eines Nemenyi-Tests mit $M = 8$ Prognoseverfahren	36
3.8	Beispielprognosen verschiedener Verfahren für unterschiedliche Zeitreihen	42
3.9	Prognose durch ein lokales k -nächste-Nachbarn-Modell	51

3.10	Ablaufdiagramm zur Prognose zukünftiger Werte einer Zeitreihe durch ein lokales k -nächste-Nachbarn-Modell	53
3.11	Konfiguration eines Multilayer-Perzeptrons als Zeitreihenprognosemodell	64
3.12	Verschiedene Formen des Trainingsprozesses in Abhängigkeit von der Lernrate η	67
4.1	Ansätze zur Selektion oder Kombination von Prognosemodellen aus einer oder mehreren möglichen Modellklassen für eine Zeitreihe	72
4.2	Taxonomie verschiedener Ansätze zur Lösung des Prognosemodellselektionsproblems	88
5.1	Training und Einsatz des neuartigen Selbbinationsverfahrens	94
5.2	Ablaufdiagramm der Trainingsphase des Selbbinationsverfahrens	96
5.3	Ablaufdiagramm zur Berechnung einer Merkmalsmatrix	97
5.4	STL-Dekomposition in Anlehnung an Hyndman und Athanasopoulos (2018)	99
5.5	Statistische Maßzahlen in der Box-Plot-Darstellung (angelehnt an Sachs und Hedderich (2006))	102
5.6	Ablaufdiagramm zur Berechnung einer Label-Matrix	105
5.7	Ablaufdiagramm zum Training eines Meta-Modells	107
5.8	Training eines Multilayer-Perzeptrons für Klassifikationsprobleme	108
5.9	Validierungsgüte der Meta-Modelle im Merkmalsselektionsprozess	111
5.10	Ablaufdiagramm zum Einsatz des Selbbinationsverfahrens zur Selektion oder Kombination geeigneter Prognosemodelle	113
5.11	Vergleich verschiedener Parameterkonfigurationen des Selbbinationsverfahrens anhand der mittleren Ränge hinsichtlich des RO-sMAPE $_{T_z, T_z+6}^{1,12}$ über die 697 Zeitreihen der Meta-Trainingsmenge	120
6.1	Balkendiagramme der durchschnittlichen Mean-RO-sMAPEs der verschiedenen Prognoseverfahren für kurze, lange und alle Zeitreihen über alle Experimente	128
6.2	Nemenyi-Test-Ergebnisse hinsichtlich der mittleren Fehler der rollierenden Prognosen der verschiedenen Prognoseverfahren gemittelt über alle durchgeführten Experimente für Signifikanzniveaus von 0,05	129
6.3	Nemenyi-Test-Ergebnisse hinsichtlich der Rechenaufwände des Selbbinationsverfahrens sowie verschiedener Benchmark-Verfahren für die Zeitreihen der NN3-Competition	136
6.4	Nemenyi-Test-Ergebnisse hinsichtlich der Fehler der Prognoseverfahrenskandidaten auf der Meta-Trainingsmenge	138
6.5	Vergleich der durchschnittlichen Prognosefehler (RO-sMAPE $_{T_z, T_z+6}^{1,12}$) verschiedener Konfigurationen des Selbbinationsverfahrens für die Zeitreihen der NN3-Competition	140
7.1	Verwendetes Lagerbestandsmodell	144

7.2	Lagerbestandsentwicklung bei Verfolgung eines (R, S_t) -Bestellrhythmusverfahrens in Anlehnung an Günther und Tempelmeier (2012)	146
7.3	Balkendiagramme der RMS(0,5/0,5)-Werte der verschiedenen Prognoseverfahren für kurze, lange und alle Zeitreihen gemittelt über alle Produktionszeiten	152
7.4	Zielkonflikte der erreichten Servicegrade und der durchschnittlichen skalierten Sicherheitsbestände für die verschiedenen Prognoseverfahren	155
C.1	Vergleich verschiedener Parameterkonfigurationen der lokal konstanten und der lokal linearen k -nächste-Nachbarn-Modelle hinsichtlich des Mean-RO-sMAPE $_{T,T+6}^{1,12}$ und der mittleren Rechenaufwände für die 808 industriellen Zeitreihen der M3-Competition	202
C.2	Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 697 langen und kurzen Zeitreihen der Meta-Trainingsmenge	205
C.3	Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 227 kurzen Zeitreihen der Meta-Trainingsmenge	205
C.4	Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 470 langen Zeitreihen der Meta-Trainingsmenge	206

Tabellenverzeichnis

3.1	Beschreibung der Datensätze der monatlichen Industriedaten der M3-Competition und der NN3-Competition	37
3.2	Taxonomie verschiedener exponentieller Glättungsmodelle nach Hyndman und Khandakar (2008)	48
4.1	Vergleich verschiedener Ansätze zur Schätzung einer erwarteten Rangfolge von Prognoseverfahren für eine Zeitreihe	76
4.2	Ansätze zur Modellselektion durch Vergleiche von Prognosefehlern	78
4.3	Ansätze zur Modellselektion durch regelbasierte Expertensysteme	81
4.4	Ansätze zur Modellselektion durch Meta-Lernen	83
5.1	Betrachtete Meta-Merkmale, um ein Prognosemodellselektionsproblem zu beschreiben	100
5.2	Nach der Merkmalsselektion verbleibende Meta-Merkmale, um ein Prognosemodellselektionsproblem zu beschreiben	112
5.3	Beispiele selektierter Prognoseverfahren basierend auf Eignungswahrscheinlichkeiten und der Hybriditätsgrenze Ω_{\min}	116
6.1	Durchschnittliche mittlere Prognosefehler rollierender Prognosen und Ränge des Selbinationsverfahrens sowie gängiger Benchmark-Verfahren für verschiedene Ursprünge, Horizonte und Zeitreihenlängen anhand der Zeitreihen der NN3-Competition	130
6.2	Anzahl der Selektionen der verschiedenen Prognoseverfahrenskandidaten durch Selbination und die anderen Selektionsverfahren mit mehreren möglichen Modellklassen	132
6.3	Mean-RO-sMAPE $_{T_z, T_z}^{1,18}$ der Teilnehmer und Benchmarks der NN3-Competition, der Teilnehmer der M3-Competition sowie der in dieser Arbeit verwendeten Prognoseverfahren für die Zeitreihen der NN3-Competition	133
6.4	Durchschnittlicher Rechenaufwand pro Training eines Modells und pro Prognose von Selbination und den Benchmark-Verfahren	135
6.5	Enthaltene Prognoseverfahrenskandidaten für alle Teilmengen der Menge \mathbf{A} . . .	139

7.1	Variablen und Evaluationskriterien der Lagerbestandssimulation für ein spezifisches Produkt	145
7.2	Konfigurationen der durchgeführten Simulationsexperimente	149
7.3	Durchschnittliche Evaluationsgrößen und Ränge von Selbination und den Benchmark-Verfahren gemittelt über alle Zeitreihen	153
B.1	Ergebnisse der M3-Competition	195
B.2	Ergebnisse der NN3-Competition	197
C.1	Verglichene Parameterkonfigurationen hinsichtlich der Mittelwertbildungen der lokal konstanten (LC) sowie der Regularisierungsparameter der lokal linearen (LL) k -nächste-Nachbarn-Prognosemodelle	201
C.2	Mittlere Fehler und mittlere Rechenaufwände pro Prognose verschiedener Parameterkonfigurationen der lokalen k -nächste-Nachbarn-Modelle für unterschiedliche Zeitreihenlängen	203
C.3	Verglichene statische und dynamische Modellselektionsmethoden für lokal konstante (LC) k -nächste-Nachbarn-Modelle und lokal lineare k -nächste-Nachbarn-Ridge-Regressionsmodelle (LL RR)	204
C.4	Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 697 langen und kurzen Zeitreihen der Meta-Trainingsmenge	207
C.5	Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 227 kurzen Zeitreihen der Meta-Trainingsmenge	208
C.6	Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 470 langen Zeitreihen der Meta-Trainingsmenge	209
D.1	Anzahl der Meta-Merkmale in den Mengen ohne und mit Merkmalsselektion für ausgewählte Konfigurationen des Selbinationsverfahrens hinsichtlich der Menge der Prognoseverfahrenskandidaten \mathbf{A}	210
D.2	Anzahl der Selektionen der verschiedenen Prognoseverfahrenskandidaten für ausgewählte Konfigurationen des Selbinationsverfahrens hinsichtlich \mathbf{A} und Ω_{\min}	211
E.1	Durchschnittliche mittlere absolute skalierte Prognosefehler rollierender Prognosen (Mean-RO-MASE) des Selbinationsverfahrens sowie gängiger Benchmark-Verfahren für verschiedene Ursprünge, Horizonte und Zeitreihenlängen anhand der Zeitreihen der NN3-Competition	212

Abkürzungs- und Formelverzeichnis

Generelle Abkürzungen

ACF	Autokorrelationsfunktion
ADF	Augmented-Dickey-Fuller-Test
agg	aggregiert
AMI	mittlere gemeinsame Information (engl. average mutual information)
ARIMA	Modellklasse der autoregressiven integrierten gleitenden Mittelwerte
ASP	Algorithmusselektionsproblem
ATO	auftragsbezogene Montage (engl. assemble to order)
BFGS	Broyden-Fletcher-Goldfarb-Shanno-Verfahren
CRISP-DM	Standardprozess für Datenanalyse in der Industrie (engl. cross-industry standard process for data mining)
ETO	auftragsbezogene Einmalfertigung (engl. engineer to order)
FNN	Algorithmus der falschen nächsten Nachbarn (engl. false nearest neighbors)
ind	individuell
<i>knn</i>	<i>k</i> -nächste-Nachbarn
LASSO	LASSO-Regression (engl. least absolute shrinkage and selection operator method)
LC	lokal konstant
LL	lokal linear
LL RR	lokal lineare Ridge-Regression
MLP	Multilayer-Perzeptron
MTO	auftragsbezogene Fertigung (engl. make to order)
MTS	Produktion auf Lager (engl. make to stock)
NaN	nicht berechenbarer Wert (engl. not a number)
NN	neuronales Netz
PACF	partielle Autokorrelationsfunktion
PCR	Hauptkomponentenregression (engl. principal component regression)
PMSP	Prognosemodellselektionsproblem
PLSR	partielle Regression der kleinsten Quadrate (engl. partial least squares regression)

PPS	Produktionsplanung und -steuerung
RBF	Rule-Based Forecasting
Rprop	Algorithmus der resilienten Fehlerrückführung mit Gewichtsrücksetzung (engl. resilient backpropagation with weight backtracking)
RQA	Rekurrenzquantifizierungsanalyse
RR	Ridge-Regression
SKU	Bestandseinheit (engl. stock keeping unit)

Prognoseverfahren

auto.arima	allgemeines ARIMA-Modell
Comb-4	Kombination der Prognoseverfahren ets, llknn, nnetar und srw
dt	exponentielle Glättung mit gedämpftem Trend
ets	allgemeine exponentielle Glättung
holt	Holts lineare exponentielle Glättung
hw	saisonale exponentielle Glättung nach Holt-Winters
lcknn	lokal konstantes k -nächste-Nachbarn-Prognosemodell
llknn	lokal lineare k -nächste-Nachbarn-Regression
ML-Sel	Selektion eines Prognosemodells durch ein Meta-Modell
nnetar	neuronales Netz mit Parameteroptimierung durch BFGS-Verfahren
nnfor	neuronales Netz mit Parameteroptimierung durch Rprop
Pool-2	Selektion von zwei Prognosemodellen für eine Kombination
Pool-3	Selektion von drei Prognosemodellen für eine Kombination
rw	nicht-saisonaler Random Walk (auch naives Verfahren)
Selbination	neuartiges Selbinationsverfahren der vorliegenden Arbeit
ses	einfache exponentielle Glättung (engl. simple exponential smoothing)
srw	saisonaler Random Walk (auch saisonales naives Verfahren)
Val-Sel	Selektion eines Prognosemodells durch einen Vergleich von Prognosefehlern auf einer Validierungsmenge

Dynamische Systeme

φ	Vektorfeld eines dynamischen Systems
$\psi(t)$	Zustandsvektor eines dynamischen Systems zum Zeitpunkt t
$\psi_1, \dots, \psi_{Y'}$	Zustände der Einflussgrößen eines dynamischen Systems
\mathcal{Y}'	Phasenraum eines dynamischen Systems
\mathbf{y}	Zeitreihe vergangener Kundenbestellungen bis zum Zeitpunkt T
\mathcal{Y}	Raum der Zeitreihen vergangener Kundenbestellungen

Überwachtes Lernen und Zeitreihenprognose

$a : \mathcal{D}^{\text{D}_{\text{train}}} \rightarrow \mathcal{M}$	überwachtes Lernverfahren, überwachter Lernalgorithmus
\mathcal{A}	Raum der Lernverfahren
\mathcal{A}_{ZRP}	Raum der Zeitreihenprognoseverfahren
AIC	klassisches Akaike-Informationskriterium
AICc	korrigiertes Akaike-Informationskriterium
α	Glättungsparameter eines exponentiellen Glättungsmodells
β	Koeffizientenvektor eines lokal linearen k nn-Regressionsmodells
\mathbf{b}_l	Schwellwertvektor der l -ten Schicht eines Multilayer-Perzeptrons
$b_{l,j}$	Schwellwert des j -ten Neurons der l -ten Schicht eines Multilayer-Perzeptrons
C	Anzahl möglicher Klassen in einem Klassifikationsproblem
$\text{CD}_{(\alpha, M)}$	kritische Distanz für ein Signifikanzniveau α und M Prognoseverfahren
d	Zeitverschiebung
$\mathcal{D} = \mathcal{X} \times \mathcal{L}$	Raum aller Datenpaare von Eingabe- und Ausgabevektoren
$\mathbf{D}_{\text{train}}$	Trainingsmenge
\mathbf{D}_{val}	Validierungsmenge
\mathbf{D}_{test}	Testmenge
$\text{dist}(\mathbf{y}_T, \mathbf{y}_i)$	euklidische Distanz zweier Delay-Koordinaten-Vektoren
δ	Ordnung der ersten Differenzbildungen eines ARIMA-Modells
Δ	Ordnung der saisonalen Differenzbildungen eines ARIMA-Modells
$e : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}$	Fehlerfunktion
$e_{o o+h}$	h -Schritt-Prognosefehler
\mathcal{E}	Raum der Fehlerfunktionen
$E_{\mathbf{D}}$	mittlerer Fehler für die Datenmenge \mathbf{D}
E_j^z	mittlerer Prognosefehler des j -ten Prognoseverfahrens a_j für die Zeitreihe \mathbf{y}^z

$E_o^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}})$	mittlerer Prognosefehler der Horizonte h_{\min} bis h_{\max} vom fixierten Ursprung o
$E_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}})$	mittlerer Prognosefehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}
ϵ_t	weißes Rauschen
$f_j : \mathcal{Y} \rightarrow \mathcal{X}$	Merkmalsextraktionsfunktion
\mathcal{F}	Raum der Merkmalsextraktionsfunktionen
$g_l : \mathbb{R}^{N_l} \rightarrow \mathbb{R}^{N_l}$	Aktivierungsfunktion der l -ten Schicht eines Multilayer-Perzeptrons
h	Prognosehorizont
ι	Lernepoche
k	Anzahl der nächsten Nachbarn
$k\%$	Prozentwert der maximal möglichen zu bestimmenden nächsten Nachbarn
ℓ	Ausgabevektor
$\hat{\ell}$	prognostizierter Ausgabevektor
ℓ_1, \dots, ℓ_C	Zielwerte, Labels
\mathcal{L}	Raum der Ausgabevektoren
λ	Regularisierungsparameter für eine Ridge-Regression oder eine LASSO-Regression
$m_{\text{true}} : \mathcal{X} \rightarrow \mathcal{L}$	wahres Modell
$\hat{m} : \mathcal{X} \rightarrow \mathcal{L}$	statistisches Modell
$\hat{m}_h : \mathcal{X} \rightarrow \mathcal{L}$	h -Schritt-Prognosemodell
M	Anzahl gegebener Prognoseverfahrenskandidaten
\mathcal{M}	Raum aller Modelle
$\mathcal{M}_{\text{Klas}}$	Raum der Klassifikationsmodelle
\mathcal{M}_{Reg}	Raum der Regressionsmodelle
\mathcal{M}_{ZRP}	Raum der Zeitreihenprognosemodelle
N_l	Anzahl der Neuronen in der l -ten Schicht eines Multilayer-Perzeptrons
N	Anzahl der neuronalen Netze zur Kombination in einem Ensemble
\mathbf{netin}_l	Eingabevektor der l -ten Schicht eines Multilayer-Perzeptrons
$\text{netin}_{l,j}$	Eingabewert des j -ten Neurons der l -ten Schicht eines Multilayer-Perzeptrons
η	Lernrate
o	Ursprung einer Prognose
\mathbf{out}_l	Ausgabevektor der l -ten Schicht eines Multilayer-Perzeptrons
$\text{out}_{l,j}$	Ausgabewert des j -ten Neurons der l -ten Schicht eines Multilayer-Perzeptrons
p	Ordnung der Autoregression eines ARIMA-Modells
P	Ordnung der saisonalen Autoregression eines ARIMA-Modells

q	Ordnung der Mittelwerte eines ARIMA-Modells
Q	Ordnung der saisonalen Mittelwerte eines ARIMA-Modells
R_j^z	Rang des j -ten Verfahrens in einer Menge von M Prognoseverfahren oder -modellen für die Zeitreihe \mathbf{y}^z
\bar{R}_j	mittlerer Rang des j -ten Prognoseverfahrens in einer Menge von M Verfahren für eine Menge von Z Zeitreihen
r	Einbettungsdimension
$r\%$	Prozentwert der maximal möglichen Einbettungsdimension
\tilde{r}	Regularisierungsparameter für eine PCR- oder eine PLSR-Regression
$\text{RSS}(\boldsymbol{\theta})$	Summe der quadrierten Residuen für den Parametervektor $\boldsymbol{\theta}$
t	Periode einer Zeitreihe
T	aktueller Zeitpunkt
$\tau_- : \mathbb{Z} \rightarrow \mathbb{Z}$	Rückverschiebungsoperator auf den letzten Zeitreihenwert
$\tau_h(\mathbf{y}_T^{\text{NN},i}) = \chi_h^i$	h -ter zukünftiger Wert des i -ten nächsten Nachbarn zum Abfragevektor
$\boldsymbol{\theta}$	Vektor der Modellparameter
\mathbf{w}	Vektor aller Gewichte und Schwellwerte eines Multilayer-Perzeptrons
$\mathbf{w}_{(\iota)}$	Parametersatz eines Multilayer-Perzeptrons in Lernepoche ι
$\Delta \mathbf{w}_{(\iota)}$	Parameteranpassungen in Lernepoche ι
\mathbf{W}_l	Gewichtungsmatrix der Ausgabewerte der l -ten Schicht eines Multilayer-Perzeptrons
\mathbf{x}	Merkmalsvektor, Eingabevektor
x_1, \dots, x_F	Merkmale
\mathbf{X}	Merkmalsmatrix
\mathcal{X}	Raum der Merkmalsvektoren
$\boldsymbol{\chi}_h$	Vektor der h -ten zukünftigen Werte der nächsten Nachbarn zum Abfragevektor
$(\mathbf{x}^i, \boldsymbol{\ell}^i)$	Trainingsbeispiel
y_j	j -ter Wert einer Zeitreihe
y_{T+h}	h -ter zukünftiger Wert einer Zeitreihe
$\hat{y}_{t t+h}$	h -Schrittprognose des Zeitreihenwertes für die Periode $t+h$ ermittelt zum Zeitpunkt t
$y_{T,j}^{\text{NN},i}$	j -te Komponente des i -ten nächsten Nachbarn zum Abfragevektor
\mathbf{y}	Zeitreihe
$\mathbf{y}_{\text{train}}$	Trainingsmenge einer Zeitreihe
\mathbf{y}_{val}	Validierungsmenge einer Zeitreihe
\mathbf{y}_{test}	Testmenge einer Zeitreihe
$\mathbf{y}_t = \mathbf{y}_{t,r,d}$	Delay-Koordinaten-Vektor der Länge r mit Zeitverschiebung d bis zum Zeitpunkt t

$\mathbf{y}_T = \mathbf{y}_{T,r,d}$	Abfrage-Vektor
$\mathbf{y}_T^{\text{NN},i}$	i -ter nächster Nachbar zum Abfragevektor
$\hat{\mathbf{y}}$	Prognosematrix
\mathbf{Y}	Menge von Zeitreihen
$\hat{\mathbf{Y}}$	Menge von Prognosematrizen
$\mathbf{Y}_{r,d}$	Trajektorienmatrix
\mathcal{Y}''	Attraktor eines dynamischen Systems
Z	Anzahl gegebener Zeitreihen
$\lfloor x \rfloor$	Abrundung der Zahl x auf die größte ganze Zahl

Prognosefehlermaße

$\text{AE}_{o o+h}$	absoluter h -Schritt-Prognosefehler
$\text{APE}_{o o+h}$	absoluter prozentualer h -Schritt-Prognosefehler
$\text{ASE}_{o o+h}$	absoluter skaliertes h -Schritt-Prognosefehler
$\text{sAPE}_{o o+h}$	symmetrischer absoluter prozentualer h -Schritt-Prognosefehler
$\text{SE}_{o o+h}$	quadratischer h -Schritt-Prognosefehler
$\text{MAE}_o^{h_{\min}, h_{\max}}$	mittlerer absoluter Prognosefehler der Horizonte h_{\min} bis h_{\max} vom fixierten Ursprung o
$\text{MAPE}_o^{h_{\min}, h_{\max}}$	mittlerer absoluter prozentualer Prognosefehler der Horizonte h_{\min} bis h_{\max} vom fixierten Ursprung o
$\text{MASE}_o^{h_{\min}, h_{\max}}$	mittlerer absoluter skaliertes Prognosefehler der Horizonte h_{\min} bis h_{\max} vom fixierten Ursprung o
$\text{MSE}_o^{h_{\min}, h_{\max}}$	mittlerer quadratischer Prognosefehler der Horizonte h_{\min} bis h_{\max} vom fixierten Ursprung o
$\text{sMAPE}_o^{h_{\min}, h_{\max}}$	symmetrischer mittlerer absoluter prozentualer Prognosefehler der Horizonte h_{\min} bis h_{\max} vom fixierten Ursprung o
$\text{RO-MAE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$	mittlerer absoluter Prognosefehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}
$\text{RO-MAPE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$	mittlerer absoluter prozentualer Prognosefehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}
$\text{RO-MASE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$	mittlerer absoluter skaliertes Prognosefehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}

$\text{RO-MSE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$	mittlerer quadratischer Prognosefehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}
$\text{RO-sMAPE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$	symmetrischer mittlerer absoluter prozentualer Prognosefehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}
$\text{Mean-RO-MASE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$	durchschnittlicher mittlerer absoluter skaliertes Fehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}
$\text{Mean-RO-sMAPE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$	durchschnittlicher symmetrischer mittlerer absoluter prozentualer Fehler der rollierenden Prognosen der Horizonte h_{\min} bis h_{\max} über die Ursprünge o_{\min} bis o_{\max}

Selektion und Kombination von Prognosemodellen

A	Menge an Prognoseverfahrenskandidaten
$\mathcal{A}_{\text{ZRP}}^{\mathcal{C}}$	Raum der Prognoseverfahren der Klasse \mathcal{C}
$\mathcal{A}_{\text{ZRP}}^{\mathcal{C}_1, \dots, \mathcal{C}_M}$	Raum aller Prognoseverfahren der Klassen $\mathcal{C}_1, \dots, \mathcal{C}_M$
$\mathbf{A}_M^{\mathcal{C}}$	Menge der M Prognoseverfahrenskandidaten für die Kandidatenauswahl nach Klassen
\mathbf{A}_M^E	Menge der M Prognoseverfahrenskandidaten für die Kandidatenauswahl nach Fehlern
\mathcal{C}	Klasse von Prognoseverfahren
E	Prognosefehlermatrix
F	Menge an Merkmalsextraktionsfunktionen
F	Anzahl verwendeter Merkmalsextraktionsfunktionen
L	Label-Matrix
μ	Momentum
\hat{m}^{NN}	Meta-Modell
M	Anzahl der Prognoseverfahrenskandidaten
$\mathcal{M}_{\text{ZRP}}^{\mathcal{C}}$	Raum der Prognosemodelle der Klasse \mathcal{C}
M_{sel}	Anzahl der zu selektierenden Prognoseverfahren
$\hat{m}_h^{\text{Comb}} : \mathcal{X} \rightarrow \mathcal{L}$	h -Schritt-Kombinationsmodell
out	Vektor der Eignungswahrscheinlichkeiten von Prognoseverfahrenskandidaten für eine Zeitreihe
$\hat{\mathbf{R}}$	erwartete Rangfolge von Prognoseverfahrenskandidaten
\hat{R}_j	erwarteter Rang des j -ten Prognoseverfahrenskandidaten

\mathcal{R}	Raum der (erwarteten) Rangfolgen von Prognoseverfahren
$s_1 : \mathcal{R} \rightarrow \mathbf{A}$	Selektionsfunktion zur Selektion eines geeigneten Prognoseverfahrens für eine Zeitreihe
$s_{M_{\text{sel}}} : \mathcal{R} \rightarrow (\mathbf{A})^{M_{\text{sel}}}$	M_{sel} -fache Selektionsfunktion
$\Theta_{\text{ZRP}}^{\mathcal{C}}$	Raum der Parametervektoren der Klasse \mathcal{C}
θ	Parametervektor eines Prognosemodells
Ω_{min}	Hybriditätsgrenze
\mathbf{X}	Merkmalsmatrix
$\mathbf{Y}_{\text{train}}$	Menge aller Trainingsmengen von Zeitreihen
\mathbf{Y}_{test}	Menge aller Testmengen von Zeitreihen

Lagerbestandssimulation

α	erreichter Servicegrad
α^*	Zielservicegrad
$\text{Bias}_{W,t}$	Verzerrung der W -Schritt-Prognosen eines Verfahrens
$e_{W,t}$	W -Schritt-Prognosefehler der am Ende von Periode t ermittelten Prognosen
ϵ_W	Zufallsvariable der W -Schritt-Prognosefehler
i_t	Nettobestand fertiger Produkte am Ende von Periode t
i_t^+	physischer Bestand fertiger Produkte auf Lager am Ende von Periode t
i_t^-	lieferrückständige Produkte am Ende von Periode t
I_t	disponibler Bestand am Ende von Periode t
p_t	Produktionsaufträge generiert am Ende von Periode t
R	Überprüfungsintervall
$\text{RMS}(\cdot/\cdot)$	Quadratmittelwert der erreichten Servicegrade und der skalierten Sicherheitsbestände
s_t	dynamischer Sicherheitsbestand am Ende von Periode t
S_t	dynamisches Zielbestellniveau am Ende von Periode t
$\hat{\sigma}_{e_t}$	empirische Standardabweichung der Einschnittprognosefehler bis Periode t
s_{scaled}	skaliertes Sicherheitsbestände
W	Produktionszeit (Zeit zur Produktion eines neuen Produktes)
w_t	Umlaufbestand am Ende von Periode t
y_t	Kundennachfrage innerhalb der Periode t
$\hat{y}_{t-j t}$	Prognose der Kundennachfrage für Periode t berechnet am Ende von Periode $t - j$

Kerndichteschätzung

B	Bandbreite
$f_{\epsilon_W}(\cdot)$	wahre Wahrscheinlichkeitsdichtefunktion
J	Stichprobengröße
$\mathcal{K}(\cdot)$	Keroglättungsfunktion
$\mathcal{K}_{\text{Epanechnikov}}(\cdot)$	Epanechnikov-Kern
ζ	adaptiver Ausdehnungswert

Kapitel 1

Einleitung

„I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year.“

Chefredakteur Business Books bei Prentice Hall (1957)

1.1 Motivation

Da produzierende Unternehmen diverse Planungsentscheidungen treffen müssen bevor die zukünftigen Entwicklungen wichtiger Einflussgrößen bekannt sind, werden Nachfrageprognosen benötigt (vgl. Syntetos et al., 2016). Neben Prognosen für Ressourcen- und Energiebedarfe (vgl. Hahn et al., 2009; Weinert et al., 2011; Hong und Fan, 2016; Khan et al., 2016), Prognosen für Elektrizitätspreise (vgl. Aggarwal et al., 2009; Weron, 2014), Standmengen- und Ausfallprognosen für die Instandhaltungsplanung von Maschinen (vgl. Peng et al., 2010; Sikorska et al., 2011; Fritzsche et al., 2014; Hellingrath und Cordes, 2014; Gao et al., 2015) sowie Ertrags- und Qualitätsprognosen (vgl. Lieber et al., 2013; Colledani et al., 2014) werden speziell Prognosen der Kundennachfrage (vgl. Kilger und Wagner, 2015; Wiendahl, 2014) benötigt. Insbesondere bei Verfolgung der Strategie einer Produktion auf Lager, wie beispielsweise häufig in der Konsumgüterindustrie, sind Absatzprognosen von besonderer Bedeutung (vgl. Bilgen und Günther, 2010). In diesem Fall basieren Rohstoffbestellungen und Produktionsprogramme hauptsächlich auf Nachfrageprognosen und Kundenbedarfe werden direkt aus einem Fertigwarenlager bedient (vgl. Wiendahl, 2014).

Häufig müssen produzierende Unternehmen die Kundennachfrage für eine breite Produktpalette basierend auf Zeitreihen historischer Kundenbestellungen simultan prognostizieren. Somit werden automatische Prognoseverfahren benötigt, um den manuellen Aufwand zur Anpassung von Prognosemodellen für jede einzelne Zeitreihe zu reduzieren. Während simple Prognoseverfahren, wie gleitende Mittelwerte, einfache exponentielle Glättungsverfahren oder lineare Regressionsverfahren weiterhin häufig in der industriellen Praxis verwendet werden, zeigen komplexe Varianten

exponentieller Glättungsverfahren, ARIMA-Verfahren oder neuronale Netze großes Potenzial in aktuellen empirischen Studien oder Prognosewettbewerben (vgl. Hyndman, 2020).

Allerdings besagt das No Free Lunch Theorem, dass kein universell bestes Prognoseverfahren existiert, welches für alle Datensätze eines breiten Anwendungsgebietes bessere Prognosen ermittelt als alle anderen Verfahren (vgl. Wolpert, 1996). Dieses theoretische Ergebnis wurde im Rahmen diverser empirischer Studien bestätigt (vgl. Makridakis und Hibon, 2000; Crone et al., 2011; Petropoulos et al., 2014; Makridakis et al., 2020). Aus diesem Grund werden Selektionsverfahren benötigt, die geeignete Prognosemodelle für jede Zeitreihenentwicklung auswählen können. Die existierenden Ansätze zur Selektion von Prognosemodellen lassen sich in drei verschiedene Kategorien einteilen: aggregierte Selektionsansätze, individuelle Selektionsansätze und Kombinationsansätze. Individuelle Selektionsverfahren bieten das Potenzial, für jede Zeitreihe ein individuell geeignetes Prognosemodell zu selektieren, während aggregierte Selektionsverfahren das gleiche Modell für die gesamte Menge gegebener Zeitreihen verwenden (vgl. Fildes und Petropoulos, 2015). Kombinationsverfahren reduzieren das Risiko zur Selektion eines schlechten Prognosemodells, da die Stärken und Schwächen mehrerer Modelle sich zu einer robusten Prognose ausgleichen (vgl. Elliott und Timmermann, 2016; Hyndman, 2020). Wenn die Anzahl der zu kombinierenden Modelle geringer ist als die Anzahl aller Prognoseverfahrenskandidaten, wird ein Kombinationsverfahren auch als Pooling-Verfahren bezeichnet (vgl. Geweke und Amisano, 2011; Kourentzes et al., 2019). Dieser Ansatz bietet die zusätzliche Möglichkeit, Prognosemodelle von der Kombination auszuschließen, für die keine guten Prognosen für eine gegebene Zeitreihe zu erwarten sind.

Zur Selektion eines oder mehrerer Prognosemodelle wurden die folgenden Selektionskriterien vorgeschlagen: klassische Informationskriterien, Vergleiche von Prognosefehlern, regelbasierte Expertensysteme sowie Meta-Lernansätze. Während Informationskriterien nur für Modellklassen mit einer soliden statistischen Theorie verwendet werden können, können die anderen genannten Kriterien verwendet werden, um Modelle aus mehreren möglichen Modellklassen zu selektieren. Insbesondere Meta-Lernansätze haben in aktuellen Studien vielversprechende Ergebnisse erzielt (vgl. Lemke und Gabrys, 2010a; Kück et al., 2016b; Talagala et al., 2018; Montero-Manso et al., 2020; Ma und Fildes, 2021).

Bei Verwendung der bisher entwickelten Selektionsansätze aus dem Stand der Forschung wird stets im Voraus festgelegt, ob ein einzelnes Modell selektiert oder mehrere Modelle kombiniert werden sollen. Allerdings bleiben hierdurch möglicherweise Potenziale ungenutzt, die zu einer höheren Prognosegüte und somit zu einer besseren Grundlage für die Produktionsplanung und die Bestandsplanung führen können. Zur Ausschöpfung dieser Potenziale wird in der vorliegenden Arbeit ein hybrides Verfahren entwickelt, welches basierend auf den Charakteristika vorliegender Zeitreihen die Eignungen verschiedener Prognoseverfahrenskandidaten schätzt und auf dieser Grundlage ein oder mehrere geeignete Prognoseverfahren selektiert.

1.2 Zielsetzung und Forschungsthesen

Die Zielsetzung dieser Arbeit besteht in der Entwicklung eines hybriden Meta-Lernverfahrens zur automatischen Selektion oder Kombination geeigneter Nachfrageprognosemodelle für die Produktionsplanung. Die Motivation dieses Ansatzes beruht auf den folgenden Forschungsthesen:

FT 1. Es existiert keine stets beste Modellklasse zur Prognose der Kundennachfrage produzierender Unternehmen.

FT 2. Meta-Lernansätze eignen sich besser zur Selektion geeigneter Prognoseverfahren als Vergleiche von Validierungsfehlern.

FT 3. Durch ein hybrides Verfahren zwischen Selektion und Kombination lässt sich eine höhere Prognosegüte erzielen als durch gängige individuelle Selektions- oder Kombinationsverfahren.

FT 4. Durch das hybride Verfahren lässt sich eine verbesserte Lagerbestandsplanung erreichen als durch gängige individuelle Selektions- und Kombinationsverfahren.

1.3 Wissenschaftlicher Beitrag der Arbeit

Teile der Arbeit wurden bereits im Vorfeld in Fachzeitschriften und Konferenzbänden publiziert (vgl. Anhang A und Kück et al., 2016b; Kück und Freitag, 2021). Insgesamt leistet die vorliegende Arbeit die folgenden wissenschaftlichen Beiträge.

- **Entwicklung eines neuartigen hybriden Verfahrens zwischen Selektion und Kombination.** Während für bisherige Verfahren aus dem Stand der Forschung stets unabhängig von den Charakteristika einer gegebenen Zeitreihe festgelegt wird, ob ein Prognosemodell selektiert oder mehrere kombiniert werden, trifft das in der vorliegenden Arbeit entwickelte, sogenannte Selbinationsverfahren diese Entscheidung datengetrieben. Das Verfahren besteht aus einem Meta-Modell zur Schätzung von Eignungswahrscheinlichkeiten für Prognoseverfahrenskandidaten sowie einer Selektionsfunktion, die basierend auf den Eignungswahrscheinlichkeiten und einer sogenannten Hybriditätsgrenze ein oder mehrere Prognoseverfahren selektiert, die darauffolgend verwendet werden, um Prognosemodelle zu trainieren. Das Verfahren wird zunächst formal durch mathematische Gleichungen und Ablaufdiagramme beschrieben. Danach wird es in umfangreichen empirischen Studien anhand realer industrieller Zeitreihen im Vergleich zu gängigen Selektions-, Kombinations- und Pooling-Verfahren aus dem Stand der Forschung evaluiert. Neben durchschnittlichen Fehlern werden Friedman- und Nemenyi-Tests durchgeführt, um die Signifikanz der Ergebnisse zu testen.
- **Evaluation in einer Lagerbestandssimulation.** Im Gegensatz zu den meisten Studien im Stand der Forschung, welche Prognoseverfahren lediglich anhand ihrer Prognosefehler

bewerten, wird in dieser Arbeit zusätzlich eine Lagerbestandssimulationsstudie durchgeführt, um die Leistung des Selbinationsverfahrens hinsichtlich der erreichten Servicegrade und der notwendigen Sicherheitsbestände zu evaluieren. Diese Untersuchungen testen zusätzlich die Übertragbarkeit der Prognoseverfahren auf die industrielle Praxis.

- **Vorstellung einer neuen Taxonomie zur Unterscheidung verschiedener Selektionsansätze.** Nach der Formulierung des sogenannten Prognosemodellselektionsproblems dieser Arbeit wird eine neue schematische Taxonomie zur Unterscheidung verschiedener Lösungsansätze für dieses Problem vorgestellt. In der Taxonomie werden sowohl Selektions- als auch Kombinations- und Pooling-Verfahren unterschieden sowie Verfahren, die aus einer vorgegebenen oder mehreren möglichen Modellklassen selektieren. Zudem werden in dieser Arbeit im Rahmen einer breit angelegten Literaturanalyse relevante Selektionsansätze für Nachfrageprognosemodelle zusammengefasst sowie Best Practices und Limitierungen dieser Ansätze abgeleitet. Dabei werden die Konzepte der Selektions-, Kombinations- und Pooling-Verfahren in einer selbst entwickelten mathematischen Notation dargestellt.
- **Einführung neuer Merkmalsextraktionsfunktionen.** Im Rahmen der Entwicklung des Selbinationsverfahrens werden zwei neue Mengen von Merkmalsextraktionsfunktionen eingeführt, die die Verhältnisse und Differenzen der Validierungsfehler der Prognoseverfahrenskandidaten angeben. Die neu eingeführten Merkmale zeigen eine gute Anwendbarkeit für das Training und die Verwendung des Meta-Modells des Selbinationsverfahrens.
- **Entwicklung einer neuen dynamischen Modellselektionsstrategie für lokale k -nächste-Nachbarn-Modelle.** Da lokale k -nächste-Nachbarn-Modelle zum Teil vielversprechende Prognoseergebnisse in aktuellen Evaluationsstudien erzielen, jedoch keine allgemein anerkannte Strategie zur Selektion der Modellparameter existiert, wird im Rahmen dieser Arbeit eine neuartige dynamische Modellselektionsstrategie eingeführt. In einer Vorstudie zu dieser Arbeit führte die Modellselektionsstrategie zu vielversprechenden Prognoseergebnissen. Daher wird sie für die Verwendung der lokalen k -nächste-Nachbarn-Modelle dieser Arbeit genutzt.

1.4 Aufbau der Arbeit

Der Aufbau der vorliegenden Arbeit orientiert sich am Standardprozess für Data-Science-Anwendungen in der Industrie (CRISP-DM, Cross-industry standard process for data mining, vgl. Abbildung 1.1 und Shearer, 2000). Zunächst wird in den Kapiteln 2-4 ein Verständnis für die zugrundeliegenden Prozesse und Daten gewonnen. Kapitel 2 beschreibt die Grundlagen der Absatzplanung. Dabei werden die notwendigen Zielgrößen produzierender Unternehmen definiert, anhand derer Prognoseverfahren zu evaluieren sind. Zudem wird der Prozess der Absatzplanung beschrieben und es wird erläutert, wie die Nachfrage eines Kunden nach einem Produkt eines

produzierenden Unternehmens ermittelt wird. Darauffolgend werden in Kapitel 3 die Grundlagen und der Stand der Forschung zur Zeitreihenprognose dargestellt. Neben gängigen Prognosemodellklassen und Prognosefehlermaßen werden insbesondere auch die genutzten Datensätze der vorliegenden Arbeit beschrieben, die monatlichen industriellen Zeitreihen der M3- und der NN3-Competition. Kapitel 4 stellt den Stand der Forschung hinsichtlich der Selektion und Kombination von Prognosemodellen dar. Die verschiedenen Ansätze werden in Form einer einheitlichen mathematischen Notation beschrieben und in einer neuartigen Taxonomie zusammengefasst.

Die folgenden Schritte im CRISP-DM-Prozess, die Datenvorverarbeitung und die Modellierung, erfolgen in Kapitel 5. Es wird die Extraktion der notwendigen Charakteristika zur Beschreibung vorliegender Zeitreihen beschrieben. Darauffolgend wird das neuartige Selbinationsverfahren dieser Arbeit eingeführt, welches aus einem Meta-Modell und einer Selektionsfunktion mit einer Hybriditätsgrenze besteht. Dieses Verfahren wird in den Kapiteln 6 und 7 hinsichtlich seiner Prognosegüte und der Leistung in einer Lagerbestandssimulation evaluiert. Der letzte Schritt des CRISP-DM-Prozesses ist der Schritt des Einsatzes eines Verfahrens. Hiermit ist der Einsatz im laufenden Prozess eines Unternehmens gemeint und somit befindet sich dieser Schritt außerhalb des Fokus der vorliegenden Arbeit. Allerdings wird im Rahmen der Schlussbetrachtung in Kapitel 8 nach einer Zusammenfassung der Arbeit sowie einer Beantwortung der aufgestellten Forschungsthesen ein Ausblick auf mögliche Anwendungen in der industriellen Praxis gegeben. Zudem werden mögliche weitere Anwendungsfelder für das Selbinationsverfahren angedeutet.

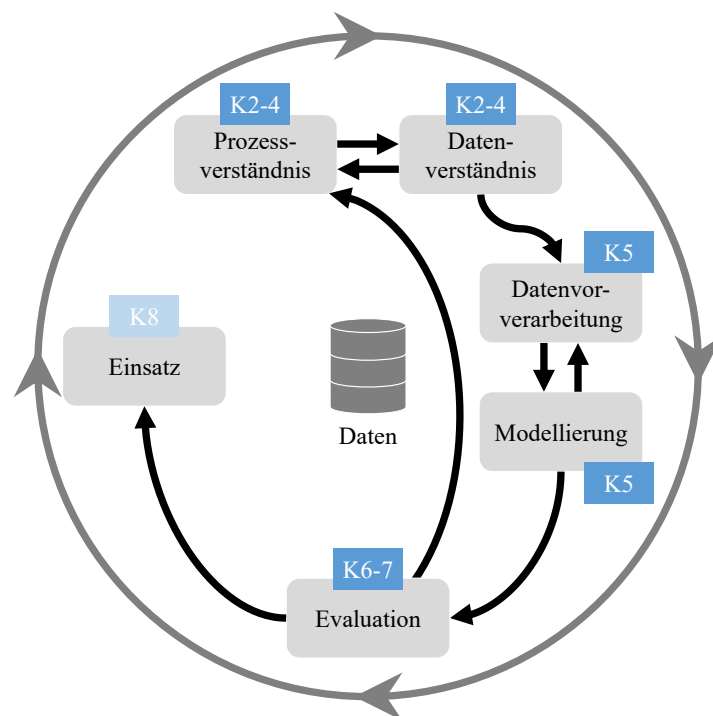


Abbildung 1.1: Aufbau der Arbeit in Anlehnung an den Cross-industry standard process for data mining (vgl. Shearer, 2000)

Kapitel 2

Grundlagen der Absatzplanung

„Gegner der Planung sind Freunde des Zufalls.“

Manfred Rommel, deutscher Politiker (1928-2013)

Dieses Kapitel erläutert die Grundlagen der Absatzplanung. Nach einer Definition relevanter Zielgrößen für produzierende Unternehmen werden die Bedeutung und der Prozess der Absatzplanung erläutert. Als Basis für die Nachfrageprognosen dieser Arbeit wird abschließend die Bedarfsplanung eines Kunden beschrieben und als dynamisches System modelliert.

2.1 Ziele produzierender Unternehmen

Ein *produzierendes Unternehmen* (engl. manufacturing company) lässt sich definieren als „eine leistungserbringende, gewinnorientierte Betriebseinheit, die aus Rohstoffen und Hilfsgütern mit Hilfe von Menschen, Energie und Kapital Fertigerzeugnisse herstellt“ (Wiendahl, 2014, S.18) und einer bestimmten Branche angehört. Das Ziel eines produzierenden Unternehmens besteht in der Erwirtschaftung eines möglichst hohen Gewinns mit möglichst niedrigem Aufwand (vgl. Wöhe et al., 2016). Zur Erreichung dieses Ziels muss das Unternehmen das richtige Produkt in der richtigen Qualität zur richtigen Zeit am richtigen Ort bereitstellen (vgl. Pfohl, 2018). Im Folgenden sei angenommen, dass ein Unternehmen über ein sinnvolles Produktprogramm verfügt, mit dem ein bestimmter Umsatz zu erreichen ist (vgl. Meffert et al., 2015). Ferner sei angenommen, dass das Unternehmen über die notwendigen Produktionsressourcen verfügt, um Produkte in ausreichender Qualität und Menge zu produzieren (vgl. Pfeifer und Schmitt, 2014). Dann stellt das Zielsystem der Produktionslogistik nach Wiendahl (2014) die wesentlichen Kriterien zur Messung der Leistungsfähigkeit eines produzierenden Unternehmens dar (vgl. Abbildung 2.1).

Demnach sollte ein Unternehmen danach streben, die Logistikleistung zu maximieren und gleichzeitig die Logistikkosten zu minimieren, um eine hohe Wirtschaftlichkeit zu erzielen. Eine hohe Logistikleistung lässt sich zum einen durch einen hohen Servicegrad und zum anderen durch kurze Lieferzeiten erreichen. In diesem Zusammenhang lässt sich der *Servicegrad* (engl. service level) für ein Produkt entweder als das Verhältnis der Anzahl an Perioden, in denen

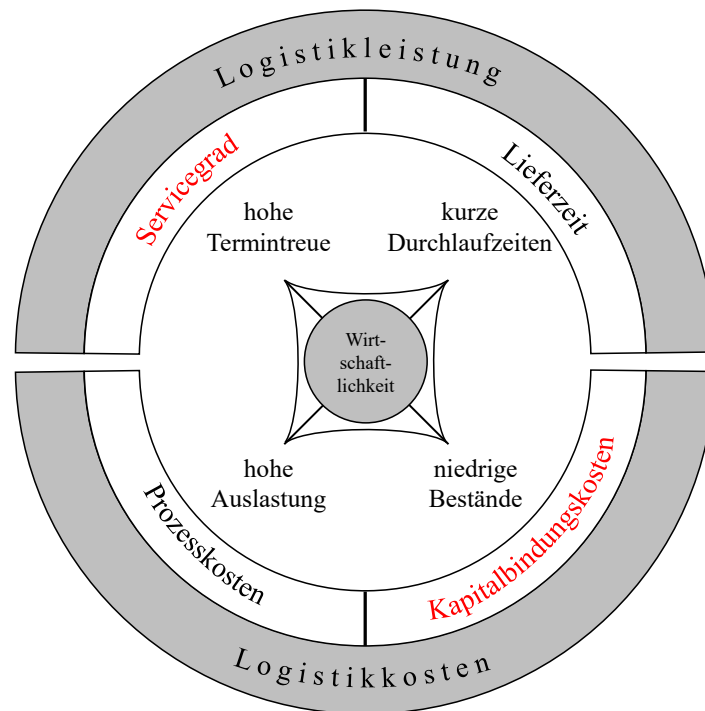


Abbildung 2.1: Zielsystem der Produktionslogistik in Anlehnung an Wiendahl (2014)

die gesamte Nachfrage erfüllt werden konnte, zu der Gesamtanzahl an Perioden definieren (α -Servicegrad) oder als das Verhältnis der Summe der erfüllten Nachfragemengen zu der Summe aller Nachfragemengen nach dem Produkt (β -Servicegrad). Die Logistikkosten setzen sich aus den Prozesskosten und den Kapitalbindungskosten zusammen. Während die Lieferzeiten und die Prozesskosten in dieser Arbeit nicht näher betrachtet werden, stellen der Servicegrad und die Kapitalbindungskosten zwei zentrale Evaluationskriterien der Arbeit dar, zwischen denen ein Zielkonflikt besteht (vgl. Kapitel 7). Geringe Kapitalbindungskosten lassen sich durch niedrige Bestände erreichen. Allerdings führt dies zu einem hohen Risiko von Fehlmengen, welche wiederum zu einem niedrigen Servicegrad führen. Eine hohe Termintreue bei der Produktion wirkt sich positiv auf den Servicegrad aus. Allerdings gilt dies nur unter der Voraussetzung, dass die Fertigung der richtigen Produkte in der richtigen Menge im Produktionsprogramm geplant wurde. Eine wichtige Grundlage für die Produktionsprogrammplanung eines Unternehmens bildet die Absatzplanung, die im folgenden Abschnitt erläutert wird.

2.2 Bedeutung und Prozess der Absatzplanung

2.2.1 Übersicht

In der vorliegenden Arbeit wird unter einer *Nachfrageprognose* (engl. demand forecast) die Vorhersage der bislang unbekanntem Menge an Einheiten eines Produktes verstanden, die zu einem bestimmten zukünftigen Zeitpunkt in einer bestimmten Region von Kunden eines produzierenden

Unternehmens nachgefragt wird (vgl. Kilger und Wagner, 2015). Nachfrageprognosen bilden die Grundlage der Absatzplanung, welche wiederum eine wichtige Basis für diverse innerbetriebliche Entscheidungsprozesse bildet. Im Folgenden wird zunächst die Bedeutung der Absatzplanung in verschiedenen Branchen beschrieben und darauffolgend der Prozess der Absatzplanung erläutert.

2.2.2 Bedeutung der Absatzplanung in verschiedenen Branchen

Nach Wiendahl (2014) bezeichnet der sogenannte *Kundenauftragsentkopplungspunkt* (engl. order decoupling point) die Stelle in der innerbetrieblichen Logistikkette aus Beschaffung, Fertigung, Montage und Versand, ab der die Aufträge konkreten Kundenbestellungen zugeordnet sind. Vor dieser Stelle werden Komponenten und Produkte auftragsunabhängig auf Basis von Nachfrageprognosen gefertigt, danach werden diese auftragsbezogen finalisiert. Dabei wird meist am Entkopplungspunkt ein Sicherheitsbestand vorgehalten, um mögliche Prognosefehler kompensieren zu können (vgl. Fleischmann und Meyr, 2004). Abbildung 2.2 stellt die Lage des Kundenauftragsentkopplungspunktes für vier unterschiedliche Bevorratungsstrategien dar: (i) Produktion auf Lager, (ii) auftragsbezogene Montage, (iii) auftragsbezogene Produktion sowie (iv) auftragsbezogene Einmalfertigung. Die Wahl einer geeigneten Bevorratungsstrategie hängt dabei vom Verhältnis der marktüblichen Lieferzeit eines Produktes zur Durchlaufzeit ab.

Für den Fall einer *Produktion auf Lager* (MTS, engl. make to stock) wird auf Basis von Nachfrageprognosen beschafft, gefertigt und montiert. Die fertigen Erzeugnisse werden in Fertigwarenlagern gelagert, aus denen Kunden bei einer Nachfrage direkt beliefert werden. Produkte, die in der Regel auf Lager produziert werden, finden sich beispielsweise häufig in der Konsumgüterindustrie. Dies können z.B. Lebensmittel, Getränke, Büroartikel oder standardisierte Elektronikprodukte, wie Kameras oder Haushaltsgeräte, sein (vgl. Bilgen und Günther, 2010; Wiendahl, 2014; Meyr und Stadtler, 2015). Diese zeichnen sich meist durch hohe Standardisierung und einen geringen Wert pro Produkteinheit aus. Sie werden häufig in großen Mengen in Fließfertigungen mit einem hohen Automatisierungsgrad gefertigt. Falls ein Produkt nicht vorrätig ist, entscheidet sich ein Endkunde aufgrund der in der Konsumgüterindustrie üblichen geringen Produktdifferenzierung oft für ein vergleichbares Produkt eines Wettbewerbers (vgl. Meyr und Stadtler, 2015). Daher müssen Konsumgüterhersteller im Allgemeinen auf Lager produzieren. Weil die Produktlebenszyklen von Konsumgütern meist mehrere Jahre betragen, lassen sich Nachfrageprognosen auf Basis historischer Daten ermitteln. Neben verschiedenen Industriezweigen wird zudem auch im Handel häufig eine MTS-Strategie verfolgt.

Bei steigender Variantenzahl ist eine Produktion aller Varianten auf Lager meist nicht mehr möglich, da die Kapitalbindungskosten ansonsten zu hoch wären. Stattdessen wählen Unternehmen in solchen Fällen häufig die Bevorratungsstrategie der *auftragsbezogenen Montage* (ATO, engl. assemble to order). Hierbei werden Standardkomponenten auf Basis von Nachfrageprognosen vorgefertigt aber die Endmontage zum fertigen Produkt erfolgt ebenso wie die Auslieferung zum Kunden erst nach Eingang von Kundenbestellungen. Beispiele für Produkte,

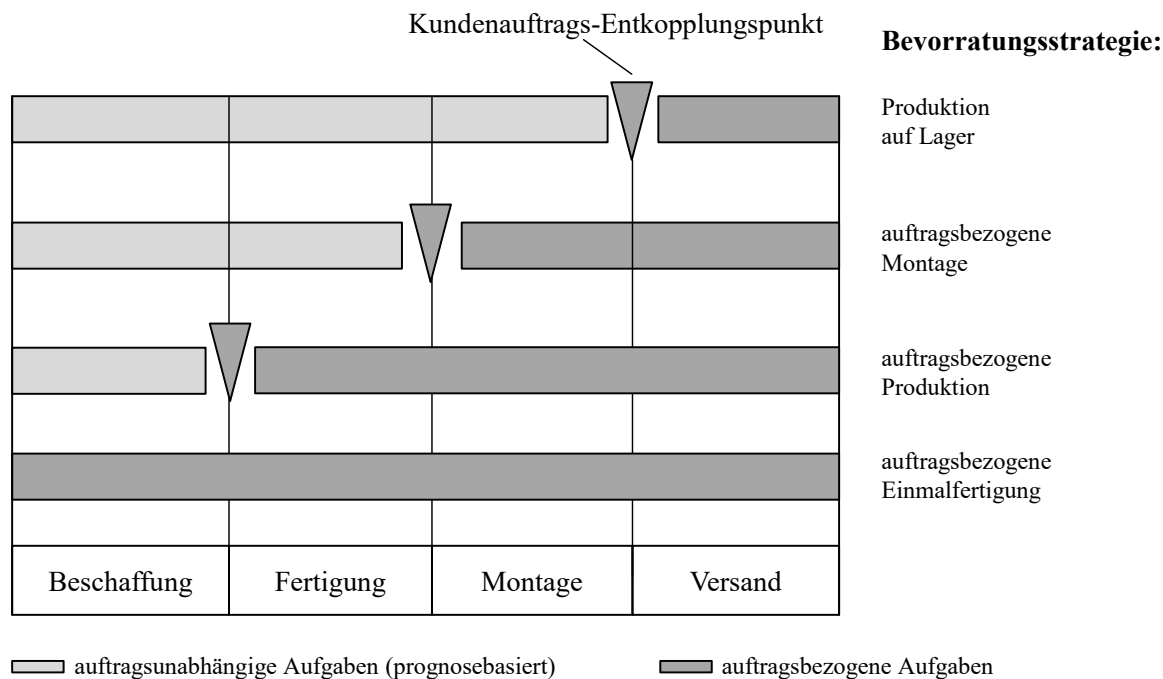


Abbildung 2.2: Bevorratungsstrategien in Anlehnung an Wiendahl (2014)

die meist auftragsbezogen montiert werden, sind Kraftfahrzeuge aus Standardkomponenten oder Computer (vgl. Wiendahl, 2014; Meyr und Stadtler, 2015). In diesen Fällen hat ein produzierendes Unternehmen in der Regel sehr viele Zulieferer, die Standardkomponenten liefern, welche in wenigen Montagewerken finalisiert werden.

Eine ähnliche Strategie, bei der der Kundenauftragsentkopplungspunkt sich jedoch früher in der Logistikkette befindet, ist die *auftragsbezogene Fertigung* (MTO, engl. make to order). In diesem Fall wird lediglich die Beschaffung von Materialien prognosebasiert durchgeführt, während Fertigung, Montage und Versand erst nach Eingang eines Kundenauftrags begonnen werden. Diese Bevorratungsstrategie ist meist notwendig, wenn es aufgrund einer sehr hohen Variantenzahl oder hohen Kosten für Komponenten nicht wirtschaftlich ist, alle Komponenten für alle möglichen Kundenwünsche vorzuproduzieren. Dies ist beispielsweise der Fall, wenn Kraftfahrzeuge oder Computer kundenindividuell mit Komponenten konfiguriert werden, die nicht dem Standard entsprechen (vgl. Wiendahl, 2014; Meyr et al., 2015).

Bei der *auftragsbezogenen Einmalfertigung* (ETO, engl. engineer to order) ist eine Neukonstruktion eines Produktes erforderlich und somit können auch Materialien erst nach Eingang eines Kundenauftrags und der Planung des Produktes beschafft werden. Beispielsweise im Anlagenbau der Fall werden spezielle Anlagen, wie Pressmaschinen oder Wasserturbinen, individuell passend für einen Kunden konstruiert und produziert (vgl. Wiendahl, 2014).

Die Wichtigkeit von Nachfrageprognosen und somit auch die Bedeutung der Absatzplanung hängen von der gewählten Bevorratungsstrategie für ein Produkt ab. Während Nachfrageprognosen für Produkte in einer auftragsbezogenen Einmalfertigung lediglich einen geringen Stellenwert

besitzen, wächst die Bedeutung der Prognosen, je näher der Kundenauftragsentkopplungspunkt am Kunden ist. Für Produkte, die auftragsbezogen gefertigt oder montiert werden, müssen Teilkomponenten des finalen Produktes prognosebasiert beschafft oder beschafft und gefertigt werden. Die größte Bedeutung haben Nachfrageprognosen im Fall einer Produktion auf Lager, da sie hierbei die Basis für die Planung der Beschaffung, der Fertigung und der Montage bilden.

2.2.3 Prozess der Absatzplanung

Um auf zunehmend globalen Märkten bestehen zu können, konzentrieren sich viele Unternehmen auf ihre Kernkompetenzen, indem sie unrentable Aufgaben an externe Dienstleister vergeben und Teilkomponenten ihrer Produkte von Zulieferern beziehen (vgl. Pfohl, 2018). Hierdurch entstehen logistische *Lieferketten* (engl. supply chains, vgl. Chopra und Meindl, 2015). In solchen Lieferketten haben produzierende Unternehmen *stromaufwärts* (engl. upstream) Zulieferer für bestimmte Komponenten und die Zulieferer haben möglicherweise selbst Zulieferer für bestimmte Teilkomponenten. *Stromabwärts* (engl. downstream) haben die produzierenden Unternehmen Kunden, die möglicherweise selbst Kunden auf der nächsten Ebene haben. Hierzu sei angemerkt, dass die meisten Lieferketten als *Liefernetzwerke* (engl. supply networks) angesehen werden können, da häufig Produkte von mehreren Zulieferern bezogen werden und die fertigen Produkte an mehrere Kunden verkauft werden. Zudem befinden sich Unternehmen, die unterschiedliche Produkte herstellen, in der Regel in mehreren Lieferketten, bzw. in mehreren Liefernetzwerken. Aus Gründen der besseren Übersichtlichkeit werden in dieser Arbeit stets Lieferketten betrachtet, die Untersuchungen lassen sich aber auf Liefernetzwerke übertragen. Abbildung 2.3 zeigt eine exemplarische Lieferkette aus der Konsumgüterindustrie. Hierbei bezieht ein produzierendes Unternehmen Komponenten von Zulieferern und beliefert Großmärkte, von denen Einzelhändler die Produkte beziehen, die diese letztlich an Endkunden verkaufen (vgl. Syntetos et al., 2016). Während das Material stromabwärts in der Lieferkette fließt, erfolgen die Kundenbestellungen stromaufwärts. Um die Kundenbedarfe zu erfüllen, muss ein Unternehmen die Kundennachfrage

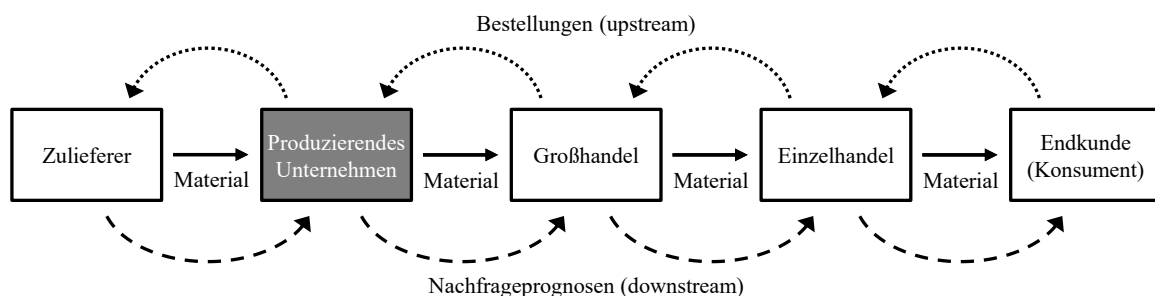


Abbildung 2.3: Exemplarische Supply Chain in der Konsumgüterindustrie in Anlehnung an Syntetos et al. (2016)

prognostizieren. Dementsprechend sind die Absatzprognosen produzierender Unternehmen, welche die Grundlage für deren Produktionsplanung und -steuerung (PPS) bilden, von signifikanter Bedeutung für die gesamte Lieferkette.

Innerbetrieblich basieren diverse wichtige Entscheidungsprozesse auf Absatzprognosen (vgl. Jain und Malehorn, 2012; Petropoulos et al., 2014). Während langfristige Prognosen mit einem Horizont mehrerer Jahre für die strategische Netzwerkplanung benötigt werden (vgl. Fleischmann et al., 2015), bildet die mittelfristige *Absatzplanung* (engl. demand planning, vgl. Jacobs et al., 2011; Chopra und Meindl, 2015; Kilger und Wagner, 2015), die Grundlage für die *Masterplanung* (engl. master planning, vgl. Abbildung 2.4 und Rohde et al., 2000; Fleischmann et al., 2015; Meyr et al., 2015). Diese koordiniert die mittelfristigen Beschaffungs-, Produktions- und Distributionsprozesse (vgl. Albrecht et al., 2015; Meyr et al., 2015) und ist somit eng verwandt mit der *Absatz und Betriebsplanung* (engl. sales and operations planning), welche die mittelfristigen Planungsaufgaben integriert, um Angebot und Nachfrage zu balancieren (vgl. Feng et al., 2008; Thomé et al., 2012; Kreuter et al., 2021). Die Masterplanung beinhaltet die *Produktionsprogrammplanung* (engl. master production scheduling), welche meist monatlich basierend auf Prognosen und vorliegenden Kundenaufträgen unter Berücksichtigung vorhandener Kapazitäten den Primärbedarf nach Produkten bestimmt (vgl. Wiendahl, 2014). Als Input für die Masterplanung werden die in der mittelfristigen Absatzplanung ermittelten Prognosen nach Produktgruppen, Verkaufsregionen sowie Produktionseigenschaften, wie bevorzugten Ressourcen oder benötigten Rüstzeiten, aggregiert. Diese Prognosen werden in der Regel auf wöchentlicher bis monatlicher

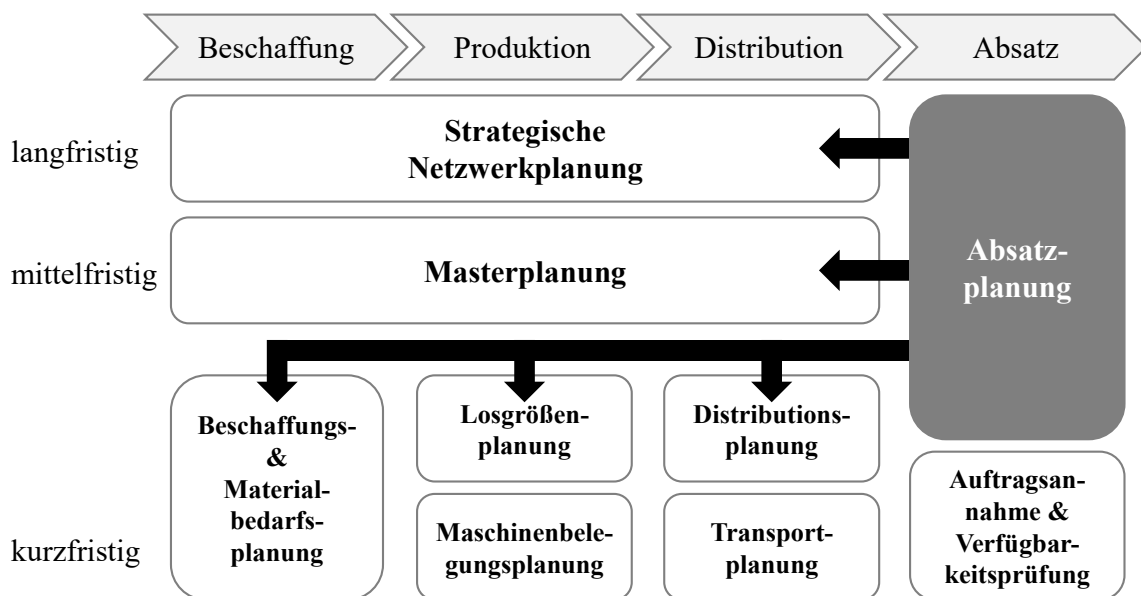


Abbildung 2.4: Einflüsse der Absatzplanung auf andere Planungsprozesse in der Supply Chain Planning Matrix in Anlehnung an Rohde et al. (2000) und Meyr et al. (2015)

Basis mit einem Horizont von ungefähr einem Jahr ermittelt (vgl. Fleischmann et al., 2015). Des Weiteren werden im Rahmen der Absatzplanung in kurzfristigen Abständen von einzelnen Tagen bis zu einer Woche detailliertere Absatzprognosen ermittelt, die zusätzlich zu den in der Masterplanung ermittelten Werten als Grundlage für die Ausführung der Beschaffungs- und Materialbedarfsplanung, der Losgrößenplanung sowie der Distributionsplanung dienen (vgl. Reuter und Rohde, 2015). Ungenaue Prognosen können negative Effekte für Unternehmen implizieren. Beispielsweise kann eine Prognose zu hoher Kundennachfrage zu hohen Lagerkosten und später zu einer Unterauslastung der Kapazitäten führen. Eine Prognose zu geringer Absatzmengen kann Überlastungen der Produktionskapazitäten, Lieferverzögerungen oder Fremdvergaben von Aufträgen zur Folge haben. Im Gegensatz dazu können sich exakte Nachfrageprognosen positiv auf die folgenden Planungsschritte auswirken und die Wirtschaftlichkeit der Produktion verbessern. Somit ist es für alle Unternehmen einer Lieferkette und insbesondere für produzierende Unternehmen erstrebenswert, die zukünftige Kundennachfrage möglichst exakt vorherzusagen.

Im Allgemeinen basiert die Erzeugung von Prognosen in der Absatzplanung auf historischen Daten, die meist in Form von Zeitreihen vorliegen (vgl. Tempelmeier, 2008; Kilger und Wagner, 2015). Aufgrund einer Vielzahl von Einflussfaktoren (vgl. Abschnitt 2.3.2), über die in der Regel kein belastbares Datenmaterial vorliegt, stellt die Prognose häufig eine schwierige Aufgabe dar. Um die dadurch entstehenden, oftmals komplex dynamischen Nachfrageverläufe vorherzusagen, besteht ein typischer Prognoseprozess aus den folgenden Schritten: (i) Datenaufnahme und Vorverarbeitung, (ii) statistische Prognose, (iii) Experten Anpassung, (iv) Konsensbildung, (v) Freigabe der Prognose sowie (vi) Evaluation und Monitoring des Outputs (vgl. Abbildung 2.5 und Kilger und Wagner, 2015; Petropoulos et al., 2016; Syntetos et al., 2016). Hierbei ist insbesondere der Schritt der statistischen Prognose von spezieller Bedeutung, bei dem Zeitreihen durch automatische Prognoseverfahren extrapoliert werden. Die daraus entstehenden Prognosen bilden die Grundlage für die folgenden Planungsschritte. Nach der Berechnung einer statistischen Prognose werden in Unternehmen Experten Anpassungen durchgeführt. Da aktuelle Studien jedoch nahelegen (vgl. Trapero et al., 2013; Petropoulos et al., 2016), dass Experten Anpassungen aufgrund systematischer Verzerrungen, wie z.B. einem *Überoptimismus* (engl. *optimism bias*), nicht notwendigerweise zu verbesserter Prognosegüte führen, werden dieser und die darauffolgenden Prozessschritte in der vorliegenden Arbeit nicht betrachtet. Stattdessen wird der Schritt der statistischen Prognose fokussiert.

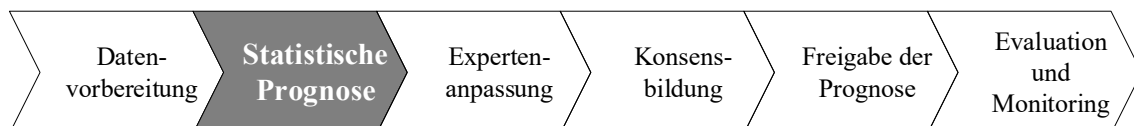


Abbildung 2.5: Typischer Prognoseprozess in Anlehnung an Kilger und Wagner (2015)

Im Zuge der Absatzplanung werden Prognosen für verschiedene Prognosehorizonte, Aggregations Ebenen (individuell für jede *Bestandseinheit* (SKU, engl. stock keeping unit) oder aggregiert für ganze Produktklassen) sowie geographische Regionen ermittelt (vgl. Kilger und Wagner, 2015). Zudem produzieren viele Unternehmen diverse Produktvarianten innerhalb verschiedener Produktklassen (vgl. ElMaraghy et al., 2013). Da somit in der Regel eine große Anzahl von Prognosen in regelmäßigen Zeitabständen erzeugt werden muss, fehlt im Allgemeinen die Zeit, jede Prognose einzeln zu ermitteln. Stattdessen werden automatische Prognoseverfahren zur Ermittlung statistischer Prognosen benötigt (vgl. Kapitel 3).

2.3 Modellierung der Kundennachfrage

Diese Arbeit beschreibt die Bedarfsplanung eines Kunden eines produzierenden Unternehmens als nichtlineares dynamisches System. Im Folgenden werden zunächst allgemeine Begriffe der Systemtheorie erläutert. Darauf folgend werden die Einflüsse auf die Kundennachfrage beschrieben.

2.3.1 Grundlagen der Systemtheorie

Abbildung 2.6 visualisiert die allgemeinen Begriffe der Systemtheorie nach Wiendahl (2014). Demnach lässt sich ein System wie folgt definieren (Wiendahl, 2014, S.22): „Ein *System* besteht aus einer Menge von Elementen und einer Menge von Beziehungen (Relationen), die zwischen diesen Elementen herrschen. Eine *Struktur* besteht aus einer Gruppe von Elementen, die über Beziehungen verknüpft sind. Jedes System kann in *Subsysteme* niedriger Ordnung zerlegt werden. Elemente, die nach einer anderen Beziehung zusammengefasst werden, heißen *Teilsysteme*. Jedes Element, jedes Subsystem und das gesamte System kann mit einer *Systemgrenze* abgegrenzt und durch *Input* (Eingang), *Output* (Ausgang) und eine *Funktion* beschrieben werden (sogenannte Black-Box-Darstellung). Wenn das System als Ganzes keine Eingangs- und Ausgangsgrößen hat, spricht man von einem geschlossenen System. Sonst handelt es sich um offene Systeme.“

2.3.2 Bedarfsplanung eines Kunden

Die Nachfrage eines Kunden nach einem Produkt eines produzierenden Unternehmens wird im Rahmen der *Materialbedarfsplanung* (engl. material requirements planning) des Kunden bestimmt. Der Kunde plant unter Berücksichtigung von Beständen den Bedarf an Eigenfertigungsteilen und Zukaufteilen basierend auf der in den Stücklisten enthaltenen Komponenten (vgl. Wiendahl, 2014). Im Allgemeinen wird der *Primärbedarf* an verkaufsfähigen Endprodukten ermittelt und aus diesem werden der *Sekundärbedarf* an Rohstoffen und Halbzeugen sowie der *Tertiärbedarf* an Betriebs- und Hilfsstoffen mit pauschalen Aufschlägen abgeleitet (vgl. Abbildung 2.7). Es lassen sich deterministische, stochastische oder heuristische Bedarfsermittlungsverfahren einsetzen. Zur sinnvollen Auswahl eines dieser Verfahren für eine bestimmte Komponente, werden in der Regel zunächst alle Komponenten in einer ABC-Analyse hinsichtlich ihrer Wertigkeit sowie in

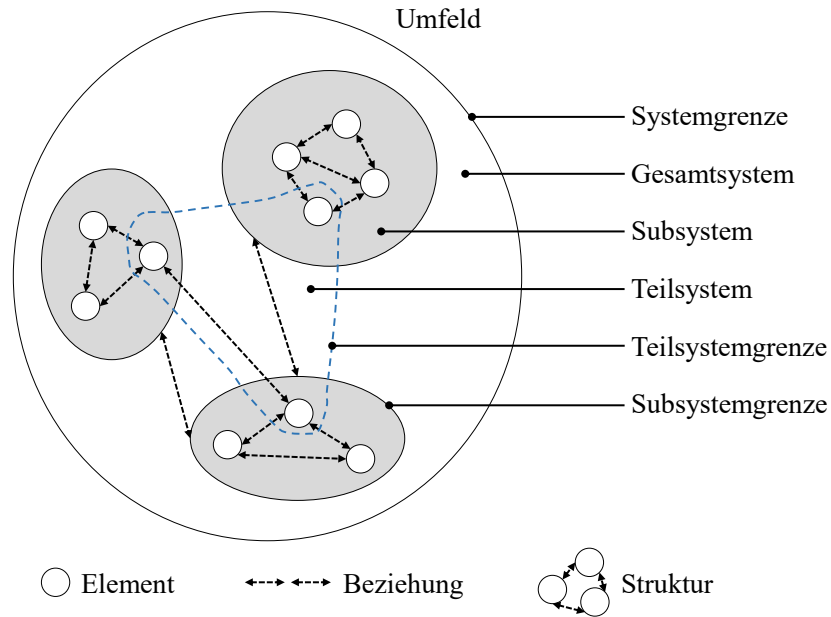


Abbildung 2.6: Systemdefinition in Anlehnung an Wiendahl (2014)

einer XYZ-Analyse hinsichtlich des Bedarfsverlaufsmusters klassifiziert (vgl. Wiendahl, 2014). Basierend auf dieser Einteilung werden beispielsweise die Bedarfe an hochwertigen Komponenten mit konstantem oder trendmäßigem Bedarfsverlauf in der Regel deterministisch basierend auf Stücklisten ermittelt. Hingegen werden Bedarfe an Komponenten mit niedrigem Wert meist stochastisch durch Prognoseverfahren ermittelt und die Mengen für bedarfsbezogene Einmalbeschaffungen lassen sich oft nur durch Heuristiken schätzen. Sobald ein Kunde die Bedarfe für alle

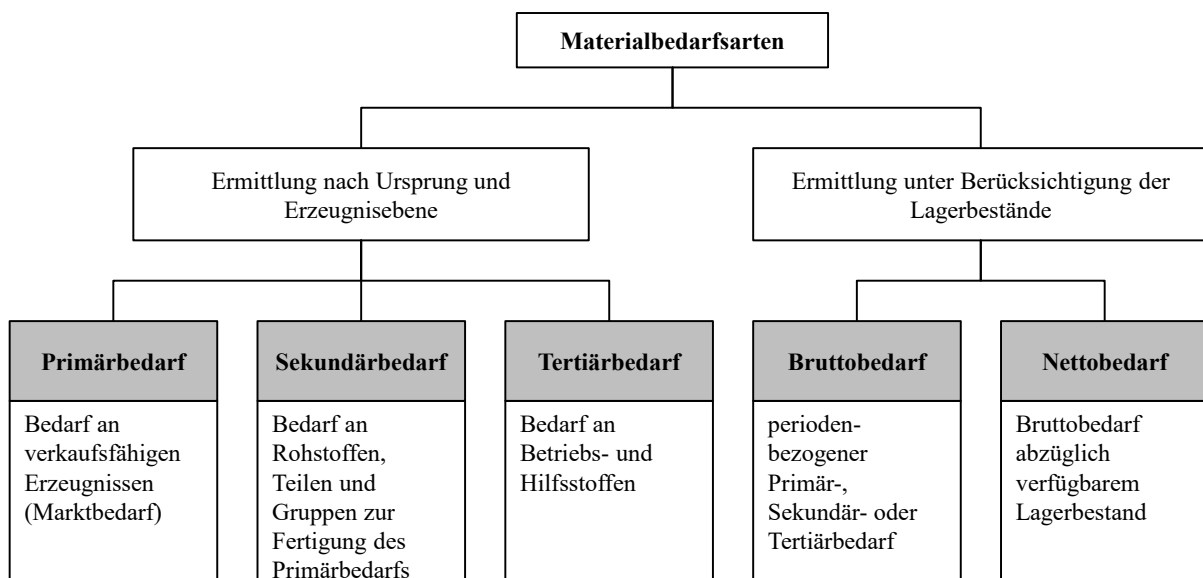


Abbildung 2.7: Materialbedarfsarten in Anlehnung an Hartmann (2002) und Wiendahl (2014)

notwendigen Komponenten ermittelt hat, wird entschieden, welche dieser Komponenten selbst gefertigt werden und welche von Zulieferern beschafft werden.

Die Bedarfsplanung lässt sich als offenes Subsystem des Gesamtsystems des Kunden betrachten (vgl. Abbildung 2.8). Ein Output der Bedarfsplanung ist die Nachfrage des Kunden nach einem Produkt des produzierenden Unternehmens. Diese Nachfrage wird sowohl durch externe Inputs als auch durch interne Elemente des Systems des Kunden beeinflusst. Externe Inputs, die die Nachfrage des Kunden beeinflussen, können beispielsweise die Anzahl verfügbarer Konkurrenzprodukte auf dem Markt, der Erfolg von Marketingkampagnen des produzierenden Unternehmens und der Konkurrenten, die aktuelle Kundenzufriedenheit für das nachgefragte Produkt des Unternehmens und für Konkurrenzprodukte, abgeschlossene Rahmenverträge sowie aktuelle ökonomische, politische, ökologische und soziale Rahmenbedingungen am Markt sein (vgl. Wiendahl et al., 2007; Porter, 2008). Interne Einflussgrößen für die Kundennachfrage sind beispielsweise der Absatzplan und das geplante Produktionsprogramm des Kunden sowie der aktuelle Lagerbestand zum nachgefragten Produkt.

Der Zweck der Absatzplanung, welche ein Subsystem des Gesamtsystems des produzierenden Unternehmens darstellt, liegt in der Prognose der zukünftigen Kundennachfrage. In diesem Zusammenhang ist es leicht ersichtlich, dass dem produzierenden Unternehmen in der Regel nicht

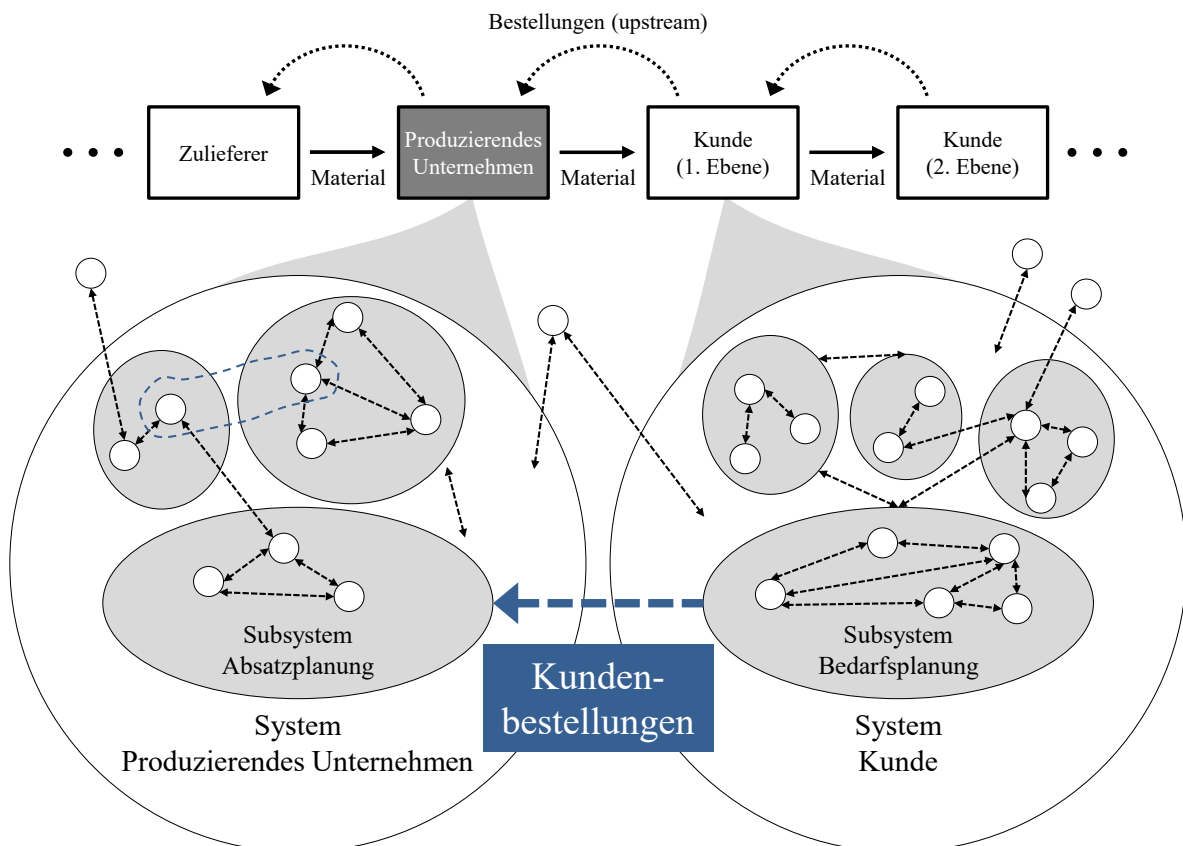


Abbildung 2.8: Kundennachfrage als Output der Bedarfsplanung des Kunden

alle externen und internen Einflussgrößen des Kunden bekannt sind und, dass die meisten der bekannten Einflussgrößen im Allgemeinen für das produzierende Unternehmen nicht messbar sind. Daher ist eine Nachfrageprognose stets mit Unsicherheit verbunden. Im folgenden Abschnitt werden die Grundannahmen dieser Arbeit hinsichtlich der Messbarkeit von Einflussgrößen auf die Nachfrage eines Kunden beschrieben.

2.3.3 Modellierung der Kundennachfrage in einem dynamischen System

Im Folgenden wird in dieser Arbeit stets die Nachfrage eines Kunden nach Produkten eines produzierenden Unternehmens betrachtet. Die Methoden und Untersuchungen sind dabei allgemeingültig und unabhängig davon, ob das Unternehmen fertige Produkte, Halbzeuge, Werkstoffe, Rohstoffe, Hilfsstoffe oder Betriebsstoffe produziert. Zudem ist es für die Untersuchungen unerheblich, ob der Kunde des Unternehmens wieder ein produzierendes Unternehmen, ein Handelsunternehmen, wie ein Groß- oder Einzelhandel, oder ein Endkunde ist. Es wird angenommen, dass das Gesamtsystem eines Kunden eines produzierenden Unternehmens sich als ein *nichtlineares dynamisches System* (engl. nonlinear dynamical system) modellieren lässt (vgl. Kantz und Schreiber, 2004; Radons und Neugebauer, 2004; Freitag, 2005; Papakostas et al., 2009; Argyris et al., 2017). Ferner wird die Bedarfsplanung des Kunden als Subsystem des Gesamtsystems des Kunden ebenfalls als ein nichtlineares dynamisches System betrachtet. Die Entwicklung eines dynamischen Systems wird durch ein System gewöhnlicher Differentialgleichungen beschrieben:

$$\frac{d}{dt}\boldsymbol{\psi}(t) = \boldsymbol{\varphi}(\boldsymbol{\psi}(t)), \quad \boldsymbol{\psi} = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{Y'} \end{pmatrix}, \quad \boldsymbol{\varphi} = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_{Y'} \end{pmatrix}, \quad t \in \mathbb{N}. \quad (2.1)$$

Der *Zustandsvektor* (engl. state vector) $\boldsymbol{\psi}(t)$ beschreibt den Zustand des dynamischen Systems zum Zeitpunkt t . Die Elemente $\psi_1, \dots, \psi_{Y'}$ des Zustandsvektors geben die Zustände aller Einflussgrößen des Systems an. Die Geschwindigkeit der Veränderung des Systemzustands über die Zeit wird durch das Vektorfeld $\boldsymbol{\varphi}$ beschrieben. Der Y' -dimensionale Raum, der durch die Komponenten von $\boldsymbol{\psi}$ multipliziert mit den Einheitsvektoren aufgespannt wird, wird als *Phasenraum* (engl. phase space) $\mathcal{Y}' \subseteq \mathbb{R}^{Y'}$ bezeichnet. Ein Punkt $\boldsymbol{\psi}$ im Phasenraum \mathcal{Y}' repräsentiert einen Zustand des dynamischen Systems zu einem bestimmten Zeitpunkt.

Unter der Annahme, dass $\boldsymbol{\varphi}$ Lipschitz-stetig ist, existiert eine Lösung des Anfangswertproblems (2.1) mit Initialbedingung $\boldsymbol{\psi}(0)$ und ist eindeutig durch den Zustandsvektor $\boldsymbol{\psi}$ gegeben. In diesem Fall lässt sich der Zustand des dynamischen Systems zu jedem Zeitpunkt t eindeutig beschreiben, falls (i) die Komponenten des Vektorfelds $\boldsymbol{\varphi}$ bekannt sind, (ii) alle Einflussgrößen des dynamischen Systems bekannt sind und (iii) die Zustände $\psi_1, \dots, \psi_{Y'}$ aller Einflussgrößen messbar sind. Allerdings ist im Allgemeinen keine dieser drei Annahmen zutreffend. Für den Fall, der Betrachtung des dynamischen Systems der Bedarfsplanung eines Kunden ist das Vektorfeld $\boldsymbol{\varphi}$ in der Regel unbekannt, es sind zumeist nur wenige Einflussgrößen des Systems bekannt und

es lassen sich nur die Zustände einer Teilmenge der bekannten Einflussgrößen messen. In der vorliegenden Arbeit wird lediglich die Verfügbarkeit einer sukzessiven, äquidistanten, skalaren Zeitreihe vergangener Kundenbestellungen

$$\mathbf{y} = \{y_1, y_2, \dots, y_T\} \in \mathcal{Y}, \quad (2.2)$$

bis zum gegenwärtigen Zeitpunkt T vorausgesetzt. Die Nachfrageentwicklung $\psi_i(t) = y_t$ ist eine Komponente des dynamischen Systems, welches die Bedarfsplanung des Kunden beschreibt und wiederum ein Subsystem des Gesamtsystems der Prozesse des Kunden ist. \mathcal{Y} bezeichnet den Raum der Zeitreihen vergangener Kundenbestellungen. Da dem produzierenden Unternehmen für die Absatzplanung somit lediglich Messungen anhand einer Komponente des möglicherweise hochdimensionalen dynamischen Systems der Bedarfsplanung des Kunden zur Verfügung stehen, müssen Abbildungen gefunden werden, die dennoch die Entwicklung des Systems über die Zeit möglichst gut beschreiben. Eine Möglichkeit zur Entwicklung solcher Abbildungen, sogenannter statistischer Modelle, ist das überwachte Lernen, welches im nächsten Kapitel beschrieben wird. In der vorliegenden Arbeit werden lediglich Zeitreihen betrachtet, die *regelmäßige Nachfrageverläufe* zeigen und somit in jeder Periode eine positive Nachfrage aufweisen. Für Anwendungen hinsichtlich *sporadischer Nachfrageverläufe* (engl. intermittent demand), welche Perioden ohne Nachfrage enthalten und beispielsweise häufig für Ersatzteile zu beobachten sind, siehe (Syntetos und Boylan, 2001; Wallström und Segerstedt, 2010; Kourentzes, 2014; Van Wingerden et al., 2014).

2.4 Zusammenfassung

In diesem Kapitel wurden die Grundlagen der Absatzplanung sowie die Annahmen dargestellt, auf denen die Untersuchungen dieser Arbeit beruhen. Die Absatzplanung ist von großer Bedeutung für die Wirtschaftlichkeit eines produzierenden Unternehmens, da sie die Basis für diverse Planungsprozesse, wie die Produktionsprogrammplanung sowie die Materialbedarfsplanung bildet. Nachfrageprognosen sind umso wichtiger, je mehr Komponenten oder fertige Produkte unabhängig von konkreten Kundenaufträgen beschafft und gefertigt werden müssen, was insbesondere häufig in der Konsumgüterindustrie sowie im Handel der Fall ist. Die Nachfrage eines Kunden nach einem Produkt eines produzierenden Unternehmens wird im Rahmen der Materialbedarfsplanung des Kunden bestimmt, welche ein Subsystem des Gesamtsystems des Kunden darstellt. Das Gesamtsystem des Kunden lässt sich als ein nichtlineares dynamisches System mit diversen internen und externen Einflussgrößen modellieren. Da die meisten dieser Einflüsse auf den Kunden für ein produzierendes Unternehmen jedoch nicht messbar sind, wird in der vorliegenden Arbeit lediglich die Verfügbarkeit einer sukzessiven, äquidistanten, skalaren Zeitreihe vergangener Kundenbestellungen angenommen. Aufgrund der häufig großen Anzahl an Produkten, für die Nachfrageprognosen in regelmäßigen Abständen ermittelt werden müssen, ist die Verwendung automatischer Prognoseverfahren notwendig. Zu diesem Zweck werden im folgenden Kapitel die Grundlagen der Zeitreihenprognose dargestellt und verschiedene Modellklassen erläutert.

Kapitel 3

Grundlagen und Stand der Forschung zur Zeitreihenprognose

„Essentially, all models are wrong, but some are useful.“

George E. P. Box, britischer Statistiker (1919 - 2013)

In diesem Kapitel werden die Grundlagen und der Stand der Forschung zur Zeitreihenprognose beschrieben. Zeitreihenprognose lässt sich als ein Spezialfall des überwachten Lernens betrachten. Im Folgenden werden zunächst die Grundlagen des überwachten Lernens beschrieben. Darauffolgend werden Zeitreihenprognoseprobleme sowie Kriterien zur Evaluation von Zeitreihenprognosen für einzelne und mehrere Zeitreihen erläutert. Im dritten Abschnitt wird der Stand der Forschung hinsichtlich gängiger Modellklassen dargestellt, wobei die Hauptergebnisse der beiden aktuellsten großen Vergleichsstudien von Prognoseverfahren als Grundlage dienen. Als Ergebnis lässt sich FT 1 bestätigen. Darauffolgend werden die in dieser Arbeit verwendeten Prognoseverfahrensklassen beschrieben. Das Kapitel endet mit einer kurzen Zusammenfassung der Erkenntnisse.

3.1 Überwachtes Lernen

3.1.1 Übersicht und Einordnung

Überwachtes Lernen ist ein Teilgebiet des maschinellen Lernens, welches wiederum der Schnittmenge aus Data Science und künstlicher Intelligenz zugeordnet werden kann (vgl. Abbildung 3.1). Im Folgenden werden diese Begriffe beschrieben und die theoretischen Grundlagen des überwachten Lernens erläutert.

Data Science

Data Science wird als die „Wissenschaft des Lernens aus Daten“ (Donoho, 2017) bezeichnet und umfasst die folgenden Teilgebiete: (i) Datensammlung, -vorverarbeitung und -exploration, (ii) Datenrepräsentation und -transformation, (iii) Datenverarbeitung, (iv) Visualisierung und (v)

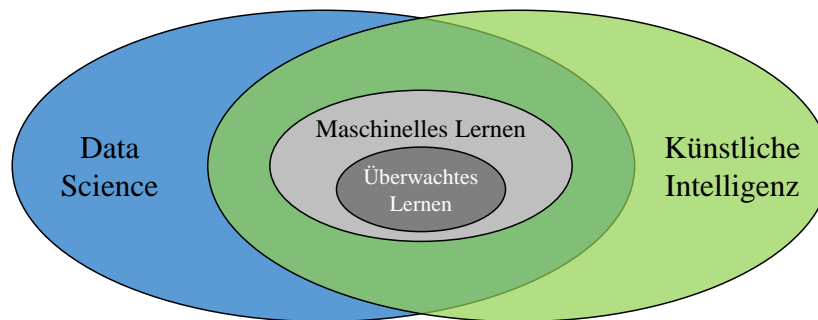


Abbildung 3.1: Mengendiagramm zur Einordnung der Gebiete Data Science, künstliche Intelligenz, maschinelles Lernen und überwachtes Lernen

Modellierung. Data Science lässt sich als Schnittmenge aus Informatik, mathematischer Statistik und einem spezifischen Anwendungsgebiet verstehen (vgl. Schutt und O’Neil, 2013; Freitag et al., 2015b). Basierend auf einer Datenmenge des Anwendungsgebietes ist das Ziel eines Data-Science-Projektes in der Regel die Entwicklung eines Modells für einen bestimmten Zweck. Ein *Modell* lässt sich als eine vereinfachte Repräsentation eines Systems verstehen (vgl. Banks et al., 2013), wobei hinsichtlich des Zwecks der Modellierung drei verschiedene Modelltypen unterschieden werden: (i) erklärende Modelle, die zur Beschreibung und Erklärung eines beobachteten Systemverhaltens dienen, (ii) prädiktive Modelle, die zur Prognose zukünftiger Entwicklungen eines Systems verwendet werden, und (iii) präskriptive Modelle, die zur Auswahl sinnvoller Entscheidungen verwendet werden können, um das Systemverhalten in eine gewünschte Richtung zu beeinflussen (vgl. Shmueli, 2010; Lepenioti et al., 2020).

Künstliche Intelligenz

Künstliche Intelligenz (engl. artificial intelligence) wird definiert als die „Lehre von Agenten, die Wahrnehmungen aus der Umgebung erhalten und Aktionen ausführen“ (Russell et al., 2021). Die Auswahl von Aktionen basiert auf einer *Agentenfunktion* (engl. agent function), die jede mögliche Wahrnehmungssequenz auf eine Aktion abbildet und als Intelligenz des Agenten betrachtet werden kann. Russell et al. (2021) unterscheiden fünf Arten von Agenten: (i) einfache Reflexagenten, die simple Wenn-dann-Regeln nutzen, um Aktionen auszuwählen, (ii) modellbasierte Reflexagenten, die mit Hilfe eines Modells den aktuellen Systemzustand beschreiben und basierend darauf Aktionen auswählen, (iii) zielbasierte Agenten, die durch eine Suche oder eine Planung Aktionen auswählen, die zum Ziel führen, (iv) nutzenbasierte Agenten, die Aktionen auswählen, welche entweder die übergeordnete Zielfunktion oder eine interne Nutzenfunktion maximieren und (v) lernende Agenten, die zusätzlich zur Wahrnehmung ihrer Umgebung auch Bewertungen ihrer ausgeführten Aktionen wahrnehmen und die Agentenfunktion basierend auf diesen Bewertungen anpassen, um zukünftig bessere Aktionen auszuwählen. Die Agentenfunktion kann von einem Menschen mit Expertenwissen definiert werden oder automatisch aus Daten gelernt werden.

Maschinelles Lernen

Die Schnittmenge zwischen Data Science und künstlicher Intelligenz beinhaltet die Entwicklung von Agenten basierend auf beobachteten Daten. Während diese Aufgabe klassischerweise von Experten durchgeführt wurde, werden heutzutage häufig maschinelle Lernverfahren eingesetzt, um auf Basis historischer Daten automatisch Modelle zu lernen, welche sich in eine Agentenfunktion einbetten lassen. *Maschinelles Lernen* (engl. machine learning) wird definiert als „die Generierung von Wissen aus Erfahrung, indem Lernalgorithmen aus Beispielen ein komplexes Modell entwickeln“ (Döbel et al., 2018). Ferner lassen sich maschinelle Lernverfahren als Methoden beschreiben, die „einem Computer dabei helfen, zu lernen, ohne explizit programmiert zu werden“ (Kaplan und Haenlein, 2019). In diesem Zusammenhang lässt sich *Lernen* (engl. learning) als der Prozess der Verbesserung der Leistungsfähigkeit durch Beobachtung eines Systems verstehen (vgl. Russell et al., 2021). Burkov (2019) unterscheidet vier Arten maschineller Lernverfahren: (i) überwachte, (ii) halbüberwachte, (iii) unüberwachte und (iv) bestärkende Lernverfahren. In der vorliegenden Arbeit werden überwachte Lernverfahren verwendet.

Überwachtes Lernen

Überwachtes Lernen (engl. supervised learning) bezeichnet die Verwendung von Lernalgorithmen, um basierend auf einer Trainingsmenge von Daten ein Modell zu bilden, welches für eine neue Datenmenge den Eingabewerten möglichst die korrekten Ausgabewerte zuordnen kann (vgl. Döbel et al., 2018; Burkov, 2019; Russell et al., 2021). Je nach Format der Ausgabewerte lassen sich Regressions- und Klassifikationsmodelle unterscheiden, wobei Zeitreihenprognose einen Spezialfall der Regression darstellt. Neben der Selektion eines einzelnen Modells kann in bestimmten Fällen die Kombination mehrerer Modelle sinnvoll sein, wobei jedes einzelne Modell entweder aus einer vorher spezifizierten Modellklasse selektiert werden kann oder aus mehreren verschiedenen Modellklassen. Während klassische Regressions- und Klassifikationsmodelle zur Erklärung oder Prognose numerischer Werte und von Klassenzugehörigkeiten verwendet werden, haben Meta-Modelle den Zweck der Ermittlung geeigneter Lernalgorithmen oder Modelle für bestimmte Lernprobleme. In den folgenden Abschnitten werden die genannten Konzepte im Detail erläutert.

3.1.2 Theoretische Grundlagen des überwachten Lernens

Überwachte Lernprobleme

Abbildung 3.2 gibt einen Überblick über die Prozesse des überwachten Lernens. Hierbei werden die Trainings- und die Einsatzphase eines Modells unterschieden. Analog zur Abbildung werden im Folgenden die theoretischen Grundlagen überwachter Lernprobleme beschrieben. Darauf folgend werden erklärende und prädiktive sowie Regressions- und Klassifikationsmodelle unterschieden.

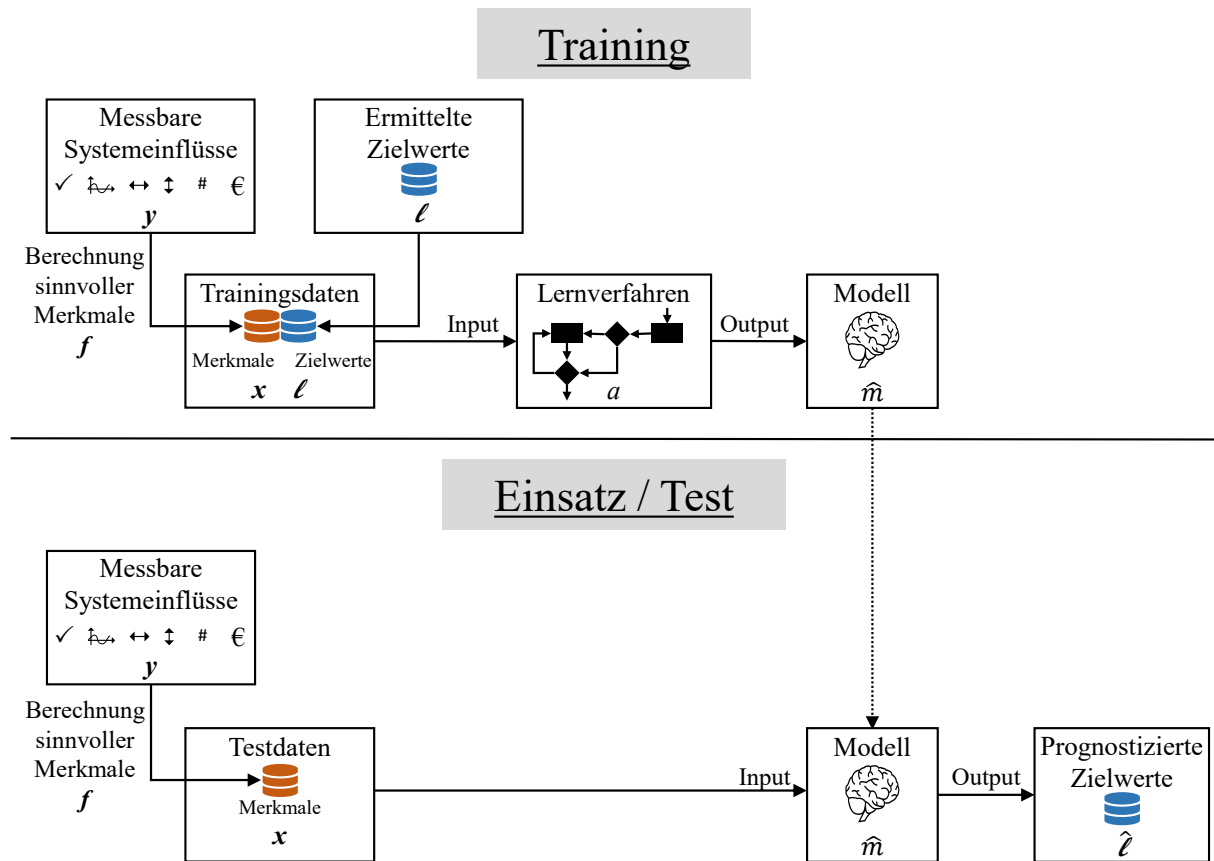


Abbildung 3.2: Training und Einsatz von Modellen beim überwachten Lernen

Sei $\mathbf{y} \in \mathcal{Y}$ ein Vektor aus dem Raum der messbaren Einflussgrößen eines Systems. Sei ferner

$$\mathbf{F} = \{f_1, f_2, \dots, f_F\}, \text{ mit } f_j \in \mathcal{F}, \quad f_j : \mathcal{Y} \longrightarrow \mathcal{X}, \quad j \in \{1, \dots, F\}, \quad (3.1)$$

eine Menge von Funktionen, die zu den gegebenen Messwerten der Einflussgrößen charakteristische Maßzahlen ermitteln. Die Funktionen f_j werden als *Merkmalsextraktionsfunktionen* (engl. feature extraction functions) bezeichnet und \mathcal{F} ist der Raum aller Merkmalsextraktionsfunktionen. Die ermittelten Vektoren charakteristischer Maßzahlen $\mathbf{x} \in \mathcal{X}$ werden *Merkmalsvektoren* (engl. feature vectors) oder auch *Eingabevektoren* (engl. query vectors) genannt und die Vektorelemente x_1, \dots, x_F heißen *Merkmale* (engl. features). Der Raum der Merkmalsvektoren $\mathcal{X} \subseteq \mathbb{R}^F$ wird als *Merkmalsraum* (engl. feature space) bezeichnet. Merkmale können entweder direkt anhand der internen und externen Einflussgrößen eines Systems gemessen werden oder sie können aus anderen Messwerten ermittelt werden. Sie dienen dazu, den Systemzustand zu beschreiben.

Zu jedem Eingabevektor $\mathbf{x} \in \mathcal{X}$, der einen spezifischen Systemzustand beschreibt, existiert ein dazugehöriger *Ausgabevektor* $\boldsymbol{\ell} \in \mathcal{L} \subseteq \mathbb{R}^C$, der den Output des Systems beschreibt. Die Elemente ℓ_1, \dots, ℓ_C des Ausgabevektors werden als *Zielwerte* (auch Labels, engl. target values) bezeichnet. Es wird angenommen, dass die Zielwerte zu jedem eintretenden Systemzustand eindeutig gemessen

werden können. Das Ziel des überwachten Lernens besteht darin, eine Abbildung zu finden, die die Zielwerte für einen bislang noch nicht eingetretenen Systemzustand vorhersagen kann.

Sei $\mathcal{D} = \mathcal{X} \times \mathcal{L}$ der Raum aller Datenpaare von Eingabe- und Ausgabevektoren, die das System (theoretisch) erzeugen könnte. Die Menge der bereits eingetretenen und gemessenen Datenpaare

$$\mathbf{D}_{\text{train}} = \left\{ (\mathbf{x}^1, \ell^1), (\mathbf{x}^2, \ell^2), \dots, (\mathbf{x}^{D_{\text{train}}}, \ell^{D_{\text{train}}}) \right\} \in \mathcal{D}^{D_{\text{train}}} \quad (3.2)$$

wird als *Trainingsmenge* (engl. training set) bezeichnet. Die Paare (\mathbf{x}^i, ℓ^i) von Eingabevektoren $\mathbf{x}^i \in \mathcal{X}$ und Ausgabevektoren $\ell^i \in \mathcal{L}$ werden *Trainingsbeispiele* (engl. training examples) genannt.

Ein *statistisches Modell* (im Folgenden als Modell bezeichnet)

$$\hat{m} \in \mathcal{M}, \quad \hat{m} : \mathcal{X} \longrightarrow \mathcal{L}, \quad \hat{m}(\mathbf{x}) = \hat{\ell} \quad (3.3)$$

ist eine Abbildung eines Merkmalsvektors $\mathbf{x} \in \mathcal{X}$ auf einen prognostizierten Ausgabevektor $\hat{\ell} \in \mathcal{L}$ eines wahren Ausgabevektors $\ell \in \mathcal{L}$. Dabei bezeichnet \mathcal{M} den *Raum aller Modelle*. Sei

$$m_{\text{true}} : \mathcal{X} \longrightarrow \mathcal{L}, \quad m_{\text{true}}(\mathbf{x}) = \ell \quad (3.4)$$

ein Modell, das jedem Eingabevektor \mathbf{x} eines Systems den korrekten Ausgabevektor ℓ zuordnet. m_{true} wird als das *wahre Modell* des Systems bezeichnet und ist im Allgemeinen unbekannt. Sei

$$e \in \mathcal{E}, \quad e : \mathcal{L} \times \mathcal{L} \longrightarrow \mathbb{R} \quad (3.5)$$

eine *Fehlerfunktion* (engl. error function) aus dem Raum der Fehlerfunktionen \mathcal{E} , welche die Güte eines Modells anhand seiner prognostizierten Ausgabevektoren im Vergleich zu den korrekten Ausgabevektoren misst. Hierbei wird eine perfekte Prognosegüte genau dann erreicht, wenn der prognostizierte Ausgabevektor mit dem wahren Ausgabevektor übereinstimmt:

$$e(\ell, \hat{\ell}) = 0 \iff \ell = \hat{\ell}. \quad (3.6)$$

Sei ferner

$$E_{\mathbf{D}} = \frac{1}{D} \sum_{(\mathbf{x}^i, \ell^i) \in \mathbf{D}} e(\ell^i, \hat{m}(\mathbf{x}^i)) \quad (3.7)$$

der *mittlere Fehler des Modells für die Datenmenge* $\mathbf{D} \in \mathcal{D}^D$ und sei

$$a \in \mathcal{A}, \quad a : \mathcal{D}^{D_{\text{train}}} \longrightarrow \mathcal{M}, \quad a(\mathbf{D}_{\text{train}}) = \hat{m} \quad (3.8)$$

ein überwachtes *Lernverfahren* (auch Lernalgorithmus, engl. learning algorithm) aus dem Raum der Lernverfahren \mathcal{A} . Dann lässt sich das folgende Problem formulieren:

Überwachtes Lernproblem. Sei $\mathbf{D}_{\text{train}}$ eine gegebene Trainingsmenge aus Paaren (\mathbf{x}^i, ℓ^i) von Eingabevektoren $\mathbf{x}^i \in \mathcal{X}$ und Ausgabevektoren $\ell^i \in \mathcal{L}$ eines Systems und sei $e : \mathcal{L} \times \mathcal{L} \longrightarrow \mathbb{R}$ eine geeignete Fehlerfunktion. Dann ist das Ziel eines überwachten Lernverfahrens $a \in \mathcal{A}$ die Entwicklung eines Modells $m^* : \mathcal{X} \longrightarrow \mathcal{L}$, welches den mittleren Fehler für alle Datenmengen $\mathbf{D} \in \mathcal{D}^D$ des Systems minimiert:

$$m^* = \underset{\hat{m} \in \mathcal{M}}{\operatorname{argmin}} E_{\mathbf{D}}. \quad (3.9)$$

Der Prozess der Verwendung eines Lernverfahrens zur Entwicklung eines Modells, welches zu einem gegebenen Eingabevektor den zugehörigen Ausgabevektor möglichst korrekt prognostiziert, wird als *Lernprozess* (auch Trainingsprozess, engl. training process) bezeichnet und die einzelnen Iterationsschritte ι werden *Lernepochen* genannt. Wie Abbildung 3.2 zeigt, lässt sich ein trainiertes Modell \hat{m} nach dem Lernprozess einsetzen, um zu gegebenen oder erdachten Messwerten der Systemeinflüsse $\mathbf{y} \in \mathcal{Y}$ den Output des Systems zu prognostizieren. Hierzu müssen die gleichen Merkmalsextraktionsfunktionen f_1, \dots, f_F verwendet werden, wie im Trainingsprozess, um den Merkmalsvektor $\mathbf{x} \in \mathcal{X}$ zu bestimmen. Dieser wird als Input für das trainierte Modell genutzt und die Prognose des Outputs wird als $\hat{m}(\mathbf{x}) = \hat{\ell}$ ermittelt.

Unterscheidung zwischen erklärenden und prädiktiven Modellen

Verschiedene überwachte Lernprobleme lassen sich hinsichtlich des Zwecks der Modellierung unterscheiden. Wenn die Menge \mathbf{D} in der Formulierung des überwachten Lernproblems (3.9) mit der Trainingsmenge $\mathbf{D}_{\text{train}}$ übereinstimmt, wird ein *erklärendes Modell* (engl. explanatory model) gesucht. Zur Entwicklung eines *prädiktiven Modells* (engl. predictive model) muss die Menge \mathbf{D} eine *Testmenge* (auch Generalisierungsmenge, engl. test set)

$$\mathbf{D}_{\text{test}} = \left\{ \left(\mathbf{x}^{D_{\text{train}}+1}, \ell^{D_{\text{train}}+1} \right), \dots, \left(\mathbf{x}^{D_{\text{train}}+D_{\text{test}}}, \ell^{D_{\text{train}}+D_{\text{test}}} \right) \right\} \in \mathcal{D}^{D_{\text{test}}} \quad (3.10)$$

sein, welche aus Paaren von Eingabe- und Ausgabevektoren besteht, die nicht in der Trainingsmenge $\mathbf{D}_{\text{train}}$ enthalten sind. Um die Fähigkeit eines Modells zur Prognose ungesehener Werte bereits während des Lernprozesses zu überprüfen, wird die Trainingsmenge in eine reduzierte Trainingsmenge und eine *Validierungsmenge* (engl. validation set) \mathbf{D}_{val} unterteilt. Dann wird nach der Vorschrift des Lernverfahrens a in jeder Lernepoche jeweils ein Modell anhand der Trainingsmenge gebildet und darauffolgend wird jeweils der mittlere Prognosefehler $E_{D_{\text{val}}}$ für die Validierungsmenge ermittelt, bis dieser Fehler eine akzeptable Genauigkeit erreicht hat oder ein anderes vorher spezifiziertes Abbruchkriterium erfüllt ist. Neben der Unterschreitung einer bestimmten Fehlergrenze sind weitere gängige Abbruchkriterien eine maximale Anzahl an Lernepochen oder ein Mindestmaß an Fehlerreduzierung über eine bestimmte Anzahl an Lernepochen. Abbildung 3.3 veranschaulicht, wie sich die optimale Trainingsepoche in einem Lernprozess bestimmen lässt.

Während für ein erklärendes Modell die Trainingsepoche mit *minimalem mittleren Trainingsfehler* $E_{D_{\text{train}}}^*$ gesucht wird, wird für ein prädiktives Modell die Lernepoche mit *minimalem mittleren Validierungsfehler* $E_{D_{\text{val}}}^*$ gewählt. Die intuitive Annahme, dass Modelle, die vorhandene Daten detailliert erklären, auch ungesehene Daten des gleichen Systems geeignet prognostizieren können, ist nicht notwendigerweise zutreffend. Der erwartete Fehler eines Modells bei der Prognose ungesehener Daten lässt sich in die drei Komponenten der quadrierten Verzerrung, der Varianz und des irreduziblen Fehlers zerlegen (vgl. Hastie et al., 2009). Die *Verzerrung* (engl. bias) beschreibt den durch falsche Modellannahmen bedingten Fehler, welcher beispielsweise

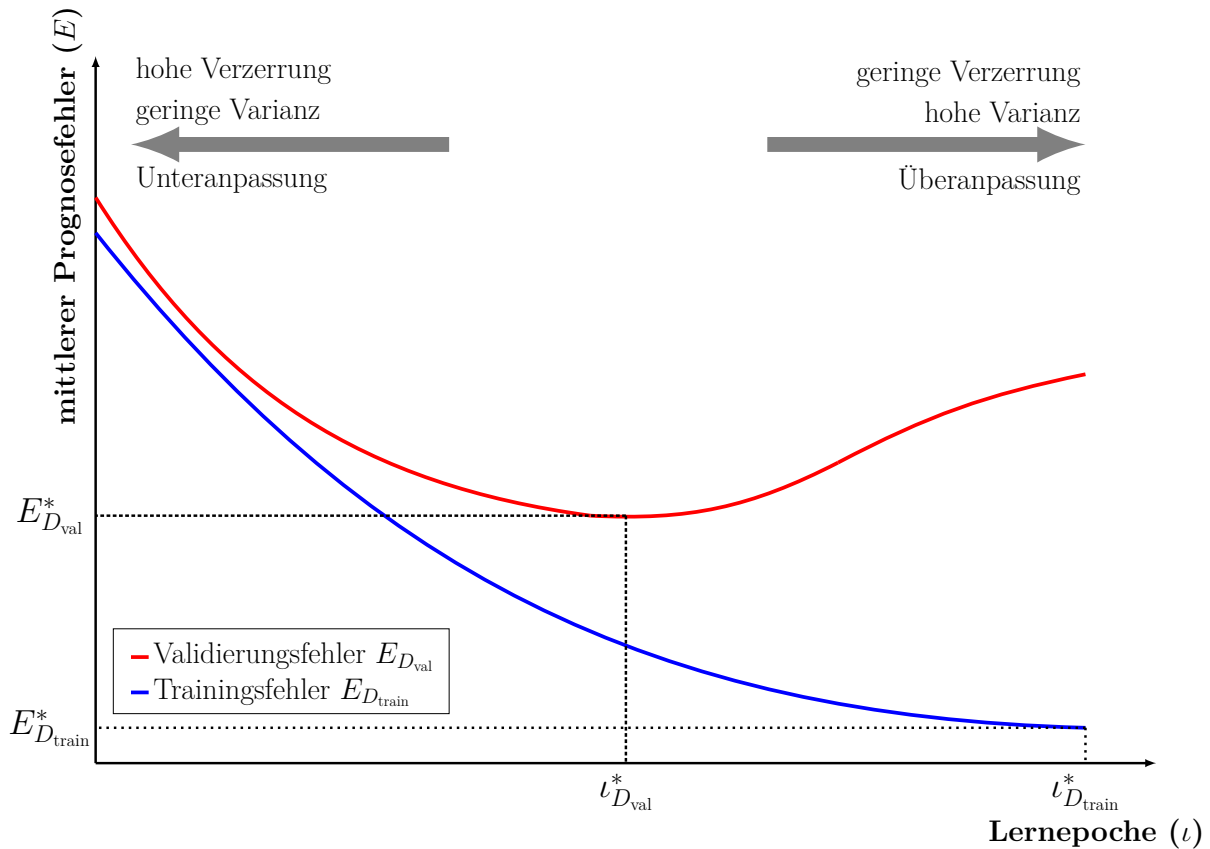


Abbildung 3.3: Bestimmung der optimalen Lernepoche für erklärende und prädiktive Modelle in Anlehnung an Hastie et al. (2009)

bei Annäherung eines nichtlinearen Prozesses durch ein lineares Modell entsteht. Die *Varianz* ist ein Maß für die Sensitivität des Modells gegenüber kleinen Schwankungen innerhalb der Trainingsdaten. Für eine gegebene Datenmenge lässt sich der *irreduzible Fehler* nicht beeinflussen. Dieser ist durch die Varianz des Rauschens innerhalb der Daten bedingt und er ist umso höher, je schlechter die Trainingsmenge das Verhalten des Gesamtsystems repräsentiert. Beeinflussbar sind die Fehlerkomponenten der Verzerrung und der Varianz, zwischen denen allerdings ein Zielkonflikt besteht, das sogenannte *Verzerrung-Varianz-Dilemma* (engl. bias-variance trade-off): In der Regel bedingt eine geringe Verzerrung eine hohe Varianz des Modells und umgekehrt. Eine gute Anpassung an Trainingsdaten lässt sich mit einem hochkomplexen Modell, wie beispielsweise einem hochdimensionalen Polynom erreichen. Dies führt zwar zu einer geringen Verzerrung, jedoch gleichzeitig zu einer hohen Varianz. Das Modell ist in der Lage, die Trainingsdaten abzubilden und somit zu erklären, lässt sich aber nur schlecht zur Prognose ungesehener Daten verwenden. Diese als *Überanpassung* (engl. overfitting) bezeichnete Modelleigenschaft liegt darin begründet, dass das komplexe Modell nicht nur die relevanten Zusammenhänge der Daten in die Modellbildung einbezieht, sondern auch irrelevantes Rauschen innerhalb der Daten. Auf der anderen Seite führt ein simpleres Modell, wie z.B. eine lineare Regression in der Regel

zu geringer Varianz, allerdings gleichzeitig zu einer hohen Verzerrung. In diesem Fall findet eine *Unteranpassung* (engl. underfitting) an die Daten statt. Sowohl eine Über- als auch eine Unteranpassung eines Modells führt zu einem hohen Prognosefehler für ungesehene Daten. Daher muss für ein prädiktives Modell ein geeigneter Kompromiss zwischen Verzerrung und Varianz bei der Modellbildung gefunden werden. Allerdings sollten Erklärung und Prognose nicht als zwei Extreme auf einer Achse betrachtet werden, sondern eher als zwei Dimensionen, denn häufig lassen sich erklärende Modelle auch zur Prognose und prädiktive Modelle auch zur Erklärung verwenden (vgl. Shmueli, 2010). Die besten Ergebnisse werden jedoch erzielt, wenn ein Modell für einen spezifischen Zweck gebildet wird. In der vorliegenden Arbeit werden prädiktive Modelle gebildet und verwendet.

Unterscheidung zwischen Regressions- und Klassifikationsproblemen

Zusätzlich zur Unterscheidung zwischen erklärenden und prädiktiven Modellen werden anhand des Formates der Elemente des Ausgabevektors ℓ eines überwachten Lernproblems Regressions- und Klassifikationsprobleme unterschieden. Wenn die Vektorelemente reelle Zahlen ohne weitere Einschränkung sind, spricht man von einem Regressionsproblem (vgl. Abschnitt 3.1.4). Wenn zusätzlich gilt, dass sie im Intervall $[0, 1]$ liegen und ihre Summe 1 ergibt, spricht man von einem Klassifikationsproblem (vgl. Abschnitt 3.1.3). Während das Ziel eines Regressionsproblems darin besteht, ein Modell zu finden, welches sich möglichst nah an vorhandene Datenpunkte anpasst, ist das Ziel eines Klassifikationsproblems ein Modell, welches Datenpunkte verschiedener Klassen möglichst gut trennt. Abbildung 3.4 visualisiert diese beiden Arten überwachter Lernprobleme und unterscheidet zusätzlich gut angepasste von über- und unterangepassten Modellen.

3.1.3 Klassifikation

Klassifikationsprobleme (vgl. Duda et al., 2001; Bishop, 2006; Hastie et al., 2009; Burkov, 2019) sind überwachte Lernprobleme (3.9), bei denen das Ziel in der Bildung eines *Klassifikationsmodells* (engl. classification model)

$$\hat{m} \in \mathcal{M}_{\text{Klas}}, \quad \hat{m} : \mathcal{X} \longrightarrow \mathcal{L} \quad (3.11)$$

aus dem Raum der Klassifikationsmodelle $\mathcal{M}_{\text{Klas}}$ besteht, welches von einem Eingabevektor $\mathbf{x}^i \in \mathbb{R}^F$ auf einen Ausgabevektor schließt, für den gilt:

$$\begin{aligned} \ell^i &= (\ell_1^i, \dots, \ell_C^i)^\top \in \mathcal{L} \subseteq \mathbb{R}^C \text{ mit } C \in \{c \in \mathbb{N} : 2 \leq C < \infty\}, \\ \forall c \in \{1, \dots, C\} : \ell_c^i &\in [0, 1] \text{ und } \sum_{c=1}^C \ell_c^i = 1. \end{aligned} \quad (3.12)$$

Die Elemente ℓ_c^i des Ausgabevektors geben die Wahrscheinlichkeit an, dass der durch \mathbf{x}^i beschriebene Systemzustand der c -ten von insgesamt C möglichen Klassen zugeordnet werden kann. Auf Basis eines Ausgabevektors wird der Systemzustand in der Regel der Klasse mit der

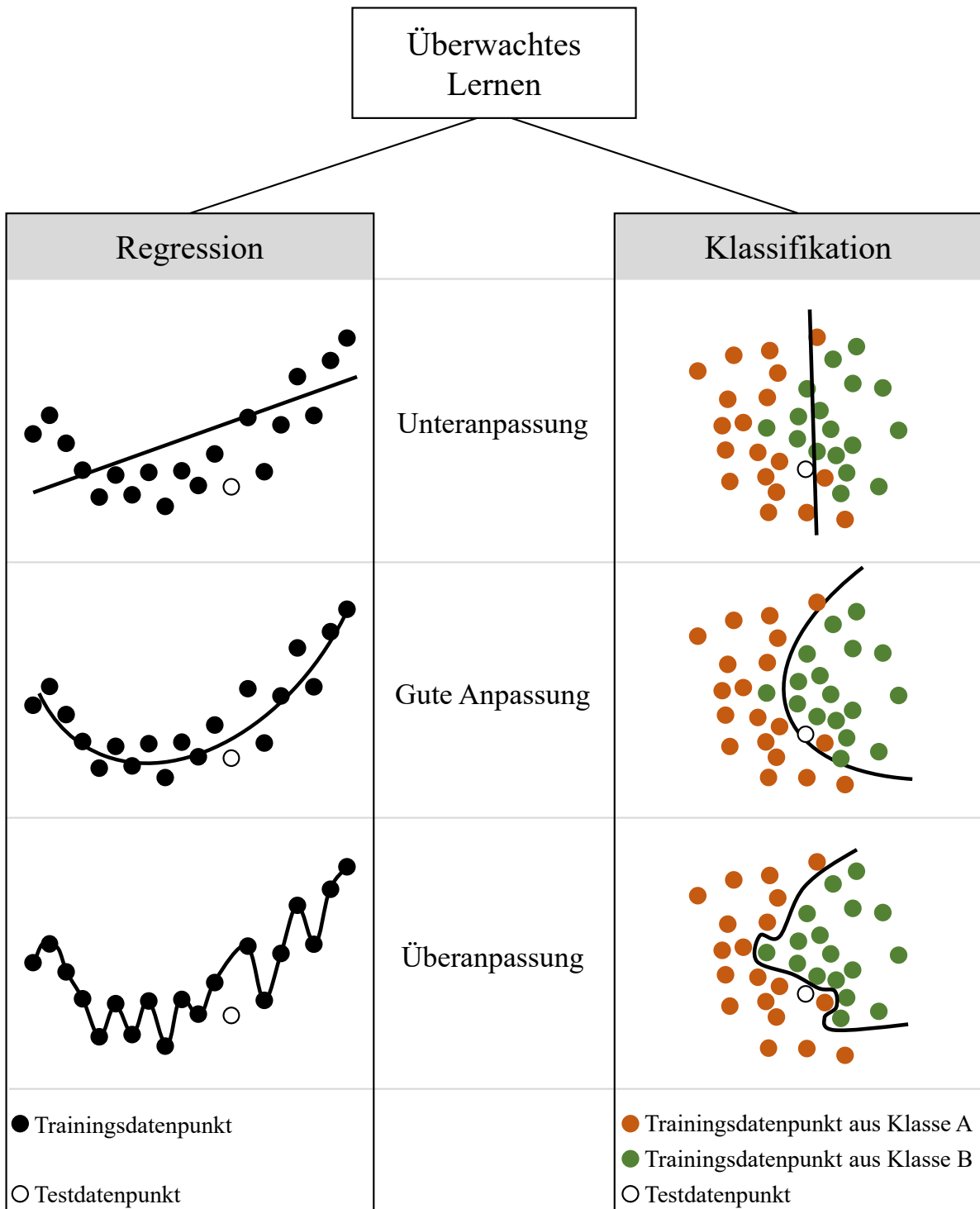


Abbildung 3.4: Unteranpassung, gute Anpassung und Überanpassung bei Regressions- und Klassifikationsmodellen

höchsten Wahrscheinlichkeit zugeordnet. Wenn für die Anzahl der möglichen Klassen $C = 2$ gilt, spricht man von einem *binären Klassifikationsproblem* (engl. binary classification problem), für $C \geq 3$ von einem *Multi-Klassen-Klassifikationsproblem* (engl. multi-class classification problem). Klassifikationsmodelle werden in der Industrie beispielsweise genutzt, um verschiedene Last- und Verschleißzustände von Maschinen und Anlagen im Rahmen einer prädiktiven Instandhaltungsplanung (engl. predictive maintenance) zu überwachen (vgl. Teti et al., 2010; Sikorska et al., 2011; Hellingrath und Cordes, 2014; Gao et al., 2015; Carvalho et al., 2019; Montero Jimenez et al., 2020; Zonta et al., 2020). In diesem Zusammenhang ist sowohl eine Konfiguration als ein binäres Klassifikationsproblem mit den beiden Klassen ‘normale Last’ und ‘hohe Last’ oder den beiden Klassen ‘Normalzustand’ und ‘Zustand kurz vor einem Ausfall’ als auch eine Konfiguration als Multi-Klassen-Klassifikationsproblem beispielsweise mit den Klassen ‘keine Last’, ‘niedrige Last’, ‘mittlere Last’ und ‘hohe Last’ denkbar. Ein weiteres Beispiel für ein Klassifikationsproblem ist die Echtzeitprognose der Qualität von Produkten bei der Fertigung (vgl. Colledani et al., 2014; Wang et al., 2021). Hierbei ist eine Konfiguration als binäres Klassifikationsproblem mit den beiden Klassen ‘in Ordnung’ und ‘nicht in Ordnung’ gängig. Als Elemente der Merkmalsvektoren \mathbf{x}^i von Klassifikationsmodellen lassen sich beispielsweise Sensordaten, wie Druckverläufe und Stromkurven, Zählerdaten, wie die Zeit seit der letzten Wartung, und Qualitätsdaten, wie die Anzahl tatsächlich fehlerhaft gefertigter Produkte des letzten Tages bestimmen, um den aktuellen Zustand des untersuchten Systems zu beschreiben.

Für die Evaluation von Klassifikationsmodellen sind die Rangfolgen der Klassenwahrscheinlichkeiten interessanter als die genauen Werte der Klassenwahrscheinlichkeiten. Ein gängiges Fehlermaß zur Evaluation von Klassifikationsmodellen ist die *Kreuzentropie* (engl. cross-entropy), die zu minimieren ist (vgl. Bishop, 1995; Hastie et al., 2009):

$$E(\boldsymbol{\theta}) = - \sum_{i=1}^{D_{\text{train}}} \sum_{c=1}^C \ell_c^i \ln \left(\frac{\hat{\ell}_c^i}{\ell_c^i} \right) \rightarrow \min_{\boldsymbol{\theta}}. \quad (3.13)$$

Dabei bezeichnet D_{train} die Anzahl der Trainingsbeispiele und $\boldsymbol{\theta}$ bezeichnet den Vektor der Modellparameter, die während des Trainingsprozesses verändert werden, um die Kreuzentropie zu minimieren. Dieses Fehlermaß ist nichtnegativ und es hat den Wert Null, wenn die prognostizierten Ausgabevektoren genau den wahren Ausgabevektoren entsprechen.

In dieser Arbeit werden neuronale Netze (vgl. Abschnitt 3.4.5) als Klassifikationsmodelle verwendet, um Eignungswahrscheinlichkeiten von Prognoseverfahren für bestimmte Zeitreihen zu bestimmen. Dieser Prozess wird auch als Meta-Lernen bezeichnet.

3.1.4 Regression

Regressionsprobleme (vgl. Hastie et al., 2009; Fahrmeir et al., 2013; Burkov, 2019) sind überwachte Lernprobleme (3.9) mit dem Ziel der Bildung eines *Regressionsmodells* (engl. regression model)

$$\hat{m} \in \mathcal{M}_{\text{Reg}}, \quad \hat{m} : \mathcal{X} \longrightarrow \mathcal{L} \quad (3.14)$$

aus dem Raum der Regressionsmodelle \mathcal{M}_{Reg} , welches von einem Eingabevektor $\mathbf{x}^i \in \mathbb{R}^F$ auf einen reellen Ausgabewert schließt:

$$\ell^i = \ell^i \in \mathcal{L} \subseteq \mathbb{R}. \quad (3.15)$$

Der Fall eines eindimensionalen Ausgabewertes wird als *univariate Regression* bezeichnet. Das Konzept lässt sich mit mehrdimensionalen Ausgabevektoren zu einer sogenannten *multivariaten Regression* erweitern, die in der vorliegenden Arbeit jedoch nicht betrachtet wird. Regressionsmodelle werden in der Industrie beispielsweise zur Vorhersage der Restnutzungsdauer (engl. remaining useful life) einer Maschine, eines Werkzeugs oder eines Bauteils genutzt, um rechtzeitig vor einem sich andeutenden Ausfall einen Wartungsauftrag zu einem möglichst günstigen Zeitpunkt zu planen (vgl. Si et al., 2011; Sikorska et al., 2011; Gao et al., 2015; Haefner und Lanza, 2017). Ebenso wie bei den genannten Anwendungen von Klassifikationsmodellen in der Industrie (vgl. Abschnitt 3.1.3) lassen sich auch hierbei Sensordaten, wie Druckverläufe und Stromkurven, Zählerdaten, wie die Zeit seit der letzten Wartung, und Qualitätsdaten, wie die Anzahl fehlerhaft gefertigter Produkte des letzten Tages, als Elemente der Merkmalsvektoren \mathbf{x}^i heranziehen, um den aktuellen Zustand des Systems zu beschreiben.

Zur Evaluation von Regressionsmodellen werden klassische Fehlermaße wie die *Summe der quadrierten Residuen* (engl. residual sum of squares)

$$\text{RSS}(\boldsymbol{\theta}) = \sum_{i=1}^{D_{\text{train}}} (\ell^i - \hat{\ell}^i)^2 \rightarrow \min_{\boldsymbol{\theta}}. \quad (3.16)$$

verwendet. Zur Bestimmung geeigneter Modellparameter ist dieses Maß zu minimieren.

3.2 Zeitreihenprognose

Zeitreihenprognoseprobleme (engl. time series forecasting problems, vgl. Ord et al., 2017; Hyndman und Athanasopoulos, 2018) lassen sich als Spezialfälle von Regressionsproblemen betrachten. Aufgrund der Bedeutung für die vorliegende Arbeit werden diese im Folgenden detailliert erläutert. Zunächst werden Zeitreihenprognoseprobleme formal definiert. Darauffolgend werden Evaluationskriterien für Zeitreihenprognosen über einzelne und mehrere Zeitreihen beschrieben.

3.2.1 Zeitreihenprognoseprobleme

Im Bereich der Zeitreihenprognose ist die einzige Datengrundlage zur Bildung eines Modells eine sogenannte *Zeitreihe* (engl. time series)

$$\mathbf{y} = \{y_1, y_2, \dots, y_T\} \in \mathcal{Y}, \quad (3.17)$$

eine geordnete Menge von Messwerten, die in regelmäßigen Zeitabständen vom Zeitpunkt 1 bis zum Zeitpunkt T gemessen wurden. Das Ziel eines Zeitreihenprognoseproblems besteht darin, ein

Modell zu finden, welches zukünftige Werte y_{T+h} der Zeitreihe prognostiziert, bevor sie bekannt werden. Die Anzahl der Perioden $h \in \mathbb{N}$, die in die Zukunft prognostiziert werden soll, wird als *Prognosehorizont* (auch *Horizont*, engl. *horizon*) bezeichnet. Zeitreihenprognosemodelle werden in der Industrie beispielsweise zur Vorhersage zukünftiger Kundenbestellungen basierend auf einer Zeitreihe vergangener Bestellungen verwendet (vgl. Gardner, 1990; Snyder et al., 2002; Warren Liao und Chang, 2010; Strijbosch et al., 2011; Acar und Gardner, 2012; Scholz-Reiter et al., 2014; Barrow und Kourentzes, 2016; Petropoulos et al., 2019; Kück und Freitag, 2021).

Basierend auf $\mathbf{y} \in \mathcal{Y}$ lassen sich Merkmalsextraktionsfunktionen $f : \mathcal{Y} \rightarrow \mathcal{X}$ verwenden, um den Systemzustand durch Merkmalsvektoren $\mathbf{x}^i \in \mathcal{X}$ zu beschreiben. Hierzu verwenden verschiedene Prognoseverfahren unterschiedliche Merkmalsextraktionsfunktionen (siehe Abschnitt 3.4). Ein häufig verwendeter Ansatz für Zeitreihenprognoseprobleme besteht darin, die Paare aus Eingabe- und Ausgabevektoren für das überwachte Lernproblem wie folgt zu bestimmen:

$$\mathbf{x}^i = (y_1, \dots, y_i) \in \mathbb{R}^i \text{ und } \ell^i = y_{i+h} \in \mathbb{R}, \text{ für } 1 \leq i \leq T - h. \quad (3.18)$$

Dies bedeutet, dass jedes Trainingsbeispiel (\mathbf{x}^i, ℓ^i) jeweils aus einem Vektor der Zeitreihenwerte vom ersten bis zum i -ten Zeitpunkt und dem um h Perioden späteren Wert der Zeitreihe besteht. Zum Training eines Prognosemodells müssen die Trainingsbeispiele in eine Trainings- und eine Validierungsmenge unterteilt werden. Während bei allgemeinen überwachten Lernproblemen die Trainingsbeispiele zufällig diesen Mengen zugeordnet werden können, bestehen die Mengen bei Zeitreihenprognoseproblemen aus den ersten T_{train} Werten und den folgenden $T_{\text{val}} = T - T_{\text{train}}$ Werten. Somit entstehen die beiden zusammenhängenden Mengen

$$\mathbf{D}_{\text{train}} = \{((y_1, y_{1+h}), \dots, ((y_1, \dots, y_{T_{\text{train}}-h}), y_{T_{\text{train}}}))\} \in \mathcal{D}^{T_{\text{train}}-h} \quad (3.19)$$

$$\mathbf{D}_{\text{val}} = \{((y_1, \dots, y_{T_{\text{train}}+1-h}), y_{T_{\text{train}}+1}), \dots, ((y_1, \dots, y_{T-h}), y_T)\} \in \mathcal{D}^{T_{\text{val}}}. \quad (3.20)$$

Basierend auf diesen beiden Mengen lässt sich mit Hilfe eines *Prognoseverfahrens* (engl. *forecasting method*) $a \in \mathcal{A}_{\text{ZRP}}$ ein sogenanntes *h-Schritt-Prognosemodell* (engl. *h-step ahead forecasting model*) trainieren:

$$\hat{m}_h \in \mathcal{M}_{\text{ZRP}}, \quad \hat{m}_h : \mathcal{X} \rightarrow \mathcal{L}, \quad \hat{m}_h(y_1, \dots, y_t) = \hat{y}_{t|t+h}. \quad (3.21)$$

Hierbei bezeichnet $\hat{y}_{t|t+h}$ eine *h-Schritt-Prognose* (engl. *h-step ahead forecast*), die Prognose des Zeitreihenwertes für die Periode $t + h$ ermittelt zum Zeitpunkt t . \mathcal{A}_{ZRP} und \mathcal{M}_{ZRP} sind die Räume aller Zeitreihenprognoseverfahren und Zeitreihenprognosemodelle. Die in dieser Arbeit verwendeten Prognoseverfahren werden in den Abschnitten 3.4 und 4.4 beschrieben.

3.2.2 Evaluationskriterien für Zeitreihenprognosen für einzelne Zeitreihen

Im Folgenden werden Kriterien für die Evaluation von Zeitreihenprognosen anhand einer einzelnen Zeitreihe beschrieben (vgl. Tashman, 2000; Küsters, 2012; Hyndman und Athanasopoulos, 2018).

Sei eine Zeitreihe $\mathbf{y} \in \mathcal{Y}$ gegeben. Es sei angemerkt, dass Prognosefehler erst dann berechnet werden können, wenn die korrekten zukünftigen Werte der Zeitreihe bekannt sind. Daher sei für den Rest dieses Abschnitts angenommen, dass zukünftige Werte der Zeitreihe bereits bekannt sind und somit alle Werte der *Trainingsmenge* (engl. training set), der *Validierungsmenge* (engl. validation set) und der *Testmenge* (auch Generalisierungsmenge, engl. test set) der Zeitreihe bekannt sind:

$$\mathbf{y}_{\text{train}} = \{y_1, \dots, y_{T_{\text{train}}}\} \quad (3.22)$$

$$\mathbf{y}_{\text{val}} = \{y_{T_{\text{train}}+1}, \dots, y_T\} \quad (3.23)$$

$$\mathbf{y}_{\text{test}} = \{y_{T+1}, \dots, y_{T+H}\}. \quad (3.24)$$

Hierbei gilt $T_{\text{train}} + T_{\text{test}} = T$. Mittlere Prognosefehler lassen sich für alle drei dieser Mengen berechnen und werden *Trainingsfehler* (engl. training error), *Validierungsfehler* (engl. validation error) und *Testfehler* (auch Generalisierungsfehler, engl. test error) genannt. Abbildung 3.5 visualisiert die Unterteilung einer Zeitreihe in eine Trainings-, eine Validierungs- und eine unbekannte Testmenge sowie die Berechnung des Validierungs- und des Testfehlers. In der Abbildung sind die Einschrittprognosen eines Random Walks zu sehen, welcher in Abschnitt 3.4.1 erläutert wird. Die im Folgenden beschriebenen Methoden zur Berechnung von Prognosefehlern sind unabhängig davon, auf welcher dieser drei Mengen sie basieren. Es müssen allerdings jeweils die Prognosehorizonte h spezifiziert werden sowie, von welchem *Ursprung* (engl. origin) o die jeweiligen Prognosen gemacht wurden.

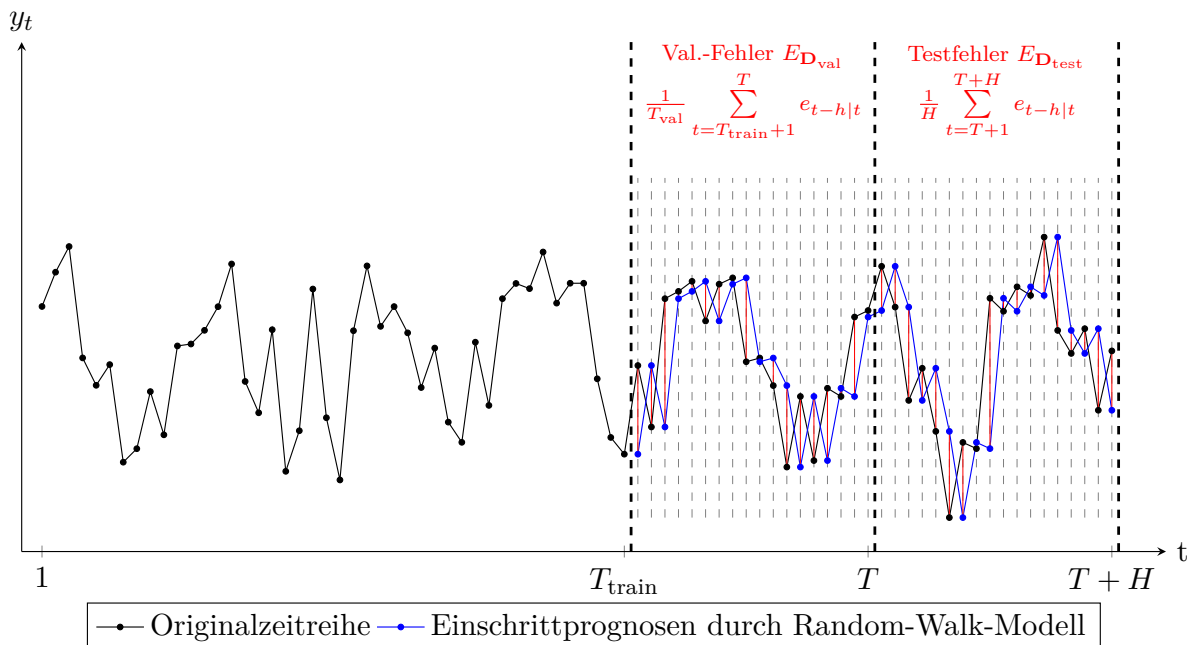


Abbildung 3.5: Unterteilung einer Zeitreihe in eine Trainings-, eine Validierungs- und eine Testmenge sowie Ermittlung des mittleren Validierungs- und des mittleren Testfehlers

Die Matrix

$$\hat{\mathbf{y}} = \begin{pmatrix} \hat{y}_{o|o+1} & \hat{y}_{o+1|o+2} & \hat{y}_{o+2|o+3} & \cdots & \hat{y}_{o+H-2|o+H-1} & \hat{y}_{o+H-1|o+H} \\ & \hat{y}_{o|o+2} & \hat{y}_{o+1|o+3} & \cdots & \hat{y}_{o+H-3|o+H-1} & \hat{y}_{o+H-2|o+H} \\ & & \hat{y}_{o|o+3} & \cdots & \hat{y}_{o+H-4|o+H-1} & \hat{y}_{o+H-3|o+H} \\ & & & & \vdots & \vdots \\ & & & & \hat{y}_{o|o+H-1} & \hat{y}_{o+1|o+H} \\ & & & & & \hat{y}_{o|o+H} \end{pmatrix}. \quad (3.25)$$

enthält die *Prognosen der Horizonte* h_{\min} bis h_{\max} von den *Ursprüngen* o bis $o + H - 1$. Sei $e : \mathcal{L} \rightarrow \mathcal{L}$ eine geeignete Fehlerfunktion, dann bezeichnet

$$e_{o|o+h} = e(y_{o+h}, \hat{y}_{o|o+h}) \quad (3.26)$$

den sogenannten *h-Schritt-Prognosefehler* (engl. *h-step ahead forecast error*) einer *h-Schritt-Prognose* für den Wert der Periode $o + h$ ermittelt zum Zeitpunkt o . Dieser Wert beschreibt die *Prognosegüte* (engl. *forecast accuracy*) eines einzelnen Prognosewertes. Allerdings lässt sich ein Modell nicht sinnvoll anhand einer einzelnen Prognose evaluieren, sondern es ist zumeist von Interesse, wie gut es zukünftige Werte über unterschiedliche Horizonte prognostiziert. Neben dem bereits in (3.7) definierten mittleren Fehler für eine Datenmenge $E_{\mathbf{D}}$ lassen sich für Zeitreihenprognosen weitere mittlere Fehler definieren. Sei

$$E_o^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{H} \sum_{h=h_{\min}}^{h_{\max}} e_{o|o+h} \quad (3.27)$$

der *mittlere Prognosefehler der Horizonte* h_{\min} bis h_{\max} vom *fixierten Ursprung* (engl. *fixed origin*) o , wobei $H = h_{\max} - h_{\min} + 1$ gilt. Dieses Maß gibt an, wie gut ein Prognosemodell für verschiedene Horizonte von einem fixierten Ursprung prognostiziert. In diesem Fall kann die Prognosegüte stark von speziellen Entwicklungen der Zeitreihe um den Ursprung sowie stochastischen Effekten abhängen. Ein robusteres Maß zur Evaluation der Prognosegüte über mehrere Horizonte und Ursprünge ist der *mittlere Prognosefehler der rollierenden Prognosen der Horizonte* h_{\min} bis h_{\max} über die *Ursprünge* o_{\min} bis o_{\max}

$$E_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{O} \sum_{o=o_{\min}}^{o_{\max}} E_o^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}}), \quad (3.28)$$

wobei $O = o_{\max} - o_{\min} + 1$ gilt. Dieses Vorgehen wird auch als *Zeitreihenkreuzvalidierung* (engl. *time series cross-validation*) bezeichnet. Wie man leicht sieht, ist $e_{o|o+h}$ ein Spezialfall von $E_o^{h_{\min}, h_{\max}}$ und dieser ist wiederum ein Spezialfall von $E_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$. Abbildung 3.6 zeigt, welche Zeitreihenwerte bei den drei verschiedenen Evaluationsarten genutzt werden. Die schwarzen Punkte werden für die Modellbildung genutzt, die blauen Punkte werden prognostiziert und die weißen Punkte bleiben bei der jeweiligen Evaluation ungenutzt. Während $e_{o|o+h}$ den Fehler eines einzelnen prognostizierten Wertes bestimmt, werden die mittleren Fehler über mehrere Prognosen

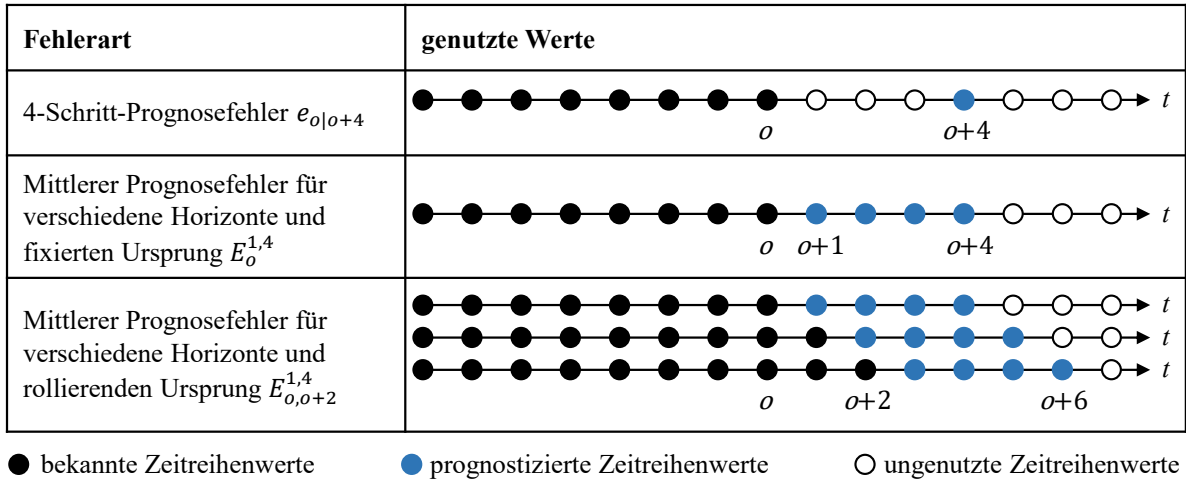


Abbildung 3.6: Genutzte Werte bei einer Zeitreihenkreuzvalidierung mit fixiertem oder rollierendem Ursprung

gemittelt. Für $E_o^{h_{\min}, h_{\max}}$ werden insgesamt $H = h_{\max} - h_{\min} + 1$ Werte berücksichtigt und für $E_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}$ werden jeweils H Werte von jedem berücksichtigten Ursprung genutzt, wodurch über $(o_{\max} - o_{\min} + 1)H$ Werte gemittelt wird.

Um die drei genannten Kriterien (3.26)-(3.28) zur Evaluation der Prognosegüte zu verwenden, muss eine geeignete Fehlerfunktion $e : \mathcal{L} \rightarrow \mathcal{L}$ gewählt werden. Klassische Fehlerfunktionen sind der *quadratische Fehler* (engl. squared error) SE, der *absolute Fehler* (engl. absolute error) AE und der *absolute prozentuale Fehler* (engl. absolute percentage error) APE (vgl. Hyndman und Athanasopoulos, 2018):

$$SE_{o|o+h} = (y_{o+h} - \hat{y}_{o|o+h})^2 \quad (3.29)$$

$$AE_{o|o+h} = |y_{o+h} - \hat{y}_{o|o+h}| \quad (3.30)$$

$$APE_{o|o+h} = \left| \frac{100(y_{o+h} - \hat{y}_{o|o+h})}{y_{o+h}} \right| \quad (3.31)$$

Analog zu (3.27) und (3.28) führen diese Fehlerfunktionen bei einer Mittelwertbildung über mehrere Horizonte und Ursprünge zum *mittleren quadratischen Fehler* (engl. mean squared error) MSE, dem *mittleren absoluten Fehler* (engl. mean absolute error) MAE und dem *mittleren absoluten prozentualen Fehler* (engl. mean absolute percentage error) MAPE. Die beiden erstgenannten Fehlermaße MSE und MAE sind skalenabhängig. Dies bedeutet, dass die mittleren Prognosefehler für Zeitreihen mit im Durchschnitt hohen absoluten Werten in der Regel höher sind als für Zeitreihen mit im Durchschnitt niedrigen absoluten Werten. Daher lassen sich diese Maße zwar zum Vergleich verschiedener Prognosen für eine Zeitreihe verwenden, jedoch nicht für den Vergleich verschiedener Prognoseverfahren oder Prognosemodelle über mehrere Zeitreihen, da aufgrund verschiedener durchschnittlicher absoluter Höhen unterschiedlicher Zeitreihen alle Zeitreihen mit einer anderen Gewichtung in den gesamten mittleren Fehler eingehen. Ein skalenunabhängiges

Fehlermaß ist der MAPE. Dieser gibt die Prognosegüte der Prognosen $\hat{\mathbf{y}}$ für die Zeitreihe \mathbf{y} als prozentuale Abweichung an, wobei die Werte der Zeitreihe die Grundwerte bilden. Allerdings hat auch dieses Fehlermaß mehrere nachteilige Eigenschaften. Der MAPE ist undefiniert falls $y_t = 0$ für mindestens einen Zeitpunkt t gilt und hat außerdem extreme Werte, falls y_t in der Nähe von 0 ist. Des Weiteren bestraft der MAPE negative Fehler stärker als positive Fehler.

Aufgrund der beschriebenen Nachteile werden die genannten Fehlermaße in dieser Arbeit nicht verwendet. Stattdessen wird die Prognosegüte mit Hilfe des *symmetrischen mittleren absoluten prozentualen Fehlers* (sMAPE, engl. symmetric mean absolute percentage error) gemessen. Dieser basiert auf dem Fehlermaß des *symmetrischen absoluten prozentualen Fehlers* (engl. symmetric absolute percentage error)

$$\text{sAPE}_{o|o+h} = \frac{200 |y_{o+h} - \hat{y}_{o|o+h}|}{|y_{o+h}| + |\hat{y}_{o|o+h}|} \quad (3.32)$$

und wird als

$$\text{sMAPE}_{o}^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{H} \sum_{h=h_{\min}}^{h_{\max}} \text{APE}_{o|o+h} \quad (3.33)$$

und

$$\text{RO-sMAPE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{O} \sum_{o=o_{\min}}^{o_{\max}} \text{sMAPE}_{o}^{h_{\min}, h_{\max}}(\mathbf{y}, \hat{\mathbf{y}}), \quad (3.34)$$

berechnet. Der sMAPE ist ein unverzerrtes, skalunenabhängiges Fehlermaß, welches sowohl für den Vergleich unterschiedlicher Prognosen für eine Zeitreihe als auch für den Vergleich mehrerer Prognoseverfahren oder -modelle über mehrere Zeitreihen unterschiedlicher Niveaus geeignet ist (vgl. Hyndman und Athanasopoulos, 2018). Ein weiteres häufig verwendetes Fehlermaß ist der *mittlere absolute skalierte Fehler* (engl. mean absolute scaled error) MASE, welcher die Prognosen durch die mittleren absoluten Einschnitt-Trainingsfehler eines Random Walks skaliert. Er basiert auf der Berechnung des *absoluten skalierten Fehlers* (engl. absolute scaled error)

$$\text{ASE}_{o|o+h} = \frac{y_{o+h} - \hat{y}_{o|o+h}}{\text{RO-MAE-RW}(\mathbf{y})}, \quad (3.35)$$

wobei

$$\text{RO-MAE-RW}(\mathbf{y}) = \frac{1}{T_{\text{train}} - 1} \sum_{i=2}^{T_{\text{train}}} |y_i - y_{i-1}| \quad (3.36)$$

den rollierenden mittleren absoluten Einschnittprognosefehler eines Random Walks auf der Trainingsmenge bezeichnet. Der MASE ist, ebenso wie der sMAPE, ein skalunenabhängiges Maß und dazu geeignet, Prognoseverfahren und -modelle hinsichtlich ihrer Prognosegüte über mehrere Zeitreihen zu vergleichen. In den Untersuchungen dieser Arbeit führten der sMAPE und der MASE zu den gleichen Schlussfolgerungen, daher werden die Ergebnisse unter Nutzung des MASE lediglich in Anhang E aufgeführt.

3.2.3 Evaluationskriterien für Zeitreihenprognosen über mehrere Zeitreihen

In dieser Arbeit werden die Prognosegüten verschiedener Zeitreihenprognoseverfahren und -modelle in Evaluationen mit fixiertem und rollierendem Ursprung über mehrere Zeitreihen verglichen. Als Evaluationskriterien werden sowohl mittlere Prognosefehler über mehrere Zeitreihen als auch statistische Tests verwendet. Es seien eine Menge von Z Zeitreihen

$$\mathbf{Y} = \{\mathbf{y}^1, \dots, \mathbf{y}^Z\} \quad (3.37)$$

und eine Menge von Prognosen

$$\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}^1, \dots, \hat{\mathbf{y}}^Z\} \quad (3.38)$$

gegeben. Dann lässt sich die mittlere Prognosegüte eines Prognoseverfahrens oder -modells als *durchschnittlicher symmetrischer mittlerer absoluter prozentualer Fehler der rollierenden Prognosen* (engl. average rolling-origin symmetric mean absolute percentage error) beschreiben

$$\text{Mean-RO-sMAPE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{Z} \sum_{z=1}^Z \text{RO-sMAPE}_{o_{\min}, o_{\max}}^{h_{\min}, h_{\max}}(\mathbf{y}^z, \hat{\mathbf{y}}^z). \quad (3.39)$$

In dieser Arbeit wird der Mean-RO-sMAPE verwendet, um die mittleren Prognosegüten verschiedener Prognoseverfahren und -modelle über eine Menge an Zeitreihen zu vergleichen. Dabei werden diverse Experimente mit verschiedenen Konfigurationen hinsichtlich der Horizonte sowie der fixierten und rollierenden Ursprünge durchgeführt. Unter anderem wird der mittlere Prognosefehler über die Horizonte 1 bis 18 von einem fixierten Ursprung $\text{Mean-RO-sMAPE}_{T,T}^{1,18}$ verwendet. Dies war sowohl das Hauptevaluationskriterium in der M3-Competition (vgl. Makridakis und Hibon, 2000) als auch in der NN3-Competition (vgl. Crone et al., 2011). Die Hauptergebnisse dieser Competitions werden in Abschnitt 3.3 erläutert.

Als skalenunabhängiges Fehlermaß ist der sMAPE zum Vergleich über mehrere Zeitreihen besser geeignet als beispielsweise die skalenabhängigen Maße MSE und MAE. Allerdings ist die Mittelwertbildung beim Mean-RO-sMAPE dennoch sensitiv gegenüber hohen Fehlern für einzelne Zeitreihen. Daher werden in dieser Arbeit statistische Tests als zusätzliches Evaluationskriterium für den Vergleich der Prognosegüten mehrerer Verfahren oder -modelle über mehrere Zeitreihen verwendet. Da die Annahmen einer Varianzanalyse mit Messwiederholungen, wie z.B. normalverteilte Stichproben, in der Regel nicht erfüllt werden können, werden nichtparametrische multiple Tests durchgeführt, sogenannte Friedman-Tests und Nemenyi-Tests. Diese werden im Folgenden erläutert.

Sei $E_j^z = E(\mathbf{y}^z, \hat{\mathbf{y}}_j^z)$ der mittlere Prognosefehler des j -ten Prognoseverfahrens a_j für die Zeitreihe \mathbf{y}^z . In diesem Zusammenhang lassen sich beliebige mittlere Prognosefehlermaße nutzen. In der vorliegenden Arbeit wird der RO-sMAPE verwendet. Der *Rang* (engl. rank) des j -ten Verfahrens in einer Menge von M Prognoseverfahren oder -modellen für die Zeitreihe \mathbf{y}^z wird definiert als

$$R_j^z = 1 + |\{i \in \{1, \dots, M\} : E_i^z < E_j^z\}| + \frac{|\{i \in \{1, \dots, M\} : E_i^z = E_j^z\}| - 1}{2}. \quad (3.40)$$

Dies bedeutet, dass das beste Verfahren den Rang 1 erhält, das zweitbeste den Rang 2 und so weiter. Falls zwei oder mehrere Verfahren exakt die gleiche Prognosegüte für eine Zeitreihe erzielen, erhalten all diese Verfahren den gleichen Rang. Wenn z.B. zwei Verfahren die geringsten mittleren Fehler erreichen, erhalten beide den Rang 1,5. Der *mittlere Rang des j-ten Prognoseverfahrens* in einer Menge von M Verfahren für eine Menge von Z Zeitreihen ist

$$\bar{R}_j = \frac{1}{Z} \sum_{z=1}^Z R_j^z. \quad (3.41)$$

Basierend auf den mittleren Rängen werden Friedman-Tests verwendet, um zu testen, ob die Prognosegüten der zu vergleichenden Prognoseverfahren sich signifikant unterscheiden (vgl. Friedman, 1937, 1940). Unter der Nullhypothese, dass die Prognosegüten aller zu vergleichenden Verfahren nicht unterschiedlich sind, ist die Friedman-Teststatistik

$$\chi_R^2 = \frac{12Z}{M(M+1)} \left(\sum_{j=1}^M (\bar{R}_j^2) - \frac{M(M+1)^2}{4} \right) \quad (3.42)$$

näherungsweise χ^2 -verteilt mit $M - 1$ Freiheitsgraden, falls die Anzahl der Zeitreihen Z und die Anzahl der zu vergleichenden Prognoseverfahren oder -modelle M groß genug sind (Demšar (2006) gibt als grobe Regel $Z > 10$ und $M > 5$ an). Falls die Nullhypothese des Friedman-Tests zum Signifikanzniveau α verworfen wird, d.h. wenn $\chi_R^2 > \chi_{(\alpha, M-1)}^2$, wird ein Nemenyi-Post-hoc-Test durchgeführt, um die signifikanten Unterschiede der Prognosegüten zu ermitteln (vgl. Nemenyi, 1962; Demšar, 2006). Zu diesem Zweck wird eine *kritische Distanz* (engl. critical distance)

$$CD_{(\alpha, M)} = \frac{q_{(\alpha, M)}}{\sqrt{2}} \sqrt{\frac{M(M+1)}{6Z}} \quad (3.43)$$

bestimmt, wobei $q_{(\alpha, M)}$ die studentisierte Spannweiten-Statistik (engl. studentized range statistic) mit unendlichen Freiheitsgraden bezeichnet. Der Nemenyi-Test betrachtet die Differenz der Prognosegüte zweier Verfahren als signifikant mit einem Signifikanzniveau α , falls die Differenz ihrer mittleren Ränge größer ist als die kritische Distanz:

$$\bar{R}_i - \bar{R}_j > CD_{(\alpha, M)}. \quad (3.44)$$

Die Ergebnisse der Nemenyi-Tests dieser Arbeit und die dazugehörigen Grafiken wurden in `Matlab` erzeugt (vgl. Kourentzes, 2012). Abbildung 3.7 zeigt eine beispielhafte Ergebnisgrafik beim Vergleich von $M = 8$ Prognoseverfahren. Der sehr niedrige p-Wert des Friedman-Tests zeigt, dass die Nullhypothese verworfen wurde und somit von signifikanten Unterschieden in der Prognosegüte der Verfahren ausgegangen werden kann. Auf der vertikalen Achse sind die verglichenen Prognoseverfahren von oben nach unten absteigend hinsichtlich ihrer Prognosegüte geordnet. Die Punkte auf der horizontalen Achse zeigen die mittleren Ränge der Prognoseverfahren. In diesem Zusammenhang werden die Prognosegüten zweier Verfahren als signifikant unterschiedlich angesehen, wenn ihre mittleren Ränge um mehr als die kritische Distanz $CD = 0.3$ voneinander

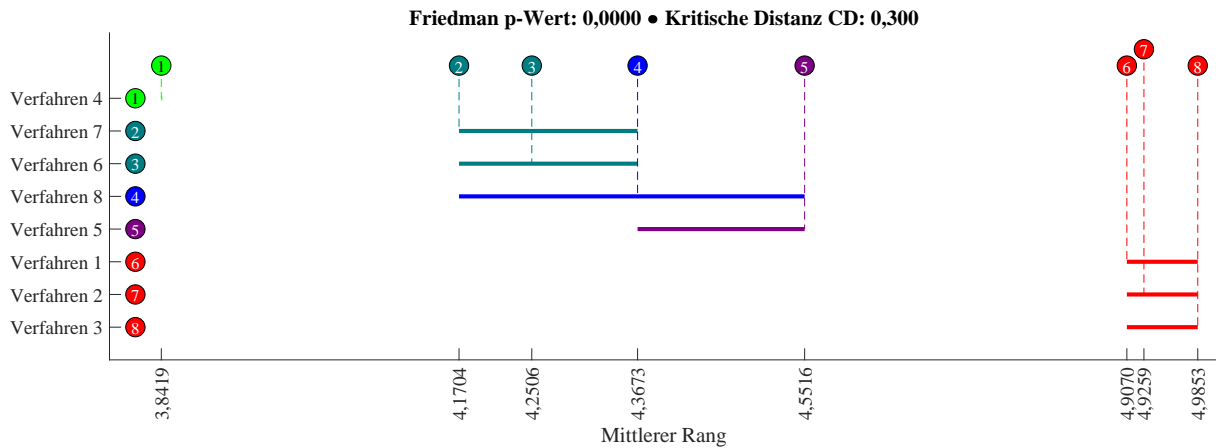


Abbildung 3.7: Beispielvisualisierung der Ergebnisse eines Nemenyi-Tests mit $M = 8$ Prognoseverfahren

abweichen. Wenn die Prognosegüte eines Verfahrens sich nicht signifikant von den Prognosegüten weiterer Verfahren unterscheidet, sind diese Verfahren durch eine horizontale Linie miteinander verbunden. In der Beispielgrafik erreicht Verfahren 4 eine signifikant bessere Prognosegüte als alle anderen Verfahren. Die Prognosegüten des 7-ten, 6-ten und 8-ten Verfahrens sind zueinander äquivalent. Die Prognosegüte von Verfahren 5 ist äquivalent zu der von Verfahren 8 jedoch signifikant schlechter als die Prognosegüten des 7-ten und des 6-ten Verfahrens. Die Verfahren 1, 2 und 3 erzielen zueinander äquivalente Prognosegüten, die signifikant schlechter sind als die der vorher genannten Verfahren.

3.3 Gängige Prognosemodellklassen und Benchmark-Datensätze

In diesem Abschnitt wird der Stand der Forschung hinsichtlich gängiger Zeitreihenprognosemodellklassen dargestellt. Zunächst werden die Datensätze, der beiden letzten großen Wettbewerbe für Zeitreihenprognoseverfahren beschrieben, der M3- und der NN3-Competition. Darauf folgend werden die Hauptergebnisse der Wettbewerbe zusammengefasst und die gängigsten Modellklassen identifiziert. Dies führt zur Bestätigung der Forschungsthese 1.

3.3.1 M3-Competition und NN3-Competition

Im Wissenschaftsgebiet der Zeitreihenprognose werden regelmäßig *Wettbewerbe* (engl. competitions) durchgeführt, bei denen die Trainingsmengen mehrerer Zeitreihen gegeben sind, für die ein Teilnehmer jeweils Prognosen der zukünftigen Werte einreichen kann. Die zukünftigen Werte sind zunächst nur den Organisatoren der Wettbewerbe bekannt und werden in der Regel nach Beendigung des Wettbewerbs und Auswertung der Ergebnisse der Öffentlichkeit zugänglich gemacht. Durch die Publikation der Ergebnisse sowie die Veröffentlichung der Datensätze inklusive der zukünftigen Werte können die Datensätze auch im Nachhinein für weitere Vergleichsstudi-

en genutzt werden, um neu entwickelte Prognoseverfahren stets gegen objektive Benchmarks vergleichen zu können. Hyndman (2020) gibt einen kurzen Überblick über die wichtigsten Zeitreihenprognosewettbewerbe. Die größten Wettbewerbe mit den meisten Teilnehmern waren die sogenannten M-Competitions (vgl. Makridakis et al., 1982, 1993; Makridakis und Hibon, 2000; Makridakis et al., 2020) und die NN3-Competition (vgl. Crone et al., 2011). Im Folgenden werden die Datensätze der M3- und der NN3-Competition beschrieben und die Hauptergebnisse erläutert, da diese beiden Datensätze in der vorliegenden Arbeit verwendet werden.

Beschreibung der Datensätze

Tabelle 3.1 beschreibt die in dieser Arbeit verwendeten Datensätze, die aus der M3- und der NN3-Competition entstammen. Beide sind etablierte Benchmark-Datensätze, die auch nach den Wettbewerben in zahlreichen empirischen Studien verwendet wurden. Der M3-Datensatz wurde beispielsweise in (Prudêncio und Ludermir, 2004; Poler und Mula, 2011; Scholz-Reiter et al., 2014; Fildes und Petropoulos, 2015; Ferbar Tratar et al., 2016; Talagala et al., 2018; Petropoulos et al., 2019; Kück und Freitag, 2021) verwendet und der NN3-Datensatz in (Lemke und Gabrys, 2010a; Barrow und Crone, 2016; Fonseca und Gomez, 2016; Kück et al., 2016b; Widodo et al., 2016).

Tabelle 3.1: Beschreibung der Datensätze der monatlichen Industriedaten der M3-Competition und der NN3-Competition

Datensatz	Anzahl Zeitreihen			Frequenz	Anzahl Zeitreihenwerte		Horizont
	kurz	lang	gesamt		minimal	maximal	
Monatliche M3-Industriedaten	278	530	808	monatlich	50	126	18
NN3-Daten	51	60	111	monatlich	50	126	18

In der vorliegenden Arbeit wird eine Teilmenge des Datensatzes der *M3-Competition* (vgl. Makridakis und Hibon, 2000) verwendet. Der komplette Datensatz der M3-Competition besteht aus 3003 univariaten Zeitreihen, die in verschiedenen Frequenzen (jährlich, vierteljährlich, monatlich, andere) aus verschiedenen Anwendungsbereichen (Mikroökonomie, Industrie, Makroökonomie, Finanzwesen, Demographie, andere) gemessen wurden. Für den Wettbewerb erhielten die Teilnehmer die Zeitreihen und mussten die zukünftigen Werte vorhersagen, die nur den Organisatoren bekannt waren. Die tatsächlichen zukünftigen Werte wurden genutzt, um die Prognosegüte der von den Teilnehmern eingereichten Prognosen zu evaluieren. Da der Fokus der vorliegenden Arbeit auf Industriedaten gerichtet ist, wird die Teilmenge der 808 monatlichen Zeitreihen der Kategorien Mikroökonomie und Industrie genutzt. Diese Teilmenge besteht aus realen Industriedaten zu Bedarfs-, Liefer-, Produktions- und Absatzmengen. Im Allgemeinen erlaubt die Struktur einer Supply Chain die Umwandlung von Nicht-Bedarfsdaten in den Bedarf nach anderen Produkten, wie beispielsweise Rohmaterialien oder Ersatzteilen, die benötigt werden, um neue Produkte zu fertigen. Daher wird angenommen, dass alle 808 Zeitreihen die

Nachfrage von Kunden eines produzierenden Unternehmens widerspiegeln. Es sei angemerkt, dass einige der Zeitreihen Dienstleistungen, wie Instandhaltungsaktivitäten oder Transporte, anstelle von physischen Produkten beschreiben. Allerdings werden diese Zeitreihen dennoch für die Evaluationsstudien dieser Arbeit mitberücksichtigt, um vergleichbare Ergebnisse zu früheren und zukünftigen Studien anhand des gleichen Datensatzes zu erzielen.

Die ursprünglich in der M3-Competition gegebenen Zeitreihen werden als Trainingsmengen genutzt und die im Nachhinein veröffentlichten zukünftigen Werte werden als Testmengen genutzt. Dabei bestehen die Trainingsmengen aus 50 bis 126 Werten. Jede Testmenge der 808 Zeitreihen besteht aus $H = 18$ Zeitreihenwerten, die prognostiziert werden müssen. Für die Evaluationsstudien dieser Arbeit wird eine Zeitreihe als lang angesehen, wenn ihre Trainingsmenge mindestens 100 Werte enthält und ansonsten als kurz. Dadurch lässt sich der Datensatz von 808 monatlichen Zeitreihen der M3-Competition in 278 kurze und 530 lange Zeitreihen unterteilen.

Während in den vorherigen Wettbewerben, wie der M1- oder der M3-Competition, fast ausschließlich statistische Verfahren verwendet wurden, sollten in der *NN3-Competition* insbesondere die Prognosegüten von maschinellen Lernverfahren und Computational-Intelligence-Verfahren, wie neuronalen Netzen, getestet werden (vgl. Crone et al., 2011). Als Datensatz für den Wettbewerb wurden von den Organisatoren 111 der beschriebenen 808 monatlichen Zeitreihen der M3-Competition ausgewählt. Diese Information war nur den Organisatoren jedoch nicht den Teilnehmern bekannt und wurde erst nach Durchführung des Wettbewerbs veröffentlicht. Die 111 Zeitreihen wurden balanciert ausgewählt, sodass der Datensatz aus 60 langen und 51 kurzen Zeitreihen besteht. Zudem lässt dieser sich in 54 saisonale und 57 nicht-saisonale Zeitreihen unterteilen. Ebenso, wie in der M3-Competition, war auch das Ziel der NN3-Competition die Vorhersage von $H = 18$ zukünftigen Werten jeder Zeitreihe.

Hauptergebnisse der M3- und der NN3-Competition

Detaillierte Beschreibungen der verwendeten Prognoseverfahren der M3- und der NN3-Competition sowie die Rangfolgen und erzielten Prognosefehler sind in Anhang B zu finden. Für ausführliche Diskussionen hierzu, siehe Makridakis und Hibon (2000) sowie Crone et al. (2011). Im Folgenden werden die Hauptergebnisse der M3- und der NN3-Competition zusammengefasst.

- **Es existiert keine stets beste Modellklasse zur Prognose der Kundennachfrage produzierender Unternehmen.** Bei Betrachtung der Rangfolgen der M3- und der NN3-Competition (siehe Anhang B) ist ersichtlich, dass die Verfahren mit den besten Prognosen jeweils nicht auf eine einzelne Modellklasse beschränkt sind. Stattdessen erreichen die besten Verfahren ihre durchschnittlich hohe Prognosegüte dadurch, dass sie für jede Zeitreihe entweder aus mehreren möglichen Modellklassen eine auswählen und hieraus ein Prognosemodell selektieren (z.B. ForecastPro und ForecastX) oder Modelle aus mehreren verschiedenen Klassen kombinieren (z.B. Theta und Wildi). Dieses Ergebnis zeigt sich ebenfalls in der Darstellung des Standes der Forschung zur Selektion geeigneter

Prognosemodelle in Kapitel 4. Somit lässt sich FT 1 bestätigen und es existiert keine stets beste Modellklasse zur Prognose der Kundennachfrage produzierender Unternehmen.

- **Die relativen Prognosegüten der Prognoseverfahren hängen von den Charakteristika der Daten ab.** Die Charakteristika einer Zeitreihe haben einen großen Einfluss darauf, welche Prognoseverfahren geeignet sind. Beispielsweise eignen sich bestimmte Verfahren gut zur Prognose langer oder kurzer Zeitreihen sowie saisonaler oder nicht-saisonaler Zeitreihen.
- **Die Prognosegüte eines Verfahrens hängt vom Prognosehorizont ab.** In den Wettbewerben erzielten andere Verfahren gute Ergebnisse für langfristige Prognosen als für kurzfristige oder mittelfristige Prognosen.
- **Die Prognosegüte eines Verfahrens hängt von der Fehlerfunktion ab.** Bei der Verwendung verschiedener Fehlerfunktionen, wie dem sMAPE oder dem MASE können sich die Rangfolgen von Prognoseverfahren unterscheiden. Allerdings zeigte sich in den Wettbewerben, dass die Tendenzen auch bei verschiedenen Fehlerfunktionen ähnlich bleiben. Wenn ein Verfahren in der Rangfolge der besten 10 Verfahren für eine Fehlerfunktion ist, gehört es vermutlich auch zu den 10 besten Verfahren für eine andere Fehlerfunktion.
- **Prognoseverfahren sollten rollierend über verschiedene Ursprünge anhand mehrerer Zeitreihen evaluiert werden.** Im Rahmen eines Prognosewettbewerbs lässt sich nur die Evaluation anhand mehrerer Zeitreihen erfüllen, nicht jedoch die Evaluation mit rollierendem Ursprung, da lediglich die Trainingsmenge jeder Zeitreihe gegeben ist und eine fixierte Anzahl zukünftiger Werte prognostiziert werden muss. Um eine größere Robustheit der Ergebnisse zu erzielen, wird jedoch für folgende Studien anhand dieses Datensatzes eine Evaluation mit rollierendem Ursprung empfohlen.
- **Kombinationen erzielen im Durchschnitt eine höhere Prognosegüte als individuelle Verfahren.** Durch die Kombination der Prognosen mehrerer Verfahren zu einer finalen Prognose lässt sich eine hohe Robustheit bzgl. der Eignung der Verfahren für bestimmte Datencharakteristika erzielen, da sich die Stärken und Schwächen verschiedener Verfahren im Durchschnitt ausgleichen. Fast alle Verfahren, die in der M3- und der NN3-Competition gute Ergebnisse erzielten, kombinierten die Prognosen mehrerer Verfahren.
- **Komplexe Verfahren prognostizieren nicht notwendigerweise besser als simple Verfahren.** Dies ist ein vieldiskutiertes Ergebnis der Prognosewettbewerbe. Simple Verfahren, wie Random Walks oder einfache exponentielle Glättungen, haben wenige oder gar keine Parameter, die basierend auf einer gegebenen Zeitreihe angepasst werden müssen. Dadurch sind diese Verfahren auch ohne Expertenwissen verwendbar, allerdings haben sie begrenzte Möglichkeiten zur Abbildung volatiler Zeitreihenverläufe. Komplexe Verfahren, wie beispielsweise neuronale Netze, bieten zwar die Möglichkeit, auch komplexere

Zeitreihenverläufe abzubilden, jedoch müssen hierfür die meist zahlreichen Parameter der Verfahren geeignet angepasst werden. Aufgrund der hohen Anzahl an Zeitreihen, die in den Wettbewerben zu prognostizieren sind, können in der Regel nicht alle Parameter jedes Modells für jede Zeitreihe manuell basierend auf Expertenwissen angepasst werden. Stattdessen werden automatische Algorithmen zur Parameteroptimierung verwendet. Allerdings besteht in diesem Zusammenhang für nichtlineare Modellklassen, wie neuronale Netze, ohne statistische Theorie zur Parameterschätzung die Schwierigkeit, dass verfügbare Trainingsalgorithmen meist große Trainingsmengen an Daten zur sinnvollen Parameteroptimierung benötigen oder generell schlecht für den Anwendungsfall der Zeitreihenprognose geeignet sind. Dies erklärt die Rangfolgen der M3- und der NN3-Competition (siehe Anhang B), in denen komplexe Verfahren sowohl die vordersten als auch die hintersten Ränge belegen, während simple Verfahren sich häufig im Mittelfeld der Rangfolgen befinden. In diesem Zusammenhang sei angemerkt, dass zum Zeitpunkt dieser Recherche gerade ein weiterer großer Prognosewettbewerb, die M4-Competition, in Planung war. Da die Ergebnisse und die Datensätze dieses Wettbewerbs jedoch noch nicht veröffentlicht waren, werden diese in der vorliegenden Arbeit nicht betrachtet. Im Großen und Ganzen bestätigen die Ergebnisse der M4-Competition die hier beschriebenen Ergebnisse. Für weitere Diskussionen, siehe Makridakis et al. (2020).

3.3.2 Gängige Prognosemodellklassen

Die Tabellen B.1 und B.2 beschreiben neben den Ergebnissen der M3- und der NN3-Competition zusätzlich die Funktionsweise und verwendeten Modellklassen aller Teilnehmer. Es zeigt sich, dass die am häufigsten verwendeten Modellklassen die Klassen der exponentiellen Glättung, der autoregressiven integrierten gleitenden Mittelwerte (ARIMA-Modelle) und der neuronalen Netze sind. Dies sind die gängigsten Modellklassen für Zeitreihenprognosen, was sich ebenfalls durch ausführliche Beschreibungen in aktuellen Standardwerken zur Zeitreihenprognose (vgl. Ord et al., 2017; Hyndman und Athanasopoulos, 2018) sowie in der Verwendung in aktuellen empirischen Studien (vgl. Fildes und Petropoulos, 2015; Fonseca und Gomez, 2016; Kück et al., 2016b; Talagala et al., 2018; Kourentzes et al., 2019; Petropoulos et al., 2019; Makridakis et al., 2020; Montero-Manso et al., 2020; Smyl, 2020; Hewamalage et al., 2021; Ma und Fildes, 2021) zeigt. Zusätzlich zu diesen drei Modellklassen werden häufig Random Walks als Benchmarks verwendet, da diese leicht anzuwenden sind und teilweise schwer zu überbietende Ergebnisse insbesondere dann erzielen, wenn Zeitreihen einen hohen Rauschanteil und nur wenig strukturelle Informationen enthalten. Für die vier genannten gängigsten Modellklassen für Zeitreihenprognosen existieren jeweils etablierte automatische Prognoseverfahren, die in dieser Arbeit verwendet werden. Diese werden in Abschnitt 3.4 beschrieben.

Eine Modellklasse, die bislang selten für die Prognose industrieller Zeitreihen verwendet wurde (vgl. Nikolopoulos et al., 2016; Kück und Freitag, 2021), jedoch erfolgreiche Anwendung in

anderen Wissenschaftsbereichen fand, ist die Klasse der lokalen k -nächste-Nachbarn-Modelle. Neben Anwendungen in der Finanzwissenschaft (vgl. Lisi und Medio, 1997; Cao und Soofi, 1999; Lisi und Schiavo, 1999; Meade, 2002; Alvarez-Diaz, 2008) wurden Modelle aus dieser Klasse insbesondere erfolgreich in den Naturwissenschaften angewandt (vgl. Ragwitz und Kantz, 2002; Kantz et al., 2004; Dimri et al., 2008; Jursa und Rohrig, 2008; Singh et al., 2015), in denen sie auch verschiedene Prognosewettbewerbe gewannen (vgl. Weigend und Gershenfeld, 1992; Sauer, 1994; McNames, 2002). Während diese Modelle im Wissenschaftsgebiet der nichtlinearen Dynamik, dem sie entstammen, gängig sind, wurden sie in den aktuellen Wettbewerben zur Zeitreihenprognose nur sporadisch verwendet. Allerdings erreichte ein Ansatz aus dieser Modellklasse den fünften Rang der NN3-Competition (vgl. D'yakonov, 2007; Crone et al., 2011). Diese Ergebnisse zeigen, dass lokale k -nächste-Nachbarn-Modelle ebenfalls das Potenzial zur erfolgreichen Anwendung für Nachfrageprognosen bieten. Da jedoch bislang kein etablierter Ansatz zur Anpassung der Modellparameter für lokale k -nächste-Nachbarn-Modelle existiert, wird in dieser Arbeit ein neuartiger Ansatz zur Modellselektion vorgestellt, der in Abschnitt 3.4.4 ausführlich beschrieben wird.

3.4 Verwendete Zeitreihenprognoseverfahren mit vorgegebener Modellklasse

Nach der Identifikation der gängigsten Prognosemodellklassen im letzten Abschnitt, werden diese im Folgenden beschrieben. Abbildung 3.8 zeigt Beispielprognosen verschiedener Prognoseverfahren für drei unterschiedliche Zeitreihen. Die schwarzen Linien stellen die gegebenen Zeitreihen dar und die blauen Linien die jeweiligen Prognosen. Es werden die Prognosen der acht in dieser Arbeit verwendeten Prognoseverfahrenskandidaten dargestellt sowie die Prognosen vier unterschiedlicher Formen exponentieller Glättungsmodelle.

3.4.1 Random-Walk-Modelle

Nicht-saisonaler Random Walk

Ein häufig als Benchmark verwendetes Prognosemodell ist der *Random Walk* (rw, auch naives Verfahren, engl. naïve method), welcher eine Zeitreihenentwicklung als global konstanten Prozess modelliert (vgl. Hyndman und Athanasopoulos, 2018). Das Modell hat keine Parameter, die angepasst werden müssen. Eine h -Schritt-Prognose wird als der aktuelle Wert der Zeitreihe ermittelt:

$$\hat{m}_h^{\text{rw}}(y_t) = \hat{y}_{t|t+h}^{\text{rw}} = y_t. \quad (3.45)$$

Abbildung 3.5 zeigt Einschrittprognosen mit Hilfe eines Random-Walk-Modells. In der vorliegenden Arbeit wird die Funktion `rwf` des `forecast`-Packages in `R` verwendet, um Random-Walk-Prognosen zu bestimmen (vgl. Hyndman et al., 2018b).

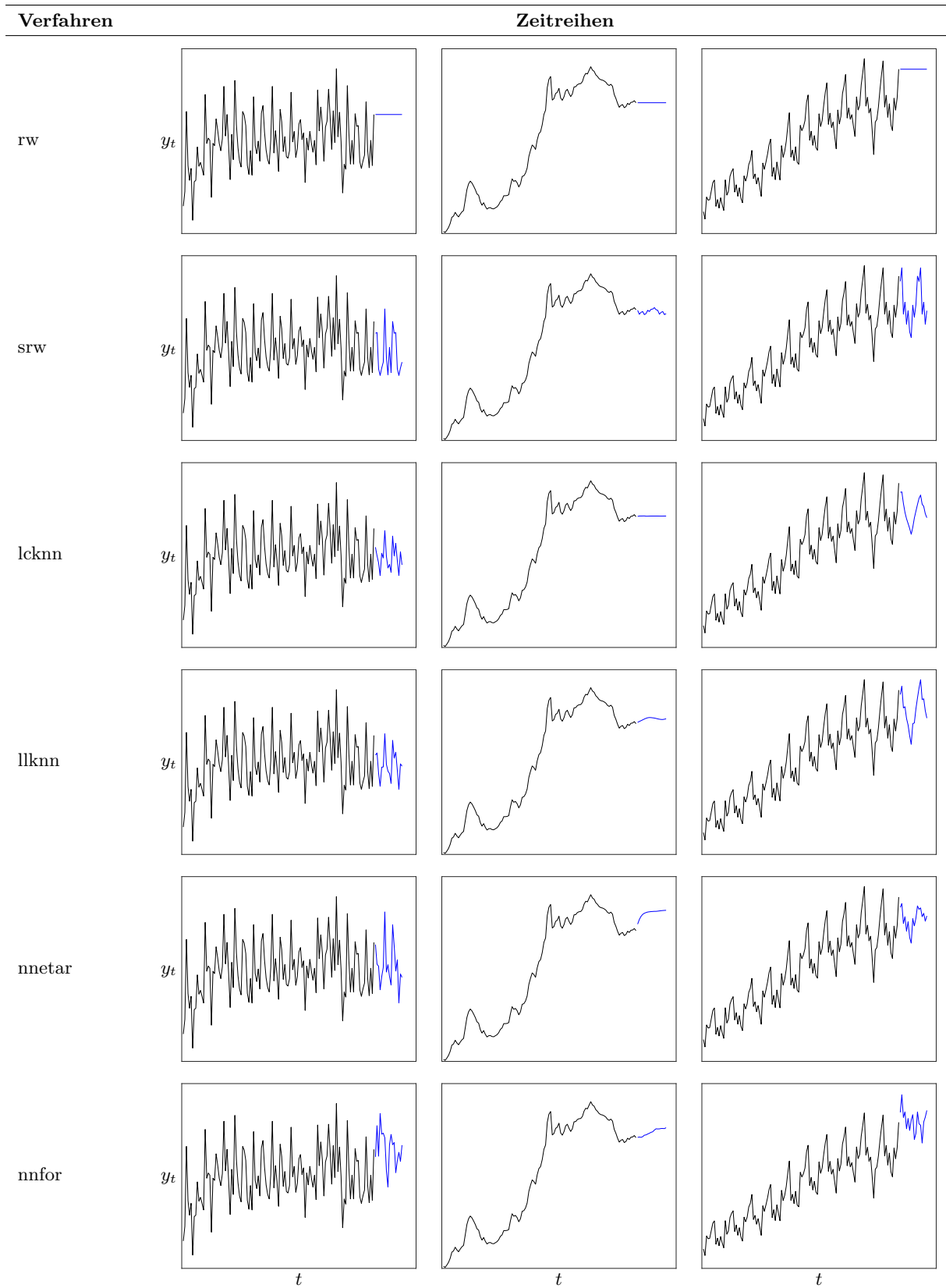


Abbildung 3.8: Beispielprognosen verschiedener Verfahren für unterschiedliche Zeitreihen

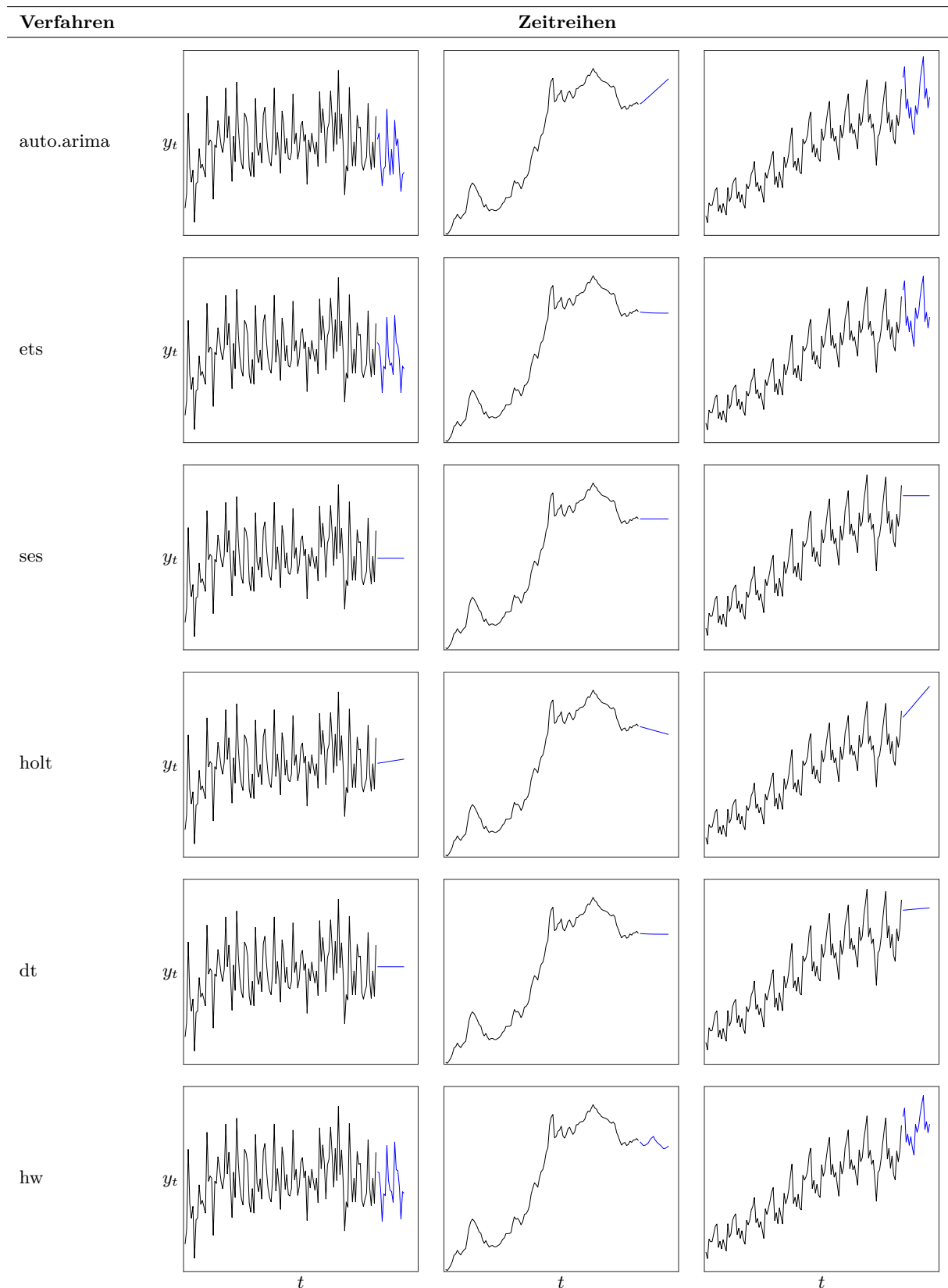


Abbildung 3.8: Beispielprognosen verschiedener Verfahren für unterschiedliche Zeitreihen (Fortsetzung)

Saisonaler Random Walk

Ein ebenfalls häufig als Benchmark verwendetes Prognosemodell, welches insbesondere für saisonale Zeitreihen nützlich ist, ist der *saisonale Random Walk* (srw, auch saisonales naives Verfahren, engl. seasonal naïve method, vgl. Hyndman und Athanasopoulos, 2018). In diesem Fall wird eine h -Schritt-Prognose als letzter Wert der gleichen Saison ermittelt. Da in der vorliegenden Arbeit lediglich monatliche Zeitreihen betrachtet werden, beträgt die Saisonalität $d = 12$. Somit wird eine h -Schritt-Prognose für einen bestimmten Monat als der Wert der Zeitreihe für den gleichen Monat im vorherigen Jahr bestimmt:

$$\hat{m}_h^{\text{srw}}(y_t) = \hat{y}_{t|t+h}^{\text{srw}} = y_{t+h-d(1+\lfloor \frac{h-1}{d} \rfloor k)} = y_{t+h-12(1+\lfloor \frac{h-1}{12} \rfloor k)}. \quad (3.46)$$

Hierbei bezeichnet

$$\lfloor x \rfloor = \max\{j \in \mathbb{Z} : j \leq x\} \quad (3.47)$$

die Abrundungsfunktion, die genutzt wird, um auf die größte ganze Zahl abzurunden. Dies bedeutet, dass beispielsweise eine Prognose des Zeitreihenwertes für den Monat Juni als Zeitreihenwert im Juni des letzten Jahres ermittelt wird. Zur Berechnung der Prognosen des srw wird in dieser Arbeit die Funktion `snaive` des `forecast`-Packages in R verwendet (vgl. Hyndman et al., 2018b).

3.4.2 ARIMA Modelle

Übersicht

Autoregressive integrierte gleitende Mittelwerte (ARIMA, engl. auto-regressive integrated moving averages) sind eine Modellklasse, die Zeitreihen als lineare Modelle basierend auf ihren Autokorrelationen beschreibt (vgl. Box et al., 2015; Hyndman et al., 2018b). Das Prinzip basiert auf der Annahme, dass betrachtete Zeitreihen stationär sind. Um eine nicht-stationäre Zeitreihe stationär zu machen, werden Differenzbildungen eingesetzt. Ein ARIMA-Modell setzt sich aus einem autoregressiven Prozess, einem gleitenden Mittelwert und einer Differenzbildung zusammen. ARIMA-Modelle basieren auf Arbeiten von Yule (1927) und erlangten Popularität durch das Werk von Box und Jenkins (1970). Sie gehören zu den am häufigsten verwendeten Zeitreihenprognosemodellen der letzten Jahrzehnte (vgl. Geurts und Patrick Kelly, 1986; Grambsch und Stahel, 1990; Goodrich, 2000; Meade, 2000a; Mélard und Pasteels, 2000; Reilly, 2000; De Gooijer und Hyndman, 2006; Hyndman und Khandakar, 2008; Babai et al., 2013; Box et al., 2015; Svetunkov und Boylan, 2019; Hyndman, 2020). In der vorliegenden Arbeit wird die `auto.arima`-Funktion des `forecast`-Packages in R verwendet, um die Parameter eines ARIMA-Modells basierend auf einer gegebenen Zeitreihe zu schätzen (vgl. Hyndman und Khandakar, 2008; Hyndman et al., 2018b). Aufgrund weitreichender Literatur zu ARIMA-Modellen (vgl. Box et al., 2015; Brockwell und Davis, 2016; Hyndman und Athanasopoulos, 2018) werden im Folgenden die theoretischen Grundlagen lediglich kurz beschrieben.

Theoretische Grundlagen von ARIMA-Modellen

Sei

$$\tau_- : \mathbb{Z} \longrightarrow \mathbb{Z}, \quad \tau_-(y_t) = y_{t-1} \quad (3.48)$$

der *Rückverschiebungsoperator* (engl. backshift operator), für den gilt

$$\tau_-^h(y_t) = \underbrace{\tau_- \tau_- \cdots \tau_-}_{h \text{ mal}}(y_t) = y_{t-h}. \quad (3.49)$$

Dann lässt sich ein *nicht-saisonales* ARIMA(p, δ, q)-Modell (engl. non-seasonal ARIMA model) durch

$$u(\tau_-) (1 - \tau_-^\delta) y_t = \zeta + v(\tau_-) \epsilon_t \quad (3.50)$$

beschreiben, wobei

$$u(\tau_-) = 1 - u_1 \tau_- - \dots - u_p \tau_-^p, \quad \text{mit } u_1, \dots, u_p \in \mathbb{R} \quad (3.51)$$

$$v(\tau_-) = 1 + v_1 \tau_- + \dots + v_q \tau_-^q, \quad \text{mit } v_1, \dots, v_q \in \mathbb{R} \quad (3.52)$$

zwei Polynome der *Ordnung der Autoregression* (engl. order of auto-regression) p sowie der *Ordnung der Mittelwerte* (engl. order of moving averages) q sind. δ ist die *Ordnung der ersten Differenzbildungen* (engl. order of first-differencing), $\zeta \in \mathbb{R}$ ist eine Konstante und ϵ_t ist ein weißes Rauschen mit Mittelwert Null und Varianz σ^2 . Ein *saisonales* ARIMA(p, δ, q)(P, Δ, Q) $_n$ -Modell (engl. seasonal ARIMA model) wird durch

$$U(\tau_-^n) u(\tau_-) (1 - \tau_-^n)^\Delta (1 - \tau_-)^\delta y_t = \zeta + V(\tau_-^n) v(\tau_-) \epsilon_t \quad (3.53)$$

beschrieben, wobei U und V die P - und Q -dimensionalen saisonalen Äquivalente von u und v sind. Δ ist die *Ordnung der saisonalen Differenzbildungen* (engl. order of seasonal-differencing) und n ist die Anzahl der Beobachtungen pro Jahr.

Selektion eines ARIMA-Modells

Um ein ARIMA-Modell zu selektieren, welches eine gegebene Zeitreihe gut abbildet, müssen die Modellordnungen p, δ, q, P, Δ und Q sinnvoll gewählt werden und die Modellkoeffizienten der Polynome u, v, U und V geeignet geschätzt werden. Es existieren verschiedene Algorithmen für diesen Zweck (vgl. Peña et al., 2001; Hyndman und Khandakar, 2008; Hyndman und Athanasopoulos, 2018). In der vorliegenden Arbeit wird zur Selektion geeigneter ARIMA-Modelle die `auto.arima`-Funktion des `forecast`-Packages in R verwendet (vgl. Hyndman und Khandakar, 2008; Hyndman et al., 2018b). Diese bestimmt die Ordnungen der ersten Differenzbildungen δ und saisonalen Differenzbildungen Δ durch Verwendung von KPSS-Tests (vgl. Kwiatkowski et al., 1992) und saisonalen Unit-Root-Tests (vgl. Wang et al., 2006). Nach Anwendung der Differenzbildungen werden die verschiedenen Polynomordnungen p, q, P und Q vorgegeben und

jeweils die Modellkoeffizienten von u, v, U und V durch Minimierung des *korrigierten Akaike-Informationenkriteriums* (AICc, engl. corrected Akaike's information criterion)

$$\text{AICc} = \text{AIC} + \frac{2(|\boldsymbol{\theta}|)(|\boldsymbol{\theta}| + 1)}{T - (|\boldsymbol{\theta}| + 1)} \quad (3.54)$$

ermittelt, wobei

$$\text{AIC} = -2 \log(L) + 2|\boldsymbol{\theta}| \quad (3.55)$$

das *klassische Akaike-Informationenkriterium* (AIC, engl. Akaike's information criterion) definiert. Hierbei bezeichnet L die Likelihood-Funktion und $(|\boldsymbol{\theta}|)$ ist die Anzahl zu schätzender Modellkoeffizienten (vgl. Hyndman und Athanasopoulos, 2018). Das AIC wurde entwickelt, um neben der Anpassungsgüte eines Modells, die in der Formel über die Likelihood-Funktion dargestellt wird und äquivalent zu einer Minimierung der Summe quadrierter Residuen ist, zusätzlich die Anzahl zu schätzender Modellkoeffizienten in die Bewertung des Modells mit einzubeziehen. Beim Vergleich mehrerer Modelle wird letztlich das Modell mit dem niedrigsten AIC-Wert ausgewählt, denn dieses bietet den besten Trade-off zwischen einer möglichst hohen Anpassungsgüte und einer möglichst niedrigen Anzahl an Modellkoeffizienten. Da das AIC für eine geringe Anzahl an Zeitreihenwerten T dazu neigt, ein Modell mit zu vielen Modellkoeffizienten zu selektieren, wird meist das korrigierte Kriterium AICc verwendet. Für weitere Infos zu Informationskriterien, siehe (Akaike, 1973, 1974; Sugiura, 1978; Hurvich und Tsai, 1989; Burnham et al., 2002).

Die `auto.arima`-Funktion betrachtet nicht alle möglichen Kombinationen der Modellordnungen, sondern es werden schrittweise sinnvolle ARIMA-Modelle so lange getestet, bis kein besserer Wert des AICc gefunden wird. Am Ende wird das Modell mit dem geringsten AICc-Wert verwendet. Für weitere Informationen zu diesem Modellselektionsprozess siehe Hyndman und Athanasopoulos (2018) und Hyndman et al. (2018b).

Prognose mit ARIMA-Modellen

Zur Nutzung eines ARIMA-Modells für eine Einschrittprognose werden die folgenden drei Schritte durchgeführt: (i) Umstellung der Gleichung (3.50), sodass y_t auf der einen Seite steht und alle anderen Terme auf der anderen Seite, (ii) Ersetzung von t durch $t + h$, (iii) Ersetzung zukünftiger Fehler durch Null und vergangener Fehler durch Residuen (vgl. Hyndman und Athanasopoulos, 2018). Eine h -Schritt-Prognose wird iterativ bestimmt, indem diese drei Schritte mehrfach ausgeführt werden und die unbekannt zukünftigen Werte jeweils durch die Prognosen ersetzt werden.

3.4.3 Exponentielle Glättungsmodelle

Übersicht

Exponentielle Glättung (engl. exponential smoothing) ist eine Modellklasse, die Prognosen als gewichtete Mittelwerte vergangener Zeitreihenwerte ermittelt, wobei die Gewichte älterer Werte

mit der Zeit exponentiell fallen (vgl. Hyndman et al., 2008b; Hyndman und Athanasopoulos, 2018). Damit erhalten aktuellere Zeitreihenwerte die höchsten Gewichte bei der Mittelwertbildung. Die Selektion eines exponentiellen Glättungsmodells basiert auf der Identifikation der Hauptkomponenten, die eine Zeitreihenentwicklung beeinflussen, wie Trends und saisonale Einflüsse, sowie der Einschätzung, wie diese zueinander in Verbindung stehen. Exponentielle Glättungsmodelle wurden in den späten 1950er Jahren eingeführt (vgl. Holt, 1957; Brown, 1959; Winters, 1960) und gehören seitdem zu den am häufigsten verwendeten Zeitreihenprognoseverfahren (vgl. Trigg und Leach, 1967; Pegels, 1969; Gardner und McKenzie, 1985; Gardner, 1999; Assimakopoulos und Nikolopoulos, 2000; Goodrich, 2000; Makridakis und Hibon, 2000; Hyndman et al., 2002; Taylor, 2003; De Gooijer und Hyndman, 2006; Gardner, 2006; Hyndman und Khandakar, 2008; Kolassa, 2011; Bergmeir et al., 2016; Kück et al., 2016b; Kourentzes et al., 2019; Hyndman, 2020; Smyl, 2020). In der vorliegenden Arbeit wird die `ets`-Funktion des `forecast`-Packages in R verwendet, um die Parameter eines exponentiellen Glättungsmodells basierend auf einer gegebenen Zeitreihe zu schätzen (vgl. Hyndman und Khandakar, 2008; Hyndman et al., 2018b). Im Folgenden wird das Prinzip exponentieller Glättung exemplarisch am Beispiel der einfachen exponentiellen Glättung erläutert und es werden andere wichtige Modellformen kurz dargestellt. Daraufaufgehend wird die Funktionsweise der `ets`-Funktion zur Modellselektion beschrieben. Für weitere Informationen sei auf die ausführliche Literatur verwiesen (vgl. Hyndman et al., 2008b; Hyndman und Athanasopoulos, 2018).

Theoretische Grundlagen der einfachen exponentiellen Glättung

Die simpelste Form exponentieller Glättungsmodelle wird als *einfache exponentielle Glättung* (ses, engl. simple exponential smoothing) bezeichnet (vgl. Holt, 1957; Brown, 1959; Hyndman et al., 2008b; Hyndman und Athanasopoulos, 2018). Die Modellierung basiert auf der Annahme, dass eine gegebene Zeitreihe keinen Trend und keine Saisonalität aufweist, sondern lediglich um einen Mittelwert schwankt. Dabei beschreitet ein ses-Modell einen Mittelweg zwischen einem Random Walk und einem gleitenden Mittelwert. Während ein nicht-saisonaler Random Walk einen zukünftigen Wert als den letzten Wert der Zeitreihe prognostiziert und frühere Zeitreihenentwicklungen unberücksichtigt lässt, bildet ein gleitender Mittelwert der Ordnung q den Durchschnitt der letzten q Zeitreihenwerte und misst allen Werten die gleiche Bedeutung bei. Im Gegensatz dazu bestimmt ein ses-Modell eine Prognose als gewichteten Mittelwert vergangener Zeitreihenwerte als

$$\hat{m}_h^{\text{ses}}(y_t) = \hat{y}_{t+h}^{\text{ses}} = \alpha y_t + \alpha(1 - \alpha)y_{t-1} + \alpha(1 - \alpha)^2 y_{t-2} + \dots = \alpha y_t + (1 - \alpha)\hat{y}_{t-1|t}^{\text{ses}}, \quad (3.56)$$

wobei die Gewichte älterer Werte mit der Zeit exponentiell fallen. Der sogenannte *Glättungsparameter* (engl. smoothing parameter) $\alpha \in [0, 1]$ bestimmt dabei, wie schnell die Gewichte fallen. Für einen großen Wert von α in der Nähe von 1 ist das Gewicht des letzten Zeitreihenwertes hoch, während ältere Werte kaum in die Prognose einbezogen werden. Für einen kleinen Wert von α in

der Nähe von 0 erhalten ältere Werte größere Gewichte. Wie in (3.56) und in Abbildung 3.8 zu sehen ist, sind die h -Schritt-Prognosen eines ses -Modells konstant, denn alle prognostizierten Werte entsprechen dem Wert der Einschrittprognose.

Vergleich verschiedener Formen exponentieller Glättungsmodelle

Durch Einführung weiterer Parameter in (3.56) sind exponentielle Glättungsmodelle dazu in der Lage, auch Trends und Saisonalitäten zu modellieren. Basierend auf den Arbeiten von Pegels (1969), Gardner und McKenzie (1985), Hyndman et al. (2002) und Taylor (2003) stellten Hyndman und Khandakar (2008) eine umfassende Taxonomie aller möglichen exponentiellen Glättungsmodelle vor (siehe Tabelle 3.2). Im Folgenden werden die wichtigsten Formen exponentieller Glättungsmodelle genannt. In Abbildung 3.8 sind Beispielprognosen dieser Modelle dargestellt, die mit Hilfe verschiedener Funktionen des `forecast`-Packages für `R` bestimmt wurden (vgl. Hyndman und Khandakar, 2008; Hyndman et al., 2018b).

Trendkomponente		Saisonkomponente		
		N (keine)	A (additiv)	M (multiplikativ)
N	(keine)	N,N	N,A	N,M
A	(additiv)	A,N	A,A	A,M
A _d	(additiv gedämpft)	A _d ,N	A _d ,A	A _d ,M
M	(multiplikativ)	M,N	M,A	M,M
M _d	(multiplikativ gedämpft)	M _d ,N	M _d ,A	M _d ,M

Tabelle 3.2: Taxonomie verschiedener exponentieller Glättungsmodelle nach Hyndman und Khandakar (2008)

Durch Verwendung der `ses`-Funktion des `forecast`-Packages lassen sich sinnvolle Parameter eines einfachen exponentiellen Glättungsmodells schätzen und Prognosen berechnen. Wie bereits beschrieben wurde, ist diese Modellform lediglich dazu in der Lage, eine Schwankung um einen Mittelwert in Form einer *Niveaugleichung* (engl. level equation) zu modellieren, während Trends und Saisonalitäten nicht modelliert werden können. Zur Einbeziehung eines Trends in die Modellierung kann *Holts lineare exponentielle Glättung* durch die Funktion `holt` angewandt werden. Hierbei wird zusätzlich zu einer Niveaugleichung eine *Trendgleichung* (engl. trend equation) genutzt. Durch Setzung des Parameters `damped=TRUE` der `holt`-Funktion lässt sich eine *exponentielle Glättung mit gedämpftem Trend* (dt) durchführen. Wie Abbildung 3.8 zeigt, sind die Prognosen dieser beiden Modellformen lineare Trendlinien, wobei der Betrag der Steigung im gedämpften Fall geringer ist. Diese Dämpfung hat insbesondere bei längerfristigen Prognosen in der Regel einen positiven Effekt auf die Prognosegüte (vgl. Gardner und McKenzie, 1985; Hyndman et al., 2008b). Zusätzlich zu einem Niveau und einem Trend lassen sich mit der

saisonalen exponentiellen Glättung nach Holt-Winters auch Saisonalitäten im Rahmen einer *Saisonalitätsgleichung* (engl. seasonality equation) abbilden. Die Prognosen können mit der `hw`-Funktion bestimmt werden und sind ebenfalls in Abbildung 3.8 dargestellt. Für weitere Informationen zur gesamten Modellklasse der exponentiellen Glättung siehe Hyndman et al. (2008b).

Selektion eines exponentiellen Glättungsmodells

Zur Selektion eines geeigneten exponentiellen Glättungsmodells aus Tabelle 3.2 müssen die Glättungsparameter sowie Initialzustände der Modelle sinnvoll an eine gegebene Zeitreihe angepasst werden. Hierfür wird in der vorliegenden Arbeit die `ets`-Funktion des `forecast`-Paketes in R verwendet (vgl. Hyndman und Khandakar, 2008; Hyndman et al., 2018b). Diese nutzt die in Hyndman et al. (2008b) beschriebene Formulierung aller Formen exponentieller Glättungsmodelle als *Zustandsraummodelle* (engl. state space models). Dabei wird jedes exponentielle Glättungsmodell durch eine *Messgleichung* (engl. measurement equation), die die beobachteten Zeitreihenwerte beschreibt, und einige *Zustandsgleichungen* (engl. state equations) beschrieben, die die Veränderungen unbekannter Komponenten und Zustände, wie Niveaus, Trends und Saisonalitäten, über die Zeit beschreiben. Durch diese Modellierung lässt sich, ähnlich wie bei ARIMA-Modellen, eine Likelihood für jedes exponentielle Glättungsmodell bestimmen und danach können verschiedene Modelle anhand eines Informationskriteriums verglichen werden. Hierzu verwendet die `ets`-Funktion das korrigierte Akaike-Informationskriterium AICc, welches in (3.54) beschrieben wurde, und wählt das Modell mit minimalem AICc-Wert. Für weitere Informationen zur Verwendung der `ets`-Funktion zur Selektion eines geeigneten exponentiellen Glättungsmodells für eine gegebene Zeitreihe, siehe Hyndman und Khandakar (2008) und Hyndman et al. (2008b).

3.4.4 Lokale k -nächste-Nachbarn-Modelle

Im Folgenden werden die notwendigen Schritte beschrieben, um ein lokales k -nächste-Nachbarn-Prognosemodell (knn) zu bilden. Da im Gegensatz zu anderen Prognoseverfahren, wie der exponentiellen Glättung oder ARIMA-Modellen, noch kein etablierter Trainingsalgorithmus zur Bestimmung geeigneter Modellparameter für lokale knn -Modelle existiert, wurden im Vorfeld dieser Arbeit Vergleichsstudien durchgeführt, um geeignete Modellselektionsmethoden zu identifizieren. In diesem Zusammenhang wurden dynamische Selektionsmethoden für knn -Modelle entwickelt, die großes Potenzial zeigten und daher in der vorliegenden Arbeit verwendet werden. Da die Modellselektionsmethoden bislang unpubliziert sind und knn -Prognosemodelle generell weniger etabliert sind als andere Prognoseverfahren, werden diese im Folgenden ausführlich beschrieben. Nach einer generellen Übersicht wird der Ablauf zum Training und der Verwendung von knn -Prognosemodellen beschrieben. Darauf folgend werden die einzelnen Schritte detailliert erläutert.

Übersicht

Lokale knn -Modelle gehören zu den Methoden der nichtlinearen Dynamik, welche aus der Theorie dynamischer Systeme stammen und auf dem Paradigma des deterministischen Chaos beruhen. Generelle Einführungen zu lokalen knn -Modellen sind in Kantz und Schreiber (2004), Aguirre und Letellier (2009) oder Broer und Takens (2011) zu finden. Neben Anwendungen zur Modellierung, Analyse und Steuerung von Produktionssystemen (vgl. Ramadge, 1993; Prabhu und Duffie, 1995; Wiendahl und Scheffczyk, 1999; Scholz-Reiter et al., 2002; Freitag, 2005; Katzorke und Pikovsky, 2006; Cho und Erkoç, 2009; Papakostas et al., 2009) und Liefernetzwerken (vgl. Wilding, 1998; Helbing, 2003; Surana et al., 2005; Scholz-Reiter et al., 2006; Donner et al., 2008; Hwang und Xie, 2008; Freitag et al., 2015a; Chankov et al., 2016; Klug, 2016; Kück et al., 2016a) wurden Methoden der nichtlinearen Dynamik auch in Prognoseanwendungen in der Form lokaler knn -Modelle verwendet. Der knn -Ansatz wurde in einem technischen Bericht von Fix und Hodges (1951) als ein Klassifikationsansatz im Kontext der nichtparametrischen Diskriminanzanalyse eingeführt. Da dieser Bericht jedoch zunächst nicht veröffentlicht wurde, erlangte der Ansatz erst durch die Arbeit von Cover und Hart (1967) Popularität. Später wurde der Ansatz in den beiden Wissenschaftsbereichen der Statistik (vgl. Stone, 1977; Cleveland, 1979; Yakowitz, 1987) und der dynamischen Systeme (vgl. Farmer und Sidorowich, 1987; Casdagli, 1989; Abarbanel et al., 1990; Sugihara und May, 1990) simultan auf Regressions- und Zeitreihenprognoseprobleme erweitert. Darauf folgend wurde der Ansatz von verschiedenen Autoren weiterentwickelt, beispielsweise zu lokal linearen Modellen basierend auf gefilterten Delay-Koordinaten-Vektoren (vgl. Sauer, 1994) oder lokal linearen Modellen mit Regularisierung (vgl. Kugiumtzis et al., 1998). Des Weiteren wurden Ansätze basierend auf nächsten Trajektorien als Alternative zu nächsten Nachbarn entwickelt (vgl. McNames, 1998) und Modellselektionsansätze entwickelt (vgl. Bontempi et al., 1999; McNames, 2002; Jayawardena et al., 2002; Yankov et al., 2006; She und Yang, 2010; Kück und Freitag, 2021). Trotz der Vielfalt an Anwendungen lokaler knn -Prognosemodelle existiert bislang kein etablierter Ansatz zur Bestimmung geeigneter Modellparameter. Um die in dieser Arbeit verwendeten dynamischen Modellselektionsmethoden einzuführen, werden im Folgenden die theoretischen Grundlagen lokaler knn -Modelle erläutert.

Abbildung 3.9 veranschaulicht den Prozess zur Prognose eines zukünftigen Wertes y_{T+h} einer gegebenen Zeitreihe $\mathbf{y} = \{y_1, \dots, y_T\}$ durch ein lokales knn -Prognosemodell. Für gegebene Modellparameter besteht dieser Prozess aus den Schritten der Delay-Koordinaten-Einbettung, der Suche nach k nächsten Nachbarn und der Bildung eines lokalen knn -Modells, welches dann zur Prognose eines zukünftigen Zeitreihenwertes verwendet wird.

Durch eine Delay-Koordinaten-Einbettung wird eine sogenannte Trajektorienmatrix $\mathbf{Y}_{r,d}$ gebildet, deren Zeilen zeitversetzte Segmente der Originalzeitreihe enthalten, welche auch als Delay-Koordinaten-Vektoren bezeichnet werden. Die Zeilenlänge r wird Einbettungsdimension genannt und der Parameter d wird als Zeitverschiebung bezeichnet. Beispielsweise enthält ein Delay-Koordinaten-Vektor jeden sechsten Wert einer Originalzeitreihe, falls $d = 6$. Zur

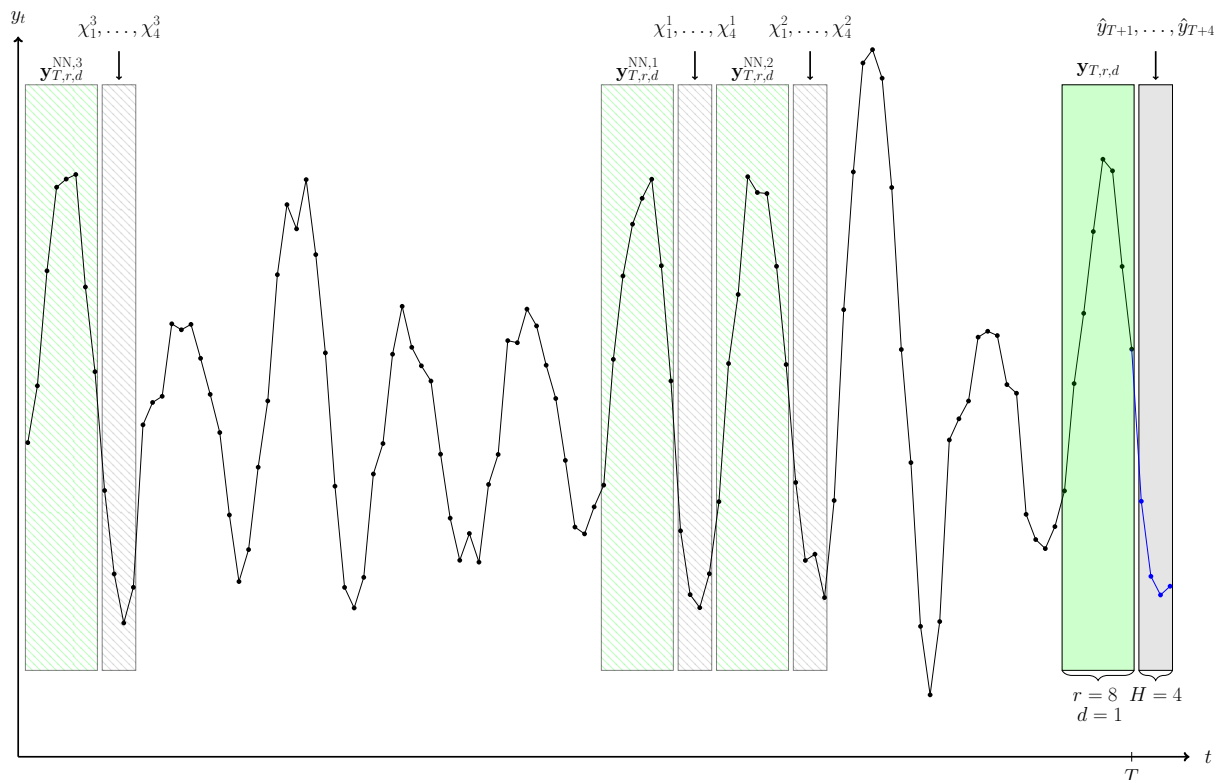


Abbildung 3.9: Prognose durch ein lokales k -nächste-Nachbarn-Modell

Veranschaulichung basiert Abbildung 3.9 auf Delay-Koordinaten-Vektoren mit einer Länge $r = 8$ und ohne Zeitverschiebung ($d = 1$).

Nach der Erzeugung einer Trajektorienmatrix werden die nächsten Nachbarn eines Abfragevektors ermittelt. Der Abfragevektor besteht aus den Punkten im grünen Kasten in Abbildung 3.9 und die nächsten Nachbarn sind als grün schraffierte Kästen dargestellt. Der Abfragevektor ist der letzte Delay-Koordinaten-Vektor der Trajektorienmatrix und er endet mit dem letzten Wert der Originalzeitreihe y_T . Die nächsten Nachbarn sind die Delay-Koordinaten-Vektoren, die dem Abfragevektor am ähnlichsten sind. In dieser Arbeit wird die Ähnlichkeit zweier Vektoren durch die euklidische Distanz bestimmt. In Abbildung 3.9 sind $k = 3$ nächste Nachbarn zum Abfragevektor abgebildet.

Da die zukünftigen Entwicklungen der gefundenen nächsten Nachbarn bekannt sind (grau schraffierte Kästen in Abbildung 3.9), lässt sich ein lokales k nn-Prognosemodell basierend auf den nächsten Nachbarn und diesen zukünftigen Entwicklungen bilden. In Abbildung 3.9 ist die Anwendung eines lokal konstanten k nn-Modells dargestellt, welches die Mittelwerte der zukünftigen Entwicklungen der nächsten Nachbarn als Prognosen verwendet. Zusätzlich zu lokal konstanten k nn-Modellen werden in dieser Arbeit auch lokal lineare k nn-Regressionsmodelle verwendet, die Prognosen auf Basis linearer Regressionsmodelle der nächsten Nachbarn und ihrer zukünftigen Entwicklungen bestimmen. Zur intuitiven Visualisierung zeigt Abbildung 3.9 direkte Einschrittprognosen der Horizonte $h = 1, \dots, 4$ basierend auf genau einer Delay-Koordinaten-

Einbettung und einer Suche nach k nächsten Nachbarn. Allerdings werden in dieser Arbeit nicht direkte, sondern iterative k nn-Prognoseverfahren verwendet, welche die Schritte der Delay-Koordinaten-Einbettung, der Suche nach nächsten Nachbarn und der Modellbildung für jeden zu prognostizierenden Wert wiederholen.

Ablauf

Abbildung 3.10 veranschaulicht die Schritte zur Prognose eines zukünftigen Zeitreihenwertes durch ein lokales k nn-Modell in einem Ablaufdiagramm. Gegeben ist eine Zeitreihe vergangener Kundenbestellungen $\mathbf{y} = \{y_1, \dots, y_T\}$. Eine Modellordnung (*order*) für das zu bildende Prognosemodell kann entweder als 0 (lokal konstant) oder als 1 (lokal linear) vorgegeben werden. H definiert den maximalen Prognosehorizont und h bezeichnet den Index des aktuellen Prognosehorizontes in der Schleife von Abbildung 3.10. Da in dieser Arbeit industrielle Daten mit regelmäßiger Nachfrage betrachtet werden, wird der Parameter *round* auf 1 gesetzt, was bedeutet, dass jeder prognostizierte Wert auf eine ganze Zahl gerundet wird. Aus demselben Grund, werden die Grenzen (*limits*) eines validen Prognosewertes auf das Intervall $[1, \infty)$ festgelegt. Falls ein Prognosewert außerhalb dieses Intervalls liegt, wird ein alternatives Prognosemodell (*alternative-function*) verwendet. In diesem Fall wird ein Random Walk verwendet (siehe Abschnitt 3.4.1) und somit wird der letzte Wert der Zeitreihe als Prognose für den zukünftigen Wert gesetzt.

Wie in Abbildung 3.10 zu sehen ist, besteht der Ablauf zur Bestimmung von Prognosen durch ein lokales k nn-Modell aus fünf Schritten. Der erste Schritt ist die Durchführung einer Hyperparameterselktion basierend auf den Eingabedaten. Es werden die Zeitverschiebung d , die Einbettungsdimension r sowie der Prozentwert der maximal möglichen zu berücksichtigen nächsten Nachbarn $k\%$ bestimmt, wie am Ende von Abschnitt 3.4.4 beschrieben wird (siehe (3.79)). Die Parameter *regtype* und *regpar* werden nur im Falle eines lokal linearen Modells (*order*=1) berücksichtigt. *regtype* spezifiziert den gewählten Typ eines Regularisierungsverfahrens und kann folgende Werte annehmen: ‘none’ (gewöhnliche lineare Regression ohne Regularisierung), ‘PCR’ (Hauptkomponentenregression), ‘PLSR’ (partielle Regression der kleinsten Quadrate), ‘RR’ (Ridge-Regression) oder ‘LASSO’ (Lasso-Regression). *regpar* definiert den Regularisierungsparameter \tilde{r} (für PCR und PLSR) oder λ (für RR und LASSO). *regpar* wird ignoriert, falls *regtype* als ‘none’ spezifiziert wird. Zur Selektion geeigneter Modellparameter werden am Ende von Abschnitt 3.4.4 bislang unpublizierte dynamische Modellselektionsmethoden vorgestellt. Basierend auf den Ergebnissen vorheriger Vergleichsstudien (siehe Anhang C.2) werden in der vorliegenden Arbeit lokal konstante k nn-Prognosemodelle mit $d = 1$, $r\% = 1$ und $k\% = 0.25$ sowie lokal lineare k nn-Ridge-Regressionsmodelle mit $d = 1$, $r\% = 1$, $k\% = 1$ und $\lambda = 100$ verwendet.

Nach dem Schritt (i) der Hyperparameterselktion wird eine Schleife durchlaufen, in der vier weitere Schritte iterativ wiederholt werden, um die Prognosewerte \hat{y}_{T+h} für $h = 1, \dots, H$ zu bestimmen, nämlich: (ii) Bestimmung der Anzahl der zu berücksichtigen nächsten Nachbarn, (iii) Erzeugung einer Trajektorienmatrix durch Delay-Koordinaten-Einbettung, (iv) Bestimmung der

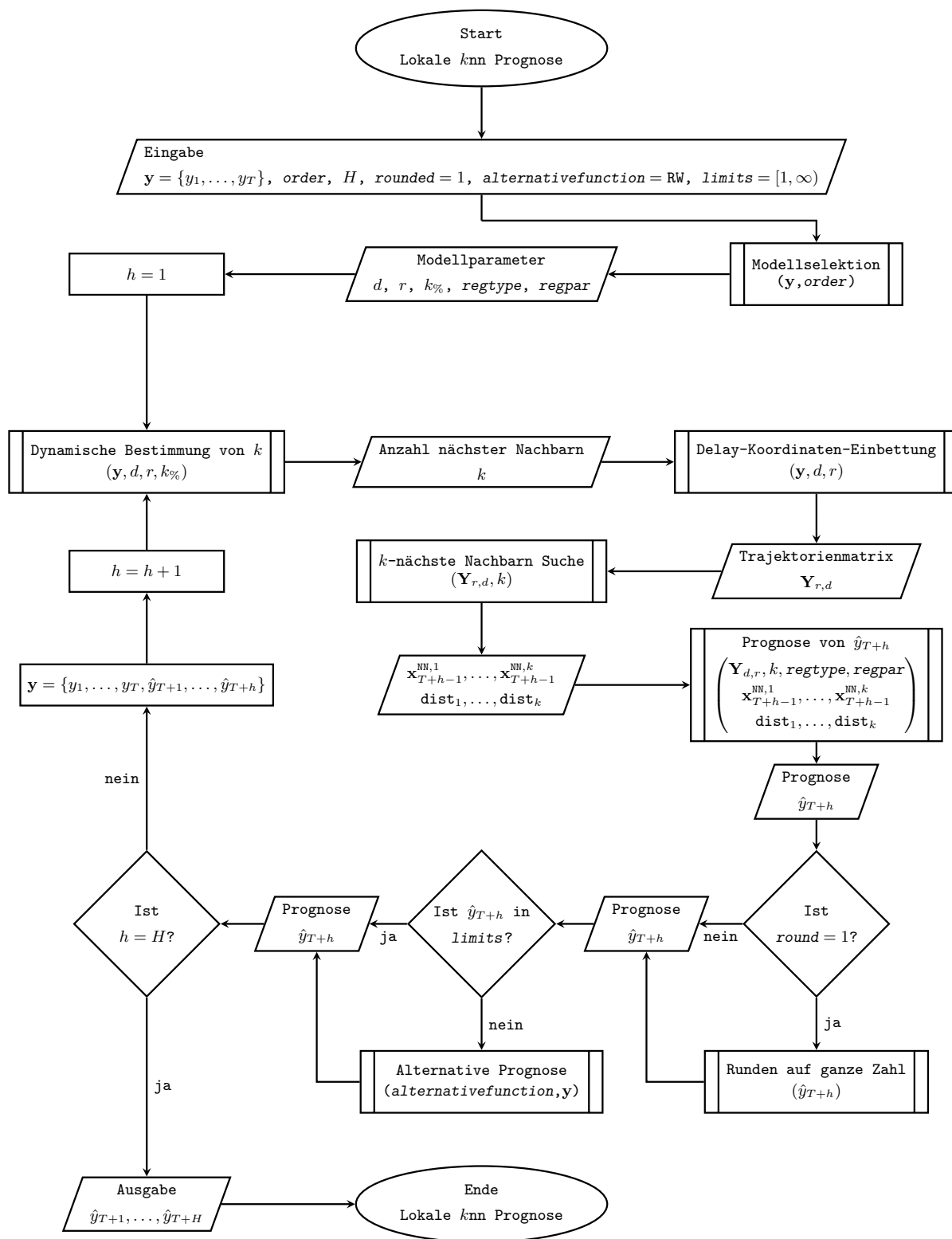


Abbildung 3.10: Ablaufdiagramm zur Prognose zukünftiger Werte einer Zeitreihe durch ein lokales k -nächste-Nachbarn-Modell

k nächsten Nachbarn zum Abfragevektor und (v) Bildung eines lokalen k nn-Modells, welches letztlich zur Prognose des zukünftigen Wertes \hat{y}_{T+h} verwendet wird. Im Nachgang wird geprüft, ob der prognostizierte Wert valide ist. Der Wert wird auf eine ganze Zahl gerundet und falls er nicht innerhalb des Intervalls $[1, \infty)$ liegt, wird eine Random-Walk-Prognose durchgeführt. Falls $h < H$, wird der prognostizierte Wert \hat{y}_{T+h} der Zeitreihe \mathbf{y} hinzugefügt, h wird um 1 erhöht und die nächste Iteration beginnt. Wenn $h = H$, endet die Schleife und die Prognosewerte $\hat{y}_{t+1}, \dots, \hat{y}_{t+H}$ werden ausgegeben.

Im Folgenden werden die fünf Schritte zur Ermittlung von Prognosen durch ein lokales k nn-Modell erläutert. Aus Gründen der besseren Verständlichkeit werden zunächst die Schritte (iii)-(v) erläutert und darauffolgend die Schritte (i) und (ii) der Modellselektion.

Delay-Koordination-Einbettung

Basierend auf einer geeigneten Selektion der Modellparameter (siehe (3.78) und (3.79)) ist der dritte Schritt zur Bildung eines lokalen k nn-Prognosemodells die *Delay-Koordinaten-Einbettung* (auch Zeitverschiebungsmethode, engl. delay coordinate embedding), welche auf der Theorie der nichtlinearen Dynamik basiert und auch als *Phasenraumrekonstruktion* (engl. phase space reconstruction) bezeichnet wird (vgl. Kantz und Schreiber, 2004). Wie in Abschnitt 2.3.3 erläutert, lässt sich die Entwicklung eines dynamischen Systems durch ein System gewöhnlicher Differentialgleichungen (2.1) beschreiben. Für dissipative dynamische Systeme gilt, dass sich nach einer gewissen Zeit eine Menge von Anfangsbedingungen auf eine Teilmenge des Phasenraums zusammenzieht. Diese Teilmenge $\mathcal{Y}'' \subseteq \mathcal{Y}'$ wird als *Attraktor* des Systems bezeichnet. Der Attraktor enthält die relevanten dynamischen Eigenschaften des gesamten Systems. Im Allgemeinen hat ein Attraktor eine geringere Dimension als der gesamte Phasenraum.

Ein dynamisches System wird von diversen Komponenten beeinflusst. Um die Entwicklung eines dynamischen Systems im Detail zu beschreiben, müssen alle Abhängigkeiten bekannt sein, was in der Regel eine unrealistische Annahme ist. Häufig sind lediglich Messwerte einer einzelnen Komponente eines dynamischen Systems gegeben. In dieser Arbeit wird nur vorausgesetzt, dass eine sukzessive, äquidistante, skalare Zeitreihe vergangener Kundenbestellungen $\mathbf{y} = \{y_1, y_2, \dots, y_T\}$ bis zum gegenwärtigen Zeitpunkt T gegeben ist. Die Nachfrageentwicklung stellt eine Komponente des dynamischen Systems dar, welches die Entwicklung des Gesamtsystems des Kunden eines produzierenden Unternehmens beschreibt. Unter recht allgemeinen Annahmen folgt jedoch aus dem Einbettungstheorem von Takens (vgl. Takens, 1981) und der Generalisierung (vgl. Sauer et al., 1991), dass unter der Vermutung eines deterministischen dynamischen Systems die Entwicklung des gesamten Systems aus der skalaren Zeitreihe einer Komponente des Systems rekonstruiert werden kann. Ein Vektor

$$\mathbf{y}_t = \mathbf{y}_{t,r,d} = (y_{t-(r-1)d}, y_{t-(r-2)d}, \dots, y_{t-d}, y_t) \in \mathbb{R}^{1 \times r} \quad (3.57)$$

wird als *Delay-Koordinaten-Vektor* (engl. delay coordinate vector) der Länge r mit Zeitverschiebung d bis zum Zeitpunkt t bezeichnet. Der Parameter r des Delay-Koordinaten-Vektors wird *Einbettungsdimension* (engl. embedding dimension) genannt und d wird als *Zeitverschiebung* (engl. delay time) bezeichnet. Ein Delay-Koordinaten-Vektor \mathbf{y}_t ist ein Segment der Originalzeitreihe \mathbf{y} . Während \mathbf{y} aus T Messungen besteht, enthält der Delay-Koordinaten-Vektor r der T Messungen mit sukzessivem Zeitabstand d . Die Einbettungsdimension r muss ausreichend groß gewählt werden, um eine hinreichende Entfaltung des Attraktors des dynamischen Systems sicherzustellen, jedoch nicht zu groß, um Redundanzen zu vermeiden, die lokale Prognosemodelle behindern könnten. Ähnlich dazu muss die Zeitverschiebung d einerseits groß genug gewählt werden, um zu starke Korrelationen zwischen sukzessiven Elementen von Delay-Koordinaten-Vektoren zu vermeiden. Andererseits sollte d nicht zu groß gewählt werden, um eine komplette Unabhängigkeit sukzessiver Elemente zu vermeiden. In dieser Arbeit wird $d = 1$ genutzt, der Wert für r wird als relativer Wert zur maximal möglichen Einbettungsdimension zum Zeitpunkt T gewählt und k wird dynamisch in jedem Iterationsschritt gewählt, wie am Ende von Abschnitt 3.4.4 beschrieben wird (siehe (3.78) und (3.79)).

Unter den Voraussetzungen, dass die Zeitreihe \mathbf{y} durch ein glattes, deterministisches dynamisches System generiert wurde und die Einbettungsparameter r und d geeignet gewählt sind, rechtfertigen die Einbettungstheoreme (vgl. Takens, 2010; Kantz und Schreiber, 2004) eine glatte funktionale Abhängigkeit

$$y_{t+1} = \tau_1(\mathbf{y}_t) = \tau_1(y_{t-(r-1)d}, y_{t-(r-2)d}, \dots, y_{t-d}, y_t). \quad (3.58)$$

Daher kann ein Prognosemodell für zukünftige Werte der gegebenen Zeitreihe basierend auf der Delay-Koordinaten-Einbettung gebildet werden. Zu diesem Zweck werden alle Delay-Koordinaten-Vektoren der Länge r mit sukzessivem Abstand d aus der gegebenen Zeitreihe \mathbf{y} gebildet und in einer sogenannten *Trajektorienmatrix* (engl. trajectory matrix) zusammengefasst

$$\mathbf{Y}_{r,d} = \begin{pmatrix} y_1 & y_{1+d} & \cdots & y_{1+(r-2)d} & y_{1+(r-1)d} \\ y_2 & y_{2+d} & \cdots & y_{2+(r-2)d} & y_{2+(r-1)d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{t-(r-1)d} & y_{t-(r-2)d} & \cdots & y_{t-d} & y_t \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_t & y_{t+d} & \cdots & y_{t+(r-2)d} & y_{t+(r-1)d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{T-(r-1)d} & y_{T-(r-2)d} & \cdots & y_{T-d} & y_T \end{pmatrix}. \quad (3.59)$$

Der *Abfragevektor* (engl. query vector)

$$\mathbf{y}_T = \mathbf{y}_{T,r,d} = (y_{T-(r-1)d}, y_{T-(r-2)d}, \dots, y_{T-d}, y_T) \quad (3.60)$$

ist der Delay-Koordinaten-Vektor bis zum gegenwärtigen Zeitpunkt. Mit der Aufstellung einer Trajektorienmatrix wurden die dynamischen Eigenschaften einer gegebenen Zeitreihe rekonstruiert.

Suche nach k -nächsten Nachbarn und Gewichtung

Der vierte Schritt des beschriebenen Prognoseverfahrens der lokalen k nn-Modelle ist die Suche nach den k nächsten Nachbarn eines Abfragevektors \mathbf{y}_T , dem letzten Vektor der Trajektorienmatrix $\mathbf{Y}_{r,d}$. Die anderen $T - (r - 1)d - 1$ Vektoren der Trajektorienmatrix werden hinsichtlich ihrer Distanz zum Abfragevektor \mathbf{y}_T sortiert. In diesem Zusammenhang können verschiedene Distanzmetriken, wie die euklidische Distanz, die diagonal gewichtete euklidische Distanz oder jegliche p -Norm genutzt werden (vgl. McNames, 1999; Engster, 2011). In dieser Arbeit wird die euklidische Distanz

$$\text{dist}(\mathbf{y}_T, \mathbf{y}_i) = \sum_{j=1}^r \left(y_{T-(r-j)d} - y_{i-(r-j)d} \right)^2 \quad (3.61)$$

verwendet. Nach der Sortierung werden die k nächsten Nachbarn $\mathbf{y}_T^{\text{NN},1}, \dots, \mathbf{y}_T^{\text{NN},k}$ von \mathbf{y}_T ermittelt, wobei der i -te nächste Nachbar der Delay-Koordinaten-Vektor mit der i -t-kürzesten Distanz dist_i zum Abfragevektor \mathbf{y}_T ist.

Zusätzlich zu den Einbettungsparametern r und d muss die Anzahl k der verwendeten nächsten Nachbarn geeignet gewählt werden. Im Allgemeinen führt ein kleiner Wert von k zu einer geringen Verzerrung eines Prognosemodells aber gleichzeitig zu einer hohen Varianz. Auf der anderen Seite führt ein hoher Wert von k zu einer geringen Varianz aber zu einer hohen Verzerrung (siehe Abschnitt 3.1.2). In dieser Arbeit wird k dynamisch bestimmt (siehe (3.79)).

Die k ermittelten nächsten Nachbarn können in einer Merkmalsmatrix

$$\mathbf{X} = \left(\mathbf{x}^1, \dots, \mathbf{x}^k \right)^\top = \left(\mathbf{y}_T^{\text{NN},1}, \dots, \mathbf{y}_T^{\text{NN},k} \right)^\top \quad (3.62)$$

zusammengefasst und darauffolgend verwendet werden, um ein Prognosemodell zu bilden. Zusätzlich bieten die berechneten Distanzen der Nachbarn zum Abfragevektor die Möglichkeit, die Wichtigkeit der Nachbarn gemäß ihrer Distanzwerte zu gewichten. In dieser Arbeit wird auf eine spezielle Gewichtung verzichtet und es werden gleiche Gewichte für alle Nachbarn verwendet

$$\mathbf{w}_{\text{equal}} = (1, \dots, 1) \in \mathbb{R}^k. \quad (3.63)$$

Ein weiterer gängiger Ansatz ist biquadratische Gewichtung der Nachbarn. Allerdings funktionierte diese Gewichtungsfunktion schlechter als der simplere Ansatz gleicher Gewichte in einer vorher durchgeführten Vergleichsstudie. Daher werden in dieser Arbeit gleiche Gewichte der Nachbarn verwendet.

Lokale k -nächste-Nachbarn-Prognosemodelle

Nach der Durchführung einer Delay-Koordinaten-Einbettung und der Suche nach k nächsten Nachbarn lässt sich ein knn -Prognosemodell bilden. Durch Verwendung der Trajektorienmatrix $\mathbf{Y}_{r,d}$, können die k nächsten Nachbarn h Schritte in die Zukunft im rekonstruierten Phasenraum abgebildet werden, um die Werte $\tau_h(\mathbf{y}_T^{\text{NN},1}), \dots, \tau_h(\mathbf{y}_T^{\text{NN},k})$ zu erhalten. Sei

$$\mathbf{y}_{T,r,d}^{\text{NN},i} = \mathbf{y}_T^{\text{NN},i} = \mathbf{y}_j = (y_{j-(r-1)d}, \dots, y_{j-d}, y_j). \quad (3.64)$$

der i -te nächste Nachbar des Abfragevektors. Dann bedeutet eine Abbildung von h Schritten in die Zukunft, den Wert

$$\tau_h(\mathbf{y}_T^{\text{NN},i}) = y_{j+h} \quad (3.65)$$

zu betrachten. Der Vektor, der die h -ten zukünftigen Werte der k nächsten Nachbarn enthält, wird als

$$\boldsymbol{\chi}_h = (\chi_h^1, \dots, \chi_h^k)^\top = (\tau_h(\mathbf{y}_T^{\text{NN},1}), \dots, \tau_h(\mathbf{y}_T^{\text{NN},k}))^\top. \quad (3.66)$$

bezeichnet. Durch Verwendung der zukünftigen Entwicklungen (3.66) der k nächsten Nachbarn, lassen sich verschiedene Prognosemodelle bilden, die die global nichtlineare Entwicklung lokal approximieren (vgl. Farmer und Sidorowich, 1987; Sauer, 1994; Kück und Freitag, 2021). In dieser Arbeit werden lokal konstante Prognosemodelle und lokal lineare Regressionsmodelle verwendet.

Lokal konstante k -nächste-Nachbarn-Modelle: Der simpelste Ansatz zur Bildung eines lokalen knn -Prognosemodells ist die Verwendung eines *lokal konstanten k -nächste-Nachbarn-Prognosemodells* (engl. locally constant k -nearest neighbor model), welches auch als *lokale Durchschnittsbildung* (engl. local averaging) bezeichnet wird (vgl. McNames, 2002). In diesem Zusammenhang ist der gängigste Ansatz die Verwendung des Stichprobenmittelwerts der Zukünfte der nächsten Nachbarn

$$\hat{m}_h^{\text{lcknn,mean}}(\mathbf{X}) = \hat{y}_{T+h}^{\text{lcknn,mean}} = \frac{1}{k} \sum_{i=1}^k \tau_h(\mathbf{y}_T^{\text{NN},i}) = \frac{1}{k} \sum_{i=1}^k \chi_h^i. \quad (3.67)$$

Alternativ lässt sich der Median der Zukünfte der Nachbarn bilden. Zu diesem Zweck wird der Vektor $\boldsymbol{\chi}_h$ der Zukünfte der nächsten Nachbarn sortiert um

$$\boldsymbol{\chi}'_h = (\chi_h^{(1)}, \chi_h^{(2)}, \dots, \chi_h^{(k)})^\top, \text{ mit } \chi_h^{(i)} \leq \chi_h^{(i+1)} \text{ für } i \in \{1, \dots, k-1\}. \quad (3.68)$$

zu erhalten. Der lokal konstante Median wird wie folgt berechnet:

$$\hat{m}_h^{\text{lcknn,median}}(\mathbf{X}) = \hat{y}_{T+h}^{\text{lcknn,median}} = \begin{cases} \chi_h^{(\frac{k+1}{2})}, & \text{falls } k \text{ gerade} \\ \frac{1}{2} (\chi_h^{(\frac{k}{2})} + \chi_h^{(\frac{k}{2}+1)}), & \text{falls } k \text{ ungerade.} \end{cases} \quad (3.69)$$

Lokal lineare k -nächste-Nachbarn-Regression: Im Gegensatz zu lokal konstanten k nn-Modellen bilden *lokal lineare k -nächste-Nachbarn-Regressionsmodelle* (engl. locally linear k -nearest neighbor regression models) ein Modell

$$\boldsymbol{\chi}_h = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\chi}_h = \begin{pmatrix} \chi_h^1 \\ \vdots \\ \chi_h^k \end{pmatrix} = \begin{pmatrix} \tau_h(\mathbf{y}_T^{\text{NN},1}) \\ \vdots \\ \tau_h(\mathbf{y}_T^{\text{NN},k}) \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & \mathbf{x}^1 \\ \vdots & \vdots \\ 1 & \mathbf{x}^k \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{y}_T^{\text{NN},1} \\ \vdots & \vdots \\ 1 & \mathbf{y}_T^{\text{NN},k} \end{pmatrix} \quad (3.70)$$

der Zukünfte der Nachbarn $\boldsymbol{\chi}_h \in \mathbb{R}^k$ und der Komponenten der Nachbarn $\mathbf{X} \in \mathbb{R}^{k \times (r+1)}$, wobei $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_r)^\top \in \mathbb{R}^{r+1}$ den Koeffizientenvektor und $\boldsymbol{\epsilon} \in \mathbb{R}^k$ ein weißes Rauschen bezeichnen. Nach der Berechnung eines geeigneten Koeffizientenvektors $\hat{\boldsymbol{\beta}}$, wird eine h -Schritt-Prognose als

$$\hat{m}_h^{\text{llknn}}(\mathbf{X}) = \hat{y}_{T+h}^{\text{llknn}} = \hat{\beta}_0 + \sum_{j=1}^r y_{T,j} \hat{\beta}_j = (\mathbf{1} \ \mathbf{y}_T) \hat{\boldsymbol{\beta}} \quad (3.71)$$

bestimmt, wobei $y_{T,j}$ die j -te Komponente des Abfragevektors \mathbf{y}_T bezeichnet und $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^r$. Aus dem Gauß-Markov-Theorem (vgl. Hastie et al., 2009) folgt, dass der beste lineare unverzerrte Schätzer des Koeffizientenvektors $\boldsymbol{\beta}$ derjenige Vektor ist, der die Summe der quadrierten Residuen

$$\text{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^k \left(\tau_h(\mathbf{y}_T^{\text{NN},i}) - \beta_0 - \sum_{j=1}^r y_{T,j} \beta_j \right)^2, \quad (3.72)$$

minimiert, wobei $y_{T,j}^{\text{NN},i}$ die j -te Komponente des i -ten nächsten Nachbarn $\mathbf{y}_T^{\text{NN},i}$ bezeichnet. Diese Methode wird als *gewöhnliche Methode der kleinsten Quadrate* (engl. ordinary least squares, OLS) bezeichnet, die den Koeffizientenvektor wie folgt bestimmt:

$$\hat{\boldsymbol{\beta}}^{\text{OLS}} = \underset{\boldsymbol{\beta}}{\text{argmin}} (\text{RSS}(\boldsymbol{\beta})) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\chi}_h. \quad (3.73)$$

In vielen Fällen von real beobachteten Zeitreihendaten können die Spalten der Matrix \mathbf{X} kollinear sein, was ein schlecht konditioniertes Matrixinverses von $\mathbf{X}^\top \mathbf{X}$ zur Folge hat und somit könnten einige Regressionskoeffizienten und der Modell-Output sehr groß sein (vgl. McNames, 1999). Um diesem Problem vorzubeugen und lokal lineare Modelle robuster zu gestalten, lassen sich Regularisierungsmethoden verwenden (vgl. Kugiumtzis et al., 1998; McNames, 1999; Kück et al., 2014). Diese Methoden verzerren den Schätzer der Regressionskoeffizienten, um die Varianz und den Fehler des Schätzers zu reduzieren und somit eine Überanpassung zu vermeiden (vgl. Abschnitt 3.1.2). Regularisierungsmethoden können in Methoden mit abgeleiteten Eingaberichtungen und Schrumpfmethoden unterteilt werden (vgl. Hastie et al., 2009). Die *Hauptkomponentenregression* (PCR, engl. principal component regression) und die *partielle Regression der kleinsten Quadrate* (PLSR, engl. partial least squares regression) nutzen *abgeleitete Eingaberichtungen* (engl. derived input directions) and lassen die Regression auf einer kleineren Anzahl von Linearkombinationen der Original-Inputs basieren (vgl. Kugiumtzis et al., 1998; Hastie

et al., 2009). *Schrumpfmethode*n (engl. shrinkage methods), wie die *Ridge-Regression* (RR) und die *LASSO-Regression* (LASSO, engl. least absolute shrinkage and selection operator method), verkleinern die Regressionskoeffizienten durch die Bestrafung hoher Werte der Koeffizienten (vgl. Tibshirani, 1996; Hastie et al., 2009).

Dynamische Modellselektionsmethoden

Lokale *knn*-Modelle wurden erfolgreich zur Zeitreihenprognose in verschiedenen Anwendungen eingesetzt und erzielten beispielsweise in mehreren Prognosewettbewerben vielversprechende Ergebnisse (vgl. Weigend und Gershenfeld, 1992; Suykens und Vandewalle, 1998; Crone et al., 2011). Allerdings existieren bislang keine zuverlässigen Methoden zur automatischen Selektion der Modellparameter d , r und k sowie einer geeigneten Regularisierungsmethode und der dazugehörigen Regularisierungsparameter im Fall lokal linearer Modelle. Stattdessen werden diese Parameter in der Regel manuell basierend auf Expertenwissen festgelegt. Im Folgenden werden Methoden zur automatischen Selektion der Parameter für lokale *knn*-Modelle vorgestellt.

Ein gängiger Ansatz zur Optimierung der Modellparameter, ist die Zeitreihenkreuzvalidierung (vgl. Abbildung 3.6), welche beispielsweise von McNames (1999) und Wichard (2011) verwendet wurde. Hierbei wird die gegebene Zeitreihe in eine Trainings- und eine Validierungsmenge geteilt. Mehrere zu vergleichende Parameterkonfigurationen werden verwendet, um Prognosemodelle anhand der Trainingsmenge zu trainieren. Diese Modelle werden verwendet, um die Werte der Validierungsmenge zu prognostizieren und anhand der Prognosefehler lassen sich die verschiedenen Parameterkonfigurationen vergleichen. Allerdings können Kreuzvalidierungsmethoden rechenintensiv sein, da häufig eine große Anzahl verschiedener Parameterkonfigurationen verglichen werden muss, um eine geeignete Konfiguration zu bestimmen. Anstelle einer vollständigen Rastersuche werden manchmal heuristische Suchverfahren, wie genetische Algorithmen verwendet (vgl. Engster, 2011; Kück und Scholz-Reiter, 2013), um gute Parameterkonfigurationen schneller zu finden. Je nach Wahl der Parameter für den Suchprozess können aber auch heuristische Suchverfahren zu einer hohen Anzahl zu vergleichender Parameterkonfigurationen führen, bis das Verfahren konvergiert und eine geeignete Lösung findet. Daher werden häufig generelle Faustregeln verwendet, um die Parameter lokaler Modelle zu bestimmen (vgl. Kantz et al., 2004). Die Zeitverschiebung d wird als erste Minimalstelle der *mittleren gemeinsamen Information* (AMI, engl. average mutual information) der Zeitreihe ermittelt (vgl. Fraser und Swinney, 1986), während die Einbettungsdimension r durch den *Algorithmus der falschen nächsten Nachbarn* (FNN, engl. false nearest neighbors algorithm) bestimmt wird (vgl. Kennel et al., 1992) und die Anzahl der nächsten Nachbarn als $k = r + 1$.

Unter Verwendung der Faustregeln zur Bestimmung der Parameter d , r und k wurden in einer Vergleichsstudie anhand der monatlichen Industriedaten der M3-Competition verschiedene Durchschnittsbildungen (Mittelwert und Median) für lokal konstante Modelle sowie verschiedene Regularisierungsmethoden (PCR, PLSR, RR und LASSO) verglichen (siehe Anhang C.1 sowie

Kück und Freitag, 2021). In der Studie erzielten lokal lineare Modelle mit starker Regularisierung, d.h. mit niedrigem Wert des Regularisierungsparameters für PCR und PLSR bzw. mit hohem Wert des Regularisierungsparameters für RR und LASSO, eine höhere Prognosegüte als lokal lineare Modelle mit geringer Regularisierung. Die beiden lokal konstanten Modelle erreichten ähnliche Ergebnisse zu den lokal linearen Modellen mit hoher Regularisierung, was intuitiv ist, da starke Regularisierung ein lineares Modell gegen ein konstantes Modell konvergieren lässt. Unter den vier verschiedenen Regularisierungsmethoden für lokal lineare Modelle erzielte die Ridge-Regression (LL RR) die besten Ergebnisse.

Basierend auf diesen Ergebnissen wurde eine zweite, bislang unpublizierte, Vergleichsstudie durchgeführt (siehe Anhang C.2), in der weitere Selektionsmethoden für lokal konstante Modelle und lokal lineare Ridge-Regressionsmodelle verglichen wurden. Für die Zeitverschiebung wurden die intuitiven Werte $d = 1, 3, 6$ oder 12 verwendet, die keine Saisonalität sowie eine vierteljährliche, halbjährliche oder jährliche Saisonalität bedeuten. Die Werte der Einbettungsdimension wurden prozentual zur maximal möglichen Einbettungsdimension basierend auf der gegebenen Zeitreihe bis zum Zeitpunkt T ermittelt. Zudem wurde eine neue *dynamische Selektionsmethode* für die Wahl der Anzahl zu berücksichtigender Nachbarn verwendet und mit klassischen statischen Methoden verglichen. Die Ergebnisse dieser Studie werden im Folgenden beschrieben.

Ein lokales knn -Modell basiert auf einer Trajektorienmatrix (3.59), die $T - (r - 1)d$ Zeilenvektoren enthält. Um sicherzustellen, dass mindestens ein nächster Nachbar zum Abfragevektor gefunden werden kann, muss die Trajektorienmatrix neben dem Eingabevektor mindestens einen weiteren Zeilenvektor enthalten. Somit gilt folgende Voraussetzung

$$T - (r - 1)d \geq 2. \quad (3.74)$$

Durch Umformung ergeben sich die folgenden Voraussetzungen für die Parameter basierend auf der gegebenen Zeitreihe bis zum Zeitpunkt T

$$1 \leq d \leq T - 2 =: d_{\max} \quad (3.75)$$

$$1 \leq r \leq \frac{T - 2}{d} + 1 =: r_{\max} \quad (3.76)$$

$$1 \leq k \leq T - (r - 1)d - 1 =: k_{\max}. \quad (3.77)$$

Da die möglichen Parameterbereiche aufeinander basieren, wird zunächst die Zeitverschiebung d gewählt, darauffolgend die Einbettungsdimension r und abschließend die Anzahl der nächsten Nachbarn k . Für die Zeitverschiebung sollte ein Wert gewählt werden, der einer möglichen Saisonalität einer gegebenen Zeitreihe entspricht, wie z.B. $d = 1$ (keine Saisonalität), $d = 3$ (vierteljährliche Saisonalität), $d = 6$ (halbjährliche Saisonalität) oder $d = 12$ (jährliche Saisonalität). Basierend auf der gewählten Zeitverschiebung d wird zunächst die Einbettungsdimension r und

daraufliegend die Anzahl der nächsten Nachbarn k gewählt. In der vorliegenden Arbeit werden diese beiden Parameter jeweils als prozentuale Anteile der maximal möglichen Werte gewählt als

$$r = \max\{r\% \lfloor r_{\max} \rfloor, 1\}, \text{ mit } 0 \leq r\% \leq 1 \quad (3.78)$$

$$k = \max\{k\% k_{\max}, 1\}, \text{ mit } 0 \leq k\% \leq 1. \quad (3.79)$$

Hierbei bezeichnet $\lfloor \cdot \rfloor$ die Abrundungsfunktion (3.47), welche genutzt wird, um auf die größte ganze Zahl abzurunden, die Bedingung (3.76) erfüllt. Die ermittelten Werte für d und r werden für die Berechnung der h -Schritt-Prognosen konstant gehalten, während k in der vorliegenden Arbeit in jedem Iterationsschritt neu bestimmt wird. Je nach Wahl der Parameter d und r kann es vorkommen, dass die Trajektorienmatrix nur wenige Zeilenvektoren enthält und somit nur wenige nächste Nachbarn zum Abfragevektor bestimmt werden können. Die dynamische Selektionsmethode nutzt die Tatsache, dass die Trajektorienmatrix in jedem Iterationsschritt einer h -Schritt-Prognose um einen Zeilenvektor und somit um einen weiteren möglichen nächsten Nachbarn erweitert wird. Dadurch vergrößert sich auch der maximal mögliche Wert k_{\max} und der Parameter k kann in jedem Iterationsschritt erneut durch (3.79) gewählt werden.

Die Ergebnisse der zweiten Studie zum Vergleich statischer und dynamischer Modellselektionsstrategien für lokale k nn-Modelle sind im Anhang C.2 zu finden. Es zeigt sich, dass eine dynamische Vergrößerung der Anzahl der nächsten Nachbarn k im Verlauf einer iterativen h -Schritt-Prognose zu signifikant besseren Prognosen führt als eine standardmäßige statische Definition dieses Parameters. Dies gilt sowohl für lokal lineare (LL) als auch für lokal konstante (LC) Modelle. Allerdings besteht in der Wahl des besten Prozentsatzes ein Unterschied zwischen diesen beiden Modellordnungen. Während für LL Modelle gilt, dass $k\%$ so groß wie möglich gewählt werden sollte, werden bei LC Modellen eher bessere Prognoseergebnisse erzielt, wenn kleinere Prozentsätze gewählt werden. Allerdings ist der Einfluss des Parameters $k\%$ auf die Prognosegüte bei LL Modellen deutlicher als bei LC Modellen. Hinsichtlich der Wahl der Zeitverschiebung führt die Konfiguration $d = 1$ zu den besten Ergebnissen. Eine mögliche Erklärung hierfür ist, dass die Methode der Delay-Koordinaten-Einbettung ursprünglich für längere Zeitreihen entwickelt wurde. Da die in dieser Arbeit betrachteten industriellen Zeitreihen höchstens 144 Werte umfassen, ist es sinnvoller, alle Werte zu verwenden, um die kompletten in der Zeitreihe enthaltenen Informationen rekonstruieren zu können. Ein ähnlich eindeutiges Ergebnis zeigt sich hinsichtlich der Einbettungsdimension. Hier werden die besten Prognoseergebnisse für eine maximale Größe von $r\% = 1$ erreicht. Dies bedeutet, dass die maximal mögliche Einbettungsdimension zum Zeitpunkt T ermittelt wird und dann für die iterative Prognose konstant gehalten wird. Insgesamt erzielte das dynamische LL-RR-Modell mit den Parametern $d = 1$, $r\% = 1$ und $k\% = 1$ sowie einem Regularisierungsparameter $\lambda = 100$ die besten Ergebnisse der Vergleichsstudie. Im Bereich der LC-Modelle erreichte das dynamische arithmetische Mittelwertmodell mit den Parametern $d = 1$, $r\% = 1$ und $k\% = 0,25$ die geringsten Prognosefehler. Daher werden diese beiden Modellkonfigurationen in den empirischen Studien der vorliegenden Arbeit genutzt.

3.4.5 Neuronale Netze

Übersicht

Ein *neuronales Netz* (NN, auch künstliches neuronales Netzwerk, engl. artificial neural network) ist ein Informationsverarbeitungssystem in Analogie an die Informationsverarbeitung in einem Gehirn eines Säugetiers (vgl. McCulloch und Pitts, 1943; Hebb, 1949; Rosenblatt, 1958; Rehkugler und Zimmermann, 1994; Bishop, 1995; Zell, 2003). Die Knotenpunkte eines neuronalen Netzes werden *Neuronen* (engl. neurons) genannt. Ein Neuron wird als elementare Einheit zur Verarbeitung von Informationen betrachtet. Durch die Verbindung mehrerer Neuronen über Synapsen wird ein neuronales Netz gebildet, welches komplexe Interaktionen von Informationskanälen verarbeiten kann. Über die letzten Dekaden wurden diverse Architekturen neuronaler Netze entwickelt. Die einfachste Form eines neuronalen Netzes ist ein sogenanntes *Perzeptron* (engl. perceptron, vgl. Rosenblatt, 1958), welches aus einer Schicht von Eingabeneuronen und einem einzelnen Ausgabeneuron besteht. Das Ausgabeneuron hat den Wert 1, wenn die gewichtete Linearkombination der Eingabewerte einen Schwellwert übersteigt und ansonsten den Wert 0. Dieses Vorgehen simuliert die Aktivierung oder Nichtaktivierung des Ausgabeneurons. Allerdings zeigten Minsky und Papert (1969), dass Perzeptrone nur linear separable Muster lernen können. Um nichtlineare Muster zu lernen, müssen weitere Schichten eingeführt werden, wodurch ein mehrschichtiges vorwärtsgerichtetes Netz, ein sogenanntes *Multilayer-Perzeptron* (MLP, engl. multilayer perceptron), entsteht (vgl. Abbildung 3.11 und Rumelhart et al., 1986b; Rehkugler und Zimmermann, 1994; Bishop, 1995; Zell, 2003). Ein MLP besteht aus mindestens drei Schichten von Neuronen, wobei die Neuronen einer Schicht jeweils nur mit den Neuronen der nächsten Schicht verbunden sind. Die Netze verallgemeinern das Vorgehen des Perzeptrons, indem die Eingabewerte jeder Schicht des Netzes erneut gewichtete Linearkombinationen der Ausgabewerte der vorherigen Schicht sind und zusätzlich in jeder Schicht eine Aktivierungsfunktion auf die Differenz zwischen Eingabewerten und Schwellwerten der Schicht angewandt wird, um die Ausgabewerte zu bestimmen. Neuronale Netzwerkarchitekturen mit höchstens drei Schichten werden *flache* (engl. shallow) neuronale Netze genannt, während Architekturen mit mehr als drei Schichten als *tiefe* (engl. deep) neuronale Netze bezeichnet werden (vgl. LeCun et al., 2015; Schmidhuber, 2015). Obwohl bereits flache MLP universale Approximatoren sind, die jede beschränkte, stetige Funktion in einem kompakten Raum annähern können, wenn die Netzarchitektur und die Netzgewichte geeignet gewählt sind (vgl. Kolmogorov, 1957; Cybenko, 1989; Hornik et al., 1989; Hornik, 1991), können tiefe Architekturen sinnvoll sein, wenn die zusätzlichen Schichten spezifische Aufgaben erfüllen. Tiefe neuronale Netze lassen sich in rekurrente neuronale Netze (engl. recurrent neural networks) und faltende neuronale Netze (engl. convolutional neural networks) unterteilen.

Rekurrente Netze, wie Hopfield-Netze (vgl. Hopfield, 1982), Jordan-Netze (vgl. Jordan, 1986), Elman-Netze (vgl. Elman, 1990), LSTM-Netze (engl. long short-term memory networks, vgl. Hochreiter und Schmidhuber, 1997) oder Echo-State-Netze (vgl. Jaeger und Haas, 2004) fügen

Rückkopplungsschleifen in die Netzarchitektur ein, wodurch die Netze eine Art Gedächtnis erhalten. Dadurch lassen sich diese zur Modellierung und Steuerung dynamischer Systeme und somit insbesondere zur Zeitreihenprognose verwenden (vgl. Smyl, 2020). Allerdings existieren bislang keine etablierten vollautomatischen Verfahren zum Training rekurrenter neuronaler Netze als Zeitreihenprognosemodelle (vgl. Hewamalage et al., 2021). Daher wird diese Klasse neuronaler Netze in der vorliegenden Arbeit nicht betrachtet. Stattdessen werden zwei unterschiedliche automatische Prognosealgorithmen verwendet, um flache vorwärtsgerichtete Netze als Zeitreihenprognosemodelle zu trainieren: die `mlp`-Funktion des `nnfor`-Packages (im Folgenden als `nnfor` bezeichnet, vgl. Kourentzes, 2017) und die `nnetar`-Funktion des `forecast`-Packages (im Folgenden als `nnetar` bezeichnet, vgl. Hyndman et al., 2018b).

Eine weitere Klasse tiefer neuronaler Netze sind faltende neuronale Netze, welche häufig als Klassifikationsmodelle zur Bilderkennung verwendet werden. Dabei sind die ersten Schichten in der Regel faltende Schichten (engl. convolutional layers) oder bündelnde Schichten (engl. pooling layers), welche automatisch Merkmale auf verschiedenen Abstraktionsebenen aus einer Matrix von Pixeln extrahieren, die Bilder repräsentieren. Die extrahierten Merkmale werden dann vorwärts durch das Netzwerk bis zu einer vollständig verbundenen Ausgabeschicht projiziert. Durch dieses Vorgehen kann das Netz schrittweise z.B. spezifische Kanten eines Bildes erkennen und darauffolgend Motive als lokale Ansammlungen von Kanten, Bildteile als Ansammlungen von Motiven und letztlich Objekte, die aus mehreren Bildteilen bestehen (vgl. LeCun et al., 2015). Ma und Fildes (2021) trainierten ein faltendes neuronales Netz als Meta-Modell zur Selektion geeigneter Zeitreihenprognosemodelle, wobei das Netz automatisch Merkmale aus gegebenen Zeitreihen extrahierte. In der vorliegenden Arbeit werden die notwendigen Merkmale zur Beschreibung von Zeitreihen jedoch durch charakteristische Maßzahlen ermittelt. Daher ist die Verwendung faltender neuronaler Netze nicht notwendig. Nach dem *Prinzip von Ockhams Rasiermesser* (auch Sparsamkeitsprinzip, engl. Occam's razor) sollten simple Modelle komplexeren Modellen vorgezogen werden, wenn sie ausreichende Erklärungen liefern (vgl. Duda et al., 2001). Daher werden in dieser Arbeit flache MLP als Meta-Modelle verwendet (siehe Kapitel 5).

Abbildung 3.11 zeigt eine beispielhafte Konfiguration eines MLP als Zeitreihenprognosemodell. Im Folgenden werden die theoretischen Grundlagen für die Konfiguration eines MLP als Zeitreihenprognosemodell erläutert. Dabei wird zunächst die Architektur eines MLP beschrieben und darauffolgend der Trainingsprozess sowie die spezielle Konfiguration für Prognosemodelle.

Architektur eines Multilayer-Perzeptrons

Ein flaches MLP ist ein neuronales Netz mit drei Schichten von Neuronen (vgl. Rehkugler und Zimmermann, 1994; Bishop, 1995; Zell, 2003). Die erste Schicht wird *Eingabeschicht* (engl. input layer) genannt und sie dient, um Merkmalsvektoren \mathbf{x} in das neuronale Netz einzugeben. Die zweite Schicht wird *verdeckte Schicht* (engl. hidden layer) genannt und die dritte Schicht heißt *Ausgabeschicht* (engl. output layer). Sei $\text{netin}_{l,j} \in \mathbb{R}$ der *Eingabewert* (engl. input value)

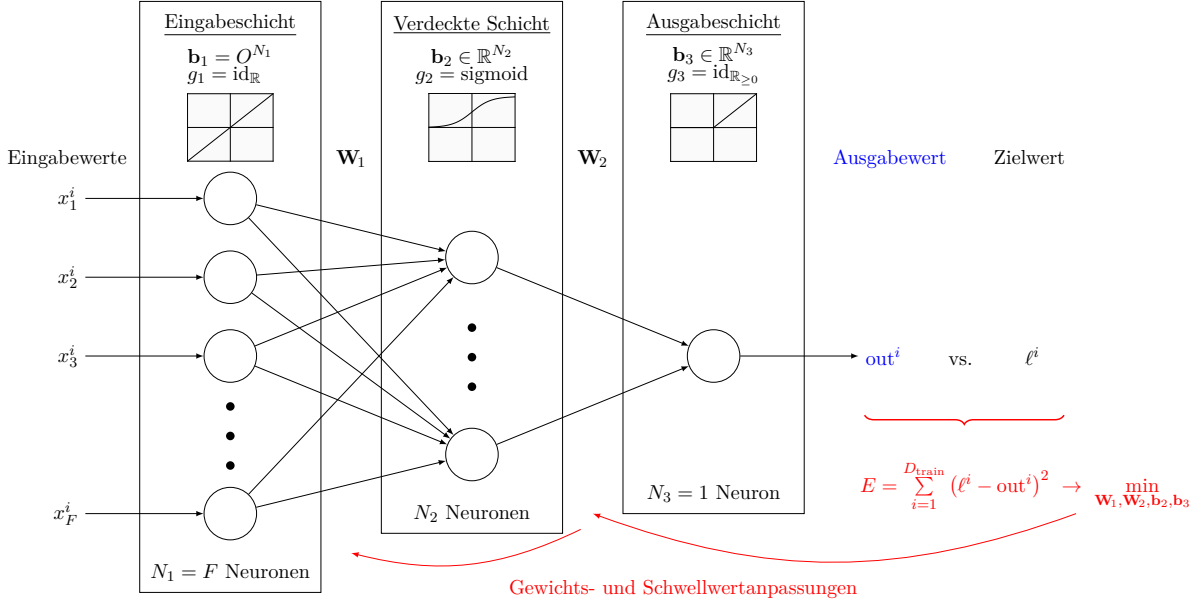


Abbildung 3.11: Konfiguration eines Multilayer-Perzeptrons als Zeitreihenprognosemodell

des j -ten Neurons in der l -ten Schicht eines MLP. Der Vektor der Eingabewerte $\mathbf{netin}_l = (\text{netin}_{l,1}, \dots, \text{netin}_{l,N_l})$ wird *Eingabevektor* (engl. input vector) der l -ten Schicht genannt. Sei ferner $\mathbf{b}_l \in \mathbb{R}^{N_l}$ der *Schwellwertvektor* (engl. bias vector) der l -ten Schicht, der aus den *Schwellwerten* (engl. bias) $b_{l,j}$ besteht. Die Elemente des *Ausgabevektors* (engl. output vector) \mathbf{out}_l der l -ten Schicht werden *Ausgabewerte* (engl. output values) genannt. Diese werden durch Anwendung einer sogenannten *Aktivierungsfunktion* (engl. activation function)

$$g_l \in \mathcal{G}, \quad g_l : \mathbb{R}^{N_l} \longrightarrow \mathbb{R}^{N_l} \quad (3.80)$$

auf die Differenz aus Eingabewerten und Schwellwerten bestimmt:

$$\text{out}_{l,j} = g_l(\text{netin}_{l,j} - b_{l,j}). \quad (3.81)$$

Um den Eingabewert des j -ten Neurons der $l+1$ -ten Schicht zu bestimmen, werden die Ausgabewerte der l -ten Schicht mit den zugehörigen Gewichten einer *Gewichtungsmatrix* (engl. weight matrix) $\mathbf{W}_l \in \mathbb{R}^{N_{l+1} \times N_l}$ multipliziert:

$$\text{netin}_{l+1,j} = \sum_{n=1}^{N_l} w_{l,j,n} \text{out}_{l,n}. \quad (3.82)$$

Im Allgemeinen wird der Schwellwertvektor der Eingabeschicht als ein Vektor mit Nullen definiert und die Aktivierungsfunktion dieser Schicht als die Identitätsfunktion, sodass die Eingabewerte der ersten Schicht, die die beschreibenden Merkmale $\mathbf{x} \in \mathcal{X}$ des Lernproblems sind, auch die Ausgabewerte der ersten Schicht sind. Durch iterative Anwendung von (3.81) und (3.82) wird der Ausgabevektor $\hat{\boldsymbol{\ell}} := \mathbf{out}_3 \in \mathbb{R}^C$ der dritten und letzten Schicht basierend auf den Eingabewerten der ersten Schicht $\mathbf{x} := \mathbf{netin}_1 \in \mathbb{R}^F$ wie folgt berechnet:

$$\hat{m}^{\text{NN}}(\mathbf{x}) = \hat{\boldsymbol{\ell}}^{\text{NN}} = \mathbf{out}_3 = g_3(\mathbf{W}_2 g_2(\mathbf{W}_1 \mathbf{x} - \mathbf{b}_2) - \mathbf{b}_3). \quad (3.83)$$

Training eines Multilayer-Perzeptrons

Um zu erreichen, dass ein MLP sinnvolle Ausgabewerte zu gegebenen Eingabewerten bestimmt, muss es basierend auf einer Trainingsmenge $\mathbf{D}_{\text{train}} = \{(\mathbf{x}^i, \ell^i)\}_{i=1}^{D_{\text{train}}}$ aus Merkmalsvektoren $\mathbf{x}^i \in \mathcal{X} \subseteq \mathbb{R}^F$ und Zielvektoren $\ell \in \mathcal{L} \subseteq \mathbb{R}^C$ trainiert werden. In diesem Zusammenhang bedeutet das Training eines Modells, geeignete Werte für die Gewichtungsmatrizen \mathbf{W}_1 und \mathbf{W}_2 sowie die Schwellwertvektoren \mathbf{b}_2 und \mathbf{b}_3 zu finden. Zu diesem Zweck muss eine geeignete Fehlerfunktion E definiert werden. Für die Anwendungen zum Training von Zeitreihenprognosemodellen, wird in dieser Arbeit die Summe der quadrierten Residuen (3.16) verwendet, während für das Training von Klassifikationsmodellen die Kreuzentropie (3.13) verwendet wird (siehe Kapitel 5). Die Gewichtungsmatrizen und Schwellwertvektoren werden in jeder Lernepoche des Trainingsprozesses so angepasst, dass der mittlere Fehler minimiert wird. Daher werden die Punkte der Fehlerfläche mit Gradient Null gesucht:

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = 0, \quad (3.84)$$

wobei \mathbf{w} den Vektor aller anpassbaren Netzwerkparameter, d.h. aller Gewichte und Schwellwerte, bezeichnet. Da die Fehlerfunktion E im Allgemeinen nicht konvex ist, ist eine Parameterkonfiguration, die (3.84) erfüllt, nicht notwendigerweise das globale Minimum der Fehlerfunktion, sondern es kann sich auch um ein lokales Minimum oder einen Sattelpunkt handeln (vgl. Bishop, 1995). In der Regel ist es nicht möglich, die Minima der Fehlerfunktion analytisch zu bestimmen. Daher wird ein iterativer *Gradientenabstieg* (engl. gradient descent) durchgeführt, um (lokale) Minima von E zu bestimmen. Zunächst werden die Gewichte und Schwellwerte eines Netzes durch die Methode von Nguyen und Widrow (1990) initialisiert. Darauf folgend wird ein Gradientenverfahren verwendet, um die Netzwerkparameter iterativ zu optimieren. In jeder Lernepoche ι des Trainingsprozesses werden die folgenden Schritte iterativ durchgeführt: (i) Anwendung des neuronalen Netzes (3.83), um die Ausgabevektoren $\hat{\ell}^i$ zu den gegebenen Eingabevektoren \mathbf{x}^i aus der Trainingsmenge zu berechnen, (ii) Bestimmung des mittleren Fehlers E , um die Ausgabevektoren $\hat{\ell}^i$ des neuronalen Netzes mit den wahren Zielwerten ℓ^i zu vergleichen, (iii) Bestimmung der Gewichts- und Schwellwertanpassungen $\Delta \mathbf{w}_{(\iota)}$ durch einen Gradientenabstieg und (iv) Anpassung der Gewichtungen und Schwellwerte als

$$\mathbf{w}_{(\iota+1)} = \mathbf{w}_{(\iota)} + \Delta \mathbf{w}_{(\iota)}. \quad (3.85)$$

Für die Bestimmung der Parameteranpassungen in jeder Lernepoche werden die partiellen Ableitungen der Fehlerfunktion benötigt. Zu diesem Zweck wird der *Algorithmus der Fehlerrückführung* (engl. error backpropagation algorithm) genutzt (vgl. Werbos, 1974; Rumelhart et al., 1986a,b), welcher eine Generalisierung des LMS-Algorithmus von Widrow und Hoff (1960) ist. Ähnlich zur Vorwärtspropagation der Eingabewerte durch das Netz durch sukzessive Anwendung von (3.81) und (3.82), um den Ausgabewert zu berechnen, wird beim Algorithmus der Fehlerrückführung der Ausgabevektor des Netzes \mathbf{out}_3 mit dem wahren Zielvektor ℓ verglichen und die Fehler werden

rückwärts durch das Netz propagiert, um zu bestimmen, wie die Gewichte und Schwellwerte angepasst werden müssen.

Der Standardansatz der Fehlerrückführung ermittelt die Parameteranpassungen mit Hilfe der partiellen Ableitungen der Fehlerfunktion als

$$\Delta \mathbf{w}_{(t)} = -\eta_{(t)} \frac{\partial E(\mathbf{w}_{(t)})}{\partial \mathbf{w}_{(t)}}, \quad (3.86)$$

wobei die Schrittweite η als *Lernrate* (engl. learning rate) bezeichnet wird. Um einen effizienten Trainingsprozess zu erreichen, muss die Lernrate η sinnvoll gewählt werden. Abbildung 3.12 zeigt verschiedene Formen eines Trainingsprozesses für verschiedene Lernraten (vgl. Zell, 2003). Wenn η zu klein gewählt ist, führt dies zu einer langsamen Konvergenz. Wenn η leicht zu groß ist, führt dies zu Oszillationen. Ist η signifikant zu groß, kann dies zu einer Divergenz führen.

Um stets einen sinnvollen Wert für die Lernrate zu gewährleisten, wird diese bei vielen Trainingsalgorithmen adaptiv in jeder Lernepoche angepasst. Es existieren diverse verschiedene Trainingsalgorithmen, die jeweils Vor- und Nachteile aufweisen (vgl. Rehkugler und Zimmermann, 1994; Zell, 2003; Neuneier und Zimmermann, 2012; Ruder, 2016). Die beiden verwendeten MLP-Prognoseverfahren dieser Arbeit nutzen unterschiedliche Trainingsalgorithmen. nnfor nutzt den Algorithmus der resilienten Fehlerrückführung mit Gewichtsrücksetzung (Rprop, resilient backpropagation with weight backtracking, vgl. Riedmiller, 1994). Hierbei werden die Parameteranpassungen über das Vorzeichen des Gradienten bestimmt und die Lernrate wird adaptiv in jeder Epoche angepasst. Falls das Vorzeichen des Gradienten für zwei aufeinanderfolgende Lernepochen gleich bleibt, wird die Lernrate erhöht, um den Trainingsprozess zu beschleunigen. Falls das Vorzeichen des Gradienten sich von einer Epoche zur nächsten ändert, wird die Lernrate verringert und die Parameteränderungen der letzten Epoche werden rückgängig gemacht. Durch dieses Vorgehen führt Rprop in der Regel zu einer deutlich schnelleren Konvergenz als der Standardansatz des Gradientenabstiegs (3.86). nnetar verfolgt einen anderen Ansatz zum Training der Netzparameter, das Broyden-Fletcher-Goldfarb-Shanno-Verfahren (BFGS, vgl. Gill et al., 1981; Ripley und Venables, 2015). Dies ist ein numerisches Optimierungsverfahren, welches der Gruppe der Quasi-Newton-Verfahren zugeordnet werden kann. Dabei wird die Hesse-Matrix iterativ approximiert und darauffolgend für die Berechnung der Parameteranpassungen des Netzes verwendet. Für weitere Informationen zum Training neuronaler Netze siehe z.B. Rehkugler und Zimmermann (1994) oder Bishop (1995).

Aufgrund des hochdimensionalen Parameterraums neuronaler Netze und der Nichtkonvexität der mittleren Fehlerfunktion hängt der Trainingsprozess und somit auch die letztlich gefundene Parameterkonfiguration stark vom Anfangszustand des neuronalen Netzes ab (vgl. Crone und Kourentzes, 2010). Aus diesem Grund werden in dieser Arbeit sowohl bei Verwendung von nnfor als auch bei Verwendung von nnetar jeweils $N = 20$ Netze pro Zeitreihe trainiert und die Prognosen dieser Netze werden jeweils in einem sogenannten *Ensemble* zu finalen Prognosen kombiniert (vgl. Breiman, 1996; Hastie et al., 2009; Crone und Kourentzes, 2010; Barrow und

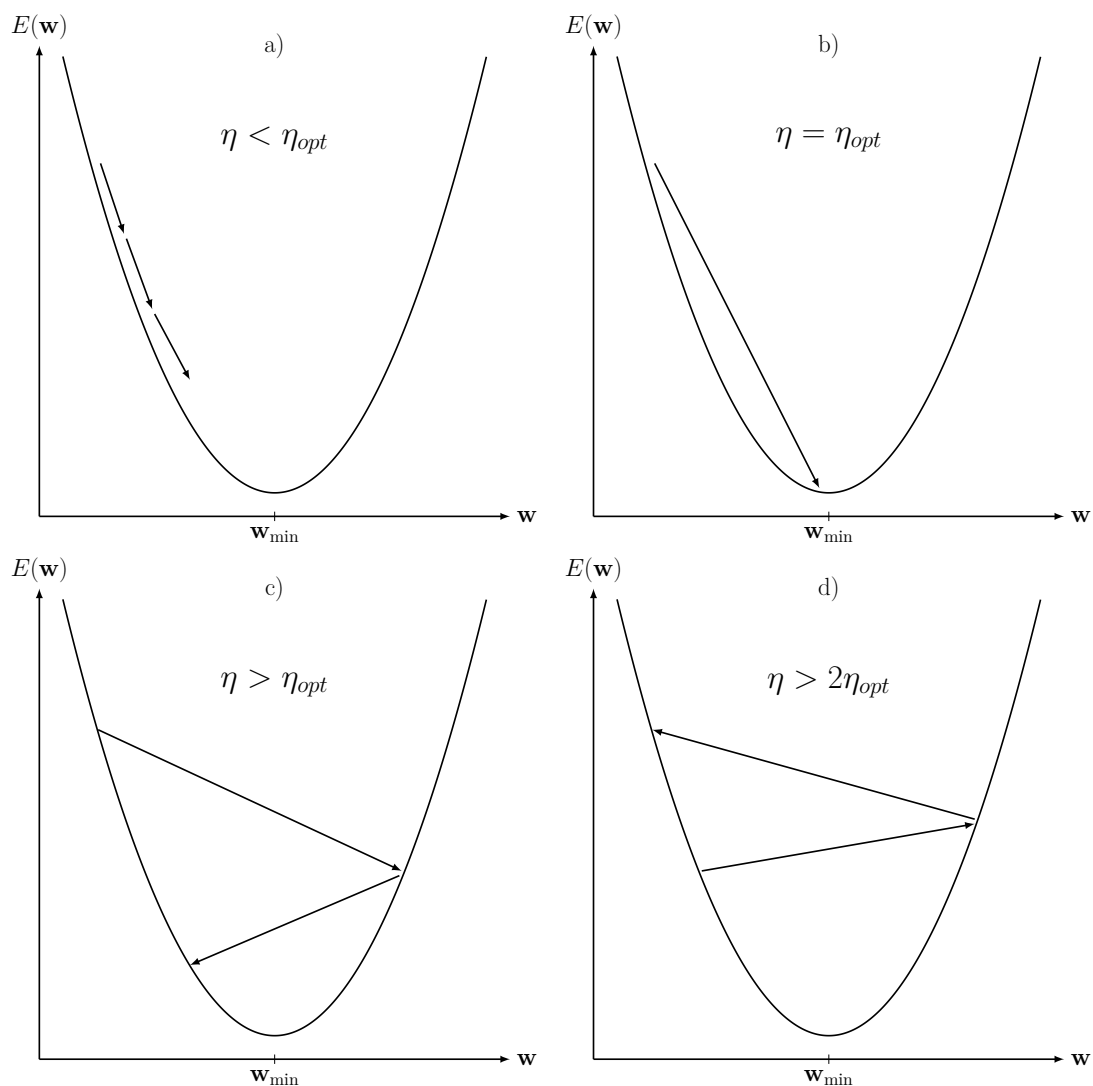


Abbildung 3.12: Verschiedene Formen des Trainingsprozesses in Abhängigkeit von der Lernrate η : (a) langsame Konvergenz, (b) optimale Konvergenz, (c) Oszillation und (d) Divergenz (angelehnt an LeCun et al. (2012))

Crone, 2016). Dabei verwendet `nnfor` den Median als Kombinationsoperator und `nnetar` bildet das arithmetische Mittel der Prognosen. Zudem wird für `nnetar` ein Gewichts­dämpfungs­faktor von 0,2 verwendet, weil dieser in Voruntersuchungen zu einer guten Prognosegüte führte. Für alle anderen Parameter beider Prognoseverfahren werden die Standardwerte genutzt.

Konfiguration eines Multilayer-Perzeptrons als Zeitreihenprognosemodell

Zur Konfiguration eines MLP als Zeitreihenprognosemodell werden in dieser Arbeit generelle Empfehlungen aus der Literatur verfolgt (vgl. Rehkugler und Zimmermann, 1994; Bishop, 1995; Zhang und Qi, 2005; Neuneier und Zimmermann, 2012; Crone und Kourentzes, 2010; The MathWorks Inc., 2017a; Hyndman et al., 2018b). Es werden die beiden automatischen

Prognoseverfahren `nnfor` und `nnetar` mit Standardparametern verwendet. In beiden Fällen werden Ensembles von $N = 20$ MLP pro Zeitreihe trainiert, wobei jedes einzelne MLP als ein autoregressives Modell konfiguriert wird. Dies bedeutet, dass der Vektor der Eingabewerte eine Teilmenge der bekannten Zeitreihenwerte ist

$$\mathbf{x}^t = (x_1^t, \dots, x_F^t) \subseteq (y_1, \dots, y_t). \quad (3.87)$$

Dabei identifiziert `nnfor` die saisonalen Frequenzen einer Zeitreihe automatisch durch einen neuronalen Filter und selektiert auf dieser Basis sinnvolle zeitversetzte Werte der gegebenen Zeitreihe als Eingabewerte für das Lernproblem (vgl. Crone und Kourentzes, 2010). `nnetar` selektiert ebenfalls automatisch die zu berücksichtigenden zeitversetzten Werte der Zeitreihe. Für nicht-saisonale Zeitreihen wird die optimale Zeitverzögerung eines linearen autoregressiven Modells durch das AIC bestimmt (vgl. (3.55)). Für saisonale Zeitreihen werden die Werte nach einer Anpassung des optimalen linearen Modells an die saisonal bereinigten Daten bestimmt (vgl. Hyndman et al., 2018b; Hyndman und Athanasopoulos, 2018).

Die Anzahl der Neuronen in der Eingabeschicht N_1 ist die Anzahl F zu berücksichtigender Werte der gegebenen Zeitreihe. In der Ausgangschicht existiert lediglich ein Neuron, welches den Einschrittprognosewert

$$\hat{m}_1^{\text{NN}}(\mathbf{x}^t) = \hat{y}_{t|t+1}^{\text{NN}} = \text{out}_3^t \quad (3.88)$$

beschreibt. Mehrschrittprognosen werden iterativ durchgeführt. Um beispielsweise den Wert y_{T+2} zu prognostizieren, wird zunächst die Prognose $\hat{y}_{T|T+1}$ basierend auf $\{y_1, \dots, y_T\}$ berechnet und dann die Prognose $\hat{y}_{T|T+2}$ basierend auf $\{y_1, \dots, y_T, \hat{y}_{T|T+1}\}$. Die Anzahl der Neuronen N_2 in der verdeckten Schicht hat einen großen Einfluss auf die Fähigkeit eines neuronalen Netzes, Muster zu lernen. Falls diese Anzahl zu groß gewählt wird, kann das Netz zu einer Überanpassung neigen, während eine zu geringe Anzahl zu einer Unteranpassung führen kann (vgl. Abschnitt 3.1.2). Bei Verwendung von `nnfor` wird die Anzahl der Neuronen in der verdeckten Schicht standardmäßig als $N_2 = 5$ gesetzt (vgl. Kourentzes, 2017). `nnetar` verwendet standardmäßig $N_2 = F/2 + 1$ gerundet auf die nächste ganze Zahl (vgl. Hyndman et al., 2018b).

Als Aktivierungsfunktion g_1 der Eingabeschicht wird normalerweise die Identitätsfunktion verwendet, damit die Eingabewerte in \mathbf{x} direkt gewichtet auf die verdeckte Schicht projiziert werden. Als Aktivierungsfunktion der verdeckten Schicht nutzen `nnfor` und `nnetar` eine *Sigmoidfunktion* (engl. sigmoid function)

$$g_2(x) = \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (3.89)$$

durch die das Netz nichtlineare Zusammenhänge abbilden kann. Für die Aktivierungsfunktion der Ausgangschicht wird eine positive Identitätsfunktion verwendet,

$$g_3(x) = \text{id}_{\mathbb{R}_{\geq 0}} = \begin{cases} x, & \text{für } x \geq 0 \\ 0, & \text{sonst,} \end{cases} \quad (3.90)$$

sodass der Prognosewert eine Linearkombination der Ausgabewerte der verdeckten Schicht ist, jedoch nicht kleiner als 0, da eine Kundennachfrage nicht negativ sein kann.

3.5 Zusammenfassung

In diesem Kapitel wurden die Grundlagen und der Stand der Forschung zur Zeitreihenprognose beschrieben. Zunächst wurde im ersten Abschnitt überwachtes Lernen als Teilgebiet des maschinellen Lernens definiert, welches sich wiederum in der Schnittmenge der Gebiete der Data Science und der künstlichen Intelligenz befindet. In den theoretischen Grundlagen des überwachten Lernens wurden überwachte Lernprobleme definiert, erklärende und prädiktive Modelle unterschieden sowie Regressions- und Klassifikationsprobleme erläutert. Im zweiten Abschnitt des Kapitels wurde das Gebiet der Zeitreihenprognose ausführlich beschrieben. Neben einer Definition von Zeitreihenprognoseproblemen wurden Evaluationskriterien für Zeitreihenprognosen anhand einzelner oder mehrerer Zeitreihen dargestellt. Im dritten Abschnitt wurden zunächst die Datensätze und die Hauptergebnisse der beiden letzten großen Zeitreihenprognosewettbewerbe beschrieben, der M3- und der NN3-Competition. Aus den Ergebnissen ließ sich FT 1 bestätigen, denn es existiert keine stets beste Modellklasse zur Prognose der Kundennachfrage produzierender Unternehmen. Des Weiteren wurden aus den Ergebnissen der Wettbewerbe Best Practices abgeleitet, die sowohl bei der Entwicklung des neuartigen Verfahrens dieser Arbeit als auch bei den Evaluationsstudien berücksichtigt werden. Darauffolgend wurden als gängigste Modellklassen im Bereich der Zeitreihenprognose die Klassen der exponentiellen Glättungsmodelle, der ARIMA-Modelle und der neuronalen Netze identifiziert. Diese Klassen wurden im vierten Abschnitt dieses Kapitels erläutert. Zusätzlich wurden zwei weitere Modellklassen beschrieben. Random-Walk-Modelle wurden dargestellt, da diese häufig als Benchmarks verwendet werden. Zudem wurden lokale k nn-Modelle ausführlich beschrieben, da diese in anderen Anwendungsbereichen, wie den Naturwissenschaften, vielversprechende Ergebnisse erzielen konnten, jedoch bislang kaum zur Prognose der Kundennachfrage eingesetzt wurden. Da aktuell kein etabliertes Verfahren zur Optimierung der Parameter von lokalen k nn-Modellen existiert, wurde im Rahmen dieser Arbeit eine neuartige dynamische Modellselektionsmethode entwickelt, die in Vorstudien evaluiert wurde (vgl. Anhang C).

Kapitel 4

Stand der Forschung und Taxonomie zur Selektion geeigneter Zeitreihenprognosemodelle

„Prediction is very difficult, especially if it's about the future.“

Niels H. D. Bohr, dänischer Nobelpreisträger für Physik (1885 - 1962)

Dieses Kapitel stellt den Stand der Forschung zur Selektion geeigneter Zeitreihenprognosemodelle dar, um Best Practices und Limitierungen bestehender Ansätze zu identifizieren. Nach einer formalen Definition des Prognosemodellselektionsproblems werden die theoretischen Grundlagen der Selektion und Kombination von Prognosemodellen erläutert. Darauf folgend wird der Stand der Forschung nach verschiedenen Selektionskriterien gegliedert dargestellt. Hieraus wird eine neuartige Taxonomie abgeleitet, um alle Ansätze zu klassifizieren. Das Kapitel schließt mit einer Identifikation der Best Practices und der Limitierungen der bestehenden Ansätze zur Selektion von Prognosemodellen.

4.1 Prognosemodellselektionsproblem

Nach dem sogenannten *No Free Lunch Theorem* existiert kein universell bestes Lernverfahren, welches für alle Probleminstanzen eines umfassenden Anwendungsgebietes bessere Ergebnisse erzielt als alle anderen möglichen Lernverfahren (vgl. Wolpert, 1996; Wolpert und Macready, 1997). Hieraus lässt sich folgern, dass kein allgemein bestes Prognoseverfahren existiert, wenn man das Anwendungsgebiet aller Zeitreihenprognoseprobleme betrachtet (vgl. Smith-Miles, 2008). Zudem geben die Ergebnisse aktueller Studien sowie der letzten großen Prognosewettbewerbe empirische Evidenz, dass diese allgemeine Aussage auch für das Teilgebiet der Nachfrageprognoseprobleme gilt (vgl. FT 1 und Abschnitt 3.3). Daher werden Ansätze zur Selektion geeigneter Zeitreihenprognosemodelle benötigt, um die bestmöglichen Prognosen für jede spezifische Zeitreihenentwicklung

zu bestimmen. In Analogie zur Formulierung des sogenannten *Algorithmusselektionsproblems* (ASP, engl. algorithm selection problem) von Rice (1976) und der Reformulierung von Smith-Miles (2008) wird im Folgenden das Hauptproblem dieser Arbeit formuliert:

Prognosemodellselektionsproblem (PMSP). Sei $\mathbf{y} \in \mathcal{Y}$ eine gegebene Zeitreihe und sei $\mathbf{A} = \{a^1, \dots, a^M\}$ eine gegebene Menge an M Prognoseverfahrenskandidaten. Dann ist das Ziel eines Selektionsverfahrens die Auswahl einer geeigneten Teilmenge an M_{sel} Prognoseverfahren, mit $M_{\text{sel}} \in \{1, \dots, M\}$, sodass eine Kombination $\hat{m}^{\text{Comb}}(\mathbf{y})$ der resultierenden Prognosemodelle $\{\hat{m}^{(1)}, \dots, \hat{m}^{(M_{\text{sel}})}\}$ den mittleren Prognosefehler minimiert.

Ein PMSP besteht somit aus zwei Hauptschritten. Zunächst wird ein Selektionsverfahren angewandt um M_{sel} Prognoseverfahren zu selektieren und darauffolgend wird jedes dieser Prognoseverfahren verwendet, um jeweils ein Prognosemodell in je einem überwachten Lernproblem (3.9) zu trainieren. Die Leistung eines Prognoseverfahrens und somit auch die Leistung eines Selektionsverfahrens lässt sich erst anhand der Prognosen der trainierten Modelle evaluieren.

Ein Selektionsverfahren zur Lösung eines PMSP beinhaltet in der Regel die folgenden Komponenten: (i) ein Verfahren zur Bestimmung einer erwarteten Rangfolge der Prognoseverfahrenskandidaten, (ii) ein Verfahren zur Bestimmung der Anzahl M_{sel} zu selektierender Prognoseverfahren und (iii) eine Selektionsfunktion, die basierend auf einer erwarteten Rangfolge M_{sel} Verfahren auswählt, die zum Training von Prognosemodellen verwendet werden. Es sei angemerkt, dass für den Fall $M_{\text{sel}} = 1$ formal keine Kombination durchgeführt wird, sondern lediglich ein Prognoseverfahren selektiert und danach verwendet wird. Dies ist jedoch keine Einschränkung der Formulierung des PMSP, da eine Selektion eines Verfahrens ein Spezialfall einer Kombination ist.

4.2 Theoretische Grundlagen zur Selektion von Zeitreihenprognosemodellen

4.2.1 Übersicht

Während klassischerweise in der Regel Experten manuell geeignete Prognosemodelle für jede Zeitreihe selektierten, erfordern heutige großskalige Anwendungen, in denen Prognosen für eine Vielzahl an Zeitreihen in regelmäßigen Abständen benötigt werden, automatische Selektionsverfahren. Aus diesem Grund werden in der vorliegenden Arbeit keine manuellen, sondern ausschließlich automatische Selektionsverfahren verwendet. Abbildung 4.1 gibt eine Übersicht über die verschiedenen Arten von Selektionsverfahren für Prognosemodelle.

Es lassen sich Ansätze zur Selektion eines einzelnen Prognosemodells und Ansätze zur Kombination der Prognosen mehrerer Modelle unterscheiden. Ohne Einschränkung sei angenommen, dass M Prognoseverfahrenskandidaten zur Auswahl stehen, die entweder aus der gleichen oder aus mehreren verschiedenen Modellklassen stammen. Bei einer Selektion wird aus den M möglichen

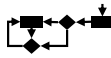













Ansatz		Anzahl der Verfahrenskandidaten	Anzahl zu selektierender Modelle
Selektion	Selektion mit vorgegebenem Verfahren	1 	1 
	Selektion aus einer vorgegebenen Modellklasse	M 	1 
	Selektion aus mehreren möglichen Modellklassen	M 	1 
Kombination	Pooling aus einer vorgegebenen Modellklasse	M 	M_{sel} 
	Pooling aus mehreren möglichen Modellklassen	M 	M_{sel} 
	Kombination aus einer vorgegebenen Modellklasse	M 	M 
	Kombination aus mehreren Modellklassen	M 	M 

Abbildung 4.1: Ansätze zur Selektion oder Kombination von Prognosemodellen aus einer oder mehreren möglichen Modellklassen für eine Zeitreihe

Verfahrenskandidaten ein Verfahren ausgewählt und dieses wird zum Training eines Prognosemodells im Rahmen eines überwachten Lernproblems (3.9) verwendet. Hierbei gilt also $M_{\text{sel}} = 1$. Die erste Zeile in Abbildung 4.1 zeigt den Spezialfall der Selektion eines Modells mit einem vorgegebenen Prognoseverfahren. In diesem Fall entfällt der erste Schritt des PMSP zur Auswahl eines Prognoseverfahrens und es wird lediglich ein überwacht Lernproblem betrachtet, um ein Prognosemodell zu trainieren.

Bei der Kombination von Prognosemodellen werden alle M Verfahrenskandidaten verwendet, um je ein Modell zu trainieren und die Prognosen der M Modelle werden kombiniert. Neben der Kombination von Modellen aus verschiedenen Klassen kann auch die Kombination mehrerer Modelle der gleichen Klasse sinnvoll sein. Beispielsweise werden häufig mehrere neuronale Netze mit verschiedenen Initialgewichten trainiert und die resultierenden Netze werden zu einem sogenannten Ensemble kombiniert.

Ein weiterer Ansatz, der häufig als Spezialfall der Kombination betrachtet wird, ist der Ansatz des Poolings. Hierbei werden aus den M zur Verfügung stehenden Verfahrenskandidaten M_{sel} Verfahren ausgewählt und diese werden verwendet, um jeweils ein Modell zu trainieren. In diesem Fall wird also ein PMSP betrachtet, bei dem M_{sel} Verfahren selektiert werden, wobei $2 \leq M_{\text{sel}} \leq M$ gilt. Darauffolgend sind M_{sel} überwachte Lernprobleme zu betrachten, um Modelle zu trainieren, deren Prognosen letztlich kombiniert werden. Im Folgenden werden zunächst Ansätze zur Selektion von Prognosemodellen beschrieben und darauffolgend Ansätze zur Kombination der Prognosen mehrerer Modelle erläutert.

4.2.2 Ansätze zur Selektion von Prognosemodellen

Bei der Selektion eines Prognosemodells lassen sich zwei verschiedene Ansätze unterscheiden: (i) die Selektion eines Modells aus einer vorgegebenen Klasse und (ii) die Selektion eines Modells aus mehreren möglichen Klassen (vgl. Fildes, 1989; Meade, 2000b; Fildes und Petropoulos, 2015; Kück et al., 2016b; Ord et al., 2017). Mit einer *Klasse* (engl. class) \mathcal{C} ist hier eine Menge von Prognosemodellen einer bestimmten Struktur mit dazugehörigen Verfahren zum Training der Parameter der Modelle gemeint. Da für jede Klasse der Raum der Modelle und der Raum der Verfahren miteinander verbunden sind, wird der Begriff der Klasse im Folgenden synonym sowohl für Mengen von Prognoseverfahren oder -modellen verwendet. Mögliche Klassen sind beispielsweise die Klasse der exponentiellen Glättungsmodelle und die Klasse der neuronalen Netze, welche in den Abschnitten 3.4.3 und 3.4.5 beschrieben wurden.

Bei der *Selektion eines Prognosemodells aus einer vorgegebenen Modellklasse* wird zunächst ein Prognoseverfahren a aus $\mathcal{A}_{\text{ZRP}}^{\mathcal{C}}$, dem *Raum der Prognoseverfahren der Klasse \mathcal{C}* , ausgewählt. Dieses Verfahren wird verwendet, um ein Modell aus $\mathcal{M}_{\text{ZRP}}^{\mathcal{C}}$, dem zugehörigen *Raum der Prognosemodelle der Klasse \mathcal{C}* , zu trainieren, welches darauffolgend zur Prognose zukünftiger Zeitreihenwerte eingesetzt werden kann. Wenn beispielsweise a ein exponentielles Glättungsverfahren ist, wird das Prognosemodell nach dem Training aus dem Raum der exponentiellen Glättungsmodelle $\mathcal{M}_{\text{ZRP}}^{\text{ets}}$ stammen. Die Selektion aus einer vorgegebenen Klasse ist also ein überwachtes Lernproblem (3.9), bei dem die Klasse \mathcal{C} vorgegeben ist und somit wird der bestmögliche *Parametervektor* $\theta^* \in \Theta_{\text{ZRP}}^{\mathcal{C}}$ in der vorgegebenen Klasse gesucht. Dies ist derjenige Parametervektor, der die besten Prognosen für eine ungesehene Testmenge \mathbf{D}_{test} liefert:

$$\theta^* = \operatorname{argmin}_{\theta \in \Theta_{\text{ZRP}}^{\mathcal{C}}} \sum_{(\mathbf{x}^i, \ell^i) \in \mathbf{D}_{\text{test}}} e\left(\ell^i, \hat{m}(\mathbf{x}^i, \theta)\right), \quad \text{mit } \hat{m} = a(\mathbf{D}_{\text{train}}). \quad (4.1)$$

Hierbei bezeichnet $\Theta_{\text{ZRP}}^{\mathcal{C}}$ den *Raum der Parametervektoren der Klasse \mathcal{C}* . Da ein Prognosemodell selektiert werden muss, bevor die zukünftigen Daten in der Testmenge \mathbf{D}_{test} bekannt werden, wird ein Selektionskriterium benötigt, über das sich die zukünftige Prognosegüte eines Modells für eine Zeitreihe möglichst genau abschätzen lässt (vgl. Abschnitt 4.3.1).

Durch die Einschränkung des Modellraums auf eine einzelne Modellklasse \mathcal{C} lassen sich bewährte Prognoseverfahren aus dieser Klasse verwenden, um die finalen Modelle zu trainieren. Allerdings birgt eine solche Einschränkung das Risiko, dass das bestmögliche Modell, das durch ein Prognoseverfahren in der vorgegebenen Klasse gefunden werden kann, eine schlechtere Leistung erzielt, als ein Modell einer anderen Klasse, aus der nicht gewählt werden kann. Um dieses Problem zu umgehen, lässt sich eine *Selektion eines Prognosemodells aus mehreren möglichen Modellklassen* durchführen. Sei

$$\mathcal{A}_{\text{ZRP}}^{\mathcal{C}_1, \dots, \mathcal{C}_M} = \mathcal{A}_{\text{ZRP}}^{\mathcal{C}_1} \cup \mathcal{A}_{\text{ZRP}}^{\mathcal{C}_2} \cup \dots \cup \mathcal{A}_{\text{ZRP}}^{\mathcal{C}_M} \quad (4.2)$$

der Raum aller Prognoseverfahren der Klassen $\mathcal{C}_1, \dots, \mathcal{C}_M$, wobei nicht notwendigerweise alle Klassen \mathcal{C}_j unterschiedlich sein müssen. Dann werden in der Regel nicht alle Verfahren des kom-

pletten Raumes $\mathcal{A}_{\text{ZRP}}^{\mathcal{C}_1, \dots, \mathcal{C}_M}$ zur Auswahl gestellt. Stattdessen werden bestimmte Prognoseverfahren aus diesen Klassen ausgewählt und es wird eine Teilmenge

$$\mathbf{A} = \{a^1, \dots, a^M\} \subseteq \mathcal{A}_{\text{ZRP}}^{\mathcal{C}_1, \dots, \mathcal{C}_M} \subseteq \mathcal{A}_{\text{ZRP}} \quad (4.3)$$

betrachtet, die als *Menge der Prognoseverfahrenskandidaten* (engl. set of forecasting method candidates) bezeichnet wird. Es seien eine Menge an Prognoseverfahrenskandidaten \mathbf{A} aus den Klassen $\mathcal{C}_1, \dots, \mathcal{C}_M$ sowie eine Zeitreihe $\mathbf{y} = \{y_1, \dots, y_T\}$ gegeben. Es seien ferner

$$\hat{\mathbf{R}} = (\hat{R}_1, \hat{R}_2, \dots, \hat{R}_M) \in \mathcal{R} \subset \mathbb{R}^M \quad (4.4)$$

eine *erwartete Rangfolge* der Verfahrenskandidaten für die Zeitreihe hinsichtlich eines geeigneten Selektionskriteriums und $a^{(j)}$ das *Prognoseverfahren mit dem j -besten erwarteten Rang*. Dann bezeichnet

$$s_1 \in \mathcal{S}, \quad s_1 : \mathcal{R} \longrightarrow \mathbf{A}, \quad s_1(\hat{\mathbf{R}}) = a^{(1)} \quad (4.5)$$

eine *Selektionsfunktion* (engl. selection function), die basierend auf der erwarteten Rangfolge ein Prognoseverfahren $a^{(1)} \in \mathbf{A}$ aus der Menge der Verfahrenskandidaten selektiert. Ziel der Selektionsfunktion ist es, ein Prognoseverfahren a^* zu selektieren, welches bei Verwendung zum bestmöglichen Prognosemodell führt:

$$a^* = \operatorname{argmin}_{a \in \mathbf{A}} \sum_{(\mathbf{x}^i, \ell^i) \in \mathbf{D}_{\text{test}}} e(\ell^i, \hat{m}(\mathbf{x}^i)), \quad \text{mit } \hat{m} = a(\mathbf{D}_{\text{train}}). \quad (4.6)$$

Die Selektion eines Prognosemodells aus mehreren möglichen Modellklassen bietet die Möglichkeit, die Eignung verschiedener Modellklassen für eine Zeitreihe in einer erwarteten Rangfolge abzuschätzen und dann diejenige Klasse auszuwählen, die am sinnvollsten erscheint. In Abschnitt 4.3.1 werden verschiedene Kriterien erläutert, um eine erwartete Rangfolge abzuschätzen.

4.2.3 Ansätze zur Kombination von Prognosemodellen

Die in Abschnitt 4.2.2 beschriebene Selektion eines Prognosemodells aus einer oder mehreren möglichen Modellklassen bietet das Potenzial, das bestmögliche Modell aus der sinnvollsten Klasse zu selektieren, wenn das verwendete Selektionsverfahren seinen Zweck erfüllt. Allerdings ist es möglich, dass ein Selektionsverfahren die Eignungen verschiedener Prognoseverfahren falsch beschreibt und somit ein Verfahren selektiert, welches ein für die Zeitreihe schlecht geeignetes Prognosemodell trainiert. Auf der anderen Seite ist es möglich, dass mehrere Prognoseverfahren gleichermaßen für eine gegebene Zeitreihe geeignet sind. In solchen Fällen bieten *Kombinationsmodelle* (engl. combination models) das Potenzial, die Unsicherheit, die bei der Selektion eines Modells entsteht, zu verkleinern, indem mehrere Modelle verwendet werden und deren Prognosen zu einer finalen Prognose kombiniert werden (vgl. Bates und Granger, 1969; Newbold und Granger, 1974; Clemen, 1989; De Menezes et al., 2000; Elliott und Timmermann, 2016). Dabei werden die Stärken und Schwächen mehrerer Modelle akkumuliert, um eine robuste finale

Prognose zu erzielen. Es sei angenommen, dass $M \geq 2$ gegebene Prognoseverfahren verwendet wurden, um h -Schritt-Prognosemodelle $\hat{m}_h^1, \dots, \hat{m}_h^M$ zu trainieren. Die Prognoseverfahren können dabei aus einer oder aus verschiedenen Klassen stammen. Dann bezeichnet

$$\begin{aligned} \hat{m}_h^{\text{Comb}} : ((\mathcal{M}_{\text{ZRP}})^M, \mathcal{X}) &\longrightarrow \mathcal{L} \\ \hat{m}_h^{\text{Comb}}(y_1, \dots, y_t) = \hat{y}_{t|t+h}^{\text{Comb}} &= \frac{1}{M} \sum_{j=1}^M w_j \hat{m}_h^j(y_1, \dots, y_t) \end{aligned} \quad (4.7)$$

ein *h-Schritt-Kombinationsmodell* (engl. *h-step ahead combination model*), wobei w_j das Gewicht des j -ten Modells ist und $\sum_{j=1}^M w_j = 1$ gilt. Zur Bestimmung passender Gewichte für eine Kombination wurden verschiedene Methoden entwickelt (vgl. Newbold und Granger, 1974; Elliott und Timmermann, 2016). Allerdings erzielen *gleichgewichtete Kombinationen* (engl. *equal weights combinations*) in der Regel eine ähnliche Prognosegüte, wie komplex gewichtete Kombinationen (vgl. Genre et al., 2013; Elliott und Timmermann, 2016). Daher werden in dieser Arbeit ausschließlich gleichgewichtete Kombinationen betrachtet.

Wenn aus der Menge von M gegebenen Prognoseverfahren lediglich eine kleinere Anzahl M_{sel} mit $2 \leq M_{\text{sel}} < M$ Verfahren verwendet wurden, um Modelle zu trainieren und diese darauffolgend zu kombinieren, wird ein Kombinationsmodell auch ein *Pooling-Modell* (engl. *pooling model*) genannt (vgl. Aiolfi und Timmermann, 2006; Geweke und Amisano, 2011; Kourentzes et al., 2019). Dieser Ansatz bietet die Möglichkeit, Verfahren oder Modelle von der Kombination auszuschließen, von denen keine guten Prognosen für eine gegebene Zeitreihe erwartet werden. Unter den gleichen Voraussetzungen, wie bei der Selektion eines Modells aus mehreren möglichen Klassen wird eine *M_{sel} -fache Selektionsfunktion*

$$s_{M_{\text{sel}}} \in \mathcal{S}, \quad s_{M_{\text{sel}}} : \mathcal{R} \longrightarrow (\mathbf{A})^{M_{\text{sel}}}, \quad s_{M_{\text{sel}}}(\hat{\mathbf{R}}) = \{a^{(1)}, \dots, a^{(M_{\text{sel}})}\} \quad (4.8)$$

gesucht, die basierend auf einer erwarteten Rangfolge M_{sel} Verfahren aus den vorgegebenen Klassen selektiert. Diese werden verwendet, um M_{sel} Modelle zu trainieren, deren Prognosen kombiniert werden, wie in (4.7) beschrieben. Für bisher entwickelte Pooling-Ansätze gilt stets $2 \leq M_{\text{sel}} \leq M$ (vgl. Kapitel 4). In der vorliegenden Arbeit wird jedoch ein hybrider Ansatz zwischen Selektion und Kombination vorgestellt, für den $1 \leq M_{\text{sel}} \leq M$ gilt (vgl. Kapitel 5). Hierdurch lassen sich die Vorteile der Selektion und der Kombination nutzen.

4.3 Stand der Forschung zur Selektion geeigneter Zeitreihenprognosemodelle

4.3.1 Übersicht

Wie in den vorherigen Abschnitten beschrieben wurde, besteht der Prozess der Modellselektion aus den beiden Schritten der Selektion eines oder mehrerer Prognoseverfahren aus einer Menge an

Prognoseverfahrenskandidaten $\mathbf{A} = \{a^1, a^2, \dots, a^M\}$ und der Verwendung dieser Prognoseverfahren zum Training von Prognosemodellen. Tabelle 4.1 vergleicht verschiedene *Selektionskriterien* (auch Rangfolgekriterien, engl. selection criteria), die sich in Vergleiche von Informationskriterien, Vergleiche von Prognosefehlern, regelbasierte Expertensysteme und Meta-Lernansätze unterteilen lassen. Während regelbasierte Expertensysteme ein oder mehrere Verfahren basierend auf definierten Entscheidungsregeln selektieren, werden anhand von Informationskriterien, Prognosefehlern sowie Meta-Lernansätzen erwartete Rangfolgen $\hat{\mathbf{R}} = (\hat{R}_1, \hat{R}_2, \dots, \hat{R}_M)$ gebildet. Diese sollen abschätzen, wie gut die verschiedenen Prognoseverfahren vermutlich für die ungesehene Testmenge der Zeitreihe funktionieren werden.

Tabelle 4.1: Vergleich verschiedener Ansätze zur Schätzung einer erwarteten Rangfolge von Prognoseverfahren für eine Zeitreihe

Ansatz	Selektion aus		Beispiele
	einer Klasse	mehreren Klassen	
Vergleich von Informationskriterien	✓*	x	ets, auto.arima
Vergleich von Prognosefehlern	✓	✓	Val-Sel, Pool-2, Pool-3
Regelbasierte Expertensysteme	✓	✓	RBF
Meta-Lernansätze	✓	✓	Selbination, ML-Sel

*wenn eine Likelihood gebildet werden kann

Im Folgenden wird der Stand der Forschung hinsichtlich verschiedener Ansätze zur Selektion von Prognosemodellen dargestellt. Dabei werden vornehmlich Ansätze zur Selektion von Verfahren oder Modellen betrachtet, während für klassische Ansätze zur Bestimmung der Gewichte in einer Prognosekombination auf die Literatur verwiesen wird (vgl. Bates und Granger, 1969; Newbold und Granger, 1974; De Menezes et al., 2000; Kolassa, 2011; Elliott und Timmermann, 2016; Barrow und Kourentzes, 2016). Die Selektionsansätze lassen sich in Ansätze zur aggregierten Selektion des gleichen Modells für alle Zeitreihen in einem Datensatz und Ansätze zur individuellen Selektion spezifischer Modelle für jede einzelne Zeitreihe unterteilen (vgl. Fildes, 1989; Fildes und Petropoulos, 2015). Aufgrund des offensichtlich größeren Potenzials liegt der Fokus der Literaturrecherche auf individuellen Selektionsansätzen, während nur wenige Ansätze zur aggregierten Selektion mit einbezogen werden. Da sich Informationskriterien nur zur Modellselektion aus spezifischen Klassen verwenden lassen, für die eine Likelihood gebildet werden kann, und zudem nicht zur Selektion aus mehreren verschiedenen Klassen, werden diese nicht betrachtet. Hierzu sei auf Abschnitt 3.4.2 und (Akaike, 1973; Schwarz, 1978; Hurvich und Tsai, 1989; Billah et al., 2005, 2006; Inoue und Kilian, 2006; Herwartz, 2010, 2011a,b; Hsu et al., 2019; Villegas und Pedregal, 2019) verwiesen. Spezifische Methoden zur Optimierung der Parameter für neuronale Netze oder Support Vector Machines können ebenfalls in der Literatur gefunden werden (vgl. Qi und Zhang, 2001; Crone und Kourentzes, 2010; Rubio et al., 2011; Florido et al., 2012; Ren et al., 2014; Aras und Kocakoç, 2016; Bredahl Kock und Teräsvirta, 2016). Der Stand

der Forschung hinsichtlich der Selektion von Prognosemodellen wird anhand folgender Kategorien unterschieden:

- **Ansatz.** Es existieren Ansätze zur Selektion eines Prognosemodells, Ansätze zur Kombination mehrerer Prognosemodelle sowie einige wenige hybride Ansätze. Einige Ansätze lassen sich automatisch verwenden, während andere nicht ohne Experteninput funktionieren.
- **Selektionskriterien.** Es werden Ansätze zum Vergleich von Prognosefehlern, regelbasierte Expertensysteme sowie Meta-Lernansätze betrachtet.
- **Merkmale.** Ein Selektionsverfahren greift auf Merkmale zurück, die ein Selektionsproblem beschreiben. Diese Merkmale lassen sich in direkte Merkmale, modellbasierte Merkmale und Landmarkers unterteilen (vgl. Brazdil et al., 2008). Direkte Merkmale, die auch als statistische oder informationstheoretische Merkmale bezeichnet werden, lassen sich direkt anhand der Zeitreihe bestimmen. Dies können beispielsweise generelle Informationen über die Zeitreihe, wie die Anzahl der Werte, oder deskriptive Statistiken, wie der Mittelwert und die Schiefe der Zeitreihe sein. Modellbasierte Merkmale charakterisieren die Daten indirekt, indem ein Modell anhand der Zeitreihe gebildet wird und Charakteristika oder Parameter des Modells als Merkmale zur Beschreibung des Selektionsproblems genutzt werden. Dies können z.B. p-Werte von statistischen Tests sein. Um die dritte Art der Merkmale zu bestimmen, die als Landmarkers bezeichnet wird (vgl. Pfahringer et al., 2000), werden die Trainings- oder Validierungsfehler von Prognosemodellen sowie Vergleiche zwischen diesen als Merkmale genutzt.
- **Prognosemodellklassen.** Ein Selektionsverfahren muss zwischen verschiedenen Prognoseverfahren oder -modellen wählen, die aus einer oder verschiedenen Klassen stammen können. Wie im letzten Kapitel beschrieben wurde, sind populäre Klassen z.B. die exponentielle Glättung, ARIMA-Modelle oder neuronale Netze.
- **Evaluationskriterien.** Die Leistung eines Selektionsverfahrens lässt sich anhand verschiedener Evaluationskriterien, wie Prognosefehlern, Bestandsmetriken oder dem benötigten Rechenaufwand, bewerten.

4.3.2 Vergleiche von Prognosefehlern

Ein gängiges Kriterium zur Bildung einer erwarteten Rangfolge ist der *Vergleich von Prognosefehlern* auf einer Trainings- oder einer Validierungsmenge. Tabelle 4.2 vergleicht verschiedene dieser Ansätze, von denen die meisten automatisch verwendet werden können.

Während frühe Ansätze meist anhand der Prognosefehler auf einer Trainingsmenge selektierten (vgl. Fildes, 1989; Franses und Koehler, 1998; Gardner, 1999), vergleichen aktuellere Ansätze in der Regel die Prognosefehler auf Validierungsmengen (vgl. Poler und Mula, 2011; Adhikari und

Tabelle 4.2: Ansätze zur Modellselektion durch Vergleiche von Prognosefehlern

Referenz	Ansatz				Selektionskriterien	Merkmale			Prognosemodellklassen	Evaluation		
	Automatisch	Selektion	Kombination	Hybrid		Direkt	Modellbasiert	Landmarking		Prognosefehler	Bestandssimulation	Rechenaufwand
Fildes (1989)	✓	✓	x	x	Trainingsfehler	x	x	✓	Exponentielle Glättung Robuster Trend	✓	x	x
Franses und Koehler (1998)	✓	✓	x	x	Trainingsfehler	x	x	✓	ARIMA	✓	x	x
Gardner (1999)	x	✓	x	x	Trainingsfehler	x	✓	✓	Exponentielle Glättung	✓	x	x
Poler und Mula (2011)	✓	✓	x	x	Gewichtete Validierungsfehler	x	x	✓	Dekomposition Exponentielle Glättung Gleitende Mittelwerte ARIMA	✓	x	x
Adhikari und Agrawal (2014)	✓	x	✓	x	Validierungsfehler	x	x	✓	ARIMA Support Vector Regression Neuronale Netze	✓	x	x
Fonseca und Gómez-Gil (2014)	✓	x	✓	x	Validierungsfehler	x	x	✓	Neuronale Netze ARIMA	✓	x	x
Fildes und Petropoulos (2015)	✓	✓	✓	x	Trainingsfehler Validierungsfehler	x	x	✓	Random Walk Exponentielle Glättung Theta ARIMA	✓	x	x
Kourentzes et al. (2019)	✓	✓	✓	x	Validierungsfehler	x	x	✓	Exponentielle Glättung	✓	x	x
Kück und Freitag (2021)	✓	✓	x	x	Validierungsfehler	x	x	✓	k -nächste-Nachbarn	✓	✓	✓

Agrawal, 2014; Fonseca und Gómez-Gil, 2014; Fildes und Petropoulos, 2015; Kourentzes et al., 2019; Kück und Freitag, 2021). Der Vergleich der Prognosefehler anhand der Trainingsmenge wird auch als *Methode der kleinsten Quadrate* (engl. least squares method) bezeichnet. Es werden diverse Prognosemodelle $a^j(\mathbf{D}_{\text{train}}) = \hat{m}^j$ trainiert und anschließend wird jeweils die Summe der quadrierten Residuen ermittelt (vgl. Hyndman und Athanasopoulos, 2018):

$$\text{RSS}(\hat{m}_1^j, \mathbf{y}_{\text{train}}) = \sum_{t=1}^T \left(y_t - \hat{m}_1^j(y_1, \dots, y_{t-1}) \right)^2. \quad (4.9)$$

Die erwartete Rangfolge der Prognoseverfahrenskandidaten wird dann aufsteigend nach diesen Werten geschätzt, sodass das Verfahren $a^{(j)}$ mit der j -niedrigsten RSS den j -ten erwarteten Rang \hat{R}_j erhält. Bei der Selektion eines einzelnen Prognosemodells wird in diesem Fall somit das Prognoseverfahren selektiert, welches die minimale RSS erzielt. Dieses Selektionskriterium ist intuitiv naheliegend und lässt sich automatisch anwenden. Allerdings führt eine Verwendung dieses Kriteriums in der Regel zu einer Überanpassung an die Trainingsdaten (siehe Abschnitt 3.1.2). Daher wird dieses Selektionskriterium in der vorliegenden Arbeit nicht verwendet. Zur Vermeidung einer Überanpassung an Trainingsdaten wird stattdessen eine erwartete Rangfolge durch einen *Vergleich von Validierungsfehlern* bestimmt. Dieses Vorgehen wird auch als *Zeitreihenkreuzvalidierung* bezeichnet und wurde in Abschnitt 3.2.2 formal beschrieben. Hierzu wird die vorhandene Trainingsmenge in eine Trainings- und eine Validierungsmenge unterteilt. Dann werden Prognosemodelle anhand der Trainingsmenge trainiert und diese Modelle werden verwendet, um die Punkte in der Validierungsmenge im Rahmen von fixierten oder rollierenden h -Schritt-Prognosen

vorherzusagen. Anschließend werden die mittleren Prognosefehler ermittelt und anhand dieser wird die erwartete Rangfolge gebildet. Hinsichtlich der Fehlerfunktion lassen sich die gängigen Metriken, wie beispielsweise der sMAPE verwenden (vgl. Abschnitt 6.1.3). Anhand der erwarteten Rangfolge lassen sich ein oder mehrere Prognoseverfahren selektieren. In den Evaluationsstudien dieser Arbeit werden erwartete Rangfolgen anhand mittlerer Validierungsfehler hinsichtlich des RO-sMAPE gebildet und darauffolgend werden verschiedene Selektionsfunktionen verwendet. Der Ansatz zur Selektion des Verfahrens mit dem besten erwarteten Rang für eine Zeitreihe wird Val-Sel genannt, während die Pooling Ansätze zur Selektion von M_{sel} Verfahren und zur Kombination der resultierenden Modelle als Pool- M_{sel} bezeichnet werden.

Fildes und Petropoulos (2015) zeigten, dass sich Vergleiche von Validierungsfehlern besser als Modellselektionskriterium eignen als Vergleiche von Trainingsfehlern. Dieses Ergebnis ist im Einklang mit den Untersuchungen von Shmueli (2010), die erläuterten, dass gute erklärende Modelle nicht notwendigerweise gut für die Prognose zukünftiger Werte geeignet sind (vgl. Abschnitt 3.1.2). Zudem zeigten Fildes und Petropoulos (2015), dass der mittlere Validierungsfehler über unterschiedliche Horizonte ein besseres Selektionskriterium ist als der mittlere Validierungsfehler von Einzschrittprognosen und der mittlere Fehler für spezifische Horizonte.

Die Artikel in Tabelle 4.2 lassen sich in Ansätze zur Selektion von Prognoseverfahren oder -modellen (vgl. Fildes, 1989; Franses und Koehler, 1998; Gardner, 1999; Poler und Mula, 2011; Kück und Freitag, 2021), Ansätze zur Selektion der Gewichte für Prognosekombinationen (vgl. Adhikari und Agrawal, 2014; Fonseca und Gómez-Gil, 2014) sowie Artikel unterteilen, in denen Selektions- und Kombinationsansätze verglichen werden (vgl. Fildes und Petropoulos, 2015; Kourntzes et al., 2019). Hybride Ansätze, die basierend auf Prognosefehlervergleichen entweder ein Modell selektieren oder mehrere Modelle kombinieren, existieren bislang nicht. Fildes und Petropoulos (2015) zeigten, dass sowohl Selektions- als auch Kombinationsansätze Vor- und Nachteile aufweisen. Während in Fällen eines dominanten Prognoseverfahrens eine Selektion sinnvoll ist, bietet eine Kombination den Vorteil der Verringerung des Risikos zur Selektion eines ungeeigneten Verfahrens. Kourntzes et al. (2019) stellten den sogenannten Forecast-Islands-Ansatz vor, einen Pooling-Ansatz mit einer automatischen, datengetriebenen Bestimmung der Anzahl M_{sel} zu selektierender Modelle. Dabei wird zunächst eine erwartete Rangfolge aller verschiedenen Formen exponentieller Glättungsmodelle hinsichtlich des Validierungsfehlers gebildet. Darauffolgend werden Fehlerdifferenzen zwischen Modellen ermittelt, die in der erwarteten Rangfolge aufeinander folgen und es werden alle Modelle bis zum ersten starken Anstieg der Differenzen selektiert. Die Prozedur beginnt mit einer Menge aus zwei Differenzwerten, einer Null für das beste erwartete Modell sowie der Fehlerdifferenz der beiden besten Modelle der erwarteten Rangfolge. Danach werden die folgenden Schritte iterativ durchgeführt: (i) Hinzufügen der nächsten Fehlerdifferenz zur Menge und (ii) Testen, ob die Menge einen Ausreißer hinsichtlich der gängigen Definition eines Boxplots enthält, d.h. die Menge der Fehlerdifferenzen enthält einen Wert, der größer ist als das dritte Quartil plus den 1,5-fachen Interquartilsabstand. Die Prozedur endet, sobald der erste

Ausreißer identifiziert wurde und selektiert alle Modelle vor demjenigen, welches zum Ausreißer führte. Es sei angemerkt, dass eine Stichprobe aufgrund der gängigen Definitionen empirischer Quantile (vgl. Hyndman und Fan, 1996) nur Ausreißer enthalten kann, wenn sie mindestens fünf Werte umfasst. Daher ist der Forecast-Islands-Ansatz äquivalent zu einer Kombination von M Modellen, falls $M < 5$ gilt und es werden $M_{\text{sel}} \leq M$ Modelle selektiert, falls $M \geq 5$ gilt. In empirischen Untersuchungen zeigten Kourentzes et al. (2019), dass verschiedene Pooling-Ansätze eine höhere Prognosegüte erzielten als ein Selektionsansatz und eine Kombination aller Modelle.

Alle Ansätze in Tabelle 4.2 basieren auf Vergleichen von Prognosefehlern und somit nutzen alle als Merkmale Landmarkers, die die Prognosegüte verschiedener Verfahren und Modelle kennzeichnen (vgl. Pfahringer et al., 2000). Gardner (1999) betrachtete zusätzlich modellbasierte Merkmale. Direkte Merkmale wurden für keinen der Ansätze verwendet. Während die frühen Ansätze häufig nur wenige Prognosemodellklassen, wie exponentielle Glättungen, ARIMA-Modelle oder klassische Dekompositionsansätze betrachteten (vgl. Fildes, 1989; Franses und Koehler, 1998; Gardner, 1999; Poler und Mula, 2011), verwenden die meisten der aktuelleren Ansätze auch andere Modellklassen, wie neuronale Netze, Support Vector Machines, Theta-Modelle oder lokale k -nächste-Nachbarn-Modelle (vgl. Adhikari und Agrawal, 2014; Fonseca und Gómez-Gil, 2014; Fildes und Petropoulos, 2015; Kück und Freitag, 2021). Alle Ansätze wurden anhand von Prognosefehlern in empirischen Studien evaluiert, während Bestandssimulationen und Rechenaufwände nur in Kück und Freitag (2021) analysiert wurden.

4.3.3 Regelbasierte Expertensysteme

Da zur Anwendung von Prognosefehlervergleichen alle Prognoseverfahrenskandidaten für jede Zeitreihe verwendet werden müssen, um Modelle zu trainieren, sind darauf basierende Selektionsverfahren in der Regel sehr rechenintensiv. Im Gegensatz dazu beruhen regelbasierte Expertensysteme und Meta-Lernansätze auf der Annahme, dass die Eignung bestimmter Prognoseverfahren von den Charakteristika einer Zeitreihe abhängt. Während Meta-Lernansätze im nächsten Abschnitt beschrieben werden, zeigt Tabelle 4.3 verschiedene Ansätze zur Selektion von Prognosemodellen durch regelbasierte Expertensysteme.

Der erste Ansatz zur Prognosemodellselektion mit einem regelbasierten Expertensystem, welcher große Aufmerksamkeit im Wissenschaftsgebiet der Prognoseverfahren erhielt, wurde Rule-Based Forecasting (RBF) genannt (vgl. Collopy und Armstrong, 1992; Adya, 2000). Für diesen Ansatz entwickelten Experten 99 Regeln und nutzten diese, um die Gewichte für eine Kombination von vier simplen Prognosemodellen zu bestimmen, einem Random Walk, einem linearen Regressionsmodell sowie zwei unterschiedlichen exponentiellen Glättungsmodellen. Die Regeln basieren auf 18 direkten und modellbasierten Merkmalen. Da diese Merkmale manuell durch Experten identifiziert werden müssen, ist es schwierig, unterschiedliche Anwendungen zu vergleichen. Durch eine spätere Modifikation des Ansatzes wurde dieser auf 28 Merkmale erweitert, von denen einige sich automatisch extrahieren lassen (vgl. Adya, 2000; Adya et al., 2001; Adya und

Tabelle 4.3: Ansätze zur Modellselektion durch regelbasierte Expertensysteme

Referenz	Ansatz				Selektionskriterien	Merkmale			Prognosemodellklassen	Evaluation		
	Automatisch	Selektion	Kombination	Hybrid		Direkt	Modellbasiert	Landmarking		Prognosefehler	Bestandssimulation	Rechenaufwand
Collopy und Armstrong (1992) Adya (2000) Adya et al. (2000) Adya et al. (2001) Adya und Lusk (2016)	x	✓	x	x	Rule-Based Forecasting	✓	✓	x	Random Walk Lineare Regression Exponentielle Glättung	✓	x	x
Vokurka et al. (1996) Flores und Pearce (2000)	✓	✓	✓	✓	Rule-Based Expert Forecasting System	x	✓	✓	Exponentielle Glättung Klassische Dekomposition Gleitende Mittelwerte	✓	x	x
Sikora et al. (2013a) Sikora et al. (2013b) Štěpničková et al. (2013)	✓	x	✓	x	Fuzzy Rule-Based Ensemble	✓	✓	x	Dekomposition Exponentielle Glättung ARIMA GARCH Gleitende Mittelwerte Random Walk	✓	x	x
Petropoulos et al. (2014)	x	✓	✓	x	Graphisches Methodenselektionsprotokoll	✓	✓	x	Random Walk Exponentielle Glättung Theta Lineare Regression Forecast Pro Autobox	✓	x	x
Štěpnička et al. (2014)	✓	x	✓	x	Fuzzy Rule-Based Ensemble	✓	✓	x	ARIMA Dekomposition Exponentielle Glättung Random Walk	✓	x	x
Burda et al. (2015)	✓	x	✓	x	Fuzzy Rule-Based Ensemble	✓	✓	x	ARIMA Exponentielle Glättung Random Walk Theta	✓	x	x

Lusk, 2016). Allerdings müssen auch für diese Modifikation einige Merkmale manuell identifiziert werden, wodurch das Expertensystem sich nicht automatisch verwenden lässt. RBF betrachtet keinerlei Prognosefehler für die Selektion, was zum Entwicklungszeitpunkt unüblich war, da zu diesem Zeitpunkt Prognosefehlervergleiche das gängigste Selektionskriterium darstellten. Obwohl RBF als sehr innovativer Ansatz angesehen wurde, war die erreichte Prognosegüte in der M3-Competition nur durchschnittlich (vgl. Makridakis und Hibon, 2000).

Um die Notwendigkeit eines manuellen Experteninputs zu reduzieren, entwickelten Vokurka et al. (1996) einen automatischen Ansatz, der als Rule-Based Expert Forecasting System benannt wurde. Nach einer Identifikation irrelevanter Startdaten, einer Detektion und Anpassung von Ausreißern sowie einer Identifikation von Trend und Saisonalität wird bei diesem Ansatz ein System aus 84 Regeln genutzt, um eines der folgenden Prognoseverfahren zu selektieren: einfache exponentielle Glättung, Gardners exponentielle Glättung mit gedämpftem Trend, klassische Dekomposition oder eine gleichgewichtete Kombination der drei Verfahren. Somit ist dies der einzige Ansatz der regelbasierten Expertensysteme, der als hybrider Ansatz zwischen Selektion und Kombination bezeichnet werden kann. Allerdings steht als Kombination lediglich die Kombination

aller Verfahren zur Auswahl, während weitere individuelle Kombinationen nicht möglich sind. Nach einer automatischen Selektion eines Verfahrens durch das Expertensystem besteht die Möglichkeit, manuelle Anpassungen vorzunehmen. In einer Evaluation anhand der Daten der M-Competition erreichte das entwickelte Expertensystem eine ähnliche Prognosegüte, wie RBF (vgl. Vokurka et al., 1996). Zudem nahmen Flores und Pearce (2000) mit einer modifizierten Version des Systems an der M3-Competition teil, erzielten aber im Durchschnitt schlechtere Ergebnisse als RBF (vgl. Makridakis und Hibon, 2000).

Aktuellere Ansätze zur Selektion durch regelbasierte Expertensysteme sind Fuzzy Rule-Based Ensembles, welche sich zur Selektion von Gewichten für Prognosekombinationen verwenden lassen (vgl. Sikora et al., 2013a,b; Štěpničková et al., 2013; Štěpnička et al., 2014; Burda et al., 2015). Diese Ansätze betrachten ebenfalls direkte und modellbasierte Merkmale. Im Gegensatz zu RBF werden hierbei mehrere unterschiedliche Modellklassen einbezogen, wie exponentielle Glättungsmodelle, ARIMA-Modelle, Random Walks oder GARCH-Modelle. Die Ansätze ermöglichen eine automatische Anwendung ohne Experteninput und erzielten vielversprechende Ergebnisse in unterschiedlichen Studien.

Basierend auf einer Regressionsanalyse entwickelten Petropoulos et al. (2014) ein graphisches Methodenselektionsprotokoll zur Bestimmung der Eignung verschiedener Prognoseverfahren für unterschiedliche Datenbedingungen hinsichtlich sieben verschiedener Zeitreihenmerkmale (Saisonalität, Trend, Zykluskomponente, Zufälligkeit, Länge, mittlere Zeit zwischen zwei Nachfragen und quadrierter Variationskoeffizient) sowie der strategischen Entscheidung hinsichtlich des Prognosehorizontes. Die Auswirkung jedes Merkmals auf die Prognosegüte wurde dafür in eine 11-stufige Skala übersetzt. Allerdings basiert die finale Selektion eines Prognoseverfahrens auf den Präferenzen des Experten. Daher lässt sich das graphische Selektionsprotokoll zwar als Richtlinie, jedoch nicht als automatisches Selektionsverfahren verwenden und es kann zu verschiedenen Ergebnissen für unterschiedliche Nutzer führen.

Alle regelbasierten Expertensysteme in Tabelle 4.3 wurden in empirischen Studien anhand von Prognosefehlern evaluiert. Bestandssimulationen oder Rechenaufwände wurden nicht betrachtet.

4.3.4 Meta-Lernansätze

Der Begriff *Meta-Lernen* (engl. meta-learning) wird in verschiedenen Wissenschaftsdisziplinen unterschiedlich definiert. Eine frühe Definition aus dem Bereich der Bildungspsychologie beschreibt den Begriff beispielsweise als „Fähigkeit, die Kontrolle über das eigene Lernen auszuüben“ (Biggs, 1985). Eine ähnlich frühe Definition aus dem Bereich der Informatik beschreibt Meta-Lernen als „lernen zu lernen“ (Schmidhuber, 1987). Diesen Definitionen entsprechend bezeichnet Meta-Lernen also nicht das Erlernen konkreter Sachkenntnis, sondern das Erlernen der Fähigkeit, Sachkenntnisse aus verschiedenen Bereichen lernen zu können. Diese abstrakte Definition lässt sich auf den Bereich des Meta-Lernens als Teilgebiet des maschinellen Lernens übertragen. In diesem Zusammenhang wird Meta-Lernen als ein Ansatz zur Lösung des Algorithmusselektionsproblems

verstanden (vgl. Rice, 1976; Vilalta und Drissi, 2002; Brazdil et al., 2008; Smith-Miles, 2008; Lemke et al., 2015). Bei der Verwendung von Meta-Lern-Ansätzen werden sogenannte *Meta-Modelle* (engl. meta-models, meta-learner) trainiert, um diese darauffolgend zur Selektion geeigneter Verfahren oder Modelle für neue Probleminstanzen verwenden zu können. Dabei lassen sich die Meta-Modelle entweder als Regressionsmodelle oder als Klassifikationsmodelle konfigurieren. Falls ein Meta-Modell als Regressionsmodell konfiguriert ist, wird es in der Regel zur Vorhersage der Prognosegüte verschiedener Verfahrenskandidaten für eine neue Probleminstanz verwendet. Im Falle der Konfiguration als Klassifikationsmodell bestimmt das Meta-Modell Eignungswahrscheinlichkeiten verschiedener Verfahren für eine Probleminstanz. Während die meisten Anwendungen des Meta-Lernens im Bereich des maschinellen Lernens die Selektion von Klassifikationsverfahren behandeln (vgl. Brazdil et al., 1994, 2008; Smith-Miles, 2008; Reif et al., 2014; van Rijn et al., 2014, 2015), existieren ebenfalls Meta-Lern-Anwendungen zur Selektion von Zeitreihenprognoseverfahren, welche in Tabelle 4.4 dargestellt sind.

Tabelle 4.4: Ansätze zur Modellselektion durch Meta-Lernen

Referenz	Ansatz				Selektionskriterien	Merkmale			Prognosemodellklassen	Evaluation		
	Automatisch Selektion	Kombination	Hybrid			Direkt	Modellbasiert	Landmarking		Prognosefehler	Bestandsimulation	Rechenaufwand
Chu und Widjaja (1994)	✓	✓	x	x	Neuronale Netze	✓	✓	x	Exponentielle Glättung	✓	x	x
Sohl und Venkatachalam (1995) Venkatachalam und Sohl (1999)	✓	✓	x	x	Neuronale Netze	✓	✓	x	Random Walk Gleitende Mittelwerte Exponentielle Glättung Lineare Regression	✓	x	x
Arinze et al. (1997)	✓	✓	✓	✓	Regelbasierte Induktion mit Ereignisbäumen	✓	✓	x	Adaptive Filtrierung Exponentielle Glättung Gleitende Mittelwerte	✓	x	x
Shah (1997)	✓	✓	x	x	Lineare Diskriminanzanalyse Quadratische Diskriminanzanalyse Diskriminanzanalyse mit Kerndichteschätzung	✓	✓	x	Exponentielle Glättung Strukturmodelle	✓	x	x
Anandarajan und Arinze (1999)	✓	✓	x	x	Neuronale Netze	✓	✓	x	Gleitende Mittelwerte Exponentielle Glättung Adaptive Filtrierung Dekomposition	✓	x	x
Meade (2000b)	✓	✓	✓	x	Lineare Regression	✓	✓	x	Gleitende Mittelwerte Random Walk Lineare Regression Exponentielle Glättung ARIMA ARARMA	✓	x	x
dos Santos et al. (2004)	✓	✓	x	x	Zoomed Ranking	✓	✓	x	Random Walk Exponentielle Glättung Autoregressive Modelle	✓	x	✓
Prudêncio und Ludermir (2004)	✓	✓	x	x	Entscheidungsbäume NOEMON	✓	✓	x	Exponentielle Glättung Neuronale Netze Random Walk Autoregressive Modelle	✓	x	x

Tabelle 4.4: Ansätze zur Modellselektion durch Meta-Lernen (Fortsetzung)

Referenz	Ansatz				Selektionskriterien	Merkmale			Prognosemodellklassen	Evaluation		
	Automatisch	Selektion	Kombination	Hybrid		Direkt	Modellbasiert	Landmarking		Prognosefehler	Bestandssimulation	Rechenaufwand
Lemke (2008) Lemke und Gabrys (2010a) Lemke und Gabrys (2010b)	✓	✓	✓	✓	Ereignisbäume Neuronale Netze Support Vector Machine Zoomed Ranking Selektion Zoomed Ranking Kombination	✓	✓	✓	Exponentielle Glättung ARIMA Neuronale Netze Gleitende Mittelwerte Strukturmodelle Lineare Regression Theta	✓	x	x
Villarreal Marroquín et al. (2009)	✓	✓	x	x	Neuronale Netze	✓	✓	x	Random Walk Mittelwerte Gleitende Mittelwerte Exponentielle Glättung ARIMA Lineare Regression	✓	x	x
Wang et al. (2009)	✓	x	✓	x	Entscheidungsbäume	✓	✓	x	Random Walk Exponentielle Glättung ARIMA Neuronale Netze	✓	x	x
Matijaš et al. (2013)	✓	✓	x	x	Ensemble: Euklidische Distanz CART Entscheidungs-bäume LVQ-Netzwerk Neuronale Netze Support Vector Machine Gaußsche Prozesse	✓	✓	x	Random Walk ARMA Algorithmus ähnlicher Tage Neuronale Netze Support Vector Regression	✓	x	x
Rossi et al. (2014)	✓	✓	x	x	Random Forest <i>k</i> -nächste-Nachbarn Naive Bayes	✓	x	x	Random Forest Support Vector Machine Klassifikations- und Regressionsbäume Project Pursuit Regression Multivariate adaptive Regressionssplines	✓	x	x
Scholz-Reiter et al. (2014)	✓	✓	x	x	Lineare Diskriminanz-analyse	✓	✓	x	Random Walk ARIMA Neuronale Netze Exponentielle Glättung <i>k</i> -nächste-Nachbarn	✓	x	x
Fonseca und Gomez (2016)	✓	x	✓	x	Self-Organizing Map	x	x	✓	Neuronale Netze ARIMA	✓	x	x
Kück et al. (2016b)	✓	✓	x	x	Neuronale Netze	✓	✓	✓	Exponentielle Glättung	✓	x	x
Talagala et al. (2018)	✓	✓	x	x	Random Forest	✓	✓	x	Weißes Rauschen ARIMA Random Walk Theta Exponentielle Glättung STL-AR SARIMA	✓	x	x
Villegas et al. (2018)	✓	✓	✓	x	Support Vector Machine	✓	✓	x	Random Walk Gleitende Mittelwerte Exponentielle Glättung ARIMA	✓	x	x
Montero-Manso et al. (2020)	✓	x	✓	x	Gradient Boosting	✓	✓	x	Random Walk Theta ARIMA Exponentielle Glättung TBATS STLM-AR	✓	x	x

Tabelle 4.4: Ansätze zur Modellselektion durch Meta-Lernen (Fortsetzung)

Referenz	Ansatz				Selektionskriterien	Merkmale			Prognosemodellklassen	Evaluation		
	Automatisch	Selektion	Kombination	Hybrid		Direkt	Modellbasiert	Landmarking		Prognosefehler	Bestandssimulation	Rechenaufwand
Ma und Fildes (2021)	✓	✓	✓	x	Neuronale Netze	✓	✓	x	Exponentielle Glättung Multiple lineare Regression ARIMAX Support Vector Regression Neuronale Netze Random Forest Gradient Boosting	✓	x	x

Im Bereich der Zeitreihenprognose wurde der Begriff des Meta-Lernens erstmals durch Prudêncio und Ludermir (2004) verwendet. Die Meta-Lernansätze in Tabelle 4.4 verwenden verschiedene Konfigurationen hinsichtlich der Meta-Modelle, Selektionsfunktionen, Merkmalsmengen und Prognosemodellklassen. Während einige Ansätze statistische Verfahren, wie Diskriminanzanalysen (vgl. Shah, 1997; Scholz-Reiter et al., 2014) oder Regressionsverfahren (vgl. Meade, 2000b; Petropoulos et al., 2014) nutzen, um Meta-Modelle abzuleiten, verwenden andere Ansätze maschinelle Lernverfahren, wie Entscheidungsbäume (vgl. Arinze et al., 1997; Prudêncio und Ludermir, 2004; Lemke, 2008; Lemke und Gabrys, 2010a; Wang et al., 2009), Random Forests (vgl. Rossi et al., 2014; Talagala et al., 2018), Gradient Boosting (vgl. Montero-Manso et al., 2020), Support Vector Machines (vgl. Lemke und Gabrys, 2010a; Villegas et al., 2018), neuronale Netze (vgl. Chu und Widjaja, 1994; Sohl und Venkatachalam, 1995; Anandarajan und Arinze, 1999; Venkatachalam und Sohl, 1999; Prudêncio und Ludermir, 2004; Villarreal Marroquín et al., 2009; Kück et al., 2016b; Ma und Fildes, 2021), Self-Organizing Maps (vgl. Fonseca und Gomez, 2016), NOEMON (vgl. Prudêncio und Ludermir, 2004), Zoomed Ranking (vgl. dos Santos et al., 2004; Lemke und Gabrys, 2010a,b) oder Kombinationen mehrerer Verfahren (vgl. Matijaš et al., 2013).

Die meisten Meta-Lernansätze lassen sich zur individuellen Selektion von Prognosemodellen verwenden (vgl. Chu und Widjaja, 1994; Sohl und Venkatachalam, 1995; Shah, 1997; Anandarajan und Arinze, 1999; Venkatachalam und Sohl, 1999; dos Santos et al., 2004; Prudêncio und Ludermir, 2004; Villarreal Marroquín et al., 2009; Matijaš et al., 2013; Rossi et al., 2014; Scholz-Reiter et al., 2014; Kück et al., 2016b; Talagala et al., 2018), während andere zur Selektion von Gewichten für Prognosekombinationen genutzt werden (vgl. Wang et al., 2009; Fonseca und Gomez, 2016; Montero-Manso et al., 2020) und in einigen Artikeln Selektions- und Kombinationsansätze verglichen werden (vgl. Meade, 2000b; Villegas et al., 2018; Ma und Fildes, 2021). Neben diesen Ansätzen, bei denen stets vor einer Selektion festgelegt wird, ob ein einzelnes oder mehrere Verfahren für eine Kombination selektiert werden sollen, wurden bislang zwei hybride Meta-Lernansätze entwickelt, die in Abhängigkeit vom Output des jeweiligen Meta-Modells für jede Zeitreihe zwischen einer Selektion und einer Kombination wählen (vgl. Arinze et al.,

1997; Lemke und Gabrys, 2010a). Allerdings stehen bei beiden Ansätzen lediglich wenige vorher zusammengestellte Kombinationen zur Auswahl und es können nicht alle individuellen Kombinationen der zur Auswahl stehenden Modelle gewählt werden. Arinze et al. (1997) verwendeten einen Entscheidungsbaum als Meta-Modell, um eines von sechs möglichen Verfahren im Rahmen eines Klassifikationsproblems auszuwählen. Die Menge der sechs Verfahrenskandidaten besteht hierbei aus einer adaptiven Filtrierung (AD), Holts exponentieller Glättung (HT), Winters exponentieller Glättung (WT), einer Kombination aus einem gleitenden Mittelwert und HT, einer Kombination aus AD und WT sowie einer Kombination aus HT und WT. Der vorgestellte Meta-Lernansatz, der für jede Zeitreihe zwischen diesen sechs Verfahrenskandidaten wählt, erzielte in einer Evaluationsstudie geringere Prognosefehler als die sechs Verfahrenskandidaten selbst (vgl. Arinze et al., 1997). Lemke (2008) konfigurierten mehrere Meta-Lernansätze zur Selektion von Prognoseverfahren sowie zur Selektion von Gewichten für Prognosekombinationen und entwickelten die Ansätze in späteren Artikeln weiter (vgl. Lemke und Gabrys, 2010a,b). Für den Selektionsansatz stehen hierbei vier verschiedene Labels zur Auswahl: ein Strukturmodell, ein neuronales Netz und zwei verschiedene Ansätze zur Kombination von 15 Prognoseverfahren mit unterschiedlichen Gewichten. Dieser Selektionsansatz wurde in einer Evaluationsstudie mit einem Kombinationsansatz verglichen, der diese vier Labels gewichtet kombinierte. Dabei erzielte der Kombinationsansatz im Durchschnitt bessere Prognoseergebnisse als der Selektionsansatz.

Hinsichtlich der Merkmale zur Beschreibung der Selektionsprobleme nutzen die meisten Meta-Lernansätze direkte oder modellbasierte Merkmale, während lediglich wenige der Ansätze Landmarkers verwenden (vgl. Lemke und Gabrys, 2010a,b; Fonseca und Gomez, 2016; Kück et al., 2016b). Allerdings wurde in einer Vorstudie zur vorliegenden Arbeit gezeigt, dass eine Einbeziehung von Landmarkers großes Potenzial zur Verbesserung der Selektionsgüte eines Selektionsverfahrens hat Kück et al. (vgl. 2016b). Während frühe Meta-Lernansätze zur Selektion von Prognosemodellen größtenteils statistische Modellklassen, wie exponentielle Glättungen oder ARIMA-Modelle betrachteten, verwenden aktuellere häufig außerdem neuronale Netze oder andere maschinelle Lernverfahren als Verfahrenskandidaten. Alle Meta-Lernansätze wurden anhand von Prognosefehlern evaluiert, während Bestandssimulationen nicht betrachtet wurden und Rechenaufwände nur von dos Santos et al. (2004) untersucht wurden.

Meta-Lernansätze zur Prognosemodellselektion erzielten bereits vielversprechende Ergebnisse in empirischen Studien, wie beispielsweise in Evaluationsstudien anhand der M3-Competition (vgl. Scholz-Reiter et al., 2014), der NN5-Competition (vgl. Lemke und Gabrys, 2010a), der NN3-Competition (vgl. Lemke und Gabrys, 2010a; Kück et al., 2016b) oder der NNGC1-Competition (vgl. Lemke und Gabrys, 2010b). Talagala et al. (2018) trainierten ein Meta-Modell basierend auf den Daten der M-Competition und nutzten dies zur Modellselektion für die Daten der M3-Competition und umgekehrt. In beiden Experimenten funktionierte der Ansatz gut, jedoch wurden bessere Ergebnisse beim Training anhand der größeren und dem Test anhand der kleineren Datenmenge erzielt. Montero-Manso et al. (2020) nutzten ein Gradient-Boosting-Verfahren, um

ein Meta-Modell zu trainieren, welches zur Selektion von Gewichten für eine Kombination diverser Prognoseverfahren eingesetzt wurde und den zweiten Platz in der M4-Competition erreichte. Ma und Fildes (2021) trainierten ein faltendes neuronales Netz als Meta-Modell, um automatisch Merkmale aus Zeitreihen zu extrahieren und darauffolgend Gewichte für eine Prognosekombination zu selektieren. Der Ansatz erreichte vielversprechende Ergebnisse bei der Prognose von Einzelhandelsumsätzen.

4.4 Taxonomie verschiedener Ansätze zur Selektion von Prognosemodellen

Basierend auf dem Stand der Forschung zur Selektion von Prognosemodellen wird in diesem Abschnitt eine neuartige Taxonomie gebildet, um die bestehenden Ansätze zu klassifizieren. Die Ansätze lassen sich in aggregierte und individuelle Ansätze sowie Selektions- und Kombinationsansätze einteilen. Abbildung 4.2 zeigt eine schematische Taxonomie, um die verschiedenen Ansätze hinsichtlich ihrer Rechenaufwände pro Modellselektion und pro Prognose zu vergleichen. In diesem Zusammenhang besteht der Rechenaufwand pro Modellselektion aus allen Schritten, die zur Selektion eines Verfahren und zum Training eines Modells notwendig sind, das zur Prognose genutzt werden soll. Hierbei kann das finale Modell entweder ein einzelnes Prognosemodell sein oder eine Menge von Prognosemodellen, die kombiniert werden. Beispielsweise umfasst das Selektionsverfahren Val-Sel, welches ein Prognosemodell nach einem Vergleich der Validierungsfehler von Verfahrenskandidaten selektiert, die folgenden Schritte: (i) Verwendung aller Prognoseverfahrenskandidaten zum Training von Prognosemodellen anhand einer Trainingsmenge, (ii) Verwendung aller trainierten Modelle zur Prognose der Werte einer Validierungsmenge, (iii) Vergleich der Validierungsfehler der verschiedenen Modelle, (iv) Selektion des Modells mit dem geringsten Validierungsfehler und möglicherweise (v) erneute Verwendung des selektierten Prognoseverfahrens zum Training eines Modells anhand aller Werte der Trainings- und der Validierungsmenge. Hingegen für eine Kombination von M bereits ausgewählten Prognoseverfahren besteht der Rechenaufwand zur Modellselektion lediglich aus der Summe der Rechenaufwände zur Verwendung aller Verfahren, um M Prognosemodelle zu trainieren. In der Regel wird der Rechenaufwand zum Training eines Modells einmalig pro Zeitreihe benötigt und das Modell kann danach zur Prognose mehrerer zukünftiger Werte der Zeitreihe verwendet werden. Im Gegensatz dazu wird der Rechenaufwand pro Prognose für jeden vorherzusagenden Wert einer Zeitreihe benötigt. Dieser besteht aus den folgenden Aufwänden: (i) Verwendung aller selektierten Modelle zur Prognose der zukünftigen Zeitreihenwerte und möglicherweise (ii) Kombination der Prognosen multipler Modelle zu einer finalen Prognose. Die verschiedenen Ansätze lassen sich anhand der folgenden Notation vergleichen:

(Selektionstyp | M_{sel} | C | Selektionskriterium).

Hierbei kann der Selektionstyp entweder aggregiert (agg) oder individuell (ind) sein. M_{sel} bezeichnet die Anzahl zu selektierender Prognosemodelle aus M Kandidaten und C bezeichnet die Anzahl verschiedener Modellklassen aus denen selektiert werden kann. Wie in Abschnitt 4.3 beschrieben wurde, können als Selektionskriterien entweder Informationskriterien, Prognosefehlervergleiche, regelbasierte Expertensysteme oder Meta-Lernansätze verwendet werden. Abbildung 4.2 ordnet die verschiedenen Ansätze hinsichtlich des Rechenaufwands zur Modellselektion für eine Menge an Zeitreihen und des Rechenaufwands pro Prognose. Hierbei wird der letzte Punkt der Notation, das Selektionskriterium, nicht formal betrachtet, da die schematische Taxonomie ansonsten unnötig komplex würde.

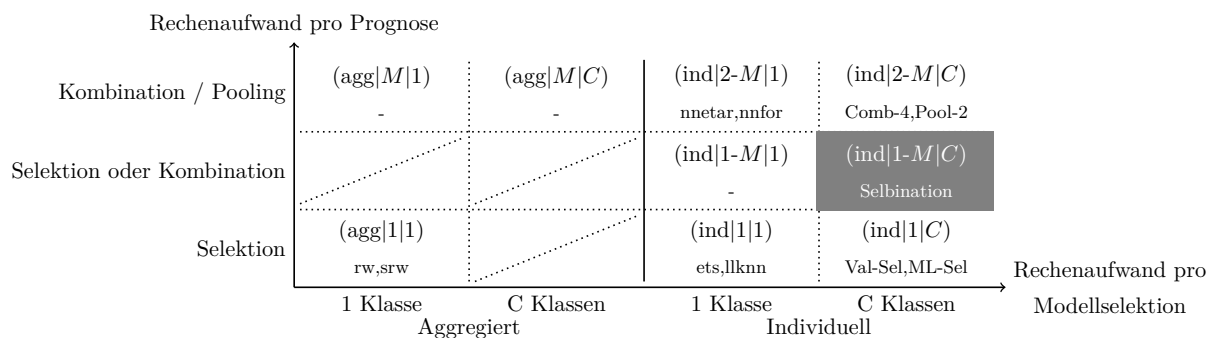


Abbildung 4.2: Taxonomie verschiedener Ansätze zur Lösung des Prognosemodellselektionsproblems

Der einfachste Ansatz zur Selektion von Prognosemodellen ist der Ansatz der aggregierten Selektion, bei dem ein einzelnes Prognosemodell selektiert wird und für alle Zeitreihen aus einer Datenmenge verwendet wird (vgl. Fildes, 1989). Es wird ein geringer Rechenaufwand zur Modellselektion benötigt und der Ansatz wird häufig in der industriellen Praxis verwendet, wenn es an Expertenwissen fehlt. Allerdings kann dieser Ansatz zu einer geringen Prognosegüte führen, falls das selektierte Modell für einige Zeitreihen aus der Datenmenge ungeeignet ist. In diesem Zusammenhang bietet der Ansatz der individuellen Selektion spezifischer Modelle für jede einzelne Zeitreihe das Potenzial, jede Zeitreihenentwicklung geeignet zu modellieren und somit eine höhere Prognosegüte zu erzielen (vgl. Fildes und Petropoulos, 2015). Wenn man eine große Datenmenge mit mehreren Zeitreihen betrachtet, benötigt dieser Ansatz einen höheren Rechenaufwand für die Modellselektion, als ein aggregierter Selektionsansatz, da für jede Zeitreihe ein individuelles Modell selektiert wird. Der Rechenaufwand pro Prognose hängt für aggregierte und individuelle Selektionsansätze vom letztlich selektierten Modell für eine Zeitreihe ab. In beiden Fällen wird ein einzelnes Modell pro Zeitreihe verwendet, um Prognosen zu berechnen.

Im Gegensatz zu Selektionsansätzen verwenden Kombinationsansätze mehrere Modelle und kombinieren deren Prognosen zu einer finalen Prognose (vgl. Bates und Granger, 1969; Elliott und Timmermann, 2016). Um eine Prognosekombination durchzuführen, muss die Anzahl zu selektierender Modelle M_{sel} definiert werden, wobei $2 \leq M_{\text{sel}} \leq M$ gilt. Falls $M_{\text{sel}} < M$ gilt, wird

ein Kombinationsansatz auch ein Pooling-Ansatz genannt (vgl. Aiolfi und Timmermann, 2006; Geweke und Amisano, 2011). Während aggregierte Kombinationsansätze die gleiche Menge an trainierten Modellen für alle Zeitreihen in einer Datenmenge verwenden, trainieren individuelle Selektionsansätze für jede Zeitreihe eine spezifische Menge an Prognosemodellen. Abhängig von der Anzahl zu selektierender Modelle M_{sel} ist der Rechenaufwand pro Prognose bei Kombinationsansätzen generell höher als bei Selektionsansätzen, während der Rechenaufwand zur Modellselektion vom spezifischen Ansatz abhängt.

In den Evaluationsstudien dieser Arbeit werden verschiedene Ansätze verwendet, von denen einige in die Taxonomie in Abbildung 4.2 eingeordnet sind: nicht-saisonale (rw) und saisonale Random Walks (srw) können als aggregierte Selektionsansätze betrachtet werden, da für jede Zeitreihe das gleiche Modell verwendet wird (vgl. Abschnitt 3.4.1 und Hyndman et al., 2018b). ets, auto.arima, llknn und lcknn sind individuelle Selektionsverfahren für die Modellklassen der exponentiellen Glättungsmodelle (vgl. Abschnitt 3.4.3 und Hyndman et al., 2008a, 2018b), der ARIMA-Modelle (vgl. Abschnitt 3.4.2 und Hyndman und Khandakar, 2008; Hyndman und Athanasopoulos, 2018) sowie der lokalen knn-Modelle (vgl. Abschnitt 3.4.4 und Kück und Freitag, 2021). Während diese Ansätze aus einer vorgegebenen Modellklasse selektieren, selektieren Val-Sel (vgl. Abschnitt 4.3.2 und Fildes und Petropoulos, 2015) und ML-Sel (vgl. Kapitel 5) ein Modell aus mehreren möglichen Modellklassen. Val-Sel verwendet mehrere Prognoseverfahrenskandidaten und selektiert das Verfahren mit dem geringsten mittleren Validierungsfehler, um ein Prognosemodell zu trainieren. ML-Sel verwendet das Meta-Modell, welches im nächsten Kapitel beschrieben wird, um das Prognoseverfahren mit der höchsten Eignungswahrscheinlichkeit für eine Zeitreihe zu verwenden. nnetar und nnfor sind individuelle Kombinationsansätze, die mehrere neuronale Netze trainieren und diese in Ensembles kombinieren (vgl. Abschnitt 3.4.5 und Kourentzes, 2017; Hyndman et al., 2018b). Comb-4 ist ein individueller Kombinationsansatz, bei dem die vier Prognoseverfahren ets, llknn, nnetar und srw verwendet werden und die Prognosen der Modelle kombiniert werden (vgl. Abschnitt 4.2.3 und Bates und Granger, 1969; Elliott und Timmermann, 2016).

Es sei angemerkt, dass die individuelle Selektion und die individuelle Kombination Generalisierungen der aggregierten Selektion und der aggregierten Kombination sind. Zudem kann eine Kombination als Generalisierung einer Selektion betrachtet werden, falls die minimale Anzahl zu selektierender Modelle nicht beschränkt wird. Allerdings zeigte die Darstellung des Standes der Forschung in Abschnitt 4.3, dass für nahezu alle bisher entwickelten Selektionsverfahren im Voraus entweder ($2 \leq M_{\text{sel}} \leq M$) oder $M_{\text{sel}} = 1$ vorgegeben wird. Im Gegensatz dazu wird im nächsten Kapitel dieser Arbeit ein hybrides Selektionsverfahren (Selbination) vorgestellt, welches basierend auf Eignungswahrscheinlichkeiten der verschiedenen Prognoseverfahrenskandidaten entweder eine individuelle Selektion oder eine individuelle Kombination durchführt.

4.5 Limitierungen und Best Practices bestehender Modellselektionsansätze

In diesem Kapitel wurde der Stand der Forschung hinsichtlich verschiedener Ansätze zur Selektion und Kombination von Prognosemodellen dargestellt und analysiert. Im Folgenden werden die Limitierungen und Best Practices der bestehenden Ansätze abgeleitet, wodurch der neuartige Ansatz dieser Arbeit motiviert wird. Das Hauptergebnis aus der Literaturanalyse ist, dass für nahezu alle bisher entwickelten Selektionsverfahren im Voraus, unabhängig von den spezifischen Charakteristika einer vorliegenden Zeitreihe und des Selektionsproblems, festgelegt wird, ob ein einzelnes Prognosemodell selektiert ($M_{\text{sel}} = 1$) oder mehrere Prognosemodelle kombiniert werden sollen ($M_{\text{sel}} > 1$). Es existieren bislang nur wenige hybride Selektionsverfahren, die datengetrieben in Abhängigkeit von den Charakteristika der vorliegenden Zeitreihe zwischen einzelnen Prognosemodellen und Prognosekombinationen entscheiden (vgl. Vokurka et al., 1996; Arinze et al., 1997; Lemke und Gabrys, 2010a). Allerdings stellen alle bisherigen hybriden Selektionsverfahren nur wenige vorher festgelegte Kombinationen von Prognosemodellen zur Auswahl. Hierdurch können weitere potenziell vielversprechende Kombinationen von Prognosemodellen nicht selektiert werden. Zur Ausschöpfung des vollen Potenzials eines hybriden Ansatzes, der datengetrieben ein oder mehrere geeignete Prognosemodelle selektiert, bietet das im nächsten Kapitel beschriebene sogenannte Selbinationsverfahren die Möglichkeit, jede Teilmenge einer gegebenen Menge an Prognoseverfahrenskandidaten für eine Zeitreihe selektieren zu können ($1 \leq M_{\text{sel}} \leq M$). Die Selektionsentscheidung basiert dabei auf den Charakteristika einer vorliegenden Zeitreihe sowie den Eignungswahrscheinlichkeiten der Verfahrenskandidaten, die durch ein Meta-Modell geschätzt werden. Dadurch ist das Selbinationsverfahren ein automatisches und gänzlich hybrides Verfahren zwischen Selektion und Kombination. Falls für eine Zeitreihe ein dominantes Prognoseverfahren erwartet wird, wird nur dieses selektiert. Wenn hingegen unklar ist, welches Verfahren zu guten Prognosen führen könnte oder mehrere ähnlich gute Verfahren erwartet werden, werden multiple Prognosemodelle kombiniert. Somit werden die Vorteile beider Ansätze in einem Verfahren vereint.

Neben der beschriebenen Limitierung bisheriger Selektionsverfahren, die das Hauptergebnis der Literaturanalyse darstellt, lassen sich mehrere Best Practices sowie einige weitere Limitierungen bestehender Ansätze identifizieren. Zusätzlich zu den bereits in Kapitel 3 beschriebenen gängigen Selektionsverfahren für vorgegebene Modellklassen wurden in diesem Kapitel gängige Verfahren zur Selektion aus mehreren möglichen Modellklassen vorgestellt. Die vielversprechendsten Ergebnisse in aktuellen Studien wurden durch Meta-Lernansätze erzielt. Daher basiert das in dieser Arbeit vorgestellte Selbinationsverfahren auf einem Meta-Lernansatz (vgl. Kapitel 5). Allerdings stellt die sinnvolle Konfiguration eines Meta-Lernansatzes bislang noch eine offene Forschungsfrage dar. Aus dem Stand der Forschung lässt sich ableiten, dass zum einen sowohl statistische als auch maschinelle Lernverfahren berücksichtigt werden sollten und zum ande-

ren sollten neben direkten und modellbasierten Merkmalen auch Landmarkers genutzt werden, um Selektionsprobleme zu beschreiben. Diese Best Practices werden bei der Konfiguration des Selbinationsverfahrens berücksichtigt. Zudem wird der gleiche Meta-Lernansatz mit einer weiteren Selektionsfunktion verwendet, sodass ein individuelles Selektionsverfahren (ML-Sel) als Benchmark für die empirischen Studien dieser Arbeit entsteht.

Neben Meta-Lernansätzen ist das gängigste Selektionskriterium für mehrere mögliche Modellklassen ein Vergleich von Validierungsfehlern. Daher wird als weiteres Benchmark-Verfahren für die empirischen Studien das Verfahren Val-Sel verwendet, welches eine erwartete Rangfolge der Prognoseverfahrenskandidaten hinsichtlich ihrer mittleren Validierungsfehler bildet und das Verfahren mit dem geringsten erwarteten Fehler selektiert. Zudem werden Pooling-Verfahren verwendet, die basierend auf der gleichen erwarteten Rangfolge mehrere Verfahren selektieren und die Prognosen der resultierenden Modelle kombinieren. Als weiteres gängiges Benchmark-Verfahren wird die Kombination aller Prognoseverfahrenskandidaten genutzt. Da gleichgewichtete Kombinationen ähnlich gute Prognosen erzielen, wie komplex gewichtete Kombinationen, erhalten alle Prognosekombinationen dieser Arbeit gleich verteilte Gewichte.

Eine weitere Limitierung bestehender empirischer Studien zu Modellselektionsansätzen ist, dass diese in der Regel lediglich anhand von mittleren Prognosefehlern evaluiert werden. In der vorliegenden Arbeit werden die genannten Limitierungen bestehender Ansätze dadurch adressiert, dass ein hybrides Meta-Lernverfahren zur automatischen Selektion oder Kombination geeigneter Prognosemodelle vorgestellt wird und anhand monatlicher industrieller Zeitreihen hinsichtlich mittlerer Prognosefehler sowie Bestandsmetriken und Rechenaufwänden evaluiert wird.

Kapitel 5

Beschreibung und Konfiguration des Selbinationsverfahrens

„Das Ganze ist mehr als die Summe seiner Teile.“

Aristoteles, griechischer Universalgelehrter (384 v.Chr. - 322 v.Chr.)

Das folgende Kapitel beschreibt das neuartige Selbinationsverfahren dieser Arbeit, ein hybrides Meta-Lernverfahren zur automatischen Selektion oder Kombination geeigneter Prognosemodelle. Abschnitt 5.1 gibt einen Überblick über den Ansatz, welcher aus einem Meta-Modell zur Bestimmung von Eignungswahrscheinlichkeiten für Prognoseverfahrenskandidaten und einer sogenannten Hybriditätsgrenze besteht. Abschnitt 5.2 erläutert den Trainingsprozess des Meta-Modells. Abschnitt 5.3 beschreibt die Verwendung eines trainierten Meta-Modells sowie die Selektion eines oder mehrerer Prognoseverfahren basierend auf bestimmten Eignungswahrscheinlichkeiten und der Hybriditätsgrenze. In Abschnitt 5.4 wird ein sinnvoller Wert für die Hybriditätsgrenze bestimmt und Abschnitt 5.5 beschreibt die Implementierung des Selbinationsverfahrens.

5.1 Übersicht

Wie im letzten Kapitel beschrieben wurde, existieren drei verschiedene Ansätze zur Selektion von M_{sel} Prognoseverfahren aus einer Menge an M Verfahrenskandidaten: (i) ein Selektionsansatz selektiert ein einzelnes Prognoseverfahren ($M_{\text{sel}} = 1$), (ii) ein Pooling-Ansatz selektiert mehrere Verfahren und kombiniert diese ($1 < M_{\text{sel}} < M$), (iii) ein Kombinationsansatz kombiniert alle Verfahrenskandidaten ($M_{\text{sel}} = M$). Selektionsansätze bieten sich an, wenn ein dominantes Verfahren für eine Zeitreihe zu erwarten ist. Pooling-Ansätze eignen sich, wenn für mehrere Verfahren gute Prognosen zu erwarten sind. Kombinationsansätze bieten Potenzial, wenn unklar ist, welche Verfahren für eine Zeitreihe geeignet sind, da die spezifischen Stärken und Schwächen verschiedener Verfahren sich im Durchschnitt ausgleichen und somit eine robuste Prognose erzielt wird. Diese Prognose ist in der Regel schlechter, als die Prognose des besten Prognoseverfahrenskandidaten

für die Zeitreihe aber besser als die Prognose des schlechtesten Verfahrenskandidaten. Somit haben die drei unterschiedlichen Ansätze in Abhängigkeit von den Charakteristika einer vorliegenden Zeitreihe jeweils Vor- und Nachteile. Allerdings weist der aktuelle Stand der Forschung die Limitierung auf, dass im Allgemeinen unabhängig von den spezifischen Charakteristika einer Zeitreihe entschieden wird, welcher der drei Ansätze verwendet werden soll (siehe Abschnitt 4.5). Im Gegensatz dazu wird in diesem Kapitel das sogenannte *Selbinationsverfahren* (Selbination) vorgestellt, welches mit Hilfe eines Meta-Modells Eignungswahrscheinlichkeiten für Prognoseverfahrenskandidaten bestimmt und darauffolgend die geeigneten Prognoseverfahren selektiert. Dadurch stellt das Verfahren einen hybriden Ansatz zwischen Selektion und Kombination dar ($1 \leq M_{\text{sel}} \leq M$). Wenn ein dominantes Verfahren erwartet wird, wird nur dieses selektiert. Wenn für mehrere Verfahren eine gute Eignung erwartet wird oder unklar ist, welche Verfahren sich für eine Zeitreihe eignen, werden mehrere Verfahren selektiert und kombiniert. Dadurch werden die Vorteile aller drei Ansätze in einem hybriden Verfahren vereint.

Abbildung 5.1 gibt einen Überblick über das neuartige Selbinationsverfahren. Es lassen sich eine Trainings- und eine Einsatzphase unterscheiden. In der Trainingsphase wird ein Meta-Lernverfahren verwendet, um ein Meta-Modell basierend auf einer Meta-Trainingsmenge zu trainieren. Die Meta-Trainingsmenge besteht aus einer Matrix charakteristischer Merkmale, die jede in der Trainingsphase genutzte Zeitreihe beschreiben, und einer Matrix mit Labels, welche die Eignungen verschiedener Prognoseverfahrenskandidaten für diese Zeitreihen angeben. Das Ergebnis der Meta-Trainingsphase ist ein trainiertes Meta-Modell zur Bestimmung von Eignungswahrscheinlichkeiten der Prognoseverfahrenskandidaten für neue Zeitreihen, die nicht in der Meta-Trainingsphase genutzt wurden. Die Meta-Trainingsphase muss einmalig ausgeführt werden und das Meta-Modell kann danach ohne weitere Trainingsphasen verwendet werden. Theoretisch besteht zusätzlich die Möglichkeit, das Meta-Modell in einer weiteren Meta-Trainingsphase neu zu trainieren, wenn zahlreiche neue Zeitreihen für das Training zur Verfügung stehen sollten. In der vorliegenden Arbeit wird die Meta-Trainingsphase jedoch nur einmalig durchlaufen.

In der Einsatzphase lässt sich das trainierte Meta-Modell verwenden, um Eignungswahrscheinlichkeiten der Prognoseverfahrenskandidaten für neue Zeitreihen zu bestimmen. Basierend auf diesen Eignungswahrscheinlichkeiten und einer Hybriditätsgrenze werden ein oder mehrere Prognoseverfahren selektiert. Insgesamt müssen die folgenden Schritte in der Einsatzphase durchlaufen werden, um zukünftige Zeitreihenwerte zu prognostizieren: (i) Bestimmung der gleichen charakteristischen Meta-Merkmale und Durchführung der gleichen Transformationen, wie in der Trainingsphase, (ii) Verwendung des trainierten Meta-Modells zur Bestimmung von Eignungswahrscheinlichkeiten für die Prognoseverfahrenskandidaten, (iii) Selektion eines oder mehrerer Prognoseverfahren basierend auf den Eignungswahrscheinlichkeiten und der Hybriditätsgrenze, (iv) Verwendung der selektierten Prognoseverfahren zum Training von Prognosemodellen, (v) Verwendung der trainierten Prognosemodelle zur Bestimmung von Prognosen und möglicherweise (vi) Kombination der Prognosen der verschiedenen Prognosemodelle zu einer finalen Prognose.

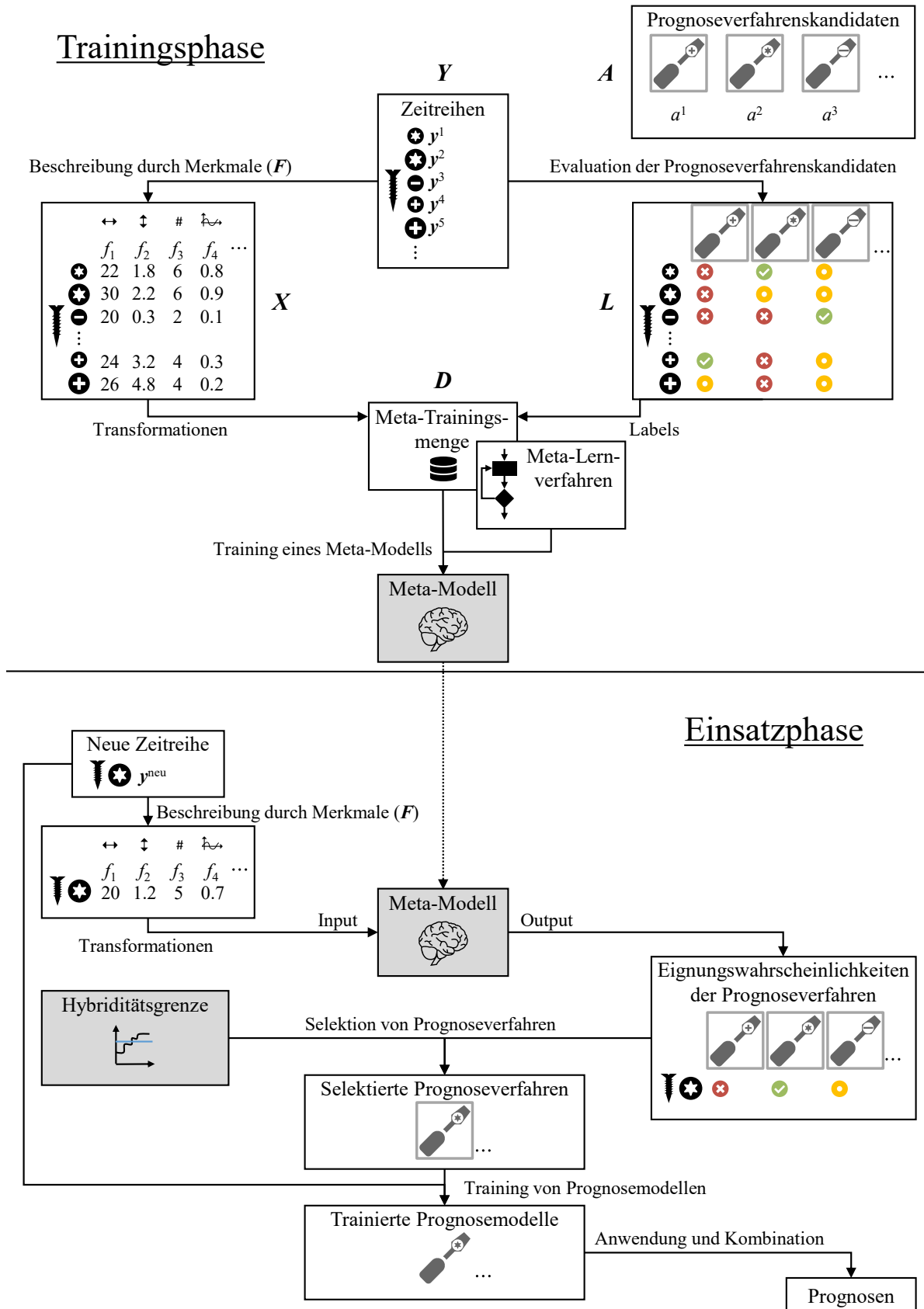


Abbildung 5.1: Training und Einsatz des neuartigen Selbbinationsverfahrens

5.2 Training des Meta-Modells

5.2.1 Übersicht

Abbildung 5.2 zeigt ein Ablaufdiagramm der notwendigen Schritte zum Training eines Meta-Modells. Die erste Komponente, die für die Trainingsphase benötigt wird, ist eine Menge an Zeitreihen \mathbf{Y} . Ähnlich zu einem Prognosemodell, welches anhand einer Trainingsmenge trainiert und anhand einer Testmenge evaluiert wird, wird ein Meta-Modell anhand von Meta-Mengen trainiert und getestet. Allerdings liegt der Zweck eines Meta-Modells nicht in der Bestimmung von Prognosen zukünftiger Zeitreihenwerte, sondern in der Bestimmung von Eignungswahrscheinlichkeiten für Prognoseverfahrenskandidaten. Diese Aufgabe lässt sich nicht als ein Zeitreihenprognoseproblem, sondern als ein Klassifikationsproblem beschreiben. Die Meta-Trainingsmenge besteht aus einer Merkmalsmatrix und einer Label-Matrix. Diese Matrizen werden basierend auf der Menge \mathbf{Y} bestimmt, welche aus Z_{train} Zeitreihen besteht:

$$\mathbf{Y} = \{\mathbf{y}^1, \dots, \mathbf{y}^{Z_{\text{train}}}\}, \text{ mit } \mathbf{y}^z = \{y_1^z, \dots, y_{T_z+H}^z\}. \quad (5.1)$$

Dieser Datensatz wird unterteilt in einen Datensatz

$$\mathbf{Y}_{\text{train}} = \{\mathbf{y}_{\text{train}}^1, \dots, \mathbf{y}_{\text{train}}^{Z_{\text{train}}}\}, \text{ mit } \mathbf{y}_{\text{train}}^z = \{y_1^z, \dots, y_{T_z}^z\}, \quad (5.2)$$

der alle Trainingsmengen der Z_{train} Zeitreihen in \mathbf{Y} enthält, und einen Datensatz

$$\mathbf{Y}_{\text{test}} = \{\mathbf{y}_{\text{test}}^1, \dots, \mathbf{y}_{\text{test}}^{Z_{\text{train}}}\}, \text{ mit } \mathbf{y}_{\text{test}}^z = \{y_{T_z+1}^z, \dots, y_{T_z+H}^z\}, \quad (5.3)$$

der alle Testmengen der Z_{train} Zeitreihen in \mathbf{Y} enthält. $\mathbf{Y}_{\text{train}}$ enthält alle Zeitreihen, die für das Training von Prognosemodellen verwendet werden und \mathbf{Y}_{test} enthält die zukünftigen Werte der Zeitreihen, die zur Evaluation der Prognosegüte eines Prognosemodells verwendet werden. Um eine Meta-Trainingsmenge für ein Meta-Modell zu bilden, wird die Merkmalsmatrix basierend auf $\mathbf{Y}_{\text{train}}$ bestimmt und die Label-Matrix wird anhand von $\mathbf{Y}_{\text{train}}$ und \mathbf{Y}_{test} bestimmt. Zu diesem Zweck müssen eine Menge von Merkmalsextraktionsfunktionen \mathbf{F}' , um die verschiedenen Zeitreihen zu beschreiben, und eine Menge an Prognoseverfahrenskandidaten \mathbf{A} definiert werden. Zudem müssen ein geeignetes Prognosegütekriterium e sowie die minimalen und maximalen Prognosehorizonte h_{\min} , h_{\max} und Ursprünge o_{\min} , o_{\max} definiert werden. Des Weiteren muss ein Lernverfahren zum Training des Meta-Modells gewählt werden. In der vorliegenden Arbeit wird ein Ensemble neuronaler Netze als Meta-Modell trainiert, da dies ein universaler Approximator ist, der dazu in der Lage ist, beliebig komplexe Strukturen zu erlernen, wenn die Datengrundlage ausreichend gut ist und die Hyperparameter geeignet gewählt werden (vgl. Abschnitt 3.4.5). Die zu wählenden Hyperparameter sind die Anzahl der Neuronen der drei verschiedenen Schichten eines Multilayer-Perzeptrons, N_1 , N_2 und N_3 , die Aktivierungsfunktionen der drei Schichten g_1 , g_2 und g_3 sowie die Trainingsspezifikationen und die Anzahl der neuronalen Netze N , die im Ensemble kombiniert werden sollen. In diesem Zusammenhang ist N_1 als die Anzahl der

betrachteten Meta-Merkmale F und N_3 als die Anzahl der Prognoseverfahrenskandidaten M gegeben. Zudem müssen die Prozentsätze der Trainingsbeispiele spezifiziert werden, die zum Training und zur Validierung der neuronalen Netze verwendet werden sollen.

Basierend auf den Eingabewerten wird zum einen die Merkmalsmatrix $\mathbf{X}' \in \mathcal{X}^{F'}$ bestimmt, indem alle Merkmalsextraktionsfunktionen in \mathbf{F}' für alle Trainingsmengen in $\mathbf{Y}_{\text{train}}$ angewandt werden. Da einige Extraktionsfunktionen sich für bestimmte Zeitreihen nicht verwenden lassen, muss die resultierende Matrix bereinigt werden, um eine Menge verwendbarer Extraktionsfunktionen $\mathbf{F} \subseteq \mathbf{F}'$ und eine bereinigte Merkmalsmatrix \mathbf{X}'' zu erhalten. Des Weiteren wird die Matrix transformiert, um besser vom Meta-Modell verarbeitet werden zu können. Die resultierende Merkmalsmatrix \mathbf{X} sowie die Mittelwerte und Standardabweichungen der Merkmale werden gespeichert. Abschnitt 5.2.2 beschreibt die Bestimmung der Merkmalsmatrix. Zudem beschreibt Abschnitt 5.2.5 einen Merkmalsselektionsansatz zum Ausschluss der Merkmale, die keinen positiven Einfluss auf die Selektionsgüte des Meta-Modells haben. Zum anderen wird eine Label-Matrix \mathbf{L} anhand der Menge der Trainingsmengen $\mathbf{Y}_{\text{train}}$, der Menge der Testmengen \mathbf{Y}_{test} , der Menge an Prognoseverfahrenskandidaten \mathbf{A} und dem Gütekriterium e bestimmt (vgl.

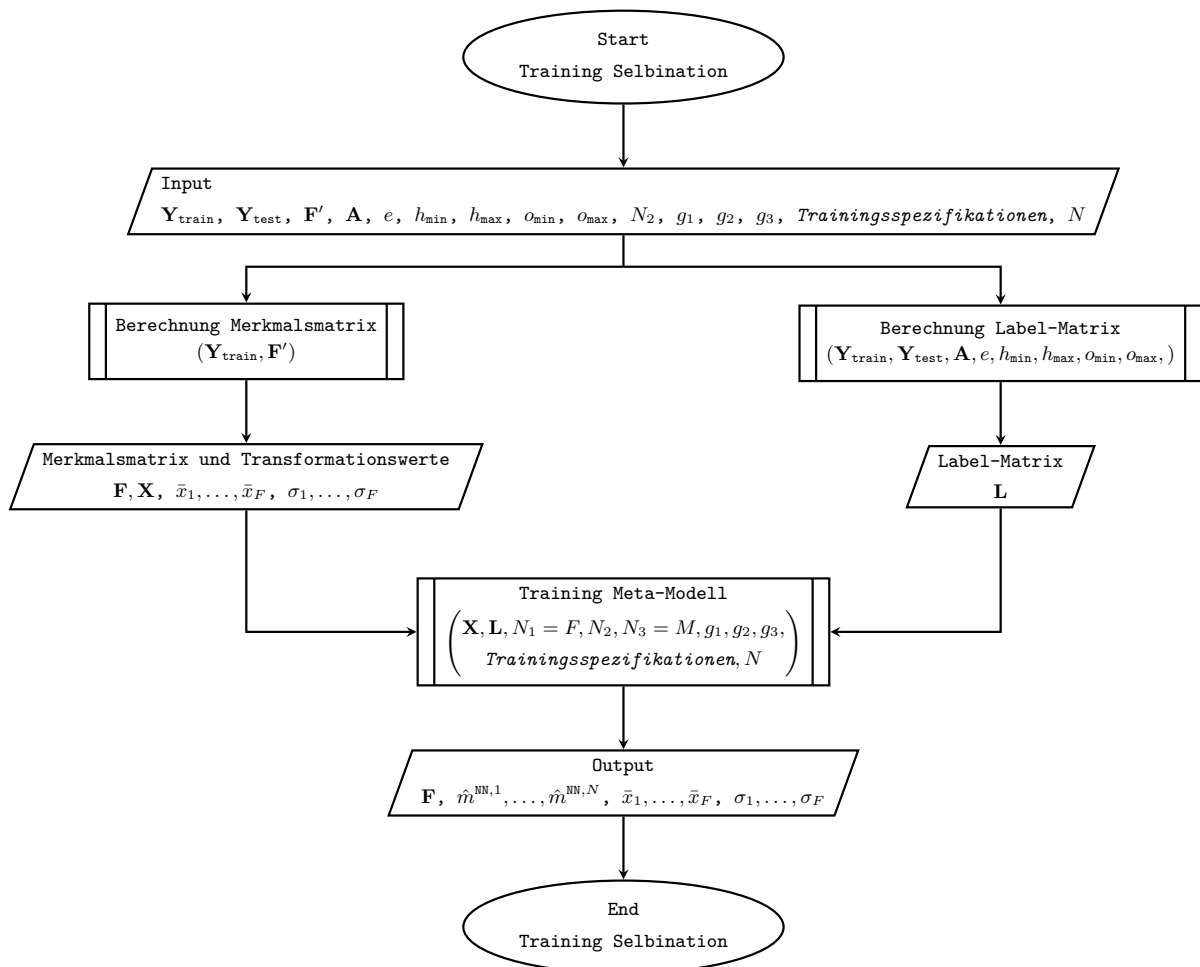


Abbildung 5.2: Ablaufdiagramm der Trainingsphase des Selbinationsverfahrens

Abschnitt 5.2.3). Darauffolgend wird das Meta-Modell, ein Ensemble neuronaler Netze, auf Basis der Merkmalsmatrix \mathbf{X} und der Label-Matrix \mathbf{L} unter Verwendung der Schichtgrößen N_1 , N_2 und N_3 , der Aktivierungsfunktionen g_1 , g_2 und g_3 , der Ensemble-Größe N sowie dem Prozentsatz zur zufälligen Aufteilung der Meta-Trainingsmenge in Meta-Trainings und Meta-Validierungsbeispiele trainiert (vgl. Abschnitt 5.2.4).

Die Ergebnisse der Trainingsphase des Selbinationsverfahrens sind das Meta-Modell, welches aus einem Ensemble von N neuronalen Netzen besteht, sowie die Mittelwerte und Standardabweichungen der Merkmale, welche für die Merkmalstransformationen für neue Zeitreihen in der Einsatzphase benötigt werden, um das gleiche Eingabeformat für das Meta-Modell zu gewährleisten.

5.2.2 Meta-Merkmale

Abbildung 5.3 zeigt ein Ablaufdiagramm zur Berechnung einer Merkmalsmatrix, welche für das Training eines Meta-Modells benötigt wird. Die Berechnung basiert auf der Menge der Trainingsmengen $\mathbf{Y}_{\text{train}}$ und einer Menge von Merkmalsextraktionsfunktionen \mathbf{F}' .

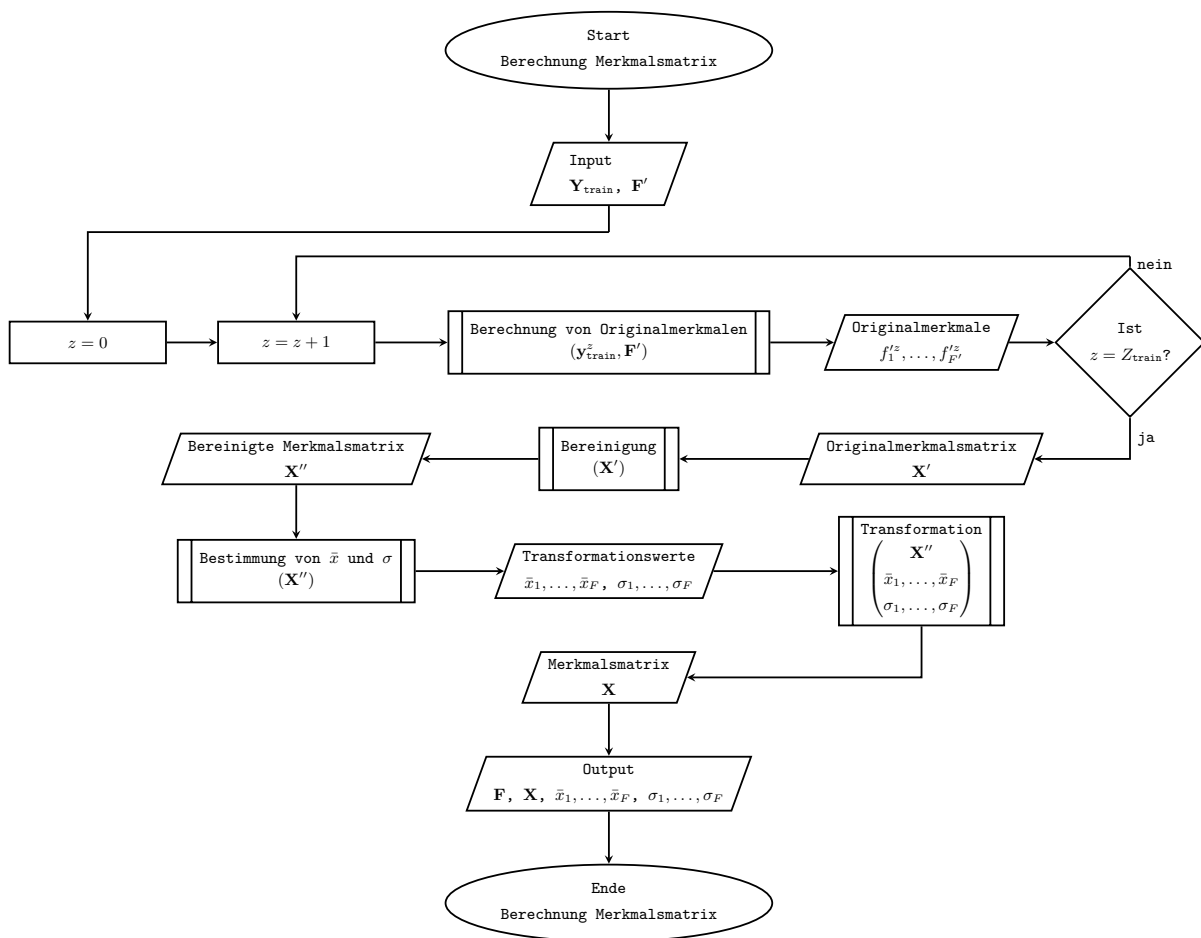


Abbildung 5.3: Ablaufdiagramm zur Berechnung einer Merkmalsmatrix

Zunächst werden alle Merkmalsextraktionsfunktionen in \mathbf{F}' , die später in diesem Abschnitt beschrieben werden, für alle Z_{train} Zeitreihen in der Menge der Trainingsmengen $\mathbf{Y}_{\text{train}}$ angewandt. Hierbei werden nur die Trainingsmengen und nicht die Testmengen der verschiedenen Zeitreihen verwendet, um die Merkmale zu berechnen, da in der späteren Einsatzphase des Selbinationenverfahrens die Testmengen der Zeitreihen nicht bekannt sind. Die resultierende Matrix mit Originalmerkmalen \mathbf{X}' muss bereinigt werden, da möglicherweise einige Merkmale für bestimmte Zeitreihen nicht berechenbar sind (NaN-Werte). Dies kann beispielsweise vorkommen, wenn mathematische Operationen aufgrund zu kurzer Zeitreihen nicht durchführbar sind. Um dieses Problem zu lösen, werden alle NaN-Werte durch den Mittelwert des jeweiligen Merkmals über alle betrachteten Zeitreihen ersetzt, falls höchstens 30% der Werte dieses Merkmals NaN-Werte sind. Falls mehr als 30% der Werte für ein Merkmal nicht berechenbar sind, wird die Spalte aus der Merkmalsmatrix gelöscht und die zugehörige Merkmalsextraktionsfunktion wird aus \mathbf{F}' entfernt. Für die bereinigte Merkmalsmatrix

$$\mathbf{X}'' = \begin{pmatrix} x_1''^1 & \cdots & x_F''^1 \\ \vdots & \vdots & \vdots \\ x_1''^{Z_{\text{train}}} & \cdots & x_F''^{Z_{\text{train}}} \end{pmatrix}, \quad (5.4)$$

werden die Mittelwerte \bar{x}_j und Standardabweichungen σ_j der Spalten berechnet. Diese Werte werden genutzt, um jede Spalte der Matrix \mathbf{X}'' zu standardisieren, indem von jedem Wert der Spaltenmittelwert subtrahiert wird und durch die Spaltenstandardabweichung dividiert wird, um die bereinigte und standardisierte Merkmalsmatrix

$$\mathbf{X} = \begin{pmatrix} x_1^1 & \cdots & x_F^1 \\ \vdots & \vdots & \vdots \\ x_1^{Z_{\text{train}}} & \cdots & x_F^{Z_{\text{train}}} \end{pmatrix} = \begin{pmatrix} \frac{x_1''^1 - \bar{x}_1}{\sigma_1} & \cdots & \frac{x_F''^1 - \bar{x}_F}{\sigma_F} \\ \vdots & \vdots & \vdots \\ \frac{x_1''^{Z_{\text{train}}} - \bar{x}_1}{\sigma_1} & \cdots & \frac{x_F''^{Z_{\text{train}}} - \bar{x}_F}{\sigma_F} \end{pmatrix}. \quad (5.5)$$

zu erhalten. Diese Transformation gewährleistet, dass alle Spalten der Merkmalsmatrix den Mittelwert 0 und die Standardabweichung 1 haben. Dies ist ein gängiges Eingabeformat für neuronale Netze (vgl. Rehkugler und Zimmermann, 1994; Neuneier und Zimmermann, 2012). Die Ergebnisse der Berechnung sind eine bereinigte und standardisierte *Merkmalsmatrix* \mathbf{X} , welche F Merkmale für alle Z_{train} Zeitreihen enthält, sowie die Mittelwerte und Standardabweichungen der Spalten der bereinigten Originalmerkmalsmatrix \mathbf{X}'' , die für die Transformation der Merkmale für eine neue Zeitreihe in der Einsatzphase des Selbinationenverfahrens genutzt werden können.

Tabelle 5.1 zeigt alle 89 Merkmale, die in den empirischen Studien dieser Arbeit verwendet werden. Die markierten Merkmale sind diejenigen, die nach der in Abschnitt 5.2.5 beschriebenen Merkmalsselektion übrig bleiben. Die Merkmale lassen sich den Typen der direkten Merkmale (D), modellbasierten Merkmale (M) und Landmarkers (L) zuordnen. Die komplette Menge an Merkmalsextraktionsfunktionen besteht aus 12 Funktionen, die in Wang et al. (2009) verwendet wurden, 14 Funktionen der Rekurrenzquantifizierungsanalyse (vgl. Marwan et al., 2007; Scholz-Reiter et al., 2014), 12 generellen Statistiken, 35 Werten aus statistischen Tests, 4 Landmarkers,

die die Validierungsfehler der Prognoseverfahrenskandidaten angeben sowie zwei neuartigen Mengen von je 6 Landmarkers: den Verhältnissen und den Differenzen der Validierungsfehler der Prognoseverfahrenskandidaten.

Die erste verwendete Menge an Merkmalsextraktionsfunktionen dieser Arbeit wurde von in einem Meta-Lernansatz von Wang et al. (2009) vorgestellt und unter anderem auch in Scholz-Reiter et al. (2014) und Kück et al. (2016b) verwendet. Die Originalmenge bestand aus 13 gängigen Metriken zur Quantifizierung der Charakteristika von Zeitreihen. Neun dieser Charakteristika werden anhand der Originalzeitreihen gemessen: Trend, Saisonalität, Autokorrelation (gemessen als der p-Wert des Box-Pierce-Autokorrelationstests), Nichtlinearität (gemessen als der p-Wert des Teräsvirta-NN-Nichtlinearitätstests), Schiefe, Wölbung, Selbstähnlichkeit (gemessen als der Hurst-Exponent), Chaos (gemessen als der Lyapunov-Exponent) und Frequenz. Außerdem wurden vier dieser Charakteristika zusätzlich anhand der um Trend und Saisonalität bereinigten Zeitreihe (cs) bestimmt: Autokorrelation (cs), Nichtlinearität (cs), Schiefe (cs) und Wölbung (cs). Abbildung 5.4 zeigt eine beispielhafte Zerlegung einer Originalzeitreihe in eine Trend-, eine Saison- und eine Restkomponente (vgl. Hyndman und Athanasopoulos, 2018). Für weitere Erläuterungen zu den Charakteristika sei auf Wang et al. (2006) und Wang et al. (2009) verwiesen. Da alle betrachteten Zeitreihen in dieser Arbeit monatlich gemessen wurden, wird das Merkmal der Frequenz nicht verwendet und es verbleibt eine Menge an 12 Merkmalsextraktionsfunktionen, welche in R (vgl. R Core Team, 2018) anhand der Beschreibungen in Wang et al. (2006) bestimmt werden.

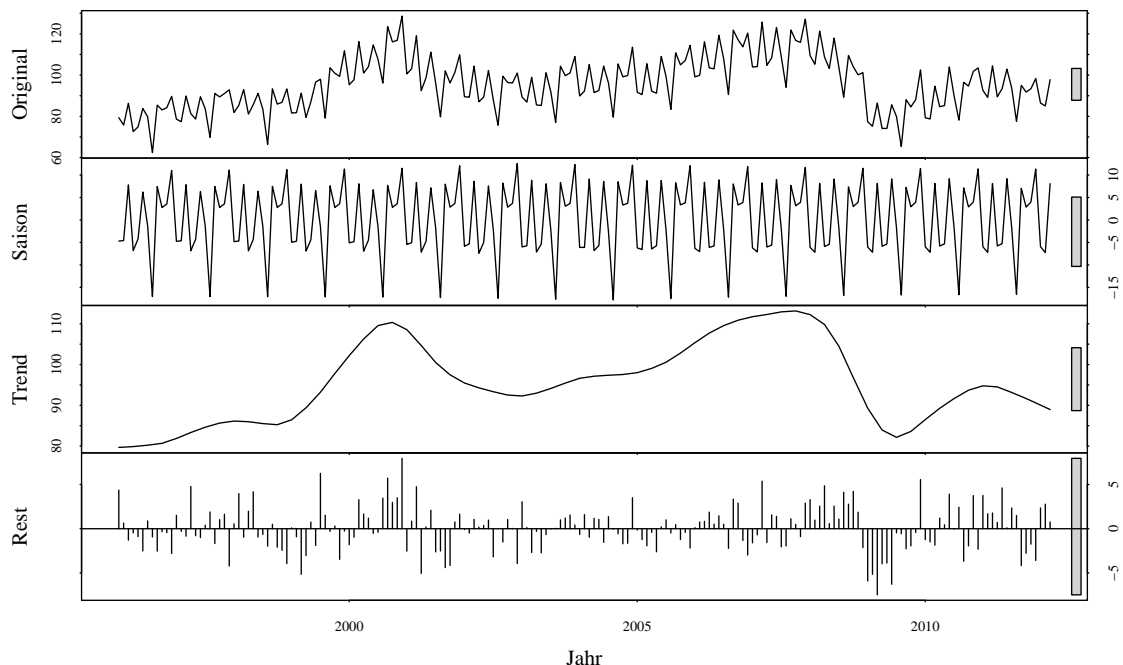


Abbildung 5.4: STL-Dekomposition in Anlehnung an Hyndman und Athanasopoulos (2018)

Tabelle 5.1: Betrachtete Meta-Merkmale der Typen direkter Merkmale (D), modellbasierter Merkmale (M) und Landmarkers (L), um ein Prognosemodellselektionsproblem zu beschreiben (verbleibende Merkmale nach der Merkmalsselektion sind mit einem * markiert)

Nr.	Merkmal	Typ	Nr.	Merkmal	Typ
1	Trend	D	46*	ADF-Stationaritätstest (c)	M
2*	Saisonalität	D	47*	ADF-Stationaritätstest (ct)	M
3	Box-Pierce-Autokorrelationstest	M	48	Chi-Quadrat-Test für 12-monatl. Saisonalität	M
4	Teräsvirta-NN-Nichtlinearitätstest	M	49	Chi-Quadrat-Test für 6-monatl. Saisonalität	M
5	Schiefe	D	50	Chi-Quadrat-Test für 3-monatl. Saisonalität	M
6	Wölbung	D	51*	Kruskal-Wallis-Test für 12-monatl. Saisonalität	M
7	Hurst-Exponent	M	52	Kruskal-Wallis-Test für 6-monatl. Saisonalität	M
8	Lyapunov-Exponent	M	53	Kruskal-Wallis-Test für 3-monatl. Saisonalität	M
9	Box-Pierce-Autokorrelationstest (cs)	M	54	F-Test für 12-monatl. Saisonalität	M
10	Teräsvirta-NN-Nichtlinearitätstest (cs)	M	55	F-Test für 6-monatl. Saisonalität	M
11	Schiefe (cs)	D	56	F-Test für 3-monatl. Saisonalität	M
12	Wölbung (cs)	D	57	ACF-Test für 12-monatl. Saisonalität	M
13	Rekurrenzrate	M	58	ACF-Test für 6-monatl. Saisonalität	M
14	Determinismus	M	59	ACF-Test für 3-monatl. Saisonalität	M
15	Mittlere Diagonallinienlänge	M	60	Anzahl nicht-signifikanter ACF-Lags	M
16*	Länge der längsten Diagonallinie	M	61	Anzahl signifikanter ACF-Lags	M
17*	Entropie der Diagonallänge	M	62*	Anzahl hoch-signifikanter ACF-Lags	M
18*	Laminarität	M	63	PACF-Test für 12-monatl. Saisonalität	M
19	Mittlere Vertikallinienlänge	M	64	PACF-Test für 6-monatl. Saisonalität	M
20	Länge der längsten Vertikallinie	M	65	PACF-Test für 3-monatl. Saisonalität	M
21*	Rekurrenzzeit erster Art	M	66*	Anzahl nicht-signifikanter PACF-Lags	M
22	Rekurrenzzeit zweiter Art	M	67	Anzahl signifikanter PACF-Lags	M
23*	Rekurrenzzeit-Entropie	M	68	Anzahl hoch-signifikanter PACF-Lags	M
24*	Transitivität	M	69*	Jarque-Bera-Normalitätstest (asymptotisch)	M
25	Einbettungsdimension	M	70*	Jarque-Bera-Normalitätstest (Monte-Carlo)	M
26	Zeitverzögerung	M	71	Runs-Test für Nichtzufälligkeit (Mittelwert)	M
27*	Anzahl der Zeitreihenwerte	D	72	Runs-Test für Nichtzufälligkeit (Median)	M
28*	Minimum	D	73	Runs-Test für Nichtzufälligkeit (up-down)	M
29	Maximum	D	74	Validierungsfehler von llknn	L
30	Mittelwert	D	75	Validierungsfehler von ets	L
31	Median	D	76	Validierungsfehler von nnetar	L
32*	unteres Quartil	D	77	Validierungsfehler von srw	L
33	oberes Quartil	D	78*	Verhältnis llknn / ets	L
34	Spannweite	D	79	Verhältnis llknn / nnetar	L
35	Standardabweichung	D	80*	Verhältnis llknn / srw	L
36	Varianz	D	81	Verhältnis ets / nnetar	L
37*	Variationskoeffizient	D	82*	Verhältnis ets / srw	L
38	Modus	D	83*	Verhältnis nnetar / srw	L
39*	Mann-Kendall-Test für monotonen Trend	M	84*	Differenz llknn - ets	L
40*	Spearman-Trend-Test für Rangkorrelation	M	85	Differenz llknn - nnetar	L
41	Pearson-Trend-Test für lineare Korrelation	M	86	Differenz llknn - srw	L
42	Noether-Test für zyklischen Trend	M	87	Differenz ets - nnetar	L
43	Cox-Stuart-Lage-Trend-Test	M	88	Differenz ets - srw	L
44	Cox-Stuart-Streuungstest	M	89	Differenz nnetar - srw	L
45	ADF-Stationaritätstest (0)	M			

Marwan et al. (2007) entwickelten die Zeitreihenkomplexitätsmaße der *Rekurrenzquantifizierungsanalyse* (RQA, engl. recurrence quantification analysis), welche als Meta-Merkmale von Scholz-Reiter et al. (2014) und Kück et al. (2016b) verwendet wurden. Diese Maße basieren auf der Theorie dynamischer Systeme und insbesondere auf einer Phasenraumrekonstruktion (vgl. Abschnitt 3.4.4 und Takens, 1981; Kantz und Schreiber, 2004). Zur Rekonstruktion der dynamischen Eigenschaften eines Systems wird die Methode der Delay-Koordinaten-Einbettung verwendet (vgl. Takens, 1981), wodurch die Zeitreihe in einen Phasenraum abgebildet wird. Die dynamischen Eigenschaften eines gesamten dynamischen Systems lassen sich auf diese Weise aus der Zeitreihe rekonstruieren, wenn die Einbettungsdimension r und die Zeitverzögerung d sinnvoll gewählt werden. Marwan et al. (2007) empfehlen die Ermittlung von r durch den Algorithmus der falschen nächsten Nachbarn (FNN, vgl. Kennel et al., 1992) und d als die erste Minimalstelle der mittleren gemeinsamen Information (AMI, vgl. Fraser und Swinney, 1986). Allerdings lassen sich einige RQA-Maße für bestimmte Zeitreihenentwicklungen nicht bestimmen, da diese ursprünglich zur Charakterisierung sehr langer Zeitreihen entwickelt wurden. Somit sind die Empfehlungen für r und d nicht umsetzbar. Um vergleichbare Charakteristika zu ermitteln, werden in der vorliegenden Arbeit die Merkmale 13-24 in Tabelle 5.1 für $r = 1$ und $d = 1$ mit Hilfe der **Cross Recurrence Plot Toolbox** für **Matlab** (vgl. Marwan, 2019) ermittelt. Die entstehende Menge aus 14 RQA-Maßen wird durch die bestimmten Werte für r durch den FNN-Algorithmus und d als erste Minimalstelle der AMI komplettiert.

Die dritte Menge an Merkmalsextraktionsfunktionen besteht aus 12 gängigen statistischen Maßzahlen, die in *iqast* (vgl. RSG Software GmbH, 2016) berechnet werden. Dies sind die Merkmale 27-38 in Tabelle 5.1. Einige der Maße sind in der Box-Plot-Darstellung in Abbildung 5.5 dargestellt (vgl. Sachs und Hedderich, 2006), für weitere Informationen zu den klassischen statistischen Maßen sei auf die Standardliteratur verwiesen (vgl. Sachs und Hedderich, 2006; Everitt und Skrondal, 2010; Fahrmeir et al., 2016).

Zur Bestimmung der Merkmale 39-73 in Tabelle 5.1 werden zahlreiche statistische Tests für Trend, Streuung, Stationarität, Saisonalität, Normalität und Nichtzufälligkeit durchgeführt. Als Merkmale zur Indikation, ob eine Zeitreihe einen Trend besitzt, werden die p-Werte eines Mann-Kendall-Tests für monotonen Trend (vgl. Mann, 1945; Kendall, 1975), eines Spearman-Trend-Tests für Rangkorrelation (vgl. Spearman, 1904; Glasser und Winter, 1961), eines Pearson-Trend-Tests für lineare Korrelation (vgl. Pearson, 1920), eines Noether-Tests für zyklischen Trend (vgl. Noether, 1956) und eines Cox-Stuart-Lage-Trend-Tests (vgl. Cox und Stuart, 1955) bestimmt sowie die p-Werte eines Cox-Stuart-Streuungstests (vgl. Cox und Stuart, 1955) und die p-Werte von Augmented-Dickey-Fuller-Tests (ADF) (vgl. Dickey und Fuller, 1979; Cheung und Lai, 1995) für Stationarität um Null (0), um eine Konstante (c) und um eine Trendlinie (ct). Zur Messung von Saisonalität werden die p-Werte von Chi-Quadrat-Tests für Homogenität (vgl. Pearson, 1900; Fisher, 1922, 1924), Kruskal-Wallis-Tests (vgl. Kruskal und Wallis, 1952) und F-Tests (vgl. Fisher, 1925; Snedecor, 1934) für 12-, 6- und 3-monatliche Saisonalität bestimmt. Des Weiteren

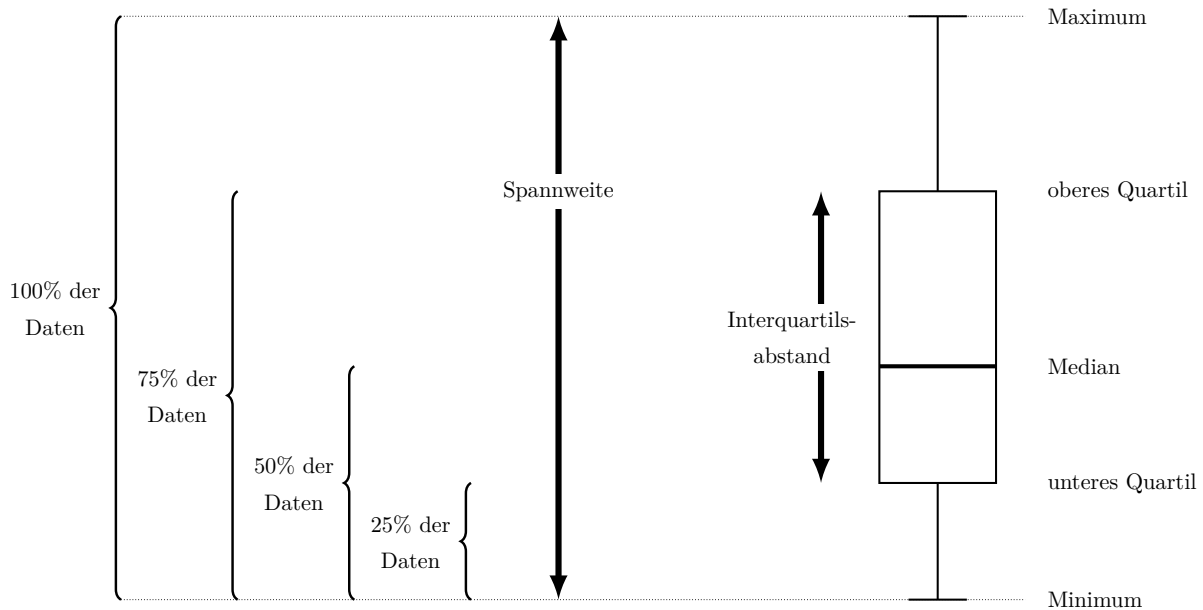


Abbildung 5.5: Statistische Maßzahlen in der Box-Plot-Darstellung (angelehnt an Sachs und Hedderich (2006))

werden Autokorrelationsfunktionen (ACF) und partielle Autokorrelationsfunktionen (PACF) ermittelt (vgl. Box et al., 2015) und die Anzahlen der nicht-signifikanten, signifikanten und hoch-signifikanten Zeitverschiebungen (engl. lags) werden bestimmt sowie die p-Werte der ACF- und der PACF-Tests für 12-, 6- und 3-monatliche Saisonalität. Außerdem werden die p-Werte zweier verschiedener Versionen des Jarque-Bera-Normalitätstests ermittelt, einer asymptotischen Version mit Verwendung einer Chi-Quadrat-Verteilung (vgl. Jarque und Bera, 1980, 1987) und einer Version mit Verwendung einer Monte-Carlo-Simulation (vgl. Bera und Jarque, 1981; Deb und Sefton, 1996). Zusätzlich zu den Tests für Trend, Streuung, Stationarität, Saisonalität und Normalität, werden Tests für Nichtzufälligkeit einer Zeitreihenentwicklung durchgeführt. Zu diesem Zweck werden die p-Werte von drei verschiedenen Wald-Wolfowitz-Runs-Tests bestimmt (vgl. Wald und Wolfowitz, 1940). Zur Transformation einer Zeitreihe in Läufe (engl. runs), werden drei verschiedene Ansätze verwendet: Im ersten Ansatz werden alle Werte, die mindestens dem Mittelwert der Zeitreihe entsprechen, mit einem + kodiert und alle niedrigeren Werte mit einem -. Daraufgehend wird die Anzahl der Wechsel zwischen Läufen mit + und - bestimmt. Im zweiten Ansatz wird anstelle des Mittelwerts der Median als Grenze genutzt. Im dritten Ansatz werden Werte, die mindestens so groß sind, wie der vorherige Zeitreihenwert mit + kodiert und niedrigere Werte mit -. Die p-Werte der Wald-Wolfowitz-Runs-Tests zeigen die Wahrscheinlichkeit, so wenige oder weniger Läufe zu beobachten, als tatsächlich beobachtet wurden und somit zeigen sie, ob die Daten zufällig verteilt sind. Alle Merkmale der vierten Merkmalsmenge werden in iqast berechnet (vgl. RSG Software GmbH, 2016).

Landmarkers wurden bislang nur in wenigen Prognosemodellselektionsansätzen verwendet (vgl. Abschnitt 4.3). Wie modellbasierte Merkmale charakterisieren Landmarkers ein Selektionsproblem indirekt, nachdem ein Modell anhand einer gegebenen Zeitreihe gebildet wurde (vgl. Brazdil et al., 2008; Pfahringer et al., 2000). Im Gegensatz zu modellbasierten Merkmalen beschreiben Landmarkers jedoch keine Modellcharakteristika, sondern Gütemaße der gebildeten Modelle. In der vorliegenden Arbeit werden die Prognoseverfahrenskandidaten genutzt, um Prognosemodelle anhand der Trainingsmengen der Zeitreihen zu trainieren. Darauf folgend werden die trainierten Prognosemodelle verwendet, um die Werte von Validierungsmengen vorherzusagen und die Validierungsfehler werden als Merkmale genutzt, um das Selektionsproblem zu beschreiben. Zu diesem Zweck muss zunächst eine gegebene Trainingsmenge einer Zeitreihe in eine kleinere Trainingsmenge und eine Validierungsmenge unterteilt werden, sodass die Validierungsmenge aus den letzten H bekannten Werten der Zeitreihe besteht. In diesem Zusammenhang werden die letzten $H = 18$ Werte jeder Zeitreihe in $\mathbf{Y}_{\text{train}}$ als Validierungsmengen genutzt. Dies ist die gleiche Länge des Prognosehorizontes, wie in den Testmengen der späteren empirischen Studie in Kapitel 6. In dieser Arbeit werden drei verschiedene Mengen von Landmarkers verwendet: Validierungsfehler sowie Verhältnisse und Differenzen der Validierungsfehler. Validierungsfehler wurden bereits in einer Vorstudie zu dieser Arbeit als Meta-Merkmale verwendet und erwiesen sich als sinnvolle Maße zur Beschreibung der Selektionsprobleme (vgl. Kück et al., 2016b). In diesem Zusammenhang werden die rollierenden symmetrischen mittleren absoluten prozentualen Fehler $\text{RO-sMAPE}_{T_z-18, T_z-12}^{1,12}(\mathbf{y}, \hat{\mathbf{y}})$ aller Prognosen der Horizonte 1 bis 12 von den Ursprüngen $T_z - 18$ bis $T_z - 12$ der Prognoseverfahrenskandidaten verwendet. Diese Konfiguration repräsentiert die Prognosegüte eines Modells über eine große Reihe verschiedener Horizonte und Ursprünge. Für weitere Informationen zur Berechnung der Prognosefehler siehe Abschnitt 3.2.2.

Neben den Validierungsfehlern werden zwei weitere Mengen von Landmarkers verwendet, die im Rahmen dieser Arbeit neu eingeführt werden. Zur Berechnung der Verhältnisse und der Differenzen der Validierungsfehler wird jedes Paar von Validierungsfehlern zweier Prognoseverfahrenskandidaten dividiert und subtrahiert. Hierdurch ergeben sich Merkmale, die beschreiben, ob ein Verfahren auf der Validierungsmenge eine höhere oder eine niedrigere Prognosegüte aufweist als ein anderes Verfahren. Alle Evaluationen hinsichtlich Prognosefehlern, und somit auch die Bestimmung der Landmarkers, werden für diese Arbeit in `Matlab` durchgeführt (vgl. The MathWorks Inc., 2017b). Es sei angemerkt, dass in dieser Arbeit verschiedene Konfigurationen des Selbinationsverfahrens verglichen werden (vgl. Abschnitt 6.4). Je nachdem, wie viele Prognoseverfahrenskandidaten die Menge \mathbf{A} enthält, vergrößert oder verkleinert sich die Menge an Landmarkers und somit die Menge an Meta-Merkmalen. Tabelle 5.1 zeigt die Merkmalsextraktionsfunktionen für eine Konfiguration des Selbinationsverfahrens mit den vier Prognoseverfahrenskandidaten `llkn`, `ets`, `nnetar` und `srw` (siehe Abschnitt 3.4). Dies ist die Konfiguration, die die besten Prognosen erzielt und in den Vergleichsstudien der beiden folgenden Kapitel verwendet wird.

5.2.3 Meta-Labels

Abbildung 5.6 stellt die Schritte zur Bestimmung einer Label-Matrix in einem Ablaufdiagramm dar. Für die Berechnung werden die Menge der Trainingsmengen $\mathbf{Y}_{\text{train}}$ und die Menge der Testmengen \mathbf{Y}_{test} benötigt. Zudem sind eine Menge an Prognoseverfahrenskandidaten \mathbf{A} , eine Prognosefehlerfunktion e sowie minimale und maximale Prognosehorizonte h_{\min} , h_{\max} und Ursprünge o_{\min} , o_{\max} notwendig. Für die Berechnungen in dieser Arbeit werden die folgenden Werte gewählt: $h_{\min} = 1$, $h_{\max} = 12$, $o_{\min} = T_z$, $o_{\max} = T_z + 6$, wodurch eine robuste Evaluation der Prognosefehler über viele verschiedene Horizonte von unterschiedlichen Ursprüngen entsteht.

Für jede Zeitreihe \mathbf{y}^z in \mathbf{Y} wird jeder der Prognoseverfahrenskandidaten a^j in \mathbf{A} verwendet, um je ein Prognosemodell \hat{m}^j basierend auf der Trainingsmenge $\mathbf{y}_{\text{train}}^z$ der Zeitreihe zu trainieren. Das trainierte Modell \hat{m}^j wird verwendet, um Prognosen der Horizonte $h_{\min}, \dots, h_{\max}$ von den Ursprüngen $o_{\min}, \dots, o_{\max}$ zu berechnen. Darauffolgend wird der mittlere Prognosefehler der Prognosen durch Anwendung der Fehlerfunktion e auf die Prognosen $m^j(\mathbf{y}_{\text{train}}^z) = \hat{\mathbf{y}}_j^z$ und die Testmenge $\mathbf{y}_{\text{test}}^z$ bestimmt. Zu diesem Zweck wird der rollierende Fehler $\text{RO-sMAPE}_{T_z, T_z+6}^{1,12}$ verwendet, welcher den Mittelwert aus 84 Prognosen pro Modell und Zeitreihe für die Horizonte 1 bis 12 von 7 verschiedenen Ursprüngen bildet. Für weitere Informationen zur Berechnung der Prognosefehler siehe Abschnitt 3.2.2. Um eine Matrix zu erhalten, die die relativen Prognosegüten der M Prognoseverfahrenskandidaten für die Z_{train} Zeitreihen in \mathbf{Y} enthält, wird die Softmax-Funktion auf die additiv inversen Werte der Zeilen der Fehlermatrix \mathbf{E} angewandt. Das Ergebnis des Prozesses ist eine *Label-Matrix*

$$\mathbf{L} = \begin{pmatrix} \ell_1^1 & \dots & \ell_M^1 \\ \vdots & \vdots & \vdots \\ \ell_1^{Z_{\text{train}}} & \dots & \ell_M^{Z_{\text{train}}} \end{pmatrix} \quad (5.6)$$

$$\text{mit } \ell_j^z \in [0, 1] \text{ und } \sum_{j=1}^M \ell_j^z = 1, \forall z \in \{1, \dots, Z_{\text{train}}\}, j \in \{1, \dots, M\}.$$

In der vorliegenden Arbeit werden verschiedene Konfigurationen des Selbinstationsverfahrens verglichen. Die Menge \mathbf{A} kann dabei zwei oder mehrere der folgenden acht Prognoseverfahrenskandidaten enthalten: `rw`, `srw`, `auto.arima`, `ets`, `nnetar`, `nnfor`, `lcknn` und `llknn`. Diese Prognoseverfahren lassen sich automatisch verwenden, um ein Modell aus einer vorgegebenen Modellklasse zu trainieren (vgl. Abschnitt 3.4). Die ersten fünf der acht Prognoseverfahren sind in das `forecast`-Package (vgl. Hyndman et al., 2018b) und das sechste Verfahren ist in das `nnfor`-Package (vgl. Kourentzes, 2017) in R implementiert. Die Prognosen der nicht-saisonalen (`rw`) und saisonalen (`srw`) Random-Walk-Modelle werden durch die Funktionen `rwf` und `snaive` ermittelt. Beide Modelle benötigen keine Parameteroptimierung. `auto.arima` ist ein ARIMA-Verfahren, welches die Ordnungen der ersten und der saisonalen Differenzbildungen durch Verwendung von KPSS-Tests und saisonalen Unit-Root-Tests ermittelt und darauffolgend ein ARIMA-Modell durch Minimierung des AICc selektiert (vgl. Abschnitt 3.4.2 und Hyndman und Khandakar, 2008; Box

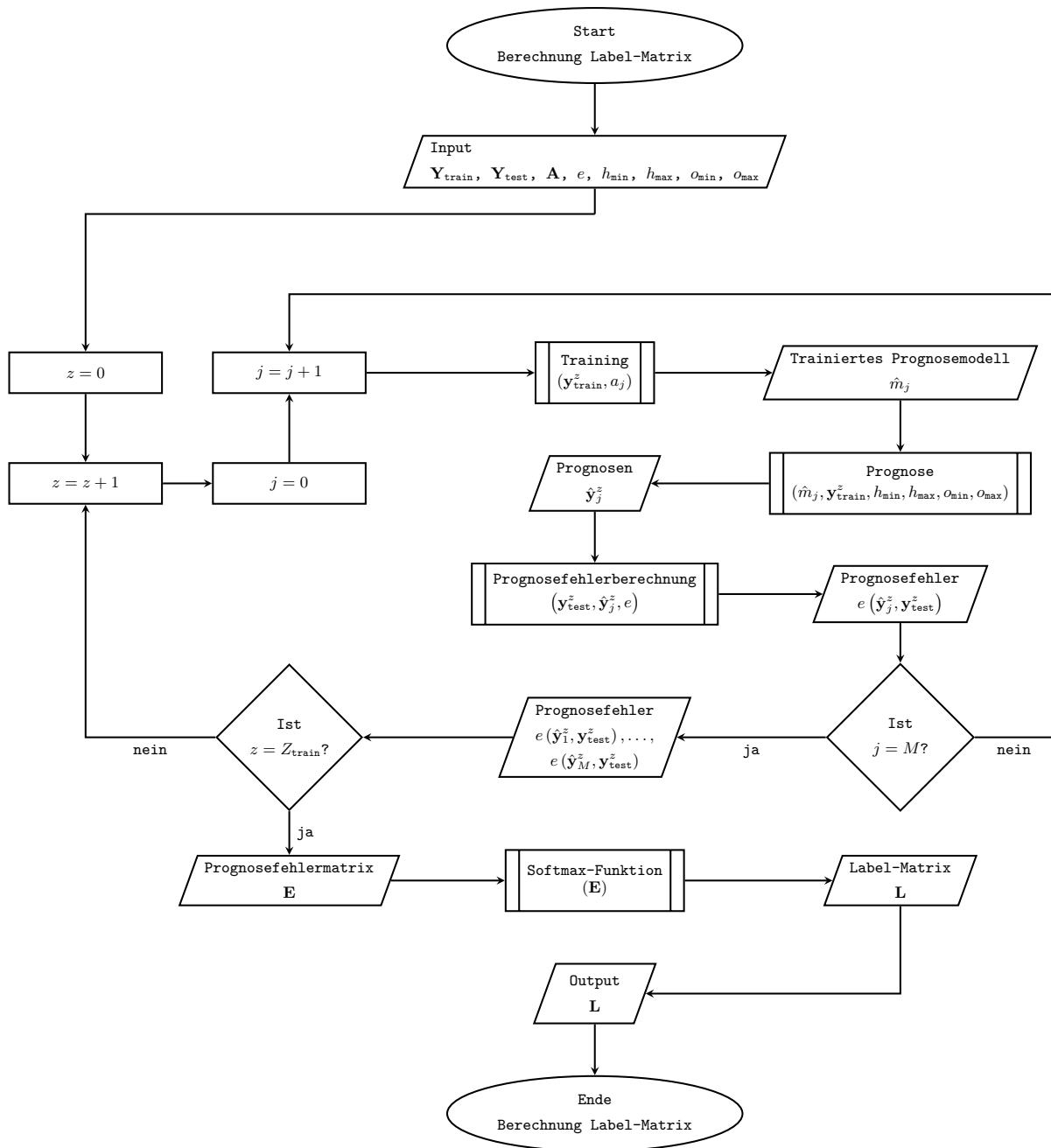


Abbildung 5.6: Ablaufdiagramm zur Berechnung einer Label-Matrix

et al., 2015; Hyndman und Athanasopoulos, 2018). ets ist ein exponentielles Glättungsverfahren, welches die Zustandsraummodellierung nutzt und ein Modell durch Minimierung des AICc selektiert (vgl. Abschnitt 3.4.3 und Hyndman und Khandakar, 2008; Hyndman et al., 2008b). Als Lernverfahren für neuronale Netze werden nnetar und nnfor verwendet (vgl. Abschnitt 3.4.5). Beide trainieren Ensembles von 20 Multilayer-Perzeptronen. nnetar nutzt das BFGS-Verfahren und einen Gewichts-dämpfungs-faktor von 0,2 zum Training der Netzparameter. nnfor nutzt den Algorithmus der resilienten Fehlerrückführung mit Gewichts-rücksetzung. Neben den sechs in R

implementierten Prognoseverfahren kann die Menge \mathbf{A} zwei im Rahmen dieser Arbeit entwickelte lokale k -nächste-Nachbarn-Verfahren enthalten (vgl. Abschnitt 3.4.4 und Kück und Freitag, 2021). $lknn$ ist ein lokal konstantes knn -Modell, welches in der Konfiguration eines arithmetischen Mittelwertes der nächsten Nachbarn mit einer dynamischen Modellselektionsstrategie und den Parametern $d = 1$, $r_{\%} = 1$ und $k_{\%} = 0,25$ verwendet wird. $llknn$ ist eine lokal lineare knn -Ridge-Regression mit einer dynamischen Modellselektion und den Parametern $d = 1$, $r_{\%} = 1$ und $k_{\%} = 1$ sowie einem Regularisierungsparameter $\lambda = 100$.

Für die empirischen Studien der nächsten beiden Kapitel enthält die Menge \mathbf{A} die folgenden vier Verfahrenskandidaten: $llknn$, ets , $nnetar$ und srw , da das Selbbinationsverfahren in dieser Konfiguration die besten Ergebnisse erzielt (vgl. Abschnitt 6.4). Es sei zudem angemerkt, dass alle Prognosen dieser Arbeit auf positive ganze Zahlen gerundet werden, da reale Zeitreihen der Kundennachfrage keine negativen Werte annehmen können.

5.2.4 Meta-Modell

Nach der Berechnung einer Merkmalsmatrix \mathbf{X} und einer Label-Matrix \mathbf{L} für eine gegebene Menge an Zeitreihen \mathbf{Y} wird eine *Meta-Trainingsmenge* (engl. meta-training set)

$$\mathbf{D}_{\text{train}} = \left\{ \left(\mathbf{x}^1, \ell^1 \right), \left(\mathbf{x}^2, \ell^2 \right), \dots, \left(\mathbf{x}^{D_{\text{train}}}, \ell^{D_{\text{train}}} \right) \right\} \in \mathcal{D}^{D_{\text{train}}} \quad (5.7)$$

aus Meta-Trainingsbeispielen gebildet. Basierend darauf lässt sich ein Meta-Modell trainieren, um die Zusammenhänge zwischen den Zeitreihencharakteristika und den Eignungen der Prognoseverfahrenskandidaten für die jeweiligen Zeitreihen zu lernen. Zu diesem Zweck wird zunächst die Meta-Trainingsmenge in eine kleinere Meta-Trainingsmenge und eine Meta-Validierungsmenge unterteilt und darauffolgend wird das Training des Meta-Modells als ein Klassifikationsproblem betrachtet (vgl. Abschnitt 3.1.3). Aus dem Raum der Klassifikationsmodelle $\mathcal{M}_{\text{Klas}}$ wird als Meta-Modell ein Ensemble von $N = 10$ Multilayer-Perzeptronen ausgewählt (vgl. Abschnitt 3.4.5), da diese universale Approximatoren sind, die jede stetige Funktion in einem kompakten Raum approximieren können, wenn Netzwerkarchitektur und die Gewichte geeignet gewählt werden (vgl. Kolmogorov, 1957; Cybenko, 1989; Hornik et al., 1989; Hornik, 1991). Abbildung 5.7 zeigt ein Ablaufdiagramm zum Training eines Meta-Modells basierend auf den folgenden Inputs: einer Meta-Trainingsmenge $\mathbf{D}_{\text{train}}$ bestehend aus einer Merkmalsmatrix \mathbf{X} und einer Label-Matrix \mathbf{L} , den Hyperparametern der Netzwerkarchitektur $N_1, N_2, N_3, g_1, g_2, g_3$ und N sowie den Parametern, die den Trainingsprozess spezifizieren, dem Lernverfahren, dem Fehlermaß, der Lernrate, der maximalen Anzahl an Lernepochen, der maximalen Anzahl aufeinanderfolgender Fehlversuche, das Fehlermaß auf der Meta-Validierungsmenge zu verbessern und den Prozentsätzen der Zeitreihen, die für die Meta-Trainings- und die Meta-Validierungsmenge verwendet werden sollen. Für das Training des Meta-Modells wird jedes der N neuronalen Netze initialisiert und darauffolgend basierend auf einer zufälligen Einteilung der Meta-Trainingsmenge in eine kleinere Meta-Trainings- und eine Meta-Validierungsmenge verwendet. In dieser Arbeit werden jeweils

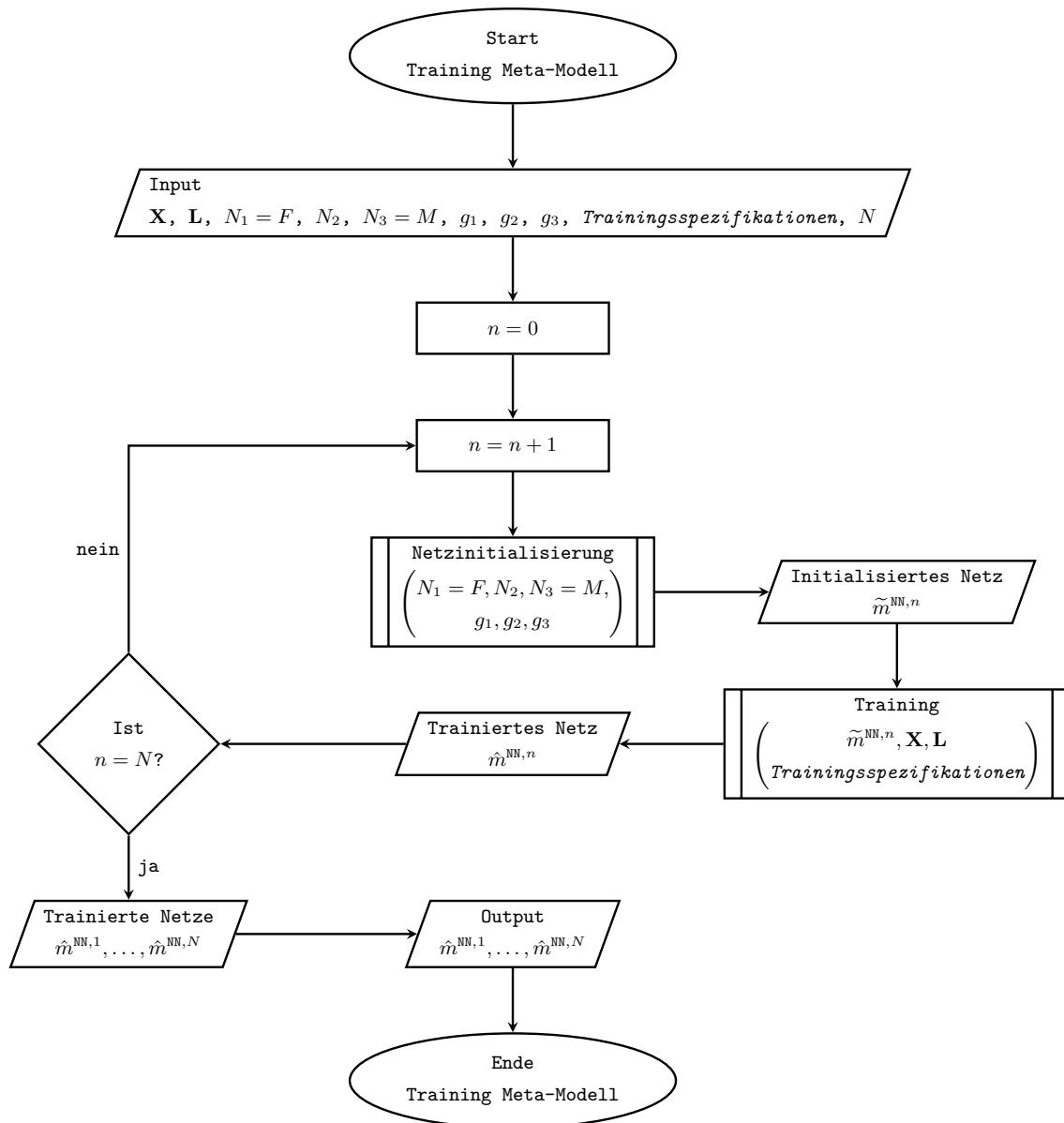


Abbildung 5.7: Ablaufdiagramm zum Training eines Meta-Modells

15% der gesamten Meta-Trainingsmenge als Meta-Validierungsmenge genutzt. Das Ergebnis der Trainingsphase ist ein trainiertes Ensemble von $N = 10$ neuronalen Netzen als Meta-Modell.

In der vorliegenden Arbeit werden Multilayer-Perzeptrene mit einer verdeckten Schicht verwendet, um die Zusammenhänge zwischen Zeitreihencharakteristika und den Eignungen verschiedener Prognoseverfahrenskandidaten zu lernen. Ein Großteil der Konfigurationen der neuronalen Netze als Prognosemodell (vgl. Abschnitt 3.4.5) lässt sich auch für die Konfiguration eines neuronalen Netzes als Klassifikationsmodell verwenden, einige sollten jedoch angepasst werden. Im Folgenden werden diejenigen Konfigurationen und Konzepte erläutert, die bei der Konfiguration als Klassifikationsmodell geändert werden.

Abbildung 5.8 visualisiert die Architektur der neuronalen Netze, die als Meta-Modell trainiert werden. Bei der Definition der Netzwerkarchitektur werden gängige Empfehlungen aus der Literatur beachtet (vgl. Rehkugler und Zimmermann, 1994; Bishop, 1995; Neuneier und Zimmermann, 2012; The MathWorks Inc., 2017a; Hyndman et al., 2018b). N_1 ist als die Anzahl der betrachteten Merkmale F gegeben und N_3 als die Anzahl der Prognoseverfahrenskandidaten M . Die Anzahl der Neuronen in der verdeckten Schicht N_2 besitzt einen großen Einfluss auf die Fähigkeit eines neuronalen Netzes, Muster zu erkennen. Falls N_2 zu groß gewählt wird, kann ein Netz zur Überanpassung neigen, während eine Unteranpassung im Fall eines zu geringen Wertes auftreten kann. Basierend auf Empfehlungen in Rehkugler und Zimmermann (1994) und Hyndman et al. (2018b) wird $N_2 = F/2 + 1$ gewählt, wobei auf die nächstgrößere ganze Zahl aufgerundet wird. Die Aktivierungsfunktion g_1 der Eingabeschicht wird als Identitätsfunktion gewählt. Als Aktivierungsfunktionen der verdeckten Schicht und der Ausgabeschicht werden die Tangens-Hyperbolicus-Funktion

$$g_2(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5.8)$$

und die Softmax-Funktion

$$g_3(x) = \text{softmax}(x_j) = \frac{e^{x_j}}{\sum_{n=1}^M e^{x_n}}. \quad (5.9)$$

gewählt. Die Tangens-Hyperbolicus-Funktion ist eine gängige Aktivierungsfunktion für verdeckte Schichten neuronaler Netze, die als Klassifikationsmodelle trainiert werden. Die Softmax-Funktion bewirkt, dass die Summe der Outputs des neuronalen Netzes 1 ergibt und somit können die Outputs als Klassenwahrscheinlichkeiten betrachtet werden. Als Fehlermaß wird die Kreuzentropie für Multi-Klassen-Klassifikationsprobleme (3.13) verwendet, welche zu minimieren ist.

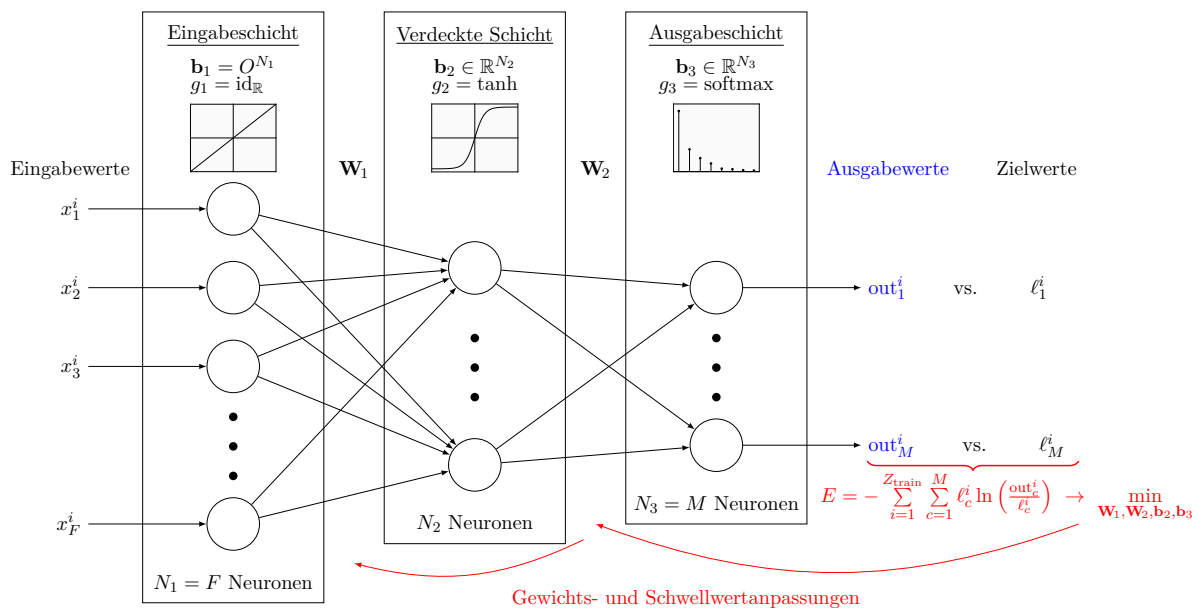


Abbildung 5.8: Training eines Multilayer-Perzeptrons für Klassifikationsprobleme

Um ein geeignetes Lernverhalten des neuronalen Netzes während des Trainingsprozesses zu erreichen wird die `traingdx`-Funktion der `Neural Network Toolbox` in `Matlab` verwendet (vgl. The MathWorks Inc., 2017a), welche einen Gradientenabstieg mit Momentum, adaptiver Lernrate und vorzeitigem Stoppen (engl. *early stopping*) durchführt. Die Parameteranpassungen in jeder Lernepoche werden durch

$$\Delta \mathbf{w}_{(t)} = -\eta_{(t)} \frac{\partial E(\mathbf{w}_{(t)})}{\partial \mathbf{w}_{(t)}} + \mu \Delta \mathbf{w}_{(t-1)} \quad (5.10)$$

ermittelt. Neben der Lernrate η hängt die Parameteranpassung vom *Momentum* $\mu \in [0, 1]$ ab. In Bereichen des Parameterraums mit geringer Krümmung der Fehleroberfläche kann ein Momentum zu einer schnelleren Konvergenz führen als ein Standardgradientenabstieg, da die Parameteranpassungen effektiver gewählt werden (vgl. Rumelhart et al., 1986a; Bishop, 1995; Qian, 1999; Duda et al., 2001). Zudem kann ein Standardgradientenabstieg ohne Momentum in Bereichen des Parameterraums mit hoher Krümmung der Fehleroberfläche zu divergenten Oszillationen führen. Auch in diesem Fall kann ein Momentum den Trainingsprozess verbessern, da hierdurch die Oszillationen vermieden werden und somit eine schnellere Konvergenz erreicht werden kann. Die `traingdx`-Funktion wird mit den Standardparametern zur Adaption der Lernrate verwendet (vgl. The MathWorks Inc., 2017a). Falls der Fehler in einer Epoche sinkt, wird die Lernrate erhöht, um den Prozess zu beschleunigen. Falls der Fehler leicht steigt, wird die Lernrate beibehalten. Dies bietet die Möglichkeit, ein lokales Minimum zu verlassen und nach einem globalen Minimum zu suchen. Im Fall eines stärkeren Anstiegs des Fehlers werden die Parameteranpassungen der vorherigen Epoche rückgängig gemacht. Außerdem wird die Lernrate vermindert, um zu versuchen, aus einer möglichen Divergenz in eine Konvergenz überzugehen.

Um eine hohe Klassifikationsgüte des Meta-Modells für ungesehene Zeitreihen in der Einsatzphase zu erreichen, werden die neuronalen Netze basierend auf der Meta-Trainingsmenge trainiert und die Klassifikationsgüte hinsichtlich der Kreuzentropie der jeweiligen Netzkonfiguration wird in jeder Lernepoche anhand der Meta-Validierungsmenge validiert. Darauffolgend werden die Netzwerkparameter angepasst und die nächste Lernepoche beginnt. Die `traingdx`-Funktion stoppt den Trainingsprozess, wenn die maximale Anzahl von 100000 Lernepochen erreicht ist oder, wenn die Kreuzentropie auf der Meta-Validierungsmenge für 200 aufeinanderfolgende Epochen nicht sinkt. Diese Werte werden unterschiedlich zu den Standardeinstellungen der `traingdx`-Funktion gewählt, um die Möglichkeit der Konvergenz zu einem niedrigeren Minimum zu ermöglichen, falls der Trainingsprozess sich für mehrere Epochen in der Nähe eines lokalen Minimums befindet. Nach der Beendigung des Trainingsprozesses wird die Netzkonfiguration mit dem niedrigsten Wert der Kreuzentropie ausgewählt.

Da die Klassifikationsgüte neuronaler Netze von der Initialisierung der Netzgewichte abhängt, werden $N = 10$ Multilayer-Perzeptrene basierend auf 10 verschiedenen Unterteilungen der Meta-Trainingsmenge in eine kleinere Meta-Trainings- und eine Meta-Validierungsmenge trainiert. Dadurch wird ermöglicht, dass die verschiedenen Netze die Zusammenhänge in verschiedenen

Trainingsbeispielen lernen und somit wird eine Überanpassung an die Daten vermieden. Das finale Meta-Modell ist ein Ensemble von 10 neuronalen Netzen

$$\hat{m}^{\text{NN}} = \left(\hat{m}^{\text{NN},1}, \dots, \hat{m}^{\text{NN},N} \right). \quad (5.11)$$

5.2.5 Merkmalsselektion

Zur Optimierung der Klassifikationsgüte des Meta-Modells wird eine *Merkmalsselektion* (engl. feature selection) durchgeführt, um die wichtigsten Merkmalsextraktionsfunktionen auszuwählen (vgl. Tabelle 5.1). Zu diesem Zweck wird eine Step-Down-Prozedur verfolgt, die mit der vollständigen Menge \mathbf{F} beginnt. In jedem Iterationsschritt werden die Wichtigkeiten aller Merkmale bestimmt und das unwichtigste Merkmal wird entfernt, bis nur noch ein Merkmal übrig ist. Letztlich wird die Menge an Merkmalsextraktionsfunktionen \mathbf{F} aus dem Iterationsschritt mit der niedrigsten Kreuzentropie gewählt. Bevor die Iteration beginnt, wird die Meta-Trainingsmenge zufällig in eine kleinere Meta-Trainings- und eine Meta-Validierungsmenge unterteilt. Diese Unterteilung wird für den gesamten Merkmalsselektionsprozess beibehalten, um zueinander vergleichbare Ergebnisse zu erhalten. Jeder Iterationsschritt besteht aus den folgenden Teilschritten: (i) Training eines Meta-Modells aus 10 neuronalen Netzen basierend auf der Meta-Trainingsmenge, (ii) Anwendung des trainierten Meta-Modells auf alle Merkmalsvektoren \mathbf{x}^i in der Meta-Validierungsmenge, um die Ausgabevektoren zu prognostizieren, (iii) Berechnung der Kreuzentropie, (iv) Bestimmung der Wichtigkeiten aller verbleibenden Merkmale, (v) Entfernung des unwichtigsten Merkmals aus der Menge \mathbf{F} . Diese Teilschritte werden im Folgenden erläutert.

Ausgehend von der kompletten Menge an Merkmalsextraktionsfunktionen \mathbf{F} und der kompletten Merkmalsmatrix \mathbf{X} , die in Abschnitt 5.2.2 beschrieben wurden, wird im ersten Teilschritt einer Iteration ein Meta-Modell basierend auf der Meta-Trainingsmenge bestehend aus Merkmalsmatrix und Label-Matrix trainiert. Das Ergebnis ist ein trainiertes Ensemble neuronaler Netze als Meta-Modell. Darauffolgend wird im zweiten Teilschritt das Meta-Modell verwendet, um für die Merkmale der Meta-Validierungsmenge die Eignungswahrscheinlichkeiten der Prognoseverfahrenskandidaten als Ausgabevektoren zu prognostizieren. Die Validierungsgüte der Konfiguration des Meta-Modells wird im dritten Teilschritt bestimmt, indem die Kreuzentropie zwischen den prognostizierten Ausgabevektoren und den wahren Labels in \mathbf{L} berechnet wird. Diese Validierungsgüte wird als Referenzwert verwendet, um zu ermitteln, ob die verschiedenen Merkmalsvektoren wichtig für die Klassifikationsgüte des Meta-Modells sind oder nicht. Zur Bestimmung der Wichtigkeit der Merkmalsvektoren werden im vierten Teilschritt der Iteration zunächst die Werte einer Spalte der Merkmalsmatrix für die Meta-Validierungsmenge zufällig permutiert. Das Meta-Modell wird auf die Meta-Validierungsmenge mit der veränderten Merkmalsmatrix angewandt und die Validierungsgüte wird durch die Kreuzentropie bestimmt. Dieser Prozess wird für alle Spalten der Merkmalsmatrix der Meta-Validierungsmenge durchgeführt, um die Wichtigkeiten aller Merkmale zu bestimmen. Da die Permutation der Elemente der Merkmalsvektoren zufällig erfolgt, werden

Mittelwerte der Kreuzentropie aus 10 Wiederholungen der Permutation und der anschließenden Bestimmung der Kreuzentropie auf der Meta-Validierungsmenge gebildet, um robuste Ergebnisse zu erzielen. Ein Merkmal wird als wichtig betrachtet, falls die Kreuzentropie des Meta-Modells steigt und somit die Validierungsgüte sinkt, wenn die Elemente des Merkmals zufällig permutiert werden. Demgegenüber wird ein Merkmal als unwichtig betrachtet, wenn die Kreuzentropie sinkt oder sich nicht verändert. Anhand dieser Regeln werden alle verbleibenden Merkmale nach Wichtigkeit geordnet und das unwichtigste Merkmal wird im fünften Teilschritt der Iteration aus der Merkmalsmatrix und der Menge der Merkmalsextraktionsfunktionen entfernt. Falls alle verbleibenden Merkmale wichtig sind, wird das am wenigsten wichtige Merkmal entfernt. Darauf folgend beginnt der nächste Iterationsschritt nach demselben Schema. Der Merkmalsselektionsprozess endet, wenn nur noch ein Merkmal übrig ist und es wird die Merkmalsmatrix aus dem Iterationsschritt mit der besten Validierungsgüte selektiert. Die Ergebnisse des Prozesses sind eine verringerte Menge an Merkmalsextraktionsfunktionen sowie eine verkleinerte Merkmalsmatrix.

Abbildung 5.9 zeigt die Validierungsgüten hinsichtlich der Kreuzentropie für die Meta-Modelle mit den verschiedenen Merkmalsmatrizen im Merkmalsselektionsprozess. Hierbei wird die Konfiguration mit den vier Prognoseverfahrenskandidaten llknn, ets, nmetar und srw sowie die Konfiguration mit den 89 Merkmalsextraktionsfunktionen aus Tabelle 5.1 verwendet. Die beste Validierungsgüte wird für eine Teilmenge von 25 Merkmalen erreicht, die in Tabelle 5.2 aufgeführt sind. Es bleiben Merkmale aus allen betrachteten Merkmalsmengen übrig. Die finale Menge an Merkmalsextraktionsfunktionen besteht aus 6 RQA-Maßen, 5 klassischen statistischen Maßen, 9 Maßen aus statistischen Tests sowie 5 Landmarkers. Hierbei ist auffällig, dass alle übrig bleibenden Landmarkers aus den beiden in dieser Arbeit neu eingeführten Mengen der

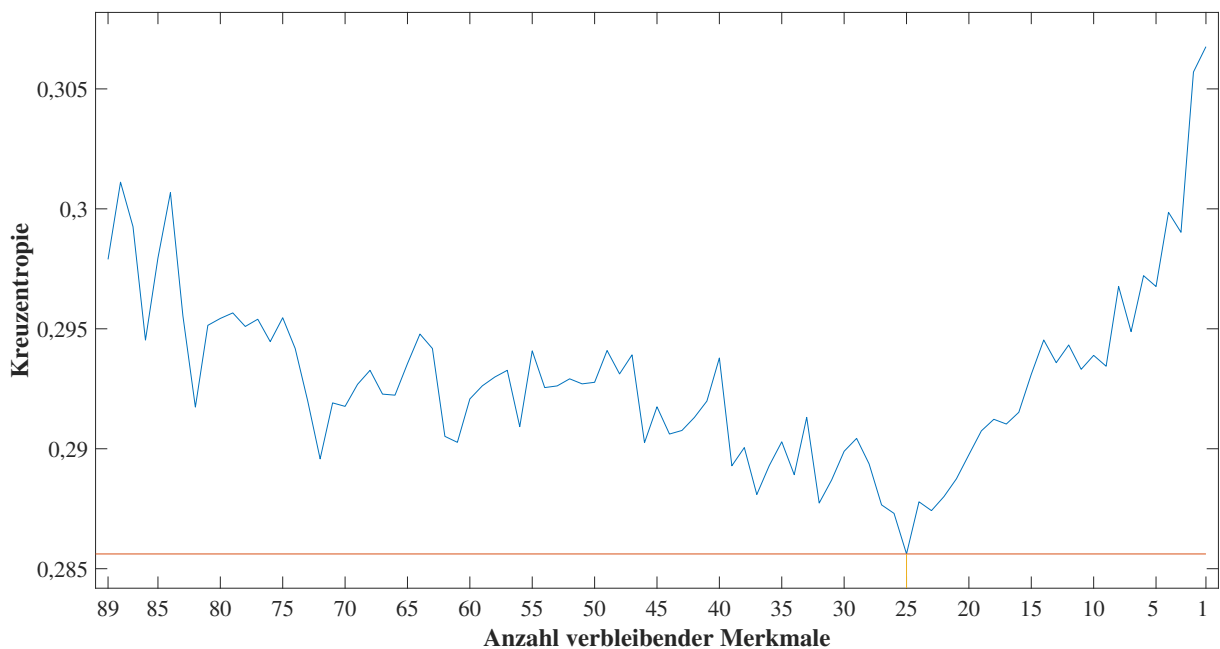


Abbildung 5.9: Validierungsgüte der Meta-Modelle im Merkmalsselektionsprozess

Tabelle 5.2: Nach der Merkmalsselektion verbleibende Meta-Merkmale der Typen direkter Merkmale (D), modellbasierter Merkmale (M) und Landmarkers (L), um ein Prognosemodellselektionsproblem zu beschreiben

Nr.	Merkmal	Typ	Nr.	Merkmal	Typ
1	Länge der längsten Diagonallinie	M	14	ADF-Stationaritätstest (c)	M
2	Entropie der Diagonallänge	M	15	ADF-Stationaritätstest (ct)	M
3	Laminarität	M	16	Kruskal-Wallis-Test für 12-monatl. Saisonalität	M
4	Rekurrenzzeit erster Art	M	17	Anzahl hoch-signifikanter ACF-Lags	M
5	Rekurrenzzeit-Entropie	M	18	Anzahl nicht-signifikanter PACF-Lags	M
6	Transitivität	M	19	Jarque-Bera-Normalitätstest (asymptotisch)	M
7	Anzahl der Zeitreihenwerte	D	20	Jarque-Bera-Normalitätstest (Monte-Carlo)	M
8	Minimum	D	21	Verhältnis llknn / ets	L
9	unteres Quartil	D	22	Verhältnis llknn / srw	L
10	Variationskoeffizient	D	23	Verhältnis ets / srw	L
11	Saisonalität	D	24	Verhältnis nnetar / srw	L
12	Mann-Kendall-Test für monotonen Trend	M	25	Differenz llknn - ets	L
13	Spearman-Trend-Test für Rangkorrelation	M			

Verhältnisse und der Differenzen der Validierungsfehler stammen. Somit lassen sich diese Maße, die jeweils die Validierungsfehler zweier Prognoseverfahren für eine Zeitreihe vergleichen, besser zur Beschreibung der Selektionsprobleme für das Meta-Modell nutzen als die Validierungsfehler selbst. In den empirischen Studien dieser Arbeit wird die reduzierte Menge von 25 Merkmalsextraktionsfunktionen für das Training und den Einsatz des Meta-Modells verwendet. Für die Evaluationen hinsichtlich der Robustheit des Selbinationsverfahrens in Abschnitt 6.4 werden verschiedene Konfigurationen des Ansatzes hinsichtlich der Menge an Verfahrenskandidaten und der Menge an Merkmalsextraktionsfunktionen verglichen. Für jede dieser Konfigurationen wird ein eigener Merkmalsselektionsprozess durchgeführt. Die Ergebnisse der Merkmalsselektionen sind in Anhang D dargestellt.

5.3 Einsatz des Selbinationsverfahrens

5.3.1 Übersicht

Abbildung 5.10 gibt einen Überblick über den Einsatz des neuartigen Selbinationsverfahrens zur Selektion oder Kombination geeigneter Prognosemodelle für eine Zeitreihe y . Der Ansatz besteht aus der Verwendung eines trainierten Meta-Modells zur Bestimmung von Eignungswahrscheinlichkeiten für Prognoseverfahrenskandidaten sowie der Selektion eines oder mehrerer Prognosemodelle basierend auf einer sogenannten Hybriditätsgrenze. Im Folgenden werden die verschiedenen Teile von Abbildung 5.10 erläutert.

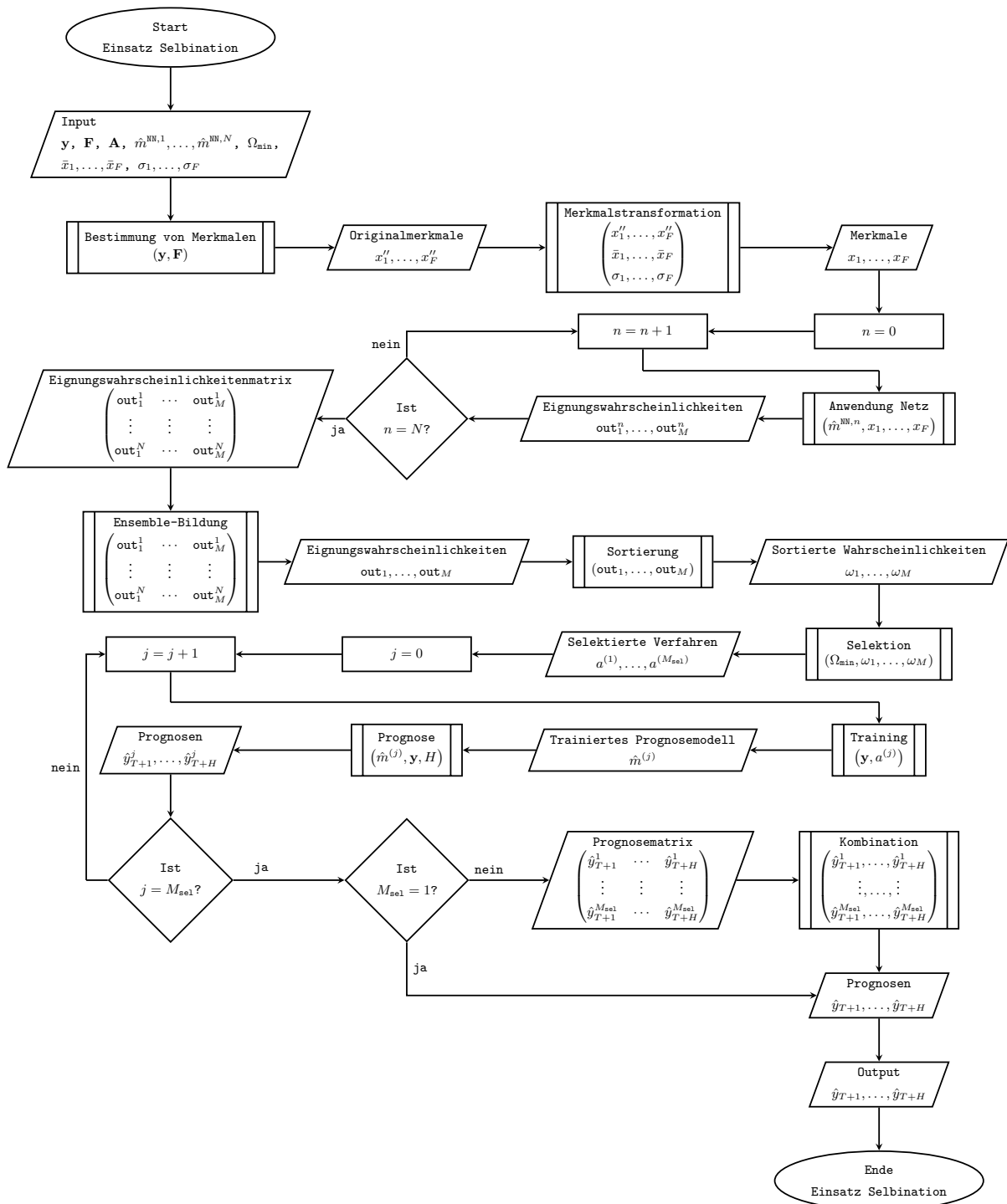


Abbildung 5.10: Ablaufdiagramm zum Einsatz des Selbinationsverfahrens zur Selektion oder Kombination geeigneter Prognosemodelle

5.3.2 Eignungswahrscheinlichkeiten von Prognoseverfahren

Der Einsatz des Selbinationsverfahrens beginnt mit der Extraktion von Merkmalen sowie der Anwendung des trainierten Meta-Modells $\hat{m}^{\text{NN}} = (\hat{m}^{\text{NN},1}, \dots, \hat{m}^{\text{NN},N})$ für eine neue Zeitreihe \mathbf{y} . Zunächst werden die gleichen Merkmalsextraktionsfunktionen in \mathbf{F} verwendet, die auch in der Meta-Trainingsphase genutzt wurden, um die Zeitreihe durch Merkmale zu beschreiben. Zur Erreichung des korrekten Eingabeformats werden die resultierenden Originalmerkmale transformiert, indem die Spaltenmittelwerte $\hat{x}_1, \dots, \hat{x}_F$ der Originalmerkmalsmatrix aus dem Trainingsprozess subtrahiert werden und durch die Standardabweichungen $\sigma_1, \dots, \sigma_F$ der Spalten dividiert wird. Darauffolgend werden alle N trainierten neuronalen Netze auf die Merkmale x_1, \dots, x_F angewandt, um N Vektoren von Eignungswahrscheinlichkeiten für die M Verfahrenskandidaten zu bestimmen. Nach einer Mittelwertbildung über die N Ausgabevektoren der neuronalen Netze ist das Ergebnis des Ensemble-Outputs ein Vektor mit M Elementen

$$\hat{m}^{\text{NN}}(x_1, \dots, x_F) = (\text{out}_1, \dots, \text{out}_M), \quad (5.12)$$

welcher die finalen Eignungswahrscheinlichkeiten der M Prognoseverfahrenskandidaten für die Zeitreihe enthält. Abschnitt 5.3.3 beschreibt die Verwendung dieses Vektors zur Selektion eines oder mehrerer Prognosemodelle.

5.3.3 Hybriditätsgrenze

Im Allgemeinen existieren zwei verschiedene Ansätze zur Verwendung von Eignungswahrscheinlichkeiten. Selektionsansätze selektieren das eine Modell mit der höchsten Eignungswahrscheinlichkeit, während Kombinationsansätze die Wahrscheinlichkeiten entweder nutzen, um mehrere geeignete Modelle für eine Kombination zu selektieren oder, um Kombinationsgewichte zu bestimmen. Beide Ansätze haben Vor- und Nachteile in Abhängigkeit von den Charakteristika einer vorliegenden Zeitreihe und den zur Verfügung stehenden Prognoseverfahrenskandidaten. Selektionsansätze lassen sich verwenden, wenn zu erwarten ist, dass ein Prognosemodell bessere Prognosen für eine Zeitreihe erzielen wird als alle anderen Modelle, während Kombinationsansätze Vorteile bieten, wenn entweder ähnlich gute Prognosen von mehreren Modellen zu erwarten sind oder unklar ist, welche Modelle für eine Zeitreihe geeignet sind. Zur Ausnutzung der Vorteile beider Ansätze stellt das neuartige Selbinationsverfahren dieser Arbeit einen hybriden Ansatz dar, welcher automatisch in Abhängigkeit von den Charakteristika einer vorliegenden Zeitreihe entweder eine Selektion oder eine Kombination durchführt. Hierbei werden formal keine Modelle direkt selektiert oder kombiniert, sondern das Selbinationsverfahren führt die folgenden Schritte durch: (i) Selektion eines oder mehrerer geeigneter Prognoseverfahren für eine Zeitreihe, (ii) Verwendung aller selektierten Prognoseverfahren, um je ein Prognosemodell zu trainieren, (iii) Anwendung aller trainierten Modelle zur Berechnung von Prognosen zukünftiger Zeitreihenwerte und möglicherweise (iv) Kombination der Prognosen der verschiedenen Modelle zu einer finalen Prognose. Im Folgenden sind stets diese Prozessschritte gemeint, wenn die Selektion eines Prognosemodells

beschrieben wird. Neben dem Vektor mit Eignungswahrscheinlichkeiten für verschiedene Verfahrenskandidaten (5.12) umfasst das Selbinationsverfahren eine sogenannte *Hybriditätsgrenze* (engl. hybridity threshold) Ω_{\min} . Dieser Parameter bezeichnet den Prozentsatz, den die Summe der Eignungswahrscheinlichkeiten der selektierten Prognoseverfahren mindestens ergeben soll. Im Folgenden wird die Hybriditätsgrenze formal beschrieben und anhand von Beispielen erläutert.

Sei $\mathbf{out} = (\text{out}_1, \dots, \text{out}_M)$ ein Vektor mit Eignungswahrscheinlichkeiten der Prognoseverfahrenskandidaten $a^1, \dots, a^M \in \mathbf{A}$ für eine Zeitreihe \mathbf{y} . Der j -te Wert out_j dieses Vektors spiegelt die Wahrscheinlichkeit wider, dass das j -te Prognoseverfahren a^j ein Modell trainieren wird, welches die höchste Prognosegüte für \mathbf{y} im Vergleich zu den anderen Kandidaten hat. Per Definition beträgt die Summe aller M Verfahrenskandidaten 1:

$$\sum_{j=1}^M \text{out}_j = 1. \quad (5.13)$$

Sei ω_j der j -höchste Wert $\text{out}_{(j)}$ des Vektors \mathbf{out} nach einer Sortierung der Werte von hoch zu niedrig, d.h.

$$\omega_1 \geq \omega_2, \dots, \geq \omega_M, \quad \text{mit} \quad \sum_{j=1}^M \omega_j = 1. \quad (5.14)$$

Sei $a^{(j)}$ das Prognoseverfahren mit der j -höchsten Eignungswahrscheinlichkeit $\text{out}_{(j)} = \omega_j$. Dann ist die *Hybriditätsgrenze* (engl. hybridity threshold) Ω_{\min} definiert als der minimale Wert der Summe der zu den selektierten Verfahren gehörenden Werte ω_j . Für einen gewählten Wert von Ω_{\min} ist die M_{sel} -fache Selektionsfunktion des Selbinationsverfahrens definiert als

$$s_{M_{\text{sel}}} \in \mathcal{S}, \quad s_{M_{\text{sel}}} : [0, 1]^M \longrightarrow (\mathbf{A})^{M_{\text{sel}}}, \quad s_{M_{\text{sel}}}(\mathbf{out}) = \{a^{(1)}, \dots, a^{(M_{\text{sel}})}\}, \quad (5.15)$$

mit

$$M_{\text{sel}} = \underset{A}{\operatorname{argmin}} \left\{ \Omega \in [0, 1] : \Omega = \sum_{j=1}^A \omega_j \geq \Omega_{\min} \right\}. \quad (5.16)$$

Somit werden nach der Bestimmung des Ausgabevektors des Meta-Modells die M_{sel} Verfahren mit den höchsten Eignungswahrscheinlichkeiten für die gegebene Zeitreihe selektiert, sodass die kumulative Summe der dazugehörigen Eignungswahrscheinlichkeiten mindestens Ω_{\min} ergibt. Die Hybriditätsgrenze Ω_{\min} bestimmt die Anzahl zu selektierender Prognoseverfahren. Kleine Werte des Parameters Ω_{\min} führen tendenziell zu einer Selektion weniger Prognoseverfahren, während hohe Werte eine Selektion mehrerer Verfahren begünstigt. Insbesondere führt $\Omega_{\min} = 0$ zu einer Selektion des einen Verfahrens mit der höchsten Eignungswahrscheinlichkeit und $\Omega_{\min} = 1$ zu einer Kombination aller Prognoseverfahrenskandidaten. Andere Werte für Ω_{\min} können zu einer Selektion eines oder zu einer Kombination mehrerer Prognoseverfahren führen. Daher lässt sich das neuartige Selbinationsverfahren dieser Arbeit als ein allgemeines Modellselektionsverfahren bezeichnen, welches die Selektion und die Kombination als Spezialfälle enthält.

Tabelle 5.3 zeigt zwei Beispiele zur Anwendung der Selektionsfunktion (5.15) zur Selektion eines oder mehrerer Prognoseverfahren aus vier Kandidaten basierend auf fiktionalen Eignungswahrscheinlichkeiten und verschiedenen Werten der Hybriditätsgrenze Ω_{\min} . Das erste Beispiel

zeigt den Fall eines dominanten Prognoseverfahrens mit einer hohen Eignungswahrscheinlichkeit. Für eine Definition der Hybriditätsgrenze von $\Omega_{\min} \leq 0,72$ wird lediglich Verfahren a^2 selektiert, während die Grenzen zur zusätzlichen Selektion der weiteren Verfahrenskandidaten und Durchführung einer Prognosekombination bei 0,84 und 0,94 liegen. Im zweiten Beispiel haben mehrere Verfahrenskandidaten ähnliche Eignungswahrscheinlichkeiten. In diesem Fall wird eine Selektion eines Verfahrens nur bis zu einem Wert $\Omega_{\min} \leq 0,4$ durchgeführt und die Grenzen für die zusätzliche Selektion der weiteren Prognoseverfahren sind 0,7 und 0,95. Beim Vergleich der beiden Beispiele lässt sich feststellen, dass für gegebene Werte der Hybriditätsgrenze Ω_{\min} die Anzahl der selektierten Prognoseverfahren M_{sel} auf den Eignungswahrscheinlichkeiten beruht. Wenn Ω_{\min} sinnvoll gewählt ist, wird durch das Selbinationsverfahren eine Selektion eines einzelnen Prognoseverfahrens immer dann durchgeführt, wenn ein dominantes Verfahren zu erwarten ist und eine Kombination wird durchgeführt, wenn zu erwarten ist, dass mehrere Verfahren ähnlich gute Ergebnisse erzielen werden. Zur Analyse der Abhängigkeit des Selbinationsverfahrens von der Hybriditätsgrenze wird in Abschnitt 5.4 ein Parametervergleich durchgeführt.

Tabelle 5.3: Beispiele selektierter Prognoseverfahren basierend auf Eignungswahrscheinlichkeiten und der Hybriditätsgrenze Ω_{\min} im Fall (a) eines erwarteten dominanten Prognoseverfahrens und (b) multipler Verfahren mit ähnlichen erwarteten Prognosegütern

(a) Ein erwartetes dominantes Verfahren

	Prognoseverfahren			
	a^1	a^2	a^3	a^4
Eignungswahrscheinlichkeiten	0,12	0,72	0,06	0,10
Sortierte Eignungswahrscheinlichkeiten	ω_2	ω_1	ω_4	ω_3
Selektion eines Verfahrens für $\Omega_{\min} \leq 0,72$		✓		
Kombination von zwei Verfahren für $0,72 < \Omega_{\min} \leq 0,84$	✓	✓		
Kombination von drei Verfahren für $0,84 < \Omega_{\min} \leq 0,94$	✓	✓		✓
Kombination von vier Verfahren für $\Omega_{\min} > 0,94$	✓	✓	✓	✓

(b) Multiple ähnliche erwartete Prognosegütern

	Prognoseverfahren			
	a^1	a^2	a^3	a^4
Eignungswahrscheinlichkeiten	0,05	0,30	0,25	0,40
Sortierte Eignungswahrscheinlichkeiten	ω_4	ω_2	ω_3	ω_1
Selektion eines Verfahrens für $\Omega_{\min} \leq 0,40$				✓
Kombination von zwei Verfahren für $0,40 < \Omega_{\min} \leq 0,70$		✓		✓
Kombination von drei Verfahren für $0,70 < \Omega_{\min} \leq 0,95$		✓	✓	✓
Kombination von vier Verfahren für $\Omega_{\min} > 0,95$	✓	✓	✓	✓

5.3.4 Einsatz von Prognoseverfahren und -modellen

Nach der Selektion der M_{sel} Prognoseverfahren $a^{(1)}, \dots, a^{(M_{\text{sel}})}$ mit den höchsten Eignungswahrscheinlichkeiten für eine gegebene Zeitreihe $\mathbf{y}_{\text{train}}$ durch Verwendung des Meta-Modells sowie der M_{sel} -fachen Selektionsfunktion (5.15) basierend auf der Hybriditätsgrenze Ω_{min} werden die selektierten Prognoseverfahren verwendet, um M_{sel} Prognosemodelle $\hat{m}^{(1)}, \dots, \hat{m}^{(M_{\text{sel}})}$ basierend auf der gegebenen Trainingsmenge der Zeitreihe zu trainieren. Darauffolgend wird jedes Modell $\hat{m}^{(j)}$ verwendet, um eine h -Schritt Prognose wie folgt zu bestimmen:

$$\hat{m}_h^{(j)}(\mathbf{y}_{\text{train}}) = \hat{\mathbf{y}}_{T|T+h}^{(j)}. \quad (5.17)$$

Falls mehrere Prognosemodelle trainiert wurden ($M_{\text{sel}} > 1$), wird der Mittelwert der Prognosen der Modelle gebildet, um die finale h -Schritt-Prognose zu erhalten:

$$\hat{\mathbf{y}}_{T|T+h}^{\text{Comb}} = \hat{m}_h^{\text{Comb}}(\mathbf{y}_{\text{train}}) = \frac{1}{M_{\text{sel}}} \sum_{j=1}^{M_{\text{sel}}} \hat{\mathbf{y}}_{T|T+h}^{(j)}. \quad (5.18)$$

Dies ist das Ergebnis der Einsatzphase des Selbinationsverfahrens.

5.4 Konfiguration der Hybriditätsgrenze

In diesem Abschnitt werden unterschiedliche Konfigurationen der Hybriditätsgrenze Ω_{min} evaluiert, welche zu verschiedenen Prognosegüten des Selbinationsverfahrens führen. Zunächst wird die Experimentkonfiguration beschrieben. Darauffolgend werden die Ergebnisse des Parametervergleichs dargestellt und es wird ein Wert der Hybriditätsgrenze ausgewählt, welcher in den Evaluationsstudien der nächsten beiden Kapitel verwendet wird.

5.4.1 Experimentbeschreibung

Parameterbereich

Im folgenden Experiment wird eine sinnvolle Hybriditätsgrenze Ω_{min} für die Konfiguration des Selbinationsverfahrens mit den vier Verfahrenskandidaten `llknn`, `ets`, `nnetar` und `srw` sowie der Menge an 25 Merkmalsextraktionsfunktionen in Tabelle 5.2, die nach der Merkmalsselektion übrig bleiben, gesucht. Dies ist die Konfiguration des Selbinationsverfahrens, die in den empirischen Studien der nächsten Kapitel verwendet wird. Ein Vergleich der Leistungsfähigkeiten verschiedener Konfigurationen des Selbinationsverfahrens wird in Abschnitt 6.4 durchgeführt.

Um eine sinnvolle Konfiguration der Hybriditätsgrenze zu bestimmen, werden die 9 Werte im Bereich $0, \frac{3}{10}, \frac{4}{10}, \dots, 1$ betrachtet. Die Konfiguration $\Omega_{\text{min}} = 0$ führt zu einer Selektion des einen Verfahrens mit der höchsten Eignungswahrscheinlichkeit und die Konfiguration $\Omega_{\text{min}} = 1$ bewirkt eine Kombination aller Prognoseverfahrenskandidaten für alle Zeitreihen. Da die höchste Eignungswahrscheinlichkeit bei M Verfahrenskandidaten mindestens $\frac{1}{M}$ beträgt, führen die Werte

im Bereich $0, \dots, \frac{1}{M}$ zum gleichen Selektionsergebnis, nämlich der Selektion des Verfahrens mit der höchsten Eignungswahrscheinlichkeit. Für die Konfiguration mit $M = 4$ Verfahrenskandidaten bewirken somit die Werte im Bereich $0, \dots, \frac{1}{4}$ die Selektion eines Prognoseverfahrens.

Datensatz und Unterteilung für den Trainingsprozess

Als Datensatz zur Evaluation der Prognosegüte des Selbbinationsverfahrens mit den verschiedenen Parameterkonfigurationen wird die Teilmenge \mathbf{Y} der 808 monatlichen Zeitreihen der Kategorien Industrie und Mikroökonomie der M3-Competition genutzt (vgl. Abschnitt 3.3). Wie in der Competition wird jede Zeitreihe in eine Trainingsmenge $\mathbf{y}_{\text{train}} = \{y_1, y_2, \dots, y_{T_z}\}$ und eine Testmenge $\mathbf{y}_{\text{test}} = \{y_{T_z+1}, y_{T_z+2}, \dots, y_{T_z+H}\}$ unterteilt. Jede Testmenge besteht aus $H = 18$ Werten. Zum Training des Meta-Modells und zur Evaluation des Selbbinationsverfahrens wird der Datensatz \mathbf{Y} der 808 Zeitreihen in zwei Datensätze unterteilt, anhand derer die Meta-Trainings- und die Meta-Testmenge bestimmt werden, welche jeweils aus einer Merkmals- und einer Label-Matrix bestehen. Die Meta-Testmenge wird anhand der 111 Zeitreihen der NN3-Competition (vgl. Abschnitt 3.3 und Crone et al., 2011) bestimmt. Die verbleibenden 697 Zeitreihen in \mathbf{Y} werden genutzt, um die Meta-Trainingsmenge zu bestimmen. Für jedes der neuronalen Netze, die zusammen das Meta-Modell bilden, wird eine Teilmenge von 104 zufällig ausgewählten Beispielen ($\approx 15\%$) der Meta-Trainingsmenge als Meta-Validierungsmenge verwendet. Da die Zeitreihen eindeutig den Mengen zugeordnet werden können, werden diese im Folgenden als Zeitreihen der Meta-Trainings-, Meta-Validierungs- und Meta-Testmenge bezeichnet.

Evaluationskriterium

Zur Auswahl eines sinnvollen Wertes der Hybriditätsgrenze wird ein geeignetes Evaluationskriterium benötigt. Hierzu werden die mittleren Ränge hinsichtlich der Prognosefehler der verschiedenen Parameterkonfigurationen über alle Zeitreihen der Meta-Trainingsmenge verwendet. Da jedes der neuronalen Netze im Ensemble des Meta-Modells basierend auf einer anderen Einteilung der Meta-Trainingsmenge in eine kleinere Meta-Trainings- und eine Meta-Validierungsmenge trainiert wird, wird das Ensemble vor einer Überanpassung geschützt und die gesamte Meta-Trainingsmenge kann als Basis für die Parameterkonfigurationen genutzt werden. Die Prognosegüte wird als durchschnittlicher symmetrischer mittlerer absoluter prozentualer Fehler der rollierenden Prognosen der Horizonte $h = 1, \dots, 12$ von verschiedenen Ursprüngen zwischen $o = T_z, \dots, T_z + 6$ auf den Testmengen der Zeitreihen bestimmt:

$$\text{RO-sMAPE}_{T_z, T_z+6}^{1,12}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{7} \sum_{o=T_z}^{T_z+6} \text{sMAPE}_o^{1,12}(\mathbf{y}, \hat{\mathbf{y}}). \quad (5.19)$$

Jede Testmenge enthält 18 zukünftige Zeitreihenwerte. Es werden 12-Schritt-Prognosen von 7 verschiedenen Ursprüngen erzeugt und die Prognosegüte wird über die so entstehenden 84 prognostizierten Werte pro Zeitreihe gemittelt.

Zur Bestimmung der mittleren Prognosegüte der verschiedenen Parameterkonfigurationen des Selbinationsverfahrens über mehrere Zeitreihen, ist es möglich, die Mean-RO-sMAPEs zu vergleichen, welche in (3.39) in Abschnitt 3.2.3 beschrieben wurden. Allerdings sind diese Werte nicht skaleninvariant und somit sensitiv gegenüber hohen Fehlerwerten für einzelne Zeitreihen. Daher könnte eine Verwendung der Mean-RO-sMAPEs als Selektionskriterium für die Hybriditätsgrenze dazu führen, dass ein Wert ausgewählt wird, welcher zu den besten Prognosen für einige wenige Zeitreihen führt, für die aber alle verwendeten Prognoseverfahren im Durchschnitt hohe Prognosefehler bedingen, und möglicherweise zu schlechten Prognosen für viele Zeitreihen, für die die durchschnittlichen Prognosefehler der meisten Verfahren gering sind. In diesem Zusammenhang ist der mittlere Rang hinsichtlich des RO-sMAPE einer Parameterkonfiguration des Selbinationsverfahrens über alle Zeitreihen ein besseres Selektionskriterium, da dieser unabhängig von der Skala der jeweiligen Prognosefehler angibt, für wie viele Zeitreihen eine Konfiguration gut funktioniert. Die Berechnung der mittleren Ränge von Prognoseverfahren wurde in (3.40) und (3.41) in Abschnitt 3.2.3 definiert. Im Folgenden wird die Hybriditätsgrenze mit dem niedrigsten mittleren Rang über alle Zeitreihen der Meta-Trainingsmenge bestimmt.

5.4.2 Evaluationsergebnisse

Für jede der 697 Zeitreihen der Meta-Trainingsmenge und jeden der 9 Werte der Hybriditätsgrenze Ω_{\min} im Bereich $0, \frac{3}{10}, \frac{4}{10}, \dots, 1$ wurden die folgenden Schritte durchgeführt: (i) Anwendung des Selbinationsverfahrens, um ein oder mehrere Prognoseverfahren zu selektieren, (ii) Anwendung der selektierten Prognoseverfahren um Prognosemodelle zu trainieren, (iii) Anwendung der trainierten Modelle, um zukünftige Zeitreihenwerte zu prognostizieren, (iv) möglicherweise Kombination der Prognosen der verschiedenen Modelle, um eine finale Prognose zu erhalten, (v) Berechnung der Prognosegüte durch (5.19). Darauf folgend wurde für jede Zeitreihe eine Rangfolge der verschiedenen Parameterkonfigurationen der Hybriditätsgrenze durch (3.40) bestimmt und die mittleren Ränge der Konfigurationen über alle 697 Zeitreihen wurden durch (3.41) ermittelt. Abbildung 5.11 zeigt die mittleren Ränge der verschiedenen Parameterwerte der Hybriditätsgrenze des Selbinationsverfahrens in der Konfiguration mit den vier Prognoseverfahrenskandidaten und den 25 Merkmalsextraktionsfunktionen, wie im vorherigen Abschnitt beschrieben wurde. Den niedrigsten mittleren Rang über die 697 Zeitreihen der Meta-Trainingsmenge erreicht die Konfiguration der Hybriditätsgrenze als $\Omega_{\min} = 0,5$. Daher wird dieser Wert für die empirischen Evaluationsstudien der nächsten beiden Kapitel genutzt.

5.5 Implementierung des Selbinationsverfahrens

Das Selbinationsverfahren wurde anhand der Ablaufdiagramme und Beschreibungen der letzten Abschnitte prototypisch in `Matlab` und `R` implementiert. Sowohl für die Bestimmung der Meta-Merkmale als auch für die Anwendung der Prognoseverfahrenskandidaten und das Training des

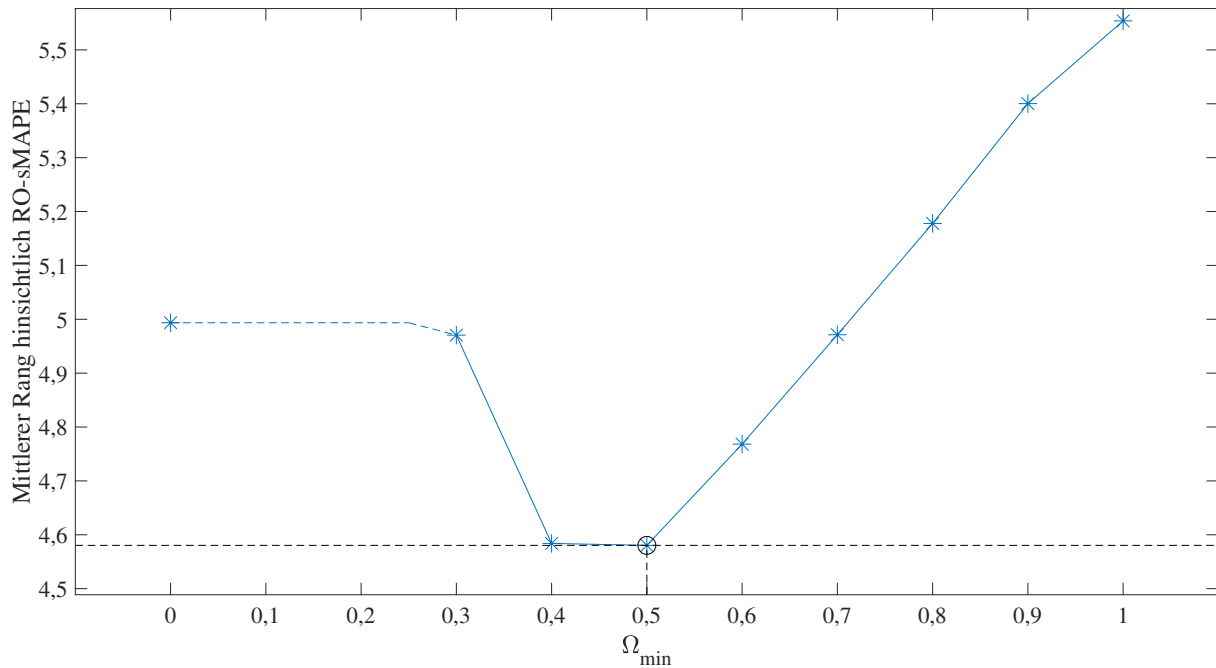


Abbildung 5.11: Vergleich verschiedener Parameterkonfigurationen des Selbinationsverfahrens anhand der mittleren Ränge hinsichtlich des $\text{RO-sMAPE}_{T_z, T_z+6}^{1,12}$ über die 697 Zeitreihen der Meta-Trainingsmenge

Meta-Modells wurden zum Teil gängige Bibliotheken verwendet. Einige Meta-Merkmale wurden mit Hilfe eines Scripts aus Wang et al. (2009), der **Cross Recurrence Plot Toolbox** für **Matlab** (vgl. Marwan et al., 2007) und Funktionen in **iqast** (vgl. RSG Software GmbH, 2016) bestimmt. Die Ermittlung der weiteren Meta-Merkmale sowie die Bereinigungen, Transformationen und Merkmalsselektion wurden in **Matlab** implementiert.

Für sechs der acht genutzten Prognoseverfahrenskandidaten wurden Funktionen der R-Bibliotheken **forecast** und **nnfor** genutzt, die bereits in zahlreichen empirischen Studien hinsichtlich ihrer Funktion verifiziert und evaluiert wurden. Für diese Verfahren wurden gängige Parametereinstellungen gewählt, um vergleichbare Ergebnisse zum Stand der Forschung zu erzielen. Die verbleibenden beiden Prognoseverfahrenskandidaten **lcknn** und **llknn** wurden im Rahmen dieser Arbeit prototypisch in **Matlab** implementiert und in Vorstudien zu dieser Arbeit mit diversen Parameterkonfigurationen evaluiert (vgl. Anhang C sowie Kück und Freitag, 2021).

Für das Training eines Meta-Modells, welches aus einem Ensemble neuronaler Netze besteht, wurde die **Neural Network Toolbox** in **Matlab** verwendet (vgl. The MathWorks Inc., 2017a). Alle weiteren Funktionen für das Training und den Einsatz des Selbinationsverfahrens wurden im Rahmen dieser Arbeit prototypisch in **Matlab** implementiert. Dies sind beispielsweise Funktionen zur Bestimmung von Prognosefehlern, zur Ermittlung von Eignungswahrscheinlichkeiten durch Verwendung des Meta-Modells, zur Selektion von M_{sel} Verfahren basierend auf Eignungswahrscheinlichkeiten und einer Hybriditätsgrenze oder zur Kombination der Prognosen

mehrerer Prognosemodelle. Die korrekte Implementierung der adaptierten und selbst implementierten Funktionen dieser Arbeit wurde im Rahmen von Debugging-Prozessen mit verschiedenen Parameterkonfigurationen anhand beispielhafter Zeitreihen verifiziert. Die Evaluation der implementierten Verfahren erfolgt in den folgenden Kapiteln, in denen das Selbinationsverfahren hinsichtlich der Prognosegüte, des Rechenaufwands und in einer Lagerbestandssimulation gegen Benchmark-Verfahren verglichen wird.

5.6 Zusammenfassung

In diesem Kapitel wurde das neuartige Selbinationsverfahren dieser Arbeit beschrieben, ein hybrides Verfahren zur automatischen Selektion oder Kombination von M_{sel} geeigneten Prognoseverfahren aus einer Menge \mathbf{A} von M Verfahrenskandidaten. Im Gegensatz zu gängigen Selektionsverfahren aus dem Stand der Forschung, für die im Allgemeinen stets im Voraus festgelegt wird, ob eine Selektion eines Prognosemodells ($M_{\text{sel}} = 1$) oder eine Kombination mehrerer Prognosemodelle ($M_{\text{sel}} \geq 2$) durchgeführt werden soll, trifft das Selbinationsverfahren diese Entscheidung automatisch in Abhängigkeit von den Charakteristika vorliegender Daten ($1 \leq M_{\text{sel}} \leq M$). Dadurch werden die Vorteile von Selektions- und Kombinationsansätzen in einem Verfahren vereint. Das Selbinationsverfahren besteht aus den beiden Hauptkomponenten eines Meta-Modells und einer M_{sel} -fachen Selektionsfunktion, welche auf einer sogenannten Hybriditätsgrenze beruht.

Das Meta-Modell des Selbinationsverfahrens ist ein Ensemble aus 10 Multilayer-Perzeptronen, die als Klassifikationsmodelle basierend auf zufälligen Unterteilungen einer Meta-Trainingsmenge in eine kleinere Meta-Trainings- und eine Meta-Validierungsmenge trainiert wurden. Die Meta-Mengen bestehen dabei jeweils aus Meta-Merkmalen, die gegebene Zeitreihen beschreiben, und Meta-Labels, die Rangfolgen der Prognoseverfahrenskandidaten für die Zeitreihen widerspiegeln. Zur Bestimmung der Meta-Merkmale werden Merkmalsextraktionsfunktionen auf die gegebenen Zeitreihen angewandt. Neben gängigen Merkmalen, die zuvor im Stand der Forschung für Meta-Lernansätze verwendet wurden, werden in dieser Arbeit zwei neue Mengen von Landmarkers eingeführt, die die Verhältnisse und Differenzen der Validierungsfehler der Verfahrenskandidaten darstellen. Im Rahmen einer Merkmalsselektion zeigte sich, dass diese neu eingeführten Merkmale besser zur Beschreibung eines Prognosemodellselektionsproblems geeignet sind, als viele gängige Merkmale und die Validierungsfehler der Verfahrenskandidaten selbst. Durch die Merkmalsselektion wurde eine Teilmenge \mathbf{F} von 25 der ursprünglich 89 Merkmalsextraktionsfunktionen ausgewählt, mit der die beste Konfiguration des Selbinationsverfahrens erreicht wurde. Zur Bestimmung der Meta-Labels wurden alle Prognoseverfahrenskandidaten für die Zeitreihen in der Meta-Trainingsmenge verwendet, um Prognosemodelle zu trainieren, die danach zur Bestimmung von Prognosen zukünftiger Werte der Zeitreihen angewandt wurden. Nach einer darauffolgenden Evaluation der Prognosegüten hinsichtlich der RO-sMAPEs wurden Rangfolgen der Prognose-

verfahren für alle Zeitreihen gebildet, aus denen die Meta-Labels bestimmt wurden. Basierend auf der Meta-Trainingsmenge aus Meta-Merkmalen und Meta-Labels wurde das Meta-Modell trainiert. Dafür wurde der Datensatz \mathbf{Y} der 808 monatlichen Zeitreihen der M3-Competition verwendet. Aus diesem Datensatz wurde die Teilmenge der 111 Zeitreihen der NN3-Competition herausgenommen und für die Meta-Testmenge der Evaluationsstudien der folgenden Kapitel vorgehalten. Anhand der verbleibenden 697 Zeitreihen wurden die Meta-Trainings- und Meta-Validierungsmengen bestimmt, die jeweils aus einer Merkmalsmatrix und einer Label-Matrix bestehen.

Nach dem Training des Meta-Modells, lässt sich dieses verwenden, um basierend auf den berechneten Meta-Merkmalen für neue Zeitreihen die Eignungswahrscheinlichkeiten der Prognoseverfahrenskandidaten zu bestimmen. Um anhand dieser Wahrscheinlichkeiten ein oder mehrere Prognoseverfahren zu selektieren, wird eine M_{sel} -fache Selektionsfunktion verwendet, die auf einer so genannten Hybriditätsgrenze beruht, für die ein Wert zwischen 0 und 1 gewählt werden kann. Die Hybriditätsgrenze ist ein Hyperparameter des Selbinationsverfahrens, der den Wert bestimmt, den die kumulative Summe der Eignungswahrscheinlichkeiten der zu selektierenden Prognoseverfahren mindestens ergeben muss. Für die Wahl eines kleinen Wertes der Hybriditätsgrenze werden tendenziell wenige Prognoseverfahren selektiert, während große Werte die Selektion mehrerer Prognoseverfahren begünstigen. Eine Konfiguration $\Omega_{\text{min}} = 0$ führt zu einer Selektionsfunktion für das einzelne Prognoseverfahren mit der höchsten Eignungswahrscheinlichkeit. Eine Konfiguration $\Omega_{\text{min}} = 1$ führt zu einer Kombination aller Prognoseverfahrenskandidaten. Für eine Konfiguration zwischen diesen beiden Grenzen ergibt sich eine M_{sel} -fache Selektionsfunktion, die datengetrieben ein oder mehrere Prognoseverfahren selektiert. Nach der Selektion der Prognoseverfahren werden diese verwendet, um Prognosemodelle zu trainieren, welche darauffolgend zur Prognose zukünftiger Zeitreihenwerte verwendet werden. Falls mehrere Prognosemodelle trainiert wurden, werden die Prognosen der verschiedenen Modelle zu einer finalen Prognose kombiniert.

In dieser Arbeit werden acht verschiedene Prognoseverfahrenskandidaten genutzt. Eine Evaluation verschiedener Konfigurationen des Selbinationsverfahrens hinsichtlich der Menge an Merkmalsextraktionsfunktionen \mathbf{F} , der Menge an Prognoseverfahrenskandidaten \mathbf{A} und der Hybriditätsgrenze Ω_{min} in Abschnitt 6.4 wird zeigen, dass das Selbinationsverfahren robust gegenüber Änderungen dieser Konfigurationen ist, da alle verschiedenen Konfigurationen ähnliche Ergebnisse erzielen. Für die Evaluationsstudien der beiden nächsten Kapitel wird die Konfiguration des Selbinationsverfahrens genutzt, die die besten Ergebnisse in der Robustheitsstudie erzielt. Dies ist eine Konfiguration mit den 25 Merkmalsextraktionsfunktionen in Tabelle 5.2, den vier Prognoseverfahrenskandidaten `llknn`, `ets`, `nnetar` und `srw` sowie der Hybriditätsgrenze $\Omega_{\text{min}} = 0,5$.

Kapitel 6

Evaluation des Selbinationsverfahrens hinsichtlich Prognosegüte und Rechenaufwand

„There is no reason anyone would want a computer in their home.“

Kenneth H. Olson, US-amerikanischer Ingenieur (1926 - 2011)

Dieses Kapitel evaluiert das neuartige Selbinationsverfahren hinsichtlich der Prognosegüte und des Rechenaufwands in empirischen Vergleichsstudien gegen gängige Benchmarks aus dem Stand der Forschung. Hierzu wird die im letzten Kapitel bestimmte Parameterkonfiguration bezüglich der Menge an Prognoseverfahrenskandidaten, der Menge an Merkmalsextraktionsfunktionen und der Hybriditätsgrenze genutzt. Die Ergebnisse der empirischen Studien führen zur Bestätigung der Forschungsthese 2 und 3. Des Weiteren enthält Abschnitt 6.4 einen Vergleich verschiedener Konfigurationen des Selbinationsverfahrens, um die Robustheit des Verfahrens zu evaluieren.

6.1 Experimentbeschreibung

6.1.1 Prognoseverfahren

Wie im letzten Kapitel erläutert wurde, müssen für eine eindeutige Konfiguration des Selbinationsverfahrens die Menge der Prognoseverfahrenskandidaten \mathbf{A} , die Menge der Merkmalsextraktionsfunktionen \mathbf{F} und die Hybriditätsgrenze Ω_{\min} festgelegt werden. Als Prognoseverfahrenskandidaten stehen in dieser Arbeit die acht in Abschnitt 3.4 beschriebenen Verfahren zur Verfügung: die beiden lokalen k -nächste-Nachbarn-Prognoseverfahren `lcknn` und `llknn`, die beiden gängigen statistischen Prognoseverfahren `auto.arima` und `ets`, die beiden Verfahren zum Training neuronaler Netze `nnetar` und `nnfor` sowie die beiden Random Walks `rw` und `srw`. Für die Vergleichsstudien gegen Benchmarks in den Abschnitten 6.2 und 6.3 wird die Teilmenge der vier Prognoseverfahrenskandidaten `llknn`, `ets`, `nnetar` und `srw` gewählt. In dieser Konfiguration

ist je ein Verfahren aus den vier verschiedenen Klassen enthalten. Wie der Parametervergleich in Abschnitt 6.4 zeigen wird, ist das Selbinationsverfahren robust und ermittelt ähnlich gute Prognoseergebnisse, für verschiedene Konfigurationen hinsichtlich der Menge an Prognoseverfahrenskandidaten. Allerdings werden für die ausgewählte Menge der vier Verfahrenskandidaten die besten Ergebnisse erzielt. Für diese Konfiguration wurde in Abschnitt 5.2.5 eine Merkmalsselektion durchgeführt, in der die optimale Teilmenge der 25 Merkmale in Tabelle 5.2 ermittelt wurde. Zudem wurde in Abschnitt 5.4 als sinnvolle Hybriditätsgrenze $\Omega_{\min} = 0,5$ bestimmt. Daher wird diese Konfiguration des Selbinationsverfahrens in den Evaluationsstudien der nächsten beiden Abschnitte verwendet und als Selbination bezeichnet:

$$\begin{aligned} \mathbf{A} &= \{\text{llknn, ets, nnetar, srw}\} \\ \mathbf{F} &= \text{Menge aller Merkmalsextraktionsfunktionen in Tabelle 5.2} \\ \Omega_{\min} &= 0,5. \end{aligned} \tag{6.1}$$

In den Evaluationsstudien wird Selbination hinsichtlich der Prognosegüte und des Rechenaufwands gegen neun verschiedene Benchmark-Verfahren verglichen, die sowohl aggregierte und individuelle Selektionsverfahren mit festgelegten oder mehreren möglichen Modellklassen sowie Kombinations- und Pooling-Verfahren umfassen. Vier dieser Verfahren sind die vier Prognoseverfahrenskandidaten, die Selbination nutzt. srw ist das aggregierte Selektionsmodell eines saisonalen Random Walks (vgl. Abschnitt 3.4.1). ets und llknn sind individuelle Selektionsverfahren mit den festgelegten Modellklassen der exponentiellen Glättungsmodelle (vgl. Abschnitt 3.4.3) und der lokalen k -nächste-Nachbarn-Modelle (vgl. Abschnitt 3.4.4). nnetar ist ein individuelles Kombinationsverfahren, welches ein Ensemble aus 20 neuronalen Netzen bildet (vgl. Abschnitt 3.4.5).

Neben diesen ersten vier Benchmark-Verfahren, welche Selektionen oder Kombinationen für festgelegte Modellklassen durchführen, verwenden die weiteren fünf Benchmark-Verfahren jeweils mehrere Modellklassen. Die Verfahren Val-Sel, Pool-2 und Pool-3 bilden zunächst für jede Zeitreihe erwartete Rangfolgen der vier Kandidaten srw, ets, llknn und nnetar anhand eines Vergleichs der mittleren Validierungsfehler hinsichtlich des RO-sMAPE und verwenden danach verschiedene Selektionsfunktionen (vgl. Abschnitt 4.3.2). Val-Sel selektiert das Verfahren mit dem besten erwarteten Rang für eine Zeitreihe. Dieses Verfahren ist sowohl in der Wissenschaft als auch in der Industrie ein häufig genutzter Benchmark. Pool-2 und Pool-3 selektieren die zwei, bzw. drei Verfahren mit den besten erwarteten Rängen für eine Zeitreihe. Zur Ermittlung der Validierungsfehler wird jede gegebene Trainingsmenge in eine kleinere Trainings- und eine Validierungsmenge unterteilt, sodass die Validierungsmenge aus den letzten $H = 18$ Werten besteht. Die Verfahrenskandidaten werden anhand der kleineren Trainingsmenge verwendet, um Modelle zu trainieren, welche darauffolgend genutzt werden, um die Werte der Validierungsmenge zu prognostizieren. Die Validierungsfehler der Verfahrenskandidaten werden als $\text{RO-sMAPE}_{T_{\text{train}}, T_{\text{train}+6}}^{1,12}$ bestimmt (vgl. Abschnitt 3.2.2) und die erwarteten Rangfolgen werden anhand der Vergleiche der Validierungsfehler gebildet (vgl. Abschnitt 4.3.2).

Zusätzlich zu der Konfiguration des Selbinationsverfahrens mit der Hybriditätsgrenze $\Omega_{\min} = 0,5$ wird eine weitere Konfiguration als Benchmark verwendet, welche als ML-Sel bezeichnet wird. Dies ist die Konfiguration des Selbinationsverfahrens, die aus der gleichen Menge der vier Prognoseverfahrenskandidaten basierend auf der gleichen Menge an 25 Merkmalsextraktionsfunktionen die Hybriditätsgrenze $\Omega_{\min} = 0$ verwendet und somit eine individuelle Selektion des einzelnen Kandidaten durchführt, für den das Meta-Modell die höchste Eignungswahrscheinlichkeit ermittelt. Das letzte Benchmark-Verfahren, welches als Comb-4 bezeichnet wird, ist eine gleichgewichtete Kombination der vier Verfahren *srw*, *ets*, *llknn* und *nnetar* für jede Zeitreihe. Es sei angemerkt, dass eine Verwendung des Selbinationsverfahrens mit einer Hybriditätsgrenze $\Omega_{\min} = 1$ zu den gleichen Prognosen führt, wie Comb-4. Zudem würde eine Anwendung des Forecast-Islands-Ansatzes von Kourentzes et al. (2019) mit den vier Verfahrenskandidaten *srw*, *ets*, *llknn* und *nnetar* ebenfalls zu den gleichen Prognosen führen, wie Comb-4. Wie in Abschnitt 4.3.2 beschrieben wurde, selektiert der Forecast-Islands-Ansatz alle Verfahren bis zum ersten Ausreißer hinsichtlich der Prognosegüte anhand der gängigen Definitionen für Boxplots und Quantile (vgl. Abbildung 5.5). In diesem Zusammenhang kann eine Stichprobe erst ab einer Mindestanzahl von fünf Werten einen Ausreißer enthalten. Somit selektiert der Forecast-Islands-Ansatz immer mindestens vier Modelle. Des Weiteren sei angemerkt, dass alle Prognosewerte der Evaluationsstudien auf positive ganze Zahlen gerundet werden, da eine Zeitreihe der Kundennachfrage nur positive ganzzahlige Werte enthalten kann.

Nach der Evaluation des Selbinationsverfahrens im Vergleich zu den beschriebenen Benchmarkverfahren, wird am Ende von Abschnitt 6.2 ein Vergleich zu den Teilnehmern und Benchmarks der NN3-Competition sowie zu den Teilnehmern der M3-Competition durchgeführt. Hierfür wurden die durchschnittlichen Prognosefehler der Teilnehmer und Benchmarks der NN3-Competition aus Crone et al. (2011) entnommen. Die Prognosen der Teilnehmer der M3-Competition wurden aus Hyndman et al. (2018a) entnommen und die durchschnittlichen Prognosefehler wurden selbst berechnet.

Zusätzlich zu der durch (6.1) beschriebenen Konfiguration des Selbinationsverfahrens werden in Abschnitt 6.4 verschiedene Parameterkonfigurationen hinsichtlich der Menge an Verfahrenskandidaten \mathbf{A} , der Menge der Merkmalsextraktionsfunktionen \mathbf{F} und der Hybriditätsgrenze Ω_{\min} in einer Robustheitsevaluation verglichen. Neben den vier Kandidaten in (6.1) werden hierbei auch die Verfahren *lcknn*, *auto.arima*, *nnfor* und *rw* verwendet und es werden verschiedene Mengen an Kandidaten verglichen.

6.1.2 Datensatz

In den Evaluationsstudien dieses Kapitels wird der gleiche Datensatz wie in Abschnitt 5.4 verwendet. Die Teilmenge \mathbf{Y} der 808 monatlichen Zeitreihen der Kategorien Industrie und Mikroökonomie der M3-Competition (vgl. Abschnitt 3.3 sowie Makridakis und Hibon, 2000) wird mit den gleichen Unterteilungen in Trainingsmengen $\mathbf{y}_{\text{train}}^z = \{y_1^z, y_2^z, \dots, y_{T_z}^z\}$ und Testmengen

$\mathbf{y}_{\text{test}}^z = \{y_{T_z+1}^z, y_{T_z+2}^z, \dots, y_{T_z+H}^z\}$ verwendet, wie in der Competition, sodass jede Testmenge aus $H = 18$ Werten besteht. Es werden die gleichen Unterteilungen in Meta-Trainings-, Meta-Validierungs- und Meta-Testmenge verwendet, wie in Abschnitt 5.4. Somit wird die Prognosegüte von Selbination sowie den Benchmark-Verfahren anhand der Meta-Testmenge evaluiert, welche aus den 111 Zeitreihen der NN3-Competition besteht (vgl. Crone et al., 2011). Wie in Abschnitt 3.3 beschrieben wurde, ist dies ein etablierter Benchmark-Datensatz, der in zahlreichen empirischen Studien zum Vergleich der Prognosegüte verschiedener Prognoseverfahren verwendet wurde. Er besteht aus einer repräsentativen Menge von 51 kurzen und 60 langen monatlichen industriellen Zeitreihen, von denen einige saisonale und andere nicht-saisonale Entwicklungen zeigen. Das balancierte Design des Datensatzes ermöglicht eine Analyse der Selektions- und Prognosegüte des Selbinationsverfahrens über verschiedene typische industrielle Datenbedingungen.

6.1.3 Evaluationskriterien

Das Selbinationsverfahren wird hinsichtlich der Prognosegüte und des Rechenaufwands evaluiert. Die Prognosegüte wird als durchschnittlicher symmetrischer mittlerer absoluter prozentualer Fehler der rollierenden Prognosen (Mean-RO-sMAPE) in diversen Experimenten mit verschiedenen Prognosehorizonten von unterschiedlichen Ursprüngen bestimmt. Zudem werden Friedman- und Nemenyi-Tests durchgeführt, um zu testen, ob die Unterschiede der Prognosegüten der verschiedenen Prognoseverfahren statistisch signifikant sind. Der Rechenaufwand von Selbination wird hinsichtlich des Aufwands pro Training eines Prognosemodells sowie hinsichtlich des Aufwands pro Prognose evaluiert.

Neben der Evaluation hinsichtlich des RO-sMAPE für $h_{\min} = 1$, $h_{\max} = H = 12$, $o_{\min} = T_z$ und $o_{\max} = T_z + 6$, dem Fehlermaß, das zur Bestimmung einer sinnvollen Hybriditätsgrenze in Abschnitt 5.4 verwendet wurde, werden in diesem Kapitel diverse weitere Experimentkonfigurationen bezüglich der Ursprünge und Prognosehorizonte durchgeführt, um eine umfassende Evaluation der Prognosegüte des Selbinationsverfahrens zu erzielen. Des Weiteren werden alle Experimente für die Teilmengen der 51 kurzen und 60 langen Zeitreihen sowie für den kompletten Datensatz der 111 Zeitreihen der NN3-Competition durchgeführt. Die Prognosegüten der verschiedenen Prognoseverfahren werden anhand der Mean-RO-sMAPEs verglichen (vgl. (3.34) in Abschnitt 3.2.3).

Zusätzlich zur Evaluation hinsichtlich der Mean-RO-sMAPEs werden in diesem Kapitel statistische Tests zur Ermittlung der Signifikanz der Unterschiede der Prognosegüten durchgeführt. Für jede Zeitreihe der NN3-Competition werden durch Verwendung von (3.40) in Abschnitt 3.2.3 Rangfolgen bezüglich des $\text{RO-sMAPE}_{T_z, T_z+6}^{1,12}$ gebildet und darauffolgend werden die mittleren Ränge aller Verfahren durch (3.41) ermittelt. Basierend darauf werden Friedman- und Nemenyi-Tests durchgeführt (vgl. Abschnitt 3.2.3).

Abgesehen von einer hohen Prognosegüte ist es von Vorteil, wenn ein Prognoseverfahren geringe Rechenaufwände für das Training eines Prognosemodells sowie für die Bestimmung von

Prognosen benötigt. Daher werden die mittleren Zeiten eines Prognoseverfahrens pro Training eines Modells und pro Prognose betrachtet. Einige der verglichenen Verfahren, wie z.B. Selbination, führen vorbereitende Schritte durch, bevor ein finales Prognosemodell trainiert wird. Daher setzt sich der Rechenaufwand für das Training eines Prognosemodells aus allen Schritten zusammen, die notwendig sind, um ein finales Prognosemodell zu trainieren. Beispielsweise umfasst der Rechenaufwand für das Training von einem oder mehreren Prognosemodellen durch das Selbinationsverfahren die folgenden Schritte: (i) Berechnung der Meta-Merkmale durch Verwendung der Merkmalsextraktionsfunktionen in \mathbf{F} , (ii) Transformation der Merkmale in das richtige Format, (iii) Anwendung der 10 neuronalen Netze des Meta-Modells, (iv) Kombination der Output-Vektoren der neuronalen Netze zu einem Ensemble-Output, welcher die Eignungswahrscheinlichkeiten der Verfahrenskandidaten widerspiegelt, (v) Selektion eines oder mehrerer Prognoseverfahren basierend auf den Eignungswahrscheinlichkeiten und der Hybriditätsgrenze sowie (vi) Verwendung der selektierten Prognoseverfahren, um Prognosemodelle zu trainieren. Der Rechenaufwand pro Prognose setzt sich aus den Zeiten pro Prognose aller trainierten Prognosemodelle sowie einer möglichen Zeit für die Kombination der Prognosen aller verwendeten Modelle zusammen. Für jedes Prognoseverfahren muss das Training eines oder mehrerer Prognosemodelle einmalig pro Zeitreihe durchgeführt werden und danach können die trainierten Modelle verwendet werden, um Prognosen für verschiedene Horizonte von unterschiedlichen Ursprüngen zu berechnen. Im Gegensatz dazu gibt der Rechenaufwand pro Prognose die Rechenzeit an, die für die Prognose eines Wertes benötigt wird. Daher wird diese Rechenzeit beispielsweise für jede Zeitreihe 18-mal für eine Evaluation mit fixiertem Ursprung mit $h_{\min} = 1$ und $h_{\max} = 18$ benötigt sowie 84-mal für eine Evaluation anhand rollierender Prognosen der Horizonte $h_{\min} = 1$ und $h_{\max} = 12$ für die Ursprünge $o_{\min} = T_z$ bis $o_{\max} = T_z + 6$. Zusätzlich zu den mittleren Trainingszeiten und den mittleren Zeiten pro Prognose werden ebenfalls Rangfolgen für alle Zeitreihen gebildet sowie Friedman- und Nemenyi-Tests durchgeführt, um auf signifikante Unterschiede hinsichtlich der Rechenaufwände zu testen. Alle Experimente dieser Arbeit wurden mit Hilfe eines Standard-Windows-Computers mit einer Intel(R) Core(TM) i5-4300U CPU @1.90GHz und 8GB RAM durchgeführt. Das Selbinationsverfahren wurde prototypisch in `Matlab` implementiert. Für eine professionelle Anwendung in einem industriellen Kontext würde eine Implementierung in einer höheren Programmiersprache das Potenzial bieten, die Rechenaufwände zu reduzieren. Allerdings liegt eine professionelle Implementierung außerhalb des Rahmens dieser Arbeit. Es sei angemerkt, dass der Rechenaufwand eines Prognoseverfahrens für eine einzelne Zeitreihe nicht deterministisch ist, da dieser von allen gleichzeitig laufenden Prozessen auf dem Computer abhängt und sich somit leicht ändert, wenn das gleiche Experiment mehrfach durchgeführt wird. Allerdings lässt sich durch die Bildung der mittleren Rechenzeiten über alle Prognosen, alle Trainingszeiten und alle Zeitreihen annehmen, dass der durchschnittliche Rechenaufwand ein robustes Maß des theoretischen Rechenaufwands der Verfahren ist und, dass verschiedene durchschnittliche Rechenaufwände verglichen werden können, um Prognoseverfahren zu evaluieren.

6.2 Evaluation hinsichtlich der Prognosegüte

6.2.1 Evaluation rollierender Prognosen für verschiedene Ursprünge und Horizonte

Tabelle 6.1 zeigt die durchschnittlichen symmetrischen mittleren prozentualen Fehler (Mean-RO-sMAPEs) der in Abschnitt 6.1 definierten Konfiguration des Selbinationsverfahrens (Selbination) im Vergleich zu den neun beschriebenen Benchmark-Verfahren. Es wurden verschiedene Experimente hinsichtlich der Ursprünge (o_{\max}), Horizonte (h_{\min} und h_{\max}) sowie der Zeitreihenlängen (kurz, lang, alle) durchgeführt. Alle Evaluationen beginnen am letzten Punkt der Trainingsmenge T_z einer Zeitreihe, somit gilt stets $o_{\min} = T_z$. Zusätzlich zu den detaillierten Ergebnissen in der Tabelle visualisiert Abbildung 6.1 die durchschnittlichen Mean-RO-sMAPEs über alle Experimente in Tabelle 6.1. Des Weiteren zeigt Abbildung 6.2 die Ergebnisse der Friedman- und Nemenyi-Tests mit Signifikanzniveau 0.05. Wie in Abschnitt 3.2.3 beschrieben wurde, ist oben auf der vertikalen Achse das Prognoseverfahren mit dem niedrigsten mittleren Rang und somit der besten mittleren Prognosegüte zu sehen. Von oben nach unten sinkt die Prognosegüte der Verfahren. Die Punkte auf der horizontalen Achse zeigen die mittleren Ränge der Prognoseverfahren und die Prognosegüten zweier Verfahren werden als signifikant unterschiedlich hinsichtlich des Signifikanzniveaus betrachtet, wenn die Differenz ihrer mittleren Ränge die kritische Distanz (CD) übersteigt. Wenn die Prognosegüte eines Verfahrens sich nicht signifikant von den Prognosegüten anderer Verfahren unterscheidet, werden die mittleren Ränge durch eine horizontale Linie verbunden. Drei verschiedene Friedman-Tests wurden für die Mengen der kurzen, der langen und aller Zeitreihen der NN3-Competition durchgeführt. In allen drei Fällen wurde die Nullhypothese verworfen, was auf Unterschiede der Prognosegüten der verschiedenen Verfahren hindeutet. Die drei Visualisierungen in Abbildung 6.2 zeigen die Ergebnisse der Nemenyi-Tests, die größtenteils mit den Evaluationsergebnissen hinsichtlich der Mean-RO-sMAPEs in Tabelle 6.1 übereinstimmen.

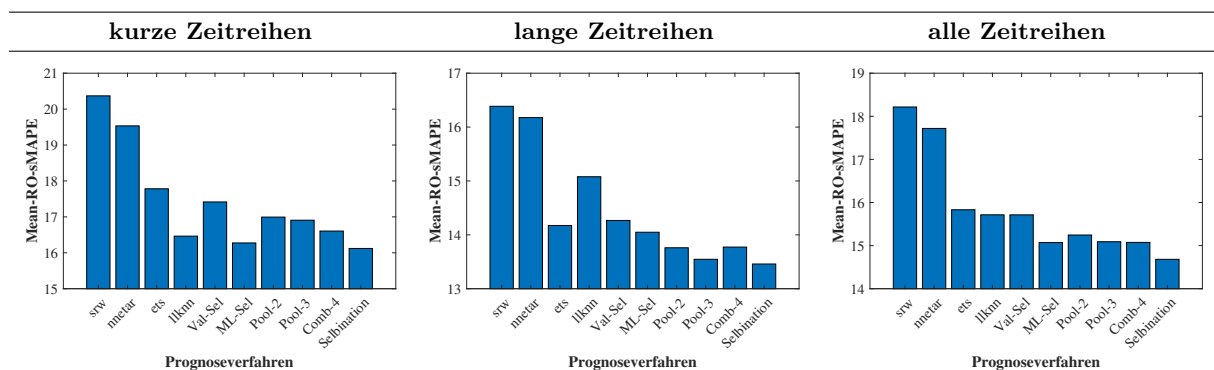


Abbildung 6.1: Balkendiagramme der durchschnittlichen Mean-RO-sMAPEs der verschiedenen Prognoseverfahren für kurze, lange und alle Zeitreihen über alle Experimente

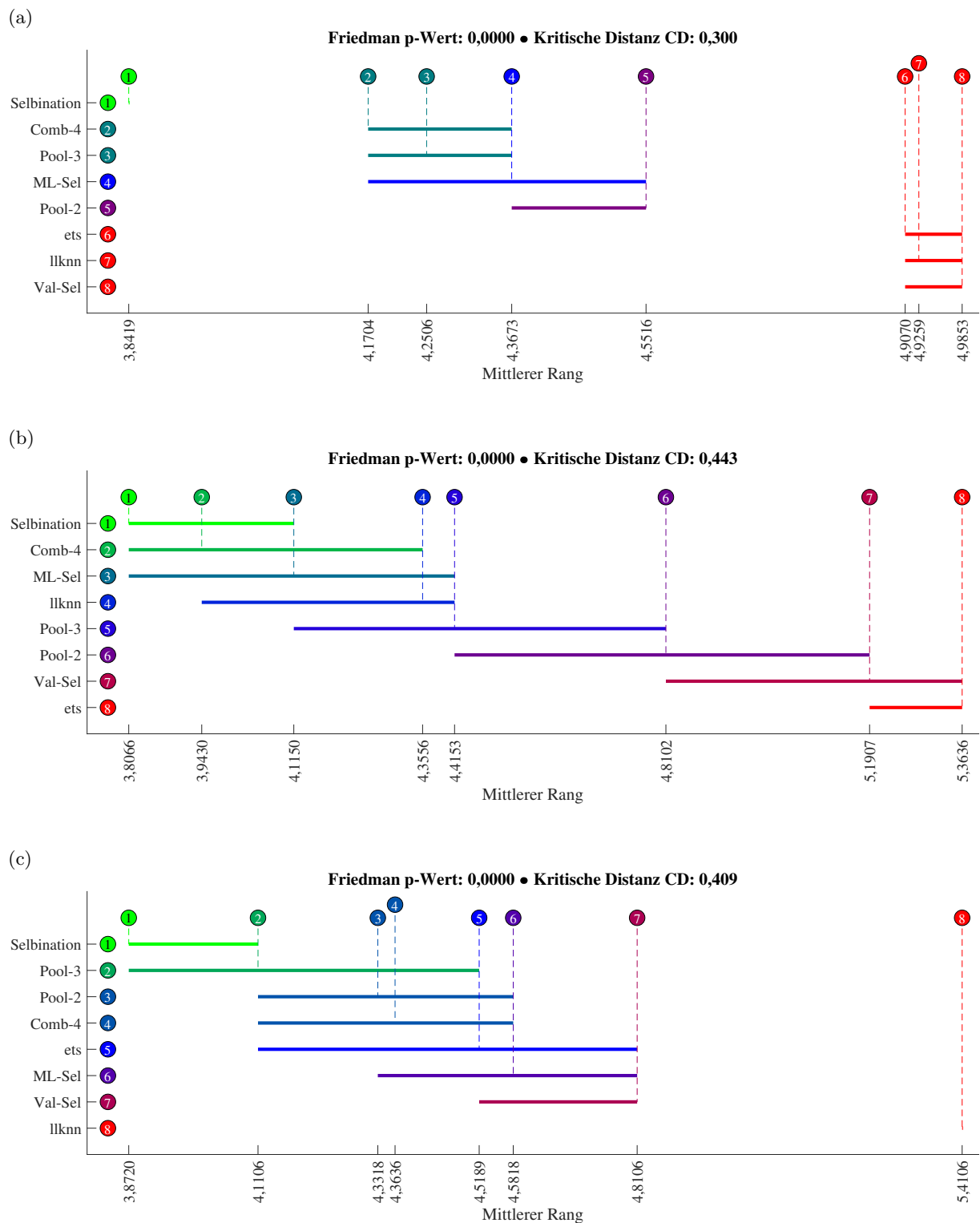


Abbildung 6.2: Nemenyi-Test-Ergebnisse hinsichtlich der mittleren Fehler der rollierenden Prognosen der verschiedenen Prognoseverfahren für (a) alle, (b) kurze und (c) lange Zeitreihen gemittelt über alle durchgeführten Experimente für Signifikanzniveaus von 0,05

Tabelle 6.1: Durchschnittliche mittlere Prognosefehler rollierender Prognosen und Ränge des Selbinationsverfahrens sowie gängiger Benchmark-Verfahren für verschiedene Ursprünge, Horizonte und Zeitreihenlängen anhand der Zeitreihen der NN3-Competition (die jeweils besten Verfahren pro Experiment sind fettgedruckt und markiert)

Experimentkonfiguration				Mean-RO-sMAPE									
o_{\max}	h_{\min}	h_{\max}	Menge	srw	nnetar	ets	llknn	Val-Sel	ML-Sel	Pool-2	Pool-3	Comb-4	Selbination
T+17	1	1	kurz	19,75 9	20,34 10	16,85 8	15,27 3	16,81 7	15,11 2	16,58 6	16,29 5	16,05 4	14,92 1*
T+17	1	1	lang	15,80 10	13,77 9	10,57 1*	12,33 8	11,96 7	11,03 3	11,03 4	11,10 5	11,48 6	10,79 2
T+17	1	1	alle	17,61 10	16,79 9	13,45 3	13,68 7	14,19 8	12,90 2	13,58 5	13,48 4	13,58 6	12,69 1*
T+15	1	3	kurz	19,45 10	19,17 9	16,73 8	15,29 3	16,54 7	15,08 2	16,18 6	16,03 5	15,73 4	14,87 1*
T+15	1	3	lang	15,88 10	14,57 9	11,53 2	13,18 8	12,30 7	11,81 5	11,61 3	11,73 4	12,08 6	11,41 1*
T+15	1	3	alle	17,52 10	16,68 9	13,92 6	14,15 7	14,25 8	13,31 2	13,71 4	13,71 3	13,76 5	13,00 1*
T+12	4	6	kurz	18,90 10	18,69 9	16,87 8	15,52 3	16,38 7	15,22 2	16,04 6	15,98 5	15,57 4	15,09 1*
T+12	4	6	lang	16,25 10	16,14 9	13,87 7	14,76 8	13,37 4	13,57 6	13,04 3	13,03 2	13,46 5	12,91 1*
T+12	4	6	alle	17,47 10	17,31 9	15,25 8	15,11 7	14,75 6	14,33 2	14,42 4	14,38 3	14,43 5	13,91 1*
T+9	7	9	kurz	18,81 10	18,54 9	17,39 8	16,05 4	16,85 7	15,71 2	16,45 6	16,30 5	15,84 3	15,62 1*
T+9	7	9	lang	16,26 10	16,22 9	13,89 7	14,99 8	13,80 6	13,67 4	13,40 2	13,48 3	13,75 5	13,16 1*
T+9	7	9	alle	17,43 10	17,28 9	15,49 8	15,48 7	15,21 6	14,60 2	14,80 5	14,77 4	14,71 3	14,29 1*
T+6	10	12	kurz	19,43 10	19,10 9	18,30 8	17,06 4	17,56 7	16,94 3	17,23 6	17,07 5	16,60 1*	16,82 2
T+6	10	12	lang	16,26 9	16,67 10	14,73 6	15,60 8	14,97 7	14,50 5	14,13 2	14,09 1*	14,41 4	14,18 3
T+6	10	12	alle	17,71 9	17,79 10	16,37 8	16,27 7	16,16 6	15,62 5	15,55 4	15,46 3	15,42 2	15,39 1*
T+3	13	15	kurz	24,31 10	21,34 9	19,15 7	18,60 3	19,52 8	18,52 2	18,95 4	19,10 6	18,98 5	18,25 1*
T+3	13	15	lang	16,93 9	17,91 10	15,79 6	16,41 8	16,20 7	15,40 4	15,54 5	14,66 2	14,65 1*	15,04 3
T+3	13	15	alle	20,32 10	19,48 9	17,33 6	17,41 7	17,73 8	16,84 4	17,11 5	16,70 3	16,64 2	16,51 1*
T	16	18	kurz	25,22 10	21,99 9	21,22 8	19,67 3	21,14 7	19,65 2	19,96 5	20,07 6	19,93 4	19,51 1*
T	16	18	lang	18,34 6	18,51 7	19,42 10	17,92 4	18,56 8	18,94 9	18,05 5	16,89 2	16,23 1*	17,18 3
T	16	18	alle	21,50 10	20,11 8	20,25 9	18,72 4	19,75 7	19,27 6	18,93 5	18,35 3	17,93 1*	18,25 2
T+12	1	6	kurz	19,11 10	18,82 9	16,68 8	15,25 3	16,24 7	15,00 2	15,95 6	15,86 5	15,52 4	14,83 1*
T+12	1	6	lang	16,03 10	15,33 9	12,69 4	14,23 8	12,78 6	12,74 5	12,44 2	12,49 3	12,97 7	12,27 1*
T+12	1	6	alle	17,45 10	16,93 9	14,53 7	14,70 8	14,37 6	13,78 2	14,05 4	14,04 3	14,14 5	13,45 1*
T+6	7	12	kurz	18,87 10	18,56 9	17,54 8	16,31 4	16,82 7	16,02 3	16,49 6	16,39 5	15,93 1*	15,95 2
T+6	7	12	lang	16,28 9	16,70 10	14,64 7	15,77 8	14,63 6	14,45 4	14,22 3	14,21 2	14,50 5	14,09 1*
T+6	7	12	alle	17,47 9	17,55 10	15,97 7	16,02 8	15,64 6	15,17 3	15,27 5	15,21 4	15,16 2	14,94 1*
T+6	1	12	kurz	19,05 10	18,63 9	16,86 8	15,50 3	16,30 7	15,26 2	16,04 6	15,91 5	15,60 4	15,12 1*
T+6	1	12	lang	16,07 10	15,81 9	14,04 7	15,15 8	13,89 6	13,88 5	13,63 3	13,51 2	13,88 4	13,46 1*
T+6	1	12	alle	17,44 10	17,11 9	15,34 8	15,31 7	14,99 6	14,51 2	14,74 5	14,61 3	14,67 4	14,22 1*
T	1	18	kurz	21,16 10	19,70 9	18,02 8	16,58 3	17,43 7	16,52 2	17,07 6	16,98 5	16,91 4	16,33 1*
T	1	18	lang	16,16 9	16,32 10	14,74 7	15,51 8	14,47 5	14,54 6	14,28 4	13,82 2	14,10 3	13,55 1*
T	1	18	alle	18,46 10	17,87 9	16,25 8	16,00 7	15,83 6	15,45 4	15,56 5	15,27 2	15,39 3	14,83 1*
alle Experimente			kurz	20,37 10	19,53 9	17,78 8	16,46 3	17,42 7	16,27 2	16,99 6	16,91 5	16,61 4	16,12 1*
alle Experimente			lang	16,39 10	16,18 9	14,17 6	15,08 8	14,27 7	14,05 5	13,76 3	13,55 2	13,77 4	13,46 1*
alle Experimente			alle	18,22 10	17,72 9	15,83 8	15,71 7	15,71 6	15,07 2	15,25 5	15,09 4	15,07 3	14,68 1*

Abbildung 6.1 zeigt, dass Selbination die niedrigsten durchschnittlichen Fehler gemittelt über alle Experimente erzielt. Dies gilt sowohl für die Menge der kurzen als auch für die Menge der langen und die Menge aller betrachteten Zeitreihen. Hinsichtlich des Nemenyi-Tests für alle Zeitreihen (Abbildung 6.2(a)) erreicht Selbination eine signifikant höhere Prognosegüte als alle Benchmark-Verfahren. Neben diesem Hauptergebnis lässt sich feststellen, dass das Verfahren ML-Sel, welches für jede Zeitreihe eine individuelle Selektion des Prognoseverfahrenskandidaten durchführt, für den das Meta-Modell die höchste Eignungswahrscheinlichkeit ermittelt, eine äquivalente Prognosegüte zu der Kombination aller vier Prognoseverfahrenskandidaten (Comb-4) sowie den beiden Pooling-Verfahren Pool-3 und Pool-2 erzielt, welche die drei bzw. zwei Prognoseverfahrenskandidaten mit den niedrigsten Validierungsfehlern pro Zeitreihe kombinieren. Somit besteht kein signifikanter Unterschied der Prognosegüten zwischen dem Selektionsverfahren ML-Sel und den betrachteten Kombinations- und Pooling-Verfahren, allerdings erzielt Selbination

eine signifikant höhere Prognosegüte als alle Selektions-, Kombinations- und Pooling-Verfahren. Durch dieses Ergebnis wird FT 3 bestätigt.

Das Verfahren Val-Sel, welches eine individuelle Selektion des Prognoseverfahrenskandidaten mit dem niedrigsten Validierungsfehler pro Zeitreihe durchführt, führt zu einer signifikant schlechteren Prognosegüte als ML-Sel, Comb-4, Pool-3 sowie Pool-2 und es erreicht äquivalente Ergebnisse zu den individuellen Selektionsverfahren mit festgelegter Modellklasse llknn und ets. Dies zeigt, dass Meta-Lernansätze ein besseres Selektionskriterium darstellen als Vergleiche von Prognosefehlern. Somit wird FT 2 bestätigt.

Tabelle 6.2 zeigt, wie häufig Selbination und die anderen Selektionsverfahren mit mehreren möglichen Modellklassen die vier verschiedenen Prognoseverfahrenskandidaten für die Meta-Testmenge selektieren. Selbination selektiert im Durchschnitt 1,77 Verfahren pro Zeitreihe. Es selektiert 87-mal llknn, 65-mal ets, 21-mal nnetar und 24-mal srw. ML-Sel selektiert ebenfalls am häufigsten llknn, gefolgt von ets, srw und nnetar, da dieses Verfahren auf dem gleichen Meta-Modell basiert, wie Selbination aber letztlich nur das jeweils eine Verfahren mit der höchsten Eignungswahrscheinlichkeit selektiert. Im Gegensatz dazu selektieren die Verfahren Val-Sel, Pool-2 und Pool-3, welche erwartete Rangfolgen auf Basis von Vergleichen der Validierungsfehler bilden, das Verfahren llknn am seltensten. Dies ist in der Funktionsweise der lokalen k nn-Verfahren begründet. Für kurze Zeitreihen lassen sich bei der Unterteilung der Trainingsmenge in eine kleinere Trainingsmenge und eine Validierungsmenge lediglich wenige nächste Nachbarn zu einem Abfragevektor bestimmen, was im Durchschnitt zu hohen Validierungsfehlern führen kann. Mit steigender Anzahl an Zeitreihenwerten vergrößert sich die Anzahl zu findender Nachbarn, wodurch die Prognosefehler für die Testmenge in der Regel geringer sind als für die Validierungsmenge. In diesem Zusammenhang ist der Validierungsfehler von llknn also kein sinnvolles Maß zur Abschätzung der Prognosefehler auf der späteren Testmenge. Hier ermittelt das Meta-Modell eine bessere Einschätzung der Verteilung der Eignungen der Prognoseverfahrenskandidaten als die Verfahren, die auf den Vergleichen der Validierungsfehler basieren.

Bei separater Betrachtung der Ergebnisse für kurze und lange Zeitreihen sind einige Unterschiede hinsichtlich der Prognosegüte der verschiedenen Verfahren zu erkennen. Während Selbination im Durchschnitt die besten Prognosen für beide Mengen erreicht, unterscheiden sich die Rangfolgen der anderen Verfahren. Comb-4 und ML-Sel erzielen äquivalente Ergebnisse für die Menge der kurzen Zeitreihen (Abbildung 6.2(b)), sie bestimmen aber signifikant schlechtere Prognosen für lange Zeitreihen (Abbildung 6.2(c)). Pool-3 erreicht äquivalente Ergebnisse zu Selbination für lange Zeitreihen, jedoch schlechtere für kurze Zeitreihen. Während keine signifikanten Unterschiede in der Prognosegüte zwischen dem Selektionsverfahren ML-Sel und den Kombinations- sowie Pooling-Verfahren Comb-4, Pool-3 und Pool-2 bestehen, erreicht Selbination die besten Ergebnisse für kurze und lange Zeitreihen. Des Weiteren bestehen signifikante Unterschiede der Prognosegüten der Verfahren llknn und ets, wenn die Länge der Zeitreihen betrachtet wird. llknn erzielt eine äquivalente Prognosegüte zum zweitbesten Verfahren für kurze Zeitreihen

Tabelle 6.2: Anzahl der Selektionen der verschiedenen Prognoseverfahrenskandidaten durch Selbination und die anderen Selektionsverfahren mit mehreren möglichen Modellklassen

Selektionsverfahren	Selektierte Prognoseverfahren				Mittlere Anzahl selektierter Verfahren
	llknn	ets	nnetar	srw	
Selbination	87	65	21	24	1,77
ML-Sel	67	32	5	7	1
Val-Sel	15	46	25	25	1
Pool-2	37	75	66	44	2
Pool-3	66	99	91	77	3
Comb-4	111	111	111	111	4

(Comb-4), erreicht jedoch nur den achten Rang für lange Zeitreihen. Im Gegensatz dazu erzielt ets nur den achten Rang für kurze Zeitreihen, erreicht aber für lange Zeitreihen eine Prognosegüte die äquivalent zum zweitbesten Verfahren (Pool-3) ist. Es lässt sich folgern, dass die Klasse der lokalen k nn-Regressionsverfahren vielversprechendes Potenzial für kurze Zeitreihen zeigt, während die Klasse der exponentiellen Glättungsverfahren die Strukturen und Muster in langen Zeitreihen besser zu modellieren scheint. Die Verfahren nnetar und srw werden in den Nemenyi-Tests nicht betrachtet, da diese in fast allen Experimenten in Tabelle 6.1 schlechtere Ergebnisse erzielen als alle anderen Verfahren. Allerdings zeigen die hohe Prognosegüte von Selbination sowie die leicht bessere Prognosegüte des Kombinationsverfahrens Comb-4 im Vergleich zu den Pooling-Verfahren, dass es dennoch sinnvoll ist, nnetar und srw als Prognoseverfahrenskandidaten in einem Selektions- oder Kombinationsansatz zu berücksichtigen.

Die detaillierten Ergebnisse in Tabelle 6.1 zeigen größtenteils die gleichen Verteilungen der Prognosegüten der verschiedenen Verfahren, wie die durchschnittlichen Ergebnisse in den Abbildungen 6.1 und 6.2 mit einigen Variationen. Für die meisten Experimente erzielt Selbination die besten Prognosen aller Verfahren und ist für alle Experimente mindestens unter den drei besten Verfahren. Während Pool-3 in den meisten Experimenten eine höhere Prognosegüte erreicht als Pool-2, besteht kein klarer Unterschied zwischen den Prognosegüten der Verfahren Pool-3 und Comb-4. Pool-3 prognostiziert besser als Comb-4 für kurze Horizonte und schlechter für mittlere bis lange Horizonte. Comb-4, Pool-3, Pool-2 und ML-Sel sind für viele der Experimente unter den besten Verfahren. ets zeigt Potenzial für lange Zeitreihen und kurze Horizonte, während llknn unabhängig vom Prognosehorizont gute Ergebnisse für kurze Zeitreihen erzielt.

6.2.2 Evaluation unter den Bedingungen der NN3-Competition

Das Evaluationskriterium der NN3-Competition war der Mean-RO-sMAPE $_{T_z, T_z}^{1,18}$, der durchschnittliche sMAPE mit festem Ursprung T_z und den Horizonten $h_{\min} = 1$ bis $h_{\max} = 18$. Tabelle 6.3 zeigt die durchschnittlichen Fehler der in dieser Arbeit verwendeten Verfahren im Vergleich zu den Teilnehmern und den Benchmarks der NN3-Competition (vgl. Crone et al., 2011) sowie den Teilnehmern der M3-Competition (vgl. Hyndman et al., 2018a).

Tabelle 6.3: Mean-RO-sMAPE $_{T_z, T_z}^{1,18}$ der Teilnehmer und Benchmarks der NN3-Competition, der Teilnehmer der M3-Competition sowie der in dieser Arbeit verwendeten Prognoseverfahren für die Zeitreihen der NN3-Competition

Rang innerhalb		Mean-RO-sMAPE	Prognoseverfahren			
aller Verfahren	NN3-Teilnehmer		NN3-Teilnehmer	NN3-Benchmarks	M3-Teilnehmer	dieser Arbeit
1		14,83				Selbination
2		14,84		B09 Wildi		
3		14,89		B07 Theta		
4	1	15,18	C27 Illies			
5		15,27				Pool-3
6		15,30			ForecastX	
7		15,36			ForecastPro	
8		15,39				Comb-4
9		15,44		B03 ForecastPro		
10		15,45				ML-Sel
11		15,56				Pool-2
12		15,56			Theta	
13		15,65			SmartFcs	
14		15,73			AutoBox2	
15		15,83				Val-Sel
16		15,85			B-J automatic	
17		15,90		B16 DES		
18		15,93		B17 Comb-S-H-D		
19		15,93			AAM1	
20		15,95		B05 Autobox		
21		16,00				llknn
22		16,16			AutoBox1	
23	2	16,17	C38 Adeodato		AutoBox3	
24		16,23				ets
25		16,25				
26	3	16,31	C03 Flores			
27		16,41			PP-Autocast	
28		16,42		B14 SES		
29		16,49		B15 HES		
30	4	16,55	C46 Chen			
31	5	16,57	C13 D'yakonov			
32		16,58			Flores/Pearce2	
33		16,74			Winter	
34	6	16,81		B00 AutomatANN		
35		16,83			ARARMA	
36		16,87			RBF	
37		16,90			AAM2	
38	7	16,92	C50 Kamel			
39		16,97				lcknn
40		17,05		B13 Njimi		
41	8	17,54	C24 Abou Nasr			
42	9	17,62	C31 Theodosiou			
43		17,78		B06 Census-X12		
44		17,79				auto.arima
45		17,84		B02 nMLP		
46		17,87				nnetar
47		17,99			Flores/Pearce1	
48	10	18,24	C26 de Vos			
49		18,32		B01 nSVR		
50		18,46				srw
51	11	18,58	C44 Yan			
52	12	18,62	C11 Perfilieva			
:		:				
:		:				
62	19	20,19	C28 Eruhimov			
63		21,06			Robust-Trend	
64		22,55				rw
65	20	22,60	C51 Papadaki			
66		23,72		B12 Hazarika		
67	21	24,09	C17 Chang			
68	22	25,13	C30 Pucheta			
69	23	32,66	C57 Corzo			
70		43,99				nnfor

Die 23 teilnehmenden Verfahren der NN3-Competition erreichten durchschnittliche Fehler zwischen 15,18% und 27,53%. Daher hätte Selbination bei einer Teilnahme an der NN3-Competition mit einem durchschnittlichen Fehler von 14,83% den ersten Rang belegt und somit den Wettbewerb gewonnen. Des Weiteren hätte Selbination ebenfalls eine höhere Prognosegüte als das beste statistische Benchmark-Verfahren erreicht, welches außer Konkurrenz an der NN3-Competition teilnahm und von keinem der NN3-Teilnehmer geschlagen werden konnte (vgl. Wildi, 2007; Crone et al., 2011). Die Benchmark-Verfahren der vorliegenden Arbeit, Pool-3, Comb-4, ML-Sel, Pool-2, Val-Sel und llknn, hätten mit durchschnittlichen Fehlern von 15,27%, 15,39%, 15,45%, 15,56%, 15,83% und 16,00% jeweils den zweiten Rang der NN3-Competition erreicht und den dritten Rang mit einem durchschnittlichen Fehler von 16,25%. nnetar und srw hätten mit durchschnittlichen Fehlern von 17,87% und 18,46% die Ränge 10 und 11 eingenommen. Die Verfahren lcknn, auto.arima, rw und nnfor, die in anderen Konfigurationen hinsichtlich der Menge an Prognoseverfahrenskandidaten des Selbinationsverfahrens verwendet werden (vgl. Abschnitt 6.4), hätten mit durchschnittlichen Fehlern von 16,97%, 17,79%, 22,55% und 43,99% die Ränge 8, 10, 20 und 24 belegt.

Tabelle 6.3 zeigt, dass viele der Teilnehmer der ursprünglichen M3-Competition ähnlich gute Ergebnisse für die Zeitreihen der NN3-Competition erzielen, wie die Teilnehmer und Benchmarks der NN3-Competition. Die folgenden Benchmarks der NN3-Competition sind Verfahren, die an der M3-Competition teilgenommen haben: B16 DES, B17 Comb-S-H-D, B14 SES, B15 HES, und B00 AutomatANN (vgl. Anhang B). Die Rangfolge der Teilnehmer der M3-Competition für die Zeitreihen der NN3-Competition ist ähnlich, wie die Verteilung für die gesamte Menge der 808 monatlichen industriellen Zeitreihen der M3-Competition, jedoch mit leichten Unterschieden. Beispielsweise erzielen für die NN3-Daten die Verfahren ForecastX und ForecastPro bessere Ergebnisse als Theta, das beste Verfahren für die M3-Daten. Das weiterentwickelte Theta-Verfahren in der NN3-Competition erreicht ebenfalls bessere Prognosen als das ursprüngliche Theta-Verfahren der M3-Competition.

Insgesamt bestätigt der Vergleich gegen die Teilnehmer und Benchmarks der NN3- sowie die Teilnehmer der M3-Competition die Ergebnisse aus Abschnitt 6.2.1. Selbination erzielt die besten Ergebnisse aller verglichenen Verfahren. ML-Sel erzielt bessere Ergebnisse als Val-Sel, was nahelegt, dass Meta-Lernverfahren sich besser zur Selektion von Prognoseverfahren eignen als Prognosefehlervergleiche. Des Weiteren zeigt sich, dass eine Selektion aus mehreren möglichen Modellklassen, wie bei ForecastX, ForecastPro oder ML-Sel, das Potenzial zur Ermittlung besserer Prognosen bietet als eine Selektion aus einer vorgegebenen Modellklasse, wie bei BJ-automatic, DES, llknn oder ets.

6.3 Evaluation hinsichtlich des Rechenaufwands

Tabelle 6.4 zeigt die durchschnittlichen Rechenaufwände von Selbination im Vergleich zu den Benchmark-Verfahren für die Zeitreihen der NN3-Competition. Es werden die durchschnittlichen Zeiten pro Training eines finalen Prognosemodells und pro Prognose bestimmt. Abbildung 6.3 visualisiert die Ergebnisse von Friedman- und Nemenyi-Tests hinsichtlich der Zeiten pro Training eines Modells und pro Prognose für Signifikanzniveaus von 0,05. In beiden Fällen werden die Nullhypothesen der Friedman-Tests verworfen und somit lässt sich folgern, dass signifikante Unterschiede zwischen den Rechenaufwänden der verschiedenen Verfahren bestehen. Die Ergebnisse der Nemenyi-Tests zeigen ähnliche Rangfolgen, wie Tabelle 6.4. Der saisonale Random Walk (srw) und das lokal lineare k nn-Verfahren (llknn) trainieren keine Modelle im Voraus, sondern führen alle Berechnungen erst bei der Bestimmung von Prognosen durch. nnetar benötigt einen geringeren durchschnittlichen Rechenaufwand für das Training von Modellen als das Verfahren Val-Sel, dessen Rechenaufwand sich aus den Aufwänden für die folgenden Schritte zusammensetzt: (i) Training der vier Prognoseverfahrenskandidaten llknn, ets, nnetar und srw anhand der Trainingsmenge, (ii) Prognose der Werte der Validierungsmenge mit allen Kandidaten, (iii) Ermittlung der Validierungsfehler, (iv) Selektion des Verfahrens mit dem geringsten Validierungsfehler und (v) Verwendung des selektierten Verfahrens, um ein finales Modell zu trainieren. Die Pooling-Verfahren Pool-2 und Pool-3 benötigen zusätzlich die Zeiten, um die Verfahren mit dem zweit- und drittniedrigsten Validierungsfehler zu verwenden und Modelle zu trainieren. ets benötigt einen äquivalenten Rechenaufwand zu Val-Sel und Pool-2, um ein Modell zu trainieren, jedoch einen höheren Aufwand als Pool-3. Der Rechenaufwand für das Training eines finalen Kombinationsmodells durch das Verfahren Comb-4 ist die Summe der Rechenaufwände der vier Prognoseverfahren llknn, ets, nnetar und srw. ML-Sel und Selbination benötigen die Zeiten zur Bestimmung der Meta-Merkmale und zur Bestimmung der Eignungswahrscheinlichkeiten durch Anwendung des Meta-Modells. Zusätzlich benötigt ML-Sel den Rechenaufwand, um das Prognoseverfahren mit der höchsten Eignungswahrscheinlichkeit zu verwenden, um ein Prognosemodell zu trainieren. Abhängig von der Verteilung der Eignungswahrscheinlichkeiten benötigt Selbination möglicherweise zusätzlich den Aufwand, um weitere Prognoseverfahren anzuwenden und weitere Modelle zu trainieren, bis die Summe der Eignungswahrscheinlichkeiten die Hybriditätsgrenze übersteigt. Tabelle 6.4 zeigt, dass Selbination den höchsten Rechenaufwand zum Training eines finalen Prognosemodells benötigt. Allerdings fällt dieser Rechenaufwand lediglich einmal pro

Tabelle 6.4: Durchschnittlicher Rechenaufwand (und Ränge) pro Training eines Modells und pro Prognose von Selbination und den Benchmark-Verfahren (Verfahren mit dem niedrigsten durchschnittlichen Rechenaufwand sind fettgedruckt)

Durchschnittliche Zeit pro	srw	nnetar	ets	llknn	Val-Sel	ML-Sel	Pool-2	Pool-3	Comb-4	Selbination
Training eines Modells [s]	0 1*	0,1198 3	0,8054 6	0 1*	0,3625 4	7,5364 9	0,6088 5	0,8166 7	0,9252 8	7,7928 10
Prognose [s]	0,0006 1*	0,0123 8	0,0037 3	0,0030 2	0,0051 5	0,0042 4	0,0109 7	0,0155 9	0,0195 10	0,0075 6

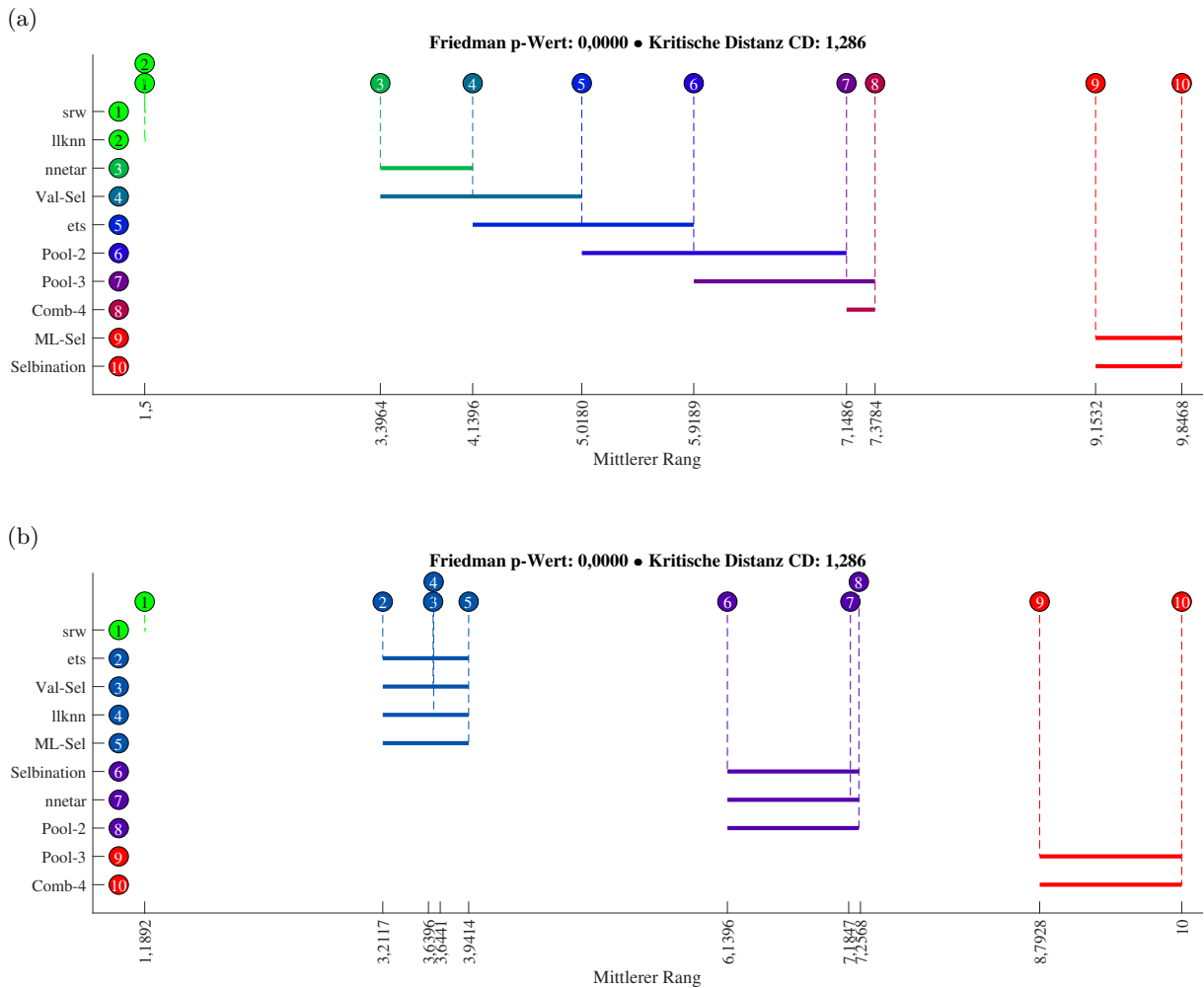


Abbildung 6.3: Nemenyi-Test-Ergebnisse hinsichtlich der Rechenaufwände (a) pro Training eines Modells und (b) pro Prognose des Selbinationsverfahrens sowie verschiedener Benchmark-Verfahren für die Zeitreihen der NN3-Wettbewerb (CD bezeichnet die kritische Distanz signifikanter Differenzen für ein Signifikanzniveau von 0,05)

Zeitreihe an. Darauf folgt dann das trainierte Modell verwendet werden, um diverse zukünftige Zeitreihenwerte zu prognostizieren.

Im Gegensatz zum Rechenaufwand für das Training eines Modells fällt der Rechenaufwand pro Prognose häufiger an, da Prognosen verschiedener Horizonte für jede Zeitreihe ermittelt werden. Tabelle 6.4 und Abbildung 6.3 b) zeigen, dass Selbination im Durchschnitt einen signifikant niedrigeren Rechenaufwand pro Prognose benötigt, als das Kombinationsverfahren Comb-4 und das Pooling-Verfahren Pool-3. Der Rechenaufwand pro Prognose von Selbination ist äquivalent zu den Aufwänden für nnetar und Pool-2. ML-Sel, llknn, Val-Sel und ets benötigen geringere Rechenaufwände pro Prognose, welche zueinander äquivalent sind. Der Rechenaufwand pro Prognose von srw ist signifikant am niedrigsten.

Es sei angemerkt, dass die Berechnungen des Selbinationsverfahrens auf selbst entwickelten Bibliotheken sowie öffentlich verfügbaren Bibliotheken in `R` und `Matlab` beruhen. Diese Konfiguration ist ausreichend für die prototypischen Implementierungen und Evaluationen der vorliegenden Arbeit. Eine professionelle Implementierung in einer effizienteren Programmiersprache würde weiteres Potenzial zur Reduzierung des benötigten Rechenaufwands bieten. Dies liegt aber außerhalb des Fokus dieser Arbeit.

6.4 Evaluation hinsichtlich der Robustheit

In diesem Abschnitt wird die Robustheit des Selbinationsverfahrens evaluiert. Zunächst werden verschiedene Strategien zur Konfiguration des Selbinationsverfahrens hinsichtlich der Menge der Prognoseverfahrenskandidaten \mathbf{A} , der Menge der Merkmalsextraktionsfunktionen \mathbf{F} und der Hybriditätsgrenze Ω_{\min} erläutert. Darauffolgend werden die Prognosegüten verschiedener Parameterkonfigurationen verglichen.

6.4.1 Strategien zur Konfiguration der Parameter des Selbinationsverfahrens

Um eine eindeutige Parameterkonfiguration des Selbinationsverfahrens zu erhalten, müssen die Menge der Prognoseverfahrenskandidaten \mathbf{A} , die Menge der Merkmalsextraktionsfunktionen \mathbf{F} und die Hybriditätsgrenze Ω_{\min} festgelegt werden (vgl. Kapitel 5). In dieser Arbeit kann die Menge der Verfahrenskandidaten \mathbf{A} jede Teilmenge der Menge

$$\mathbf{A}_8 = \{\text{lknn}, \text{llknn}, \text{auto.arima}, \text{ets}, \text{nnfor}, \text{nnetar}, \text{rw}, \text{srw}\} \quad (6.2)$$

sein, die mindestens zwei Verfahren enthält. Mit der ausgewählten Konfiguration wird das Meta-Modell des Selbinationsverfahrens anhand der Meta-Trainingsmenge trainiert und darauffolgend anhand der Meta-Testmenge evaluiert (vgl. Abschnitt 6.1.2). Um eine geeignete Teilmenge von \mathbf{A}_8 auszuwählen, werden zunächst die Prognosegüten der acht Prognoseverfahrenskandidaten für die Meta-Trainingsmenge verglichen. Hierzu wurden alle acht Verfahrenskandidaten verwendet, um Modelle für alle Zeitreihen der Meta-Trainingsmenge zu trainieren. Danach wurden mit Hilfe dieser Modelle die zukünftigen Werte der Zeitreihen prognostiziert und die Prognosefehler ($\text{RO-sMAPE}_{T_z, T_z+6}^{1,12}$) ermittelt. Die Nullhypothese eines Friedman-Tests zum Vergleich der Prognosefehler der acht Verfahren für die Zeitreihen der Meta-Trainingsmenge wurde zu einem Signifikanzniveau von 0,05 verworfen. Abbildung 6.4 visualisiert die Ergebnisse eines darauffolgenden Nemenyi-Tests zum Signifikanzniveau 0,05. Das lokal lineare k nn-Regressionsverfahren `llknn` erzielt die signifikant besten Prognoseergebnisse der acht Kandidaten für die Meta-Trainingsmenge. Das lokal konstante k nn-Verfahren `lknn` und das exponentielle Glättungsverfahren `ets` erreichen mit zueinander äquivalenten Ergebnissen die zweitbesten Prognosen, gefolgt von `auto.arima`. Die durchschnittliche Prognosegüte des Verfahrens zum Training eines Ensembles neuronaler Netze `nnetar` ist äquivalent zu der des saisonalen Random Walks `srw` und signifikant besser als die Güte

des nicht-saisonalen Random Walks `rw`. Das zweite Verfahren zum Training eines Ensembles neuronaler Netze `nnfor` bestimmt die signifikant schlechtesten Prognosen der acht Verfahren für die Meta-Trainingsmenge.

Anhand des Vergleichs der durchschnittlichen Fehler für die Meta-Trainingsmenge werden zwei verschiedene Strategien zur Bestimmung der Menge \mathbf{A} verglichen. Die erste Strategie wird als *Kandidatenauswahl nach Fehlern* bezeichnet. Hierbei wird die Teilmenge \mathbf{A}_M^E mit M Verfahrenskandidaten anhand der Rangfolge der Verfahren hinsichtlich der Prognosefehler für die Meta-Trainingsmenge bestimmt. Somit enthält beispielsweise die Teilmenge der vier Kandidaten nach dieser Strategie die Verfahren `llknn`, `lcknn`, `ets` und `auto.arima`. Diese Strategie zur Auswahl der Menge an Prognoseverfahrenskandidaten ist naheliegend. Allerdings hat sie den Nachteil, dass ähnliche Prognoseverfahren, wie beispielsweise `llknn` und `lcknn` auch ähnliche Prognosen ermitteln, wodurch eine Kombination dieser Verfahren aufgrund einer fehlenden Diversität weniger vielversprechend ist als eine Kombination von Verfahren aus unterschiedlichen Modellklassen (vgl. Lemke und Gabrys, 2010a). Aus diesem Grund wird in der vorliegenden Arbeit eine zweite Strategie verfolgt, die als *Kandidatenauswahl nach Klassen* bezeichnet wird. Hierbei wird für die Bestimmung der Teilmenge \mathbf{A}_M^C mit M Kandidaten die gleiche Rangfolge der Prognoseverfahrenskandidaten hinsichtlich ihrer Prognosefehler für die Meta-Trainingsmenge verwendet, allerdings werden zunächst die besten Verfahren aus anderen Modellklassen ausgewählt, bevor ein zweites Verfahren aus einer bereits vorhandenen Modellklasse hinzugefügt wird. In diesem Zusammenhang werden die acht Verfahrenskandidaten in vier verschiedene Klassen eingeteilt. Die Klasse der lokalen *knn*-Verfahren enthält `llknn` und `lcknn`. Die Klasse der Random Walks besteht aus `srw` und `rw`. Die Klasse der neuronalen Netze enthält `nnetar` und `nnfor`. Zudem werden die Verfahren `ets` und `auto.arima` zur Klasse der statistischen Prognoseverfahren zusammengefasst. Hierdurch enthält die Menge der vier Prognoseverfahrenskandidaten nach dieser Strategie beispielsweise die Verfahren `llknn`, `ets`, `nnetar` und `srw`. Obwohl `lcknn` und

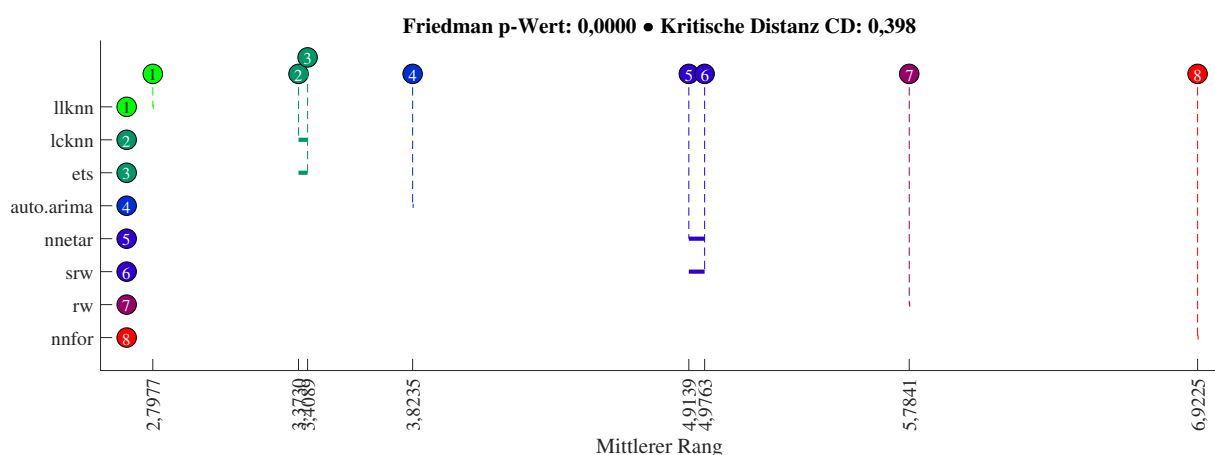


Abbildung 6.4: Nemenyi-Test-Ergebnisse hinsichtlich der Fehler der Prognoseverfahrenskandidaten auf der Meta-Trainingsmenge

auto.arima die zweit- und vierbesten Prognosen für die Meta-Trainingsmenge ermitteln (vgl. Abbildung 6.4), werden diese erst als fünftes und sechstes Verfahren der Menge \mathbf{A} hinzugefügt, da zunächst mindestens ein Verfahren aus jeder Klasse in der Menge enthalten sein muss, bevor ein zweites Verfahren aus einer bereits vorhandenen Klasse hinzugefügt wird. Tabelle 6.5 zeigt die Konfigurationen der Menge der Prognoseverfahrenskandidaten nach den beiden beschriebenen Strategien. Für 2-5 Kandidaten führen die beiden Strategien zu unterschiedlichen Teilmengen, während sich für 6-8 Kandidaten die gleichen Teilmengen ergeben.

Nach der Auswahl der Menge an Prognoseverfahrenskandidaten \mathbf{A} wird die Menge an Merkmalsextraktionsfunktionen \mathbf{F} festgelegt. Hierzu werden ebenfalls zwei unterschiedliche Strategien verglichen. Für jede Menge \mathbf{A} wird sowohl die Konfiguration mit allen möglichen Merkmalsextraktionsfunktionen verwendet als auch die Konfiguration mit der Menge nach einer Merkmalsselektion. Die Anzahl der verwendeten Merkmale für alle verglichenen Konfigurationen sind in Anhang D aufgeführt. Für die Hybriditätsgrenze Ω_{\min} werden die Werte im Bereich $0, \frac{1}{10}, \frac{2}{10}, \dots, 1$ betrachtet.

6.4.2 Vergleich der Prognosegüten verschiedener Parameterkonfigurationen

Abbildung 6.5 zeigt die durchschnittlichen Prognosefehler der im letzten Abschnitt definierten Konfigurationen des Selbinationsverfahrens für die Meta-Testmenge der Zeitreihen der NN3-Competition. Als Prognosefehler wird der Mean-RO-sMAPE $_{T_z, T_z+6}^{1,12}$ verwendet, da dieser den durchschnittlichen Fehler über verschiedene Horizonte von unterschiedlichen Ursprüngen bildet und somit ein robustes Maß der Prognosegüte für verschiedene Experimentkonfigurationen darstellt. Wie die Grafik zeigt, bestehen keine signifikanten Unterschiede der Prognosegüten des Selbinationsverfahrens für die verschiedenen Konfigurationen hinsichtlich der Menge an Kandidaten \mathbf{A} . Die einzige Teilmenge mit im Durchschnitt höheren Prognosefehlern als für die anderen Teilmengen ist die Konfiguration mit $\mathbf{A}_2^E = \{\text{llknn}, \text{lcknn}\}$. Dies ist intuitiv naheliegend,

Tabelle 6.5: Enthaltene Prognoseverfahrenskandidaten für alle Teilmengen der Menge \mathbf{A}

\mathbf{A}	Kandidatenauswahl nach		Enthaltene Prognoseverfahrenskandidaten							
	Fehlern	Klassen	llknn	ets	nnetar	srw	lcknn	auto.arima	rw	nnfor
\mathbf{A}_2^E	✓		✓				✓			
\mathbf{A}_2^C		✓	✓	✓						
\mathbf{A}_3^E	✓		✓	✓			✓			
\mathbf{A}_3^C		✓	✓	✓	✓					
\mathbf{A}_4^E	✓		✓	✓			✓	✓		
\mathbf{A}_4^C		✓	✓	✓	✓	✓				
\mathbf{A}_5^E	✓		✓	✓	✓		✓	✓		
\mathbf{A}_5^C		✓	✓	✓	✓	✓	✓			
\mathbf{A}_6	✓	✓	✓	✓	✓	✓	✓	✓		
\mathbf{A}_7	✓	✓	✓	✓	✓	✓	✓	✓	✓	
\mathbf{A}_8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

da beide Verfahren aus der gleichen Modellklasse stammen und ähnliche Prognoseergebnisse liefern. Daher fehlen in dieser Konfiguration des Selbinationsverfahrens die Potenziale, die die Einbeziehung anderer Verfahrensklassen bietet. Alle anderen Konfigurationen hinsichtlich der Menge \mathbf{A} erreichen zueinander ähnliche durchschnittliche Prognosefehler. Diese Ergebnisse zeigen, dass das neuartige Selbinationsverfahren der vorliegenden Arbeit robust hinsichtlich der Wahl der Menge an Prognoseverfahrenskandidaten ist. Die für die Evaluationsstudien gewählte Konfiguration (6.1) mit den vier Verfahrenskandidaten in $\mathbf{A}_4^C = \{\text{llknn, ets, nnetar, srw}\}$ erzielt die besten Prognoseergebnisse im Vergleich aller Konfigurationen des Selbinationsverfahrens. Die anderen Konfigurationen erreichen aber ähnlich gute Ergebnisse. Die Anzahl der Selektionen der verschiedenen Prognoseverfahrenskandidaten für ausgewählte Konfigurationen des Selbinationsverfahrens werden in Anhang D dargestellt.

Hinsichtlich der Menge an Merkmalsextraktionsfunktionen \mathbf{F} werden in den meisten Fällen bessere Ergebnisse erzielt, wenn nicht alle möglichen Funktionen verwendet werden, sondern diejenigen, die nach der Merkmalsselektion für die jeweilige Konfiguration hinsichtlich \mathbf{A} übrig bleiben. Für alle verschiedenen Konfigurationen der Mengen \mathbf{A} und \mathbf{F} werden die jeweils besten durchschnittlichen Prognosefehler für Werte der Hybriditätsgrenze im Bereich $0 < \Omega_{\min} < 1$ erreicht. Dies zeigt, dass hybride Ansätze, die die Möglichkeit der Selektion eines oder mehrerer Verfahren haben, stets besser funktionieren als reine Selektions- ($\Omega_{\min} = 0$) und reine Kombinationsansätze ($\Omega_{\min} = 1$). Dies bestätigt erneut FT 3 und somit das große Potenzial des neuartigen Selbinationsverfahrens im Vergleich zu gängigen Selektions- und Kombinationsverfahren.

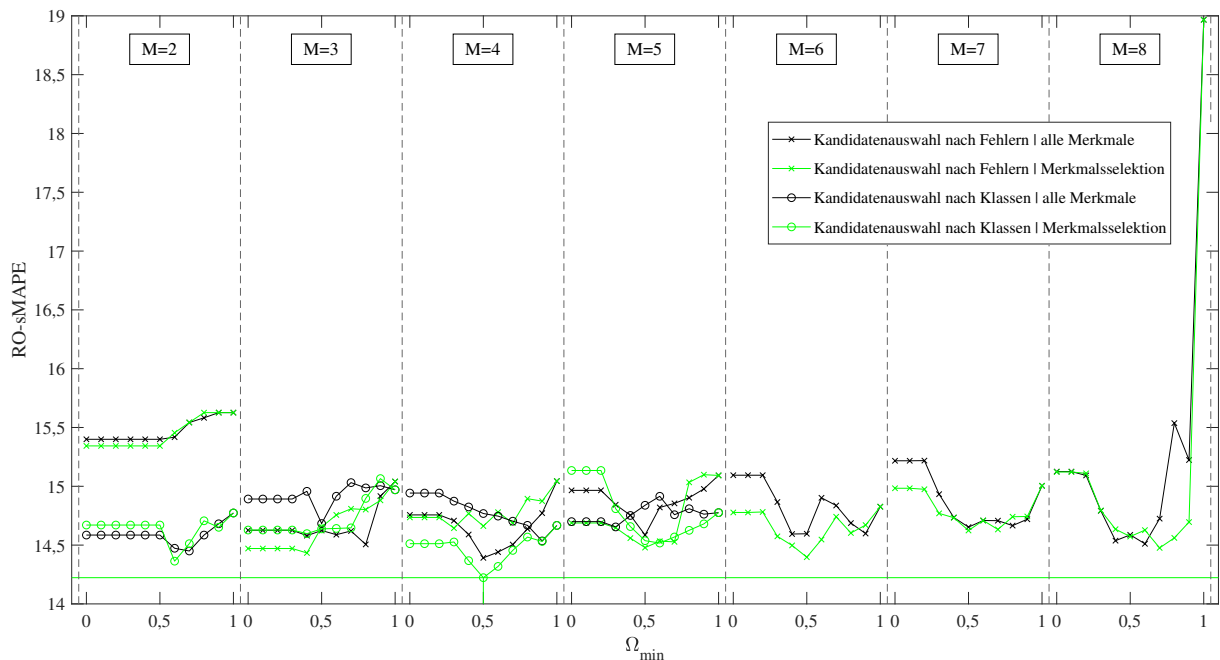


Abbildung 6.5: Vergleich der durchschnittlichen Prognosefehler ($\text{RO-sMAPE}_{T_z, T_z+6}^{1,12}$) verschiedener Konfigurationen des Selbinationsverfahrens für die Zeitreihen der NN3-Competition

6.5 Zusammenfassung

In diesem Kapitel wurde das neuartige Selbinationsverfahren dieser Arbeit hinsichtlich der Prognosegüte und des Rechenaufwands anhand des Datensatzes der 111 monatlichen Zeitreihen der NN3-Competition evaluiert. Zunächst wurde eine spezifische Konfiguration des Selbinationsverfahrens gegen gängige Benchmark-Verfahren verglichen. Daraufaufgehend wurden verschiedene Parameterkonfigurationen verglichen, um die Robustheit des Verfahrens zu untersuchen.

In Abschnitt 6.2 wurde das Selbinationsverfahrens in der Konfiguration mit den vier Prognoseverfahrenskandidaten `llknn`, `ets`, `nnetar` und `srw`, den 25 Merkmalsextraktionsfunktionen in Tabelle 5.2 sowie der Hybriditätsgrenze $\Omega_{\min} = 0,5$ gegen gängige Benchmark-Verfahren in diversen Experimenten mit verschiedenen Prognosehorizonten und unterschiedlichen Ursprüngen evaluiert. Hierzu wurden durchschnittliche symmetrische mittlere absolute prozentuale Fehler rollierender Prognosen ermittelt sowie zusätzlich Friedman- und Nemenyi-Tests durchgeführt, um die Signifikanz der Unterschiede in der Prognosegüte zu testen. Als Benchmark-Verfahren wurden sowohl Selektionsverfahren mit vorgegebener Modellklasse (`llknn`, `ets`, `srw`) als auch Kombinationsverfahren mit vorgegebener Modellklasse (`nnetar`), Selektionsverfahren mit mehreren möglichen Modellklassen (Val-Sel, ML-Sel) sowie Kombinations- und Pooling-Verfahren aus verschiedenen Modellklassen (Comb-4, Pool-3, Pool-2) verwendet. In den Experimenten erreichte das Selbinationsverfahren die signifikant besten Ergebnisse aller verglichenen Prognoseverfahren gemittelt über alle durchgeführten Experimente. Das Verfahren ermittelte sowohl die besten Prognosen für kurze als auch für lange Zeitreihen und gemittelt über alle betrachteten Zeitreihen. Die Ergebnisse bestätigen FT 3, denn das Selbinationsverfahren erzielt durch seinen hybriden Ansatz zwischen Selektion und Kombination bessere Prognoseergebnisse als alle in dieser Arbeit verwendeten Selektions-, Kombinations- und Pooling-Verfahren. Dieses Ergebnis wurde zudem in einer weiteren Evaluationsstudie bestätigt, in der eine Teilnahme des Selbinationsverfahrens unter den Originalbedingungen der NN3-Competition simuliert wurde. Dabei erreichte es einen niedrigeren durchschnittlichen Prognosefehler als alle Teilnehmer und Benchmarks der NN3- sowie alle Teilnehmer der M3-Competition. Somit hätte das Selbinationsverfahren die NN3-Competition gewonnen.

Neben den beschriebenen Hauptergebnissen lassen sich aus den Evaluationsstudien weitere Erkenntnisse ableiten. Das Selektionsverfahren ML-Sel, welches sich als Spezialfall des Selbinationsverfahrens mit einer Konfiguration der Hybriditätsgrenze von $\Omega_{\min} = 0$ ergibt, selektiert für jede Zeitreihe jeweils das eine Prognoseverfahren, für welches das Meta-Modell die höchste Eignungswahrscheinlichkeit ermittelt. Dieses Verfahren erreicht eine signifikant höhere Prognosegüte als das Verfahren Val-Sel, welches eine erwartete Rangfolge basierend auf den Validierungsfehlern der Prognoseverfahrenskandidaten bildet und das Verfahren mit dem geringsten Validierungsfehler pro Zeitreihe selektiert. Dies zeigt, dass Meta-Lernansätze sich besser zur Selektion von Prognoseverfahren eignen als Vergleiche von Validierungsfehlern, wodurch FT 2 bestätigt wird.

Zudem zeigen die Evaluationsergebnisse keine klaren Unterschiede in der Prognosegüte zwischen dem Selektionsverfahren ML-Sel und den Kombinations- und Pooling-Verfahren Comb-4, Pool-3 und Pool-2. All diese Verfahren erzielen im Durchschnitt bessere Prognoseergebnisse als llknn, ets, nnetar und srw. Dieses Ergebnis deutet darauf hin, dass eine Selektion oder Kombination mit Verfahren aus mehreren möglichen Modellklassen mehr Potenzial bietet als eine Selektion oder Kombination aus einer vorgegebenen Modellklasse. Von den vier Prognoseverfahrenskandidaten erreicht llknn gute Prognoseergebnisse für kurze Zeitreihen und ets für lange Zeitreihen. nnetar und srw ermitteln im Durchschnitt schlechtere Prognosen als alle anderen verglichenen Verfahren der Evaluationsstudien, allerdings zeigen die guten Ergebnisse des Selbinationsverfahrens, dass eine Einbeziehung dieser Verfahren als Prognoseverfahrenskandidaten dennoch sinnvoll ist.

Nach der Evaluation hinsichtlich der Prognosegüte wurden in Abschnitt 6.3 die Rechenaufwände des Selbinationsverfahrens und der Benchmark-Verfahren pro Training eines Modells und pro Prognose verglichen. Das Selbinationsverfahren benötigt zwar im Durchschnitt den größten Rechenaufwand für das Training eines finalen Prognosemodells, allerdings benötigt es einen signifikant geringeren Rechenaufwand pro Prognose als die Verfahren Pool-3 und Comb-4. Da das Training eines Modells lediglich einmalig pro Zeitreihe durchgeführt werden muss, der Rechenaufwand pro Prognose aber für jeden einzelnen prognostizierten Zeitreihenwert anfällt, lässt sich folgern, dass das Selbinationsverfahren einen Vorteil der Recheneffizienz gegenüber gängigen Kombinations- und Pooling-Verfahren bietet.

Zur Bestimmung der Robustheit des Selbinationsverfahrens hinsichtlich der Wahl der Menge an Verfahrenskandidaten \mathbf{A} , der Menge an Merkmalsextraktionsfunktionen \mathbf{F} und der Hybriditätsgrenze Ω_{\min} wurden in Abschnitt 6.4 die Prognosegüten verschiedener Parameterkonfigurationen verglichen. Die Untersuchung zeigte, dass das Selbinationsverfahren robust gegen die Wahl verschiedener Parameterkonfigurationen ist und jeweils eine ähnlich hohe Prognosegüte erzielt, wenn ein paar Vorgaben beachtet werden. Die Menge \mathbf{A} sollte mindestens je einen Kandidaten aus zwei verschiedenen Modellklassen enthalten, es sollte eine Merkmalsselektion durchgeführt werden und als Hybriditätsgrenze sollte ein Wert zwischen $\frac{1}{M} < \Omega_{\min} < 1$ gewählt werden, um das volle Potenzial des Verfahrens auszuschöpfen. Insgesamt zeigen die Evaluationsergebnisse dieses Kapitels, dass das neuartige Selbinationsverfahren bessere Prognoseergebnisse erzielt als gängige Selektions- und Kombinationsverfahren und robust gegenüber der Wahl der Parameterkonfiguration ist.

Kapitel 7

Evaluation des Selbinationsverfahrens mittels Lagerbestandssimulation

„I think there is a world market for maybe five computers.“

Thomas Watson, US-amerikanischer Unternehmer, (1874 - 1956)

Nach der Analyse der Prognosegüte und des Rechenaufwands des Selbinationsverfahrens im letzten Kapitel wird das Verfahren in diesem Kapitel hinsichtlich verschiedener Leistungskennzahlen in einer Lagerbestandssimulation evaluiert. Hierbei wird erneut die Konfiguration des Verfahrens genutzt, die in (6.1) definiert wurde und als Selbination bezeichnet wird. Abschnitt 7.1 erläutert das implementierte ereignisdiskrete Simulationsmodell inklusive der verwendeten Lagerhaltungspolitik. Für eine gegebene Zeitreihe der Kundennachfrage werden zukünftige Werte prognostiziert und es werden mit Hilfe eines Bestellrhythmusverfahrens Produktionsaufträge generiert. Abschnitt 7.2 beschreibt die Experimentkonfiguration und die Evaluationskriterien. Abschnitt 7.3 stellt die Ergebnisse der Simulationsstudie dar.

7.1 Simulationsmodell und Lagerhaltungspolitik

7.1.1 Zustandsvariablen und Ablauf der Simulation

Abbildung 7.1 gibt einen Überblick über das verwendete Lagerbestandsmodell, dessen Verhalten im Folgenden explizit durch mathematische Gleichungen beschrieben wird (vgl. Kück und Freitag, 2021). Aus Gründen der Übersichtlichkeit wird angenommen, dass jede der $Z = 111$ monatlichen industriellen Zeitreihen des NN3-Datensatzes die Nachfrage eines spezifischen Kunden für ein spezielles Produkt des produzierenden Unternehmens widerspiegelt. Für jedes der Z Produkte existiert eine eigene Produktionslinie. Dabei wird angenommen, dass sowohl die verschiedenen Produktionslinien als auch die Zeitreihen der Kundennachfrage untereinander unabhängig sind. Rohmaterial und Produktionskapazitäten hinsichtlich Maschinen und Personal werden als verfügbar angenommen und nicht näher betrachtet. Lieferzeiten für fertige Produkte und

Rohmaterialien sowie Fixkosten für Rüstvorgänge und Bestellungen werden ebenfalls vernachlässigt. Dies bedeutet, dass für jede Zeitreihe der *Wiederaufstockungsprozess* (engl. replenishment process) der *Fertigwarenlager* (engl. finished goods inventory) als ein *unabhängiges einstufiges Einprodukt-Lagerbestandsproblem* (engl. independent single-stage single-item inventory control problem) mit dem Ziel der Erreichung eines hohen Servicegrades mit geringen Sicherheitsbeständen betrachtet werden kann. Tabelle 7.1 listet alle Variablen und Evaluationsgrößen der Lagerbestandssimulation auf. Diese werden im Folgenden erläutert.

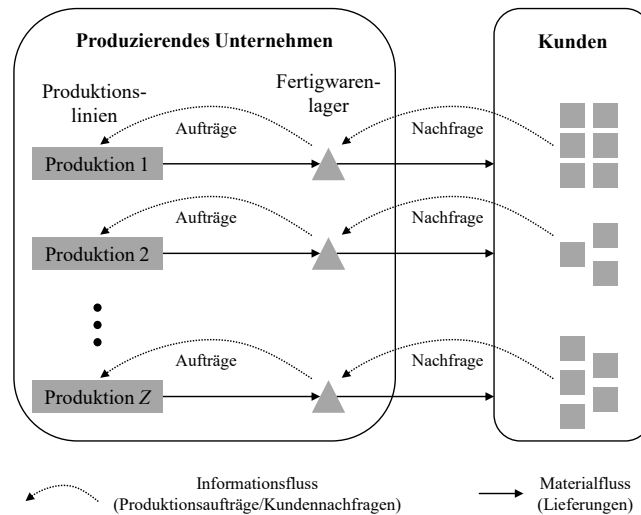


Abbildung 7.1: Verwendetes Lagerbestandsmodell

Sei $y_t^z = y_t$ die Kundennachfrage für Produkt z in Periode t , d.h. zwischen den Zeitpunkten $t - 1$ und t . Aus Gründen der Übersichtlichkeit wird der Index des Produktes z in den folgenden Beschreibungen nicht aufgeführt. Sei $\hat{y}_{t-j|t}$ die Prognose für die Kundennachfrage in Periode t , die am Ende von Periode $t - j$ ermittelt wurde. Der *Nettobestand* (engl. net inventory of finished goods) am Ende von Periode t ist definiert als die Differenz des *physischen Bestands auf Lager* (engl. physical inventory on hand) und der *Lieferrückstände* (engl. backorders)

$$i_t = i_t^+ - i_t^-, \quad (7.1)$$

mit

$$i_t^+ = \max(i_t, 0) \quad (7.2)$$

$$i_t^- = \max(-i_t, 0). \quad (7.3)$$

Es wird angenommen, dass offene Nachfragen zurückgestellt werden, wenn sie nicht erfüllt werden können, weil der physische Bestand auf Lager erschöpft ist. Zurückgestellte Nachfragen können in einer der folgenden Perioden bedient werden. Des Weiteren wird angenommen, dass die *Produktionszeit* (auch Wiederbeschaffungszeit, engl. replenishment time) zur Herstellung oder Beschaffung neuer Produkte W Zeitperioden entspricht. Somit sind die Produkte p_t , für die die

Tabelle 7.1: Variablen und Evaluationskriterien der Lagerbestandssimulation für ein spezifisches Produkt

y_t	Kundennachfrage innerhalb der Periode t
$\hat{y}_{t-j t}$	Prognose der Kundennachfrage für Periode t berechnet am Ende von Periode $t - j$
i_t	Nettobestand fertiger Produkte am Ende von Periode t
i_t^+	physischer Bestand fertiger Produkte auf Lager am Ende von Periode t
i_t^-	lieferrückständige Produkte am Ende von Periode t
p_t	Produktionsaufträge generiert am Ende von Periode t
W	Produktionszeit (Zeit zur Produktion eines neuen Produktes)
w_t	Umlaufbestand am Ende von Periode t
I_t	disponibler Bestand am Ende von Periode t
R	Überprüfungsintervall
s_t	dynamischer Sicherheitsbestand am Ende von Periode t
S_t	dynamisches Zielbestellniveau am Ende von Periode t
α^*	Zielservicegrad
$e_{W,t}$	W -Schritt-Prognosefehler der am Ende von Periode t ermittelten Prognosen
ϵ_W	Zufallsvariable der W -Schritt-Prognosefehler
$\hat{\sigma}_{e_t}$	empirische Standardabweichung der Einschrittprognosefehler bis Periode t
α	erreichter Servicegrad
s_{scaled}	skalierter Sicherheitsbestand
$\text{Bias}_{W,t}$	Verzerrung der W -Schritt-Prognosen eines Verfahrens
$\text{RMS}(\cdot/\cdot)$	Quadratmittelwert erreichter Servicegrade und skalierten Sicherheitsbestände

Produktionsaufträge zum Ende der Periode t generiert wurden, W Perioden später zu Beginn der Periode $t + W + 1$ verfügbar. Die Produktionszeiten und die Produktionsauftragsmengen werden als deterministisch angenommen. Die zum Ende einer Periode t noch nicht abgeschlossenen Produktionsaufträge werden als *Umlaufbestand* (engl. work-in-process) bezeichnet

$$w_t = \begin{cases} 0, & \text{wenn } W = 0 \\ \sum_{j=1}^W p_{t-j}, & \text{sonst.} \end{cases} \quad (7.4)$$

Mit Hilfe dieser Werte wird der *disponible Bestand* (engl. inventory position) am Ende von Periode t als die Summe des Nettobestandes und des Umlaufbestandes definiert:

$$I_t = i_t + w_t. \quad (7.5)$$

Für den Wiederaufstockungsprozess wird ein (R, S_t) -Bestellrhythmusverfahren mit ange-strebttem Zielbestandsniveau (engl. periodic-review order-up-to policy) verwendet, wobei ein monatliches Überprüfungsintervall ($R = 1$) vorgegeben ist und das Zielbestandsniveau S_t dynamisch ist. Der Ablauf der Ereignisse in jeder Periode der Simulation ist wie folgt: (i) Ankunft gefertigter Produkte, (ii) Erfüllung der Kundennachfrage aus dem physischen Bestand auf Lager oder Lieferrückstellung, (iii) Aktualisierung des Nettobestands, des Umlaufbestands und des disponiblen Bestands, (iv) Berechnung von Nachfrageprognosen für die zukünftigen Perioden,

(v) Bestimmung des dynamischen Zielbestandsniveaus und (vi) Generierung neuer Produktionsaufträge. Diese Ereignisse werden in Abbildung 7.2 exemplarisch visualisiert und im Folgenden detailliert beschrieben.

Zu Beginn einer Periode t werden die Produktionsaufträge p_{t-W-1} fertiggestellt, die W Perioden früher, d.h. am Ende von Periode $t-W-1$, generiert wurden. Diese werden zum physischen Lagerbestand i_t^+ hinzugefügt. Nach einer Bedienung möglicher zurückgestellter Nachfragemengen der letzten Perioden wird die Kundennachfrage y_t erfüllt, die im Laufe der Periode auftrat, indem die Produkte aus dem Lagerbestand entnommen werden, so lange physischer Lagerbestand vorhanden ist ($i_t^+ > 0$). Wenn dieser aufgebraucht ist ($i_t^+ = 0$), werden unerfüllte Nachfragen zurückgestellt. Am Ende der Periode ist die gesamte Nachfrage y_t der t -ten Periode bekannt und der Nettobestand, der Umlaufbestand sowie der disponible Bestand werden wie folgt aktualisiert:

$$i_t = i_{t-1} + p_{t-W-1} - y_t \quad (7.6)$$

$$w_t = w_{t-1} + p_{t-1} - p_{t-W-1} \quad (7.7)$$

$$I_t = I_{t-1} + p_{t-1} - y_t. \quad (7.8)$$

Darauffolgend werden Nachfrageprognosen $\hat{y}_{t|t+h}$ für $h = 1, \dots, W$ berechnet, indem ein selektiertes Prognosemodell auf die Zeitreihe vergangener Kundenbestellungen angewandt wird. Für einen vorgegebenen *Zielservicegrad* (engl. target service level) α^* soll das *dynamische Zielbestandsniveau* (engl. dynamic order-up-to level) S_t so gewählt werden, dass die Wahrscheinlichkeit,

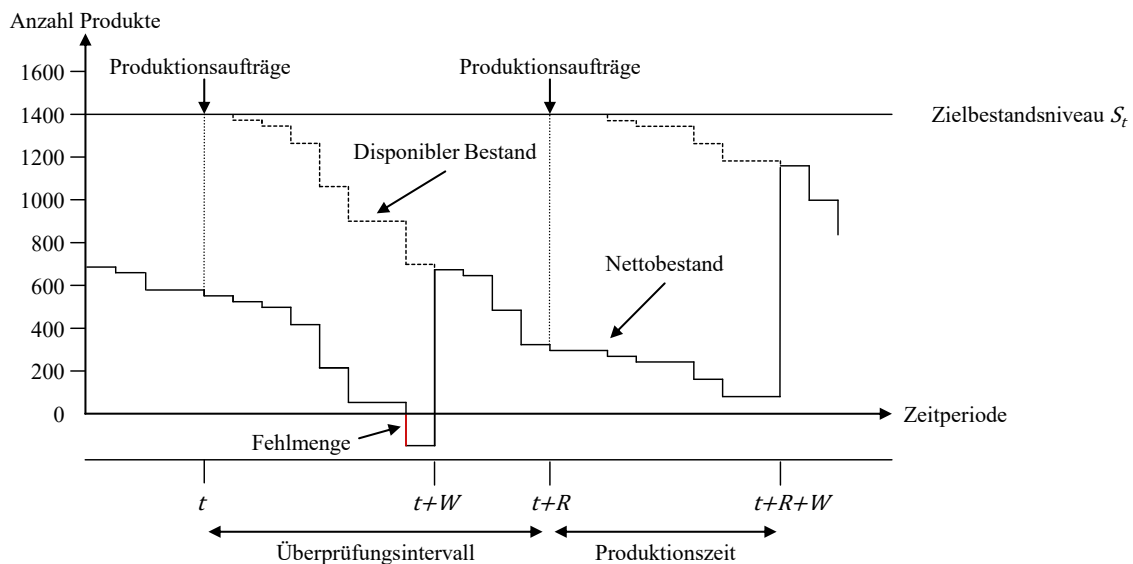


Abbildung 7.2: Lagerbestandsentwicklung bei Verfolgung eines (R, S_t) -Bestellrhythmusverfahrens in Anlehnung an Günther und Tempelmeier (2012)

alle Kundennachfragen der nächsten $W + 1$ Perioden ohne Rückstellungen erfüllen zu können, α^* beträgt. Dieser wird durch

$$S_t = \sum_{h=1}^{W+1} \hat{y}_{t|t+h} + s_t \quad (7.9)$$

bestimmt, wobei s_t den *dynamischen Sicherheitsbestand* (engl. dynamic safety stock) am Ende der Periode t bezeichnet. Der Sicherheitsbestand ist aufgrund der Unsicherheit bei der Prognose sinnvoll und wird vorgehalten, um Fehlmengen in Folge von W -Schritt-Prognosefehlern vorzubeugen. Sei

$$e_{W,t} = \sqrt{\sum_{h=1}^{W+1} (y_{t+h} - \hat{y}_{t|t+h})^2} \quad (7.10)$$

der W -Schritt-Prognosefehler der Prognosen, die am Ende der Periode t ermittelt wurden und sei ϵ_W die Zufallsvariable der W -Schritt-Prognosefehler. Der Sicherheitsbestand ist definiert als die Produktmenge, die der W -Schritt-Prognosefehler mit einer Wahrscheinlichkeit von α^* nicht übersteigen wird:

$$\Pr(\epsilon_W \leq s_t) \geq \alpha^*. \quad (7.11)$$

Sei zudem $F_{\epsilon_W}^{-1}(\cdot)$ die inverse Verteilungsfunktion der W -Schritt-Prognosefehler. Dann lässt sich der dynamische Sicherheitsbestand der Periode t als Quantil des Zielservicegrades ermitteln:

$$s_t = Q_{\epsilon_W}(\alpha^*) = F_{\epsilon_W}^{-1}(\alpha^*). \quad (7.12)$$

In diesem Zusammenhang wird gängigerweise in vielen theoretischen und praktischen Anwendungen die Annahme getroffen, dass die Prognosefehler unabhängige und identisch normalverteilte Zufallsvariablen sind. Allerdings wird diese Annahme häufig verletzt, da Prognosefehler über die Produktionszeit W selbst dann korrelieren, wenn die Nachfragen nicht korrelieren (vgl. Prak et al., 2017) und ihre Verteilung von einer Normalverteilung abweichen kann (vgl. Manary et al., 2009; Lee, 2014). Zudem wurde in aktuellen Arbeiten gezeigt, dass traditionelle Methoden zur Ermittlung der Sicherheitsbestände auf Basis der Annahme unabhängig und identisch normalverteilter Zufallsvariablen in der Regel zu einer signifikanten Unterschätzung der notwendigen Sicherheitsbestände führen, wenn die Annahme verletzt wird (vgl. Prak et al., 2017). Daher wird in der vorliegenden Arbeit die nichtparametrische Methode der *Kerndichteschätzung* (engl. kernel density estimation, vgl. Silverman, 1986) verwendet, um die Verteilung der W -Schritt-Prognosefehler zu schätzen, wie von Manary et al. (2009) und Trapero et al. (2019) vorgeschlagen wurde. Für eine gegebene Stichprobe vergangener W -Schritt-Prognosefehler $e_{W,1}, \dots, e_{W,J}$ wird die wahre Wahrscheinlichkeitsdichtefunktion $f_{\epsilon_W}(\cdot)$ als

$$\hat{f}_{\epsilon_W}(e) = \frac{1}{JB} \sum_{j=1}^J \kappa\left(\frac{e - e_{W,j}}{B}\right), \quad (7.13)$$

geschätzt, wobei J die Stichprobengröße angibt, B die Bandbreite bezeichnet und $\mathcal{K}(\cdot)$ die Kernglättungsfunktion, welche zu 1 integriert. Wie von Silverman (1986) und Trapero et al. (2019) vorgeschlagen, wird der Epanechnikov-Kern

$$\mathcal{K}_{\text{Epanechnikov}}(u) = \begin{cases} \frac{3}{4\sqrt{5}} \left(1 - \frac{1}{5}u^2\right), & -\sqrt{5} \leq u \leq \sqrt{5} \\ 0, & \text{sonst} \end{cases} \quad (7.14)$$

verwendet und die Bandbreite wird als

$$B_{\text{opt}} = 0.9\zeta J^{-\frac{1}{5}}, \quad (7.15)$$

gewählt, wobei der adaptive Ausdehnungswert ζ wie folgt definiert wird:

$$\zeta = \min(\text{Standardabweichung}, \text{Interquartilsabstand}/1.34). \quad (7.16)$$

Durch Verwendung der geschätzten Wahrscheinlichkeitsdichtefunktion wird der Sicherheitsbestand durch (7.12) bestimmt und das dynamische Zielbestandsniveau durch (7.9). Letztlich wird eine Menge p_t an neuen Produktionsaufträgen am Ende von Periode t generiert, sodass der disponible Bestand auf das Zielbestandsniveau erhöht wird:

$$p_t = \max(S_t - I_t, 0). \quad (7.17)$$

Es wird angenommen, dass keine Produktionsaufträge generiert werden, wenn der disponible Bestand am Ende einer Periode höher ist als das dynamische Zielbestandsniveau. Zu Beginn der folgenden Periode wird die gleiche Abfolge von Ereignissen erneut durchgeführt.

Das Lagerbestandssimulationsmodell dieser Arbeit ist ein ereignisdiskretes Simulationsmodell (vgl. Banks et al., 2013; Law, 2015). Da das Überprüfungsintervall als $R = 1$ festgelegt ist, ändert das System jeweils zum Ende einer Periode seinen Zustand. Dann ist die komplette Kundennachfrage der Periode bekannt und die Zustandsvariablen des Systems werden durch Ausführung von (7.6)-(7.17) aktualisiert.

7.1.2 Implementierung des Simulationsmodells

Das Simulationsmodell wurde in `Matlab` (vgl. The MathWorks Inc., 2017b) implementiert. Zur Verifikation der korrekten Implementierung wurden die Werte der Zustandsvariablen manuell für verschiedene Beispiele für Kundennachfrage und Prognosen bestimmt. Darauffolgend wurden diese Werte mit den Werten der Zustandsvariablen aus der Simulation verglichen. Dieser Prozess wurde mit Hilfe des Debuggers in `Matlab` durchgeführt. Zusätzlich zur formalen Verifikation, dass das Lagerbestandssimulationsmodell korrekt implementiert wurde, muss validiert werden, dass es ein reales Lagerbestandssystem sinnvoll repräsentiert. Da einstufige Einprodukt-Lagerbestandsmodelle mit der beschriebenen Lagerhaltungspolitik nach dem (R, S_t) -Bestellrhythmusverfahren in diversen theoretischen und praktischen Anwendungen verwendet

wurden und diese als Modelle für gängige reale Lagerbestandssysteme angesehen werden (vgl. Lee et al., 1997; Minner, 2003; Babai und Dallery, 2009; Wang und Petropoulos, 2016; Silver et al., 2016; Petropoulos et al., 2019), lässt sich das implementierte Simulationsmodell als ein valides Modell zur Evaluation der Prognoseverfahren dieser Arbeit hinsichtlich der Evaluationskriterien der Lagerhaltung betrachten, die im folgenden Abschnitt beschrieben werden.

7.2 Experimentbeschreibung

7.2.1 Untersuchungsdimensionen

Tabelle 7.2 zeigt die Untersuchungsdimensionen der Simulationsstudie dieses Kapitels. Wie in Kapitel 6 wird der Datensatz der 111 Zeitreihen der NN3-Competition verwendet, welcher in Abschnitt 3.3 beschrieben wurde. Es wird angenommen, dass jede der Zeitreihen die monatliche Kundennachfrage für ein spezielles Produkt beschreibt und die Zeitreihen werden als untereinander unabhängig betrachtet. Für jede Zeitreihe werden die Leistungsfähigkeiten der zehn in Abschnitt 6.1 beschriebenen Prognoseverfahren für drei verschiedene Zielservicegrade und vier verschiedene Produktionszeiten ermittelt. Somit umfasst die Simulationsstudie dieses Kapitels insgesamt 13320 voneinander unabhängige Simulationsläufe. Jeder Simulationslauf wird für die Werte der Testmenge $\mathbf{y}_{\text{test}} = y_{T+1}, \dots, y_{T+H}$ einer Zeitreihe durchgeführt. Da die Werte der Zustandsvariablen für jede Periode im Laufe einer Simulation deterministisch durch Ausführung von (7.6)-(7.17) bestimmt werden, sind keine Wiederholungen der Simulationsläufe notwendig.

7.2.2 Bestimmung der Initialwerte für die Simulation

Um einen Simulationslauf beginnen zu können, werden Initialwerte für den Nettobestand, den Umlaufbestand und den disponiblen Bestand benötigt. Zu diesem Zweck werden Prognosen und Fehler zusätzlich für die Validierungsmenge $\mathbf{y}_{\text{val}} = \{y_{T-17}, \dots, y_T\}$ einer Zeitreihe berechnet. Als Initialwert des Nettobestandes wird der Sicherheitsbestand verwendet:

$$i_T = s_T = \hat{F}_{\epsilon_w}^{-1}(\alpha^*). \quad (7.18)$$

Tabelle 7.2: Konfigurationen der durchgeführten Simulationsexperimente

Dimension	Ausprägungen
Zeitreihen	111 Zeitreihen der NN3-Competition
Prognoseverfahren	llknn ets nnetar srw Val-Sel ML-Sel Pool-2 Pool-3 Comb-4 Selbination
Zielservicegrad	0,90 0,95 0,99
Produktionszeit	1 2 3 4

Zur Ermittlung eines Initialwertes für den Umlaufbestand, müssen vergangene Produktionsaufträge bekannt sein, welche als Mittelwert der Kundennachfrage der Validierungsmenge initialisiert werden:

$$p_{T-j} = \frac{1}{T} \sum_{t=0}^{17} y_{T-t}, \text{ für alle } j \in \{1, \dots, W\}. \quad (7.19)$$

Mit Hilfe dieser Initialwerte werden die Werte des Umlaufbestands, des disponiblen Bestands und des dynamischen Zielbestandsniveaus am Ende der Periode T durch Ausführung von (7.4), (7.5) und (7.9) bestimmt. Die Produktionsaufträge p_T am Ende der Validierungsmenge werden durch (7.17) generiert. Danach wird eine Simulation durch die iterative Ausführung von (7.6)-(7.17) durchgeführt. Es sei angemerkt, dass alle Werte für Prognosen, Bestände und Produktionsaufträge auf positive ganze Zahlen gerundet werden.

7.2.3 Evaluationskriterien

Die Lagerbestandssimulation dieser Arbeit wird durchgeführt, um die Prognosen des neuartigen Selbinationsverfahren hinsichtlich ihrer Auswirkungen auf die Logistikleistung und die Logistikkosten zu evaluieren. Wie in Abschnitt 2.1 beschrieben wurde, lässt sich die Logistikleistung durch den Servicegrad und die Lieferzeit beschreiben. Die Logistikkosten bestehen aus den Kapitalbindungs- und den Prozesskosten. Da in dieser Arbeit Lieferzeiten und Prozesskosten als gegeben angenommen werden, besteht das Ziel der folgenden Simulationsstudie in der Erreichung eines hohen Servicegrades mit möglichst geringen Sicherheitsbeständen und somit niedrigen Kapitalbindungskosten. Zur Evaluation der Leistung eines Prognoseverfahrens in einem Simulationslauf werden die folgenden drei Evaluationskriterien verwendet: (i) der erreichte Servicegrad, (ii) der durchschnittliche benötigte Sicherheitsbestand und (iii) das Verhältnis zwischen den beiden Größen. Der erreichte Servicegrad ist definiert als der Anteil der Perioden, in denen die gesamte Kundennachfrage aus dem physischen Lagerbestand erfüllt werden kann, ohne Lieferrückstellungen aufgrund von Fehlmengen zu erzeugen. Ein hoher Wert begünstigt eine hohe Kundenzufriedenheit, während ein niedriger Wert zu Lieferverzügen oder zu einem Verlust des Kunden führen kann. Der durchschnittliche benötigte Sicherheitsbestand für ein Produkt wird als mittlerer Sicherheitsbestand über alle Perioden der Testmenge bestimmt. Ein hoher Sicherheitsbestand impliziert hohe Kapitalbindungskosten. Daher liegt das Ziel der Simulation in der Erreichung eines hohen Servicegrades mit einem niedrigen durchschnittlichen Sicherheitsbestand. Wie in Abschnitt 2.1 beschrieben wurde, besteht ein Zielkonflikt zwischen diesen beiden Größen. Daher ist das Hauptevaluationskriterium der Leistungsfähigkeit eines Prognoseverfahrens in der Lagerbestandssimulation der *Quadratmittelwert* (RMS, engl. root mean square) der beiden Evaluationsgrößen

$$\text{RMS}(0, 5/0, 5) = \sqrt{\left(0, 5 \left(\frac{\bar{\alpha}}{\alpha}\right)^2 + 0, 5 \left(\frac{s_{\text{scaled}}}{\bar{s}_{\text{scaled}}}\right)^2\right)}, \quad (7.20)$$

wobei α den erreichten Servicegrad eines Prognoseverfahrens bezeichnet, s_{scaled} den benötigten Sicherheitsbestand skaliert durch die mittlere Kundennachfrage für das betrachtete Produkt und $\bar{\alpha}$ sowie $\overline{s_{\text{scaled}}}$ die durchschnittlichen Werte über alle verglichenen Prognoseverfahren darstellen. Es sei angemerkt, dass α zu maximieren ist, während s_{scaled} zu minimieren ist. Daher wird der inverse Wert des ersten Terms in (7.20) gebildet. Während ein niedriger Wert des RMS einen guten Kompromiss zwischen dem erreichten Servicegrad und dem benötigten Sicherheitsbestand und daher eine hohe Leistung des verwendeten Prognoseverfahrens anzeigt, deuten hohe Werte des RMS auf eine schlechte Leistung hin.

Um die Simulationsergebnisse zu erklären, werden die Verzerrungen der W -Schritt-Prognosen sowie die Varianzen der W -Schritt-Prognosefehler als zusätzliche Evaluationskriterien betrachtet. Die Verzerrung der W -Schritt-Prognosen eines Verfahrens für eine Zeitreihe wird als mittlere Differenz zwischen den W -Schritt-Prognosen und der W -Schritt-Nachfrage ermittelt:

$$\text{Bias}_{W,t}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{h=1}^{W+1} (\hat{y}_{t+h} - y_{t+h}). \quad (7.21)$$

Eine positive Verzerrung deutet auf die Tendenz eines Verfahrens hin, zu hohe Werte zu prognostizieren, während eine negative Verzerrung auf im Durchschnitt zu niedrige Prognosen hindeutet. Die Varianz der W -Schritt-Prognosefehler $e_{W,t}$ hat einen großen Einfluss auf die Berechnung der dynamischen Sicherheitsbestände (vgl. (7.12)), wobei eine höhere Varianz zu einem höheren Sicherheitsbestand führt.

Für den Vergleich der verschiedenen Evaluationskriterien über mehrere Zeitreihen und Prognoseverfahren werden der durchschnittliche Sicherheitsbestand, die Verzerrung der W -Schritt-Prognosen und die Varianz der W -Schritt-Prognosefehler skaliert, indem sie durch die mittlere Kundennachfrage für die betrachtete Zeitreihe dividiert werden. Die beschriebenen Lagerbestandsimulationen werden für alle $Z = 111$ Zeitreihen separat durchgeführt und die mittleren Werte der Evaluationskriterien werden bestimmt, indem die Mittelwerte über die 111 Simulationsläufe für ein ausgewähltes Prognoseverfahren sowie gewählte Werte des Zielservicegrades und der Produktionszeit gebildet werden.

7.3 Ergebnisse

Abbildung 7.3 zeigt die durchschnittlichen Quadratmittelwerte der zehn verglichenen Prognoseverfahren gemittelt über alle Zeitreihen, alle Produktionszeiten $W = 1, \dots, 4$ und den Zielservicegrad $\alpha^* = 0,9$ in der durchgeführten Simulationsstudie. Der RMS-Wert zeigt, dass Selbination den besten Kompromiss zwischen dem erreichten Servicegrad und dem benötigten Sicherheitsbestand für die Mengen der langen und aller Zeitreihen erreicht und somit die höchste Leistung in der Lagerbestandssimulation erzielt. Für die Menge der kurzen Zeitreihen, erreicht Selbination den zweitbesten RMS-Wert, knapp hinter dem Verfahren Pool-3. Die Pooling-Verfahren Pool-2 und Pool-3 erreichen für alle drei Mengen der Zeitreihen einen der ersten drei Ränge hinsichtlich der

RMS-Werte. Im Gegensatz zu den Evaluationen anhand der Prognosegüte in Kapitel 6 erreicht das Kombinationsverfahren Comb-4 schlechtere Ergebnisse in der Lagerbestandssimulation. Es erzielt einen vergleichbar guten Kompromiss zwischen dem erreichten Servicegrad und den benötigten Sicherheitsbeständen, wie ML-Sel, Val-Sel und ets, jedoch deutlich schlechtere Ergebnisse als Selbination und die Pooling-Verfahren. nnetar und srw erzielen erneut schlechtere Ergebnisse als alle anderen Verfahren. Allerdings ermittelt llknn den schlechtesten Kompromiss zwischen dem erreichten Servicegrad und den benötigten Sicherheitsbeständen für lange Zeitreihen, während es gute Ergebnisse für kurze Zeitreihen erreicht.

Zur Erklärung der RMS-Werte der verschiedenen Prognoseverfahren zeigt Tabelle 7.3 alle durchschnittlichen Evaluationsgrößen gemittelt über alle Produktionszeiten $W = 1, \dots, 4$ für den Zielservicegrad $\alpha^* = 0.9$. Für die komplette Menge aller kurzen und langen Zeitreihen erreicht Selbination den besten Kompromiss und somit die beste Leistung aller Verfahren. Es erzielt den dritthöchsten Servicegrad und benötigt dafür im Durchschnitt den drittniedrigsten Sicherheitsbestand. Nur llknn und ets erzielen leicht höhere Servicegrade, benötigen dafür aber signifikant höhere durchschnittliche Sicherheitsbestände. Die Pooling-Verfahren benötigen zwar leicht niedrigere Sicherheitsbestände als Selbination, sie erzielen aber deutlich geringere Servicegrade. Val-Sel führt zu ähnlichen aber leicht schlechteren Ergebnissen als die Pooling-Verfahren. ML-Sel erreicht einen guten Kompromiss durch den viertbesten Servicegrad mit dem fünftniedrigsten durchschnittlichen Sicherheitsbestand. Trotz einer hohen Prognosegüte (vgl. Kapitel 6) erreicht Comb-4 nur den sechsthöchsten Servicegrad und den sechsniedrigsten durchschnittlichen Sicherheitsbestand. srw und nnetar erzielen schlechtere Simulationsergebnisse als alle anderen Verfahren. Bei separater Betrachtung der Mengen der kurzen und der langen Zeitreihen lassen sich einige Unterschiede der Leistungen der Prognoseverfahren feststellen. Wie in den Evaluationen hinsichtlich der Prognosegüte, erreicht llknn gute Ergebnisse für kurze Zeitreihen und ets für lange Zeitreihen. Die Rangfolgen der anderen Prognoseverfahren verändert sich nur geringfügig mit der Länge der Zeitreihen.

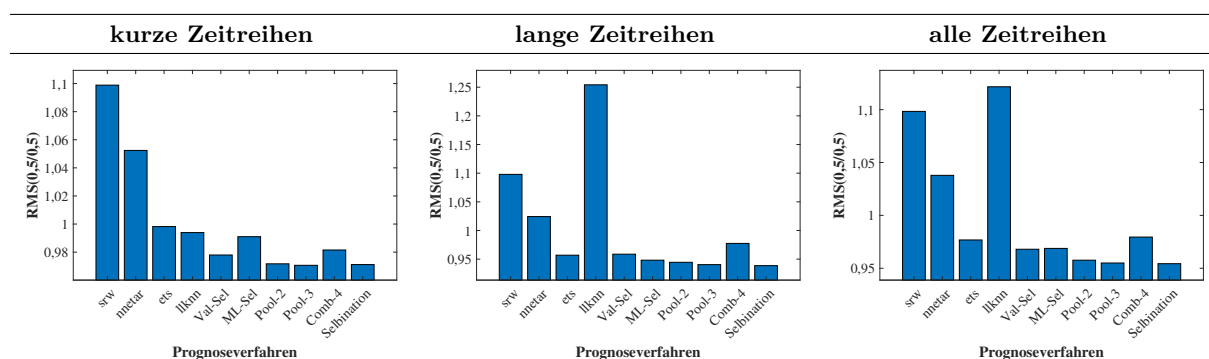


Abbildung 7.3: Balkendiagramme der $RMS(0,5/0,5)$ -Werte der verschiedenen Prognoseverfahren für kurze, lange und alle Zeitreihen gemittelt über alle Produktionszeiten

Tabelle 7.3: Durchschnittliche Evaluationsgrößen und Ränge von Selbination und den Benchmark-Verfahren gemittelt über alle Zeitreihen, für Produktionszeiten $W = 1, \dots, 4$ und einen Zielservicegrad von 0,9 (die jeweils besten Verfahren sind fettgedruckt und markiert)

(a) Durchschnittliche Leistung für die komplette Menge der kurzen und langen Zeitreihen

Durchschnittliche(r)	srw	nnetar	ets	llknn	Val-Sel	ML-Sel	Pool-2	Pool-3	Comb-4	Selbination
RMS(0,5/0,5)	1,098 9	1,038 8	0,977 6	1,122 10	0,968 4	0,969 5	0,958 3	0,955 2	0,979 7	0,954 1*
erreichter Servicegrad	0,952 5	0,940 10	0,957 2	0,967 1*	0,944 9	0,953 4	0,945 7	0,945 8	0,949 6	0,954 3
skalierter Sicherheitsbestand	0,891 9	0,796 8	0,718 7	0,932 10	0,695 4	0,703 5	0,679 2	0,675 1*	0,716 6	0,681 3
skalierte Prognoseverzerrung	-0,24 4	0,671 9	0,746 10	0,216 2	0,406 8	0,104 1	0,255 5	0,332 6	0,347 7	0,224 3
skalierte Fehlervarianz	0,184 9	0,117 8	0,102 7	0,247 10	0,092 5	0,089 4	0,087 2	0,088 3	0,101 6	0,086 1

(b) Durchschnittliche Leistung für die Menge der kurzen Zeitreihen

Durchschnittliche(r)	srw	nnetar	ets	llknn	Val-Sel	ML-Sel	Pool-2	Pool-3	Comb-4	Selbination
RMS(0,5/0,5)	1,099 10	1,052 9	0,998 8	0,994 7	0,978 4	0,991 6	0,972 3	0,971 1*	0,982 5	0,971 2
erreichter Servicegrad	0,963 8	0,965 5	0,965 6	0,972 1*	0,956 10	0,966 4	0,960 9	0,963 7	0,966 3	0,969 2
skalierter Sicherheitsbestand	0,943 10	0,875 9	0,791 8	0,790 7	0,751 3	0,780 6	0,744 1*	0,745 2	0,765 5	0,751 4
skalierte Prognoseverzerrung	0,452 1	3,352 10	1,806 9	0,650 2	1,510 6	0,720 3	1,418 5	1,607 8	1,565 7	1,029 4
skalierte Fehlervarianz	0,198 10	0,106 9	0,090 5	0,094 8	0,091 7	0,091 6	0,085 2	0,084 1	0,090 4	0,086 3

(c) Durchschnittliche Leistung für die Menge der langen Zeitreihen

Durchschnittliche(r)	srw	nnetar	ets	llknn	Val-Sel	ML-Sel	Pool-2	Pool-3	Comb-4	Selbination
RMS(0,5/0,5)	1,098 9	1,024 8	0,957 5	1,254 10	0,959 6	0,948 4	0,945 3	0,940 2	0,977 7	0,939 1*
erreichter Servicegrad	0,942 3	0,918 10	0,950 2	0,963 1*	0,934 6	0,942 4	0,933 8	0,929 9	0,934 7	0,941 5
skalierter Sicherheitsbestand	0,846 9	0,729 8	0,657 6	1,053 10	0,647 5	0,637 4	0,624 3	0,615 1*	0,674 7	0,622 2
skalierte Prognoseverzerrung	-0,83 9	-1,61 10	-0,16 2	-0,15 1	-0,53 5	-0,42 3	-0,73 7	-0,75 8	-0,69 6	-0,46 4
skalierte Fehlervarianz	0,172 9	0,127 8	0,112 7	0,378 10	0,093 5	0,088 2	0,090 3	0,091 4	0,110 6	0,086 1

Die offensichtlichsten Unterschiede zwischen Prognosegüte und Leistung in der Lagerbestandssimulation zeigen sich für die Verfahren Comb-4 und llknn. Im Vergleich zu den anderen Verfahren erreichen diese beiden Verfahren deutlich schlechtere Leistungen in der Simulation als ihre Prognosegüte vermuten ließe. Ein Grund für die hohen Sicherheitsbestände dieser Verfahren liegt in den hohen Varianzen der W -Schritt-Prognosefehler (vgl. Tabelle 7.3). Wie in (7.12) zu sehen ist, haben die Varianzen der Prognosefehler einen großen Einfluss auf die Dichteschätzung und daher auf die Berechnung der dynamischen Sicherheitsbestände in den Simulationen. Es sei angemerkt, dass alle Verfahren eine Tendenz zur Prognose zu niedriger Werte für die Menge der langen Zeitreihen und zu hoher Werte für die Menge der kurzen Zeitreihen zeigen. Neun der zehn Verfahren haben skalierte Verzerrungen zwischen -15% und -83% der mittleren Kundennachfrage für die langen sowie zwischen 45% und 181% für die kurzen Zeitreihen. nnetar zeigt besonders starke Verzerrungen der Prognosen von -161% für die langen und 335% für die kurzen Zeitreihen.

Zusätzlich zu den durchschnittlichen Ergebnissen in Tabelle 7.3 stellt Abbildung 7.4 die Zielkonflikte zwischen skalierten Sicherheitsbeständen und erreichten Servicegraden für kurze, lange und alle Zeitreihen für die verschiedenen Produktionszeiten und Zielservicegrade dar. Da das Ziel der Lagerbestandssimulation in der Erreichung hoher Servicegrade mit niedrigen Sicherheitsbeständen liegt, deuten Kurven, die weiter links und weiter oben sind, als andere, auf eine bessere Leistung hin. Die Kurven zeigen, dass Selbination, Pool-3, Pool-2 und ML-Sel unter den besten Verfahren für alle Zeitreihenmengen und alle Produktionszeiten sind. Comb-4

erzielt eine geringere Leistung als diese vier Verfahren. llkn erreicht gute Ergebnisse für kurze Zeitreihen, jedoch schlechte für lange Zeitreihen, für die es zwar hohe Servicegrade erzielt, aber sehr hohe Sicherheitsbestände benötigt. ets erreicht bessere Ergebnisse für lange Zeitreihen als für kurze. Val-Sel benötigt geringe Sicherheitsbestände aber erzielt keine hohen Servicegrade. nnetar und srw erreichen keine zufriedenstellende Leistung im Vergleich zu den anderen Verfahren. Alle Prognoseverfahren benötigen höhere skalierte Sicherheitsbestände für die kurzen Zeitreihen als für die langen und die benötigten Sicherheitsbestände wachsen mit der Produktionszeit.

Insgesamt zeigen die Simulationsergebnisse dieses Abschnitts, dass das Selbinationsverfahren eine höhere Leistung in der Lagerbestandsplanung erzielt als die gängigen Selektions-, Kombinations- und Pooling-Verfahren. Somit wird FT 4 bestätigt.

7.4 Zusammenfassung

Dieses Kapitel evaluierte die Leistung des Selbinationsverfahrens im Vergleich zu den Benchmark-Verfahren im Rahmen einer Lagerbestandssimulationsstudie. Zu diesem Zweck wurde ein ereignisdiskretes Simulationsmodell eines Lagerbestandssystems mit einem (R, S_t) -Bestellrhythmusverfahren mit angestrebtem Zielbestandsniveau implementiert. Es wurden insgesamt 13320 Simulationen durchgeföhrt, um verschiedene Konfigurationen hinsichtlich des Zielservicegrades und der Wiederbeschaffungszeit anhand des Datensatzes der NN3-Competition zu untersuchen.

In der durchgeföhrt Simulationenstudie erreichte das neuartige Selbinationsverfahren dieser Arbeit im Durchschnitt die höchste Leistung hinsichtlich des Zielkonfliktes aus Logistikleistung und Logistikkosten, welcher als durchschnittlicher Quadratmittelwert der erreichten Servicegrade und der skalierten benötigten Sicherheitsbestände ermittelt wurde. Die Ergebnisse zeigen, dass das Selbinationsverfahren eine bessere Eignung zum Einsatz in der Lagerbestandsplanung besitzt als gängige Selektions-, Kombinations- und Pooling-Verfahren. Somit wird FT 4 bestätigt.

Ein weiteres Ergebnis der Simulationenstudie dieses Kapitels ist, dass neben dem Haupteinflussfaktor der Prognosegüte eines Verfahrens auch die Varianz der W -Schritt-Prognosefehler einen großen Einfluss auf die Leistung eines Prognoseverfahrens hinsichtlich der Lagerbestandsgrößen hat. Diese wirkt sich auf die Berechnung des Sicherheitsbestandes aus. Einige der Selektionsverfahren mit einer vorgegebenen Verfahrensklasse erreichen im Durchschnitt hohe Servicegrade, aber all diese Verfahren benötigen hohe durchschnittliche Sicherheitsbestände. Daher erreichen sie keinen guten Kompromiss zwischen den beiden Zielgrößen des erreichten Servicegrades und des benötigten Sicherheitsbestandes. Das Kombinationsverfahren Comb-4, welches im Durchschnitt eine hohe Prognosegüte in den Evaluationsstudien in Kapitel 6 erreichte, erzielt in der Lagerbestandssimulation dieses Kapitels lediglich eine leicht bessere Leistung als die Selektionsverfahren mit vorgegebener Verfahrensklasse. Bessere Kompromisse erreichen die Selektionsverfahren mit mehreren möglichen Verfahrensklassen Val-Sel und ML-Sel sowie die Pooling-Verfahren Pool-2 und Pool-3, welche leicht niedrigere Servicegrade erzielen aber signifikant niedrigere durchschnittli-

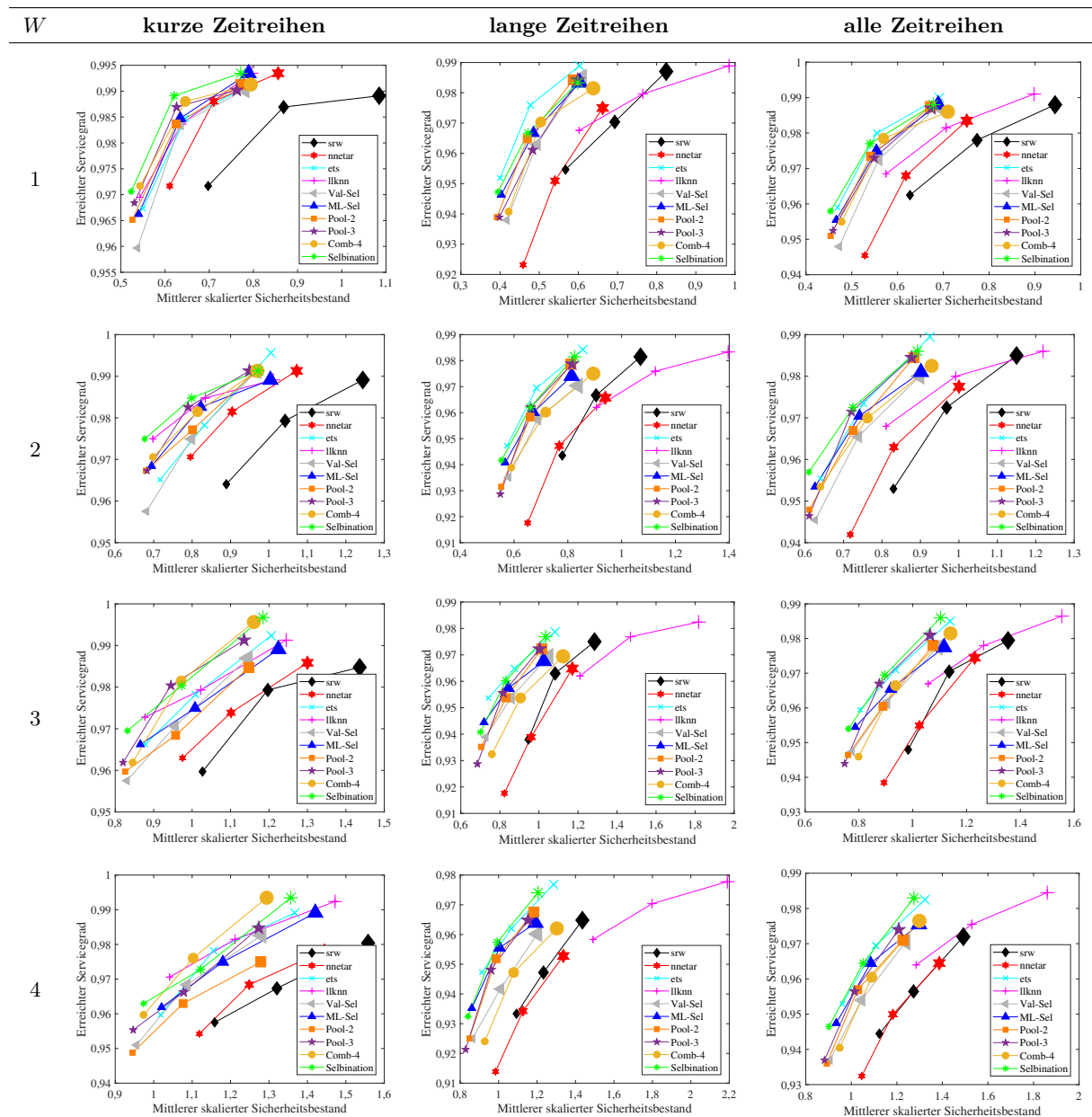


Abbildung 7.4: Zielkonflikte der erreichten Servicegrade und der durchschnittlichen skalierten Sicherheitsbestände für die verschiedenen Prognoseverfahren für kurze, lange und alle Zeitreihen und verschiedene Zielservicegrade von 0,9; 0,95 und 0,99 (größere Symbole bezeichnen die Untersuchungen zu einem höheren Zielservicegrad)

che Sicherheitsbestände benötigen. Den besten Kompromiss zwischen erreichtem Servicegrad und benötigten Sicherheitsbeständen und somit die höchste Leistung in der Lagerbestandssimulation erreicht das Selbinationverfahren. Dies zeigt, dass das hybride Verfahren zwischen Selektion und Kombination neben einer hohen Prognosegüte ebenfalls großes Potenzial zur Erreichung einer hohen Leistung hinsichtlich einer Lagerbestandsplanung im Vergleich zu klassischen Selektions-, Kombinations- und Pooling-Verfahren bietet.

Kapitel 8

Schlussbetrachtung

„My interest is in the future because I am going to spend the rest of my life there.“

Charles F. Kettering, US-amerikanischer Ingenieur (1876 - 1958)

8.1 Zusammenfassung der Hauptergebnisse

In der vorliegenden Arbeit wurde ein hybrides Meta-Lernverfahren zur automatischen Selektion oder Kombination geeigneter Nachfrageprognoseverfahren vorgestellt. Das sogenannte Selbina-tionsverfahren wurde in umfassenden empirischen Studien anhand realer industrieller Zeitreihen-daten im Vergleich zu gängigen Benchmark-Verfahren evaluiert.

Nachdem in Kapitel 1 die Motivation, die Forschungsthese und der Aufbau der Arbeit dargestellt wurden, stellte Kapitel 2 die Grundlagen der Absatzplanung vor. Zunächst wurden ein hoher Servicegrad und niedrige Kapitalbindungskosten als relevante Zielgrößen produzierender Unternehmen definiert, hinsichtlich derer die Verfahren dieser Arbeit in den folgenden Kapiteln evaluiert wurden. Darauffolgend wurden die Bedeutung und der Prozess der Absatzplanung produzierender Unternehmen erläutert. Nachfrageprognosen sind insbesondere von signifikanter Bedeutung für Unternehmen, die Produkte auf Lager produzieren, wie es beispielsweise häufig in der Konsumgüterindustrie der Fall ist. Da zumeist Prognosen für zahlreiche Produkte in regelmäßigen Intervallen benötigt werden, sind automatische Prognoseverfahren notwendig.

In Kapitel 3 wurden die Grundlagen und der Stand der Forschung zur Zeitreihenprognose beschrieben. Auf Basis der Grundlagen des überwachten Lernens wurden Zeitreihenprognose-probleme sowie Kriterien zur Evaluation von Prognosen für einzelne und mehrere Zeitreihen erläutert. Es wurden Random Walks, ARIMA-Modelle, exponentielle Glättungsmodelle, lokale k -nächste-Nachbarn-Modelle sowie neuronale Netze beschrieben. Hierbei wurden jeweils automa-tische Verfahren definiert, die zum Training eines Prognosemodells anhand der Trainingsmenge einer Zeitreihe verwendet werden können. Mit Hilfe eines trainierten Prognosemodells lassen sich zukünftige Zeitreihenwerte prognostizieren. Als Grundlage für die Identifikation der gängigsten Modellklassen im Stand der Forschung dienten die Ergebnisse der beiden aktuellsten Vergleichs-

studien für Prognoseverfahren, der M3- und der NN3-Competition. Es wurde festgestellt, dass keine stets beste Modellklasse zur Prognose der Kundennachfrage produzierender Unternehmen existiert. Dies motivierte die Untersuchung von Selektions- und Kombinationsverfahren über mehrere mögliche Modellklassen.

Kapitel 4 stellte den Stand der Forschung zur Selektion geeigneter Zeitreihenprognosemodelle dar. Nach einer formalen Definition des Prognosemodellselektionsproblems wurden die theoretischen Grundlagen der Selektion und Kombination von Prognosemodellen erläutert und umfassend in Form einer neuartigen mathematischen Notation präsentiert. Hierbei wurde in der gesamten Arbeit davon ausgegangen, dass die Selektion eines Prognosemodells durch die Selektion eines Prognoseverfahrens und die anschließende Verwendung dieses Verfahrens zum Training eines Modells erreicht wird. Aus einer detaillierten Literaturanalyse wurde eine neuartige Taxonomie abgeleitet, um verschiedene Ansätze zur Selektion von Prognosemodellen zu klassifizieren. Diese unterscheidet zwischen Selektionsverfahren, die aus einer Menge von M Prognoseverfahrenskandidaten ein Verfahren selektieren, Kombinationsverfahren, die alle M Verfahren verwenden und ihre Prognosen kombinieren, sowie Pooling-Verfahren, die zunächst eine festgelegte Anzahl M_{sel} (mit $1 < M_{\text{sel}} < M$) an Verfahren auswählen und diese danach kombinieren. Zudem unterscheidet die Taxonomie die Selektion von Verfahren aus einer vorgegebenen oder mehreren möglichen Verfahrensklassen. Abschließend wurden die Best Practices und die Limitierungen der bestehenden Ansätze erläutert. Hierbei wurde festgestellt, dass bislang kein hybrides Verfahren existiert, welches automatisch anhand der Charakteristika einer vorliegenden Zeitreihe entscheidet, ob ein einzelnes Prognoseverfahren selektiert oder mehrere Verfahren kombiniert werden sollten. Dies motivierte die Entwicklung des neuartigen Selbinationsverfahrens der vorliegenden Arbeit. Zur Ermittlung der Leistung des Selbinationsverfahrens wurde auf Basis der Best Practices der Literaturanalyse eine große Menge an Benchmark-Verfahren definiert, welche Selektions-, Kombinations- und Pooling-Verfahren mit vorgegebenen oder mit mehreren möglichen Verfahrensklassen umfasst, wobei sowohl Vergleiche von Validierungsfehlern als auch Meta-Lernansätze als Selektionskriterien verwendet wurden.

In Kapitel 5 wurde das neuartige Selbinationsverfahren dieser Arbeit beschrieben, ein hybrides Meta-Lernverfahren zur automatischen Selektion oder Kombination von M_{sel} geeigneten Prognoseverfahren aus einer Menge \mathbf{A} von M Verfahrenskandidaten. Der Ansatz besteht aus einem Meta-Modell zur Bestimmung von Eignungswahrscheinlichkeiten für Prognoseverfahrenskandidaten und einer Selektionsfunktion, welche auf Basis einer sogenannten Hybriditätsgrenze ein oder mehrere Prognoseverfahren selektiert. Das Meta-Modell ist ein Ensemble aus neuronalen Netzen, welche als Klassifikationsmodelle anhand des Datensatzes der monatlichen industriellen Zeitreihen der M3-Competition ohne die Teilmenge der Daten der NN3-Competition trainiert wurden. Der Datensatz der NN3-Competition wurde für die Evaluationsstudien der folgenden Kapitel vorgehalten. Zur Beschreibung vorliegender Zeitreihen werden Merkmalsextraktionsfunktionen verwendet, um charakteristische Zeitreihenmaße zu bestimmen. Hierzu wurden in dieser Arbeit

neben gängigen Zeitreihenmerkmalen aus dem Stand der Forschung zusätzlich zwei neue Mengen von Landmarkers eingeführt, welche die Verhältnisse und Differenzen der Validierungsfehler der Verfahrenskandidaten beschreiben. Basierend auf den Meta-Merkmalen als Input schätzt das Meta-Modell die Eignungswahrscheinlichkeiten der Prognoseverfahrenskandidaten, für die in der vorliegenden Arbeit automatische Verfahren der gängigsten Verfahrensklassen verwendet wurden. Zur Selektion eines oder mehrerer Prognoseverfahren auf Basis der geschätzten Eignungswahrscheinlichkeiten wird eine Selektionsfunktion verwendet, die auf einer Hybriditätsgrenze beruht. Die Hybriditätsgrenze Ω_{\min} bestimmt den Wert, den die kumulative Summe der Eignungswahrscheinlichkeiten der zu selektierenden Prognoseverfahren mindestens ergeben muss. Für die Wahl eines kleinen Wertes der Hybriditätsgrenze werden tendenziell wenige Prognoseverfahren selektiert, während große Werte die Selektion mehrerer Verfahren begünstigen. Die Spezialfälle der Selektion eines Verfahrens und der Kombination aller Verfahrenskandidaten werden durch die Konfigurationen $\Omega_{\min} = 0$ und $\Omega_{\min} = 1$ erreicht. Für eine Konfiguration zwischen diesen beiden Grenzen ergibt sich datengetrieben die Selektion eines oder mehrerer Prognoseverfahren. Im Gegensatz zu gängigen Verfahren aus dem Stand der Forschung wird also nicht im Voraus festgelegt, ob eine Selektion ($M = 1$) oder eine Kombination ($M \geq 2$) durchgeführt werden soll, sondern das Selbinationsverfahren trifft diese Entscheidung selbstständig auf Basis der vorhandenen Daten. Damit vereint das Verfahren die Vorteile beider Ansätze.

Kapitel 6 evaluierte das Selbinationsverfahren in einer umfangreichen empirischen Studie hinsichtlich der Prognosegüte und des notwendigen Rechenaufwands im Vergleich zu gängigen Benchmark-Verfahren sowie bezüglich der Robustheit gegenüber der Konfiguration der Parameter des Verfahrens. Für die Evaluationsstudien wurde der Datensatz der monatlichen industriellen Zeitreihen der NN3-Competition verwendet, ein gängiger Datensatz, der bereits in diversen Studien verwendet wurde, um neue Prognoseverfahren zu evaluieren. Um eine eindeutige Konfiguration des Selbinationsverfahrens zu bestimmen, müssen die Menge der Prognoseverfahrenskandidaten \mathbf{A} , die Menge der Merkmalsextraktionsfunktionen \mathbf{F} und die Hybriditätsgrenze Ω_{\min} festgelegt werden. Die Robustheitsuntersuchungen ergaben, dass verschiedene Konfigurationen des Verfahrens zu einer ähnlichen Prognosegüte führen, wenn einige Vorgaben beachtet werden. Die Menge \mathbf{A} sollte mindestens je einen Kandidaten aus zwei verschiedenen Verfahrensklassen enthalten, es sollte eine Merkmalsselektion durchgeführt werden und für die Hybriditätsgrenze sollte ein Wert zwischen $\frac{1}{M} < \Omega_{\min} < 1$ gewählt werden, um das volle Potenzial des Verfahrens auszuschöpfen. Unter diesen Voraussetzungen ist das Selbinationsverfahren robust gegen die Wahl der Parameter und erzielt eine hohe Prognosegüte. Für die empirischen Vergleichsstudien dieser Arbeit wurde daher eine Konfiguration des Selbinationsverfahrens mit den vier Prognoseverfahrenskandidaten llknn, ets, nnetar und srw, den 25 Merkmalsextraktionsfunktionen nach einer Merkmalsselektion in Tabelle 5.2 sowie der Hybriditätsgrenze $\Omega_{\min} = 0,5$ gewählt. In diversen Experimenten mit rollierenden Prognosen über verschiedene Prognosehorizonte von unterschiedlichen Ursprüngen erzielte das Selbinationsverfahren signifikant bessere Prognoseergebnisse als alle Selektions-

Kombinations- und Pooling-Verfahren, die als Benchmarks verwendet wurden. Das Verfahren ermittelte sowohl die besten Prognosen für kurze als auch für lange und die Menge aller betrachteten Zeitreihen. Dieses Ergebnis wurde zudem in einer weiteren Evaluationsstudie bestätigt, in der eine Teilnahme des Selbinationsverfahrens unter den Originalbedingungen der NN3-Competition simuliert wurde. Dabei erreichte es einen niedrigeren durchschnittlichen Prognosefehler als alle Teilnehmer und Benchmarks der NN3- sowie alle Teilnehmer der M3-Competition. Somit hätte das Selbinationsverfahren die NN3-Competition gewonnen. Die Evaluationen hinsichtlich des Rechenaufwands zeigten, dass das Selbinationsverfahren zwar den größten Aufwand aller verglichenen Verfahren für das Training eines finalen Prognosemodells benötigt, jedoch einen signifikant niedrigeren Aufwand als das Kombinationsverfahren Comb-4 sowie das Pooling-Verfahren Pool-3 pro prognostiziertem Zeitreihenwert. Da ein Prognosemodell in der Regel lediglich einmalig für eine Zeitreihe trainiert werden muss, der Aufwand pro prognostiziertem Wert jedoch deutlich häufiger anfällt, lässt sich folgern, dass das Selbinationsverfahren effizienter hinsichtlich des Rechenaufwandes ist, als die verglichenen Kombinations- und Pooling-Verfahren. Neben den beschriebenen Hauptergebnissen wurden aus den Evaluationsstudien weitere Erkenntnisse abgeleitet. Die Untersuchungen zeigten, dass Meta-Lernansätze ein besseres Selektionskriterium darstellen als die gängigerweise verwendeten Vergleiche von Validierungsfehlern. Zudem erreichten Selektions- und Kombinationsverfahren mit mehreren möglichen Verfahrensklassen bessere Prognoseergebnisse als Verfahren mit vorgegebener Klasse.

Kapitel 7 evaluierte das Selbinationsverfahren im Vergleich zu den beschriebenen Benchmark-Verfahren hinsichtlich verschiedener Leistungskennzahlen in einer Lagerbestandssimulationsstudie. Es wurde ein ereignisdiskretes Simulationsmodell eines Lagerbestandssystems mit einem Bestellrhythmusverfahren mit angestrebtem Zielbestandsniveau implementiert. In insgesamt 13320 Simulationsläufen wurden verschiedene Konfigurationen hinsichtlich des Zielservicegrades und der Wiederbeschaffungszeit anhand des Datensatzes der NN3-Competition untersucht. Das Selbinationsverfahren erreichte auch in dieser Studie im Durchschnitt die höchste Leistung aller Verfahren. Es erzielte den besten Kompromiss hinsichtlich des Zielkonfliktes aus Logistikleistung und Logistikkosten, welcher durch den durchschnittlichen Quadratmittelwert der erreichten Servicegrade und der skalierten benötigten Sicherheitsbestände ermittelt wurde. Ein weiteres Ergebnis der Simulationsstudie ist, dass neben dem Haupteinflussfaktor der Prognosegüte eines Verfahrens auch die Varianz der Prognosefehler einen großen Einfluss auf die Leistung eines Prognoseverfahrens hinsichtlich der Lagerbestandsgrößen hat. Diese wirkt sich auf die Berechnung des Sicherheitsbestandes aus. Im Vergleich zu den Benchmark-Verfahren, welche entweder hohe Servicegrade erzielten aber dafür hohe Sicherheitsbestände benötigten oder umgekehrt, erreichte das Selbinationsverfahren im Durchschnitt den besten Kompromiss zwischen den beiden Zielgrößen und somit die höchste Leistung in der Lagerbestandssimulation für verschiedene Werte der Wiederbeschaffungszeit und des Zielservicegrades. Dies zeigt, dass es neben einer hohen Prognosegüte ebenfalls großes Potenzial für den Einsatz in der Lagerbestandsplanung bietet.

8.2 Beantwortung der Forschungsthesen

Die Selektion geeigneter Nachfrageprognoseverfahren für die Produktionsplanung wurde in der vorliegenden Arbeit umfassend betrachtet. Nach einer Darstellung des Standes der Forschung zur Ermittlung der Eignung verschiedener Modellklassen für Zeitreihenprognoseprobleme wurden die Eignungen verschiedener Kriterien zur Selektion von Prognoseverfahren verglichen. Basierend auf den Erkenntnissen dieser Untersuchungen wurde das neuartige Selbinationsverfahren dieser Arbeit entwickelt und sowohl hinsichtlich der Prognosegüte und des Rechenaufwands als auch im Rahmen einer Lagerbestandssimulation evaluiert. Für jeden dieser Aspekte wurde in Abschnitt 1.2 eine Forschungsthese definiert. Im Folgenden werden diese Forschungsthesen beantwortet.

FT 1. Es existiert keine stets beste Modellklasse zur Prognose der Kundennachfrage produzierender Unternehmen.

Die erste Forschungsthese ist naheliegend, da das sogenannte No Free Lunch Theorem besagt, dass kein universell bestes Lernverfahren existiert, welches für alle Probleminstanzen eines umfassenden Anwendungsgebietes bessere Ergebnisse erzielt als alle anderen möglichen Lernverfahren (vgl. Wolpert, 1996; Wolpert und Macready, 1997). Hieraus lässt sich folgern, dass kein allgemein bestes Prognoseverfahren existiert, wenn man das Anwendungsgebiet aller Zeitreihenprognoseprobleme betrachtet (vgl. Smith-Miles, 2008). Eine Einschränkung des weitreichenden Gebiets der Zeitreihenprognoseprobleme auf den Anwendungsbereich der Nachfrageprognoseprobleme könnte die Möglichkeit bieten, ein universell bestes Prognoseverfahren für diesen Teilbereich zu finden. Allerdings wurde diese Möglichkeit im Rahmen der Literaturanalyse zum Stand der Forschung hinsichtlich Zeitreihenprognoseverfahren in Abschnitt 3.3 negiert und somit die erste Forschungsthese bestätigt. Es wurden die Hauptergebnisse der beiden aktuellsten Prognosewettbewerbe, der M3- und der NN3-Competition, untersucht, deren Rangfolgen in Anhang B aufgeführt sind. Es zeigt sich, dass die Verfahren mit den besten Prognoseergebnissen jeweils nicht auf eine einzelne Modellklasse beschränkt sind. Stattdessen selektieren die besten Verfahren der Wettbewerbe entweder für jede Zeitreihe ein Prognosemodell aus mehreren möglichen Modellklassen oder sie kombinieren Modelle aus mehreren verschiedenen Klassen. Dieses Ergebnis wurde ebenfalls in der Literaturanalyse zu Selektions- und Kombinationsverfahren in Kapitel 4 bestätigt sowie in den späteren Evaluationsstudien, in denen Selektions- und Kombinationsverfahren mit mehreren möglichen Modellklassen im Durchschnitt bessere Prognoseergebnisse erzielten als Verfahren mit vorgegebener Modellklasse.

FT 2. Meta-Lernansätze eignen sich besser zur Selektion geeigneter Prognoseverfahren als Vergleiche von Validierungsfehlern.

Zur Untersuchung der zweiten Forschungsthese wurden in der vorliegenden Arbeit die beiden Benchmark-Verfahren Val-Sel und ML-Sel zur Selektion jeweils eines Prognoseverfahrens aus

der Menge der vier Prognoseverfahrenskandidaten $llknn$, ets , $nnetar$ und srw verwendet und hinsichtlich der Prognosegüte in den Evaluationsstudien verglichen. Val-Sel ermittelt eine erwartete Rangfolge basierend auf den Validierungsfehlern der Prognoseverfahrenskandidaten für eine Zeitreihe und selektiert das Verfahren mit dem niedrigsten Validierungsfehler. Im Gegensatz dazu beschreibt das Verfahren ML-Sel eine Zeitreihe durch Meta-Merkmale und nutzt ein Meta-Modell, um Eignungswahrscheinlichkeiten für die Kandidaten zu schätzen. In den Evaluationsstudien in Kapitel 6 erreichte ML-Sel im Durchschnitt signifikant bessere Prognoseergebnisse als Val-Sel. Dies zeigt, dass Meta-Lernansätze ein besseres Selektionskriterium darstellen als Vergleiche von Prognosefehlern und bestätigt somit FT 2. Den Hauptgrund für den Unterschied hinsichtlich der Selektionsgüten der beiden Verfahren liefert Tabelle 6.2, die zeigt, wie häufig die Selektionsverfahren die vier Prognoseverfahrenskandidaten selektierten. Val-Sel selektierte das lokale knn -Verfahren $llknn$ deutlich seltener als ML-Sel. Da die Validierungsmengen der vorliegenden Zeitreihen weniger Werte enthalten als die für die Prognose der zukünftigen Werte verwendeten kompletten Zeitreihen, lassen sich teilweise nur wenige nächste Nachbarn zu einem Abfragevektor bestimmen, was im Durchschnitt zu hohen Validierungsfehlern führen kann. Mit steigender Anzahl an Zeitreihenwerten vergrößert sich die Anzahl zu findender Nachbarn, wodurch die Prognosefehler des Verfahrens $llknn$ für die Testmenge in der Regel geringer sind als für die Validierungsmenge. Es zeigt sich, dass der Validierungsfehler von $llknn$ also möglicherweise kein sinnvolles Maß zur Abschätzung der Prognosefehler auf der späteren Testmenge ist. In diesem Zusammenhang ermittelt das Meta-Modell eine bessere Einschätzung der Verteilung der Eignungen der Prognoseverfahrenskandidaten als die Verfahren, die auf den Vergleichen der Validierungsfehler basieren.

FT 3. Durch ein hybrides Verfahren zwischen Selektion und Kombination lässt sich eine höhere Prognosegüte erzielen als durch gängige individuelle Selektions- oder Kombinationsverfahren.

Im Rahmen der Literaturanalyse in Kapitel 4 wurde festgestellt, dass für die bislang entwickelten Verfahren im Stand der Forschung stets im Voraus, unabhängig von den spezifischen Charakteristika einer vorliegenden Zeitreihe und des Selektionsproblems, festgelegt wird, ob ein einzelnes Prognosemodell selektiert ($M_{sel} = 1$) oder mehrere Prognosemodelle kombiniert werden sollen ($M_{sel} > 1$). Selektionsverfahren bieten das Potenzial, das beste Prognoseverfahren aus einer Menge von Kandidaten für eine vorliegende Zeitreihe zu selektieren, sofern ein dominantes Prognosemodell für die Zeitreihe existiert. Allerdings besteht dadurch das Risiko, ein schlechtes Modell zu selektieren, falls entweder das Selektionskriterium die erwartete Rangfolge der Verfahrenskandidaten schlecht ermittelt oder für mehrere Modelle eine ähnliche Prognosegüte zu erwarten ist. In solchen Fällen reduzieren Kombinationsverfahren das Risiko der Selektion eines schlechten Modells, da sich die Stärken und Schwächen der verschiedenen Modelle zu einer robusten Prognose ausgleichen. Zur Vereinigung der Vorteile beider Ansätze in einem Verfahren trifft das neuartige Selbinationsverfahren dieser Arbeit die Entscheidung zwischen der Selektion

eines oder der Kombination mehrerer Verfahrenskandidaten datengetrieben für jede Zeitreihe. Basierend auf der Verteilung der durch das Meta-Modell ermittelten Eignungswahrscheinlichkeiten der Verfahrenskandidaten für eine Zeitreihe sowie der festgelegte Hybriditätsgrenze wird eine geeignete Menge von Prognoseverfahren selektiert. In den empirischen Vergleichsstudien in Kapitel 6 erzielte das neuartige Selbinationsverfahren signifikant bessere Prognoseergebnisse als alle Selektions-, Kombinations- und Pooling-Verfahren, die als Benchmarks verwendet wurden. Das Verfahren erreichte sowohl die höchste Prognosegüte für kurze als auch für lange sowie für die Menge aller betrachteten Zeitreihen. Des Weiteren ermittelte das Selbinationsverfahren die besten Prognosen aller Teilnehmer und Benchmarks der NN3-Competition in einer Studie unter den Originalbedingungen des Wettbewerbs. Da die Robustheitsuntersuchungen am Ende von Kapitel 6 zeigten, dass das Verfahren robust gegenüber der Wahl der Parameter ist, lässt sich folgern, dass das neuartige Selbinationsverfahren dieser Arbeit, welches ein hybrides Verfahren zwischen Selektion und Kombination darstellt, im Durchschnitt eine höhere Prognosegüte erzielt als individuelle Selektions- und Kombinationsverfahren. Somit wird die dritte Forschungsthese bestätigt.

FT 4. Durch das hybride Verfahren lässt sich eine verbesserte Lagerbestandsplanung erreichen als durch gängige individuelle Selektions- und Kombinationsverfahren.

Zur Beantwortung der vierten Forschungsthese wurde das Selbinationsverfahren in Kapitel 7 hinsichtlich der Leistung in einer Lagerbestandssimulationsstudie evaluiert. Es erzielte den besten Kompromiss aus Logistikleistung und Logistikkosten in Form des durchschnittlichen Quadratmittelwertes der erreichten Servicegrade und der skalierten benötigten Sicherheitsbestände im Vergleich zu den Selektions-, Kombinations- und Pooling-Verfahren, die als Benchmarks der Studie verwendet wurden. Einige der Benchmark-Verfahren erreichten zwar leicht höhere Servicegrade als das Selbinationsverfahren, benötigten dafür aber deutlich höhere durchschnittliche Sicherheitsbestände. Andere Benchmark-Verfahren benötigten leicht niedrigere Sicherheitsbestände, erzielten damit aber deutlich niedrigere Servicegrade. Dementsprechend erreichte das Selbinationsverfahren den besten Kompromiss der beiden Zielgrößen. Die Ergebnisse der Simulationsstudie zeigen, dass das neuartige Selbinationsverfahren großes Potenzial für den Einsatz in der Lagerbestandsplanung bietet, wodurch die vierte Forschungsthese bestätigt wird.

8.3 Ausblick

Das in der vorliegenden Arbeit entwickelte Selbinationsverfahren erreichte durch seinen hybriden Ansatz zwischen Selektion und Kombination eine hohe Prognosegüte in den durchgeführten Evaluationsstudien und empfiehlt sich somit für die Verwendung in der Absatzplanung produzierender Unternehmen. Im Folgenden werden mögliche zukünftige Forschungsthemen im Zusammenhang mit dem entwickelten Verfahren beschrieben.

- **Einsatz des Verfahrens bei einem produzierenden Unternehmen.** Die Ergebnisse der Evaluationsstudien dieser Arbeit zeigen, dass das Selbinationsverfahren ein hohes Potenzial zur Selektion geeigneter Nachfrageprognoseverfahren bietet. Daher ist davon auszugehen, dass das Verfahren ebenfalls im Rahmen einer Anwendung im laufenden Prozess eines produzierenden Unternehmens eine hohe Prognosegüte erzielen sollte. In diesem Zusammenhang wäre es interessant, die Akzeptanz der Mitarbeiter hinsichtlich der Auswahl der Prognoseverfahren für verschiedene Zeitreihen zu betrachten. Dabei ließe sich die Erklärbarkeit der Selektionsergebnisse durch Verwendung von Ansätzen erklärbarer künstlicher Intelligenz (engl. explainable artificial intelligence) (vgl. Adadi und Berrada, 2018; Dosilovic et al., 2018) verbessern.
- **Erweiterung des Verfahrens um Prognoseverfahren für sporadische Nachfrage.** Das entwickelte Selbinationsverfahren lässt sich zur Prognose regelmäßiger Nachfrageentwicklungen verwenden. Sporadische Nachfragen, welche Perioden ohne eine Nachfrage enthalten und beispielsweise häufig für Ersatzteilbedarfe beobachtet werden, benötigen andere Prognoseverfahren, um geeignet prognostiziert zu werden (vgl. Syntetos und Boylan, 2001; Wallström und Segerstedt, 2010; Kourentzes, 2013, 2014; Van Wingerden et al., 2014; Lolli et al., 2017). Durch die Einbeziehung von Prognoseverfahren für sporadische Nachfrage sowie die Erweiterung der Meta-Trainingsmenge um sporadische Zeitreihen ließe sich das Selbinationsverfahren um die Eignung für sporadische Zeitreihen erweitern.
- **Transfer des Verfahrens auf andere industrielle oder betriebswirtschaftliche Anwendungsgebiete.** Das entwickelte Selbinationsverfahren besteht aus einem Meta-Modell und einer Selektionsfunktion mit einer Hybriditätsgrenze. Durch Änderung der Datenbasis ließe sich dieser generische Ansatz auf andere industrielle oder betriebswirtschaftliche Anwendungsgebiete erweitern. Tendenziell wären hierbei alle Anwendungsfälle möglich, in denen geeignete Verfahren zur Lösung bestimmter Problemstellungen selektiert oder kombiniert werden müssen. Beispielsweise ließe sich das Verfahren anpassen, um in Abhängigkeit von aktuellen Prozessdaten einer Produktionsanlage sowie den dazugehörigen Stammdaten, Verschleißdaten sowie Instandhaltungsdaten hinsichtlich durchgeführter und geplanter Wartungen und Reparaturen geeignete Prognosemodelle für eine vorausschauende Instandhaltungsplanung (engl. predictive maintenance, vgl. Peng et al., 2010; Hellingrath und Cordes, 2014; Gao et al., 2015) oder für die Prognose der Produktqualität (vgl. Lieber et al., 2013; Colledani et al., 2014) zu selektieren. Ein weiterer Anwendungsfall könnte die Selektion oder Kombination geeigneter Verfahren zur Prognose von Energiebedarfen sein (vgl. Hahn et al., 2009; Weinert et al., 2011; Hong und Fan, 2016).
- **Übertragung des Verfahrens zur Selektion oder Anpassung unüberwachter Lernverfahren.** Die Prognoseverfahren dieser Arbeit lassen sich als überwachte Lernverfahren charakterisieren, da jeder prognostizierte Wert im Nachhinein mit einem wahren Zielwert

verglichen werden kann. Zusätzlich wäre es denkbar, das Selbinationsverfahren zur Selektion oder Anpassung unüberwachter Lernverfahren zu verwenden. Solche Lernverfahren eignen sich insbesondere dann, wenn keine Zielwerte existieren, um getroffene Entscheidungen zu evaluieren, weil beispielsweise erst geeignete Kategorien gebildet werden müssen. Dieser Fall kann z.B. bei der Bildung von Ausfallursachen von Produktionsanlagen oder der Suche nach Gründen für Qualitätsprobleme eintreten. Mit Hilfe unüberwachter Lernverfahren lassen sich hierbei mehrere Datenpunkte aufgrund ihrer Ähnlichkeit clustern. In diesem Zusammenhang könnte das Selbinationsverfahren so konfiguriert werden, dass es entweder geeignete Clustering-Verfahren selektiert oder deren Parameter geeignet anpasst.

Literaturverzeichnis

- Abarbanel, H. D. I., Brown, R., und Kadtko, J. B. (1990). Prediction in chaotic nonlinear systems: Methods for time series with broadband Fourier spectra. *Physical Review A*, 41(4):1782–1807.
- Abou-Nasr, M. (2007). Time Series forecasting with Recurrent Neural Networks NN3 Competition. Technischer bericht, neural-forecasting-competition.com/NN3.
- Acar, Y. und Gardner, E. S. (2012). Forecasting method selection in a global supply chain. *International Journal of Forecasting*, 28(4):842–848.
- Adadi, A. und Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160.
- Adeodato, P. J., Arnaud, A. L., Vasconcelos, G. C., Cunha, R. C., und Monteiro, D. S. (2011). MLP ensembles improve long term prediction accuracy over single networks. *International Journal of Forecasting*, 27(3):661–671.
- Adhikari, R. und Agrawal, R. K. (2014). Performance evaluation of weights selection schemes for linear combination of multiple forecasts. *Artificial Intelligence Review*, 42(4):529–548.
- Adya, M. (2000). Corrections to rule-based forecasting: findings from a replication. *International Journal of Forecasting*, 16(1):125–127.
- Adya, M., Armstrong, J. S., Collopy, F., und Kennedy, M. (2000). An application of rule-based forecasting to a situation lacking domain knowledge. *International Journal of Forecasting*, 16(4):477–484.
- Adya, M., Collopy, F., Armstrong, J. S., und Kennedy, M. (2001). Automatic identification of time series features for rule-based forecasting. *International Journal of Forecasting*, 17(2):143–157.
- Adya, M. und Lusk, E. J. (2016). Development and validation of a rule-based time series complexity scoring technique to support design of adaptive forecasting DSS. *Decision Support Systems*, 83:70–82.
- Aggarwal, S. K., Saini, L. M., und Kumar, A. (2009). Electricity price forecasting in deregulated markets: A review and evaluation. *International Journal of Electrical Power and Energy Systems*, 31(1):13–22.

- Aguirre, L. A. und Letellier, C. (2009). Modeling nonlinear dynamics and chaos: A review. *Mathematical Problems in Engineering*, 2009:1–35.
- Ahmed, N., Atiya, A., El Gayar, N., und El-Shishiny, H. (2007). A Combined Neural Network/Gaussian Process Regression Time Series Forecasting System for the NN3 Competition. Technischer bericht, neural-forecasting-competition.com/NN3.
- Aiolfi, M. und Timmermann, A. (2006). Persistence in forecasting performance and conditional combination strategies. *Journal of Econometrics*, 135(1-2):31–53.
- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. und Csaki, F., Herausgeber, *2nd International Symposium on Information Theory*, 267–281, Budapest. Akademiai Kiado.
- Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.
- Albrecht, M., Rohde, J., und Wagner, M. (2015). Master Planning. In Stadler, H., Kilger, C., und Meyr, H., Herausgeber, *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, Kapitel 9, 155–175. Springer-Verlag, Berlin Heidelberg.
- Alvarez-Diaz, M. (2008). Exchange rates forecasting: local or global methods? *Applied Economics*, 40(15):1969–1984.
- Anandarajan, M. und Arinze, B. (1999). A Comparative Study of Multilayer Perceptron Topologies for the Selection of Forecasting Methods. *Journal of Computer Information Systems*, 39(2):75–80.
- Aras, S. und Kocakoç, p. D. (2016). A new model selection strategy in time series forecasting with artificial neural networks: IHTS. *Neurocomputing*, 174:974–987.
- Argyris, J., Faust, G., Haase, M., und Friedrich, R. (2017). *Die Erforschung des Chaos - Dynamische Systeme*. Springer Berlin Heidelberg, Berlin, 3. Auflage.
- Arinze, B., Kim, S.-L., und Anandarajan, M. (1997). Combining and selecting forecasting models using rule based induction. *Computers & Operations Research*, 24(5):423–433.
- Assimakopoulos, V. und Nikolopoulos, K. (2000). The theta model: A decomposition approach to forecasting. *International Journal of Forecasting*, 16(4):521–530.
- Babai, M. Z., Ali, M. M., Boylan, J. E., und Syntetos, A. A. (2013). Forecasting and inventory performance in a two-stage supply chain with ARIMA(0,1,1) demand: Theory and empirical analysis. *International Journal of Production Economics*, 143(2):463–471.

- Babai, M. Z. und Dallery, Y. (2009). Dynamic versus static control policies in single stage production-inventory systems. *International Journal of Production Research*, 47(2):415–433.
- Balkin, S. D. und Ord, J. K. (2000). Automatic neural network modeling for univariate time series. *International Journal of Forecasting*, 16(4):509–515.
- Banks, J., Carson, J. S., Nelson, B. L., und Nicol, D. M. (2013). *Discrete-Event System Simulation*. Pearson Education Limited, 5. Auflage.
- Barrow, D. K. und Crone, S. F. (2016). Cross-validation aggregation for combining autoregressive neural network forecasts. *International Journal of Forecasting*, 32(4):1120–1137.
- Barrow, D. K. und Kourentzes, N. (2016). Distributions of forecasting errors of forecast combinations: Implications for inventory management. *International Journal of Production Economics*, 177:24–33.
- Bates, J. M. und Granger, C. W. J. (1969). The Combination of Forecasts. *Journal of the Operational Research Society*, 20(4):451–468.
- Beadle, L. und Johnson, C. G. (2007). Composite Forecasting Strategy Using Seasonal Schemata. Technischer bericht, neural-forecasting-competition.com/NN3.
- Beliakov, G. (2006). Interpolation of Lipschitz functions. *Journal of Computational and Applied Mathematics*, 196(1):20–44.
- Bera, A. K. und Jarque, C. M. (1981). Efficient tests for normality, homoscedasticity and serial independence of regression residuals: Monte Carlo Evidence. *Economics Letters*, 7(4):313–318.
- Bergmeir, C., Hyndman, R. J., und Benítez, J. M. (2016). Bagging exponential smoothing methods using STL decomposition and Box-Cox transformation. *International Journal of Forecasting*, 32(2):303–312.
- Biggs, J. B. (1985). The Role of Metalearning in Study Processes. *British Journal of Educational Psychology*, 55(3):185–212.
- Bilgen, B. und Günther, H. O. (2010). Integrated production and distribution planning in the fast moving consumer goods industry: A block planning application. *OR Spectrum*, 32(4):927–955.
- Billah, B., Hyndman, R. J., und Koehler, A. B. (2005). Empirical information criteria for time series forecasting model selection. *Journal of Statistical Computation and Simulation*, 75(10):831–840.
- Billah, B., King, M. L., Snyder, R. D., und Koehler, A. B. (2006). Exponential smoothing model selection for forecasting. *International Journal of Forecasting*, 22(2):239–247.

- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1. Auflage.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York.
- Bontempi, G., Birattari, M., und Bersini, H. (1999). Local Learning for Iterated Time Series Prediction. In *Machine learning: proceedings of the Sixteenth International Conference (ICML'99)*, 32–38. Morgan Kaufmann Publishers.
- Box, G. E. P. und Jenkins, G. M. (1970). *Time series analysis: forecasting and control*. Holden-Day series in time series analysis. Holden-Day, San Francisco, Calif. [u.a.], 1. Auflage.
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., und Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. Wiley, 4. Auflage.
- Brazdil, P., Carrier, C. G., Soares, C., und Vilalta, R. (2008). *Metalearning: Applications to Data Mining*, Band 26. Springer Science & Business Media, 1. Auflage.
- Brazdil, P., Gama, J., und Henery, B. (1994). Characterizing the applicability of classification algorithms using meta-level learning. In Bergadano, F. und De Raedt, L., Herausgeber, *Machine Learning: ECML-94*, 83–102, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Bredahl Kock, A. und Teräsvirta, T. (2016). Forecasting Macroeconomic Variables Using Neural Network Models and Three Automated Model Selection Techniques. *Econometric Reviews*, 35(8-10):1753–1779.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.
- Brockwell, P. J. und Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer International Publishing, 3. Auflage.
- Broer, H. H. W. und Takens, F. (2011). *Dynamical Systems and Chaos*. Springer, New York, 1. Auflage.
- Brown, R. G. (1959). *Statistical Forecasting for Inventory Control*. McGraw-Hill, New York, 1. Auflage.
- Burda, M., Štěpnička, M., und Štěpničková, L. (2015). Fuzzy Rule-Based Ensemble for Time Series Prediction: Progresses with Associations Mining. In *Strengthening Links Between Data Analysis and Soft Computing*, 261–271. Springer, Cham.
- Burkov, A. (2019). *The Hundred-Page Machine Learning Book*. Andriy Burkov, Quebec City, Canada, 1. Auflage.
- Burnham, K. P., Anderson, D. R., und Burnham, K. P. (2002). *Model Selection and Multimodel Inference : A Practical Information-Theoretic Approach*. Springer-Verlag New York, 2. Auflage.

- Cao, L. und Soofi, A. S. (1999). Nonlinear deterministic forecasting of daily dollar exchange rates. *International Journal of Forecasting*, 15:421–430.
- Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. d. P., Basto, J. P., und Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers and Industrial Engineering*, 137:106024.
- Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Physica D: Nonlinear Phenomena*, 35(3):335–356.
- Chankov, S., Hütt, M. T., und Bendul, J. (2016). Synchronization in manufacturing systems: quantification and relation to logistics performance. *International Journal of Production Research*, 54(20):6033–6051.
- Chen, H. und Yao, X. (2007). Ensemble Regression Trees for Time Series Predictions. Technischer bericht, neural-forecasting-competition.com/NN3.
- Cheung, Y.-W. und Lai, K. S. (1995). Lag Order and Critical Values of the Augmented Dickey-Fuller Test. *Journal of Business & Economic Statistics*, 13(3):277–280.
- Cho, S. und Erkoc, M. (2009). Design of predictable production scheduling model using control theoretic approach. *International Journal of Production Research*, 47(11):2975–2993.
- Chopra, S. und Meindl, P. (2015). *Supply Chain Management: Strategy, Planning, and Operation*. Pearson Education, 6. Auflage.
- Chu, C.-H. und Widjaja, D. (1994). Neural network system for forecasting method selection. *Decision Support Systems*, 12(1):13–24.
- Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, 5(4):559–583.
- Cleveland, W. S. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.
- Colledani, M., Tolio, T., Fischer, A., Iung, B., Lanza, G., Schmitt, R., und Váncza, J. (2014). Design and management of manufacturing systems for production quality. *CIRP Annals - Manufacturing Technology*, 63(2):773–796.
- Collopy, F. und Armstrong, J. S. (1992). Rule-Based Forecasting: Development and Validation of an Expert Systems Approach to Combining Time Series Extrapolations. *Management Science*, 38(10):1394–1414.
- Cover, T. M. und Hart, P. E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

- Cox, D. R. und Stuart, A. (1955). Some Quick Sign Tests for Trend in Location and Dispersion. *Biometrika*, 42(1/2):80–95.
- Crone, S. F., Hibon, M., und Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, 27(3):635–660.
- Crone, S. F. und Kourentzes, N. (2010). Feature selection for time series prediction - A combined filter and wrapper approach for neural networks. *Neurocomputing*, 73(10-12):1923–1936.
- Crone, S. F. und Pietsch, S. (2007). A naïve support vector regression benchmark for the NN3 forecasting competition. In *IEEE International Conference on Neural Networks - Conference Proceedings*, 2454–2459.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.
- De Gooijer, J. G. und Hyndman, R. J. (2006). 25 Years of Time Series Forecasting. *International Journal of Forecasting*, 22(3):443–473.
- De Menezes, L. M., Bunn, D. W., und Taylor, J. W. (2000). Review of guidelines for the use of combined forecasts. *European Journal of Operational Research*, 120(1):190–204.
- de Vos, N. (2007). Multi-Resolution Time Series Forecasting Using Wavelet Decomposition. Technischer bericht, neural-forecasting-competition.com/NN3.
- Deb, P. und Sefton, M. (1996). The distribution of a Lagrange multiplier test of normality. *Economics Letters*, 51(2):123–130.
- Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30.
- Dickey, D. A. und Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*, 74(366a):427–431.
- Dimri, A. P., Joshi, P., und Ganju, A. (2008). Precipitation forecast over western Himalayas using k -nearest neighbour method. *International Journal of Climatology*, 28(14):1921–1931.
- Döbel, I., Leis, M., Vogelsang, M. M., Neustroev, D., Petzka, H., Rüping, S., Voss, A., Wegele, M., und Welz, J. (2018). Maschinelles Lernen - Kompetenzen, Anwendungen und Forschungsbedarf. Technischer bericht, Fraunhofer-Gesellschaft, Sankt Augustin.
- Donner, R., Scholz-Reiter, B., und Hinrichs, U. (2008). Nonlinear characterization of the performance of production and logistics networks. *Journal of Manufacturing Systems*, 27(2):84–99.

- Donoho, D. (2017). 50 Years of Data Science. *Journal of Computational and Graphical Statistics*, 26(4):745–766.
- dos Santos, P. M., Ludermir, T. B., und Prudencio, R. B. C. (2004). Selection of Time Series Forecasting Models based on Performance Information. In *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, 366–371. IEEE.
- Dosilovic, F. K., Brcic, M., und Hlupic, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 210–215. Institute of Electrical and Electronics Engineers Inc.
- Duda, R. O., Stork, D. G., und Hart, P. E. (2001). *Pattern classification*. A Wiley-Interscience publication. Wiley, New York, 2. Auflage.
- D'yakonov, A. G. (2007). Simple kNN-Method for Times Series Prediction. Technischer bericht, neural-forecasting-competition.com/NN3.
- Elliott, G. und Timmermann, A. (2016). *Economic Forecasting*. Princeton University Press, Princeton, 1. Auflage.
- Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211.
- ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., und Bernard, A. (2013). Product variety management. *CIRP Annals - Manufacturing Technology*, 62(2):629–652.
- Engster, D. (2011). *Local-and Cluster Weighted Modeling for Prediction and State Estimation of Nonlinear Dynamical Systems*. Dissertation, Niedersächsische Staats-und Universitätsbibliothek Göttingen.
- Everitt, B. S. und Skrondal, A. (2010). *The Cambridge Dictionary of Statistics*. Cambridge Univ. Press, 4. Auflage.
- Fahrmeir, L., Heumann, C., Künstler, R., Pigeot, I., und Tutz, G. (2016). *Statistik: Der Weg zur Datenanalyse*. Springer, Berlin Heidelberg, 8. Auflage.
- Fahrmeir, L., Kneib, T., Lang, S., und Marx, B. (2013). *Regression : models, methods and applications*. Springer, Berlin Heidelberg, 1. Auflage.
- Farmer, J. und Sidorowich, J. (1987). Predicting chaotic time series. *Physical review letters*, 59(8):845–848.
- Feng, Y., D'Amours, S., und Beaugerard, R. (2008). The value of sales and operations planning in oriented strand board industry with make-to-order manufacturing system: Cross functional integration under deterministic demand and spot market recourse. *International Journal of Production Economics*, 115(1):189–209.

- Ferbar Tratar, L., Mojšker, B., und Toman, A. (2016). Demand forecasting with four-parameter exponential smoothing. *International Journal of Production Economics*, 181:162–173.
- Fildes, R. (1989). Evaluation of Aggregate and Individual Forecast Method Selection Rules. *Management Science*, 35(9):1056–1065.
- Fildes, R. und Petropoulos, F. (2015). Simple versus complex selection rules for forecasting many time series. *Journal of Business Research*, 68(8):1692–1701.
- Fisher, R. (1925). *Statistical Methods for Research Workers*. Oliver & Boyd, Edinburgh.
- Fisher, R. A. (1922). On the Interpretation of χ^2 from Contingency Tables, and the Calculation of P. *Journal of the Royal Statistical Society*, 85(1):87–94.
- Fisher, R. A. (1924). The Conditions Under Which χ^2 Measures the Discrepancy Between Observation and Hypothesis.
- Fix, E. und Hodges, J. L. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. Technischer bericht, University of California, Berkeley.
- Fleischmann, B. und Meyr, H. (2004). Customer Orientation in Advanced Planning Systems. In Dyckhoff, H., Lackes, R., und Reese, J., Herausgeber, *Supply Chain Management and Reverse Logistics*, 297–321. Springer, Berlin Heidelberg.
- Fleischmann, B., Meyr, H., und Wagner, M. (2015). Advanced planning. In *Supply Chain Management and Advanced Planning (Fourth Edition): Concepts, Models, Software, and Case Studies*, 71–95. Springer, Berlin, Heidelberg.
- Flores, B. E. und Pearce, S. L. (2000). The use of an expert system in the M3 competition. *International Journal of Forecasting*, 16(4):485–496.
- Flores, P., Anaya, C., Ramírez, H. M., und Morales, L. B. (2007). Automated linear modeling of time series with self adaptive genetic algorithms. In *IEEE International Conference on Neural Networks - Conference Proceedings*, 1389–1396.
- Florido, J. P., Pomares, H., Rojas, I., Urquiza, J. M., und Lopez-Gordo, M. A. (2012). A deterministic model selection scheme for incremental RBFNN construction in time series forecasting. *Neural Computing and Applications*, 21(3):595–610.
- Fonseca, R. und Gomez, P. (2016). Automatic Model Selection in Ensembles for Time Series Forecasting. *IEEE Latin America Transactions*, 14(8):3811–3819.
- Fonseca, R. und Gómez-Gil, P. (2014). Temporal validated meta-learning for long-term forecasting of chaotic time series using Monte Carlo cross-validation. In Castillo, O., Melin, P., Pedrycz, W., und Kacprzyk, J., Herausgeber, *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, Band 547, 353–367. Springer International Publishing.

- Franses, P. H. und Koehler, A. B. (1998). A model selection strategy for time series with increasing seasonal variation. *International Journal of Forecasting*, 14(3):405–414.
- Fraser, A. M. und Swinney, H. L. (1986). Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134–1140.
- Freitag, M. (2005). *Modellierung und Analyse von Produktionssystemen mit Methoden der Nichtlinearen Dynamik*. GITO-Verlag, Berlin, 1. Auflage.
- Freitag, M., Becker, T., und Duffie, N. A. (2015a). Dynamics of resource sharing in production networks. *CIRP Annals - Manufacturing Technology*, 64(1):435–438.
- Freitag, M., Kück, M., Ait Alla, A., und Lütjen, M. (2015b). Potenziale von Data Science in Produktion und Logistik: Teil 1 - Eine Einführung in aktuelle Ansätze der Data Science. *Industrie 4.0 Management*, 31(5):22–26.
- Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200):675–701.
- Friedman, M. (1940). A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *The Annals of Mathematical Statistics*, 11:86–92.
- Fritzsche, R., Gupta, J. N., und Lasch, R. (2014). Optimal prognostic distance to minimize total maintenance cost: The case of the airline industry. *International Journal of Production Economics*, 151:76–88.
- Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., und Helu, M. (2015). Cloud-enabled prognosis for manufacturing. *CIRP Annals - Manufacturing Technology*, 64(2):749–772.
- Gardner, E. S. (1990). Evaluating forecast performance in an inventory control system. *Management Science*, 36(4):490–499.
- Gardner, E. S. (1999). Note: Rule-Based Forecasting vs. Damped-Trend Exponential Smoothing. *Management Science*, 45(8):1169–1176.
- Gardner, E. S. (2006). Exponential smoothing: The state of the art-Part II. *International Journal of Forecasting*, 22(4):637–666.
- Gardner, E. S. und McKenzie, E. (1985). Forecasting Trends in Time Series. *Management Science*, 31(10):1237–1246.
- Genre, V., Kenny, G., Meyler, A., und Timmermann, A. (2013). Combining expert forecasts: Can anything beat the simple average? *International Journal of Forecasting*, 29(1):108–121.
- Geurts, M. D. und Patrick Kelly, J. (1986). Forecasting retail sales using alternative models. *International Journal of Forecasting*, 2(3):261–272.

- Geweke, J. und Amisano, G. (2011). Optimal prediction pools. *Journal of Econometrics*, 164(1):130–141.
- Gill, P. E., Murray, W., und Wright, M. H. (1981). *Practical optimization*. Academic Press, London, 1. Auflage.
- Glasser, G. J. und Winter, R. F. (1961). Critical Values of the Coefficient of Rank Correlation for Testing the Hypothesis of Independence. *Biometrika*, 48(3/4):444–448.
- Goodrich, R. L. (2000). The Forecast Pro methodology. *International Journal of Forecasting*, 16(4):533–535.
- Grambsch, P. und Stahel, W. A. (1990). Forecasting demand for special telephone services. A case study. *International Journal of Forecasting*, 6(1):53–64.
- Günther, H.-O. und Tempelmeier, H. (2012). *Produktion und Logistik*. Springer, Berlin, 9. Auflage.
- Haefner, B. und Lanza, G. (2017). Function-oriented measurements and uncertainty evaluation of micro-gears for lifetime prognosis. *CIRP Annals - Manufacturing Technology*, 66(1):475–478.
- Hahn, H., Meyer-Nieberg, S., und Pickl, S. (2009). Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research*, 199(3):902–907.
- Hartmann, H. (2002). *Materialwirtschaft : Organisation, Planung, Durchführung, Kontrolle*. Deutscher Betriebswirte-Verlag, Gernsbach, 8. Auflage.
- Hastie, T., Tibshirani, R., und Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, NY, USA, 2. Auflage.
- Hazarika, N. (2007). Time Series Prediction Using Decomposition onto a System of Random Sequence Basis Functions and a Temperature-Dependent SOFTMAX Combiner. Technischer bericht, neural-forecasting-competition.com/NN3.
- Hebb, D. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, 1. Auflage.
- Helbing, D. (2003). Modelling supply networks and business cycles as unstable transport phenomena. *New Journal of Physics*, 5(90):1–28.
- Hellingrath, B. und Cordes, A.-K. (2014). Conceptual approach for integrating condition monitoring information and spare parts forecasting methods. *Production & Manufacturing Research*, 2(1):725–737.
- Herwartz, H. (2010). A note on model selection in (time series) regression models - general-to-specific or specific-to-general? *Applied Economics Letters*, 17(12):1157–1160.

- Herwartz, H. (2011a). Forecast accuracy and uncertainty in applied econometrics: a recommendation of specific-to-general predictor selection. *Empirical Economics*, 41(2):487–510.
- Herwartz, H. (2011b). Keep it simple: on specific-to-general predictor selection for time series forecasting in the short, medium and long run. *Journal of Statistical Computation and Simulation*, 81(8):931–943.
- Hewamalage, H., Bergmeir, C., und Bandara, K. (2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427.
- Hochreiter, S. und Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Holt, C. C. (1957). Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages,. Technischer bericht, Carnegie Institute of Technology, Pittsburgh, USA.
- Hong, T. und Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.
- Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257.
- Hornik, K., Stinchcombe, M., und White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- Hsu, H.-L., Ing, C.-K., und Tong, H. (2019). On model selection from a finite family of possibly misspecified time series models. *The Annals of Statistics*, 47(2):1061–1087.
- Hurvich, C. M. und Tsai, C. L. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307.
- Hwarng, H. B. und Xie, N. (2008). Understanding supply chain dynamics: A chaos perspective. *European Journal of Operational Research*, 184(3):1163–1178.
- Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14.
- Hyndman, R. J., Akram, M., und Archibald, B. C. (2008a). The admissible parameter space for exponential smoothing models. *Annals of the Institute of Statistical Mathematics*, 60(2):407–426.

- Hyndman, R. J., Akram, M., Bergmeir, C., und O'Hara-Wild, M. (2018a). Mcomp - R package.
- Hyndman, R. J. und Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts, Melbourne, Australia, 2. Auflage.
- Hyndman, R. J., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., und Yasmeeen, F. (2018b). *forecast: Forecasting functions for time series and linear models - R package*.
- Hyndman, R. J. und Fan, Y. (1996). Sample Quantiles in Statistical Packages. *American Statistician*, 50(4):361–365.
- Hyndman, R. J. und Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, 27(3):1–22.
- Hyndman, R. J., Koehler, A., Ord, K., und Snyder, R. (2008b). *Forecasting with Exponential Smoothing*. Springer-Verlag Berlin Heidelberg, 1. Auflage.
- Hyndman, R. J., Koehler, A. B., Snyder, R. D., und Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454.
- Ilies, I., Jaeger, H., Kosuchinas, O., Rincon, M., Sakenas, V., und Vaskevicius, N. (2007). Stepping forward through echoes of the past: forecasting with echo state networks, Short report on the winning entry to the NN3 financial forecasting competition. Technischer bericht, neural-forecasting-competition.com/NN3.
- Inoue, A. und Kilian, L. (2006). On the selection of forecasting models. *Journal of Econometrics*, 130(2):273–306.
- Iša, J. (2007). Growing fuzzy inference neural network applied to the NN3 neural forecasting competition. Technischer bericht, neural-forecasting-competition.com/NN3.
- Jacobs, F. R., Berry, W., Whybark, D. C., und Vollmann, T. (2011). *Manufacturing Planning and Control for Supply Chain Management*. McGraw-Hill Education, New York.
- Jaeger, H. und Haas, H. (2004). Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80.
- Jain, C. L. und Malehorn, J. (2012). *Fundamentals of Demand Planning and Forecasting*. Graceway Publishing Company, New York, 1. Auflage.
- Jarque, C. M. und Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3):255–259.

- Jarque, C. M. und Bera, A. K. (1987). A Test for Normality of Observations and Regression Residuals. *International Statistical Review / Revue Internationale de Statistique*, 55(2):163–172.
- Jayawardena, A., Li, W., und Xu, P. (2002). Neighbourhood selection for local modelling and prediction of hydrological time series. *Journal of Hydrology*, 258(1-4):40–57.
- Jordan, M. I. (1986). Serial Order: A Parallel Distributed Processing Approach. Technischer bericht, Institute for Cognitive Science, San Diego.
- Jursa, R. und Rohrig, K. (2008). Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. *International Journal of Forecasting*, 24(4):694–709.
- Kantz, H., Holstein, D., Ragwitz, M., und K. Vitanov, N. (2004). Markov chain model for turbulent wind speed data. *Physica A: Statistical Mechanics and its Applications*, 342(1-2):315–321.
- Kantz, H. und Schreiber, T. (2004). *Nonlinear Time Series Analysis*. Cambridge University Press, Cambridge, 2. Auflage.
- Kaplan, A. und Haenlein, M. (2019). Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, 62(1):15–25.
- Katzorke, I. und Pikovsky, A. (2006). Chaos and complexity in a simple model of production dynamics. *Discrete Dynamics in Nature and Society*, 5(3):179–187.
- Kendall, M. (1975). *Rank Correlation Methods*. Charles Griffin, London, 4. Auflage.
- Kennel, M. B., Brown, R., und Abarbanel, H. D. I. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403–3411.
- Khan, A. R., Mahmood, A., Safdar, A., Khan, Z. A., und Khan, N. A. (2016). Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renewable and Sustainable Energy Reviews*, 54:1311–1322.
- Kilger, C. und Wagner, M. (2015). Demand Planning. In Stadtler, H., Kilger, C., und Meyr, H., Herausgeber, *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, 125–154. Springer-Verlag Berlin Heidelberg.
- Klug, F. (2016). Analysing bullwhip and backlash effects in supply chains with phase space trajectories. *International Journal of Production Research*, 54(13):3906–3926.
- Kolassa, S. (2011). Combining exponential smoothing forecasts using Akaike weights. *International Journal of Forecasting*, 27(2):238–251.

- Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114(5):953–956.
- Kourentzes, N. (2012). Nemenyi tests.
- Kourentzes, N. (2013). Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, 143(1):198–206.
- Kourentzes, N. (2014). On intermittent demand model optimisation and selection. *International Journal of Production Economics*, 156:180–190.
- Kourentzes, N. (2017). *nnfor: Time Series Forecasting with Neural Networks - R package*.
- Kourentzes, N., Barrow, D., und Petropoulos, F. (2019). Another look at forecast selection and combination: Evidence from forecast pooling. *International Journal of Production Economics*, 209:226–235.
- Kreuter, T., Scavarda, L. F., Thomé, A. M. T., Hellingrath, B., und Seeling, M. X. (2021). Empirical and theoretical perspectives in sales and operations planning.
- Kruskal, W. H. und Wallis, W. A. (1952). Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*, 47(260):583–621.
- Kück, M., Becker, T., und Freitag, M. (2016a). Emergence of Non-predictable Dynamics Caused by Shared Resources in Production Networks. *Procedia CIRP*, 41:520–525.
- Kück, M., Crone, S., und Freitag, M. (2016b). Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. In *Proceedings of the International Joint Conference on Neural Networks*, 1499–1506. IEEE.
- Kück, M. und Freitag, M. (2021). Forecasting of customer demands for production planning by local k-nearest neighbor models. *International Journal of Production Economics*, 231:107837.
- Kück, M. und Scholz-Reiter, B. (2013). A genetic algorithm to optimize lazy learning parameters for the prediction of customer demands. In *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, 160–165.
- Kück, M., Scholz-Reiter, B., und Freitag, M. (2014). Robust Methods for the Prediction of Customer Demands based on Nonlinear Dynamical Systems. *Procedia CIRP*, 19:93–98.
- Kugiumtzis, D., Lingjaerde, O. C., und Christophersen, N. (1998). Regularized local linear prediction of chaotic time series. *Physica D*, 112:344–360.

- Kurogi, S., Koyama, R., Tanaka, S., und Sanuki, T. (2007). Forecasting using first-order difference of time series and bagging of competitive associative nets. In *IEEE International Conference on Neural Networks (IJCNN)*, 166–171.
- Küsters, U. (2012). Evaluation, Kombination und Auswahl betriebswirtschaftlicher Prognoseverfahren. In Mertens, P. und Rässler, S., Herausgeber, *Prognoserechnung*, Kapitel 18, 423—467. Physica-Verlag HD, Heidelberg.
- Kwiatkowski, D., Phillips, P. C., Schmidt, P., und Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1-3):159–178.
- Law, A. M. (2015). *Simulation modeling and analysis*. McGrawHill Education, New York, NY, 5. Auflage.
- LeCun, Y., Bengio, Y., und Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- LeCun, Y. A., Bottou, L., Orr, G. B., und Müller, K. R. (2012). Efficient backprop. In Montavon, G., Orr, G., und Müller, K., Herausgeber, *Neural Networks: Tricks of the Trade*, 9–48. Springer Verlag, Berlin Heidelberg.
- Lee, H. L., Padmanabhan, V., und Whang, S. (1997). The Bullwhip Effect In Supply Chains. *Sloan Management Review*, 38(3):93–102.
- Lee, Y. S. (2014). A semi-parametric approach for estimating critical fractiles under autocorrelated demand. *European Journal of Operational Research*, 234(1):163–173.
- Lemke, C. (2008). On the benefit of using time series features for choosing a forecasting method. *2nd European Symposium on Time Series Prediction*.
- Lemke, C., Budka, M., und Gabrys, B. (2015). Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, 44(1):117–130.
- Lemke, C. und Gabrys, B. (2010a). Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(10-12):2006–2016.
- Lemke, C. und Gabrys, B. (2010b). Meta-learning for time series forecasting in the NN GC1 competition. In *International Conference on Fuzzy Systems*, 1–5.
- Lepenioti, K., Bousdekis, A., Apostolou, D., und Mentzas, G. (2020). Prescriptive analytics: Literature review and research challenges. *International Journal of Information Management*, 50:57–70.
- Levenbach, H. (2000). The PP (Autocast) System. *International Journal of Forecasting*, 16(4):536.

- Lewicke, E. T. (2007). NN3 Forecasting Competition Methodology. Technischer bericht, neural-forecasting-competition.com/NN3.
- Lieber, D., Stolpe, M., Konrad, B., Deuse, J., und Morik, K. (2013). Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. *Procedia CIRP*, 7:193–198.
- Lisi, F. und Medio, A. (1997). Is a random walk the best exchange rate predictor? *International Journal of Forecasting*, 13(2):255–267.
- Lisi, F. und Schiavo, R. A. (1999). A comparison between neural networks and chaotic models for exchange rate prediction. *Computational Statistics & Data Analysis*, 30(1):87–102.
- Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T., und Gucci, S. (2017). Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics*, 183:116–128.
- Ma, S. und Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, 288(1):111–128.
- Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., und Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1(2):111–153.
- Makridakis, S., Chatfield, C., Hibon, M., Lawrence, M., Mills, T., Ord, K., und Simmons, L. R. F. (1993). The M2-competition: A real-time judgmentally based forecasting study. *International Journal of Forecasting*, 9(1):5–22.
- Makridakis, S. und Hibon, M. (2000). The M3-competition: Results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476.
- Makridakis, S., Spiliotis, E., und Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74.
- Manary, M. P., Willems, S. P., und Shihata, A. F. (2009). Correcting heterogeneous and biased forecast error at intel for supply chain optimization. *Interfaces*, 39(5):415–427.
- Mann, H. B. (1945). Nonparametric Tests Against Trend. *Econometrica*, 13(3):245–259.
- Marwan, N. (2019). Cross Recurrence Plot Toolbox for MATLAB®.
- Marwan, N., Carmen Romano, M., Thiel, M., und Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5-6):237–329.
- Matijaš, M., Suykens, J. a. K., und Krajcar, S. (2013). Load forecasting using a multivariate meta-learning system. *Expert Systems with Applications*, 40(11):4427–4437.

- McCulloch, W. S. und Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.
- McNames, J. (1998). A nearest trajectory strategy for time series prediction. *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*, 112–128.
- McNames, J. (1999). *Innovations in local modeling for time series prediction*. Dissertation, Stanford University.
- McNames, J. (2002). Local averaging optimization for chaotic time series prediction. *Neurocomputing*, 48(1-4):279–297.
- Meade, N. (2000a). A note on the robust trend and ARARMA methodologies used in the M3 Competition. *International Journal of Forecasting*, 16(4):517–519.
- Meade, N. (2000b). Evidence for the selection of forecasting methods. *Journal of Forecasting*, 19(6):515–535.
- Meade, N. (2002). A comparison of the accuracy of short term foreign exchange forecasting methods. *International Journal of Forecasting*, 18(1):67–83.
- Meffert, H., Burmann, C., und Kirchgeorg, M. (2015). *Marketing - Grundlagen marktorientierter Unternehmensführung Konzepte - Instrumente - Praxisbeispiele*. Springer Fachmedien, Wiesbaden, 12. Auflage.
- Mélard, G. und Pasteels, J. M. (2000). Automatic ARIMA modeling including interventions, using time series expert software. *International Journal of Forecasting*, 16(4):497–508.
- Meyr, H. und Stadtler, H. (2015). Types of Supply Chains. In Stadtler, H., Kilger, C., und Meyr, H., Herausgeber, *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, Kapitel 5, 55–69. Springer-Verlag, Berlin Heidelberg, 5. Auflage.
- Meyr, H., Wagner, M., und Rohde, J. (2015). Structure of Advanced Planning Systems. In Stadtler, H., Kilger, C., und Meyr, H., Herausgeber, *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, Kapitel 5, 99–106. Springer-Verlag, Berlin Heidelberg, 5. Auflage.
- Minner, S. (2003). Multiple-supplier inventory models in supply chain management: A review. *International Journal of Production Economics*, 81-82:265–279.
- Minsky, M. und Papert, S. (1969). *Perceptrons; an introduction to computational geometry*. MIT Press.
- Montero Jimenez, J. J., Schwartz, S., Vingerhoeds, R., Grabot, B., und Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics.

- Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., und Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting*, 36(1):86–92.
- Nemenyi, P. (1962). Distribution-free multiple comparisons. *Biometrics*, 18(2):263.
- Neuneier, R. und Zimmermann, H. G. (2012). How to Train Neural Networks. In Montavon, G., Orr, G. B., und Müller, K.-R., Herausgeber, *Neural Networks: Tricks of the Trade: Second Edition*, 369–418. Springer Berlin Heidelberg.
- Newbold, P. und Granger, C. W. J. (1974). Experience with Forecasting Univariate Time Series and the Combination of Forecasts. *Journal of the Royal Statistical Society. Series A (General)*, 137(2):131.
- Nguyen, D. und Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *IJCNN. International Joint Conference on Neural Networks*, 21–26. Publ by IEEE.
- Nikolopoulos, K. I., Babai, M. Z., und Bozos, K. (2016). Forecasting supply chain sporadic demand with nearest neighbor approaches. *International Journal of Production Economics*, 177:139–148.
- Njimi, H. und Mélard, G. (2007). Automatic ARIMA modelling, using TSE-AX. Technischer bericht, neural-forecasting-competition.com/NN3.
- Noether, G. E. (1956). Two Sequential Tests Against Trend. *Journal of the American Statistical Association*, 51(275):440–450.
- Omrod, A. (2000). John Galt’s ForecastX Engine. *International Journal of Forecasting*, 16(4):535.
- Ord, K., Fildes, R., und Kourentzes, N. (2017). *Principles of Business Forecasting*. Wessex Press, Inc., New York, NY, USA, 2. Auflage.
- Papadaki, G. und Amaxopoulos, F. (2007). Forecasting using Neural Networks : A new approach to dynamic architecture network. Technischer bericht, neural-forecasting-competition.com/NN3.
- Papakostas, N., Efthymiou, K., Mourtzis, D., und Chryssolouris, G. (2009). Modelling the complexity of manufacturing systems using nonlinear dynamics approaches. *CIRP Annals - Manufacturing Technology*, 58(1):437–440.
- Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, 50(302):157–175.
- Pearson, K. (1920). Notes on the History of Correlation. *Biometrika*, 13(1):25–45.

- Pegels, C. C. (1969). Exponential Forecasting: Some New Variations. *Management Science*, 15(5):311–315.
- Peña, D., Tiao, G. C., und Tsay, R. S. (2001). *A course in time series analysis*. J. Wiley, 1. Auflage.
- Peng, Y., Dong, M., und Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: A review. *International Journal of Advanced Manufacturing Technology*, 50(1-4):297–313.
- Perfileva, I., Novák, V., Pavliska, V., Dvorák, A., und Stepnicka, M. (2007). Forecasting Time Series Using Fuzzy Transform. Technischer bericht, neural-forecasting-competition.com/NN3.
- Petropoulos, F., Fildes, R., und Goodwin, P. (2016). Do big losses in judgmental adjustments to statistical forecasts affect experts behaviour? *European Journal of Operational Research*, 249(3):842–852.
- Petropoulos, F., Makridakis, S., Assimakopoulos, V., und Nikolopoulos, K. (2014). 'Horses for Courses' in demand forecasting. *European Journal of Operational Research*, 237(1):152–163.
- Petropoulos, F., Wang, X., und Disney, S. M. (2019). The inventory performance of forecasting methods: Evidence from the M3 competition data. *International Journal of Forecasting*, 35(1):251–265.
- Pfahringer, B., Bensusan, H., und Giraud-Carrier, C. (2000). Meta-Learning by Landmarking Various Learning Algorithms. *Proceedings of the Seventeenth International Conference on Machine Learning ICML2000*, 951(2000):743–750.
- Pfeifer, T. und Schmitt, R., Herausgeber (2014). *Masing Handbuch Qualitätsmanagement*. Hanser, München, 6. Auflage.
- Pfohl, H.-C. (2018). *Logistiksysteme - Betriebswirtschaftliche Grundlagen*. Springer Vieweg, Berlin, 9. Auflage.
- Poler, R. und Mula, J. (2011). Forecasting model selection through out-of-sample rolling horizon weighted errors. *Expert Systems with Applications*, 38(12):14778–14785.
- Porter, M. E. (2008). The Five Competitive Forces that Shape Strategy. *Harvard Business Review*, 86(January):78–94.
- Prabhu, V. V. und Duffie, N. A. (1995). Modelling and Analysis of Nonlinear Dynamics in Autonomous Heterarchical Manufacturing Systems Control. *CIRP Annals - Manufacturing Technology*, 44(1):425–428.

- Prak, D., Teunter, R., und Syntetos, A. (2017). On the calculation of safety stocks when demand is forecasted. *European Journal of Operational Research*, 256(2):454–461.
- Prudêncio, R. B. C. und Ludermir, T. B. (2004). Meta-learning approaches to selecting time series models. *Neurocomputing*, 61(1-4):121–137.
- Pucheta, J., Patiño, D., und Kuchen, B. (2007). Neural Networks-Based Time Series Prediction Using Long and Short Term Dependence in the Learning Process. Technischer bericht, neural-forecasting-competition.com/NN3.
- Qi, M. und Zhang, G. P. (2001). An investigation of model selection criteria for neural network time series forecasting. *European Journal of Operational Research*, 132(3):666–680.
- Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Radons, G. und Neugebauer, R., Herausgeber (2004). *Nonlinear Dynamics of Production Systems*. Wiley, Weinheim, 1. Auflage.
- Ragwitz, M. und Kantz, H. (2002). Markov models from data by simple nonlinear time series predictors in delay embedding spaces. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 65(5):1–12.
- Ramadge, P. J. (1993). Periodicity and Chaos from Switched Flow Systems: Contrasting Examples of Discretely Controlled Continuous Systems. *IEEE Transactions on Automatic Control*, 38(1):70–83.
- Rehkgugler, H. und Zimmermann, H. G. (1994). *Neuronale Netze in der Ökonomie: Grundlagen und finanzwirtschaftliche Anwendungen*. Vahlen, München, 1. Auflage.
- Reif, M., Shafait, F., Goldstein, M., Breuel, T., und Dengel, A. (2014). Automatic classifier selection for non-experts. *Pattern Analysis and Applications*, 17(1):83–96.
- Reilly, D. (2000). The AUTOBOX system. *International Journal of Forecasting*, 16(4):531–533.
- Ren, C., An, N., Wang, J., Li, L., Hu, B., und Shang, D. (2014). Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting. *Knowledge-Based Systems*, 56:226–239.
- Reuter, B. und Rohde, J. (2015). Coordination and Integration. In Stadtler, H., Kilger, C., und Meyr, H., Herausgeber, *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*, Kapitel 13, 241–256. Springer-Verlag, Berlin Heidelberg, 5. Auflage.

- Rice, J. R. (1976). The Algorithm Selection Problem. *Advances in Computers*, 15(C):65–118.
- Riedmiller, M. A. (1994). Rprop - Description and Implementation Details.
- Ripley, B. und Venables, W. (2015). nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models.
- Rohde, J., Meyr, H., und Wagner, M. (2000). Die Supply Chain Planning Matrix. *PPS Management*, 5(1):10–15.
- Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65–386.
- Rossi, A. L. D., de Carvalho, A. C. P. d. L. F., Soares, C., und de Souza, B. F. (2014). MetaStream: A meta-learning based method for periodic algorithm selection in time-changing data. *Neurocomputing*, 127:52–64.
- RSG Software GmbH (2016). iqast - Intelligent Forecasting.
- Rubio, G., Pomares, H., Rojas, I., und Herrera, L. J. (2011). A heuristic method for parameter selection in LS-SVM: Application to time series prediction. *International Journal of Forecasting*, 27(3):725–739.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms.
- Rumelhart, D. E., Hinton, G. E., und Williams, R. J. (1986a). Learning internal representations by error propagation. In Rumelhart, D. E. und McClelland, J. L., Herausgeber, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, 318–362. MIT Press, Cambridge, MA, USA.
- Rumelhart, D. E., Hinton, G. E., und Williams, R. J. (1986b). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.
- Russell, S. J., Russell, S., und Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson series in artificial intelligence. Pearson, Hoboken, NJ, 4. Auflage.
- Sachs, L. und Hedderich, J. (2006). *Angewandte Statistik: Methodensammlung mit R*. Springer, Berlin, 12. Auflage.
- Sauer, T. (1994). Time series prediction by using delay coordinate embedding. In Weigend, A. S. und Gershenfeld, N. A., Herausgeber, *Time Series Prediction: forecasting the future and understanding the past*, 175–193. Addison Wesley, Harlow, UK.
- Sauer, T., Yorke, J. a., und Casdagli, M. (1991). Embedology. *Journal of Statistical Physics*, 65(3):579–616.

- Schmidhuber, J. (1987). *Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook*. Diplomarbeit, Technische Universität München.
- Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, 61:85–117.
- Scholz-Reiter, B., Freitag, M., und Schmieder, A. (2002). Modelling and control of production systems based on nonlinear dynamics theory. *CIRP Annals - Manufacturing Technology*, 51(1):375–378.
- Scholz-Reiter, B., Kück, M., und Lappe, D. (2014). Prediction of customer demands for production planning - Automated selection and configuration of suitable prediction methods. *CIRP Annals - Manufacturing Technology*, 63(1):417–420.
- Scholz-Reiter, B., Tervo, J. T., und Freitag, M. (2006). Phase-synchronisation in continuous flow models of production networks. *Physica A: Statistical Mechanics and its Applications*, 363(1):32–38.
- Schutt, R. und O’Neil, C. (2013). *Doing data science: Straight talk from the frontline*. O’Reilly Media, Inc., Sebastopol, 1. Auflage.
- Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464.
- Shah, C. (1997). Model selection in univariate time series forecasting using discriminant analysis. *International Journal of Forecasting*, 13(4):489–500.
- She, D. und Yang, X. (2010). A New Adaptive Local Linear Prediction Method and Its Application in Hydrological Time Series. *Mathematical Problems in Engineering*, 2010:1–15.
- Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 5(4):13–22.
- Shmueli, G. (2010). To Explain or to Predict? *Statistical Science*, 25(3):289–310.
- Si, X. S., Wang, W., Hu, C. H., und Zhou, D. H. (2011). Remaining useful life estimation - A review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14.
- Sikora, D., Štěpnička, M., und Vavříčková, L. (2013a). Fuzzy Rule-Based Ensemble Forecasting: Introductory Study. In Kruse, R., Berthold, M. R., Moewes, C., Gil, M. Á., Grzegorzewski, P., und Hryniewicz, O., Herausgeber, *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, 379–387, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Sikora, D., Štěpnička, M., und Vavříčková, L. (2013b). On the Potential of Fuzzy Rule-Based Ensemble Forecasting. In Herrero, Á., Snášel, V., Abraham, A., Zelinka, I., Baruaque, B.,

- Quintián, H., Calvo, J. L., Sedano, J., und Corchado, E., Herausgeber, *International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions*, 487–496, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Sikorska, J. Z., Hodkiewicz, M., und Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836.
- Silver, E. R., Pyke, D. F., und Thomas, D. J. (2016). *Inventory and Production Management in Supply Chains*. Taylor & Francis, 4. Auflage.
- Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, London, 1. Auflage.
- Singh, A., Damir, B., Deep, K., und Ganju, A. (2015). Calibration of nearest neighbors model for avalanche forecasting. *Cold Regions Science and Technology*, 109:33–42.
- Smart, C. N. (2000). SmartForecasts' Automatic Forecasting System. *International Journal of Forecasting*, 16(4):536–537.
- Smith-Miles, K. A. (2008). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1):1–25.
- Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85.
- Snedecor, G. (1934). *Calculation and Interpretation of Analysis of Variance and Covariance*. Collegiate Press, Iowa, 1. Auflage.
- Snyder, R. D., Koehler, A. B., und Ord, J. K. (2002). Forecasting for inventory control with exponential smoothing. *International Journal of Forecasting*, 18(1):5–18.
- Sohl, J. E. und Venkatachalam, A. R. (1995). A neural network approach to forecasting model selection. *Information and Management*, 29(6):297–303.
- Sorjamaa, A. und Lendasse, A. (2007). Time series prediction as a problem of missing values: Application to ESTSP2007 and NN3 competition benchmarks. In *IEEE International Conference on Neural Networks (IJCNN)*, 2948–2953.
- Spearman, C. (1904). The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1):72–101.
- Štěpnička, M., Štěpničková, L., und Burda, M. (2014). Fuzzy rule-based ensemble for time series prediction: The application of linguistic associations mining. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 505–512. IEEE.

- Štěpničková, L., Štěpnička, M., und Sikora, D. (2013). Fuzzy rule-based ensemble with use linguistic associations mining for time series prediction. In *Proceedings of the 8th conference of the European Society for Fuzzy Logic and Technology*, 408–415, Paris, France. Atlantis Press.
- Stone, C. J. (1977). Consistent Nonparametric Regression. *The Annals of Statistics*, 5(4):595–645.
- Strijbosch, L. W., Syntetos, A. A., Boylan, J. E., und Janssen, E. (2011). On the interaction between forecasting and stock control: The case of non-stationary demand. *International Journal of Production Economics*, 133(1):470–480.
- Sugihara, G. und May, R. M. (1990). Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344(6268):734–741.
- Sugiura, N. (1978). Further Analysis of the Data by Anaike’s Information Criterion and the Finite Corrections. *Communications in Statistics - Theory and Methods*, 7(1):13–26.
- Surana, A., Kumara, S., Greaves, M., und Raghavan, U. N. (2005). Supply-chain networks: a complex adaptive systems perspective. *International Journal of Production Research*, 43(20):4235–4265.
- Suykens, J. A. K. und Vandewalle, J. (1998). The K.U.Leuven Time Series Prediction Competition. In *Nonlinear Modeling*, 241–253. Springer US.
- Svetunkov, I. und Boylan, J. E. (2019). State-space ARIMA for supply-chain forecasting. *International Journal of Production Research*, 1–10.
- Syntetos, A. A., Babai, Z., Boylan, J. E., Kolassa, S., und Nikolopoulos, K. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *European Journal of Operational Research*, 252(1):1–26.
- Syntetos, A. A. und Boylan, J. E. (2001). On the bias of intermittent demand estimates. *International Journal of Production Economics*, 71(1-3):457–466.
- Takens, F. (1981). Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence, Warwick 1980*, 898:366–381.
- Takens, F. (2010). Reconstruction theory and nonlinear time series analysis. In *Handbook of Dynamical Systems*, Band 3, Kapitel 7, 345–377. Elsevier B.V.
- Talagala, T. S., Hyndman, R. J., und Athanasopoulos, G. (2018). Meta-learning how to forecast time series.
- Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 16(4):437–450.

- Taylor, J. W. (2003). Exponential smoothing with a damped multiplicative trend. *International Journal of Forecasting*, 19(4):715–725.
- Tempelmeier, H. (2008). *Material-Logistik: Modelle und Algorithmen für die Produktionsplanung und -steuerung in Advanced Planning-Systemen*. Springer, 7. Auflage.
- Teti, R., Jemielniak, K., O'Donnell, G., und Dornfeld, D. (2010). Advanced monitoring of machining operations. *CIRP Annals - Manufacturing Technology*, 59(2):717–739.
- The MathWorks Inc. (2017a). MATLAB Neural NetworkToolbox Release 2017b.
- The MathWorks Inc. (2017b). MATLAB Release 2017b.
- Theodosiou, M. (2011). Disaggregation & aggregation of time series components: A hybrid forecasting approach using generalized regression neural networks and the theta method. *Neurocomputing*, 74(6):896–905.
- Theodosiou, M. und Murali, S. (2007). A hybrid forecasting approach: structural decomposition, generalized regression neural networks and theta method. Technischer bericht, neural-forecasting-competition.com/NN3.
- Thomé, A. M. T., Scavarda, L. F., Fernandez, N. S., und Scavarda, A. J. (2012). Sales and operations planning: A research synthesis.
- Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.
- Trapero, J. R., Cardós, M., und Kourentzes, N. (2019). Empirical safety stock estimation based on kernel and GARCH models. *Omega*, 84:199–211.
- Trapero, J. R., Pedregal, D. J., Fildes, R., und Kourentzes, N. (2013). Analysis of judgmental adjustments in the presence of promotions. *International Journal of Forecasting*, 29(2):234–243.
- Trigg, D. W. und Leach, A. G. (1967). Exponential Smoothing with an Adaptive Response Rate. *Journal of the Operational Research Society*, 18(1):53–59.
- van Rijn, J. N., Holmes, G., Pfahringer, B., und Vanschoren, J. (2014). Algorithm Selection on Data Streams. In Džeroski, S., Panov, P., Kocev, D., und Todorovski, L., Herausgeber, *Discovery Science*, 325–336, Cham. Springer International Publishing.
- van Rijn, J. N., Holmes, G., Pfahringer, B., und Vanschoren, J. (2015). Having a Blast: Meta-Learning and Heterogeneous Ensembles for Data Streams. In Aggarwal, C., Zhou, Z.-H., Tuzhilin, A., Xiong, H., und Wu, X., Herausgeber, *2015 IEEE International Conference on Data Mining (ICDM)*, 1003–1008, Atlantic City, NJ, USA.

- Van Wingerden, E., Basten, R. J., Dekker, R., und Rustenburg, W. D. (2014). More grip on inventory control through improved forecasting: A comparative study at three companies. *International Journal of Production Economics*, 157(1):220–237.
- Venkatachalam, A. R. und Sohl, J. E. (1999). An Intelligent Model Selection and Forecasting System. *Journal of Forecasting*, 180(June 1998).
- Vilalta, R. und Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95.
- Villarreal Marroquín, M. G., Acosta Cervantes, M. C., Martínez Flores, J. L., und Cabrera-Ríos, M. (2009). Time series: Empirical characterization and artificial neural network-based selection of forecasting techniques. *Intelligent Data Analysis*, 13(6):969–982.
- Villegas, M. A. und Pedregal, D. J. (2019). Automatic selection of unobserved components models for supply chain forecasting. *International Journal of Forecasting*, 35(1):157–169.
- Villegas, M. A., Pedregal, D. J., und Trapero, J. R. (2018). A support vector machine for model selection in demand forecasting applications. *Computers & Industrial Engineering*, 121:1–7.
- Vokurka, R. J., Flores, B. E., und Pearce, S. L. (1996). Automatic feature identification and graphical support in rule-based forecasting: a comparison. *International Journal of Forecasting*, 12(4):495–512.
- Wald, A. und Wolfowitz, J. (1940). On a Test Whether Two Samples are from the Same Population.
- Wallström, P. und Segerstedt, A. (2010). Evaluation of forecasting error measurements and techniques for intermittent demand. *International Journal of Production Economics*, 128(2):625–636.
- Wang, B., Li, Y., Luo, Y., Li, X., und Freiheit, T. (2021). Early event detection in a deep-learning driven quality prediction model for ultrasonic welding. *Journal of Manufacturing Systems*, 60:325–336.
- Wang, X. und Petropoulos, F. (2016). To select or to combine? The inventory performance of model and expert forecasts. *International Journal of Production Research*, 54(17):5271–5282.
- Wang, X., Smith, K., und Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery*, 13(3):335–364.
- Wang, X., Smith-Miles, K., und Hyndman, R. (2009). Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. *Neurocomputing*, 72(10-12):2581–2594.

- Warren Liao, T. und Chang, P. C. (2010). Impacts of forecast, inventory policy, and lead time on supply chain inventory A numerical study. *International Journal of Production Economics*, 128(2):527–537.
- Weigend, A. S. und Gershenfeld, N. A., Herausgeber (1992). *Time series prediction: Forecasting the future and understanding the past*. Addison-Wesley, Reading, MA.
- Weinert, N., Chiotellis, S., und Seliger, G. (2011). Methodology for planning and operating energy-efficient production systems. *CIRP Annals - Manufacturing Technology*, 60(1):41–44.
- Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Dissertation, Harvard University.
- Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030–1081.
- Wichard, J. D. (2011). Forecasting the NN5 time series with hybrid models. *International Journal of Forecasting*, 27(3):700–707.
- Widodo, A., Budi, I., und Widjaja, B. (2016). Automatic lag selection in time series forecasting using multiple kernel learning. *International Journal of Machine Learning and Cybernetics*, 7(1):95–110.
- Widrow, B. und Hoff, M. E. (1960). Adaptive switching circuits. Technischer bericht, Stanford University, CA, Stanford Electronics Labs.
- Wiendahl, H.-P. (2014). *Betriebsorganisation für Ingenieure*. Hanser, München, 7. Auflage.
- Wiendahl, H. P., ElMaraghy, H. A., Nyhuis, P., Zäh, M. F., Wiendahl, H. H., Duffie, N., und Brieke, M. (2007). Changeable Manufacturing - Classification, Design and Operation. *CIRP Annals - Manufacturing Technology*, 56(2):783–809.
- Wiendahl, H. P. und Scheffczyk, H. (1999). Simulation based analysis of complex production systems with methods of nonlinear dynamics. *CIRP Annals - Manufacturing Technology*, 48(1):357–360.
- Wildi, M. (2007). NN3-Forecasting Competition: An Adaptive Robustified Multi-Step-Ahead Out-Of-Sample Forecasting Combination Approach. Technischer bericht, neural-forecasting-competition.com/NN3.
- Wilding, R. D. (1998). Chaos Theory: Implications for Supply Chain Management. *The International Journal of Logistics Management*, 9(1):43–56.
- Winters, P. R. (1960). Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science*, 6(3):324–342.

- Wöhe, G., Döring, U., und Brösel, G. (2016). *Einführung in die Allgemeine Betriebswirtschaftslehre*. Vahlen, München, 26. Auflage.
- Wolpert, D. H. (1996). The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1341–1390.
- Wolpert, D. H. und Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.
- Yakowitz, S. (1987). Nearest-neighbour methods for time series analysis. *Journal of Time Series Analysis*, 8(2):235–247.
- Yan, W. (2007). Multiple-Model Fusion for Robust Time-Series Forecasting. Technischer bericht, neural-forecasting-competition.com/NN3.
- Yankov, D., DeCoste, D., und Keogh, E. (2006). Ensembles of Nearest Neighbor Forecasts. In Fürnkranz, J., Scheffer, T., und Spiliopoulou, M., Herausgeber, *Machine Learning: ECML 2006. ECML 2006. Lecture Notes in Computer Science, vol 4212*, 545–556. Springer, Berlin, Heidelberg.
- Yule, G. (1927). On a method of investigating periodicities disturbed series, with special reference to Wölfer’s sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A*, 226(636-646):267–298.
- Zell, A. (2003). *Simulation neuronaler Netze*. Oldenbourg, München, 4. Auflage.
- Zhang, G. P. und Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, 160(2):501–514.
- Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., und Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers and Industrial Engineering*, 150:106889.

Anhang A

Eigene Publikationen und Würdigung studentischer Arbeiten

Teile dieser Arbeit wurden bereits publiziert. Zudem enthält die Arbeit Ergebnisse und Einflüsse aus der Betreuung studentischer Abschlussarbeiten. Im Folgenden wird zunächst ein Auszug der eigenen Publikationen aufgelistet und darauffolgend werden die studentischen Arbeiten aufgeführt.

Eigene Publikationen mit Bezug zur vorliegenden Arbeit

- Kück, M. und Freitag, M. (2021). Forecasting of customer demands for production planning by local k-nearest neighbor models. *International Journal of Production Economics*, 231:107837.
- Kück, M., Crone, S., und Freitag, M. (2016). Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. In *Proceedings of the International Joint Conference on Neural Networks*, 1499–1506. IEEE.
- Freitag, M., Kück, M., Ait Alla, A., und Lütjen, M. (2015a). Potenziale von Data Science in Produktion und Logistik: Teil 1 - Eine Einführung in aktuelle Ansätze der Data Science. *Industrie 4.0 Management*, 31(5):22–26.
- Freitag, M., Kück, M., Ait Alla, A., und Lütjen, M. (2015b). Potenziale von Data Science in Produktion und Logistik: Teil 2 - Vorgehensweise zur Datenanalyse und Anwendungsbeispiele. *Industrie 4.0 Management*, 31(6):39–46.
- Scholz-Reiter, B., Kück, M., und Lappe, D. (2014). Prediction of customer demands for production planning - Automated selection and configuration of suitable prediction methods. *CIRP Annals - Manufacturing Technology*, 63(1):417–420.

Studentische Arbeiten mit Bezug zur vorliegenden Arbeit

Broda, E. (2018). *Adaptives Scheduling mittels simulationsbasierter Optimierung*. Masterarbeit, Universität Bremen.

Kebel, J. (2018). *Entwicklung eines neuen Frameworks zur integrierten Planung und Steuerung von Supply Chains*. Bachelorarbeit, Universität Bremen.

Anhang B

Ergebnisse der M3- und der NN3-Competition

B.1 Ergebnisse der M3-Competition

Tabelle B.1 beschreibt alle Verfahren, die an der M3-Competition teilgenommen haben (vgl. Makridakis und Hibon, 2000). Der angegebene Mean-RO-sMAPE ist der mittlere Fehler der Horizonte 1 bis 18 mit fixiertem Ursprung über die 808 monatlichen Zeitreihen der Kategorien Industry und Micro (vgl. Abschnitt 3.3).

Tabelle B.1: Ergebnisse der M3-Competition

Rang	Verfahren	Mean-RO-sMAPE	Referenz	Beschreibung	Verfahrensklassen
1	Theta	17,65	Assimakopoulos und Nikolopoulos (2000)	Dekomposition in Theta-Linien und anschließende Kombination einer einfachen exponentiellen Glättung und einer linearen Regression	Exponentielle Glättung Lineare Regression
2	ForecastPro	17,93	Goodrich (2000)	Individuelle Selektion einer Verfahrensklasse und automatische Parameteroptimierung durch das Forecast Pro System von Business Forecast Systems	Exponentielle Glättung ARIMA Gleitende Mittelwerte Croston's Methode Poissonverteilung Negative Binomialverteilung
3	ForecastX	18,66	Omrod (2000)	Individuelle Selektion eines Prognosemodells basierend auf einer Kombination verschiedener Selektionskriterien: Summe der quadrierten Residuen, Bayes Informationskriterium, rollierende Kreuzvalidierung	Exponentielle Glättung ARIMA Croston's Methode
4	Dampen	18,91	Gardner und Mckenzie (1985)	Exponentielle Glättung mit gedämpftem Trend	Exponentielle Glättung
5	Comb-S-H-D	18,92	Makridakis und Hibon (2000)	Kombination der Verfahren Single, Holt und Dampen	Exponentielle Glättung
6	B-J automatic	19,18	Goodrich (2000)	ARIMA-Modellierung nach dem Box-Jenkins-Verfahren implementiert in das Forecast Pro System von Business Forecast Systems	ARIMA
7	Automat ANN	19,20	Balkin und Ord (2000)	Individuelle Selektion zwischen drei verschiedenen Modellen basierend auf Bayes Informationskriterium: automatisch trainiertes Multilayer-Perzeptron mit einer verdeckten Schicht und Parameteroptimierung durch Backpropagation, lineare Autoregression, Random Walk	Neuronale Netze Lineare Autoregression Random Walk

Tabelle B.1: Ergebnisse der M3-Competition (Fortsetzung)

Rang	Verfahren	Mean-RO-sMAPE	Referenz	Beschreibung	Verfahrensklassen
8	RBF	19,38	Adya (2000)	Rule-Based forecasting: Regelbasiertes Expertensystem, welches basierend auf Zeitreihencharakteristika und 99 Regeln die Gewichte für eine individuelle Kombination der folgenden drei Verfahren bestimmt: Random Walk, lineare Regression, Holts lineare exponentielle Glättung	Random Walk Lineare Regression Exponentielle Glättung
9	SmartFcs	19,40	Smart (2000)	Individuelle Selektion eines Prognosemodells nach Vergleich der Validierungsfehler von vier exponentiellen Glättungs- und zwei gleitenden Mittelwertmodellen	Exponentielle Glättung Gleitende Mittelwerte
10	Flores/Pearce2	19,43	Flores und Pearce (2000)	Regelbasiertes Expertensystem, welches basierend auf Zeitreihencharakteristika und 84 Regeln ein individuelles Prognosemodell aus den folgenden selektiert und nachträgliche Änderungen ermöglicht: Einfache exponentielle Glättung, Gardners exponentielle Glättung mit gedämpftem Trend, klassische Dekomposition, gleitende Mittelwerte	Gleitende Mittelwerte Exponentielle Glättung Klassische Dekomposition
11	PP-Autocast	19,50	Levenbach (2000)	Exponentielle Glättung mit gedämpftem Trend	Exponentielle Glättung
12	Theta-sm	19,75	Assimakopoulos und Nikolopoulos (2000)	Exponentielle Glättung mit gedämpftem Trend	Exponentielle Glättung
13	Single	19,93	Makridakis und Hibon (2000)	Einfache exponentielle Glättung	Exponentielle Glättung
14	AutoBox2	20,22	Reilly (2000)	ARIMA-Modellierung nach dem Box-Jenkins-Verfahren mit nachträglicher Ausreißer-Anpassung implementiert in das Autobox-System von Automatic Forecasting Systems	ARIMA
15	Holt	20,34	Makridakis und Hibon (2000)	Holts lineare exponentielle Glättung	Exponentielle Glättung
16	Winter	20,50	Makridakis und Hibon (2000)	Holt-Winters lineare und saisonale exponentielle Glättung	Exponentielle Glättung
17	AutoBox1	20,58	Reilly (2000)	ARIMA-Modellierung nach dem Box-Jenkins-Verfahren implementiert in das Autobox-System von Automatic Forecasting Systems	ARIMA
18	Flores/Pearce1	20,70	Flores und Pearce (2000)	Regelbasiertes Expertensystem, welches basierend auf Zeitreihencharakteristika und 84 Regeln ein individuelles Prognosemodell aus den folgenden selektiert: Einfache exponentielle Glättung, Gardners exponentielle Glättung mit gedämpftem Trend, klassische Dekomposition, gleitende Mittelwerte	Gleitende Mittelwerte Exponentielle Glättung Klassische Dekomposition
19	AAM1	21,20	Mélard und Pastels (2000)	ARIMA-Modellierung nach dem Box-Jenkins-Verfahren implementiert in das TSE-AX-System	ARIMA
20	ARARMA	21,32	Meade (2000a)	ARARMA-Modellierung nach der Parzen-Methode	ARARMA
21	AAM2	21,56	Mélard und Pastels (2000)	ARIMA-Modellierung nach dem Box-Jenkins-Verfahren mit nachträglicher Ausreißer-Anpassung implementiert in das TSE-AX-System	ARIMA
22	AutoBox3	21,67	Reilly (2000)	ARIMA-Modellierung nach dem Box-Jenkins-Verfahren mit vorheriger Ausreißer-Anpassung implementiert in das Autobox-System von Automatic Forecasting Systems	ARIMA
23	Naive 2	22,18	Makridakis und Hibon (2000)	Saisonaler Random Walk	Random Walk
24	Robust-Trend	25,91	Meade (2000a)	Nichtparametrische Version von Holts linearer exponentieller Glättung mit Median-basierter Trendschätzung	Exponentielle Glättung

B.2 Ergebnisse der NN3-Competition

Tabelle B.2 beschreibt alle Verfahren, die an der NN3-Competition teilgenommen haben (vgl. Crone et al., 2011). Der angegebene Mean-RO-sMAPE ist der mittlere Fehler der Horizonte 1 bis 18 mit fixiertem Ursprung über alle 111 Zeitreihen der NN3-Competition (vgl. Abschnitt 3.3). Die Verfahren mit den Identifikationsnummern C03-C59 waren die tatsächlichen Teilnehmer an der Vergleichsstudie. Da die Vorgabe zur Teilnahme eine Verwendung von Verfahren aus den Bereichen der neuronalen Netze oder Computational Intelligence war, konnten klassische statistische Verfahren lediglich außer Konkurrenz teilnehmen. Die Identifikationsnummern B00-B17 markieren Verfahren, die außer Konkurrenz als Benchmarks an der Vergleichsstudie teilnahmen. In der Tabelle werden sowohl die Ränge über alle Verfahren als auch die Ränge der tatsächlichen Teilnehmer angegeben.

Tabelle B.2: Ergebnisse der NN3-Competition

Rang*	Verfahren	Mean-RO-sMAPE	Referenz	Beschreibung	Verfahrensklassen
1/-	B09 Wildi	14,84	Wildi (2007)	Kombination von 18 Kalman-Filtern basierend auf Zustandsraummodellen mit einer exponentiell abnehmenden Wichtigkeit vergangener Beobachtungen	Klassische Dekomposition Zustandsraummodelle
2/-	B07 Theta	14,89	Assimakopoulos und Nikolopoulos (2000)	Dekomposition in Theta-Linien und anschließende Kombination einer einfachen exponentiellen Glättung und einer linearen Regression	Exponentielle Glättung Lineare Regression
3/1	C27 Illies	15,18	Illies et al. (2007)	Ensemble aus 500 rekurrenten Echo-State Netzen	Neuronale Netze
4/-	B03 ForecastPro	15,44	Goodrich (2000)	Individuelle Selektion einer Verfahrensklasse und automatische Parameteroptimierung durch das Forecast Pro System von Business Forecast Systems	Exponentielle Glättung ARIMA Gleitende Mittelwerte Croston's Methode Poissonverteilung Negative Binomialverteilung
5/-	B16 DES	15,90	Gardner und McKenzie (1985)	Exponentielle Glättung mit gedämpftem Trend	Exponentielle Glättung
6/-	B17 Comb-S-H-D	15,93	Makridakis und Hibon (2000)	Kombination der Verfahren SES, HES und DES	Exponentielle Glättung
7/-	B05 Autobox	15,95	Reilly (2000)	ARIMA-Modellierung nach dem Box-Jenkins-Verfahren mit nachträglicher Ausreißer-Anpassung implementiert in das Autobox-System von Automatic Forecasting Systems	ARIMA
8/2	C38 Adeodato	16,17	Adeodato et al. (2011)	Ensemble aus 15 Multilayer-Perzeptronen mit vorheriger Trendbereinigung und Datennormalisierung	Neuronale Netze
9/3	C03 Flores	16,31	Flores et al. (2007)	Automatische lineare Modellierung durch selbst-adaptierende genetische Algorithmen	ARIMA
10/-	B14 SES	16,42	Makridakis und Hibon (2000)	Einfache exponentielle Glättung	Exponentielle Glättung
11/-	B15 HES	16,49	Makridakis und Hibon (2000)	Holts lineare exponentielle Glättung	Exponentielle Glättung
12/4	C46 Chen	16,55	Chen und Yao (2007)	Ensemble aus 500 Regressionsbäumen	Klassifikations- und Regressionsbäume
13/5	C13 D'yakonov	16,57	D'yakonov (2007)	Lokal konstante k -nächste-Nachbarn-Modelle mit manueller Parameterselektion	k -nächste-Nachbarn

Tabelle B.2: Ergebnisse der NN3-Competition (Fortsetzung)

Rang*	Verfahren	Mean-RO-sMAPE	Referenz	Beschreibung	Verfahrensklassen
14/6	B00 AutomatANN	16,81	Balkin und Ord (2000)	Individuelle Selektion zwischen drei verschiedenen Modellen basierend auf Bayes Informationskriterium: automatisch trainiertes Multilayer-Perzeptron mit einer verdeckten Schicht und Parameteroptimierung durch Backpropagation, lineare Autoregression, Random Walk	Neuronale Netze Lineare Autoregression Random Walk
15/7	C50 Kamel	16,92	Ahmed et al. (2007)	Kombination eines Multilayer-Perzeptrons und einer Gauß-Prozess-Regression	Neuronale Netze Gauß-Prozess-Regression
16/-	B13 Njimi	17,05	Njimi und Mélard (2007)	Expertensystem zur Vorverarbeitung einer Zeitreihe durch verschiedene Transformationen und anschließende Selektion eines ARIMA-Modells durch Maximum Likelihood Schätzung	ARIMA
17/8	C24 Abou Nasr	17,54	Abou-Nasr (2007)	Ensemble aus 10 rekurrenten neuronalen Netzen	Neuronale Netze
18/9	C31 Theodosiou	17,62	Theodosiou und Murali (2007), Theodosiou (2011)	Kombination von generalisierten Regressionsnetzen, Theta-Methode und saisonalem Random Walk nach einer STL-Dekomposition	STL-Dekomposition Random Walk Neuronale Netze Lineare Regression Exponentielle Glättung
19/-	B06 Census-X12	17,78	Crone et al. (2011)	Automatische ARIMA-Modellierung durch Census X-12-ARIMA Software	ARIMA
20/-	B02 nMLP	17,84	Crone et al. (2011)	Naives Multilayer-Perzeptron	Neuronale Netze
21/10	C26 de Vos	18,24	de Vos (2007)	Kombination von drei Modellen nach einer Wavelet-Dekomposition: Multilayer-Perzeptron, radiale Basifunktionsnetze, Support Vector Regression	Neuronale Netze Support Vector Regression
22/-	B01 nSVR	18,32	Crone und Pietsch (2007)	Naive Support Vector Regression	Support Vector Regression
23/11	C44 Yan	18,58	Yan (2007)	Ensemble aus 3 generalisierten Regressionsnetzen	Neuronale Netze
24/12	C11 Perfilieva	18,62	Perfilieva et al. (2007)	Kombination von zwei Verfahren: Fuzzy-Transformation und perzeptionsbasierte logische Deduktion	Fuzzy-Transformation Perzeptionsbasierte logische Deduktion
25/13	C37 Duclos	18,68	Crone et al. (2011)	Rekurrentes neuronales Netzwerk welches durch die Maximum-a-posteriori-Methode, dem Levenberg-Marquardt-Algorithmus und genetische Algorithmen trainiert wird	Neuronale Netze
26/14	C49 Schliebs	18,72	Crone et al. (2011)	Neuronale Netze mit einer Quantum-Merkmalselektion	Neuronale Netze
27/15	C59 Beliakov	18,73	Beliakov (2006)	Optimale Lipschitz-Interpolation	Interpolation
28/16	C20 Kurogi	18,97	Kurogi et al. (2007)	Kombination mehrerer kompetitiver assoziativer neuronaler Netze	Neuronale Netze
29/-	B10 Beadle	19,14	Beadle und Johnson (2007)	Saisonschematamethode, welche für jeden Kalendermonat eines von acht verschiedenen Prognosemodellen selektiert (Random Walk, Holts und Holt-Winters exponentielle Glättung, additive und multiplikative Dekomposition, logarithmische multiple lineare Regression, multiple lineare Regression und segmentierte kleinste Quadrate)	Random Walk Exponentielle Glättung Dekomposition Lineare Regression Segmentierte kleinste Quadrate
30/-	B11 Lewicke	19,17	Lewicke (2007)	Prognosegleichungen bestehend aus einer Trendlinie und einer Serie von sinusförmigen Fehlertermen berechnet durch Paracaster J von Parabolic Systems	Lineare Regression Sinusfunktionen
31/17	C36 Sorjamaa	19,51	Sorjamaa und Lendasse (2007)	Kombination von Self-Organizing Map und empirischer orthogonaler Funktionsmodelle	Self-Organizing Map Empirische orthogonale Funktionsmodelle
32/18	C15 Isa	20,00	Iša (2007)	Wachsendes Fuzzy-Inferenz neuronales Netz	Neuronale Netze
33/19	C28 Eruhimov	20,19	Crone et al. (2011)	Wavelet-Dekomposition und Gradient-Boosting-Bäume	Gradient-Boosting

Tabelle B.2: Ergebnisse der NN3-Competition (Fortsetzung)

Rang*	Verfahren	Mean-RO-sMAPE	Referenz	Beschreibung	Verfahrensklassen
34/20	C51 Papadaki	22,60	Papadaki und Amaxopoulos (2007)	Dynamische Architektur für neuronale Netze, bei der im Trainingsprozess sukzessive weitere verdeckte Schichten hinzugefügt werden, bis ein Abbruchkriterium erfüllt ist	Neuronale Netze
35/-	B04 Naive	22,69	Crone et al. (2011)	Nicht-saisonaler Random Walk	Random Walk
36/-	B12 Hazarika	23,72	Hazarika (2007)	Softmax-Kombination von 100 Zufallszahlensequenzen nach einer klassischen Dekomposition	Dekomposition Zufallszahlensequenzen
37/21	C17 Chang	24,09	Crone et al. (2011)	k -nächste-Nachbarn-Regression und Support Vector Regression	k -nächste-Nachbarn Support Vector Regression
38/22	C30 Pucheta	25,13	Pucheta et al. (2007)	Time-Delay Netze, die in einem dynamischen Lernprozess basierend auf Lang- und Kurzzeitabhängigkeit trainiert werden	Neuronale Netze
39/23	C57 Corzo	32,66	Corzo and Hong 2007	Ensemble eines Multilayer-Perzeptrons und eines Regressionsbaumes	Neuronale Netze Regressionsbäume

*Gesamtrang aller Verfahren inklusive Benchmarks/Rang der Teilnehmer an der NN3-Competition

Anhang C

Vergleich verschiedener Selektionsmethoden für lokale k -nächste-Nachbarn-Modelle

C.1 Vergleich verschiedener Regularisierungsmethoden

Im Folgenden werden verschiedene Parametermengen der lokal konstanten (LC) und lokal linearen (LL) k -nächste-Nachbarn-Prognosemodelle hinsichtlich ihrer Prognosegüten und Rechenaufwände verglichen. Tabelle C.1 zeigt die verglichenen Parameterkonfigurationen. Die lokal lineare k nn-Regression nach der Methode der kleinsten Quadrate (OLS) besitzt keinen Regularisierungsparameter. Der Regularisierungsparameter der lokal linearen k nn-Hauptkomponentenregression (LL PCR) und der lokal linearen partiellen k nn-Regression der kleinsten Quadrate (LL PLSR) spezifiziert den minimalen prozentualen Wert der kumulativen Varianz aller Hauptkomponenten oder aller Komponenten der partiellen kleinsten Quadrate, die von den letztlich genutzten Komponenten abgedeckt wird. Der Regularisierungsparameter der lokal linearen k nn-Ridge-Regression (LL RR) und der lokal linearen k nn-LASSO-Regression (LL LASSO) ist ein Schrumpfungparameter, welcher große Modellkoeffizienten bestraft. Für weitere Informationen hinsichtlich der Regularisierungsmethoden sei auf (Hastie et al., 2009; Kugiumtzis et al., 1998; Kück und Freitag, 2021) verwiesen. Abbildung C.1 visualisiert die mittleren Prognosefehler (a) und die mittleren Rechenaufwände (b) der verschiedenen Parameterkonfigurationen der lokalen k nn-Prognosemodelle. Tabelle C.2 fasst beide Evaluationen zusammen.

Tabelle C.1: Vergleichene Parameterkonfigurationen hinsichtlich der Mittelwertbildungen der lokal konstanten (LC) sowie der Regularisierungsparameter der lokal linearen (LL) k -nächste-Nachbarn-Prognosemodelle

Modelltyp	Parameterkonfigurationen											
LC	Mean	Median										
LL OLS	–											
LL PCR	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,75	0,8	0,85	0,9	0,95
LL PLSR	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,75	0,8	0,85	0,9	0,95
LL RR	0,01	0,05	0,1	0,5	1	5	10	50	100	500	1000	5000
LL LASSO	0,01	0,05	0,1	0,5	1	5	10	50	100	500	1000	5000

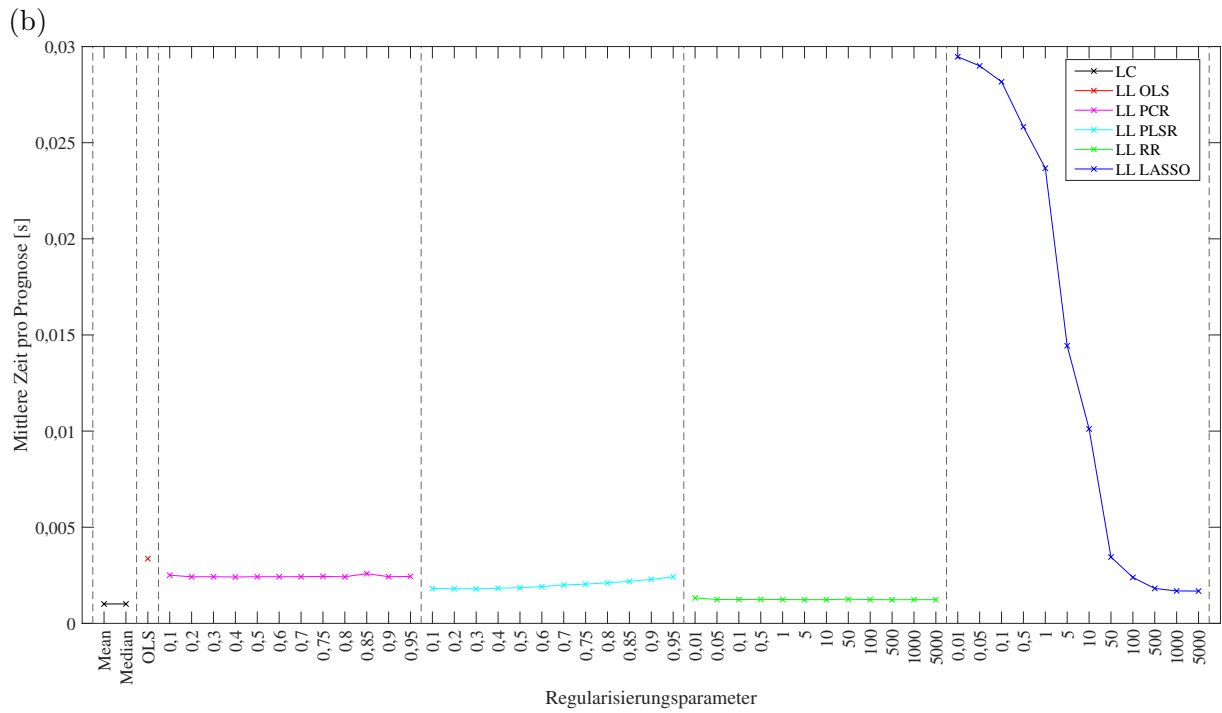
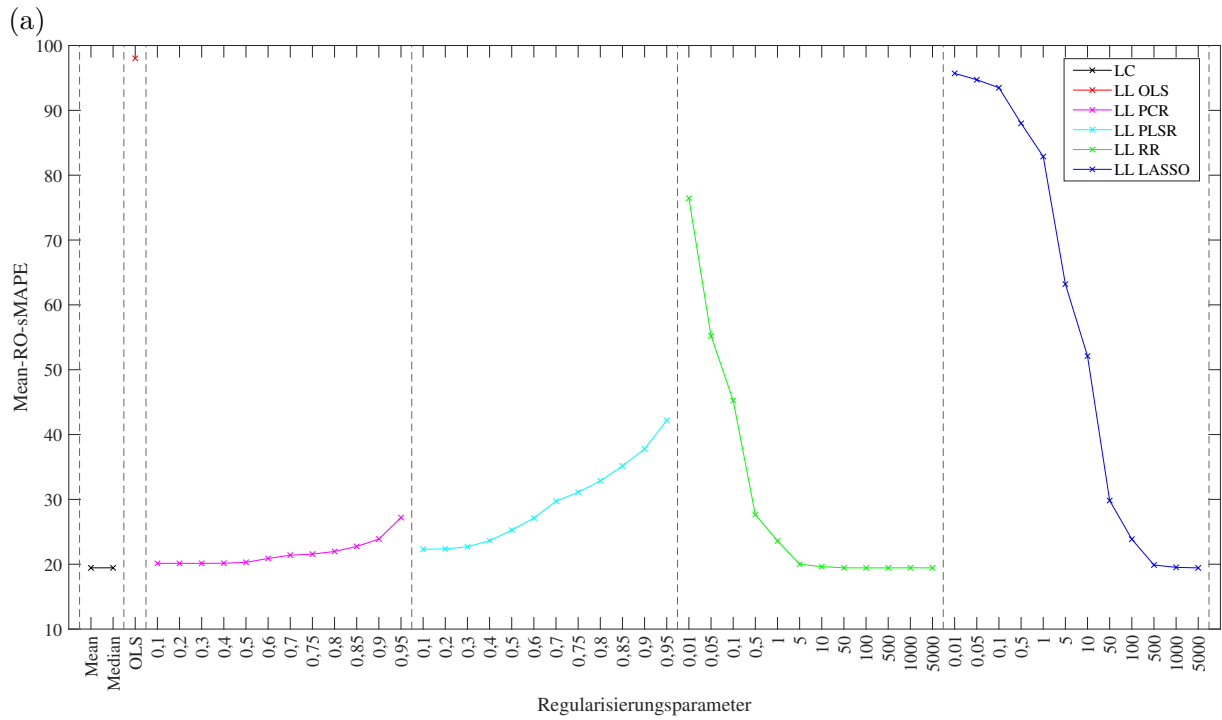


Abbildung C.1: Vergleich verschiedener Parameterkonfigurationen der lokal konstanten (LC) und der lokal linearen (LL) k -nächste-Nachbarn-Modelle hinsichtlich des (a) $\text{Mean-RO-sMAPE}_{T,T+6}^{1,12}$ und (b) der mittleren Rechenaufwände für die 808 industriellen Zeitreihen der M3-Wettbewerb

Tabelle C.2: Mittlere Fehler und mittlere Rechenaufwände pro Prognose verschiedener Parameterkonfigurationen der lokalen k -nächste-Nachbarn-Modelle für unterschiedliche Zeitreihenlängen

Mean-RO-sMAPE						Modell						Mittl. Zeit pro Prognose	
alle		kurz		lang		LC	LL				Wert [s]	Rang	
Rang	Wert	Rang	Wert	Rang	Wert		OLS	PCR	PLSR	RR			LASSO
1	19,44	4	26,19	2	15,89	–	–	–	–	5000	0,0017	15	
2	19,44	6	26,19	3	15,89	–	–	–	500	–	0,0012	3	
3	19,44	7	26,21	1	15,89	–	–	–	100	–	0,0012	8	
4	19,44	3	26,18	5	15,91	–	–	–	5000	–	0,0012	4	
5	19,44	4	26,19	4	15,90	Mean	–	–	–	–	0,0010	1	
6	19,45	1	25,99	9	16,02	Median	–	–	–	–	0,0010	2	
7	19,45	8	26,21	6	15,91	–	–	–	1000	–	0,0012	6	
8	19,45	2	26,17	7	15,93	–	–	–	50	–	0,0013	13	
9	19,52	9	26,28	8	15,97	–	–	–	–	1000	0,0017	16	
10	19,62	10	26,37	10	16,09	–	–	–	10	–	0,0012	5	
11	19,88	16	26,96	11	16,17	–	–	–	–	500	0,0018	20	
12	19,99	14	26,85	12	16,40	–	–	–	5	–	0,0012	7	
13	20,14	13	26,84	13	16,62	–	–	0,3	–	–	0,0024	36	
14	20,14	11	26,84	15	16,62	–	–	0,1	–	–	0,0025	41	
14	20,14	11	26,84	15	16,62	–	–	0,2	–	–	0,0024	33	
16	20,15	15	26,89	14	16,62	–	–	0,4	–	–	0,0024	30	
17	20,29	17	27,07	17	16,73	–	–	0,5	–	–	0,0024	35	
18	20,89	18	27,65	18	17,35	–	–	0,6	–	–	0,0024	34	
19	21,39	19	28,10	19	17,88	–	–	0,7	–	–	0,0024	38	
20	21,56	20	28,34	20	18,01	–	–	0,75	–	–	0,0024	40	
21	21,97	21	28,84	21	18,37	–	–	0,8	–	–	0,0024	32	
22	22,31	22	29,62	22	18,47	–	–	–	0,1	–	0,0018	19	
23	22,37	23	29,70	23	18,52	–	–	–	0,2	–	0,0018	18	
24	22,70	25	30,01	25	18,86	–	–	–	0,3	–	0,0018	17	
25	22,74	24	29,78	26	19,05	–	–	0,85	–	–	0,0026	42	
26	23,61	26	31,01	28	19,73	–	–	–	–	1	0,0012	9	
27	23,64	27	31,14	27	19,70	–	–	–	0,4	–	0,0018	21	
28	23,86	30	33,41	24	18,86	–	–	–	–	100	0,0024	29	
29	23,88	28	31,28	29	20,00	–	–	0,9	–	–	0,0024	37	
30	25,27	29	32,83	30	21,30	–	–	–	0,5	–	0,0019	22	
31	27,12	32	35,35	31	22,81	–	–	–	0,6	–	0,0019	23	
32	27,19	31	34,99	32	23,10	–	–	0,95	–	–	0,0024	39	
33	27,65	33	35,79	34	23,38	–	–	–	–	0,5	0,0012	12	
34	29,71	34	38,96	35	24,86	–	–	–	0,7	–	0,0020	24	
35	29,82	37	42,54	33	23,15	–	–	–	–	50	0,0034	44	
36	31,10	35	40,54	36	26,14	–	–	–	0,75	–	0,0020	25	
37	32,86	36	42,52	37	27,79	–	–	–	0,8	–	0,0021	26	
38	35,16	38	44,78	38	30,12	–	–	–	0,85	–	0,0022	27	
39	37,78	39	47,85	39	32,50	–	–	–	0,9	–	0,0023	28	
40	42,20	40	52,83	40	36,62	–	–	–	0,95	–	0,0024	31	
41	45,25	41	57,17	41	38,99	–	–	–	–	0,1	0,0012	10	
42	52,10	43	72,01	42	41,65	–	–	–	–	10	0,0101	45	
43	55,22	42	68,52	43	48,24	–	–	–	–	0,05	0,0012	11	
44	63,20	44	84,23	44	52,17	–	–	–	–	5	0,0144	46	
45	76,44	45	90,51	45	69,07	–	–	–	–	0,01	0,0013	14	
46	82,88	46	100,0	46	73,90	–	–	–	–	1	0,0237	47	
47	87,99	47	102,6	47	80,32	–	–	–	–	0,5	0,0258	48	
48	93,49	48	105,8	48	87,06	–	–	–	–	0,1	0,0282	49	
49	94,72	49	106,3	49	88,67	–	–	–	–	0,05	0,0290	50	
50	95,70	50	106,9	50	89,83	–	–	–	–	0,01	0,0295	51	
51	98,02	51	108,1	51	92,73	–	OLS	–	–	–	0,0034	43	

C.2 Vergleich statischer und dynamischer Modellselektionsmethoden

Im Folgenden werden verschiedene statische und dynamische Modellselektionsstrategien für lokal konstante k nn-Modelle und lokal lineare k nn-Ridge-Regressionsmodelle verglichen. In diesem Zusammenhang werden die Strategien, die die Modellparameter durch Benchmark-Methoden selektieren, mit b bezeichnet. Hierbei wird die Zeitverschiebung d als erste Minimalstelle der mittleren gemeinsamen Information einer Zeitreihe ermittelt (vgl. Fraser und Swinney, 1986), die Einbettungsdimension r durch den Algorithmus der falschen nächsten Nachbarn (vgl. Kennel et al., 1992) und die Anzahl der nächsten Nachbarn als $k = r + 1$. Tabelle C.3 zeigt die verglichenen Parameterkonfigurationen für die Modellselektionsstrategien. Die Abbildungen C.2-C.4 visualisieren die mittleren Prognosefehler der fünf besten Konfigurationen jeder Parametermenge für die Mengen aller, kurzer und langer Zeitreihen der in der vorliegenden Arbeit betrachteten Meta-Trainingsmenge. Die Tabellen C.4-C.6 zeigen die mittleren Fehler und die Ränge der jeweils fünf besten Konfigurationen.

Tabelle C.3: Vergleichene statische und dynamische Modellselektionsmethoden für lokal konstante (LC) k -nächste-Nachbarn-Modelle und lokal lineare k -nächste-Nachbarn-Ridge-Regressionsmodelle (LL RR)

Modelltyp	Wertebereiche										
	LC			LL RR (λ)							
	Mean	Median		0,5	1	5	10	50	100	500	
d	1	3	6	12	b						
$r\%$	0	0,25	0,5	0,75	1	b					
$k\%$	0	0,25	0,5	0,75	1	b					
k -Typ	dynamisch		statisch								

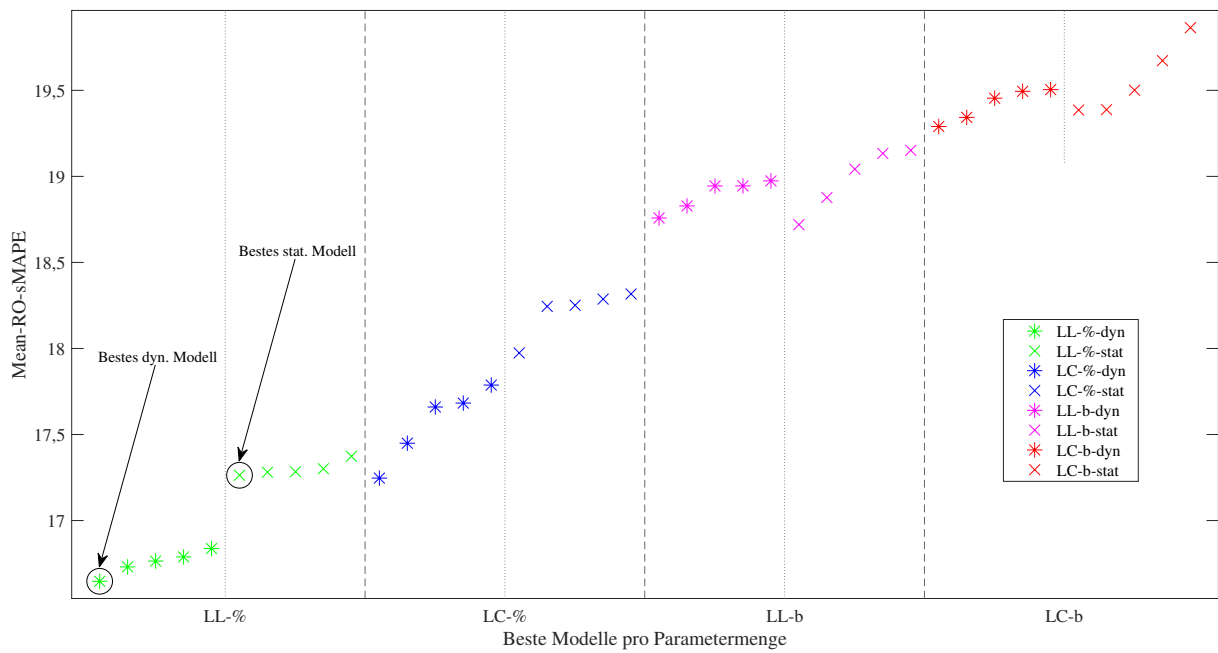


Abbildung C.2: Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 697 langen und kurzen Zeitreihen der Meta-Trainingsmenge

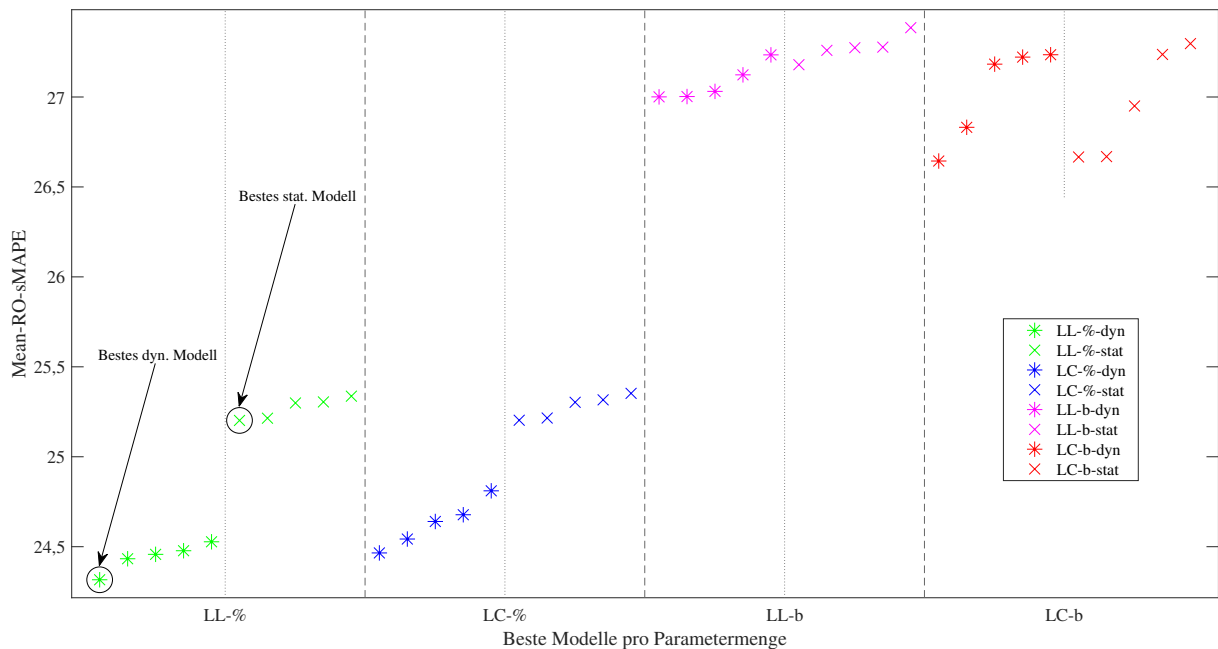


Abbildung C.3: Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 227 kurzen Zeitreihen der Meta-Trainingsmenge

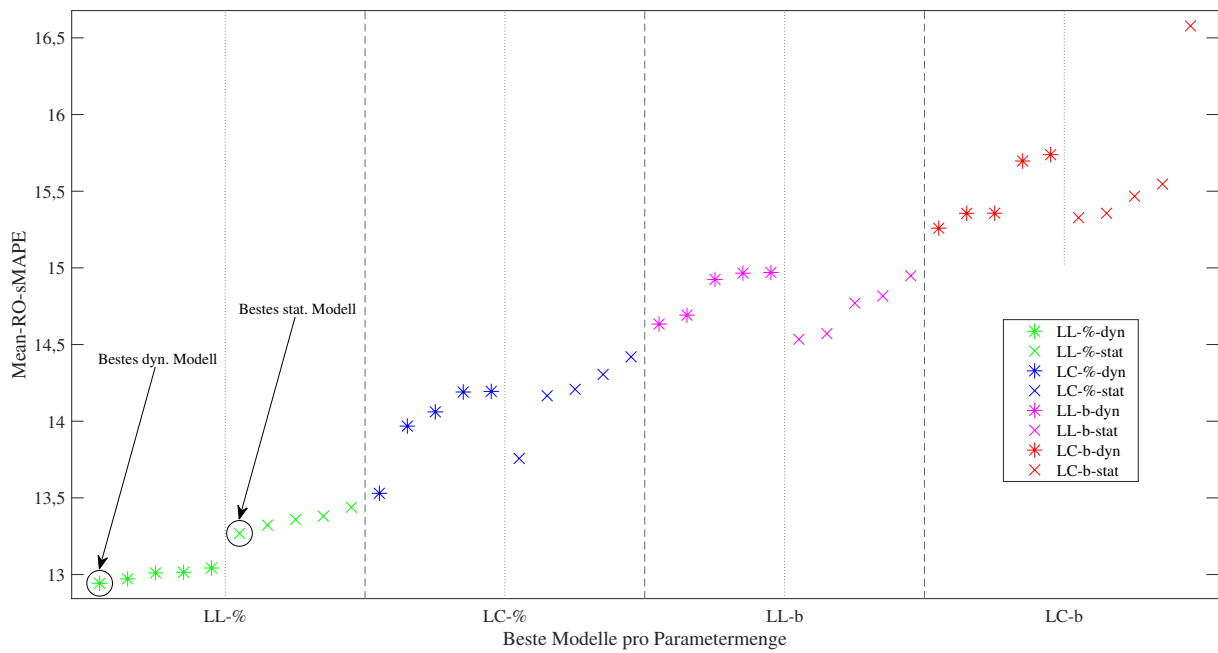


Abbildung C.4: Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 470 langen Zeitreihen der Meta-Trainingsmenge

Tabelle C.4: Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 697 langen und kurzen Zeitreihen der Meta-Trainingsmenge

Mean-RO-sMAPE		Modelltyp		Selektionsmethode			
Rang	Wert	LC	LL RR	d	$r\%$	$k\%$	k -Typ
1	16,65		100	1	1	1	dynamisch
2	16,73		100	1	1	0,75	dynamisch
3	16,76		100	1	1	0,5	dynamisch
4	16,79		50	1	1	0,5	dynamisch
5	16,84		50	1	1	1	dynamisch
14	17,25	Mean		1	1	0,25	dynamisch
17	17,26		50	1	0,75	0,75	statisch
19	17,28		100	1	0,75	0,75	statisch
21	17,28		100	1	0,75	1	statisch
22	17,30		50	1	0,75	1	statisch
24	17,37		50	1	0,75	0,5	statisch
29	17,45	Mean		1	1	0,5	dynamisch
46	17,66	Mean		3	1	0,25	dynamisch
48	17,68	Median		1	1	0,25	dynamisch
58	17,79	Mean		1	0,75	0,25	dynamisch
72	17,97	Mean		1	0,75	0,5	statisch
120	18,24	Mean		1	0,75	0,75	statisch
122	18,25	Mean		6	1	1	statisch
126	18,29	Mean		1	0,75	0,25	statisch
133	18,32	Mean		1	0,5	0,25	statisch
227	18,72		10	b	b	1	statisch
241	18,76		10	b	b	0,75	dynamisch
261	18,83		5	b	b	0,75	dynamisch
273	18,88		5	b	b	1	statisch
290	18,94		1	b	b	1	dynamisch
291	18,94		0.5	b	b	1	dynamisch
296	18,97		5	b	b	1	dynamisch
309	19,04		10	b	b	0,75	statisch
339	19,13		10	b	b	0,5	statisch
347	19,15		50	b	b	0,25	statisch
396	19,29	Median		b	b	0,25	dynamisch
417	19,34		500	b	b	0,25	statisch
436	19,39	Mean		b	b	0,25	statisch
437	19,39	Median		b	b	0,25	statisch
462	19,45	Mean		b	b	0,25	dynamisch
475	19,49		500	b	b	b	statisch
477	19,50	Mean		b	b	b	statisch
478	19,50		100	b	b	b	statisch
530	19,67	Median		b	b	b	statisch
572	19,86	Median		b	b	0,5	statisch

Tabelle C.5: Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 227 kurzen Zeitreihen der Meta-Trainingsmenge

Mean-RO-sMAPE		Modelltyp		Selektionsmethode			
Rang	Wert	LC	LL RR	d	$r\%$	$k\%$	k -Typ
1	24,32		100	1	1	1	dynamisch
2	24,43		100	1	1	0,75	dynamisch
3	24,46		500	1	1	0,5	dynamisch
4	24,47	Mean		1	1	0,5	dynamisch
5	24,48		500	1	1	0,75	dynamisch
6	24,53		100	1	1	0,5	dynamisch
7	24,54	Median		1	1	0,5	dynamisch
8	24,64	Mean		1	1	0,75	dynamisch
9	24,68	Median		1	1	0,75	dynamisch
14	24,81	Median		3	1	0,5	dynamisch
37	25,20		500	1	0,75	1	statisch
38	25,20	Median		1	0,75	1	statisch
41	25,21		100	1	0,75	1	statisch
42	25,22	Mean		1	0,75	1	statisch
50	25,30		50	1	0,75	1	statisch
52	25,30	Mean		1	0,75	0,75	statisch
55	25,30		500	1	0,75	0,75	statisch
57	25,32	Median		1	0,5	0,75	statisch
59	25,34	Mean	100	1	0,5	0,75	statisch
64	25,35	Mean		1	0,5	0,75	statisch
326	26,64	Median		b	b	0,25	dynamisch
336	26,67	Median		b	b	0,75	statisch
338	26,67	Median		b	b	0,5	statisch
384	26,83	Median		b	b	0,5	dynamisch
418	26,95	Median		b	b	1	statisch
430	27,00		1	b	b	1	dynamisch
431	27,00		0.5	b	b	1	dynamisch
437	27,03		5	b	b	1	dynamisch
458	27,12		10	b	b	1	dynamisch
477	27,18		50	b	b	0,75	statisch
478	27,18		100	b	b	0,75	statisch
489	27,22		500	b	b	0,75	statisch
493	27,23		100	b	b	0,25	dynamisch
495	27,23	Mean		b	b	0,25	dynamisch
496	27,24	Mean		b	b	0,75	statisch
505	27,26		50	b	b	1	statisch
511	27,27		100	b	b	0,5	statisch
512	27,28		50	b	b	0,5	statisch
517	27,30	Mean		b	b	0,5	statisch
538	27,39		10	b	b	1	statisch

Tabelle C.6: Mean RO-sMAPEs der fünf besten Konfigurationen jeder Parametermenge der lokalen k -nächste-Nachbarn-Modelle für die 470 langen Zeitreihen der Meta-Trainingsmenge

Mean-RO-sMAPE		Modelltyp		Selektionsmethode			
Rang	Wert	LC	LL RR	d	$r\%$	$k\%$	k -Typ
1	12,94		100	1	1	1	dynamisch
2	12,97		50	1	1	0,5	dynamisch
3	13,01		100	1	1	0,75	dynamisch
4	13,02		100	1	1	0,5	dynamisch
5	13,04		50	1	1	1	dynamisch
10	13,27		50	1	0,75	0,5	statisch
13	13,32		50	1	0,75	0,75	statisch
16	13,36		100	1	0,75	0,5	statisch
17	13,38		100	1	0,75	0,75	statisch
26	13,44		50	1	0,75	1	statisch
32	13,53	Mean		1	1	0,25	dynamisch
46	13,76	Mean		1	0,75	0,25	statisch
70	13,97	Mean		3	1	0,25	dynamisch
76	14,06	Mean		1	1	0,5	dynamisch
88	14,17	Median		1	0,75	0,25	statisch
90	14,19	Median		1	1	0,25	dynamisch
93	14,19	Mean		1	0,75	0,25	dynamisch
94	14,21	Mean		1	0,75	0,5	statisch
108	14,31	Mean		1	0,5	0,25	statisch
137	14,42	Mean		3	0,75	0,25	statisch
153	14,53		10	b	b	1	statisch
156	14,57		5	b	b	1	statisch
167	14,63		10	b	b	0,75	dynamisch
179	14,69		5	b	b	0,75	dynamisch
191	14,77		1	b	b	1	statisch
203	14,82		0,5	b	b	1	statisch
223	14,92		1	b	b	0,75	dynamisch
226	14,95		10	b	b	0,75	statisch
228	14,97		10	b	b	0,5	dynamisch
230	14,97		0,5	b	b	0,75	dynamisch
333	15,26		500	b	b	0,25	statisch
348	15,33	Mean		b	b	0,25	statisch
359	15,36		500	b	b	b	statisch
360	15,36	Mean		b	b	b	statisch
361	15,36		100	b	b	b	statisch
393	15,47	Median		b	b	0,25	statisch
418	15,55	Median		b	b	b	statisch
469	15,70	Mean		b	b	0,25	dynamisch
480	15,74	Median		b	b	0,25	dynamisch
702	16,58	Median		b	b	0,5	statisch

Anhang D

Selektionen verschiedener Parameterkonfigurationen des Selbinationsverfahrens

Im Folgenden werden die Selektionen verschiedener Parameterkonfigurationen des Selbinationsverfahrens beschreiben. Tabelle D.1 zeigt, wie viele Merkmale für verschiedene Konfigurationen verwendet werden. Hierbei werden sowohl Konfigurationen mit und ohne eine Merkmalsselektion betrachtet. Die Mengen A_M^E enthalten die M Prognoseverfahrenskandidaten mit den geringsten durchschnittlichen Fehlern auf der Meta-Trainingsmenge, während die Mengen A_M^C zunächst jeweils mindestens ein Verfahren aus jeder Klasse enthalten muss, bevor nach dem Fehler auf der Meta-Trainingsmenge entschieden wird (vgl. Abschnitt 6.4). Tabelle D.2 zeigt wie häufig Prognoseverfahrenskandidaten für verschiedene Parameterkonfigurationen des Selbinationsverfahrens selektiert wurden.

Tabelle D.1: Anzahl der Meta-Merkmale in den Mengen ohne und mit Merkmalsselektion für ausgewählte Konfigurationen des Selbinationsverfahrens hinsichtlich der Menge der Prognoseverfahrenskandidaten \mathbf{A}

Anzahl Merkmale F	\mathbf{A}										
	\mathbf{A}_2^E	\mathbf{A}_2^C	\mathbf{A}_3^E	\mathbf{A}_3^C	\mathbf{A}_4^E	\mathbf{A}_4^C	\mathbf{A}_5^E	\mathbf{A}_5^C	\mathbf{A}_6	\mathbf{A}_7	\mathbf{A}_8
ohne Merkmalsselektion	77	77	82	82	89	89	98	98	109	122	137
nach Merkmalsselektion	35	32	8	19	8	25	44	17	22	9	45

Tabelle D.2: Anzahl der Selektionen der verschiedenen Prognoseverfahrenskandidaten für ausgewählte Konfigurationen des Selbbinationsverfahrens hinsichtlich \mathbf{A} und Ω_{\min} (alle Werte betreffen die Konfigurationen mit Merkmalsselektion)

\mathbf{A}	Kandidatenauswahl nach		Ω_{\min}	Selektierte Prognoseverfahren							Mittlere Anzahl selektierter Verfahren	
	Fehlern	Klassen		llknn	ets	nnetar	srw	lcknn	auto.arima	rw		
\mathbf{A}_2^E	✓		0	91	0	0	0	20	0	0	0	1
	✓		0,1	91	0	0	0	20	0	0	0	1
	✓		0,2	91	0	0	0	20	0	0	0	1
	✓		0,3	91	0	0	0	20	0	0	0	1
	✓		0,4	91	0	0	0	20	0	0	0	1
	✓		0,5	91	0	0	0	20	0	0	0	1
	✓		0,6	107	0	0	0	57	0	0	0	1,48
	✓		0,7	110	0	0	0	93	0	0	0	1,83
	✓		0,8	111	0	0	0	111	0	0	0	2
	✓		0,9	111	0	0	0	111	0	0	0	2
✓		1	111	0	0	0	111	0	0	0	2	
\mathbf{A}_2^C		✓	0	69	42	0	0	0	0	0	0	1
		✓	0,1	69	42	0	0	0	0	0	0	1
		✓	0,2	69	42	0	0	0	0	0	0	1
		✓	0,3	69	42	0	0	0	0	0	0	1
		✓	0,4	69	42	0	0	0	0	0	0	1
		✓	0,5	69	42	0	0	0	0	0	0	1
		✓	0,6	88	65	0	0	0	0	0	0	1,38
		✓	0,7	99	96	0	0	0	0	0	0	1,76
		✓	0,8	105	111	0	0	0	0	0	0	1,95
		✓	0,9	110	111	0	0	0	0	0	0	1,99
	✓	1	111	111	0	0	0	0	0	0	2	
\mathbf{A}_4^E	✓		0	65	23	0	0	1	22	0	0	1
	✓		0,1	65	23	0	0	1	22	0	0	1
	✓		0,2	65	23	0	0	1	22	0	0	1
	✓		0,3	72	28	0	0	1	27	0	0	1,15
	✓		0,4	76	54	0	0	39	37	0	0	1,86
	✓		0,5	77	61	0	0	43	38	0	0	1,97
	✓		0,6	83	78	0	0	49	59	0	0	2,42
	✓		0,7	99	111	0	0	50	61	0	0	2,89
	✓		0,8	111	111	0	0	59	64	0	0	3,11
	✓		0,9	111	111	0	0	101	110	0	0	3,90
✓		1	111	111	0	0	111	111	0	0	4	
\mathbf{A}_4^C		✓	0	67	32	5	7	0	0	0	0	1
		✓	0,1	67	32	5	7	0	0	0	0	1
		✓	0,2	67	32	5	7	0	0	0	0	1
		✓	0,3	68	33	6	8	0	0	0	0	1,04
		✓	0,4	77	45	11	16	0	0	0	0	1,34
		✓	0,5	87	65	21	24	0	0	0	0	1,77
		✓	0,6	89	82	23	30	0	0	0	0	2,02
		✓	0,7	95	101	37	43	0	0	0	0	2,49
		✓	0,8	102	106	65	54	0	0	0	0	2,95
		✓	0,9	109	110	95	94	0	0	0	0	3,68
	✓	1	111	111	111	111	0	0	0	0	4	
\mathbf{A}_6	✓	✓	0	43	23	9	3	16	17	0	0	1
	✓	✓	0,1	43	23	9	3	16	17	0	0	1
	✓	✓	0,2	43	24	9	3	17	17	0	0	1,02
	✓	✓	0,3	65	34	15	8	39	30	0	0	1,72
	✓	✓	0,4	73	42	16	8	48	33	0	0	1,98
	✓	✓	0,5	84	52	20	14	58	52	0	0	2,52
	✓	✓	0,6	90	71	25	19	70	60	0	0	3,02
	✓	✓	0,7	94	90	39	25	75	86	0	0	3,68
	✓	✓	0,8	104	101	55	36	80	105	0	0	4,33
	✓	✓	0,9	110	111	98	57	93	108	0	0	5,20
✓	✓	1	111	111	111	111	111	111	0	0	6	
\mathbf{A}_8	✓	✓	0	44	23	10	2	13	18	1	0	1
	✓	✓	0,1	44	23	10	2	13	18	1	0	1
	✓	✓	0,2	47	23	12	2	14	22	1	1	1,10
	✓	✓	0,3	63	35	17	7	42	31	1	2	1,78
	✓	✓	0,4	68	45	21	10	51	36	3	2	2,13
	✓	✓	0,5	75	63	28	15	61	48	4	2	2,67
	✓	✓	0,6	80	89	40	25	67	61	5	4	3,34
	✓	✓	0,7	89	98	56	41	73	81	6	4	4,04
	✓	✓	0,8	98	105	78	57	82	98	13	6	4,84
	✓	✓	0,9	110	109	103	91	96	109	24	7	5,85
✓	✓	1	111	111	111	111	111	111	111	111	8	

Anhang E

Evaluation des Selbinationsverfahrens hinsichtlich des MASE

Tabelle E.1: Durchschnittliche mittlere absolute skalierte Prognosefehler rollierender Prognosen (Mean-RO-MASE) des Selbinationsverfahrens sowie gängiger Benchmark-Verfahren für verschiedene Ursprünge, Horizonte und Zeitreihenlängen anhand der Zeitreihen der NN3-Competition

Experimentkonfiguration				Mean-RO-MASE									
o_{\max}	h_{\min}	h_{\max}	Menge	srw	nnetar	ets	llknn	Val-Sel	ML-Sel	Pool-2	Pool-3	Comb-4	Selbination
T+17	1	1	kurz	0,863	0,965	0,786	0,681	0,771	0,678	0,756	0,734	0,728	0,666
T+17	1	1	lang	1,593	1,368	0,815	0,973	1,218	1,097	1,043	1,033	1,019	0,949
T+17	1	1	alle	1,257	1,183	0,801	0,839	1,013	0,904	0,912	0,895	0,885	0,819
T+15	1	3	kurz	0,857	0,882	0,785	0,691	0,760	0,686	0,741	0,727	0,712	0,668
T+15	1	3	lang	1,578	1,523	0,954	1,089	1,300	1,217	1,132	1,117	1,106	1,050
T+15	1	3	alle	1,246	1,228	0,877	0,906	1,052	0,973	0,953	0,938	0,925	0,875
T+12	4	6	kurz	0,851	0,863	0,801	0,711	0,760	0,703	0,743	0,735	0,713	0,684
T+12	4	6	lang	1,611	1,801	1,299	1,352	1,533	1,496	1,370	1,344	1,328	1,306
T+12	4	6	alle	1,262	1,370	1,070	1,058	1,177	1,132	1,082	1,064	1,045	1,020
T+9	7	9	kurz	0,854	0,871	0,839	0,752	0,798	0,740	0,776	0,761	0,739	0,722
T+9	7	9	lang	1,604	1,874	1,396	1,461	1,631	1,592	1,450	1,423	1,383	1,388
T+9	7	9	alle	1,259	1,414	1,140	1,135	1,248	1,200	1,140	1,119	1,087	1,082
T+6	10	12	kurz	0,862	0,880	0,855	0,785	0,818	0,780	0,799	0,781	0,756	0,762
T+6	10	12	lang	1,592	1,919	1,572	1,562	1,789	1,727	1,554	1,513	1,485	1,569
T+6	10	12	alle	1,257	1,441	1,243	1,205	1,343	1,292	1,207	1,176	1,150	1,198
T+3	13	15	kurz	1,073	0,997	0,881	0,845	0,916	0,855	0,889	0,884	0,864	0,827
T+3	13	15	lang	1,948	2,102	1,858	1,840	1,988	1,910	1,871	1,754	1,745	1,823
T+3	13	15	alle	1,546	1,594	1,409	1,383	1,495	1,425	1,420	1,354	1,340	1,365
T	16	18	kurz	1,113	1,043	0,974	0,868	0,975	0,887	0,909	0,913	0,898	0,877
T	16	18	lang	2,272	2,343	2,296	2,223	2,398	2,273	2,265	2,116	2,107	2,186
T	16	18	alle	1,739	1,746	1,689	1,600	1,744	1,636	1,642	1,563	1,552	1,584
T+12	1	6	kurz	0,848	0,867	0,792	0,698	0,750	0,690	0,736	0,727	0,709	0,671
T+12	1	6	lang	1,577	1,642	1,116	1,219	1,395	1,353	1,245	1,220	1,210	1,172
T+12	1	6	alle	1,242	1,286	0,967	0,980	1,099	1,049	1,011	0,993	0,980	0,942
T+6	7	12	kurz	0,846	0,866	0,841	0,766	0,793	0,755	0,777	0,763	0,740	0,735
T+6	7	12	lang	1,572	1,890	1,512	1,507	1,714	1,674	1,525	1,477	1,436	1,498
T+6	7	12	alle	1,239	1,419	1,204	1,167	1,291	1,252	1,181	1,149	1,117	1,148
T+6	1	12	kurz	0,848	0,863	0,806	0,720	0,759	0,712	0,750	0,738	0,721	0,691
T+6	1	12	lang	1,579	1,747	1,338	1,374	1,556	1,528	1,414	1,366	1,337	1,359
T+6	1	12	alle	1,243	1,341	1,094	1,073	1,190	1,153	1,109	1,078	1,054	1,052
T	1	18	kurz	0,925	0,913	0,852	0,766	0,807	0,767	0,793	0,783	0,776	0,744
T	1	18	lang	1,692	1,843	1,531	1,514	1,659	1,623	1,555	1,470	1,465	1,460
T	1	18	alle	1,339	1,416	1,219	1,170	1,268	1,230	1,205	1,154	1,149	1,131
alle Experimente			kurz	0,904	0,910	0,838	0,753	0,810	0,750	0,788	0,777	0,760	0,732
alle Experimente			lang	1,692	1,823	1,426	1,465	1,653	1,590	1,493	1,439	1,420	1,433
alle Experimente			alle	1,330	1,403	1,156	1,138	1,265	1,204	1,169	1,135	1,117	1,111