



---

# Deep Learning für Zeitreihenvorhersage

Multi-step ahead Prognose von intralogistischen Auftragsdaten

---

## Autor

Lukas Dech

[lukas.dech@nordakademie.de](mailto:lukas.dech@nordakademie.de)

W18b, 9376

06.12.2021 – 07.02.2022

Bachelorarbeit zur Erlangung des akademischen Grades eines Bachelor of Science  
(Wirtschaftsingenieurwesen)

---

### Erstgutachter

M. Sc. Dirk Johannßen

FH Nordakademie

---

### Zweitgutachter

Prof. Dr. Michael Schulz

FH Nordakademie

---

### Betreuer

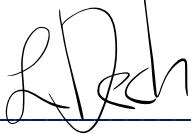
Marcel Andruchiewicz

BUDNI Handels- und Service  
GmbH Co KG

## **Eidesstattliche Erklärung des Studenten / der Studentin**

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder von mir noch von jemand anderem als Prüfungsleistung vorgelegt.

Datum: 06.02.2022

Unterschrift: 

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis .....</b>	<b>III</b>
<b>Abbildungsverzeichnis.....</b>	<b>IV</b>
<b>Tabellenverzeichnis.....</b>	<b>V</b>
<b>Formelverzeichnis .....</b>	<b>VI</b>
<b>1. Einleitung.....</b>	<b>7</b>
<i>1.1 Aufgabenstellung.....</i>	<i>8</i>
<i>1.2 Vorgehensweise und Zielsetzung .....</i>	<i>9</i>
<i>1.3 Verwandte Arbeiten.....</i>	<i>10</i>
<b>2. Zeitreihen.....</b>	<b>11</b>
<i>2.1 Zeitreihenkomponenten.....</i>	<i>11</i>
<i>2.2 Zeitreihenanalyse.....</i>	<i>12</i>
<b>3. Machine-Learning.....</b>	<b>13</b>
<i>3.1 Grundlagen .....</i>	<i>13</i>
<i>3.2 Lernverfahren.....</i>	<i>14</i>
<i>3.3 Instanzbasiertes und modellbasiertes Lernen .....</i>	<i>14</i>
<i>3.4 Overfitting und Underfitting .....</i>	<i>14</i>
<i>3.5 Testen und Validieren .....</i>	<i>15</i>
<b>4. Deep Learning .....</b>	<b>16</b>
<i>4.1 Grundlagen .....</i>	<i>16</i>
<i>4.1.1 Künstliche Neuronen.....</i>	<i>16</i>
<i>4.1.2 Künstliche Neuronale Netze .....</i>	<i>18</i>
<i>4.2 Trainingsprozess .....</i>	<i>19</i>
<i>4.2.1 Forward Propagation.....</i>	<i>20</i>
<i>4.2.2 Verlustfunktion .....</i>	<i>20</i>
<i>4.2.3 Backpropagation .....</i>	<i>21</i>
<i>4.2.4 Gradientenabstieg.....</i>	<i>22</i>
<i>4.2.5 Aktivierungsfunktion .....</i>	<i>23</i>
<i>4.2.6 Naive Prognose .....</i>	<i>24</i>
<i>4.3 Architekturen.....</i>	<i>25</i>
<i>4.3.1 CNN .....</i>	<i>25</i>
<i>4.3.2 RNN .....</i>	<i>26</i>
<i>4.3.3 Encoder-Decoder .....</i>	<i>28</i>
<b>5. Methodik .....</b>	<b>28</b>

<b>6. Artefakt .....</b>	<b>30</b>
<i>6.1 Datenverständnis .....</i>	30
6.1.1 Datenübersicht .....	30
6.1.2 Qualitätsprüfung.....	31
6.1.3 Explorative Datenanalyse .....	32
<i>6.2 Datenvorverarbeitung.....</i>	36
6.2.1 Datenaufbereitung.....	37
6.2.2 Merkmalsextraktion .....	38
<i>6.3 Modellierung.....</i>	40
6.3.1 Vorgehen.....	40
6.3.2 Sequenzielle Modelle: Aufbau.....	43
6.3.3 Sequenzielle Modelle: Ergebnisse .....	44
6.3.4 Multi-Head-Modelle: Aufbau .....	48
6.3.5 Multi-Head-Modelle: Ergebnisse.....	49
<i>6.4 Evaluierung.....</i>	50
<i>6.5 Bereitstellung .....</i>	52
<b>7. Diskussion und Ausblick .....</b>	<b>53</b>
<i>7.1 Diskussion .....</i>	53
<i>7.2 Ausblick.....</i>	54
<i>7.3 Fazit .....</i>	55
<b>Literaturverzeichnis.....</b>	<b>VII</b>
<b>Anhang .....</b>	<b>XII</b>

## Abkürzungsverzeichnis

ANN .....	<i>Artificial neural network</i>
CNN .....	<i>Convolutional neural networks</i>
ConvLSTM .....	<i>Convolutional-LSTM</i>
CRISP-DM.....	<i>Cross Industry Standard Process for Data Mining</i>
DASC-PM.....	<i>Data Science Process Model</i>
MAE.....	<i>Mean absolute error</i>
MAPE.....	<i>Mean absolute percentage error</i>
MLP .....	<i>Multilayer perceptron</i>
MSE .....	<i>Mean squared error</i>
RMSE.....	<i>Root mean square error</i>
RNN .....	<i>Recurrent neural network</i>
TE.....	<i>Transporteinheit</i>

## Abbildungsverzeichnis

Abbildung 1: Zeitreihenkomponenten .....	12
Abbildung 2: Ein Einzelnes Neuron .....	17
Abbildung 3: Einfaches Neuronales Netz.....	18
Abbildung 4: Gradientenabstieg mit unterschiedlichen Lernraten .....	23
Abbildung 5 Aktivierungsfunktionen Sigmoid und ReLU .....	24
Abbildung 6: Beispiel Naive Prognose.....	24
Abbildung 7 Convolutional Neural Network.....	26
Abbildung 8: Recurrent Neural Network.....	27
Abbildung 9: CRISP-DM.....	29
Abbildung 10 Qualitätskriterien Datensätze .....	31
Abbildung 11 Auftragsdaten Liniendiagramm 2020 .....	32
Abbildung 12 Auftragsdaten Heatmap, Monate und Jahre .....	33
Abbildung 13 Auftragsdaten Heatmap, Monate und Wochentage .....	33
Abbildung 14 Auftragsdaten Liniendiagramm, 2020 und 2019 .....	34
Abbildung 15 Auftragsdaten Histogramm.....	35
Abbildung 16 Suchbaum Modellierung.....	41
Abbildung 17 Ebene im Suchbaum .....	42
Abbildung 18 Knoten im Suchbaum.....	42
Abbildung 19 Kennzahlen Naive Prognose .....	45
Abbildung 20 Liniendiagramm Sequenzielle Modelle; Historische Auftragsdaten .....	46
Abbildung 21 Liniendiagramm Sequenzielle Modelle; Weitere Datensätze.....	47
Abbildung 22 Beispiel Multi-Head-Modell.....	48
Abbildung 23 Liniendiagramm Multi-Head; LSTM-CNN Historische Auftragsdaten.....	49
Abbildung 24 Evaluierung Label.....	51

## Tabellenverzeichnis

Tabelle 1 Deskriptive Kennzahlen, Auftragsdaten .....	35
Tabelle 2 Eingabesequenzen, Historische Auftragsdaten .....	39
Tabelle 3 Eingabesequenzen, weitere Datensätze.....	40
Tabelle 4 Ergebnisse Sequenzielle Modelle; Historische Auftragsdaten .....	46
Tabelle 5 Ergebnisse Sequenzielle Modelle; Weitere Datensätze .....	47
Tabelle 6 Ergebnisse Multi-Head Modelle; Historische Auftragsdaten .....	49
Tabelle 7 Ergebnisse Multi-Head Modelle; Alle Datensätze.....	50
Tabelle 8 Prognosefehler bei unterschiedlicher Schrittweite.....	51
Tabelle 9 Vergleich Prognosen für Dezember 2021 .....	52

## Formelverzeichnis

Formel 1: Augabe Neuron.....	17
Formel 2: Gewichtete Summe der Inputs .....	17
Formel 3: Ausgabewerte, vollständig verbundene Schicht.....	19
Formel 4: MSE; Mean squared error .....	20
Formel 5: MAE; Mean absolute error.....	20
Formel 6: Huber-Funktion .....	21
Formel 7: Gradientenabstieg, Aktualisierungsregel.....	22

## 1. Einleitung

In den vergangenen Jahren erfreut sich der Data-Science-Bereich einer immer größer werdenden Beliebtheit. Dies ist vorwiegend auf den Umstand zurückzuführen, dass die Menge an jährlich gesammelten Daten stetig steigt und in immer vielfältigeren Fachgebieten Datenauswertungen als sinnvoll betrachtet werden (Reinsel, Gantz und Rydning 2018). Zusätzlich verfügen immer mehr mittelständische Unternehmen über eine ausgebauten IT-Infrastruktur, wodurch das Sammeln und Auswerten von Daten auch mit internen Ressourcen bewältigt werden kann (Zimmermann, S. 15). Daten stellen das Abbild von in der Vergangenheit liegenden Ereignissen dar. Das Treffen von Entscheidung anhand von Daten, welches im Zuge der Revolution zur Industrie 4.0 auch immer wieder als „data-driven decision making“ betitelt wird, nimmt damit einen immer größer werdenden Stellenwert ein (Provost und Fawcett 2013). Mithilfe verschiedener Analysemethoden und der gesammelten Daten können innerbetriebliche Prozesse, deren Gestaltung bisher auf Erfahrungswerten und Beobachtungen beruhte, optimiert werden. Außerdem können mit der Betrachtung der Daten genaue Aussagen über den Istzustand und über zurückliegende Ereignisse getroffen werden. Für das Auswerten aktueller oder vergangener Zustände werden bei quantitativen Daten vermehrt Methoden aus der deskriptiven Statistik eingesetzt.

Die Auswertung von qualitativen Daten wie Bild, Ton und Schrift wird seit geraumer Zeit auch vermehrt zur Beschreibung von Zuständen herangezogen. Viele Methoden der deskriptiven Statistik versagen bei qualitativen Daten sowie bei quantitativen Datensätzen, welche ein großes Spektrum an Datenpunkten umfassen. Durch die Notwendigkeit, große Mengen an unstrukturierten Datensätzen auszuwerten, werden immer häufiger Methoden aus der Domäne des Machine-Learnings im wirtschaftlichen Kontext eingesetzt (Mills 2017). Dabei erstreckt sich der Aufgabenbereich von der Bild Klassifizierung bis hin zur Übersetzung von natürlicher Schrift. Machine-Learning beschreibt dabei den Prozess, Computerprogramme so zu definieren, dass sie aus Erfahrungen und zurückliegenden Ereignissen lernen können und anhand dessen in der Lage sind, Prognosen für neue Datenpunkte oder zukünftige Ereignisse zu treffen. Der Unterschied zur herkömmlichen Programmierung ist dabei, dass nicht alle notwendigen Einschränkungen definiert werden müssen, da das Programm in der Lage sein sollte diese selbstständig herauszufiltern und zu erkennen (Alpaydin 2020, S. 3).

Wie beschrieben, ist ein Anwendungsgebiet für Machine-Learning die Vorhersage von zukünftigen Ereignissen auf der Grundlage historischer Daten, mit welchen sich auch die vorliegende Arbeit beschäftigen wird. Das Ziel der Arbeit ist es, eine Zeitreihe, welche Information zu intralogistischen Auftragsdaten beinhaltet, zu untersuchen und anhand der vorliegenden historischen Daten eine möglichst genaue Prognose für den weiteren Verlauf der Zeitreihe zu erstellen. Damit soll neben der Problematik der Zeitreihenanalyse, die Fragestellung beantwortet werden, inwiefern sich der Prognosefehler der aktuell genutzten Prognose, mit der Verwendung von Methoden aus dem Deep-Learning-Bereich minimiert lässt. Deep Learning ist dabei ein Teilbereich des maschinellen Lernens und beschreibt den Prozess hinter der Erstellung von tiefgreifenden neuronalen Netzen.

Bei den untersuchten Auftragsdaten handelt es sich um Betriebsinterne Daten, welche in der Intralogistik gesammelt worden sind. Die Aufträge bestimmen maßgeblich die Prozesse sowie die Auslastung des Lagers. Mit einer Zielgenauen Vorhersage der Zeitreihe wäre es damit möglich, benötigte Ressourcen optimal zu verteilen und einzuplanen.

## 1.1 Aufgabenstellung

Durch den erhöhten Praxisbezug der Arbeit und die Notwendigkeit, Daten zu besitzen, an welchen die später dargestellten theoretischen Aspekte des Deep Learnings angewendet werden können, erfolgt die Bearbeitung und Erstellung der Bachelorarbeit in Kooperation mit dem in Hamburg ansässigen Drogeriehändler Budnikowsky.

Für die Beschreibung der Problemstellung spielt die Domäne, in welcher das Projekt umgesetzt wird, eine übergeordnete Rolle, weshalb im Folgenden die Problemstellung unter Bezugnahme zu der Domäne erläutert wird. Dafür wird der Prozess dargestellt, durch welchen die intralogistischen Auftragsdaten generiert werden. Anhand der Prozessbeschreibung kann anschließend die konkrete Aufgabenstellung, sowie die daraus resultierende Zielsetzung definiert werden.

Das im Laufe der Arbeit erstellte Modell sollte anhand von historischen Daten lernen und basierend auf den daraus gesammelten Erfahrungen den weiteren Verlauf der Auftragsdaten prognostizieren können. Eine Vorhersage anhand historischer Daten ist nur unter der Voraussetzung möglich, dass die Daten einen hohen Zusammenhang zum prognostizierenden Zeitraum der Zeitreihe aufweisen.

Sollte dies nicht der Fall sein, kann es dazu kommen, dass die Vorhersage unpräzise ist oder dass während der Erstellung des Modells auftretende Korrelationen fälschlicherweise als Kausalzusammenhänge angenommen werden. Die damit Betrachteten Scheinkorrelationen ließe sich mit der Inbetriebnahme des Modells nicht reproduzieren, was zu einer unpräziseren Vorhersage und der Nichtverwendbarkeit des Modells führen würde. Um dem vorzubeugen und auszuschließen, dass Daten verwendet werden, welche nicht genügend Informationen über den weiteren Verlauf der Zeitreihe beinhalten, werden im Folgenden die Bestellprozesse im Logistikzentrum von Budnikowsky dargestellt. Für die Darstellung des Prozesses wird auf erworbenes Wissen aus Experteninterviews zurückgegriffen; der Leitfaden zu den geführten Interviews ist im Anhang 1 einsehbar, die daraus entstandenen Dialoge befinden sich im Anhang 2-3.

Die konkrete Mitarbeiterplanung im Logistikzentrum von Budnikowsky erfolgt eine Woche im Voraus, weshalb das Modell darauf ausgerichtet sein sollte, mindestens sieben Zeitschritte prognostizieren zu können. Aktuell erfolgt die Ressourcenplanung anhand einer Prognose, welche auf Durchschnitts- und Erfahrungswerten beruht.

Zum Zeitpunkt der Erstellung der Arbeit besitzt Budnikowsky in etwa 200 Filialen, welche sich größtenteils im Raum Hamburg befinden. Die Lieferketten aller Filialen laufen über das gleiche Logistikzentrum, damit bilden die Bestellungen aller Filialen die Grundgesamtheit der zu prognostizierenden Zeitreihe. Der Bestellauftrag einer Filiale wird automatisch anhand der aktuellen Lagerbestände der Filiale generiert, anschließend kann der Auftrag noch von einem befugten Mitarbeiter der Filiale manuell angepasst werden. Sobald die Bestellung zusammengestellt und bestätigt worden ist, wird sie an das Logistikzentrum weitergeleitet. Die Frequenz, in welcher eine Filiale im Lager Ware bestellt, ist von deren Größe sowie Standort abhängig und liegt zwischen zwei- und sechsmal pro Woche. Die Bestellhäufigkeit einer Filiale ist dabei gleichbleibend, kann jedoch im Saisongeschäft abweichen. Sobald alle Bestellungen für den jeweiligen Tag eingegangen sind, kann mit der Kommissionierung der Ware im Lager begonnen werden.

Die Kommissionierung beschreibt den Prozessschritt, in welchem aus einer großen Warenmenge eine spezifische, auftragsabhängige Teilmenge zusammengestellt wird. Damit übernimmt Kommissionierung einen essenziellen Part in der Wertschöpfungskette der Intralogistik (Stinson 2012).

Nachdem der Auftrag einer Filiale kommissioniert worden ist, wird die Ware für den Versand bereitgestellt. Der Prozess vom Bestelleingang bis hin zur Bereitstellung im Versand nimmt in etwa 48 Stunden in Anspruch. Alle Filialen werden in Abhängigkeit der zugeordneten Priorität in Batches unterteilt. Ein Batch beschreibt dabei eine Teilmenge der gesamten Auftragslast. Der Batch und die dazugehörigen Aufträge durchlaufen zuerst den gesamten Lagerprozess, bevor der nächste Batch gestartet wird. Dies hat den Vorteil, dass verschiedene Aufträge priorisiert werden und damit die Kapazitäten sowie Ressourcen im Lager gleichmäßig verteilt werden können. Pro Tag werden alle Aufträge in drei bis vier Batches aufgeteilt und die Kommissionierung erfolgt in drei unterschiedlichen Lagerbereichen.

Die Mitarbeiterplanung der einzelnen Lagerbereiche ist unabhängig voneinander, wodurch eine Vorhersage der Auftragsdaten pro Lagerbereich notwendig ist, da sonst kein Mehrwert aus den Analyseergebnissen generiert werden kann. Schlussfolgernd besteht die konkrete Aufgabenstellung darin, die im Logistikzentrum einlaufenden Auftragsdaten der verschiedenen Filialen mit einer Vorlaufzeit von sieben Tagen vorherzusagen, wobei eine Vorhersage pro Lagerbereich notwendig ist.

Der Prozess, von der Bestellung einer Filiale bis hin zur Auslieferung der Ware ist außerdem in Anhang 4 grafisch dargestellt.

## 1.2 Vorgehensweise und Zielsetzung

Für die Bearbeitung der Problemstellung wird zu Beginn der Arbeit grundlegende Aspekte von Zeitreihen sowie deren Komponenten dargestellt. Anschließend werden die Grundlagen und die Entwicklung des Machine-Learnings erläutert, wobei sich bei der Vorstellung auf Aspekte beschränkt wird, welche auch für das weiterführende Deep Learning relevant bleiben. Aufbauend auf den Grundlagen des Machine-Learnings werden dann aus dem Bereich des Deep Learnings konkrete Architekturen zur Analyse erläutert. Das anschließende Vorgehen zur Umsetzung der dargestellten Theorie orientiert sich an üblichen Vorgehensmodellen aus dem Data-Science-Bereich. Dabei wird anhand einzelner Projektphasen die vorgestellte Theorie praktisch umgesetzt und die daraus entstehenden Ergebnisse anhand verschiedener Metriken evaluiert. Als Resultat wird die Architektur, welche die besten Ergebnisse erzielen konnte, weiter untersucht, um in der Zukunft liegende Auftragsdaten zu prognostizieren.

Die Wertschöpfung im Lager beruht auf der Lagerung, Kommissionierung und Auslieferung von Ware. Dies ermöglicht es, Artikel in großen Mengen zu kaufen und den Preis pro Stück zu reduzieren. Durch die Lagerung und die anschließende Kommissionierung kann die Ware bedarfsspezifisch ausgeliefert werden, wodurch die Ware wiederum zu einem höheren Stückpreis verkauft werden kann. Damit stellen die Überwachung und das Reporting von Auftragsdaten ein essenzielles Werkzeug in der Logistik dar. Die Vorhersage der Auftragsdaten würde somit Einfluss auf die Verteilung von Ressourcen und auf die Planung von Lagerbeständen nehmen. Außerdem könnten Engpässe verhindert werden, indem ein ungewöhnlich hohes Auftragsvolumen bereits im Vorfeld bekannt ist.

Resultierend daraus hat die vorliegende Arbeit das Ziel, grundlegende theoretische Aspekte von Zeitreihen, Machine-Learning und Deep Learning darzustellen und anzuwenden. Mit erfolgreicher Umsetzung und Modellierung des Modells werden die erreichten Analyseergebnisse mit der aktuell im Lager getroffenen Prognose verglichen, um so eine Aussage über die Verwendbarkeit des Modells treffen zu können. Damit sollte sich die Fragestellung beantworten lassen, ob eine Inbetriebnahme des Modells in den Live-Betrieb ratsam ist und inwiefern sich mittels des Einsatzes, der auf Deep Learning beruhenden Analysemethoden, ein Mehrwert generieren lässt.

### 1.3 Verwandte Arbeiten

Im folgenden Kapitel werden verwandte Arbeiten vorgestellt, welche sich bereits mit der Anwendung von Methoden aus dem Deep-Learning-Bereich für die Zeitreihenprognose oder die Bedarfsplanung in der Logistik beschäftigt haben.

Das Prognostizieren des zukünftigen Verhaltens einer Zeitreihen ist für verschiedene Domänen relevant. Damit wurden für die Vorhersage von Zeitreihen bereits verschiedene Modelle entworfen und untersucht. Traditionelle Ansätze sind dabei der gleitende Durchschnitt, das exponentielle Glätten, Regressionsmodelle oder das ARIMA-Modell (Hyndman und Athanasopoulos 2021). Zusätzlich wurden auch vermehrt Prognosemodelle mittels Methoden aus dem Machine-Learning erstellt und untersucht. Ein Beispiel dafür ist die Studie von García (Turrado García, García Villalba und Portela 2012).

Zeitreihenanalysen mittels Deep-Learning-Algorithmen konnten lange keine nennenswerten Ergebnisse erreichen. Einer der ersten Erfolge im Vergleich zu traditionellen Ansätzen konnten neuronale Netze 1996 verzeichnen (Hill, O'Connor und Remus 1996, S.1088). Spätere Arbeiten basierend auf verschiedenen Varianten von Recurrent neural networks (RNN) (Hochreiter und Schmidhuber 1997) konnten bei der Analyse von Sequenziellen Daten weitere außergewöhnliche Ergebnisse erzielen. Dabei sind zu nennen die Arbeiten von Li und Yu (Yu u. a. 2019) (Y. Li u. a. 2018).

Ein Problem in den vorgestellten Arbeiten war, dass die Genauigkeit der Modelle mit der steigenden Länge der Prognosezeitraumes stark abnahm. Damit konnten einzelne Zeitschritte, mit der mittlerweile entwickelten, Encoder-Decoder-Architektur zwar mit hoher Präzision vorhergesagt werden, jedoch nahm die Genauigkeit im Laufe der Ausgabesequenz stark ab (Qin u. a. 2017).

Ein weiteres Problem war die Verarbeitung von langen Eingabesequenzen; mit steigender Länge der Eingabesequenz nahm besonders die Genauigkeit von RNN-Architekturen ab (Zhou u. a. 2021). Die Probleme, welche bei dem Einlesen von langen Eingabesequenzen auftraten, konnten durch den Einsatz von Convolutional neural networks (CNN) verringert werden (Bai, Kolter und Koltun 2018) (Stoller u. a. 2019).

Allerdings blieben die beschriebenen Probleme der abnehmenden Genauigkeit bei der Vorhersage von mehreren Zeitschritten bestehen. Mit dem Einsatz von Transformern (Vaswani u. a. 2017) konnte sowohl die Genauigkeit einzelner Vorhersagen als auch die Sequenzieller Prognosen signifikant verbessert werden (Zhou u. a. 2021) (Wu u. a. 2020).

In den folgenden Kapiteln wird die Theorie hinter den genannten Ansätzen besprochen. Anschließend kann darauf aufbauend ein Artefakt erstellt werden, welches das beschriebene Problem von langen Ein- und Ausgabesequenzen aufgreift.

## 2. Zeitreihen

Zeitreihen beschreiben alle Datensätze, welche die Entwicklung einer oder mehrerer Variablen über einen Zeitraum abbilden. Dabei lassen sie sich der Gruppe der Längsschnittdaten zuordnen. Längsschnittdaten sind das Ergebnis von Datenerhebungen, die in einer zeitlichen Abfolge eine gleichbleibende statistische Einheit messen (Backhaus u. a. 2018, S. 126). Damit besteht eine Zeitreihe aus einer Reihe von Datenpunkten, die in einer zeitlichen Reihenfolge aufgelistet werden. Schlussfolgernd lassen sich diverse Kennzahlen als Zeitreihe darstellen, wie die Entwicklung von Umsätzen, Preise von Produkten oder des Wetters. Die Zeitschritte können dabei in ihrer Größe variieren, damit können Zeitreihen die Abfolge von Daten im Sekundentakt bis hin zu gemessenen Zeitpunkten im Jahrestakt darstellen. Durch die Einfachheit und das damit verknüpfte intuitive Verständnis, eignen sich Zeitreihen besonders in der deskriptiven Statistik für Visualisierung verschiedener Ereignisse. (Frick u. a. 2021, S. 203)

### 2.1 Zeitreihenkomponenten

Allgemein wird davon ausgegangen, dass Datenpunkte in Zeitreihen, in regelmäßigen Abständen erzeugt werden; sollte dies nicht der Fall sein, wird von einer unregelmäßigen Zeitreihe gesprochen. Weiterhin können Zeitreihen anhand der betrachteten Variablen, welche sich im untersuchten Zeitraum verändert haben, unterschieden werden. Verändert sich nur eine Variable im betrachteten Zeitraum oder wird nur eine Variable betrachtet, wird von einer univariaten Zeitreihe gesprochen. Eine univariate Zeitreihe liegt häufig bei Sensordaten vor, da diese zu jedem Zeitpunkt einen eindimensionalen Wert erzeugen. Sollten sich mehrere Werte über einen betrachteten Zeitraum verändern, wird von einer multivariaten Zeitreihe gesprochen. Damit stellt eine multivariate Zeitreihe die Betrachtung von mehreren univariaten Zeitreihen im gleichen Zeitraum da.

Die Variablen in der multivariaten Zeitreihe stehen allerdings im Verhältnis zueinander, wodurch die eigentliche Aussagekraft der Zeitreihe erst mit der Betrachtung der verschiedenen Parameter erkennbar ist. Sollte es nicht klar sein, inwiefern die Veränderung der Zeitreihe von anderen gemessenen Parametern abhängig ist, kann eine Falsche Annahme in der Analyse zur Folge haben, dass Scheinkorrelationen zwischen den einzelnen untersuchten Zeitreihen als Kausalzusammenhang angenommen werden, wodurch die Gefahr besteht, dass eine falsche Schlussfolgerung aus den Analyseergebnissen gezogen wird. Damit sollte unter Bezug von Domänenexperten der Kausalzusammenhang zwischen den beobachteten Parametern geklärt werden (Backhaus u. a. 2018, S. 134).

Weiterhin können die observierten Bestandteile einer Zeitreihe in folgenden Komponenten aufgeteilt werden (vgl. Abbildung 1):

- Ein Trend beschreibt die langfristige Veränderung des mittleren Niveaus; damit zeichnet sich bei der Trendanalyse das lineare Fallen oder Steigen der Zeitreihe ab.
- Saisonalitäten beschreiben das wiederholte Auftreten von ähnlichen Mustern entlang der Zeitachse.
- Außerdem können Zeitreihen durch zufällige Ereignisse geprägt worden sein, wodurch ein unstrukturiertes Muster entsteht, welches als Rauschen oder Zufalls Komponente bezeichnet wird und kaum bis gar nicht vorhersagbar ist.

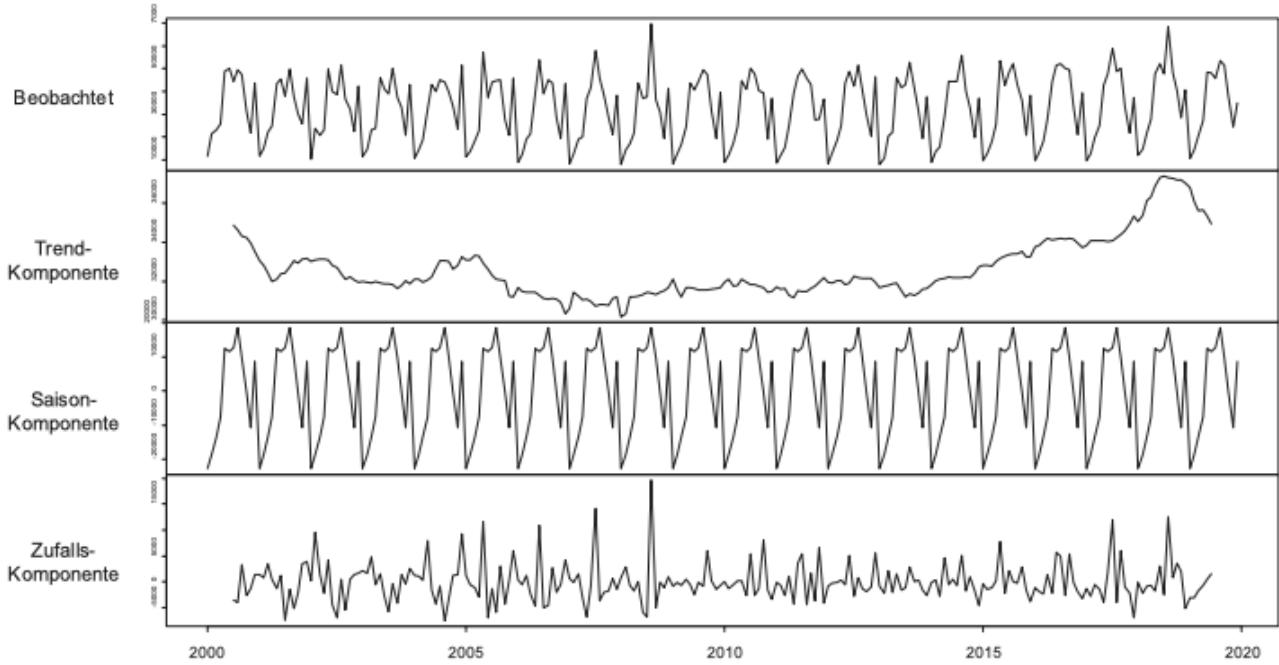


Abbildung 1: Zeitreihenkomponenten (Frick u. a. 2021, S. 203)

## 2.2 Zeitreihenanalyse

Neben der visuellen Darstellung lassen sich die erhobenen Daten auch für eine Analyse verwenden. Die Analyse von Zeitreihen verfolgt dabei zweierlei Ziele: Zum einen können vergangene Ereignisse mithilfe der festgehaltenen Daten erklärt werden und zum anderen kann das bisherige Verhalten der Zeitreihe genutzt werden, um die zukünftige Entwicklung der Zeitreihe zu prognostizieren (Eckstein 1999, S. 273).

Die Prognose von zukünftigen zeitschritten, lässt sich anhand der getätigten Vorhersage in folgende Gruppen einteilen:

- Punktprognose: Bei dieser wird eine genaue Vorhersage für einen zukünftigen Zeitschritt getroffen; das Ergebnis der Vorhersage kann anschließend anhand eines ausgewählten Fehlermaßes evaluiert werden umso eine Aussage über die Genauigkeit der getätigten Prognose treffen zu können. Das Ziel ist es, damit eine Prognose zu liefern, bei welcher die errechnete statistische Abweichung möglichst klein ist (Backhaus u. a. 2018, S. 134).
- Probabilistische Vorhersagen: Bei der Prognose wird ein Zeitschritt nicht punktuell vorhergesagt, sondern die Wahrscheinlichkeitsverteilung des zukünftigen Wertes dargestellt. Damit kann eine Aussage darüber getroffen werden, mit welcher Wahrscheinlichkeit sich der zukünftige Wert in einem bestimmten Intervall befindet. Häufig werden dafür Vorhersageintervalle verwendet (Hyndman und Athanasopoulos 2021, S. 25).

Das Modell, welches im weiteren Verlauf der Arbeit erstellt wird, soll zum einen in der Lage sein, eine punktuelle Prognose für einen zukünftigen Zeitschritt abzugeben, und zum anderen soll anhand von Prognoseintervallen eine probabilistische Vorhersage getroffen werden. Anschließend kann mit dem Vergleich der beiden Prognosen eine Aussage über die Sicherheit der Vorhersage getroffen werden.

### 3. Machine-Learning

Im Folgenden werden die Grundlagen des Machine-Learnings dargestellt, wobei sich auf die Teilbereiche beschränkt wird, welche auch für den darauffolgenden Abschnitt des Deep Learnings relevant sind.

#### 3.1 Grundlagen

Machine-Learning ist die Wissenschaft, Computerprogramme so zu definieren, dass es für diese möglich ist, anhand von Daten zu lernen, entsprechend der folgenden Definition:

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E ”. (Mitchell 1997)

Sollte die Aufgabe des Machine-Learning-Programms damit die Klassifikation von Hatespeech im Internet sein, ist die Aufgabe T klar definiert. Die Erfahrungen E können dann mithilfe von bereits gespeicherter Hate- und Nicht-Hatespeech dargestellt werden. Diese Daten können vom System als Lernbeispiele verwendet werden und stellen damit den Trainingsdatensatz dar. Jedes Beispiel innerhalb des Datensatzes wird als Trainingsdatenpunkt bezeichnet. Das Definieren des Leistungsmaßes T wird dabei individuell und in Abhängigkeit von der Aufgabe bestimmt und kann im Beispiel der Hatespeech-Klassifizierung der Anteil der Grundgesamtheit sein, welchen das Programm korrekt klassifiziert hat.

Der Vorteil von Machine-Learning-Systemen im Verhältnis zu herkömmlichen Programmietechniken besteht darin, dass nicht jede Bedingung eigenständig definiert werden muss. Im Beispiel der Erkennung von Hatespeech im Internet wäre es mit den herkömmlichen Methoden notwendig, einzelne Bedingungen zu definieren. Mit der Definition aller notwendigen Bedingungen, wäre das Programm einsatzbereit. Sollte eine Lücke gefunden werden, muss das Programm nachträglich nachgebessert und eine weitere Bedingung hinzugefügt werden. Damit wäre es notwendig, dass ein solches Programm, geschrieben mit der herkömmlichen Programmietechnik, stetig unter Beobachtung steht. Ein Machine-Learning-Programm hingegen könnte mit einem erweiterten Trainingsdatensatz erneut und automatisiert trainiert werden, wodurch die gefundene Systemlücke geschlossen werden würde. Weiterhin kann es in diversen Aufgaben, wie auch im genannten Beispiel, dazu kommen, dass es nicht leicht oder nicht möglich ist, alle Bedingungen mit der nötigen Komplexität manuell zu definieren. Dies kann zum einen aus der Menge an zu definierenden Bedingungen resultieren, zum anderen kann es auch der Fall sein, dass die Bedingungen in ihrer notwendigen Komplexität unbekannt oder manuell nicht definierbar sind.

Damit sind Programme, die auf dem maschinellen Lernen beruhen, vor allem für Aufgaben geeignet, bei welchen sowohl in der Programmierung als auch im Nachhinein ein hohes Maß an Genauigkeit und Feinarbeit benötigt wird oder nicht alle Parameter und Abhängigkeiten bekannt sind, wodurch es mit den herkömmlichen Methoden in Abhängigkeit vom gewählten Leistungsparameter zu keiner zufriedenstellenden Lösung kommen würde. Resultierend daraus ist ein Programm, das auf maschinellem Lernen basiert, nicht zwingend herkömmlichen Methoden überlegen, kann aber gegebenenfalls in verschiedenen Anwendungsfällen zu einem besseren Ergebnis führen oder den benötigten Aufwand verringern (Géron 2019, S. 4).

### 3.2 Lernverfahren

Systeme, welche auf Machine-Learning beruhen, lassen sich anhand der Art und Weise, wie sie während des Trainings überwacht werden, unterscheiden. Dabei erfolgt einer Unterteilung in die folgenden vier Kategorien: Überwachtes Lernen, unüberwachtes Lernen, halbüberwachtes Lernen und verstärkendes Lernen.

Bei der Erstellung des Modells wird lediglich das überwachte Lernen verwendet. Beim überwachten Lernen wird das Modell anhand von Daten, welche bereits ein Label besitzen, trainiert. Das bedeutet, dass zu jedem Datenpunkt, welcher als Eingabe dient, auch die bereits passende Lösung vorhanden ist. Im Beispiel der Zeitreihen ist damit zu jeder Eingabesequenz auch die zu prognostizierende Ausgabesequenz vorhanden. Dies hat den Vorteil, dass direkt mit der Ausgabe des Modells die Qualität der Aussage anhand eines definierten Leistungsmaßes gemessen werden kann. Anschließend kann in Abhängigkeit des Prognosefehlers Anpassungen am Modell vorgenommen werden (Han, Pei und Kamber 2011).

Die Begrifflichkeit des Labels für die richtige Prognose wird im Folgenden beibehalten.

### 3.3 Instanzbasiertes und modellbasiertes Lernen

Bei fast allen Systemen, welche auf der Anwendung von Machine-Learning-Algorithmen beruhen, geht es entweder um das Treffen einer Vorhersage von Gruppenzugehörigkeiten, die Klassifikation, oder um eine Vorhersage von stetigen Werten, die Regression. Das Ziel ist es somit in beiden Fällen, in Abhängigkeit vom definierten Leistungsmaß einen möglichst geringen Prognosefehler auf unbekannten Datenpunkten zu erreichen. Für die Erreichung des Ziels können Machine-Learning-Systeme instanzbasiert oder modellbasiert lernen.

Beim instanzbasierten Lernen lernt das Modell die Beispiele, die sich im Trainingsdatensatz befinden, auswendig und kategorisiert neue Datenpunkte anhand eines Ähnlichkeitsmaßes. Mithilfe des Ähnlichkeitsmaßes und der Datenpunkte, welche dem neuen Datenpunkt am nächsten sind, kann anschließend eine Prognose erstellt werden (Aha, Kibler und Albert 1991).

Beim modellbasierten Lernen ist es das Ziel, die Aussage, welche die Datenpunkte des Trainingsdatensatzes beinhalten, zu verallgemeinern. Dabei passt sich das Modell analysierten Trends sowie wiederkehrenden Mustern, welche im Trainingsdatensatz erkannt worden sind, an. Durch die Anpassung anhand verschiedener Datenpunkte kann damit eine verallgemeinerte Aussage für neue Datenpunkte getroffen werden (Quinlan 1993).

Für die Erstellung des Artefaktes wird ein modellbasierter Ansatz verwendet, weshalb im Folgenden der Begriff Modell für die Beschreibung des zu erstellenden Programms gewählt wird.

### 3.4 Overfitting und Underfitting

Durch das Erkennen von Mustern im Trainingsdatensatz ist es möglich, dass das Modell eine Prognose für zukünftige Datenpunkte treffen kann. Datenpunkte beinhalten oftmals eine rauschende Komponente. Das Rauschen stellt Informationen im Datenpunkt dar, welche von äußerem, nicht vorhersehbaren Ereignissen generiert worden sind (vgl. Kapitel 2.1).

Sollte sich das Modell zu stark an die Trainingsdaten anpassen und die identifizierten Informationen verallgemeinern, kann dies dazu führen, dass das Modell Muster in der rauschenden Komponente

erkennt und auch bei neuen Datenpunkten ein ähnliches Rauschen prognostiziert. Dies führt oftmals zu einem guten Ergebnis auf dem Trainingsdatensatz, jedoch lassen sich die erkannten Muster der rauschenden Komponente bei neuen Datenpunkten nicht reproduzieren; resultierend daraus nimmt die Leistung des Modells mit der Inbetriebnahme merklich ab. In einem solchen Fall wird von Overfitting gesprochen, da sich das Modell zu stark an den Trainingsdatensatz angepasst hat. Präventiv kann Overfitting mit einem früheren Abbruch des Trainingsprozesses oder mit einer Erweiterung des Trainingsdatensatzes verhindert werden.

Sollte die Datenlage nicht ausreichend sein oder die Komplexität der Realität die des Modells übersteigen, kann es dazu kommen, dass das Modell keine Lösung oder Vorhersage treffen kann. In einem solchen Fall wird von Underfitting gesprochen. Underfitting kann auch aus dem verfrühten Abbrechen des Trainingsprozesses resultieren (Badillo u. a. 2020, S. 6).

### 3.5 Testen und Validieren

Mithilfe der Betrachtung der Ergebnisse während des Trainingsprozesses kann keine Aussage darüber getroffen werden, wie gut die Prognosen des Modells auf unbekannten Datenpunkten und damit im eigentlichen Livebetrieb sein werden. Zu begründen ist diese Aussage mit der vorgestellten Problematik des Overfittings (vgl. Kapitel 3.4).

Resultierend daraus bleibt die einzige Möglichkeit für die Evaluierung eines Modells das Erstellen von Prognosen für unbekannte Datenpunkte. Um zu garantieren, dass eine ausreichende Datenmenge, zum Testen zur Verfügung steht, lässt sich der untersuchte Datensatz in Trainings- und Testdaten unterteilen.

Für die Erstellung des Testdatensatzes wird von der Datengrundlage ein Teil abgetrennt und nicht für den Trainingsprozess des Modells verwendet. Damit sind die Datenpunkte im Testdatensatz für das Modell unbekannt. Mithilfe dieser Methodik kann das Modell, welches auf den Trainingsdaten trainiert worden ist, anschließend auf den Testdaten evaluiert werden und die Testphase kann bereits vor der Inbetriebnahme des Modells abgeschlossen werden.

Mit der Unterteilung ergibt sich das Problem, dass mit der Evaluierung des Modells auf dem Testdatensatz keine Änderungen mehr am Modell vorgenommen werden dürfen. Das ist mit der Tatsache zu begründen, dass mit einer Optimierung nach der Testphase der Bias des Entwicklers in die Veränderung des Modells einfließen würde. Dies hat zur Folge, dass nach der Optimierung des Modells erneut keine Aussage über die zu erwartende Performance auf unbekannten Datenpunkten und im Livebetrieb getroffen werden kann.

Damit das Modell mehrfach auf den Testdaten evaluiert werden kann, besteht die Möglichkeit, den Datensatz weiter zu unterteilen; diese Methode wird als K-Fold Cross-Validation beschrieben. Dabei kann der Datensatz beliebig oft unterteilt werden, woraus sich die Möglichkeit ergibt, die Ergebnisse anhand verschiedener Testinstanzen zu validieren. Allerdings muss dabei beachtet werden, dass mit jeder weiteren Unterteilung, der Trainingsdatensatz verkleinert wird, was sich gegebenenfalls auf die Ergebnisse auswirken kann (Hastie, Tibshirani und Friedman 2001, S. 241).

In der vorliegenden Arbeit erfolgt die Unterteilung in einen Trainings-, einen Test- und einen Evaluierungsdatensatz. Dies hat den Vorteil, dass nach erfolgreichem Trainingsprozess und Evaluierung auf dem Evaluierungsdatensatz das Modell weiterhin optimiert werden kann, ohne dass der Testdatensatz in den Prozess einzbezogen werden muss.

Schlussfolgernd daraus kann das Modell trainiert und anhand unbekannter Datenpunkte optimiert werden und der Testdatensatz bleibt sowohl für das Modell als auch für den Entwickler unbekannt. Sobald alle Anpassungen vorgenommen worden sind und das Modell auf dem Evaluierungsdatensatz eine entsprechende Performance aufweist, kann das Modell mithilfe des Testdatensatzes erneut evaluiert werden. Damit kann eine zielsichere Aussage über die zu erwartende Performance im Livebetrieb getroffen werden. Jedoch gilt dabei weiterhin, dass nach Evaluierung auf dem Testdatensatz keine Anpassungen mehr am Modell vorgenommen werden dürfen.

## 4. Deep Learning

Wie beschrieben, ist es das Ziel der vorliegenden Arbeit, mithilfe von Methoden aus dem Bereich des Deep Learnings den weiteren Verlauf einer Zeitreihe zu prognostizieren (vgl. Kapitel 1.2). Im Folgenden werden die Grundlagen von Deep Learning dargestellt. Anschließend werden theoretische Aspekte, welche für die Erstellung des Modells und die Bearbeitung der Fragestellung relevant sind, spezifiziert. Ziel im Anschluss des Kapitels sollte es sein, dass alle nötigen Grundlagen, welche für die Bearbeitung der Fragestellung sowie für die Umsetzung und Erstellung des Modells notwendig sind, dargestellt worden sind.

### 4.1 Grundlagen

Deep Learning beschreibt den Einsatz von Deep Neuronal Networks als Methodik zur Datenanalyse. Der Begriff „Deep“ ist dabei unscharf definiert und teilweise wird bereits ein künstliches neuronales Netz, welches mehr als eine Schicht besitzt, zu der Kategorie des Deep Learnings gezählt. Damit eine einheitliche Notation gegeben ist, umschließt im Folgenden die Begrifflichkeit der künstlichen neuronalen Netze (ANN) auch Deep Neural Networks. Die Grundzüge von ANNs werden anhand folgender Definition dargestellt:

“A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification.” (Deng 2014)

Demzufolge handelt es sich um eine Technik aus der Domäne des Machine-Learning, bei welchem Informationen mithilfe von nicht linearen Operationen verarbeitet werden. Sie basiert auf einem ähnlichen Grundprinzip wie das Machine-Learning und das Trainieren eines Modells anhand von Daten, wobei auch zwischen verschiedenen Trainingsmethoden wie überwachtem und unüberwachtem Lernen unterschieden wird (vgl. Kapitel 3.1). Die Hauptaufgabe ist dabei, innerhalb der Daten Merkmale und Muster zu erkennen, welche es ermöglichen, die Daten zu klassifizieren, zu analysieren oder zu transformieren.

#### 4.1.1 Künstliche Neuronen

Die Grundidee hinter ANNs stammt aus der Biologie und aus der Arbeitsweise der sich im menschlichen Gehirn befindenden Neuronen. Im Laufe der Entwicklung hat sich die Arbeitsweise der künstlichen Neuronen weitestgehend vom biologischen Vorbild entfernt. Mittlerweile wird deshalb auch vermehrt die Begrifflichkeit Unit anstelle von Neuron verwendet.

Die Funktionsweise neuronaler Netze wird im Folgenden anhand einer mehrschichtigen Perzeptron-Architektur (MLP) dargestellt. Das Perzeptron wurde 1957 von Frank Rosenblatt entwickelt und bildete den Grundstein für weiterführende Architekturen, welche heutzutage verwendet werden.

Das Perzeptron wird dabei in der Literatur vermehrt als einzelnes Neuron beschrieben, weshalb diese Notation im Weiteren beibehalten wird. Das Neuron, welches in Abbildung 2 dargestellt ist, ist in der Lage, numerische Werte als Eingabe zu verarbeiten.

Die Eingabewerte werden dabei als die Inputs bezeichnet. Die Inputs werden an das Neuron weitergegeben, der Transfer der Information wird dabei mit Verbindungen dargestellt.

Jeder Verbindung wird dabei ein Gewicht zugewiesen, wobei es sich auch um einen numerischen Wert handelt. Mithilfe der Inputs und der zugewiesenen Gewichte ist das Neuron in der Lage, die gewichtete Summe der Eingabewerte zu berechnen (vgl. Formel 1).

Anschließend wird eine Aktivierungsfunktion auf die errechnete Summe angewendet (vgl. Formel 2). Die Aktivierungsfunktion definiert einen Schwellenwert und bestimmt damit den Aktivitätszustand des Neurons. Überschreitet das Aktivitätslevel den definierten Schwellenwert der Aktivierungsfunktion, wird von einem aktiven Neuron gesprochen; sollte das Aktivitätsniveau unter dem Schwellenwert liegen, wird von einem inaktiven Neuron gesprochen und es wird keine Ausgabe weitergegeben (Rosenblatt 1958).

Mit der beschriebenen Funktionsweise ist ein einzelnes Neuron bereits in der Lage, einfache binäre Klassifikationsaufgaben zu lösen. Dies geschieht, wie beschrieben, mit der Berechnung der Linear-kombination der Eingabewerte. Anschließend gibt das Neuron eine positive Kategorie zurück, sobald das Aktivitätslevel den Schwellenwert der Aktivierungsfunktion überschreitet, anderenfalls gibt das Neuron eine negative Kategorie aus (Suzuki 2011, S. 3).

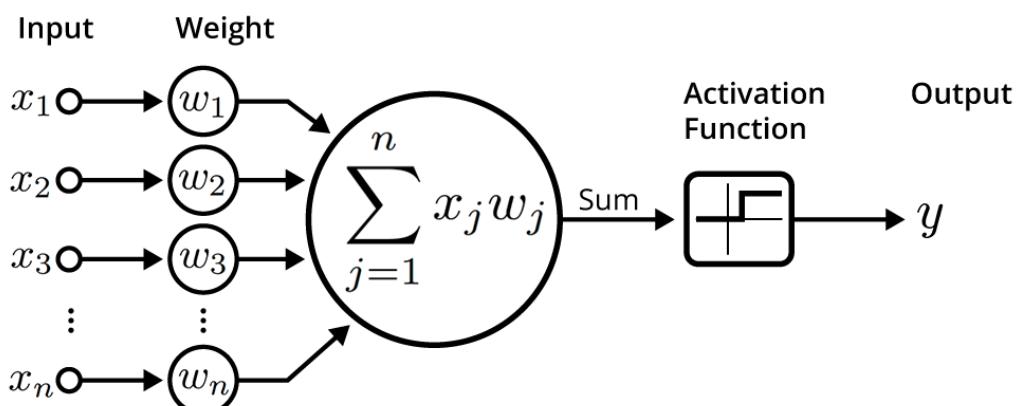


Abbildung 2: Ein Einzelnes Neuron (Saxena 2017)

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n$$

$$z = w^T \cdot x$$

Formel 2: Gewichtete Summe der Inputs

$$h_w(x) = \text{Aktivierungsfunktion}(z) \text{ mit } z = w^T \cdot x$$

Formel 1: Ausgabe Neuron

#### 4.1.2 Künstliche Neuronale Netze

Neuronale Netze werden zur Analyse von Daten verwendet. Damit müssen sie in der Lage sein, Informationen aufzunehmen, zu verarbeiten und auszugeben. Entsprechend lassen sich die Hauptaufgaben damit wie folgt unterteilen:

- Informationsaufnahme
- Informationsverarbeitung
- Informationsausgabe

Ein ANN besteht zur Bewerkstelligung der Aufgaben aus mehreren Schichten von Neuronen. Innerhalb jeder Schicht wird eine beliebige Anzahl von Neuronen parallel angeordnet; die Ausgabe der Neuronen der vorgelagerten Schicht dient dabei als Eingabe für jedes Neuron der nachgelagerten Schicht (vgl. Abbildung 3).

Die verschiedenen Schichten innerhalb eines ANN bewerkstelligen dabei verschiedene Aufgaben und lassen sich anhand dessen klassifizieren:

- Input-Layer: Der Input-Layer, in Abbildung 3 links dargestellt, stellt die erste Schicht eines ANNs dar. Der Layer verarbeitet Eingabedaten von der Außenwelt, anschließend werden diese an die nachgelagerte Schicht weitergegeben. Die Hauptaufgabe des Input-Layer ist damit das Einlesen der übergebenen Daten; damit muss er in der Lage sein, die zur Verfügung gestellten Informationen in ihrer Dimension verarbeiten zu können.
- Output-Layer: Diese Schicht stellt die letzte Schicht des neuronalen Netzes dar, entsprechend ist sie auch rechts in der Abbildung 3 dargestellt. Die Anzahl verwendeter Neuronen im Output-Layer ist dabei abhängig von der Anzahl der gewünschten Prognosen und Ausgaben. Für jede Ausgabe wird ein Neuron benötigt. Beispielsweise wird bei einer Klassifikationsaufgabe ein Neuron pro Klasse benötigt; die Ausgabe eines Neurons stellt dann die Wahrscheinlichkeit dar, mit welcher es sich bei dem Input um die jeweilige Klasse handelt. Die Klasse mit der höchsten Wahrscheinlichkeit kann anschließend als Vorhersage identifiziert werden.
- Hidden Layer: Alle Schichten, welche weder Input- noch Output-Layer sind, werden als Hidden Layer bezeichnet und befinden sich zwischen den beiden zuerst genannten Layern. Ein ANN kann dabei beliebig viele Hidden Layer besitzen. Innerhalb jedes Hidden Layers werden die Eingabedaten weiter transformiert, wodurch die Komplexität des ANN weiter ansteigt.

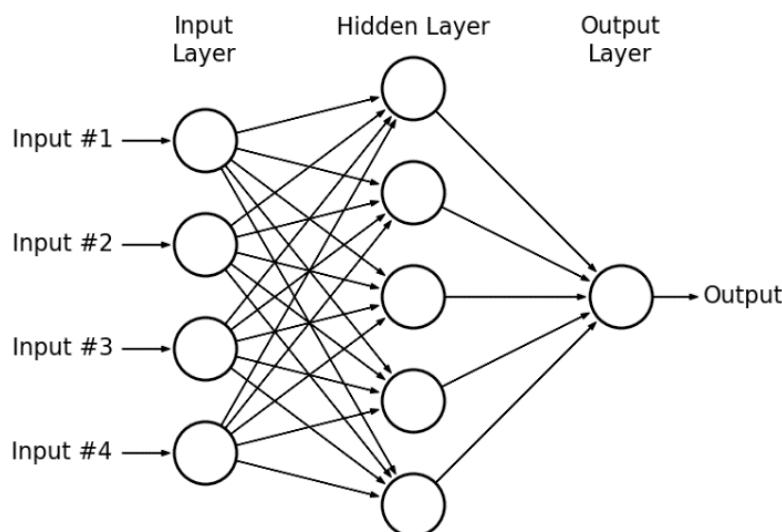


Abbildung 3: Einfaches Neuronales Netz (Hassan u.a 2015, S.252)

Außerdem wird von einem Fully-Connected- oder Dense-Layer gesprochen, sobald alle Neuronen einer Schicht mit allen Neuronen der nachgelagerten Schicht verbunden sind. Damit handelt es sich bei jedem Layer des in Abbildung 3 dargestellten neuronalen Netzes um einen Dense Layer. Jede Schicht, mit Ausnahme des Output Layers, wird zusätzlich mit einem Bias-Neuron initialisiert. Das Bias-Neuron wird mit dem Wert 1 initialisiert und bleibt im Laufe des Trainingsprozesses konstant. Das Bias-Neuron ist mit jedem Neuron der nachgelagerten Schicht verbunden; den Verbindungen des Neurons sind dabei ebenfalls Gewichte zugewiesen. Das Bias-Neuron kann so direkten Einfluss auf das durchschnittliche Aktivitätsniveau jedes Neurons nehmen, da der Bias-Therm konstant bleibt und somit die Veränderung der Gewichte unabhängig von der Eingabe der vorgelagerten Schicht Einfluss auf die Ausgabe nimmt (Goodfellow, Bengio und Courville 2016, S. 163).

Wie beschrieben, kann ein einzelnes Neuron bereits für einfache binäre Klassifikationsaufgaben eingesetzt werden. Durch die parallele Anordnung der Neuronen innerhalb einer Schicht und das Hintereinanderschalten mehrerer Schichten ist das neuronale Netz in der Lage, deutlich komplexere Aufgaben zu lösen. Dies resultiert daraus, dass der Trainingsprozess eines ANN auf dem Verändern von Verbindungsgewichten anhand des berechneten Prognosefehlers beruht. Durch die höhere Anzahl von Schichten und Neuronen lassen sich dabei mehr Verbindungsgewichte verändern, wodurch das Netz in der Lage ist, vielschichtigere Merkmale innerhalb der Eingabesequenz zu identifizieren (Géron 2019, S. 290). Der Trainingsprozess des neuronalen Netzes wird in Kapitel 4.2 weiter spezifiziert.

Beim ersten Durchlauf der Eingabedaten durch das neuronale Netz werden alle Verbindungsgewichte mit zufälligen Werten initialisiert. Damit lassen sich die Ausgabewerte einer vollständig verbundenen Schicht ähnlich wie bei den einzelnen Neuronen berechnen.

Formel 3 ermöglicht dabei das effiziente Berechnen von allen Ausgabewerten einer Schicht innerhalb des neuronalen Netzes. X steht dabei für die Matrix von Eingabemerkmalen, W ist die Gewichtsmatrix und enthält dabei alle Gewichte mit Ausnahme der Gewichte des Bias-Neurons. Der Bias-Vektor wird mit b notiert und enthält alle Gewichte des Bias-Neurons. Die Funktion  $\Phi$  stellt die Aktivierungsfunktion dar, welche auf den berechneten Term angewendet wird.

Unter der Anwendung von Formel 3 und dem Berechnen der Ausgabewerte jeder Schicht, welche sich im neuronalen Netz befindet, kann das neuronale Netz einmal durchlaufen werden bis hin zur Ausgabeschicht. Die Ergebnisse der Ausgabeschicht stellen anschließend die Prognose des Netzes dar. (Goodfellow, Bengio und Courville 2016, S. 194)

$$h_{W,b}(X) = \Phi(X \cdot W + b)$$

*Formel 3: Ausgabewerte, vollständig verbundene Schicht*

## 4.2 Trainingsprozess

Im vorangegangenen Kapitel wurden die Funktionsweise und der Aufbau eines einfachen ANN beschrieben. Außerdem wurde die Transformation der Eingabewerte innerhalb des Netzes bis hin zur Ausgabe vorgestellt. Durch die Initialisierung der Verbindungsgewichte mit zufälligen Werten wird das Ergebnis jedoch lediglich durch den Zufall bestimmt. Um ein Ergebnis zu erreichen, welches auf Erfahrungen und auf dem Training anhand von verschiedenen Datenpunkten beruht, besteht die Notwendigkeit, nach jedem Durchlauf Änderungen am Netz vorzunehmen. Mit Anpassungen am neuronalen Netz sollte die Prognose nach mehreren Iterationsschritten zu einem richtigen Ergebnis konvergieren. Der Prozess, welcher das Verändern von Parametern innerhalb des Netzes umfasst, wird als Trainingsprozess beschrieben. Im Folgenden wird dieser vorgestellt, wobei sich die Beschreibung

des Trainingsprozesses auf Modelle beschränkt, welche mittels des überwachten Lernens trainiert werden (vgl. Kapitel 3.2).

Das Training eines ANNs lässt sich in folgende vier Phasen unterteilen:

- Forward Propagation
- Bestimmen des Fehlers mittels Verlustfunktion
- Backpropagation
- Aktualisieren von Parametern

#### 4.2.1 Forward Propagation

Die Datentransformation des neuronalen Netzes vom Input bis hin zur Ausgabe, welche im vorangegangenen Kapitel vorgestellt worden ist, wird als Forward Propagation beschrieben. Dabei werden alle Verbindungsgewichte mit zufälligen Werten initialisiert. Die Initialisierung anhand von zufälligen Werten ist notwendig, da sonst alle Neuronen einer Schicht identisch bleiben, wodurch der Trainingsprozess fehlschlagen würde. Mit der zufälligen Initialisierung, wird die Symmetrie innerhalb jeder Schicht gebrochen und ein effizienter Trainingsprozess wird ermöglicht (Svozil, Kvasnicka und Pospichal 1997).

Nachdem die Gewichte mit den zufälligen Werten initialisiert worden sind, wird das neuronale Netz mittels der Forward Propagation durchlaufen (vgl. Kapitel 4.1.2). Anschließend kann die Abweichung des Ausgabewertes im Verhältnis zum Label anhand einer Verlustfunktion berechnet werden.

#### 4.2.2 Verlustfunktion

Die Verlustfunktion berechnet den Fehler, der sich durch die Abweichung der Prognose vom Label ergibt. Damit stellt die Verlustfunktion das Qualitätsmaß dar, anhand dessen die Performance des ANN gemessen wird. In diesem Zusammenhang wird auch die Kostenfunktion erwähnt, mit welcher anschließend der Verlust beziehungsweise Prognosefehler über den gesamten Trainingsdatensatz berechnet wird. Im Folgenden werden dafür verschiedene Arten von Verlustfunktionen vorgestellt, welche sich als Fehlermaß bei einem Regressionsproblem eignen. Weiterhin ist zu erwähnen, dass die Berechnungen der Verlust- und der Kostenfunktion übereinstimmend sind.

Mithilfe des Mittlere quadratische Abweichung (MSE) wird die mittlere Summe der quadratischen Abstände zwischen der Prognose und dem Label bestimmt (vgl. Formel 4)(Chai und Draxler 2014). Üblicherweise wird das Label als  $y$  und die Prognose als  $\hat{y}$  beschrieben; diese Form der Notation wird auch in den weiter vorgestellten Verlustfunktionen beibehalten.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Formel 4: MSE; Mean squared error

Mittels des Mittleren absoluten Fehlers (MAE) kann die mittlere Summe der absoluten Abweichung zwischen der Prognose und dem Label bestimmt werden. Durch ausschließliche Betrachtung des Betrages spielt dabei die Richtung der Abweichung keine Rolle. Im Verhältnis zum MSE fallen Ausreißer dabei nicht so stark ins Gewicht (vgl. Formel 5).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Formel 5: MAE; Mean absolute error

Bei der Verwendung des MAE kann es zu Problemen kommen, sobald die Differenz zwischen der Prognose und dem Label minimal ist. Der MSE umgeht dieses Problem durch das Quadrieren der Summe. Allerdings können bei der Verwendung des MSE Probleme auftreten, sobald die Differenz zwischen Prognose und Label zu groß ist, Ausreißer fallen damit stark ins Gewicht (Chai und Draxler 2014).

Die Huber-Funktion ist eine Verlustfunktion, die die beiden Ansätze des MAE und MSE kombiniert und damit versucht, die auftretenden Probleme der Funktion zu lösen. Damit dies möglich ist, verwendet die Huber-Funktion für kleine Differenzen die Methodik des MSE und bei größeren Unterschieden wird auf den MAE zurückgegriffen (vgl. Formel 6). Der Schwellenwert, ab welchem eine jeweilige Methode eingesetzt wird, kann dabei in Abhängigkeit des Datensatzes definiert werden (Cavazza und Murino 2016).

$$\text{Huber} = \begin{cases} \frac{1}{2} \cdot (y_i - \hat{y}_i)^2 & \text{for } |y_i - \hat{y}_i| \leq \delta \\ \delta \cdot (|y_i - \hat{y}_i| - \frac{1}{2} \cdot \delta^2) & \text{otherwise} \end{cases}$$

Formel 6: Huber-Funktion

Im Trainingsprozess gilt es die Verlustfunktion, durch die Veränderung von Modellparametern zu minimieren. Das Training kann beendet werden, sobald die Funktion ausreichend minimiert worden ist oder im weiteren Trainingsverlauf keine signifikanten Änderungen mehr zu erwarten sind. Ein zu kleiner Prognosefehler kann allerdings auch ein Indiz für ein mögliches Overfitting des Modells sein (vgl. Kapitel 3.4).

#### 4.2.3 Backpropagation

Das eigentliche Konvergieren der Prognose wird durch den Backpropagation-Algorithmus und das Anpassen von Verbindungsgewichten ermöglicht. Der Backpropagation-Algorithmus wurde 1986 bekannt und wird auch heute noch für das Training von neuronalen Netzen verwendet.

Bei der Backpropagation handelt es sich um eine effiziente Technik für das Bestimmen des Gradienten und das Ausführen eines Schrittes im Gradientenverfahren.

Die Vorstellung des Algorithmus wird in zwei Teile unterteilt: Zum einen wird die Methodik zur Bestimmung des Gradienten mithilfe der Reverse-Mode-Autodiff-Methode beschrieben und zum anderen die Optimierung mithilfe des Gradientenverfahrens, welches in Kapitel 4.2.4 weiter spezifiziert wird.

Bei der Anwendung des Algorithmus wird innerhalb von zwei Durchgängen – einem Vorwärts- und einem Rückwärtsdurchlauf – durch die Architektur des neuronalen Netzes der Gradient des Fehlers für jeden Modellparameter im Netz bestimmt. Damit kann mithilfe der Backpropagation ermittelt werden, wie jedes Verbindungsgewicht zum Prognosefehler beigetragen hat.

Damit ein effizientes Training mithilfe des Backpropagation-Algorithmus möglich ist, wird der Datensatz in Batche unterteilt, wobei ein Batch eine festgelegte Anzahl von Datenpunkten beinhaltet. Alle Datenpunkte, welche sich im Batch befinden, werden gleichzeitig verarbeitet.

Im Trainingsprozess wird der Gesamtdatensatz mehrmals durchlaufen, wobei jeder Durchlauf des Datensatzes als Epoche bezeichnet wird. Eine Epoche beinhaltet damit den gesamten Datensatz, welcher in mehrere Batches unterteilt ist.

Um nun anhand der einzelnen Batches trainieren zu können, durchläuft jede Instanz des Batches das neuronale Netz mittels der Forward Propagation (vgl. Kapitel 4.1.2). Dabei werden neben der schlussendlichen Prognose auch alle Zwischenergebnisse der einzelnen Neuronen gespeichert. Im nächsten Schritt wird der Prognosefehler jeder Ausgabe mittels der Verlustfunktion bestimmt. Nachdem der Prognosefehler bestimmt worden ist, wird mit der Backpropagation fortgefahren. Dafür werden die Schritte der Forward Propagation rückwärts durchlaufen.

Im ersten Schritt wird die Ableitung der Verlustfunktion in Bezug auf die Gewichte und den Bias-Theta jedes Modellparameters bestimmt. Die Ableitung repräsentiert damit die Steigung der Funktion und wird unter Anwendung der Kettenregel gebildet. Mithilfe dessen kann festgestellt werden, inwiefern jede Ausgabeverbindung zum Prognosefehler beigetragen hat. Damit diese Information für jeden Modellparameter vorliegt, wird die Architektur des neuronalen Netzes rückwärts durchlaufen bis hin zum Input-Layer. Sobald die Backpropagation erfolgreich durchgeführt worden ist, liegt damit der Fehlerbeitrag jedes Modellparameters in Bezug auf die Verbindungsgewichte vor und es kann mit der Aktualisierung der Gewichte fortgefahren werden (Rumelhart, Hinton und Williams 1986).

#### 4.2.4 Gradientenabstieg

Die Parameter werden mithilfe einer Optimierungsfunktion und des Durchführens eines Schrittes im Gradientenverfahren angepasst, mit dem Ziel, die Verlustfunktion zu minimieren.

Das Anpassen der Parameter wird im Folgenden am Beispiel des stochastischen Gradientenabstiegsverfahrens erläutert. Das stochastische Gradientenabstiegsverfahren ist eine Methodik aus der numerischen Mathematik. Der Grundgedanke dabei ist, Parameter iterativ so zu verändern, dass der ermittelte Fehler der Kostenfunktion minimiert wird und dementsprechend ein Minimum der Funktion gefunden werden kann. Weiterhin ist zu erwähnen, dass das Verfahren lediglich irgendein Minimum bestimmt, wobei jedoch keine Aussage darüber getroffen werden kann, ob es sich um ein lokales oder ein globales Minimum handelt (Ruder 2017, S. 2).

Die Verbindungs gewichte jedes Neurons werden auf der Basis der Formel 7 aktualisiert. Dabei wird die Ableitung der Verlustfunktion in Bezug auf das jeweilige Verbindungs gewicht gebildet, anschließend wird die Ableitung vom Verbindungs gewicht abgezogen. Außerdem ist zu erkennen, dass die Ableitung mit einem weiteren Parameter multipliziert wird. Dabei wird von der Lernrate des Netzes gesprochen; die Lernrate bestimmt, wie groß die Veränderung der Gewichte in Abhängigkeit von der errechneten Ableitung pro Durchlauf ist. Der berechnete Term nach der Formel 7 wird abschließend als neues Verbindungs gewicht verwendet. Der Vorgang wird dabei wiederholt, bis alle Verbindungs gewichte im Netz aktualisiert worden sind.

Darauffolgend kann mit der nächsten Trainingsinstanz fortgefahren werden. Mit jeder Iterationsstufe durch das neuronale Netz sollte der festgestellte Prognosefehler kleiner werden. Das Training wird beendet, sobald die Verlustfunktion ausreichend minimiert worden ist (Goodfellow, Bengio und Courville 2016, S. 200).

$$w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w_t}$$

Formel 7: Gradientenabstieg, Aktualisierungsregel

In Abbildung 4 ist das Konvergieren der Verlustfunktion hin zu einem Minimum in einem zweidimensionalen Koordinatensystem dargestellt. Dabei sind auf der Abszisse die Verbindungs gewichte zu erkennen und auf der Ordinate der errechnete Prognosefehler.

Die Geschwindigkeit, mit welcher die Verlustfunktion zu einem Minimum konvergiert, wird maßgeblich durch die definierte Lernrate beeinflusst. Eine zu niedrige Lernrate kann dazu führen, dass das Konvergieren der Lösung hin zu einem Minimum viel Zeit beansprucht und sehr ineffizient verläuft, was im linken Koordinatensystem in Abbildung 4 zu erkennen ist, wohingegen eine zu hoch eingestellte Lernrate das Konvergieren der Lösung generell verhindern kann, was im rechten Koordinatensystem in Abbildung 4 erkennbar ist (Smith 2018, S. 6).

Das stochastische Gradientenabstiegsverfahren stellt die herkömmliche Herangehensweise für die Veränderung der Parameter in neuronalen Netzen dar. Im Laufe der Zeit wurden verschiedene Variationen der Gleichung entwickelt, welche den Gradientenabstieg effizienter gestalten sollen (Ruder 2017, S. 3). Eine der Optimierungen des Algorithmus, welche häufig im Trainingsprozess von neuronalen Netzen eingesetzt wird, ist der Adam-Optimierungsalgorithmus. Der Algorithmus betrachtet bei der Optimierung zusätzlich den Durchschnitt der vorangegangenen Gradienten und auch den Durchschnitt der quadrierten Gradienten. Die Optimierung mittels Adam hat sich in vergangenen Arbeiten als äußerst effizient herausgestellt und wird auch bei der weiterführenden Umsetzung der Fragestellung verwendet (Kingma und Ba 2017).

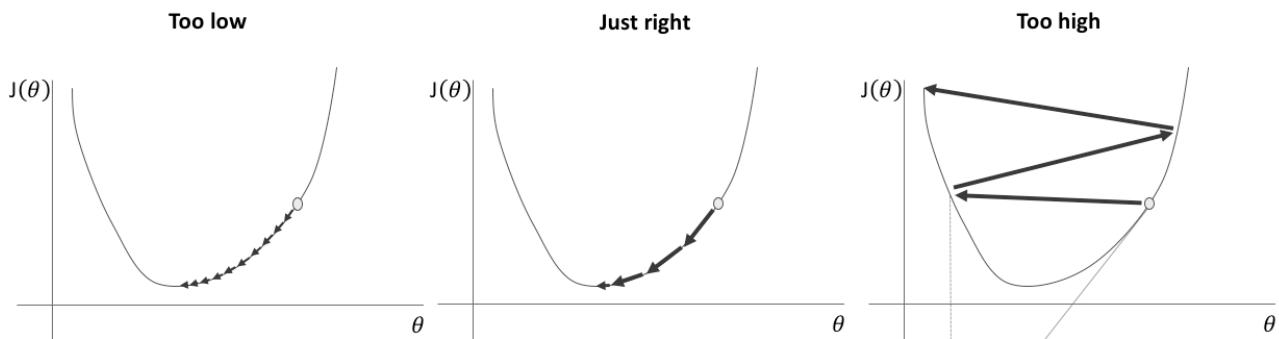


Abbildung 4: Gradientenabstieg mit unterschiedlichen Lernraten (Jordan 2018)

#### 4.2.5 Aktivierungsfunktion

Die Aktivierungsfunktion, welche bereits in Kapitel 4.1.1 erwähnt worden ist, bestimmt die Ausgabe jedes Neurons einer Schicht. Damit stellt die Aktivierungsfunktion den Zusammenhang zwischen dem Input und dem Aktivitätslevel eines Neurons dar. Das Neuron gibt damit nur einen Wert an die nachgelagerte Schicht aus, sobald ein Schwellenwert überschritten worden ist. Im Folgenden werden die Sigmoid- und die ReLU-Aktivierungsfunktion weiter spezifiziert. Die Funktionen können mithilfe eines zweidimensionalen Koordinatensystems beschrieben werden, bei welchem auf der Abszisse der Netzinput und auf der Ordinate das entsprechende Aktivitätslevel abgetragen wird (vgl. Abbildung 5).

Die Sigmoidfunktion liegt in einem Wertebereich von 0 bis 1, womit sie vor allem bei binären Klassifikationsaufgaben verwendet werden kann. Ein Problem der Sigmoidfunktion ist jedoch die kurze Reichweite der Funktion. Damit werden größere negative oder positive Werte stark komprimiert und die Kurve verläuft beinahe parallel zur Abszisse, wodurch – wenn überhaupt – nur eine kleine Steigung während der Backpropagation festgestellt werden kann. Besonders bei vielschichtigen neuronalen Netzen kann dies zu Problemen führen (Ramachandran, Zoph und Le 2017, S. 1).

Die ReLU-Funktion umgeht die Problematik der Sigmoidfunktion und gibt den Wert null zurück, wenn die Eingabe kleiner als null ist, und den unveränderten Eingabewert, sobald der Wert größer null ist. Damit gehen für die Prognose wichtige Informationen, auch bei großen Inputs in vielschichtigen neuronalen Netzen nicht verloren. Lange Zeit wurde in der Forschung die Sigmoidfunktion anstelle der ReLU-Funktion präferiert, da die ReLU-Funktion am Punkt null nicht differenzierbar ist. In der Praxis hat sich jedoch gezeigt, dass dies nicht zwangsläufig zu Problemen führt (Goodfellow, Bengio und Courville 2016, S. 192). Weiterhin konnte festgestellt werden, dass sich die ReLU-Funktion in vielen Anwendungsfällen als am effizientesten erwies. Damit ist die ReLU-Funktion auch die aktuell meistverwendete Aktivierungsfunktion in neuronalen Netzwerken (Ramachandran, Zoph und Le 2017, S. 1).

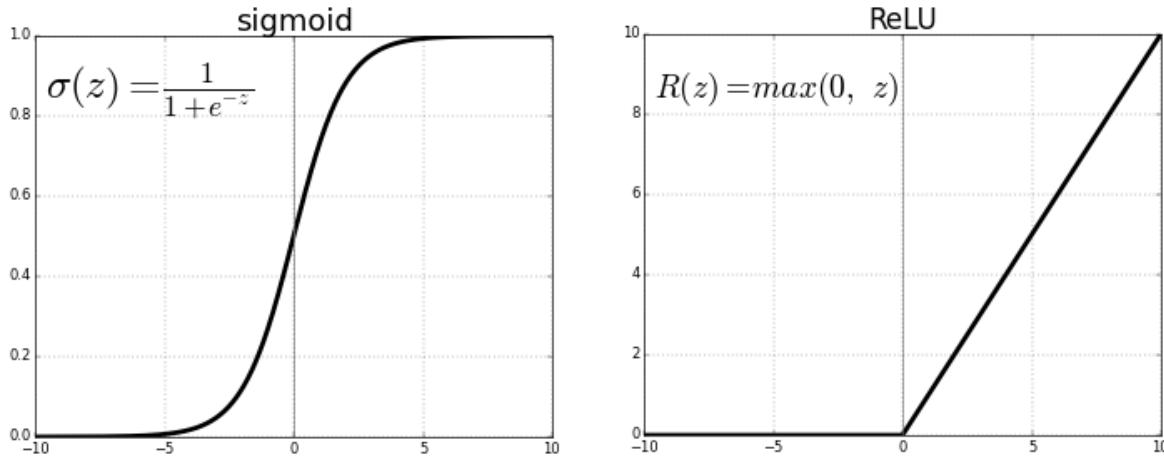


Abbildung 5 Aktivierungsfunktionen Sigmoid und ReLU (Sharma 2017)

#### 4.2.6 Naive Prognose

Nachdem der Trainingsprozess des Modells abgeschlossen ist, kann der Prognosefehler der Vorhersage über den Evaluierungsdatensatz anhand von verschiedenen Kostenfunktionen wie dem MAE oder MSE berechnet werden. Vermehrt reicht jedoch die Betrachtung des Prognosefehlers nicht aus, um eine Aussage über die Verwendbarkeit des Modells treffen zu können.

Für die weitere Evaluierung des Modells können die Prognosen ins Verhältnis zu herkömmlichen Vorhersagemodellen gesetzt werden. Häufig wird dafür die Naive Prognose verwendet. Bei der Naiven Prognose wird angenommen, dass der kommende Zeitschritt dem zuletzt erfassten Zeitschritt entspricht. Damit gilt  $t + 1 = t$ , woraus sich eine Prognose ergibt, welche die Zeitreihe um einen Zeitschritt verschoben darstellt (vgl. Abbildung 6). Die Naive Prognose ist damit neben einer zufälligen Schätzung das einfachste Vorhersagemodell.

	M	A	M	J	J	A	S	O	N	D	J
Realisierter Auftragseingang	23	25	28	32	29	26	28	30	33	36	32
Naiver Forecast	23	25	28	32	29	26	28	30	33	36	36

Abbildung 6: Beispiel Naive Prognose (Kühnapsel 2015, S.167)

Sollte das Modell den Prognosefehler der Naiven Vorhersage nicht minimieren können, ist anzunehmen, dass die Zeitreihe nicht vorhersagbar ist oder die Datengrundlage nicht ausreichend Information enthält. (Hyndman und Athanasopoulos 2021, S. 48).

Falls der zu untersuchende Datensatz saisonale Schwankungen in regelmäßigen Abständen aufweist, kann die Notwendigkeit bestehen, das Naive Modell in Abhängigkeit der auftretenden Saisonalitäten anzupassen. Bei diesem Vorgehen wird auch von einer saisonalen Naiven Prognose gesprochen (Hyndman und Athanasopoulos 2021, S. 48).

## 4.3 Architekturen

### 4.3.1 CNN

Convolutional-Neural-Networks (CNN) stellen eine Sonderform der MLP-Architektur dar. Die Architektur ist darauf optimiert, Daten zu verarbeiten, welche eine gitterartige Struktur aufweisen. Damit wird sie häufig für die Analyse von Bildern verwendet, da sich jedes Bild hinsichtlich der Pixel in ein zweidimensionales Gitter aufspalten lässt. Die Architektur kann auch bei der Analyse von Zeitreihen verwendet werden; die Zeitreihe stellt dabei eindimensionales Gitter dar, welches Daten in regelmäßigen Abständen enthält. Der Unterschied zur herkömmlichen MLP-Architektur besteht darin, dass das CNN anstelle einer Matrixmultiplikation eine diskrete Convolution für die Berechnung der Inputs verwendet. Die Funktionsweise beruht damit auf der Verwendung von Filtern. Mit der Filterung der Daten können Merkmale aus den Trainingsdatenpunkten extrahiert werden. Bei der Betrachtung einer Zeitreihe können mithilfe der CNN-Architektur Merkmale wie Muster, Saisonalitäten oder Trends innerhalb der Zeitreihe erkannt werden. Mit der steigenden Tiefe des neuronalen Netzes steigen sowohl die Komplexität als auch die Anzahl identifizierter Merkmale (LeCun u. a. 1998).

Grundsätzlich benötigt ein klassisches CNN-Modell folgende drei Layer:

- Convolutional Layer
- Pooling Layer
- Fully Connected Layer

### Convolutional Layer

Der Convolutional Layer ist der wichtigste Bestandteil für die Funktionsweise der CNN-Architektur. Im Convolutional Layer findet die Filterung der Daten statt. Dafür wird jedes Neuron des Convolutional Layers nur mit einem Teil der Input-Daten verknüpft, sodass verschiedene Filter entstehen, welche auch Feature-Maps genannt werden. Jeder Filter betrachtet dabei nur einen Teil der Eingabesequenz und identifiziert verschiedene Merkmale im betrachteten Bereich. Durch die Verwendung mehrerer Convolutional Layer können die herausgefilterten Merkmale der vorgelagerten Schichten miteinander in Verbindung gebracht werden (vgl. Abbildung 7). Die Dimension des Betrachtungsreiches jedes Neurons, wird auch Wahrnehmungsfeld genannt und wird im Trainingsprozess definiert. (Géron 2019, S. 452).

### Pooling Layer

Für den alleinigen Einsatz der Convolutional Layer wäre es notwendig, über ein hohes Maß an Rechenleistung zu verfügen; das Training und die Optimierung des Netzes wären damit nur eingeschränkt möglich. Um dem entgegenzuwirken, werden Pooling Layer in Verbindung mit den Convolutional Layern verwendet. Damit befindet sich innerhalb des neuronalen Netzes nach einem Convolutional Layer ein Pooling Layer. Der Pooling Layer ist wie der bereits vorgestellte Convolutional Layer aufgebaut. Damit sind die Neuronen der Pooling Layer auch nur mit einer begrenzten Anzahl an Neuronen der vorgelagerten Schicht verbunden. Der Layer verfügt über keine Gewichte, sondern

aggregiert lediglich die Eingabedaten über eine Aggregatfunktion, wodurch die Dimension der zu verarbeitenden Datenpunkte deutlich abnimmt und ein positiver Effekt auf die Trainingsdauer zu verzeichnen ist. Damit verhält sich der Pooling Layer wie ein zustandsloses gleitendes Fenster. Außerdem verhindert der Layer ein frühzeitiges Overfitting, da schwach ausgeprägte Merkmale bei dem Aggregieren der Daten verworfen werden können (Géron 2019, S. 460).

### Fully Connected Layer

Für die Ausgabe des CNN wird der bereits vorgestellte Fully Connected Layer verwendet. Die Anzahl von Neuronen, welche sich im Output-Layer befinden, ist entsprechend vom Problem und der gewünschten Ausgabe abhängig.

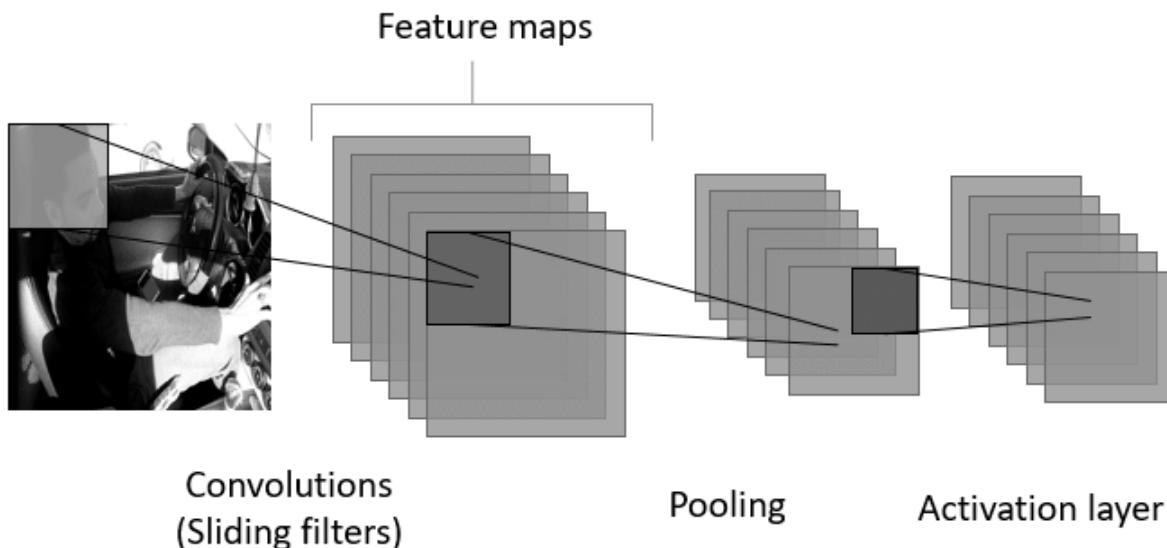


Abbildung 7 Convolutional Neural Network (Mafeni Mase u. a. 2020)

### 4.3.2 RNN

Die bisher vorgestellten Architekturen verfolgen einen Feedforward-Ansatz. Bei einer Feedforward-Architektur verläuft die Verarbeitungsrichtung der Datenpunkte innerhalb des neuronalen Netzes chronologisch von der Eingabe hin zur Ausgabe. Im Folgenden werden Recurrent Neural Networks (RNN) vorgestellt, welche sich dadurch auszeichnen, dass die Verarbeitungsrichtung der Daten vom Feedforward-Ansatz abweicht. Durch das Abweichen der Verarbeitungsrichtung sind Neuronen im RNN in der Lage, mittels Rückkopplung den eigenen Zustand oder den Zustand von Neuronen in der gleichen Schicht zu beeinflussen.

Im Feedforward-Netz werden alle Eingaben, welche zu einer Trainingssequenz gehören, im gleichen Zeitschritt verarbeitet. Das Problem, welches dabei entsteht, ist, dass bei der Untersuchung von sequenziellen Daten die Informationen über die zeitlichen Unterschiede der erfassten Datenpunkte verloren gehen. Im RNN wird eine Eingabe für jeden Zeitschritt verwendet, wodurch Informationen über die zeitlichen Unterschiede innerhalb der Eingabesequenz während der Datentransformation erhalten bleiben. Das ermöglicht es, den Erfassungszeitpunkt in der Prognose zu berücksichtigen. Weshalb RNNs besonders bei der Analyse von sequenziellen Daten zum Einsatz kommen.

Die Arbeitsweise eines einzelnen Neurons innerhalb einer RNN-Architektur ist in Abbildung 8 dargestellt. Dabei entspricht die linke Abbildung dem eigentlichen Neuron, wobei zu erkennen ist, dass

neben dem normalen Output eine weitere Ausgabe an das Neuron selbst zurückgeleitet wird. In der rechten Abbildung ist die Funktionsweise des Neurons entlang der Zeitachse dargestellt. Dabei enthält das Neuron bei jeder Eingabe eines Zeitschrittes auch den Ausgabevektor des vorangegangenen Zeitschrittes, wodurch die zeitlichen Unterschiede der Eingabedaten in die Analyse einfließen können. (Hopfield 1982)

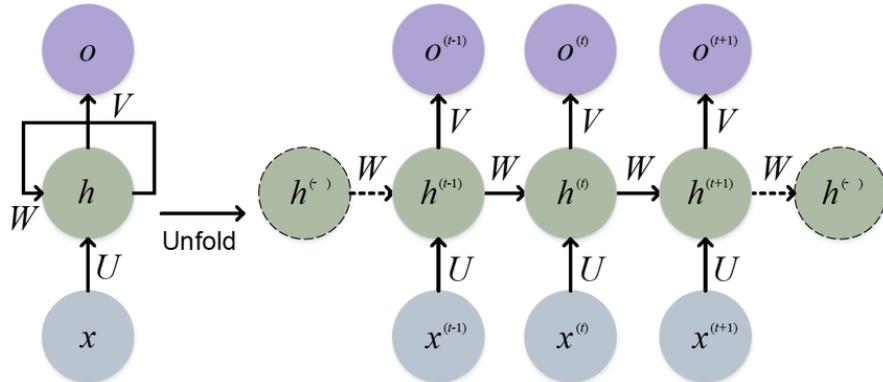


Abbildung 8: Recurrent Neural Network (Feng u. a. 2017)

Bei der Transformation der Daten innerhalb des RNN gehen bei jedem Zeitschritt Informationen verloren. Das kann zur Folge haben, dass die Daten innerhalb des RNN keinerlei Informationen mehr über die eigentlichen Inputs enthalten, was die Ergebnisse signifikant beeinflussen kann. Weiterhin können bei der Untersuchung von langen Sequenzen innerhalb eines RNNs, die Problematik der schwindenden und explodierenden Gradienten auftreten. Schwindende Gradienten treten auf, wenn die berechnete Ableitung innerhalb der Backpropagation kleiner 1 ist, wodurch im weiteren Verlauf die Gradienten, einen immer kleiner werdenden Wert annehmen und ein effizientes Training verhindert wird. Im Gegensatz dazu steht die Problematik des explodierenden Gradienten welches auftritt, sobald die Ableitungen hohe Werte annehmen. Im Laufe der Propagation steigt der Gradient damit weiter an und das Netz wird instabil, ein effizientes Training ist damit ebenfalls nicht mehr möglich.

Eine Möglichkeit die Auftrittswahrscheinlichkeit der genannten Problematiken zu reduzieren, ist die Verwendung von LSTM-Zellen. Die LSTM-Zelle verfügt über ein Langzeitgedächtnis, mit welchen die Möglichkeit besteht wichtige Merkmale über einen längeren Zeitraum zu speichern und andere Merkmale wieder zu vergessen. Die Ausgabe einer LSTM-Zelle ist dabei an einem Punkt innerhalb der Zeit von folgenden drei Faktoren abhängig:

- Dem Output des vorangegangenen Zeitschrittes
- Der Eingabe des aktuellen Zeitschrittes
- Dem aktuellen Zellenzustand

Für die Realisierung der genannten Eigenschaften besitzt jede Zelle ein Input-Gate, ein Output-Gate und ein Forget-Gate. Die Gates steuern dabei, wie die einfließenden Informationen im Netzwerk gespeichert werden und das Netz wieder verlassen.

Der Zustand der LSTM-Unit ist dabei in zwei Vektoren aufgeteilt: einen Zustand, welcher das Kurzzeitgedächtnis beschreibt, und einen Zustand, welcher das Langzeitgedächtnis beschreibt. Der Grundgedanke dabei ist, dass die LSTM-Unit in der Lage ist, relevante Eingaben zu erkennen und diese im Langzeitgedächtnis zu speichern, wofür das Input-Gate verantwortlich ist. Dort sollen wichtige Merkmale extrahiert und die Eingaben so lange wie notwendig gespeichert werden.

Sobald eine Information nicht mehr benötigt wird, kann sie vergessen werden, was die Aufgabe des Forget-Gates ist. Anschließend werden die benötigten Merkmale für die Ausgabe extrahiert und die Informationen weitergegeben, wofür letztendlich das Output-Gate zuständig ist (Hochreiter und Schmidhuber 1997).

#### 4.3.3 Encoder-Decoder

Encoder-Decoder gehören zu der in Kapitel 4.3.2 vorgestellten Gruppe der RNNs und werden vermehrt für das Lösen von Sequenz-zu-Sequenz-Problemen verwendet. Bei den bisher vorgestellten Netzen wurde mithilfe einer Eingabesequenz ein Ausgabevektor erzeugt, wobei jeder Wert des Vektors einen zukünftigen Zeitschritt darstellt. Der Unterschied zu einer sequenziellen Ausgabe besteht darin, dass bei einer vektoriellen Ausgabe alle zukünftigen Zeitschritte gleichzeitig prognostiziert werden. Bei der Verwendung einer Encoder-Decoder-Architektur wird jeder Zeitschritt einzeln prognostiziert, woraus schlussendlich eine Sequenz von Werten entsteht. Das hat den Vorteil, dass der zu prognostizierende Zeitschritt von der vorherigen Prognose beeinflusst werden kann. Das Encoder-Decoder-Modell kann dabei als zweistufiges Netz beschrieben werden, wobei der Encoder die erste Stufe darstellt und der Decoder die zweite.

Die Eingabesequenz wird dabei an den Encoder weitergegeben und mithilfe einer der bereits vorgestellten Architekturen analysiert. Der Ausgabevektor des Encoders beschreibt den internen Zustand der Eingabesequenz. Dieser Vorgang wird mehrmals wiederholt, in Abhängigkeit von der Länge der gewünschten Ausgabesequenz. Die erstellte Vektorsequenz wird anschließend an den Decoder weitergegeben, wobei jeder Vektor auch als Kontextvektor beschrieben wird. Der Decoder analysiert den Kontextvektor und trifft anhand dessen eine Prognose, wobei die Prognose für jeden Zeitschritt separat erfolgt. Damit ist es mittels des Decoders möglich, für jeden Zeitschritt in der Ausgabesequenz den erforderlichen Kontext aus dem Kontextvektor herauszufiltern, wodurch mittels des Decoder-Encoder-Modells eine Prognose zum einen beruhend auf dem internen Zustand der Eingabesequenz und zum anderen basierend auf der Prognose des vorangegangenen Zeitschrittes getroffen werden kann (Sutskever, Vinyals und Le 2014).

## 5. Methodik

Die Vorgehensweise zur Bearbeitung der Problemstellung und Anwendung der Theorie orientiert sich am Cross Industry Standard Process for Data Mining (CRISP-DM). Der CRISP-DM stellt ein Vorgehensmodell zur Bearbeitung von Projekten aus der Domäne des Data-Minings dar. Das Vorgehensmodell wurde 1996 konzipiert und wurde zum Projekt der Europäischen Union erklärt. Veröffentlicht wurde es 2000 für das damals noch neue und unerforschte Feld des Data-Minings. Data-Mining beschreibt das Extrahieren von Informationen aus großen Datensätzen mittels systematisch angewandter statistischer Methoden. Ein Anstoß zur Entwicklung des CRISP-DM war die Beobachtung, dass vermehrt datenorientierte Projekte scheiterten, da es nicht gelang, das wirtschaftliche Problem in eine datenorientierte Aufgabe zu überführen. Unter diesem Hintergrund soll das Modell einen industrieunabhängigen Leitfaden bieten, für die Durchführung von datenorientierten Projekten (Chapman u. a. 2000).

Trotz der zurückliegenden Veröffentlichung des Modells wird es weiterhin häufig bei der Bearbeitung von Daten-Projekten verwendet (Mariscal, Marbán und Fernández 2010, S. 139).

Im CRISP-DM werden sechs Phasen definiert, die bei der Bearbeitung der Fragestellung zu durchlaufen sind. In Abhängigkeit der Ergebnisse der Phase kann mit der darauffolgenden fortgefahren werden oder die Notwendigkeit bestehen zu einer vorangegangenen Phase zurückzuspringen (vgl. Abbildung 9).

- - Das Geschäftsverständnis
- - Das Datenverständnis
- - Die Datenvorbereitung
- - Die Modellierung
- - Die Evaluierung
- - Die Bereitstellung

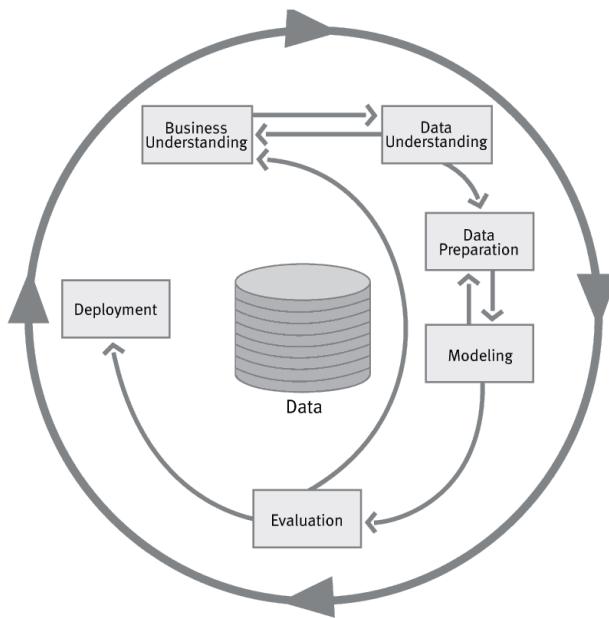


Abbildung 9: CRISP-DM (Chapman u. a. 2000)

Weiterhin ist zu erwähnen, dass sich für die Bearbeitung der Problemstellung lediglich am CRISP-DM orientiert wird und sofern notwendig vom vorgeschriebenen Vorgehen abweicht.

Dies ist damit zu begründen, dass das Modell an verschiedenen Stellen Lücken aufweist und nicht detailliert genug für die Umsetzung des Deep Learning Modells ist, was sich vor allem auf die voranschreitende Entwicklung und den Entwurf des Modells für die Data-Mining-Domäne zurückführen lässt. Zum Beispiel spielt Evaluierung der Ergebnisse im Vorgehensmodell nur bedingt eine Rolle und auf ein weiterführendes Monitoring nach Inbetriebnahme des Modells wird nicht eingegangen. Daraus ergibt sich die Schlussfolgerung, dass für die Umsetzung eines Projektes anhand des CRISP-DM die Notwendigkeit besteht, einzelne Prozessschritte hinzuzufügen oder detaillierter auszuführen. Zur Ergänzung des Vorgehensmodells wird sich damit am Data Science Process Model (DASC-PM) orientiert. Das DASC-PM ist ein Vorgehensmodell für Data-Science-Projekte, welches 2020 veröffentlicht wurde (Schulz u. a. 2020). Durch die Aktualität des Modells und der Entwicklung für den Data-Science-Bereich können auftretende Schwächen des CRISP-DM ausgeglichen und ergänzt werden. Bei Bearbeitung der Fragestellung wird zu Beginn jedes Prozessschrittes das definierte Ziel im Vorgehensmodells dargestellt. Anschließend können die einzelnen Phasen durchlaufen und ein Fazit gezogen werden. In Abhängigkeit der erreichten Ergebnisse kann anschließend mit der nächsten Phase fortgefahren oder zu einer vorangegangenen Phase zurückgesprungen werden.

## 6. Artefakt

Im ersten Schritt des Vorgehensmodells wird das notwendige Geschäftsverständnis aufgebaut. Zu der Phase gehört neben dem Erfassen der notwendigen Daten das Aufbauen eines weitreichenden Verständnisses für die Domäne, in welcher die Aufgabenstellung bearbeitet wird (Chapman u. a. 2000, S. 13). Die Fragestellung des Projektes unter der Betrachtung der Domäne wurden bereits in Kapitel 1.2 weitreichend erläutert, außerdem wurden die benötigten Daten bereits im Vorfeld generiert und gesammelt und stehen im weiteren Verlauf zur Analyse zur Verfügung, weshalb auf eine weitere Darstellung der Aufgabenstellung und der Domäne verzichtet wird.

Weiterhin ist vor Bearbeitung zu erwähnen, dass in der Aufgabenstellung beschrieben worden ist, dass eine eigenständige Vorhersage für jeden Lagerbereich getroffen werden muss. Unter der Betrachtung, dass die Datenlage pro Lagerbereich gleich ist und sich damit die Ergebnisse eines Lagerbereiches auf die anderen Bereiche transferieren lassen, beschränkt sich die folgende Darstellung auf die Analyse eines einzigen Lagerbereiches.

### 6.1 Datenverständnis

Im Prozessschritt des Datenverständnisses soll ein erster Überblick über die Datengrundlage geschaffen werden. Dafür werden alle für die Aufgabe verwendeten Datensätze vorgestellt und deren Eigenarten und Qualität geprüft. Außerdem soll mithilfe einer explorativen Datenanalyse ein erster Einblick in die Daten generiert werden. Bei der explorativen Datenanalyse werden zum einen die vorgestellten Datensätze visualisiert und zum anderen bekannte Kennzahlen aus der deskriptiven Statistik berechnet. (Chapman u. a. 2000, S. 17).

#### 6.1.1 Datenübersicht

Die verwendeten Daten werden von einer relationalen Datenbank abgerufen. Die Verteilung der Informationen innerhalb der verschiedenen Tabellen ist in Anhang 5 zu finden. Im Folgenden werden die Informationen, welche aus den einzelnen Tabellen in die Analyse einfließen, dargestellt und der Zusammenhang zwischen den Datenreihen und dem Prognosezeitraum erläutert.

#### Auftragsdaten

Die Auftragsdaten, welche es zu prognostizieren gilt, werden seit 2015 gesammelt und gespeichert. Um vorherzusagen Komponenten wie Trends und Saisonalitäten (vgl. Kapitel 2.1) in den Zeitreihen und im Bestellverhalten der Filialen zu entdecken, werden unter anderem die historischen Daten im Trainingsprozess verwendet.

#### Lagerauslastung

Der Datensatz beinhaltet sowohl in absoluten Zahlen als auch in relativen Zahlen die Lagerauslastung des zu analysierenden Lagerbereiches. Die Daten wird in der Analyse unter der Annahme verwendet, dass eine höhere Lagerauslastung auf bevorstehende Saisonalitäten, Feiertage oder andere Besonderheiten hinweisen kann, was sich wiederum auf die zukünftigen Auftragsdaten auswirken würde.

#### Eingelagerte Transporteinheiten

Die Datenreihe beinhaltet Informationen über die Anzahl eingelagerter Transporteinheiten (TE), welche im betrachteten Lagerbereich kommissioniert werden. Eine TE beschreibt dabei eine Europalette nach der europäischen Norm UNE-EN 1368-1. Die Datenreihe wird dabei mit der gleichen Begründung wie bei den zuvor vorgestellten Datensatz herangezogen.

## Zu kommissionierende Filialen

Die Anzahl an Filialen, welche an den zu prognostizierenden Tagen im Lager zu kommissionieren sind, ist von den geplanten Auslieferungen im Tourenplan abhängig. Damit kann eventuell mittels der Information bereits vorab, ein Richtwert über die Höhe der Auftragsdaten prognostiziert werden. Es zeigte sich außerdem im Verlauf der geführten Experteninterviews, dass die befragten Personen auch ein Zusammenhang zwischen den zu Kommissionierenden Filialen und den Auftragsdaten annehmen (vgl. Anhang 3).

Inwiefern sich durch die vorgestellten Informationen in der Analyse einen Mehrwert generieren lässt, ist fraglich und wird sich zum einen in der explorativen Datenanalyse und zum anderen in der Analysephase zeigen. Sollte sich durch die Betrachtung der verschiedenen Informationen kein Mehrwert für die Prognose generieren lassen, kann die Notwendigkeit bestehen, die selektierte Datengrundlage anzupassen.

### **6.1.2 Qualitätsprüfung**

Bei einer ersten Betrachtung des Rohdatensatzes wird häufig von der Qualität der Daten gesprochen. Die Qualität eines Datensatzes ist dabei kein klar definierter Begriff und kann nicht anhand eines Maßstabes gemessen werden. Meist wird von qualitativ hochwertigen Daten gesprochen, sobald die Daten vollständig und aktuell vorliegen sowie über einen großen Bezug zum Projektziel verfügen (Gitzel 2016). Eine weitere Möglichkeit ist, die Datenqualität anhand verschiedener Parameter zu prüfen. Eine beispielhafte Auflistung dafür infrage kommender Kriterien ist in Abbildung 10 dargestellt.

Aktualität	Allgemeingültigkeit	Alter	Änderungshäufigkeit	Aufbereitungsgrad
Bedeutung	Benutzbarkeit	Bestätigungsgrad	Bestimmtheit	Detailliertheit
Effizienz	Eindeutigkeit	Fehlerfreiheit	Flexibilität	Ganzheit
Geltungsdauer	Genauigkeit	Glaubwürdigkeit	Gültigkeit	Handhabbarkeit
Integrität	Informationsgrad	Klarheit	Kompaktheit	Kompression
Konsistenz	Konstanz	Korrektheit	Neutralität	Objektivität
Operationalität	Performance	Portabilität	Präzision	Problemadäquatheit
Prognosegehalt	Prüfbarkeit	Quantifizierbarkeit	Rechtzeitigkeit	Relevanz
Reliabilität	Richtigkeit	Robustheit	Seltenheit	Sicherheit
Signifikanz	Speicherbedarf	Standardisierungsgrad	Subjektdäquatheit	Testbarkeit
Umfang	Unabhängigkeit	Überprüfbarkeit	Übertragbarkeit	Validität
Verdichtungsgrad	Verfügbarkeit	Verfügungsmacht	Verknüpfbarkeit	Verlässlichkeit
Verschlüsselungsgrad	Verständlichkeit	Vertrauenswürdigkeit	Verwendungsbereitschaft	Vollständigkeit
Wahrheitsgehalt	Wahrscheinlichkeit	Wartungsfreundlichkeit	Wiederverwendbarkeit	Wirkungsdauer
Zeitadäquanz	Zeitbezug	Zeitoptimal	Zugänglichkeit	Zuverlässigkeit

Abbildung 10 Qualitätskriterien Datensätze (Helfert 2001, S.7)

Eine Qualitätsprüfung mittels der aufgelisteten Kriterien soll dabei keine vollumfängliche Analyse der Datenqualität darstellen, sondern lediglich die Prüfung von Eigenschaften ermöglichen, welche für die spezifische Aufgabenstellung notwendigerweise vorhanden sein müssen. Damit kann zwar kein Anspruch auf Vollständigkeit erhoben werden, allerdings kann dennoch mit der Prüfung der relevantesten Parameter ein unvorhersehbares Scheitern, bedingt durch die Qualität und Form des Rohdatensatzes, auf ein Minimum reduziert werden. Teilweise wird für die Prüfung verschiedener Eigenschaften ein tiefer Einblick in die Daten benötigt und erfolgt demnach erst mit der Durchführung der explorativen Datenanalyse (Helfert 2001, S. 43).

Eine Analyse bezüglich der Datenqualität wurde bereits im Vorfeld der Bachelorarbeit als Vorstudie durchgeführt und ist in Anhang 6 einsehbar. In der beschriebenen Studie konnte festgestellt werden, dass die Daten eine hohe Konsistenz aufweisen und redundanzfrei sind. Weiterhin wird von verständlichen Daten gesprochen, sobald alle Datenpunkte in einem geeigneten Datentyp vorliegen. Bei den vorliegenden Daten ist dies der Fall, abgesehen von den erfassten Zeitschritten. Diese sollten im Schritt der Datenaufbereitung in ein geeignetes Zeitstempel-Format transformiert werden. Außerdem konnte festgestellt werden, dass die betrachteten Daten ohne Plausibilitätsprüfung in die Datenbank geschrieben werden. Schlussfolgernd daraus besteht die Notwendigkeit, den Plausibilitätscheck im weiteren Verlauf nachzuholen, da fehlerhaft erfasste Messwerte sich negativ auf die Analyseergebnisse auswirken können.

### 6.1.3 Explorative Datenanalyse

In der explorativen Datenanalyse gilt es, Einblicke in die Zeitreihe der Auftragsdaten zu generieren und zu überprüfen, inwiefern sich bereits vorhersagbare Komponenten der Zeitreihe erkennen lassen. Außerdem kann geklärt werden, inwiefern Korrelationen zwischen den Auftragsdaten und den in Kapitel 6.1.1 vorgestellten weiteren Zeitreihen vorhanden sind. Mit diesem Hintergrund wird die explorative Datenanalyse in zwei Abschnitte unterteilt. Im ersten Abschnitt wird dabei nur die Zeitreihe der Auftragsdaten betrachtet und visualisiert. Anschließend wird im zweiten Teil, sofern vorhanden, ein Bezug zwischen den Datensätzen hergestellt.

#### 6.1.3.1 Darstellung Auftragsdaten

Die Zeitreihe der Auftragsdaten umfasst in etwa 2100 Datenpunkte. Mit der Betrachtung der Auftragsdaten des Jahres 2020 im Liniendiagramm können nur wenige bis gar keine vorhersagbaren Komponenten erkannt werden (vgl. Abbildung 11). Die gemessenen Datenpunkte sind vermeintlich zufällig verteilt, wodurch nur eine rauschende, nicht vorhersagbare Komponente erkennbar ist.

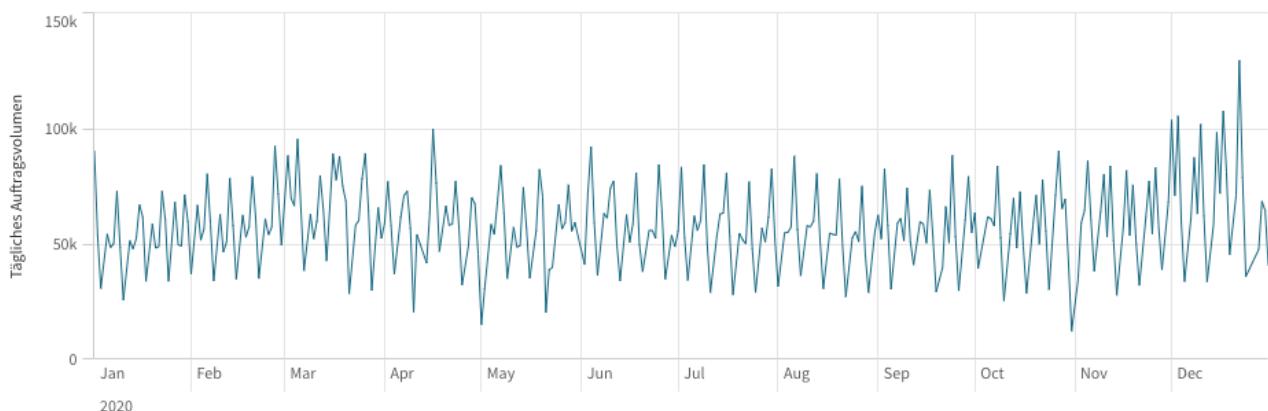


Abbildung 11 Auftragsdaten Liniendiagramm 2020 (Eigene Darstellung)

Das saisonale Verhalten der Zeitreihe wird erst mit der Visualisierung in verschiedenen Heatmaps deutlich. In Abbildung 12 ist dafür das durchschnittliche tägliche Auftragsvolumen mit der Unterteilung in Jahre und Monate dargestellt. Dabei ist zu erkennen, dass die durchschnittliche Höhe der Auftragsdaten vom Monat abhängt, in welchem diese erfasst worden sind.

Dabei steigt zum Ende jeden Jahres das durchschnittliche Auftragsvolumen an und fällt zum Jahresbeginn auf deutlich niedrigeren Wert. Dieses Verhalten ist dabei unabhängig vom betrachteten Jahr zu beobachten.

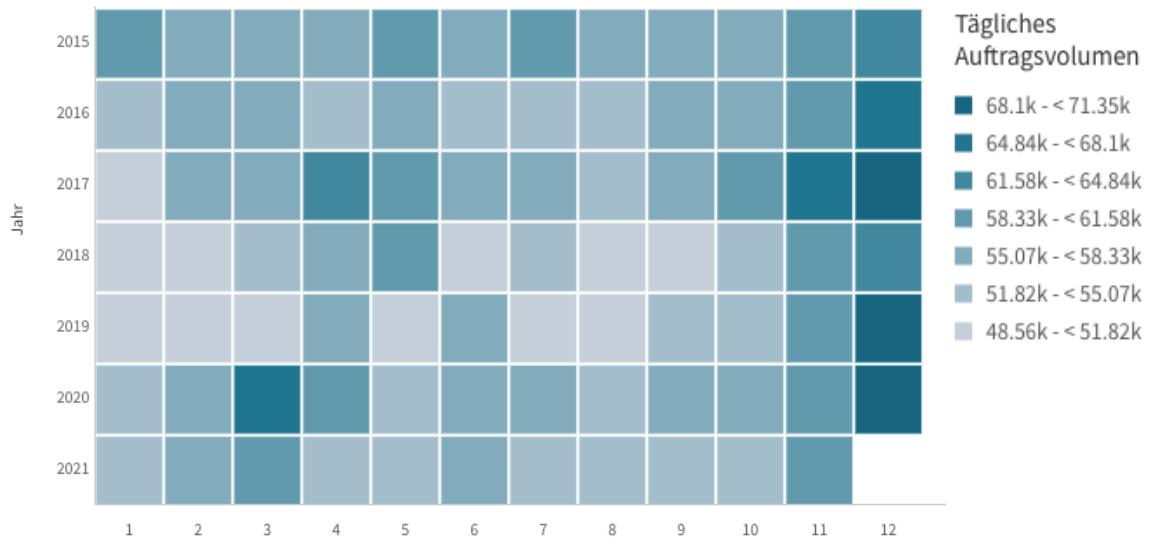


Abbildung 12 Auftragsdaten Heatmap, Monate und Jahre (Eigene Darstellung)

Mit einer weiteren Heatmap und der Unterteilung in Monate und Wochentage konnten weitere Korrelationen zwischen der Höhe der Auftragsdaten und den Wochentagen festgestellt werden (vgl. Abbildung 13). Dabei ist erkennbar, dass am Donnerstag und Montag im Verhältnis zu den restlichen Wochentagen ein erhöhtes Auftragsvolumen zu verzeichnen ist. Außerdem wird deutlich, dass an Samstagen ein geringes Auftragsvolumen zu erwarten ist. Die Zusammenhänge können unabhängig vom jeweiligen Monat festgestellt werden. Das Auftragsvolumen steigt zwar weiterhin zum Jahresende, jedoch bleibt das Verhältnis der Wochentage untereinander weitestgehend bestehen.

Ein weiteres Muster innerhalb der Zeitreihe konnte mit der Betrachtung der Woche im Monat, in welchem die Auftragsdaten erfasst worden sind, festgestellt werden. In Anhang 7 ist dies ebenfalls als Heatmap visualisiert. Das Verhalten lässt sich möglicherweise auf ein erhöhtes Kaufverhalten im Zeitraum der Gehaltsauszahlung zurückführen.

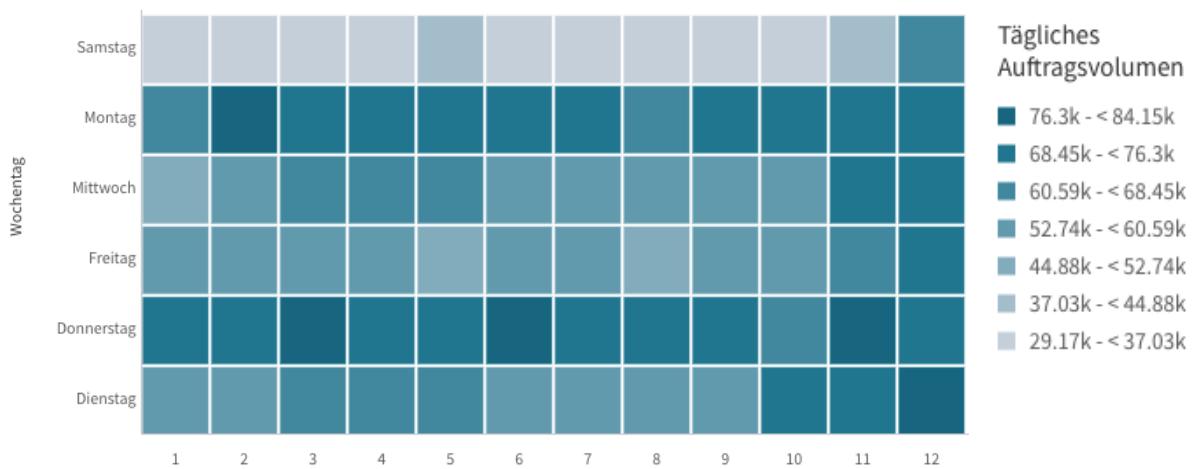


Abbildung 13 Auftragsdaten Heatmap, Monate und Wochentage (Eigene Darstellung)

Die verschiedenen Muster, welche innerhalb der Zeitreihe im Laufe eines Jahres auftreten, verdeutlichen sich mit dem Vergleich der gemessenen Auftragsdaten aus unterschiedlichen Jahren. Dafür sind in Abbildung 14 die erfassten Auftragsdaten aus dem Jahr 2020 (blau) und 2019 (violett) mithilfe eines Liniendiagramms dargestellt, wobei die Schrittweite der Ordinate eine Woche beträgt, wodurch

das Diagramm die Veränderungen des durchschnittlichen Auftragsvolumens, der Kalenderwochen, im Laufe eines Jahres darstellt.

Die Abbildung verdeutlicht, dass die Auftragsvolumina der unterschiedlichen Jahre bei der Betrachtung absoluter Werte stark voneinander abweichen. Allerdings können Ähnlichkeiten bei der relativen Veränderung der Zeitreihen beobachtet werden. Tendenziell sind die Veränderung und Ausreißer in der Zeitreihe des Jahres 2019 nicht so stark ausgeprägt wie in der des Jahres 2020. Jedoch sind die erkennbaren Veränderungen der Zeitreihe sowie Abstände zwischen den Veränderungen, in beiden Jahren weitestgehend gleich. Damit sind die Auftragsdaten des Jahres 2019 zwar nicht so stark ausgeprägt wie die des folgenden Jahrs, dennoch enthält die Zeitreihe damit wichtige Informationen über die relative Veränderung der Auftragsdaten in verschiedenen Jahresabschnitten.



Abbildung 14 Auftragsdaten Liniendiagramm, 2020 und 2019 (Eigene Darstellung)

Mit den bereits erkannten Mustern innerhalb der Auftragsdaten kann davon ausgegangen werden, dass die Prognose der zukünftigen Zeitschritte zum Teil schon mit der Betrachtung der historischen Auftragsdaten möglich ist.

Für die weitere Betrachtung wird die Verteilung der Werte innerhalb der Zeitreihe untersucht. Die Darstellung erfolgt dabei mithilfe eines Histogramms, mit welchem die Auftrittshäufigkeit verschiedener Werte innerhalb der Zeitreihe dargestellt werden kann, und mit der Berechnung verschiedener deskriptiver Kennzahlen. In Abbildung 15 ist damit erkennbar, dass die Werte innerhalb der Zeitreihe annähernd normalverteilt sind. Die geringe Streuung wird auch unter Betrachtung der deskriptiven Kennzahlen deutlich (vgl. Tabelle 1), da zu erkennen ist, dass etwa 50 % der Werte weniger als 10000 Einheiten vom errechneten Durchschnitt entfernt liegen. Außerdem lässt das Histogramm darauf schließen, dass sich innerhalb der Zeitreihe auch Ausreißer befinden. Dies wird weiter verdeutlicht mit der Betrachtung der Maxima und Minima innerhalb der Zeitreihe (vgl. Tabelle 1).

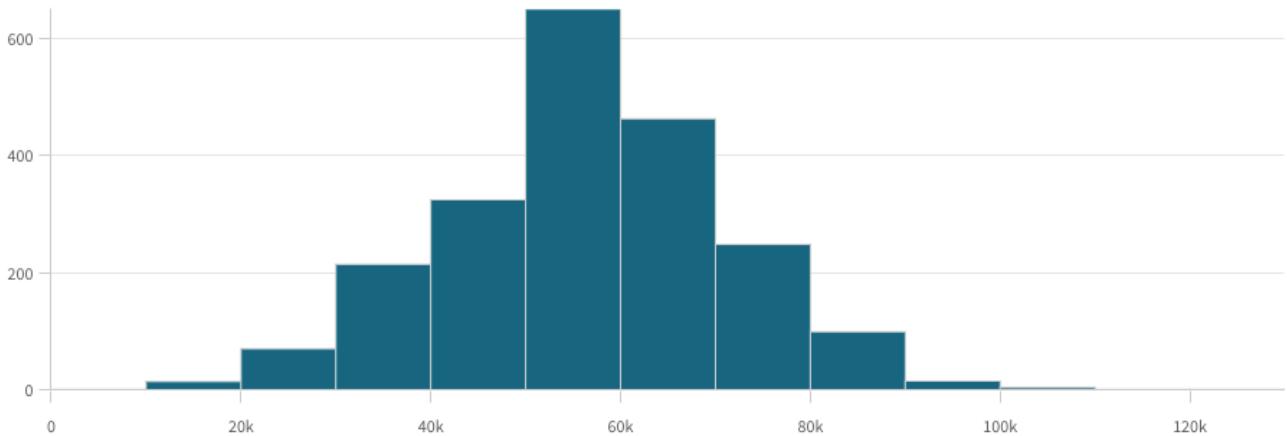


Abbildung 15 Auftragsdaten Histogramm (Eigene Darstellung)

Durchschnitt	56566	25%-Quantil	66146
Median	56630	75%-Quantil	48227
Minimum	1365	Maximum	129499

Tabelle 1 Deskriptive Kennzahlen, Auftragsdaten (Eigene Darstellung)

Für eine weiterführende detailliertere Betrachtung der Ausreißer werden Boxplots verwendet, welche in Anhang 8 zu finden sind. In den Boxplots ist dabei zu erkennen, dass das Auftreten von Ausreißern stark vom betrachteten Wochentag abhängig ist. Der Mittwoch besitzt nach der Darstellung im Boxplot zwar Ausreißer, allerdings würden diese der Betrachtung des gesamten Datensatzes nicht zu der Kategorie der Ausreißer zählen. Schlussfolgernd daraus würde eine Bereinigung, welche 3 % der oberen und unteren Werte aus dem Datensatz entfernt, vermehrt Auftragsdaten, welche an Donnerstagen und Samstagen gemessen worden sind, entfernen, da an diesen Tagen im Schnitt das größte und niedrigste Auftragsvolumen zu verzeichnen ist. Im Schritt der Datenaufbereitung kann damit eine Bereinigung von Ausreißern in Abhängigkeit der Wochentage in Erwägung gezogen werden.

#### 6.1.3.2 Darstellung zusätzlicher Datensätze

Im Folgenden werden die Auftragsdaten mit den in Kapitel 6.1.1 bereits vorgestellten weiteren Datenreihen ins Verhältnis gesetzt. Durch das Erkennen von Mustern und Korrelationen kann ein erster Eindruck gewonnen werden, wie sich die Hinzunahme der Datensätze auf die Prognose auswirken.

Bedingt durch die unterschiedliche Skalierung der Daten und die damit einhergehenden Schwierigkeiten, die Zeitreihen miteinander zu vergleichen, werden die Daten normalisiert. Die Normalisierung geschieht dabei mithilfe einer Min-Max-Normalisierung, wobei alle Werte der Zeitreihen in einem Bereich von 0 bis 1 dargestellt werden. Anschließend können die Zeitreihen mithilfe von Liniendiagrammen verglichen werden. Sollten sich in der Darstellung keine Zusammenhänge zwischen den Zeitreihen erkennen lassen, kann die Verwendung der zusätzlichen Daten in der Analysephase zu einer Verzerrung der Prognoseergebnisse führen. Die untersuchten Diagramme, in welchen die normalisierten Zeitreihen miteinander verglichen werden, sind in den Anhängen 10-12 einsehbar. Dabei wird jeweils das Verhalten der unterschiedlichen Zeitreihen zum Jahresende und zum Jahresanfang untersucht, da besonders in diesen Zeiträumen saisonale Veränderungen deutlich geworden sind (vgl. Kapitel 6.1.3.1).

## **Zu kommissionierende Filialen**

Der Vergleich zwischen den Auftragsdaten und den zu kommissionierenden Filialen (vgl. Anhang 9) konnte aufzeigen, dass leichte Überschneidungen im Verhalten der beiden Zeitreihen erkennbar sind, wodurch sich mithilfe dieser Information womöglich ein Mehrwert in der Analyse generieren lässt. Ein weiterer Vorteil besteht darin, dass die Informationen über die zu kommissionierenden Filialen bereits vor dem Prognosezeitraum bekannt sind.

## **Eingelagerte Transporteinheiten**

Im Vergleich zwischen den Auftragsdaten und den eingelagerten TE (vgl. Anhang 10) ist erkennbar, dass die Zeitreihen zum Jahresende ein ähnliches Verhalten aufweisen. Damit kann ein niedriges Bestellaufkommen zeitgleich mit einer geringen Anzahl eingelagerter TE festgestellt werden. Der Vergleich mit dem Jahresanfang spiegelt jedoch ein gegensätzliches Verhalten wider, wodurch die beiden Zeitreihen gegensätzlich verlaufen und ein niedriges Bestellaufkommen eher auf eine höhere Anzahl eingelagerter TE hinweist. Inwiefern es sich dabei um eine Korrelation oder um ein zufälliges Muster handelt, ist unklar und sollte mittels Iteration und der Verwendung unterschiedlicher Datengrundlagen in der Analysephase geklärt werden.

## **Lagerauslastung**

Beim Vergleich zwischen den Auftragsdaten und der aktuellen Auslastung des Lagers (vgl. Anhang 11) konnte kein bis kaum ein Zusammenhang festgestellt werden. Es wurde lediglich erkennbar, dass eine geringe Veränderung der Lagerkapazitäten meist auf ein niedriges Auftragsvolumen hinweist. Inwiefern damit ein Mehrwert in der Analyse generiert werden kann, ist ebenfalls fraglich und sollte, wie bei der Zeitreihe der eingelagerten TE, mittels des iterativen Prozesses in der Analysephase untersucht werden.

## **6.2 Datenvorverarbeitung**

Unter der Betrachtung der Ergebnisse der Qualitätsprüfung und der explorativen Datenanalyse erfolgt nun die Datenvorverarbeitung. Der Prozessschritt hat zum einen das Ziel, mittels verschiedener Änderungen am Datensatz die bekannten Mängel und somit die Datenqualität zu erhöhen. Außerdem müssen die Daten in eine Form gebracht werden, in welcher sie sich für ein weiterführendes Analyseverfahren eignen. (Chapman u. a. 2000, S. 20).

Die Datenvorverarbeitung lässt sich damit in zwei Schritte unterteilen: Die Datenaufbereitung und die Extraktion von Merkmalen, welche im weiteren Verlauf als Input für das zu erstellende Modell dienen werden.

Im Schritt der Datenaufbereitung werden anhand der Erkenntnisse aus den vorangegangenen Kapiteln Änderungen am Datensatz vorgenommen, mit dem Ziel, die Qualität des Datensatzes zu verbessern und damit Einfluss auf die Analyseergebnisse zu nehmen.

Im darauffolgenden Schritt der Merkmalsextraktion wird der Datensatz unterteilt, sodass anschließend für jede vorherzusagende Sequenz beziehungsweise jedes vorherzusagende Label Eingabesequenzen vorliegen, die für den Trainingsprozess des Modells genutzt werden können. Die extrahierten Eingabesequenzen bilden damit die Grundlage des Trainingsprozesses, wodurch es notwendig ist, dass diese über einen möglichst großen Bezug zum vorherzusagenden Zeitraum verfügen und dass sich mithilfe dieser die vorhersagbaren Komponenten und Muster der Zeitreihe herausfiltern lassen.

### 6.2.1 Datenaufbereitung

Wie sich in der explorativen Analyse gezeigt hat, befinden sich in der Zeitreihe der Auftragsdaten Ausreißer. Bei einem Analyseverfahren mittels neuronaler Netzen ist es nicht zwangsweise notwendig, Ausreißer zu entfernen oder die Zeitreihe zu glätten, da sich gezeigt hat, dass ein rauschender Anteil in den Trainingsdaten nur eine geringe bis kaum bemerkbare Auswirkung auf die Performance des Modells hat (Rolnick u. a. 2018).

Ausreißer lassen sich in zwei Gruppen einteilen: Die natürlichen und die unnatürlichen Ausreißer. Datenpunkte, welche bedingt durch nicht vorhersagbare Umwelteinflüsse stark von anderen Datenpunkten abweichen, werden zu der Kategorie der unnatürlichen Ausreißer gezählt. Das Modell sollte nicht anhand unnatürlicher Ausreißer trainiert werden, da die Ereignisse, welche die Abweichungen hervorgerufen haben, sich nicht reproduzieren lassen. Das Training anhand dieser kann damit zu einer Verzerrung der Prognoseergebnisse führen.

Ausreißer, die durch das Verhalten der Zeitreihe erklärbar sind, gehören zu der Gruppe der natürlichen Ausreißer und sollten in den Trainingsprozess mit einfließen, da sonst das Modell nicht in der Lage wäre, die Veränderung der Zeitreihe in den entsprechenden Abschnitten zu prognostizieren (Aggarwal 2017, S. 3).

Schlussfolgernd daraus ist eine genauere Untersuchung der in Kapitel 6.1.3 erkannten Ausreißer notwendig, um eine Aussage darüber treffen zu können, ob das Entfernen dieser aus der Datenreihe sinnvoll ist. Die Ausreißer, welche im März 2020 festgestellt worden sind (vgl. Abbildung 12), können auf den Ausbruch der Corona-Pandemie in Deutschland zurückgeführt werden. Damit wäre die Veränderung der Zeitreihe an diesem Punkt durch äußere, nicht vorhersagbare Einflüsse hervorgerufen worden und die Ausreißer zählen zu der Kategorie der unnatürlichen Ausreißer. Die Anpassung des Zeitreihenabschnittes wird damit als ratsam betrachtet.

Weiterhin ist zu erkennen, dass es bei im Dezember gemessenen Auftragsdaten vermehrt zu Ausreißern kommt (vgl. Abbildungen 13). Dies ist durch ein erhöhtes Kaufverhalten von Kunden in der Weihnachtsperiode erklärbar. Das Auftreten der Ausreißer verfolgt damit ein Muster und das Modell sollte in der Lage sein, ein höheres Auftragsvolumen in der Weihnachtsperiode prognostizieren zu können. Ein ähnliches Verhalten kann dabei auch in der Periode vor Ostern beobachtet werden.

Unter der Betrachtung der sich im Datensatz befindenden Maxima und Minima wird deutlich, (vgl. Tabelle 1), dass nicht alle Abweichung der Zeitreihe erklärbar sind und auch vermehrt unnatürliche Ausreißer vorhanden sind. Im Folgenden wird daher mit einem Datensatz weitergearbeitet, bei welchem 3 % der oberen und unteren Werte entfernt worden sind.

Zu der Bereinigung der Ausreißer ist außerdem zu erwähnen, dass in einem ersten Durchlauf des Prozessschrittes 3 % der Werte über den gesamten Datensatz entfernt worden sind. Bei der anschließenden Evaluierung der Ergebnisse war der Prognosefehler für das zu erwartende Auftragsvolumen an Donnerstagen deutlich höher als bei allen anderen Wochentagen (vgl. Anhang 12). Die erhöhte Abweichung ließ sich, wie bereits in Kapitel 6.1.3.1 beschrieben, durch die unterschiedlichen Auftragsvolumina, in Abhängigkeit von den Wochentagen, erklären. Mit dem Zurückspringen zur Phase der Datenvorverarbeitung wurde dies angepasst, sodass nun mit Auftragsdaten weitergearbeitet wird, bei welchen das Entfernen der Ausreißer in Abhängigkeit der Wochentage durchgeführt worden ist.

Bedingt durch die Bereinigung der Ausreißer weist die Zeitreihe Lücken auf. Damit eine Analyse weiterhin möglich ist, werden die entstandenen Lücken mit Werten aufgefüllt. Eine häufig verwendete Methode für das Ausfüllen fehlender Zeitschritte ist das lineare Ausfüllen, bei welchem aus dem letzten und dem folgenden Wert des fehlenden Zeitschrittes der Durchschnitt gebildet und eingesetzt wird (Pratama u. a. 2016). Unter der Betrachtung, dass die Zeitreihe in Abhängigkeit der Wochentage starke saisonale Schwankungen aufweist, werden die fehlenden Zeitschritte saisonal-linear aufgefüllt. Dafür wird, ausgehend von der Lücke in der Zeitreihe, der Durchschnitt aus dem sieben vor- und zurückliegenden Wert gebildet. Mit diesem Vorgehen bleiben die beobachteten wöchentlichen Schwankungen erhalten.

### 6.2.2 Merkmalsextraktion

Die Eingabedaten, anhand welcher das Modell eine Prognose trifft, werden auch Merkmale oder Features genannt. Die Features, welche das im weiteren Verlauf erstellte Modell verwendet, sollten aus einer Sequenz verschiedener Zeitschritte bestehen, welche im Folgenden aus den verschiedenen Datenreihen herausgearbeitet werden. Die Selektion der Features lässt sich dabei in zwei Schritte unterteilen. Im ersten Schritt werden verschiedene Sequenzen aus den historischen Auftragsdaten herausgearbeitet. Im zweiten Schritt werden Sequenzen aus den zusätzlich verwendeten Datenreihen extrahiert.

Bei der Verwendung von Deep-Learning-Methoden ist oftmals die manuelle Selektion spezifischer Features innerhalb des Datensatzes nicht notwendig, da das Modell in der Lage sein sollte, die für die Vorhersage wichtigen Informationen aus den Eingabedaten herauszufiltern (Xu, Mak und Cheung 2017). Die folgende spezifische Selektion der Eingabesequenzen ist damit zu begründen, dass in einem bereits durchgeföhrten Prozessdurchlauf lediglich die erfassten Auftragsdaten in chronologischer Reihenfolge und unterschiedliche Längen als Eingabesequenz verwendet worden sind. Das damit trainierte Modell war anschließend kaum in der Lage, den Prognosefehler des naiven Modells zu unterbieten (vgl. Kapitel 4.2.6). Die vollständigen Ergebnisse der Modelle ohne manuelle Selektion der Features sind in Anhang 13 zu finden.

### Auftragsdaten

Mit der Betrachtung der Zeitschritte der Auftragsdaten in chronologischer Abfolge wirkte es so, als wären innerhalb der Zeitreihe größtenteils eine rauschende, nicht vorhersagbare Komponente vorhanden (vgl. Abbildung 11). Erst in der explorativen Datenanalyse konnten Muster und vorhersagbare Komponenten innerhalb der Zeitreihe erkennbar gemacht werden. Schlussfolgernd daraus werden die Eingabesequenzen, welche für das Training des Modells genutzt werden, so modelliert, dass sich mit Hilfe dieser die Merkmale und Muster innerhalb der Zeitreihe erfassen lassen.

Die erste Eingabesequenz, welche herausgearbeitet wird, soll den Zusammenhang zwischen dem Erfassungsmonat und der Höhe der Auftragsdaten verdeutlichen (vgl. Abbildung 12). Die Sequenz umfasst damit neben Datenpunkten aus dem aktuellen Jahr auch Datenpunkte der beiden vorangegangenen Jahren. Dabei werden in jedem Jahr die letzte 30 Zeitschritte, welche vor Beginn der vorherzusagenden Sequenz gemessen worden sind, zusammengefügt, womit die erste Eingabesequenz aus 90 Zeitschritten besteht (vgl. Tabelle 2).

Die zweite Sequenz betrachtet alle Auftragsdaten, welche in den letzten Jahren um den zu prognostizierenden Zeitraum erfasst worden sind. Die Sequenz umfasst damit alle Zeitschritte, welche eine Woche vor dem Prognosezeitraum bis hin zu einer Woche nach dem Prognosezeitraum in den vergangenen Jahren lagen. (vgl. Tabelle 2).

Eine dritte Eingabesequenz, welche aus den Auftragsdaten extrahiert wird, beruht auf der Erkenntnis, dass sich das Bestellverhalten in Abhängigkeit von der Woche im Monat verändert (vgl. Anhang 7). Die Sequenz umfasst damit die äquivalenten Zeitschritte der vergangenen Monate in Bezug zum vorherzusagenden Zeitraum, wodurch sich die auftretenden Schwankungen innerhalb eines Monates berücksichtigen lassen (vgl. Tabelle 2).

Beispiel Vorhersagezeitraum	01.12.2021 – 07.12.2021
Erste Eingabesequenz	01.11.2021 – 31.11.2021 01.11.2020 – 31.11.2020 01.11.2019 – 31.11.2019
Zweite Eingabesequenz	24.11.2020 – 14.12.2020 24.11.2019 – 14.12.2019
Dritte Eingabesequenz	01.11.2021 – 07.11.2021 01.10.2021 – 07.10.2021 01.09.2021 – 07.09.2021

Tabelle 2 Eingabesequenzen, Historische Auftragsdaten (Eigene Darstellung)

## Weitere Datensätze

Die weiteren Datensätze, welche zusätzliche Informationen für die Prognose der zukünftigen Auftragsdaten liefern sollen, werden erst ab dem Jahr 2017 erfasst. Somit umfassen die Datensätze weit aus weniger Datenpunkte als die gesammelten Auftragsdaten. Eine jahresübergreifende Betrachtung der im Folgenden herausgearbeiteten Features wäre damit nur mit der Einschränkung des Trainingsdatensatzes möglich. Unter der Betrachtung, dass die Performance neuronaler Netze maßgeblich von dem Volumen des Trainingsdatensatzes abhängig ist (Alom u. a. 2019, S. 7) und unter der Annahme, dass die historischen Auftragsdaten den größten Zusammenhang mit den zu prognostizierenden Zeitschritten aufweisen, was mit den Ergebnissen der explorativen Datenanalyse zu begründen ist, wird die Einschränkung des Trainingsdatensatzes vermieden, wodurch die jahresübergreifende Betrachtung innerhalb der weiteren Features nicht möglich ist.

Schlussfolgernd daraus bestehen die Eingabesequenzen, welche aus den weiteren Datensätzen herausgearbeitet werden, lediglich aus der chronologischen Abfolge der Zeitschritte, welche vor dem Prognosezeitraum erfasst worden sind.

Die Eingabefrequenz der Zeitreihe, welche die Belegung des Lagers beschreibt, besteht dabei aus vier Zeitschritten, da die Aggregation der Daten später begonnen wurde als bei anderen Datensätzen. Die Features, welche die eingelagerten TE und die zu kommissionierenden Filialen beinhalten, umfassen jeweils 10 Zeitschritte (vgl. Tabelle 3).

Sollte sich in der Analysephase zeigen, dass ein Feature keine Informationen über das zukünftige Verhalten der Auftragsdaten beinhaltet, können Anpassungen an der Selektion der vorgestellten Features vorgenommen werden.

Vorherzusagende Zeitschritte	01.12.2021 – 07.12.2021
Vierte Eingabesequenz	28.11.2021 – 31.11.2021
Fünfte Eingabesequenz	22.11.2021 – 31.11.2021
Sechste Eingabesequenz	22.11.2021 – 31.11.2021

Tabelle 3 Eingabesequenzen, weitere Datensätze (Eigene Darstellung)

### 6.3 Modellierung

Im Prozessschritt der Modellierung wird ein für die Aufgabenstellung geeignetes Analyseverfahren selektiert und anschließend auf den aufbereiteten Datensatz angewendet (Chapman u. a. 2000, S. 23). Vor der Modellierung und dem Trainingsprozess werden alle Labels und Eingabesequenzen in Trainings-, Evaluierungs- und Testdatensatz unterteilt (vgl. Kapitel 3.5). Test- und Evaluierungsdatensatz umfassen jeweils 10% des Gesamtdatensatzes. Die Evaluierung aller Modelle erfolgt demnach anhand des Evaluierungsdatensatzes. Der anschließende Test anhand des Testdatensatzes erfolgt lediglich mit dem Modell, welches am Ende des Modellierungsprozesses selektiert wird.

Bei der Erstellung eines Deep-Learning-Modells lassen sich diverse Parameter anpassen und damit für verschiedene Aufgabenstellungen optimieren. Die veränderbaren Parameter innerhalb eines neuronalen Netzes werden dabei Hyperparameter genannt. Zu den Hyperparametern zählen beispielsweise die Anzahl der Epochen, die Batchgröße oder die ausgewählte Aktivierungsfunktion (vgl. Kapitel 4). Die verfügbaren Hyperparameter können wiederum in Abhängigkeit von der gewählten Architektur variieren. Die Optimierung der Hyperparameter ist ein iterativer Prozess während der Analysephase, da sich einer optimalen Konfiguration von Hyperparametern nur mittels des Testens verschiedenster Kombinationen angenähert werden kann. Weiterhin können die verschiedenen Eingabesequenzen mit unterschiedlichen Architekturen untersucht werden. Mit der Betrachtung verschiedener Architekturen, Eingabesequenzen und Konfigurationen von Hyperparametern ergibt sich eine außerordentliche Anzahl verschiedener Kombinationsmöglichkeiten.

#### 6.3.1 Vorgehen

Im Prozess der Modellierung soll für eine gegebene Eingabe eine bestmögliche Lösung gefunden werden. Für die Bearbeitung des Problems lässt sich dieses im Suchbaum visualisieren. Jede Ebene des Suchbaums beschreibt dabei die Betrachtung einer Eingabesequenz. Die Knoten der Ebene beschreiben die verwendeten Architekturen, welche zur Analyse der jeweiligen Sequenz der Ebene

genutzt werden. In der beispielhaften Darstellung des Suchbaums in Abbildung 16 wurde das Problem lediglich mit der Untersuchung drei verschiedener Architekturen und dreier Eingabesequenzen visualisiert. Die vereinfachte Darstellung verdeutlicht bereits, wie die Komplexität mit der Hinzunahme weiterer Eingabesequenzen und Architekturen mit jeder Ebene zunimmt. Durch die verschiedenen Ebenen und die enorme Anzahl unterschiedlichster Parameter übersteigt das Testen aller Kombinationsmöglichkeiten den verfügbaren Zeitraum, wodurch sich einer optimalen Konfiguration lediglich angenähert werden kann.

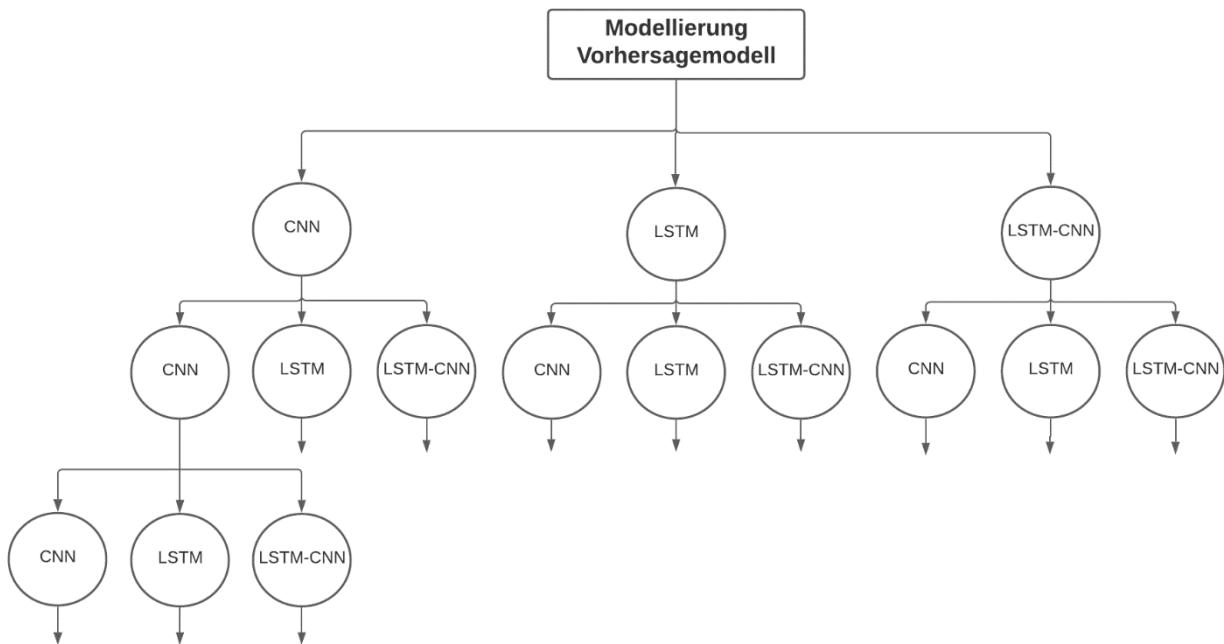


Abbildung 16 Suchbaum Modellierung (Eigene Darstellung)

Die Methodik für das Durchlaufen des Suchbaumes orientiert sich am Vorgehen einer Strahlensuche. Die Strahlensuche stellt eine Erweiterung der Best-First-Suche dar. Bei der Best-First-Suche werden alle Knoten einer Ebene untersucht, anschließend wird von dem Knoten, welcher die besten Ergebnisse erzielen konnte, die nächsttiefere Ebene untersucht und der Prozess wiederholt. Die Strahlensuche verfolgt ein ähnliches Prinzip, jedoch wird anstelle eines einzelnen Knotens eine definierte Anzahl Knoten weiter betrachtet. Die Anzahl Knoten, welche weiter betrachtet werden, wird auch Strahlenbreite genannt (Daumé III und Marcu 2009).

Die Ebenen des Suchbaumes separat voneinander untersucht, indem anhand der jeweiligen Eingabesequenz verschiedene Modelle trainiert werden, welche eigenständig in der Lage sind, die Auftragsdaten zu prognostizieren. Sofern die Features, welche in Kapitel 6.2.2 aus der Datengrundlage extrahiert worden sind, über einen großen Bezug zum Prognosezeitraum verfügen, sollte es mit der separaten Betrachtung jedes Features möglich sein, ein Modell zu trainieren, welches in der Lage ist, die zukünftigen Auftragsdaten vorherzusagen.

Die erreichten Ergebnisse der verschiedenen Architekturen werden anschließend untereinander verglichen, um im weiteren Verlauf in Abhängigkeit von der Strahlenbreite und Ergebnisse verschiedene Architekturen zu Selektieren. Damit lässt sich die Untersuchung der ersten Ebene wie in Abbildung 17 darstellen.

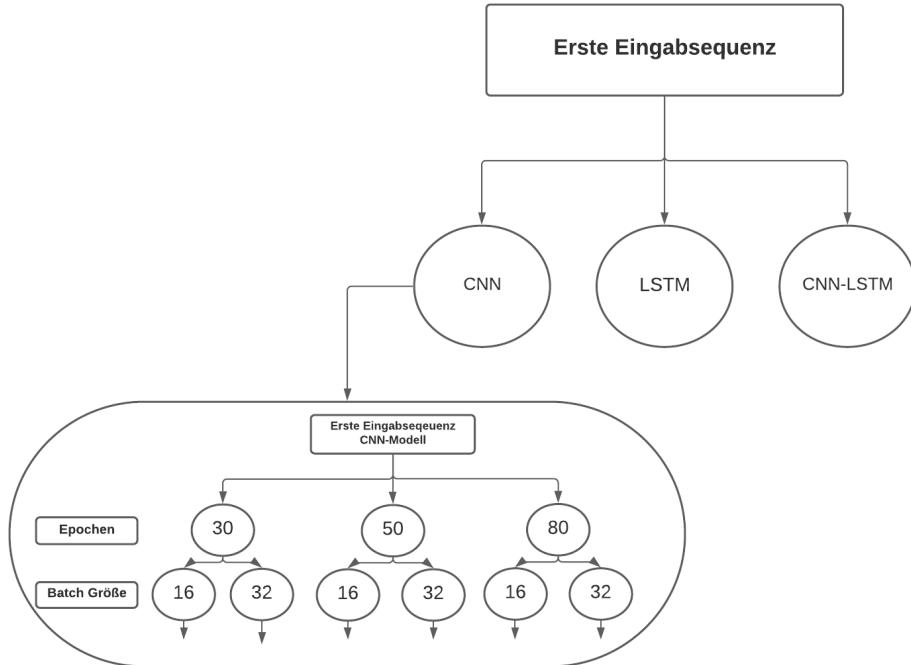


Abbildung 17 Ebene im Suchbaum (Eigene Darstellung)

Damit die Ergebnisse der unterschiedlichen Architekturen vergleichbar sind, erfolgt eine Optimierung der Hyperparameter bei jedem trainierten Modell (vgl. Abbildung 18).

Die Optimierung der Hyperparameter verfolgt dabei das Prinzip zufälligen Suche. Bei der zufälligen Suche wird jeder Hyperparameter mit einem zufälligen Wert initialisiert und das Modell anhand der zufälligen Konfiguration trainiert. Mit dem mehrfachen Wiederholen des Trainingsprozesses können so die Hyperparametern für die Verarbeitung der Eingabesequenzen angepasst werden (Bergstra und Bengio 2012, S.303).

Sobald anhand jeder Eingabesequenz verschiedene Architekturen trainiert und optimiert worden sind, sollte eine Auswahl geeigneter Modelle für die Prognose vorliegen. Außerdem können

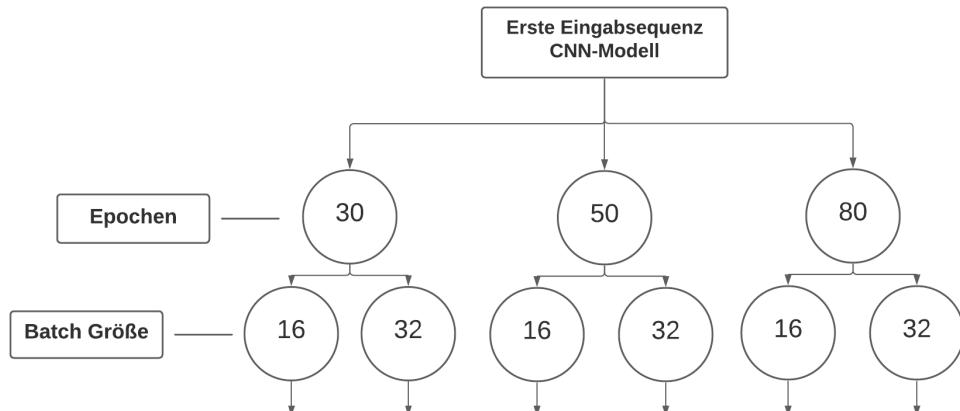


Abbildung 18 Knoten im Suchbaum (Eigene Darstellung)

Eingabesequenzen, welche sich im Vergleich zur Naiven Prognose als nicht geeignet gezeigt haben, aussortiert werden.

Im nächsten Schritt wird aufbauend auf den Ergebnissen ein Bezug zwischen den einzelnen Ebenen hergestellt. Dafür werden Modelle betrachtet, welche auf Grundlage verschiedener Eingabesequenzen trainiert worden sind. Bei der Modellierung erfolgt die Analyse jeder Eingabesequenz anhand von Architekturen, welche sich in der vorherigen Untersuchung als geeignet gezeigt haben. Mit den verschiedenen Trainingsetappen und Iterationsstufen kann der Suchbaum durchlaufen werden. Schlussendlich kann damit ein Modell selektiert werden, mit welchem im weiteren Projektverlauf gearbeitet wird, und es kann davon ausgegangen werden, dass sich durch die verschiedenen Iterationsstufen einer optimalen Lösung angenähert worden ist.

### 6.3.2 Sequenzielle Modelle: Aufbau

Die Architekturen, welche für das Training verwendet werden, stellen Varianten der klassischen CNN- und LSTM-Architektur dar (vgl. Kapitel 4.3). Der Aufbau sowie die Besonderheiten der genutzten Modelle werden im Folgenden kurz erläutert und dargestellt.

Weiterhin ist zu erwähnen, dass die verschiedenen Modelle, welche genutzt werden, keine einheitliche Namensnotation besitzen, wodurch für die Modelle eine Bezeichnung gewählt wird, welche die verwendeten Ansätze bestmöglich beschreiben sollen.

#### CNN

Das CNN-Modell, welches verwendet wird, entspricht der klassischen CNN-Architektur, die in Kapitel 4.3.1 bereits vorgestellt worden ist. Der genaue Aufbau des Modells ist außerdem in Anhang 14 zu finden.

#### LSTM

Das erste LSTM-Modell, welches für die Vorhersage verwendet wird, entspricht dem in Kapitel 4.3.2 vorgestellten klassischen LSTM. Die benutzte Architektur ist in Anhang 15 hinterlegt.

#### Mehrschichtiges LSTM

Das mehrschichtige LSTM-Modell ist ähnlich aufgebaut wie das traditionelle LSTM, allerdings werden hierbei mehrere LSTM-Layer hintereinandergeschaltet. Jeder LSTM-Layer mit Ausnahme des letzten gibt dabei eine Sequenz an Werten zurück, wodurch die Verarbeitung in mehreren Schichten von LSTMs ermöglicht wird. Die Verwendung mehrerer LSTM-Schichten erhöht die Komplexität des Netzes und ermöglicht die mehrfache Transformation einzelner Zeitschritte (Géron 2019, S. 511). Die verwendete Architektur ist in Anhang 16 zu finden.

#### Bidirektionales LSTM

Das bidirektionale LSTM ähnelt von der Funktionsweise dem klassischen LSTM. Allerdings erfolgt die Rückkopplung im bidirektionalen LSTM in beide Richtungen. Damit verfügt das bidirektionale Netz über ein Forward Layer und Backward Layer, wodurch der Zustand der LSTM-Unit sowohl vom vorgelagerten als auch vom nachgelagerten Zeitschritt beeinflusst wird (Géron 2019, S.511). Der Aufbau der Architektur ist außerdem in Anhang 17 zu finden.

## CNN-LSTM

Die vorgestellten Ansätze des CNN- und des LSTM-Modells können auch zusammen in einem Modell kombiniert werden. Dabei wird die Sequenz im ersten Schritt mit dem CNN- und anschließend mit dem LSTM-Ansatz untersucht. Dies hat den Vorteil, dass mithilfe der CNN-Architektur Merkmale aus Teilabschnitten der Zeitreihe herausgearbeitet werden können; anschließend werden die Daten des reduzierten Betrachtungszeitraums an den LSTM-Layer weitergegeben, in welchem die Daten entlang der Zeitachse miteinander ins Verhältnis gesetzt werden können. Damit ermöglicht die Verwendung des kombinierten Ansatzes sowohl die Betrachtung von sequenziellen Abhängigkeiten der Daten als auch die Extraktion von Merkmalen mittels des Convolutional-Ansatzes (Kim und Cho 2019). Die Struktur des Modells ist in Anhang 18 dargestellt.

## Encoder/Decoder

Die im Folgenden vorgestellte Architektur basiert auf der in Kapitel 4.3.3 vorgestellten Encoder-Decoder-Struktur. Der Decoder besteht dabei bei jedem der folgenden Modelle aus einem LSTM-Layer und einem Fully Connected Layer, mit welchen die Untersuchung der Zeitschritte separat erfolgt. Im Folgenden wird deshalb lediglich der Encoder weiter spezifiziert wird.

## Encoder-Decoder-LSTM

Die Architektur des Encoders auf Basis eines LSTM überschneidet sich mit dem bereits vorgestellten LSTM-Modell, das Modell ist in Anhang 19 abgebildet.

## Encoder-Decoder-CNN

Das Gleiche gilt für das Modell des Encoders auf Basis eines CNN-Modells. Die Architektur des Modells ist in Anhang 20 zu finden.

## Encoder-Decoder-ConvLSTM

Der Encoder des letzten verwendeten Modells zur Untersuchung der Eingabesequenzen beruht auf der Verwendung eines 2D-Convolutional-LSTM-Layers, mit welchem es die Erweiterung des CNN-LSTM-Ansatzes darstellt. Im Vergleich zum CNN-LSTM-Modell, bei welchen zuerst mittels Filter Merkmale der Zeitreihe extrahiert und diese anschließend mithilfe der LSTM-Architektur interpretiert, verwendet das Convolutional-LSTM (ConvLSTM) die Convolution-Methodik direkt während des Prozesses des Einlesens der Daten in das LSTM. Damit erfolgt die Untersuchung der Datenreihe mithilfe der Convolution innerhalb jeder LSTM-Zelle. Das hat zur Folge, dass sowohl die Transformation des Inputs als auch die Rückkopplung im LSTM mithilfe verschiedener Filter untersucht werden (Shi u. a. 2015). Der Aufbau des Modells ist in Anhang 21 einsehbar.

### 6.3.3 Sequenzielle Modelle: Ergebnisse

Im Folgenden werden die Prognoseergebnisse vorgestellt, welche mithilfe der beschriebenen Modelle auf dem Evaluierungsdatensatz erreicht wurden. Eine Gesamtübersicht der Ergebnisse ist außerdem in Anhang 22-23 zu finden.

Für die Evaluierung der Ergebnisse wird die Abweichung der prognostizierten Zeitschritte im Verhältnis zum Label berechnet. Die Abweichung oder der Prognosefehler kann mittels verschiedener Metriken bestimmt werden. Für die Evaluierung der Ergebnisse wird sich auf die folgenden Fehlermaße beschränkt: RMSE, MAE, MAPE.

Für den Root-Mean-Square-Error (RMSE) wird der MSE berechnet (vgl. Kapitel 4.2.2) und anschließend die Wurzel gezogen. Mittels des RMSE wird die mittlere quadratische Abweichung der Prognosewerte bestimmt. Die Berechnung des MAE ist deckungsgleich mit der Berechnung der vorgestellten Loss-Function (vgl. Kapitel 4.2.2). Als letzte Metrik wird der Mean-Absolute-Percentage-Error (MAPE) bestimmt. Der MAPE berechnet die mittlere prozentuale Abweichung der Prognosewerte im Verhältnis zum Label. Mit der Betrachtung der vorgestellten Metriken lassen sich zum einen die Ergebnisse der unterschiedlichen Modelle miteinander vergleichen, zum anderen kann ein Überblick über die Genauigkeit der prognostizierten Zeitschritte geschaffen werden. Mit dem vorgestellten Vorgehen orientiert, an einer Strahlensuche, wird im Folgenden mit einer Strahlenbreite von 3 weitergearbeitet (vgl. Kapitel 6.3.1), wodurch für jede Sequenz nur die 3 Architekturen weiter untersucht werden, welche gemessen am RMSE den geringsten Prognosefehler auf den Evaluierungsdatensatz aufweisen konnten.

Für die weitere Untersuchung der Ergebnisse, werden die Fehlermaße der Naiven Prognose bestimmt (vgl. Abbildung 19). Der Prognosefehler der naiven Vorhersage stellt die Mindestanforderung an jedes Modell dar (vgl. Kapitel 4.2.6). Sollte es unter der Betrachtung einer Eingabesequenz nicht möglich sein, den Fehler der naiven Prognose zu minimieren, kann von einem schwachen bis kaum vorhandenen Zusammenhang zwischen Sequenz und Label ausgegangen werden.

```

Naives Forecasting
RMSE_Naive: 14151.96
MAE_Naive: 11831.02
MAPE_Naive: 0.23

```

*Abbildung 19 Kennzahlen Naive Prognose (Eigene Darstellung)*

Verschiedene Modelle, welche anhand der extrahierten Sequenzen aus den Auftragsdaten trainiert worden sind, waren in der Lage, den Prognosefehler des naiven Modells zu unterbieten. In Tabelle 4 ist eine Auflistung der Modelle und des RMSE der Prognose auf dem Evaluierungsdatensatz dargestellt, eine Gesamtübersicht mit den weiteren Metriken ist in Anhang 22 zu finden. Bei allen vorgestellten Modellen erfolgte die Optimierung der Hyperparameter mithilfe der zufälligen Suche über 50 Iterationsschritte (vgl. Kapitel 6.3.1). Eine beispielhafte Darstellung des Optimierungsprozesses eines Modells ist in Anhang 24 zu finden.

Die Prognosen lassen erkennen, dass die ersten Sequenzen Informationen zum Prognosezeitraum beinhalten, da der RMSE im Vergleich zum naiven Modell bei verschiedenen Architekturen um bis zu 50 % reduziert werden konnte (vgl. Tabelle 4). Außerdem ist zu bemerken, dass die auf dem CNN-Ansatz basierenden Modelle deutlich bessere Prognoseergebnisse liefern konnten als die LSTM-Modelle. Zu erklären ist dies durch den Umstand, dass die Sequenzen, welche aus den Auftragsdaten extrahiert worden sind, über 40 Zeitschritte beinhalteten. Wie beschrieben kann das Verarbeiten langer Sequenzen kann zu verschiedenen Problemen innerhalb eines RNN führen (vgl. Kapitel 4.3.2).

Trotz der schwachen Ergebnisse der LSTM-Modelle konnten mittels der CNN-basierten Architekturen gute Ergebnisse auf dem Evaluierungsdatensatz erreichen werden. Der Prognosefehler gemessen am MAE betrug bei diesen Modellen im Schnitt ca. 6000 Einheiten. Außerdem konnte mittels des MAPE eine durchschnittliche prozentuale Abweichung von 10 % festgestellt werden.

Architekturen	Erste Sequenz	Zweite Sequenz	Dritte Sequenz
CNN	8223	8562	8562
LSTM	24346	55493	55493
LSTM-Stacked	16101	14076	14076
BI-LSTM	41811	41807	41807
CNN-LSTM	8290	9301	9301
ED-LSTM	11408	12083	12083
ED-LSTM-CNN	22891	13681	13681
ED-ConvLSTM	10588	11245	11245

Tabelle 4 Ergebnisse Sequenzielle Modelle; Historische Auftragsdaten (Eigene Darstellung)

Nach dem in Kapitel 6.3.1 vorgestellten Vorgehen und einer definierten Strahlenbreite von 3 werden im weiteren Verlauf lediglich das CNN-, das CNN-LSTM- und das ConvLSTM-Modell für die Untersuchung der ersten Eingabesequenzen weiter betrachtet, da bei diesen Modellen der Prognosefehler, gemessen am RMSE, am geringsten ausgefallen ist.

Die Unterschiede zwischen den einzelnen Prognosen und den dazugehörigen Labels lassen sich mithilfe eines Liniendiagramms darstellen und analysieren. In Abbildung 20 wurden die Prognosen für 30 Zeitschritte der Modelle, welche anhand der ersten Eingabesequenz trainiert worden sind, dargestellt. Dabei ist deutlich zu erkennen, wie die verschiedenen Modelle das Verhalten der Zeitreihe adaptiert haben und in der Lage waren, das zukünftige Verhalten der Zeitreihe zu prognostizieren. Dennoch ist zu bemerken, dass die Prognosen teilweise entgegengesetzt der eigentlichen Zeitreihe verlaufen. Die Abschnitte, in welchen dieses Verhalten zu beobachten ist, überschneiden sich bei der Betrachtung der verschiedenen Modelle.

Weiterhin konnte festgestellt werden, dass es zu stärkeren Abweichungen zwischen Prognose und Label kommt, sobald Schwankungen und Anomalien innerhalb der Zeitreihe auftreten. Besonders deutlich wird das mit der Darstellung der prognostizierten Werte über den gesamten Evaluierungsdatensatz (vgl. Anhang 22).

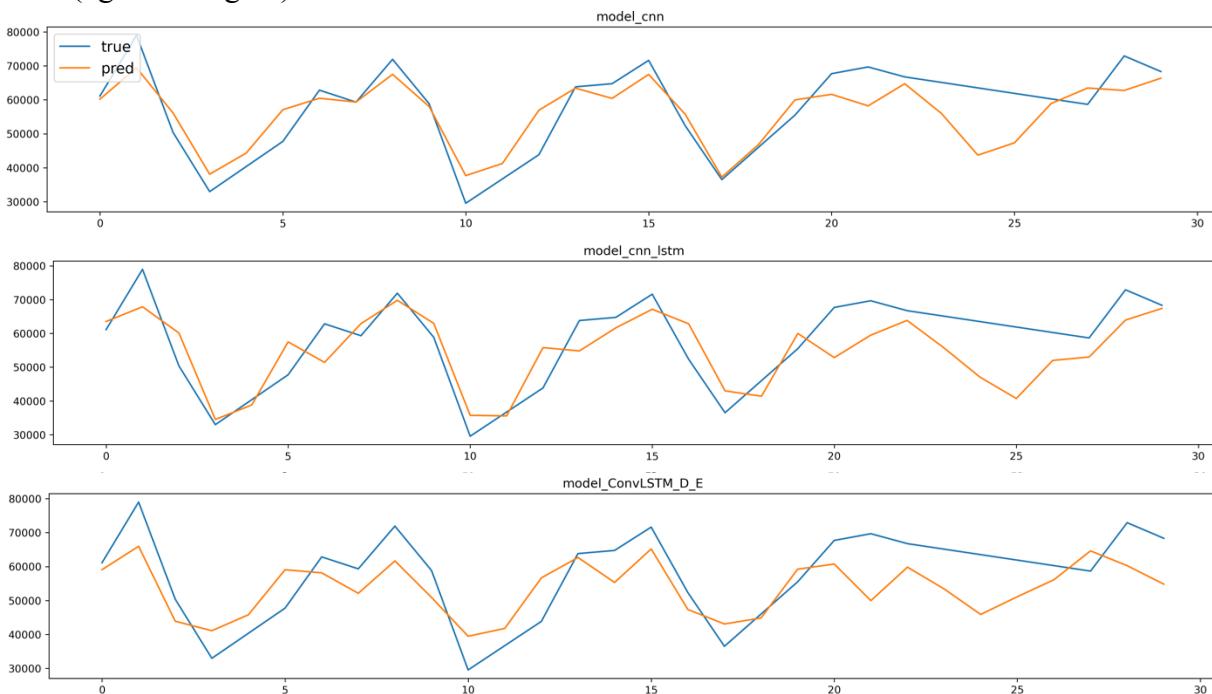


Abbildung 20 Liniendiagramm Sequenzielle Modelle; Historische Auftragsdaten (Eigene Darstellung)

Im Folgenden werden die Prognoseergebnisse der Modelle vorgestellt, welche auf Grundlage der übrigen Eingabesequenzen trainiert worden sind. Bei der Betrachtung der Ergebnisse der explorativen Datenanalyse war zu erwarten, dass die Ergebnisse schlechter ausfallen werden als bei den zuvor betrachteten Sequenzen (vgl. Kapitel 6.3.1). Diese Vermutung konnte in der Analysephase bestätigt werden und der Fehler des naiven Modells konnte mithilfe der weiteren Modelle kaum bis gar nicht minimiert werden.

Eine vollständige Übersicht der Analyseergebnisse sowie die weiteren Fehlermetriken ist in Anhang 23 zu finden. Eine Auflistung der verschiedenen Prognosefehler, gemessen am RMSE, ist in Tabelle 5 dargestellt. In der Tabelle ist dabei zu erkennen, dass im Vergleich zu den vorherigen Eingabesequenzen die auf der LSTM-Architektur basierenden Modelle die besten Ergebnisse erzielen konnten. Dies ist darauf zurückzuführen, dass die Eingabesequenzen im Gegensatz zu den vorher getesteten Sequenzen nur aus 10 Zeitschritten bestehen (vgl. Kapitel 6.2.2), wodurch die Problematiken, welche durch die langen Eingabesequenzen bei RNNs hervorgerufen werden, nicht so stark ausfallen.

Architekturen	Vierte Sequenz	Fünfte Sequenz	Sechste Sequenz
CNN	22148	19774	19812
LSTM	13668	14236	13621
LSTM-Stacked	13628	16796	13916
BI-LSTM	55180	55192	55190
CNN-LSTM	55492	20404	20384
ED-LSTM	19141	20158	20469
ED-LSTM-CNN	13693	15181	13974
ED-ConvLSTM	13650	32473	15259

Tabelle 5 Ergebnisse Sequenzielle Modelle; Weitere Datensätze (Eigene Darstellung)

Die stärkere Abweichung zwischen den Prognosen und Labels und der damit höher ausfallende RMSE wird bei Betrachtung von Abbildung 21 deutlich. Dabei ist zu erkennen, dass auch die Modelle, welche den geringsten Prognosefehler aufwiesen, mittels der vorhandenen Informationen nicht in der Lage waren, Muster in der Zeitreihe zu erkennen und das zukünftige Verhalten zu prognostizieren.

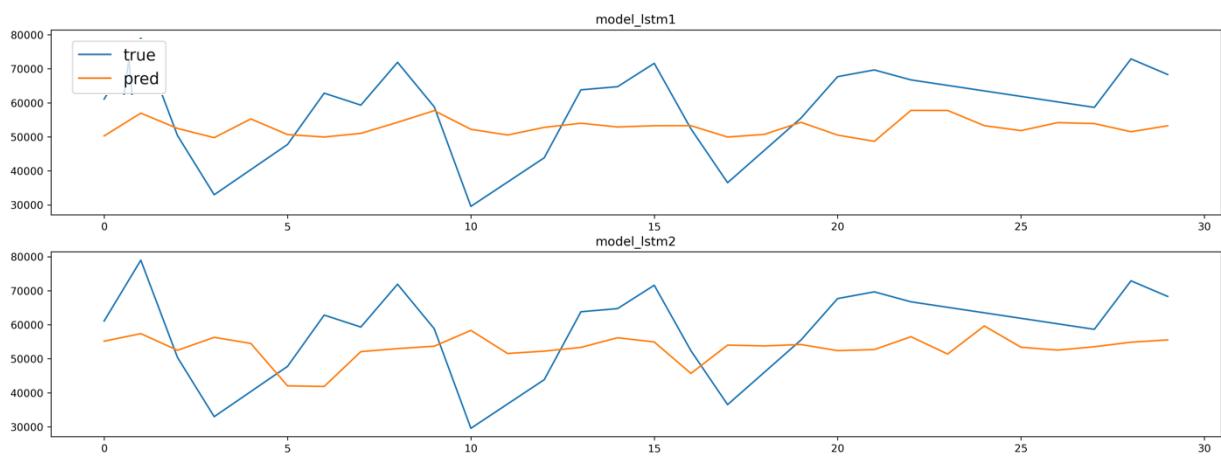


Abbildung 21 Liniendiagramm Sequenzielle Modelle; Weitere Datensätze (Eigene Darstellung)

#### 6.3.4 Multi-Head-Modelle: Aufbau

Im Folgenden werden Modelle vorgestellt, welche auf Grundlage verschiedener Eingabesequenzen die zukünftigen Zeitschritte prognostizieren. Dabei wird auf die Ergebnisse des vorangegangenen Kapitels zurückgegriffen und die einzelnen Sequenzen werden mithilfe von Architekturen verarbeitet, welche gemessen am RMSE den geringsten Fehler aufgewiesen haben. Die Modellierung und die Funktionsweisen der verschiedenen Modelle erfolgen nach den bereits vorgestellten Prinzipien aus Kapitel 6.3.2

Die bisher vorgestellten Modelle gehörten zu der Gruppe der sequenziellen Modelle, da jede Schicht einen Inputvektor und einen Ausgabevektor besitzt. Bei dem im Folgenden verwendeten Architekturen werden durch die Betrachtung mehrerer Eingabesequenzen innerhalb einer Architektur mehrere

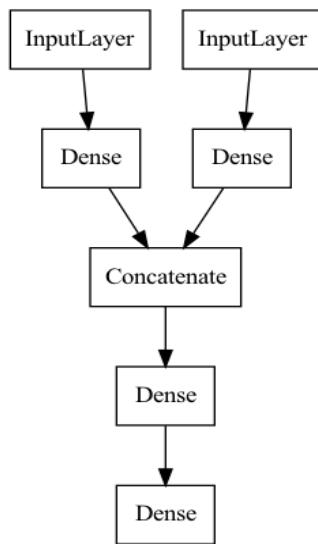


Abbildung 22 Beispiel Multi-Head-Modell (Eigene Darstellung)

Inputvektoren benötigt, wobei auch von einem Multi-Head-Modell gesprochen wird. Bei der Untersuchung von zwei Eingabesequenzen innerhalb eines Modells kann sich damit eine Struktur wie in Abbildung 22 ergeben. Dabei durchlaufen alle Eingabesequenzen unabhängig voneinander verschiedene Stufen der Transformation, bis die Zwischenergebnisse zusammengefügt werden. Mit der Aggregation der Zwischenergebnisse können diese im weiteren Verlauf gemeinsam weiterverarbeitet werden.

Im ersten Schritt werden Modelle auf Basis der Eingabesequenzen erstellt, welche den Prognosefehler des naiven Modells unterbieten konnten (vgl. Kapitel 6.3.3). Bei der Erstellung werden dafür die Architekturen verwendet, welche sich im vorangegangenen Kapitel als am geeignetsten gezeigt haben. In der Modellierung wird dabei der Ansatz verfolgt, dass jede Eingabesequenz separat eingelesen und mit einer ausgewählten Architektur untersucht wird. Anschließend werden die Ergebnisse zusammengeführt. Die aggregierten Ergebnisse durchlaufen anschließend weitere Fully Connected Layer. Sofern sich ergänzende Informationen innerhalb betrachteter Sequenzen befinden, sollten sich diese auf die Prognoseergebnisse auswirken und der Fehler gemessen am RMSE, im Vergleich zur sequenziellen Betrachtung, minimiert werden.

Basierend auf der Beschreibung der Modellierung werden CNN-, CNN-LSTM- und Conv-LSTM-Ansätze erstellt, welche auf Grundlage der ersten drei Eingabesequenzen eine Prognose treffen verwenden. Eine grafische Darstellung des Aufbaus der Modelle ist in Anhang 25-27 zu finden.

Um überprüfen zu können, ob die letzten drei Eingabesequenzen trotz der schwachen Ergebnisse in der separaten Betrachtung in Verbindung mit den weiteren Eingabesequenzen einen Mehrwert in der Analyse darstellen, werden im zweiten Schritt Modelle erstellt, welche auf Grundlage aller Sequenzen eine Vorhersage treffen. Für die Modellierung der Modelle werden dabei die Architekturen verwendet, welche den geringsten Prognosefehler aufweisen konnten. Dabei werden die ersten drei Sequenzen mit einer CNN-basierten Architektur untersucht. Für die übrigen Sequenzen wird ein LSTM-basierter Ansatz verwendet. Bei der Modellierung der Modelle wird dabei der gleiche Ansatz verfolgt wie bei den bereits vorgestellten Multi-Head-Modellen. Eine Gesamtübersicht der Modelle ist in Anhang 28-30 zu finden.

### 6.3.5 Multi-Head-Modelle: Ergebnisse

Im Folgenden folgt die Betrachtung der Prognoseergebnisse, welche mit den verschiedenen Multi-Head-Architekturen erreicht werden konnten. Die Evaluierung erfolgt dabei anhand der gleichen Fehlermetriken, welche auch schon für die sequenziellen Modelle verwendet worden sind (vgl. Kapitel 6.3.2). Der Prognosefehler konnte mittels der Modelle, welche auf Grundlage der ersten drei Eingabesequenzen trainiert worden sind, weiter minimiert werden (vgl. Tabelle 6). Dabei konnten, anders als bei der sequenziellen Betrachtung, mithilfe der CNN-LSTM-Architektur die besten Ergebnisse erzielt werden. Die mittlere prozentuale Abweichung, gemessen mittels des MAPE, konnte auf einen Wert von 9 % gebracht werden, außerdem wurde der RMSE um knapp 1000 Einheiten verringert.

Architekturen	RMSE	MAE	MAPE
CNN	8488	6080	0,11
CNN-LSTM	7464	5123	0,09
ConvLSTM	10754	8630	0,17

Tabelle 6 Ergebnisse Multi-Head Modelle; Historische Auftragsdaten (Eigene Darstellung)

Mit der Visualisierung der Prognosen und der Labels kann erkannt werden, dass das Modell nun ein leicht verbessertes Verhalten aufweist, die auftretenden Veränderungen der Zeitreihe des Evaluierungsdatensatzes besser zu antizipieren und damit auch zu prognostizieren (vgl. Abbildung 23).

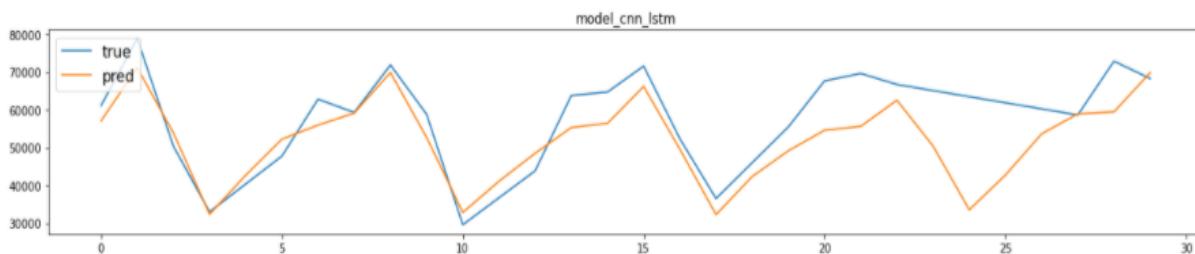


Abbildung 23 Liniendiagramm Multi-Head; LSTM-CNN Historische Auftragsdaten (Eigene Darstellung)

Nach der Untersuchung der Modelle, welche auf der Verarbeitung der historischen Auftragsdaten basieren, werden nun die Ergebnisse der Modelle vorgestellt, bei welchen für die Prognose alle Eingabesequenzen betrachtet werden. Mit der Betrachtung der Ergebnisse sollte eine darüber getroffenen werden können, inwiefern die Prognoseergebnisse durch das Hinzufügen der zusätzlichen

Informationen verzerrt werden oder ob der Prognosefehler mittels der Kombination verschiedener Information weiter minimiert werden konnte.

Die erreichten Ergebnisse sind in der Tabelle 7 aufgelistet. Dabei ist zu erkennen, dass der Prognosefehler bei der Betrachtung aller Eingabesequenzen nicht weiter minimiert werden konnte. Außerdem haben sich die Ergebnisse durch die Hinzunahme der Information leicht verschlechtert. Mit dem festgestellten geringen Zusammenhang der weiteren Information erscheint lediglich die Verwendung der historischen Auftragsdaten zum Prognostizieren der zukünftigen Zeitschritte als sinnvoll.

Architekturen	RMSE	MAE	MAPE
CNN	9735.0	7005.0	0.12
ConvLSTM	8518.0	6396.0	0.12
CNN-ConvLSTM	8057	5648	0.11

Tabelle 7 Ergebnisse Multi-Head Modelle; Alle Datensätze (Eigene Darstellung)

## 6.4 Evaluierung

Im Prozessschritt der Evaluierung erfolgt die Untersuchung der Ergebnisse, welches mittels des selektierten Analyseverfahrens erreicht worden sind. Zum Teil ist dies bereits in Kapitel 6.3 geschehen, da die Berechnung verschiedener Fehlermetriken notwendig war, um die verwendeten Analyseverfahren miteinander zu vergleichen und die Modellierung des Modells weiter zu optimieren. Entsprechend werden im Folgenden nur die Prognosen des CNN-LSTM-Modells, basierend auf den ersten drei Eingabesequenzen, weiter betrachtet, da hierbei der geringste Prognosefehler festgestellt worden ist. Mit der weiteren Untersuchung der Ergebnisse können Ursachen für Fehler oder Schwächen des Modells ausfindig gemacht werden.

Anschließend wird das Modell zum einen anhand des Testdatensatzes erneut evaluiert, um Rückschlüsse auf die zu erwartenden Ergebnisse im Live-Betrieb ziehen zu können, zum anderen werden die Prognose des Modells und die aktuell im Lager getätigte Prognose verglichen, um anhand dessen eine Aussage über den wirklichen Mehrwert des Modells treffen zu können.

Für die Untersuchung der einzelnen Prognosen auf dem Evaluierungsdatensatz werden die Prognose und die Labels mittels eines Reports dargestellt. Außerdem werden die oberen und unteren Grenzen der Prognoseintervalle, mit einer 95% Sicherheitswahrscheinlichkeit bestimmt (vgl. Kapitel 2.2). Damit trifft das Modell neben der Punktprognose auch eine Probabilistische Vorhersage. Mit dem Vergleich der beiden Prognosen kann eine Aussage über die Eintrittswahrscheinlichkeit des prognostizierten Wertes und damit über die Unsicherheit der Prognose getroffen werden.

## Evaluationsreports

Die verschiedenen Kennzahlen lassen sich anschließend mithilfe eines Linien- und Balkendiagramms miteinander ins Verhältnis setzen. Ein beispielhafter Auszug des Evaluierungsreportes für den Prognosezeitraum 05.08.2021 – 11.08.2021 ist in Abbildung 24 zu erkennen. Vollständige Evaluierungsreports sind in Anhang 31 einzusehen. Mit der Betrachtung unterschiedlicher Ergebnisse lässt sich erkennen, dass sich mittels der berechneten Prognoseintervalle eine präzise Aussage über die Sicherheit der Eintrittswahrscheinlichkeit treffen lässt. Weiterhin kann erkannt werden, dass der Prognosefehler für verschiedene Labels weit unter den bestimmten Fehlermetriken liegt. Der erhöhte Prognosefehler über den gesamten Datensatz lässt sich mit auf vereinzelt auftretende extremer Abweichung zurückführen. Eine direkte Ursache für das Verhalten konnte nicht ausfindig gemacht werden, eine Möglichkeit kann die fehlende Plausibilitätsprüfung bei der Datenerfassung sein (vgl. Kapitel 6.1.2).

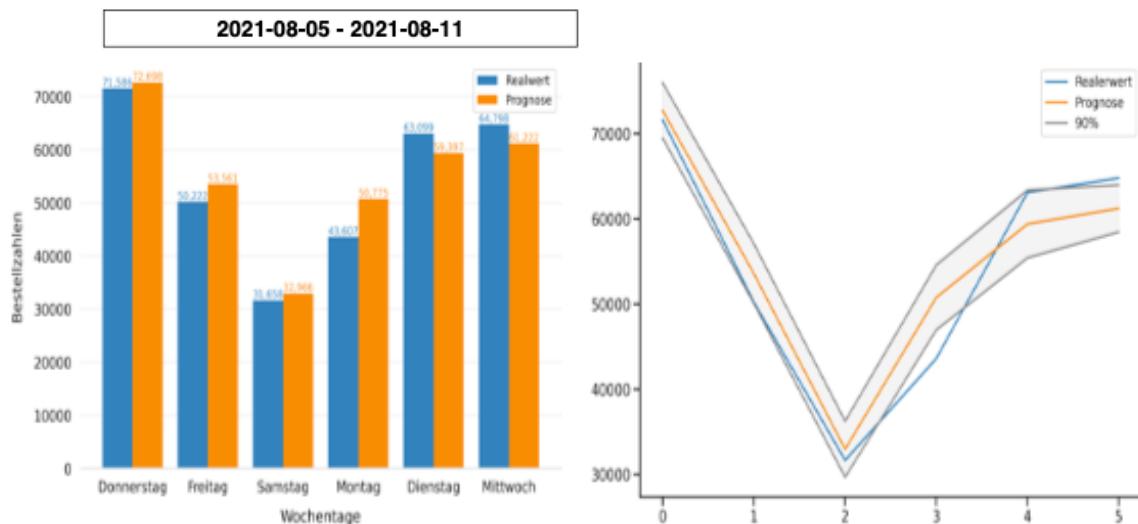


Abbildung 24 Evaluierung Label (Eigene Darstellung)

Für die weitere Untersuchung wird die Abweichung des Labels von der Prognose über den gesamten Evaluierungsdatensatz in einem weiteren Report betrachtet. Zum einen wird dafür untersucht, wie sich die Genauigkeit im Laufe der sieben Schritte verändert (vgl. Tabelle 8), zum anderen wird betrachtet, in welchen Abschnitten der Zeitreihe vermehrt Fehler auftreten. Dafür werden die bekannten Fehlermetriken in Abhängigkeit von den Wochentagen bestimmt. Ein Beispiel des gesamten Reports ist in Anhang 32 zu finden.

Mit der Betrachtung der Tabelle ist erkennbar, dass sich die Prognosegenauigkeit im Laufe der sieben Zeitschritte nur marginal verschlechtert hat. Falls notwendig, kann somit eine Verlängerung des Prognosezeitraumes in Erwägung gezogen werden.

Step	RMSE	MAE	MAPE
1	7363	5065	0.09
2	7434	5137	0.09
3	7493	5125	0.09
4	7486	4951	0.09
5	7560	5158	0.09
6	7423	4959	0.09
7	7517	5003	0.09

Tabelle 8 Prognosefehler bei unterschiedlicher Schrittweite (Eigene Darstellung)

## Testdatensatz Evaluation

Die Evaluierung des Modells auf den Testdatensatz konnte ähnliche Prognoseergebnisse erreichen wie die auf den Evaluierungsdatensatz. Der Prognosefehler hat sich dabei nur leicht erhöht, wodurch mit einer ähnlichen Leistung im Live-Betrieb zu rechnen ist. Eine Gesamtübersicht der Analyseergebnisse auf den Testdatensatz ist in Anhang 33 zu finden.

## Vergleich bisher genutzte Prognose

Um eine schlussendliche Aussage über den Mehrwert des Modells treffen zu können, wird die Prognose des Modells mit der im Lager manuell erstellten Prognose, welche aktuell für die Planung verwendet wird, verglichen. Der Untersuchungszeitraum umfasst dabei den gesamten Dezember 2021.

Das Verhalten der Zeitreihe im Dezember weicht bedingt durch Feiertage und das Weihnachtsgeschäft stark vom normalen Verhalten ab (vgl. Abbildung 12), was die Prognose zusätzlich erschwert. Weiterhin ist zu erwähnen, dass bedingt durch die Anforderung, die zu erwartenden Auftragsdaten des gesamten Dezembers zu prognostizieren, der Outputlayer des Modells angepasst worden ist, so dass eine Vorhersage von 30 Zeitschritten möglich ist.

Eine Gesamtübersicht über die Vorhersagen der einzelnen Tage ist in Anhang 34 zu finden. Die Prognosefehler der beiden Vorhersagen für den Monat Dezember sind in Tabelle 9 aufgelistet. Bei der manuellen Prognose basierend auf statistischen Methoden und Erfahrungswerten, ist zu erkennen, dass der Prognosefehler nur knapp unter dem des naiven Modells liegt. Der Prognosefehler des neuronalen Netzes, gemessen an den verschiedenen Fehlermetriken, ist höher aus gefallen als auf den Evaluierungs- und Testdatensatz zuvor, was möglicherweise auf den verlängerten Prognosezeitraum von 30 Zeitschritten zurückzuführen ist. Dennoch konnte sowohl der absolute als auch der prozentuale Fehler im Vergleich zur manuellen Prognose halbiert werden, was den Mehrwert des Einsatzes des Modells bestätigt. Eine Inbetriebnahme in den Live-Betrieb kann somit als sinnvoll betrachtet werden.

Prognose	RMSE	MAE	MAPE
Manuell	16185	12602	0,2
CNN-LSTM	8406	6534	0,1

Tabelle 9 Vergleich Prognosen für Dezember 2021 (Eigene Darstellung)

## 6.5 Bereitstellung

Im Schritt der Bereitstellung wird der eigentliche Mehrwert gewonnen, indem das vorgestellte Modell in den Live-Betrieb eingebunden wird. Damit kann die Entscheidung oder der Prozess anhand der Prognosen angepasst werden. Im Laufe der Zeit können neue Faktoren innerhalb der Zeitreihen auftreten, wodurch ein weiterführendes Monitoring der Prognoseergebnisse nach Inbetriebnahme des Modells notwendig ist (Chapman u. a. 2000, S.28).

Im Live-Betrieb verwendet das Modell aktuell erfasste Daten, um zukünftige Zeitschritte zu prognostizieren. Diese werden anschließend mithilfe eines Reports dargestellt und an eine weisungsbefugte Stelle weitergegeben. Ein dafür geeigneter Report ist exemplarisch in Anhang 35 dargestellt.

Der Prozessschritt wurde der Vollständigkeit halber beschrieben, jedoch wird auf ein konkretes Vorgehen zur Einbindung des Modells in den Live-Betrieb sowie auf die dafür nötige IT-Infrastruktur im Rahmen dieser Arbeit nicht eingegangen. Dies ist damit zu begründen, dass die Problemstellung sich auf die Prognose und Analyse der Auftragsdaten beschränkt, was mit dem Beenden des Kapitels 6.4 vervollständigt worden ist.

## 7. Diskussion und Ausblick

### 7.1 Diskussion

Im Rahmen der Bachelorarbeit konnten die Ergebnisse der Untersuchung zeigen, dass sich die zukünftigen Zeitschritte der betrachteten Zeitreihe prognostizieren lassen, mithilfe eines Modells, welches auf Methoden aus dem Bereich des Deep Learnings beruht. Das Vorgehen zur Erstellung des Modells orientierte sich dabei am Vorgehensmodell des CRISP-DM. In einer explorativen Datenanalyse konnten auftretende Saisonalitäten und Trends innerhalb der Zeitreihe ausfindig gemacht werden. Basierend auf den daraus gewonnenen Informationen konnten Zusammenhänge zwischen den zurückliegenden Auftragsdaten und den zukünftig zu erwartenden Auftragsdaten festgestellt werden. Anschließend konnten anhand dessen verschiedene Sequenzen aus den Auftragsdaten extrahiert werden, welche darauffolgend als Trainingsgrundlage für die Modelle dienten.

Der Modellierungsprozess wurde systematisch, orientiert am Vorgehen einer Strahlensuche, durchlaufen. Durch die verschiedenen Ebenen der Optimierung und das iterative Testen verschiedenster Architekturen konnte zum einen ein umfassender Überblick über die Eignung verschiedenster Modelle zur Zeitreihenanalyse generiert werden, zum anderen konnte mit der Veränderung der Datengrundlage eine Aussage über den Zusammenhang zwischen den verwendeten Informationen und den zukünftigen Zeitschritten getroffen werden. Zum Prognostizieren der Auftragsdaten konnte dabei lediglich mit der Verwendung historischer Auftragsdaten ein Mehrwert generiert werden.

Bei der Analyse der verschiedenen Modelle und Prognosen konnte erkennbar gemacht werden, dass die Modelle, welche auf dem Einsatz eines Convolutional Layer beruhen, weitaus bessere Ergebnisse erreichen konnten als verschiedene Arten von RNNs. Schlussendlich konnten die besten Ergebnisse mit einer LSTM-CNN-Architektur erreicht werden, wobei die nächstfolgenden sieben Schritte der Zeitreihe mit einer durchschnittlichen prozentualen Abweichung von 9 % vorhergesagt werden konnten. Im Vergleich zu der naiven Prognose mit einer Abweichung von 23 % und der aktuell verwendeten Prognose im Lager mit einer Abweichung von 20 % konnten somit signifikante Verbesserungen erzielt werden.

Zu Beginn der Forschung wurde davon ausgegangen, dass sich die Auftragsdaten durch verschiedene Einflussfaktoren beschreiben lassen. Damit wurde vermutet, dass auftretende Anomalien in der Zeitreihe mit der Analyse verschiedener Faktoren prognostizierbar sind. Dies konnte nur teilweise bestätigt werden, da mittels der Hinzunahme verschiedener Datensätze kein Mehrwert generiert werden konnte. Jedoch mit der Betrachtung der historischen Auftragsdaten die Prognose der zukünftigen Zeitschritte ermöglicht worden ist.

Außerdem wurde zunächst damit gerechnet, dass die genaueste Prognose mit der Verwendung eines RNN erreicht werden würde, da diese oftmals für die Analyse sequenzieller Daten und Zeitreihen verwendet werden und in der Vergangenheit bereits beachtliche Ergebnisse mit diesen erreicht werden konnten (Siami-Namini, Tavakoli, und Namin 2019). Bei der Auswertung der Prognosen zeigte sich jedoch, dass erst mit der Verwendung von CNN-basierten Architekturen nennenswerte Ergebnisse erreicht werden konnten. Schlussendlich konnte das CNN-LSTM-Modell die besten Prognoseergebnisse erreichen. Die Effektivität des kombinierten Ansatzes von CNN- und LSTM-Units, bei der Untersuchung von Zeitreihen, konnte bereits in verschiedenen Studien festgestellt werden (P. Li, Abdel-Aty, und Yuan 2020)(Kim und Cho 2019).

Es ist zu beachten, dass die Untersuchung der vorliegenden Bachelorarbeit unter der Betrachtung einer spezifischen Domäne erfolgt ist, weshalb die Frage offenbleibt, inwiefern sich die Ergebnisse in weiteren Untersuchungen reproduzieren lassen. Außerdem sollte der Lesende berücksichtigen, dass mithilfe dieser Arbeit lediglich eine oberflächliche Aussage darüber getroffen werden kann, inwiefern sich verschiedene Architekturen für die Zeitreihenanalyse eignen.

Dies ist damit zu begründen, dass zwar verschiedene Hyperparameter in der Modellierungsphase optimiert worden sind, jedoch sind für diverse aufgetretenen Probleme, bereits mögliche Lösungsansätze bekannt. Beispielsweise kann das Problem des explodierenden Gradienten, welches häufig bei der Verwendung von RNNs und langen Eingabesequenzen auftritt, mit einem Gradienten-Clipping minimiert werden (Zhang u. a. 2020). Möglicherweise könnte so die spezifischere Optimierung dafür sorgen, dass auch bei der Verwendung eines RNN nennenswerte Ergebnisse erzeugt werden könnten.

Weiterhin ist zu erwähnen, dass im Schritt der Modellierung für die Optimierung des Modells lediglich die Anpassung von Hyperparametern beschrieben worden ist. Im Laufe der Anfertigung der Bachelorarbeit wurden auch weitere Schritte zur Optimierung durchgeführt, wie die Regularisierung der Lernrate oder das Verwenden verschiedener Dropouts, um ein mögliches Overfitting zu verhindern. Allerdings konnten mittels dieser Schritte die Prognoseergebnisse nicht signifikant beeinflusst werden, weshalb auf eine ausführliche Beschreibung dieser Prozesse verzichtet worden ist.

## 7.2 Ausblick

Mit der Untersuchung konnten erste Erkenntnisse für die Prognose intralogistischer Auftragsdaten im Bezug zur Projektdomäne dargestellt werden. Die Analyse kann mit weiteren, in dieser Arbeit nicht betrachteten Ansätzen aus dem Bereich des Deep Learning fortgeführt werden. Vielversprechend dafür ist zum Beispiel die Verwendung einer Transformer-Architektur (Vaswani u. a. 2017). Ursprünglich wurden die Transformer für die Analyse natürlicher Sprache eingesetzt, jedoch konnten in jüngster Vergangenheit auch beachtliche Ergebnisse bei Zeitreihenanalysen erreicht werden (Zhou u. a. 2021).

Neben dem Einsatz neuer Analysemethoden kann auch das CNN-LSTM-Modell weiter optimiert werden. Ein Ansatz dafür kann die Steigerung der Iterationsschritte im Optimierungsprozess der Hyperparametern sein oder die Veränderung der Architektur an sich. Für die weitere Evaluierung der Ergebnisse wäre eine Überführung des erstellten Modells in den Live-Betrieb ratsam. Durch die Verbesserung des aktuell genutzten Prognosewertes erschließt sich auch die Frage für weiterführende Forschung, inwiefern sich andere Zeitreihen in der Logistik mit dem Einsatz von Deep-Learning-Modellen prognostizieren lassen.

### 7.3 Fazit

Ziel der vorliegenden Bachelorarbeit war es, ein Modell zu erstellen, welches auf Grundlage bisher gesammelter Daten die zukünftig zu erwartenden intralogistischen Auftragsdaten prognostizieren kann. Mit dem Abarbeiten verschiedener Prozessschritte, orientiert am Vorgehensmodell des CRISP-DM, und der Anwendung verschiedenster Methoden aus dem Bereich des Deep Learnings konnte schlussendlich ein Modell erstellt werden, welches in der Lage war, die zukünftigen sieben Zeitschritte mit einer mittleren prozentualen Abweichung von 9 % vorherzusagen.

Damit konnte der Prognosefehler im Vergleich zu der aktuell in der Praxis verwendeten Prognose halbiert werden. Unter der Betrachtung der Ergebnisse, konnte die Problemstellung damit erfolgreich bearbeitet werden. Außerdem konnte die Fragestellung, inwiefern eine Prognose intralogistischer Auftragsdaten mithilfe von Deep-Learning-Modellen einen Mehrwert bietet, für die spezifischen Projektomäne beantwortet werden.

Für weitere Forschung können andere Ansätze untersucht werden, mit welchen eventuell eine weitere Verbesserung der Ergebnisse möglich ist. Außerdem können bei der Bearbeitung von verwandten Problemstellungen auf die erlangten Erkenntnisse zurückgegriffen werden, um so auch die Prognosegenauigkeit anderer Modelle zu verbessern.

Der Code, welcher für die Bewertung, Analyse und Evaluierung der Daten erstellt worden ist, kann im folgende-Repository eingesehen werden: [https://github.com/lukas328/BA\\_TimeSeriesAnalyse](https://github.com/lukas328/BA_TimeSeriesAnalyse). Dazu ist zu erwähnen, dass es sich bei der bereitgestellten Datengrundlage um zufällige Werte handelt, da in der Bachelorarbeit vertrauliche betriebsinterne Daten untersucht worden sind. Weshalb bei einer Reproduktion der Analyseergebnisse abweichende Ergebnisse zu erwarten sind.

## Literaturverzeichnis

- Aggarwal, Charu C. 2017. „An Introduction to Outlier Analysis“. In *Outlier Analysis*, herausgegeben von Charu C. Aggarwal, 1–34. Cham: Springer International Publishing. [https://doi.org/10.1007/978-3-319-47578-3\\_1](https://doi.org/10.1007/978-3-319-47578-3_1).
- Aha, David W., Dennis Kibler, und Marc K. Albert. 1991. „Instance-Based Learning Algorithms“. *Machine Learning* 6 (1): 37–66. <https://doi.org/10.1007/BF00153759>.
- Alom, Md. Zahangir, Tarek Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Nasrin, Mahmudul Hasan, Brian Essen, Abdul Awwal, und Vijayan Asari. 2019. „A State-of-the-Art Survey on Deep Learning Theory and Architectures“. *Electronics* 8 (März): 292. <https://doi.org/10.3390/electronics8030292>.
- Alpaydin, Ethem. 2020. *Introduction to Machine Learning, Fourth Edition*. MIT Press.
- Backhaus, Klaus, Bernd Erichson, Wulff Plinke, und Rolf Weiber. 2018. „Zeitreihenanalyse“. In *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*, herausgegeben von Klaus Backhaus, Bernd Erichson, Wulff Plinke, und Rolf Weiber. Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-662-56655-8\\_3](https://doi.org/10.1007/978-3-662-56655-8_3).
- Badillo, Solveig, Balazs Banfai, Fabian Birzele, Iakov Davydov, Lucy Hutchinson, Tony Kam-Thong, Juliane Siebourg-Polster, Bernhard Steiert, und Jitao David Zhang. 2020. „An Introduction to Machine Learning“. *Clinical Pharmacology & Therapeutics* 107 (März). <https://doi.org/10.1002/cpt.1796>.
- Bai, Shaojie, J. Zico Kolter, und Vladlen Koltun. 2018. „Convolutional Sequence Modeling Revisited“, Februar. <https://openreview.net/forum?id=rk8wKk-R->.
- Bergstra, James, und Yoshua Bengio. 2012. „Random Search for Hyper-Parameter Optimization“. *Journal of Machine Learning Research* 13 (10): 281–305.
- Cavazza, Jacopo, und Vittorio Murino. 2016. „Active Regression with Adaptive Huber Loss“. *arXiv:1606.01568 [cs]*, Juni. <http://arxiv.org/abs/1606.01568>.
- Chai, T., und R. R. Draxler. 2014. „Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?“. *Numerical Methods*. <https://doi.org/10.5194/gmdd-7-1525-2014>.
- Chapman, P., J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, und R. Wirth. 2000. „CRISP-DM 1.0: Step-by-Step Data Mining Guide“.
- Daumé III, Hal, und Daniel Marcu. 2009. „Learning as Search Optimization: Approximate Large Margin Methods for Structured Prediction“. *arXiv:0907.0809 [cs]*, Juli. <http://arxiv.org/abs/0907.0809>.
- Deng, Li. 2014. „Deep Learning: Methods and Applications“. *Foundations and Trends® in Signal Processing* 7 (3–4): 197–387. <https://doi.org/10.1561/2000000039>.
- Eckstein, Peter P. 1999. „Zeitreihenanalyse“. In *Angewandte Statistik mit SPSS: Praktische Einführung für Wirtschaftswissenschaftler*, herausgegeben von Peter P. Eckstein, 273–328. Wiesbaden: Gabler Verlag. [https://doi.org/10.1007/978-3-322-92004-1\\_7](https://doi.org/10.1007/978-3-322-92004-1_7).

- Feng, Weijiang, Naiyang Guan, Yuan Li, Xiang Zhang, und Zhigang Luo. 2017. „Audio visual speech recognition with multimodal recurrent neural networks“. In , 681–88. <https://doi.org/10.1109/IJCNN.2017.7965918>.
- Frick, Detlev, Andreas Gadatsch, Jens Kaufmann, Birgit Lankes, Christoph Quix, Andreas Schmidt, und Uwe Schmitz, Hrsg. 2021. *Data Science: Konzepte, Erfahrungen, Fallstudien und Praxis*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-33403-1>.
- Géron, Aurélien. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc.
- Gitzel, R. 2016. „Data Quality in Time Series Data: An Experience Report“. In *CBI*.
- Goodfellow, Ian, Yoshua Bengio, und Aaron Courville. 2016. *Deep Learning*. MIT Press.
- Han, Jiawei, Jian Pei, und Micheline Kamber. 2011. *Data Mining: Concepts and Techniques*. Elsevier.
- Hassan, Hassan, Abdelazim Negm, Mohamed Zahran, und Oliver Saavedra. 2015. „ASSESSMENT OF ARTIFICIAL NEURAL NETWORK FOR BATHYMETRY ESTIMATION USING HIGH RESOLUTION SATELLITE IMAGERY IN SHALLOW LAKES: CASE STUDY EL BURULLUS LAKE.“ *International Water Technology Journal* 5 (Dezember).
- Helfert, M. 2001. „Datenqualitätsmanagement“. St. Gallen: Institut für Wirtschaftsinformatik.
- Hill, Tim, Marcus O'Connor, und William Remus. 1996. „Neural Network Models for Time Series Forecasts“. *Management Science* 42 (7): 1082–92. <https://doi.org/10.1287/mnsc.42.7.1082>.
- Hinton, Geoffrey E., Simon Osindero, und Yee-Whye Teh. 2006. „A Fast Learning Algorithm for Deep Belief Nets“. *Neural Computation* 18 (7): 1527–54. <https://doi.org/10.1162/neco.2006.18.7.1527>.
- Hochreiter, Sepp, und Jürgen Schmidhuber. 1997. „Long Short-Term Memory“. *Neural Computation* 9 (8): 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Hopfield, J. J. 1982. „Neural Networks and Physical Systems with Emergent Collective Computational Abilities“. *Proceedings of the National Academy of Sciences* 79 (8): 2554–58. <https://doi.org/10.1073/pnas.79.8.2554>.
- Hyndman, Rob J., und George Athanasopoulos. 2021. *Forecasting: Principles and Practice*. OTexts.
- Jordan, Jeremy. 2018. „Setting the Learning Rate of Your Neural Network.“ JeremyJordan. 2. März 2018. <https://www.jeremyjordan.me/nn-learning-rate/>.
- Kim, Tae-Young, und Sung-Bae Cho. 2019. „Predicting Residential Energy Consumption Using CNN-LSTM Neural Networks“. *Energy* 182 (September): 72–81. <https://doi.org/10.1016/j.energy.2019.05.230>.

- Kingma, Diederik P., und Jimmy Ba. 2017. „Adam: A Method for Stochastic Optimization“. *arXiv:1412.6980 [cs]*, Januar. <http://arxiv.org/abs/1412.6980>.
- Kühnnapfel, Jörg B. 2015. „Prognosen für Eilige: Naive Forecasts und ihre Rolle als Richtmaß“. In *Vertriebsprognosen: Methoden für die Praxis*, herausgegeben von Jörg B. Kühnnapfel, 166–71. Wiesbaden: Springer Fachmedien. [https://doi.org/10.1007/978-3-658-03002-5\\_7](https://doi.org/10.1007/978-3-658-03002-5_7).
- LeCun, Yann, Patrick Haffner, Léon Bottou, und Yoshua Bengio. 1998. „Object Recognition with Gradient-Based Learning“. In *Shape, Contour and Grouping in Computer Vision*, herausgegeben von David A. Forsyth, Joseph L. Mundy, Vito di Gesù, und Roberto Cipolla, 319–45. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. [https://doi.org/10.1007/3-540-46805-6\\_19](https://doi.org/10.1007/3-540-46805-6_19).
- Li, Pei, Mohamed Abdel-Aty, und Jinghui Yuan. 2020. „Real-Time Crash Risk Prediction on Arte-rials Based on LSTM-CNN“. *Accident Analysis & Prevention* 135 (Februar): 105371. <https://doi.org/10.1016/j.aap.2019.105371>.
- Li, Yaguang, Rose Yu, Cyrus Shahabi, und Yan Liu. 2018. „Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting“. *ArXiv:1707.01926 [Cs, Stat]*, Februar. <http://arxiv.org/abs/1707.01926>.
- Mariscal, Gonzalo, Oscar Marbán, und Covadonga Fernández. 2010. „A survey of data mining and knowledge discovery process models and methodologies“. *Knowledge Eng. Review* 25 (Juni): 137–66. <https://doi.org/10.1017/S0269888910000032>.
- Mills, Kathy. 2017. „What are the threats and potentials of big data for qualitative research?“ *Qualitative Research* 18 (November): 146879411774346. <https://doi.org/10.1177/1468794117743465>.
- Mitchell, Tom Michael. 1997. *Machine Learning*. McGraw-Hill Education.
- Mafeni Mase, Jimiama, Grazziela Figueredo, Mercedes Torres Torres, und Peter Chapman. 2020. „A Hybrid Deep Learning Approach for Driver Distraction Detection“. In . <https://doi.org/10.1109/ICTC49870.2020.9289588>.
- Pratama, Irfan, Adhistya Erna Permanasari, Igi Ardiyanto, und Rini Indrayani. 2016. „A review of missing values handling methods on time-series data“. In *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, 1–6. <https://doi.org/10.1109/ICITSI.2016.7858189>.
- Provost, Foster, und Tom Fawcett. 2013. „Data Science and its Relationship to Big Data and Data-Driven Decision Making“. *Big Data* 1 (1): 51–59. <https://doi.org/10.1089/big.2013.1508>.
- Qin, Yao, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, und Garrison Cottrell. 2017. „A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction“. *arXiv:1704.02971 [cs, stat]*, August. <http://arxiv.org/abs/1704.02971>.
- Quinlan, J. R. 1993. „Combining Instance-Based and Model-Based Learning“. In , 236–43. Morgan Kaufmann.
- Ramachandran, Prajit, Barret Zoph, und Quoc V. Le. 2017. „Searching for Activation Functions“. *arXiv:1710.05941 [cs]*, Oktober. <http://arxiv.org/abs/1710.05941>.

- Reinsel, David, John Gantz, und John Rydning. 2018. „The Digitization of the World from Edge to Core“, 28.
- Rolnick, David, Andreas Veit, Serge Belongie, und Nir Shavit. 2018. „Deep Learning is Robust to Massive Label Noise“. *arXiv:1705.10694 [cs]*, Februar. <http://arxiv.org/abs/1705.10694>.
- Rosenblatt, F. 1958. „The perceptron: A probabilistic model for information storage and organization in the brain“. *Psychological Review* 65 (6): 386–408. <https://doi.org/10.1037/h0042519>.
- Ruder, Sebastian. 2017. „An overview of gradient descent optimization algorithms“. *arXiv:1609.04747 [cs]*, Juni. <http://arxiv.org/abs/1609.04747>.
- Rumelhart, D., Geoffrey E. Hinton, und Ronald J. Williams. 1986. „Learning representations by back-propagating errors“. *Nature*. <https://doi.org/10.1038/323533a0>.
- Saxena, Shubh. 2017. „Artificial Neuron Networks(Basics) | Introduction to Neural Networks“. Medium. 27. Oktober 2017. <https://becominghuman.ai/artificial-neuron-networks-basics-introduction-to-neural-networks-3082f1dcca8c>.
- Schulz, Michael, Uwe Neuhaus, Jens Kaufmann, Daniel Badura, Ulrich Kerzel, Felix Welter, Maik Prothmann, u. a. 2020. *DASC-PM v1.0 - Ein Vorgehensmodell für Data-Science-Projekte*. <https://doi.org/10.25673/32872.2>.
- Sharma, Sagar. 2017. „Activation Functions in Neural Networks“. Medium. 6. September 2017. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, und Wang-chun Woo. 2015. „Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting“. *arXiv:1506.04214 [cs]*, September. <http://arxiv.org/abs/1506.04214>.
- Siami-Namini, Sima, Neda Tavakoli, und Akbar Siami Namin. 2019. „The Performance of LSTM and BiLSTM in Forecasting Time Series“. In *2019 IEEE International Conference on Big Data (Big Data)*, 3285–92. <https://doi.org/10.1109/BigData47090.2019.9005997>.
- Smith, Leslie N. 2018. „A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay“. *arXiv:1803.09820 [cs, stat]*, April. <http://arxiv.org/abs/1803.09820>.
- Stoller, Daniel, Mi Tian, Sebastian Ewert, und Simon Dixon. 2019. „Seq-U-Net: A One-Dimensional Causal U-Net for Efficient Sequence Modelling“. *arXiv:1911.06393 [cs, eess, stat]*, November. <http://arxiv.org/abs/1911.06393>.
- Stinson M, Wehking K. 2012. „Leistungsbewertung und -optimierung in der manuellen Kommissionierung.“ *Logistics Journal*.

- Sutskever, Ilya, Oriol Vinyals, und Quoc V Le. 2014. „Sequence to Sequence Learning with Neural Networks“. In *Advances in Neural Information Processing Systems*. Bd. 27. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html>.
- Suzuki, Kenji, Hrsg. 2011. *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. InTech. <https://doi.org/10.5772/644>.
- Svozil, Daniel, Vladimír Kvasnicka, und Jiří Pospichal. 1997. „Introduction to Multi-Layer Feed-Forward Neural Networks“. *Chemometrics and Intelligent Laboratory Systems* 39 (1): 43–62. [https://doi.org/10.1016/S0169-7439\(97\)00061-0](https://doi.org/10.1016/S0169-7439(97)00061-0).
- Turrado García, Fernando, Luis Javier García Villalba, und Javier Portela. 2012. „Intelligent System for Time Series Classification Using Support Vector Machines Applied to Supply-Chain“. *Expert Systems with Applications* 39 (12): 10590–99. <https://doi.org/10.1016/j.eswa.2012.02.137>.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, und Illia Polosukhin. 2017. „Attention Is All You Need“. *arXiv:1706.03762 [cs]*, Dezember. <http://arxiv.org/abs/1706.03762>.
- Wu, Sifan, Xi Xiao, Qianggang Ding, P. Zhao, Ying Wei, und Junzhou Huang. 2020. „Adversarial Sparse Transformer for Time Series Forecasting“. In *NeurIPS*.
- Xu, Sean, Man-Wai Mak, und Chi-Chung Cheung. 2017. „Deep neural networks versus support vector machines for ECG arrhythmia classification“. In , 127–32. <https://doi.org/10.1109/ICMEW.2017.8026250>.
- Yu, Rose, Stephan Zheng, Anima Anandkumar, und Yisong Yue. 2019. „Long-Term Forecasting Using Higher Order Tensor RNNs“. *ArXiv:1711.00073 [Cs]*, August. <http://arxiv.org/abs/1711.00073>.
- Zhang, Jingzhao, Tianxing He, Suvrit Sra, und Ali Jadbabaie. 2020. „Why gradient clipping accelerates training: A theoretical justification for adaptivity“. *arXiv:1905.11881 [cs, math]*, Februar. <http://arxiv.org/abs/1905.11881>.
- Zhou, Haoyi, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, und Wancai Zhang. 2021. „Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting“. *arXiv:2012.07436 [cs]*, März. <http://arxiv.org/abs/2012.07436>.
- Zimmermann, Volker. o. J. „KfW Digitalisierungsbericht Mittelstand 2020 – Rückgang der Digitalisierungsaktivitäten vor Corona, ambivalente Entwicklung während der Krise“,

## Anhang

Anhang 1 Experteninterview Leitfaden.....	XIII
Anhang 2 Experteninterview J. S .....	XIV
Anhang 3 Experteninterview O. H .....	XV
Anhang 4 Darstellung Bestellprozess.....	XVI
Anhang 5 Aufbau Datenbank .....	XVII
Anhang 6 Vorstudie, Aufbereitung und Qualitätsprüfung von Multivariaten Zeitreihen.....	XVIII
Anhang 7 Heatmap, Bestellverhalten in Abhängigkeit der Woche im Monat .....	XIX
Anhang 8 Boxplot-Auftragsdaten.....	XX
Anhang 9 Liniendiagramm, Kommissionierte Filialen .....	XXI
Anhang 10 Liniendiagramm Eingelagerte TE.....	XXII
Anhang 11 Liniendiagramm Lagerkapazitäten .....	XXIII
Anhang 12 Modell Evaluation, Ausreißer über gesamten Datensatz entfernt .....	XXIV
Anhang 13 Modell Evaluation, Chronologische Sequenz von Zeitschritten als Feature .....	XXV
Anhang 14 Aufbau Sequenzielles CNN .....	XXVI
Anhang 15 Aufbau Sequenzielles LSTM.....	XXVII
Anhang 16 Aufbau Sequenzielles Mehrschichtiges-LSTM .....	XXVIII
Anhang 17 Aufbau Sequenzielles Bidirektionales-LSTM .....	XXIX
Anhang 18 Aufbau Sequenzielles CNN-LSTM .....	XXX
Anhang 19 Aufbau Sequenzielles Encoder-Decoder-LSTM .....	XXXI
Anhang 20 Aufbau Sequenzielles Encoder-Decoder-LSTM-CNN .....	XXXII
Anhang 21 Aufbau Sequenzielles Encoder-Decoder-ConvLstm .....	XXXIII
Anhang 22 Ergebnisse Sequenzielle Modelle Trainingsgrundlage: Historische Auftragsdaten....	XXXIV
Anhang 23 Ergebnisse Sequenzielle Modelle Trainingsgrundlage: Weitere Datensätze .....	XXXV
Anhang 24 Darstellung Optimierungsprozess.....	XXXVI
Anhang 25 Aufbau Multi-Head-3-CNN.....	XXXVII
Anhang 26 Aufbau Mutli-Head-3 CNN-LSTM .....	XXXVIII
Anhang 27 Aufbau Multi-Head-3-ConvLSTM .....	XXXIX
Anhang 28 Aufbau Multi-Head-5-CNN.....	XL
Anhang 29 Aufbau Mutli-Head-5-ConvLSTM .....	XLI
Anhang 30 Aufbau Mutli-Head-5-CNN-ConvLSTM .....	XLII
Anhang 31 Report Evaluierungsdatensatz Pro-Label.....	XLIII
Anhang 32 Report Evaluierungsdatensatz Gesamt .....	XLIV
Anhang 33 Ergebnisse Testdatensatz .....	XLV
Anhang 34 Prognosen Dezember-2021.....	XLVI
Anhang 35 Bereitstellungs-Report .....	XLVII

Anhang 1  
Experteninterview Leitfaden

## **Interviewleitfaden**

### **Deep Learning für Zeitreihenvorhersage - Multi-step ahead Prognose von intralogistischen Auftragsdaten**

#### **Interesse:**

Das Ziel der Arbeit ist es, eine Zeitreihe, welche Information zu intralogistischen Auftragsdaten beinhaltet, zu untersuchen und anhand der vorliegenden historischen Daten eine möglichst genaue Prognose für den weiteren Verlauf der Zeitreihe zu erstellen. Anschließend können anhand der Analyseergebnisse Handlungsempfehlungen erstellt, oder Prozesse angepasst werden.

#### **Forschungsfrage:**

Inwiefern lassen sich die zukünftigen erfassten die intralogistische Auftragsdaten prognostizieren, mit der Bezugnahme zu Methoden aus dem Bereich des Deep-Learning.

#### **Inhalt:**

- Fragen zur Person und Unternehmen
- Fragen zur Lieferkette und Erfassung der Auftragsdaten
- Fragen zu einer möglichen Prognose der Auftragsdaten

#### **Hinweise zum Vorgehen und Datenschutz:**

- Das Interview besteht aus 9 Hauptfragen und 28 optionalen Nachfragen.
- Die voraussichtliche Dauer des Interviews beträgt 30-45 Minuten.
- Eine Einwilligungserklärung zur Verwendung des Interviews ist angehängt.

## **Block 1: Fragen zur Person und zum Unternehmen**

In welcher Abteilung arbeiten Sie und was genau ist ihr Verantwortungsbereich?

- Wie lange sind Sie schon in der Logistik tätig
- Wie lange Arbeiten sie bereits für Budnikowsky
- Für welche Prozesse in der Logistik sind Sie mitverantwortlich

Was genau ist ihr Tätigkeitsbereich in der Logistik?

- Wie nehmen Sie Einfluss darauf
- Welche Abteilung sind davon betroffen

Wie ist das Unternehmen Budnikowsky aufgebaut?

- Wie viele Filialen besitzt Budnikowsky ca. zum aktuellen Zeitpunkt
- In welchen Raum ist Budnikowsky ansässig

## **Block 2: Fragen zur Lieferkette und Erfassung der Auftragsdaten**

Können Sie kurz den Prozess, von der Bestellung einer Filiale bis hin zur Auslieferung der Ware beschreiben?

- Wie setzt sich die Bestellung einer Filiale zusammen
- Laufen alle Bestellung über ein Logistikzentrum
- Welche Prozesse sind innerhalb des Logistikzentrums dafür notwendig
- In welcher Frequenz erhält eine Filiale Ware
- Wie lange dauert der Prozess von Bestellung bis hin zum Erhalt der Ware

Inwiefern beeinflusst die Höhe der erfassten Auftragsdaten Ihre Arbeit?

- Welche Prozesse werden davon beeinflusst
- Werden Entscheidung Anhand der Höhe Auftragsdaten getroffen
- Inwiefern sind Ihre Tätigkeiten davon langfristig oder kurzfristig betroffen

## **Block 3 Fragen zu einer möglichen Prognose der Auftragsdaten**

Falls die Höhe der erfassten Auftragsdaten ihre Arbeit beeinflusst, inwiefern probieren Sie diese vorherzusagen?

- Wie ist die Prognose aufgebaut
- Werden statistische Methoden für die Prognose herangezogen
- Inwiefern wird die Prognose durch Erfahrungswerte beeinflusst
- Welche Kennzahlen werden in der Prognose berücksichtigt
- Wie präzise waren die Prognosen in der Vergangenheit

Wie sich eine Präzise Prognose der Auftragsdaten auf ihre aktuelle Arbeit auswirken?

- Könnten Prozesse langfristig angepasst oder optimiert werden
- Welche kurzfristigen Änderungen würden sich aus der Vorhersage ergeben

Wie müsste die Prognose aufgebaut sein, damit sich ein Mehrwert generieren lassen würde?

- Welcher Zeitraum müsste vorhergesagt werden
- Müsste die Prognose in verschiedenen Teile unterteilt werden
- Wie detailliert müssten die Daten vorliegen
- In welcher Form müssten Ergebnisse bereitgestellt werden
- Was gilt es weiterhin zu beachten

Durch welche Faktoren werden Ihrer Meinung nach die Auftragsdaten beeinflusst?

- Stehen die genannten Kennzahlen zur Analyse zur Verfügung, Falls nicht ist es möglich diese Daten zu erfassen
- Können diese Daten verarbeitet werden

Anhang 2  
Experteninterview J. S

## **Interview 1 – mit J. S. (J)**

### **Block 1: Fragen zur Person und zum Unternehmen**

L: In welcher Abteilung arbeiten Sie und was genau ist ihr Verantwortungsbereich?

- J: Ich arbeite bei Budni und bin im Supply-Chain-Management Tätig, dabei bin ich Teamleiter von dem Bereichen Warenausgang und Leergut.

L: Wie lange sind Sie schon in der Logistik tätig?

- J: Bei Budni bin ich mittlerweile seit 7 Jahren angestellt und in der Logistik arbeite ich bestimmt schon seit über 14 Jahren.

L: Für welche Prozesse in der Logistik sind Sie mitverantwortlich?

- J: Für alle Prozesse die sich im Bereich des Warenausgangs und des Leerguts abspielen, sowie für die Planung der Warenauslieferung und damit auch für den Fuhrpark.

L: Was genau ist ihr Tätigkeitsbereich in der Logistik?

- J: In den Bereichen, für die ich verantwortlich bin, bestehen meine Tätigkeiten vor allem aus der Planung und Organisation, allerdings besteht auch ein Großteil der Arbeit aus der Optimierung von Arbeitsabläufen und Prozessen.

L: Wie viele Filialen besitzt Budnikowsky ca. zum aktuellen Zeitpunkt?

- J: Durch die Kooperation mit der Edeka, ist es schwer zu sagen und kommt auf die Betrachtungsweise an, allerdings werden auf jeden Fall ca. 200 Filialen vom Logistikzentrum versorgt.

L: In welchen Raum ist Budnikowsky ansässig?

- J: Der größte Teil der Filialen befindet sich in der Metropolregion Hamburgs, ein paar weitere Filialen sind allerdings auch in anderen Teilen von Deutschland zu finden. Zu nennen sind dabei auf jeden Fall die Standorte in Berlin, Hagenow und Sylt.

### **Block 2: Fragen zur Lieferkette und Erfassung der Auftragsdaten**

L: Können Sie kurz den Prozess, von der Bestellung einer Filiale bis hin zur Auslieferung der Ware beschreiben?

- J: Im ersten Schritt bekommt jede Filiale einen automatisch generierten Bestellvorschlag, falls notwendig kann der Bestellvorschlag noch von Kollegen aus der Filiale angepasst werden. Anschließend wird die Bestellung zu einer von uns definierten Zeit an das Logistikzentrum übermittelt, die Zeiten sind dabei entweder 18 oder 20 Uhr. Mithilfe der unterschiedlichen Bestelleingänge ist es uns möglich die Ressourcen im Lager besser zu verteilen. Sobald die Bestellungen eingegangen sind, werde diese vom SAP-System und anschließend vom Warenwirtschaftssystem verarbeitet. Das Warenwirtschaftssystem verteilt die eingegangenen Bestellungen auf verschiedene Batches, zu Erklärung ein Batch stellt eine Teilmenge der Gesamten Auftragslast dar. Alle Filialen, die um 18 Uhr bestellt haben, rutschen in den ersten Batch, wodurch die Aufträge zuerst kommissioniert werden und wir die Möglichkeit haben, Filialen unterschiedliche Prioritäten zuzuordnen.

Im Schnitt werden am Tag, 4 Batch kommissioniert. Der Prozess von der Verarbeitung der Bestellung bis hin zum zur Auslieferung der Ware nimmt in etwa 35std- 48std in Anspruch, dass variiert je nachdem in welchen Batch sich die Bestellung, der Filiale befindet.

Die Filialen bestellen im Schnitt 3,8-mal pro Woche, die Bestellhäufigkeit variiert unter den Filialen und liegt zwischen 2- und 6-mal pro Woche. Nachdem die Ware kommissioniert, worden ist, wird sie Warenausgang der bereitgestellt und der Filiale zugeteilt werden.

L: Inwiefern beeinflusst die Höhe der erfassten Auftragsdaten Ihre Arbeit?

- J: In meinen Tätigkeitsbereichen wirkt sich die Höhere der Bestellungen, vor allem auf die Planung des Fuhrparks aus.  
Sollten zum Beispiel extrem wenig Bestellungen ausgeliefert werden müssen, kann sich das negativ auf die Auslastung jedes LKWs auswirken, wodurch letztendlich viel Platz bei der Auslieferung verschenkt wird.  
Umgekehrt kann es genauso ein Problem sein, wenn zu viele Bestellungen ausgeliefert werden und nicht genug LKW-Fahrer zur Verfügung stehen.  
Neben der Auslastung des Fuhrparks können hohe Bestellzahlen auch dafür sorgen, dass der Verfügbare Raum im Warenausgang ausgelastet ist und kaum noch Platz für die Zwischenlagerung von Ware zur Verfügung steht.

L: Welche Entscheidungen werden Anhand der Höhe Auftragsdaten getroffen?

- J: Die Anzahl der benötigten LKWs orientiert sich an den erfasste Auftragsdaten, allerdings müsste es schon zu extremen Abweichungen kommen, damit sich bei der Planung entscheidende Veränderungen ergeben würden.

### **Block 3 Fragen zu einer möglichen Prognose der Auftragsdaten**

L: Falls die Höhe der erfassten Auftragsdaten ihre Arbeit beeinflusst, inwiefern probieren Sie diese vorherzusagen?

- J: Da Ich lediglich eine Prognose für die Planung des Fuhrparks benötige, prognostizieren wir nicht direkt die Bestellzahlen, sondern konzentrieren uns auf die Vorhersage der Versandeinheiten. Die unterschiedlichen Prognosen hängen zwar irgendwo zusammen, allerdings hat sich in der Vergangenheit gezeigt, dass man vom einen nicht direkt auf das schließen kann.

L: Werden statistische Methoden für die Prognose verwendet?

- J: Unsere Prognose stützt sich vor allem, auf die geplante Anzahl an Filialen, welche am jeweiligen Tag zu beliefern sind. Sollte wir uns in einem Saisongeschäft, wie Ostern oder Weinachten befinden, betrachten wir auch vermehrt die Durchschnittswerte, der Vergangen Jahre.

L: Inwiefern wird die Prognose durch Erfahrungswerte beeinflusst?

- J: Tendenziell fließen keine Erfahrungswerte in die Prognose hinein, allerdings wird meistens für den Fuhrpark ein Puffer miteinberechnet, um Engpässe vorzubeugen.

L: Wie präzise waren die Prognosen in der Vergangenheit?

- J: Schwierig zu sagen, da geringe Abweichung eigentlich keine Auswirkung auf die Planung hätte.

L: Wie würde sich eine Präzise Prognose, der Auftragsdaten auf ihre aktuelle Arbeit auswirken?

- J: Bedingt dadurch, dass wir unsere Prognose auf die Versandeinheiten konzentrieren, würde sich erstmal nichts großartig ändern. Wenn wir jedoch die Auftragsdaten zielsicher Vorhersagen können, kann sicherlich nochmal betrachtet werden, inwiefern unsere Prognose davon profitieren würde.

L: Durch welche Faktoren werden Ihrer Meinung nach die Auftragsdaten beeinflusst?

- J: Besonders bei uns in der Logistik spielt der Betrachtungszeitraum eine große Rolle, damit sind vor allem die unterschiedlichen Saisongeschäfte in Laufe eines Jahres gemeint. Vor allem Faktoren wie Ostern, Weinachten oder beginnende Schulferien sollten in der Analyse betrachtet werden. Ein weiterer Faktor sind sicherlich Feiertage, durch welche die Bestellung einer Filiale wegfällt und mit einem größeren Bestellvolumen in der nächsten Bestellung gerechnet werden kann. Darauf aufbauend spielt auch die Anzahl und Größe der Filialen, welche am Betrachteten Tag bestellen, eine Rolle. Durch welche weiteren Kennzahlen im Lager, sich die Auftragsdaten prognostizieren lassen, ist schwierig zu sagen, pauschal würde mir in meinen Verantwortungsbereichen allerdings keine einfallen.

## **Einverständniserklärung zur Erhebung und Verarbeitung von Interviewdaten**

Ich bin damit einverstanden, im Kontext der genannten Bachelorarbeit an der Befragung teilzunehmen. Darüber hinaus akzeptiere ich die o.g. Form der Weiterverarbeitung und wissenschaftlichen Verwertung des mit mir geführten Interviews und den daraus entstehenden Daten bzw. Informationen. Außerdem bestätige ich damit auch die Richtigkeit der Protokollierten Dialoge und den daraus entstandenen Daten.

Mir ist bekannt, dass ich diese Einwilligung jederzeit gegenüber Lukas Dech widerrufen kann, mit der Folge, dass die Verarbeitung meiner personenbezogenen Daten, nach Maßgabe der Widerrufserklärung, durch diesen für die Zukunft unzulässig wird. Dies berührt die Rechtmäßigkeit der aufgrund der Einwilligung bis zum Widerruf erfolgten Verarbeitung jedoch nicht.

Jakob Schuster

Vorname; Nachname in Druckschrift

3.2.22 Hamburg

Ort, Datum / Unterschrift

Anhang 3  
Experteninterview O. H

## **Interview 2 – mit O. H. (O)**

### **Block 1: Fragen zur Person und zum Unternehmen**

L: In welcher Abteilung arbeiten Sie und was genau ist ihr Verantwortungsbereich?

- H: Ich arbeite im Supply-Chain-Management bei Budni und bin Teamleiter der Kommissionierung.

L: Wie lange sind Sie schon in der Logistik tätig?

- H: In der Logistik arbeite ich seit 2013, bei Budni habe im September 2017 angefangen.

L: Für welche Prozesse in der Logistik sind Sie mitverantwortlich?

- H: Ich betreue alle Abläufe rund um die Kommissionierung, welche bei uns im Logistikzentrum in drei verschiedenen Lagerbereichen durchgeführt wird.

L: Was genau ist ihr Tätigkeitsbereich in der Logistik?

- H: Im Bereich der Kommissionierung kümmere ich mich vor allem um die Planung, Steuerung und Überwachung verschiedenster Abläufe, aber auch die Optimierung diverser Prozesse gehört zu meinen Tätigkeiten.

L: Welche Abteilung sind davon betroffen?

- H: Viele verschiedene Abläufe innerhalb der Supply-Chain hängen miteinander zusammen, weshalb das gar nicht so leicht zu beantworten ist. Aber sicherlich wirkt sich die Kommissionierung stark auf Abläufe im Warenausgang, Wareneingang, Einkauf oder in der Disposition aus.

### **Block 2: Fragen zur Lieferkette und Erfassung der Auftragsdaten**

L: Können Sie kurz den Prozess, von der Bestellung einer Filiale bis hin zur Auslieferung der Ware beschreiben?

- H: Es beginnt damit, dass jede Filiale einen automatisch generierten Bestellvorschlag bekommt, der dann noch manuell angepasst werden kann. Anschließend werden die Aufträge entweder um 18 oder 20 Uhr an Logistikzentrum übermittelt. Dafür durchlaufen die Daten verschieden SAP-System in welchen in erster Linie, die Verfügbarkeiten, sowie Stati der Artikel geprüft werden. Sollte alles passen, kann die Bestellungen von unserem Warenwirtschaftssystem verarbeitet werden. Dabei werden die unterschiedliche Bestellung in Batches unterteilt. Ein Batch stellt eine Teilmenge der gesamten Aufträge dar und ist bereits vor dem Einlaufen der Bestellungen geschrieben und definiert. Damit ist bereits klar, in welchem Batch eine Filiale kommissioniert wird.  
Anschließend kann mit der Kommissionierung und der anschließenden Bereitstellung, der Ware fortgefahrt werden. Mit der Fertigstellung eines Auftrages, bekommt jede Filiale ein Lieferschein, im welchen nochmal aufgelistet ist, was in der Bestellung enthalten sein sollte. Sollte es im Laufe des Prozesses zu Abschreibungen gekommen sein, kann der Lieferschein leicht von der ursprünglichen Bestellung abweichen.  
Der Prozess läuft in der Regel im 48 Stunden Rhythmus, damit wird beispielsweise Montag bestellt, Dienstag kommissioniert und Mittwoch ausgeliefert.

L: In welcher Frequenz erhält eine Filiale Ware?

- H: Im Schnitt wird jede Filiale 3,7-Mal die Woche beliefert

L: Inwiefern beeinflusst die Höhe der erfassten Auftragsdaten Ihre Arbeit?

- H: Vor allem die Personalplanung, in der Kommissionierung orientiert sich an den zu erwartenden Bestellzahlen. Dafür wird im ersten Schritt eine Jahresplanung aufgestellt, der dort geplante Personalbedarf, wird dann in der Vorwoche erneut bewertet und kann gegeben falls angepasst werden.

L: Inwiefern sind Ihre Tätigkeiten davon langfristig oder kurzfristig betroffen?

- H: Durch die Orientierung der Personalplanung an den zu erwartenden Auftragsdaten, sind vor allem Organisation und Prozessplanung betroffen.

### **Block 3 Fragen zu einer möglichen Prognose der Auftragsdaten**

L: Falls die Höhe der erfassten Auftragsdaten ihre Arbeit beeinflusst, inwiefern probieren Sie diese vorherzusagen?

- H: Ja, auf jeden Fall müssen wir für die Personalplanung, die Auftragsdaten in irgendeiner Form prognostizieren.

L: Wie ist die Prognose aufgebaut?

- Große Faktoren in der Prognose sind zum einen die Vorjahres Zahlen und Trends der letzten Wochen, aber auch die geplante Anzahl an zu kommissionierenden Filialen am jeweiligen Tage wird betrachtet.

L: Werden statistische Methoden für die Prognose herangezogen?

- H: vorwiegend wird auf historische Daten und Durchschnittswerte zurückgegriffen.

L: Inwiefern wird die Prognose durch Erfahrungswerte beeinflusst?

- H: Zum einen werden verschiedene Informationen betrachtet, welche uns bereits vorab bekannt sind, zum Beispiel kam es im Zuge der Coronapandemie immer wieder zu Sonderzuteilung, die wir natürlich bei der Prognose berücksichtigen müssen. Aber auch Bachgefühl und Erfahrungswerte aus den vergangenen Jahren spielen sicherlich eine Rolle.

L: Wie würde sich eine Präzisere Prognose der Auftragsdaten auf ihre aktuelle Arbeit auswirken?

- H: Eine präzise Vorhersage würde die Personal- und Einsatzplanung vereinfachen. Außerdem könnte der Aufbau von Überstunden verhindert werden und Aufgebaute Überstunden gezielt abgebaut werden, ohne das Engpässe entstehen würden.

L: Wie müsste die Prognose aufgebaut sein, damit sich ein Mehrwert generieren lassen würde?

- Die Auftragsdaten müssten pro Tag vorhergesagt werden, außerdem wäre für die Personalplanung eine Unterteilung der Lagerbereiche notwendig.  
Für die Einsatzplanung würde eine Prognose der Auftragsdaten, 1-2 Wochen im Voraus genügen. Um zum Beispiel die Urlaubsplanung daran zu orientieren, müssten eine Prognose für das gesamte Jahr vorliegen.

L: Durch welche Faktoren werden Ihrer Meinung nach die Auftragsdaten beeinflusst?

- H: Ich denke ältliche Faktoren nehmen Einfluss auf die Höhe der Auftragsdaten. Sicherlich besteht eine Korrelation zwischen den bereits gemessenen Auftragsdaten und den zukünftigen zu erwartenden Daten. Außerdem kann die Anzahl an Filialen, welche am entsprechenden Tag kommissioniert werden, ein Richtwert zu dem Daten liefern.  
Neben den gemessenen Kennzahlen sollten, aber auch Umwelteinflüsse wie die Jahreszeiten, Saisongeschäfte oder Schulferien nicht außeracht gelassen werden, in diesem Zusammenhang ist sicherlich auch Corona zu erwähnen, wodurch es in jüngster Vergangenheit immer wieder zu starken Schwankungen in Auftragsdaten kam.  
Ein weiterer Faktor kann auch die Entwicklung von Budni an sich sein, damit sind vor allem Neueröffnungen oder die Expansion in weitere Bereiche gemeint.

## **Einverständniserklärung zur Erhebung und Verarbeitung von Interviewdaten**

Ich bin damit einverstanden, im Kontext der genannten Bachelorarbeit an der Befragung teilzunehmen. Darüber hinaus akzeptiere ich die o.g. Form der Weiterverarbeitung und wissenschaftlichen Verwertung des mit mir geführten Interviews und den daraus entstehenden Daten bzw. Informationen. Außerdem bestätige ich damit auch die Richtigkeit der Protokollierten Dialoge und den daraus entstandenen Daten.

Mir ist bekannt, dass ich diese Einwilligung jederzeit gegenüber Lukas Dech widerrufen kann, mit der Folge, dass die Verarbeitung meiner personenbezogenen Daten, nach Maßgabe der Widerrufserklärung, durch diesen für die Zukunft unzulässig wird. Dies berührt die Rechtmäßigkeit der aufgrund der Einwilligung bis zum Widerruf erfolgten Verarbeitung jedoch nicht.

Oscar, Hauw

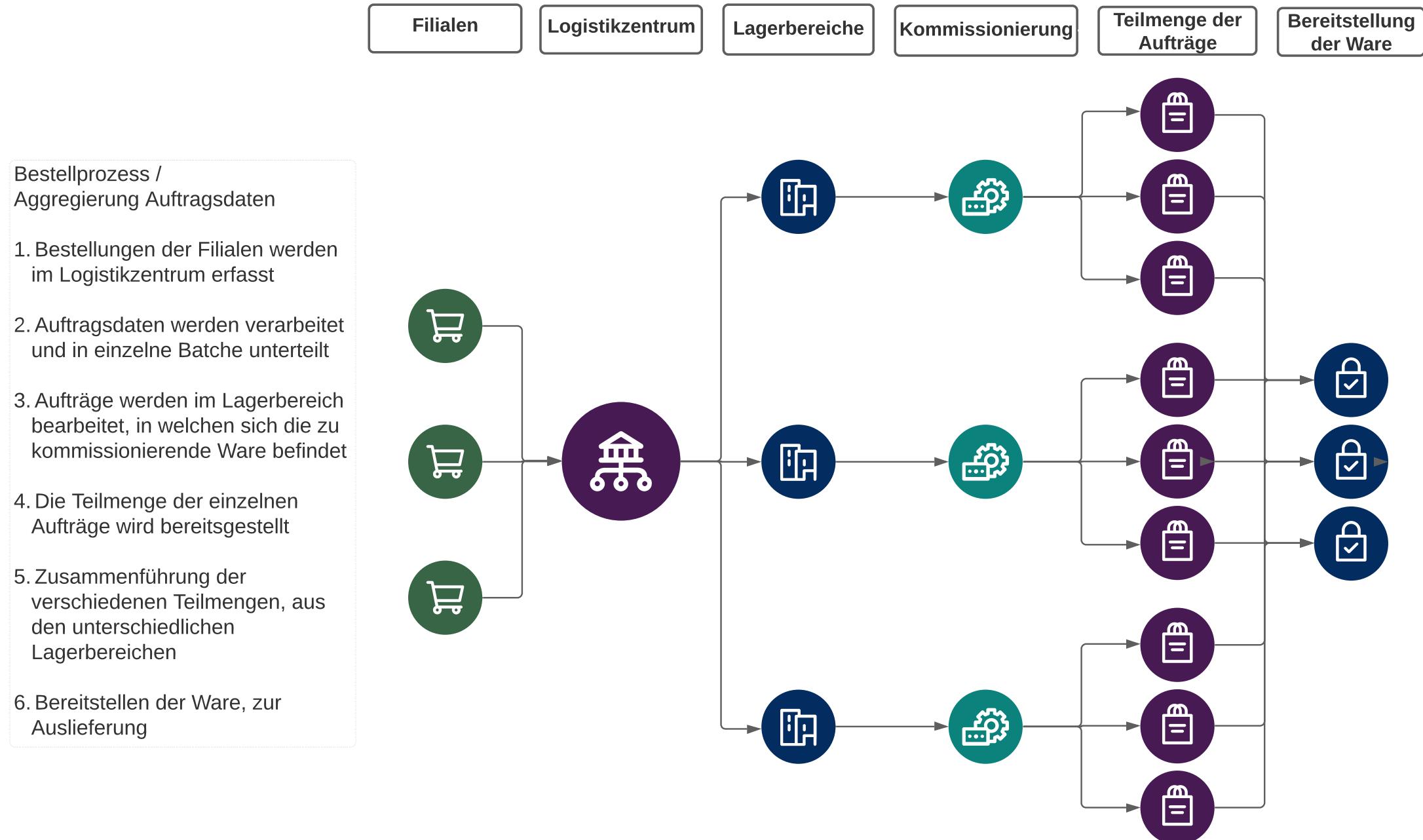
Vorname; Nachname in Druckschrift

Hamburg, 03.02.2022

Ort, Datum / Unterschrift



Anhang 4  
Darstellung Bestellprozess



Anhang 5  
Aufbau Datenbank

dbo.Batchzahlen_Gesamt
<b>Datum</b>
VPE_Gsamt
Komm_VPE
Voll
Displays
<b>Lager</b>

dbo.DPS_Lagerbelegung
<b>Datum</b>
Bereich
Belegt
Frei
Fehler
Gesperrt
Gesamt

dbo.We_Te_Eingelagert
<b>Datum</b>
Lager
TE

dbo.Filialen_Pro_Tag
<b>Datum</b>
<b>Filialen</b>
Lager

Anhang 6  
Vorstudie, Aufbereitung und Qualitätsprüfung  
von Multivariaten Zeitreihen



## Transferleistung Theorie/Praxis □

Matrikelnummer:	
Freigegebenes Thema:	
Studiengang, Zenturie:	

## Gliederung

Abbildungsverzeichnis .....	1
1. Einführung .....	2
2. Zeitreihen.....	3
3. Methodik .....	4
4. Projektdomäne.....	5
5. Datensatz .....	6
6. Ursprungsdatenquelle .....	7
7. Explorative Datenanalyse.....	9
8. Datenaufbereitung .....	13
9. Fazit .....	14
Literaturverzeichnis .....	15

## Abbildungsverzeichnis

Abbildung 1: Zeitreihen Komponenten.....	3
Abbildung 2: DASC-PM Daten .....	4
Abbildung 3: Daten Info.....	6
Abbildung 4: Daten Ausschnitt.....	6
Abbildung 5: Qualitätsmerkmale Daten.....	7
Abbildung 6: Daten Kennzahlen.....	9
Abbildung 7: Liniendiagramm gesamter Datensatz .....	10
Abbildung 8: Histogramm gesamter Datensatz.....	10
Abbildung 9: Liniendiagramm 100 Tage.....	10
Abbildung 10: Box-Plot Wochentage .....	11
Abbildung 11: Heat-Map Wochentage .....	11
Abbildung 12: Box-Plot Wochen .....	12
Abbildung 13: Heat-Map Monate .....	12

## 1. Einführung

Durch die Voranschreitende Digitalisierung, der unverzichtbare Einsatz von Maschinen und eine ausgebauten IT-Infrastruktur, wird die gesammelte und gespeicherte Datenmenge von Jahr zu Jahr größer (Cisco Systems, 2018). Daten stellen Fakten da, anhand dessen können datengetriebene Entscheidungen basierend auf Fakten getroffen werden. Mit der steigernden Relevanz von Daten und deren Verarbeitung im wirtschaftlichen Kontext wird das Sammeln und Auswerten von Daten zu einer essenziellen Aufgabe im Unternehmen.

Durch die Durchführung des Dualen Studiums in der Logistik, kam ich in den vergangenen Praxisphasen immer wieder in Kontakt mit Projekten, bei welchen Daten und die Auswertung dieser eine Zentralrolle spielten. Im Unternehmen werden häufig die gesammelten Daten mithilfe Methoden der deskriptiven Statistik ausgewertet und analysiert. Daraufhin können anhand der Ergebnisse, Schlüsse über zukünftige Ereignisse oder den Ist-Zustand getroffen werden. Die Korrektheit der Vorhersage oder die Auswertung des Ist-Zustandes kann maßgeblichen Einfluss auf die Effiziente Gestaltung von Prozessen sowie die Mitarbeiterplanung haben. Da die Vorhersage aktuell auf statistischen Methoden und gesammelten Erfahrungswerten beruht, wird die bevorstehende Bachelorarbeit dafür genutzt den Prozess zu erweitern und zukünftigen Ereignissen mithilfe von Analysemethoden aus dem Bereich des Machine-Learning vorherzusagen. Da ein Großteil der Data-Science Projekte aufgrund der Datengrundlage nicht umgesetzt wird oder die Projekte bedingt durch die geringe Aussagekraft der Daten zwangsweise abgebrochen werden müssen (Gartner, 2015), wird die vorliegende Transferleistung als Vorstudie dienen, in welcher die ausgewählten Daten auf ihre Qualität geprüft werden außerdem wird eine erste Explorativen Datenanalyse durchgeführt. Anschließend können anhand der Ergebnisse weitere Schritte zur Datenaufbereitung definiert werden.

Das Duale Studium wird bei den in Hamburg ansässigen Drogeriehändler Budnikowsky durchgeführt, mit dem Schwerpunkt in der Logistik. In der Bachelorarbeit wird das Problem der Vorhersage von Bestellzahlen behandelt. Jede Bestellung muss im Nachgang kommissioniert und zur Auslieferung Bereitgestellte werden. Damit hängt die Mitarbeiterplanung maßgeblich von den täglich einkauften Bestellzahlen ab. Die Bestellzahlen wurden bereits in der Vergangenheit über einen längeren Zeitraum gesammelt und können somit als Datengrundlage für die Modellierung dienen.

Zu Beginn der Arbeit werden die Grundzüge von Zeitreihen dargestellt und erläutert, um das nötige Grundverständnis zu von diesen zugewinnen. Daraufhin wird der Datensatz mit dem vorhandenen Informationen betrachtet. Anschließend werden die ersten Schritte eines Data-Science Projektes detailliert durchlaufen und erklärt. In den ersten Schritten wird der Prozess hinter den Bestellung erläutert damit das nötige Domänenwissen für die Durchführung der Analyse vorhanden ist. Danach wird die Qualität der Daten bewertet, die Betriebsinterne Infrastruktur geprüft sowie mithilfe einer Explorativen Datenanalyse, Einblicke in die Daten gewonnen. Anhand der Ergebnisse können weiter Schritte zur Aufbereitung des Datensatzes definiert werden, welche die Daten für die eigentlich Datenanalyse vorbereiten.

## 2. Zeitreihen

Als Zeitreihe werden alle Datensätze beschrieben, die gewonnen werden indem eine Reihe von Variablen in regelmäßigen Abständen gemessen wird. Viele Ereignisse die wiederholt auftreten lassen sich ein Zeitpunkt zuordnen, damit sind Zeitreihen allgewärtig und stellen auch im wirtschaftlichen Kontext ein wichtiges Werkzeug dar. Eines der wichtigsten Ziele, bei der Analyse von Zeitreihen, ist die Vorhersage von zukünftigen Werten. Durch die Vorhersage von zukünftigen Werten, kann die Abnahme oder Zunahme des betrachteten Wertes frühzeitig erkannt werden und Prozesse entsprechend angepasst werden. Damit ist die Fähigkeit eine Zeitreihe mit hoher Präzision Vorhersagen ein wichtiger Faktor im Unternehmen wie auch im Wissenschaftlichen Kontext sein. (Kreiß, 2006)

Allgemein wird davon ausgegangen das Zeitreihen in regelmäßigen Abständen erzeugt werden, sollte dies nicht der Fall sein wird von einer unregelmäßigen Zeitreihe gesprochen. Weiterhin können Zeitreihen anhand der betrachteten Variablen, welche sich im untersuchten Zeitraum verändert haben, unterschieden werden. Verändert sich nur eine Variable im betrachteten Zeitraume, wird von einer univariaten Zeitreihe gesprochen. Eine univariante Zeitreihe liegt häufig bei Sensordaten vor, da diese zu jedem Zeitpunkt einen eindimensionalen Wert erzeugen. Sollten sich mehrere Werte über den betrachteten Zeitraum verändern, wird von einer multivariaten Zeitreihe gesprochen. Schlussfolgend daraus ist eine multivariate Zeitreihe, die Betrachtung mehrerer univariater Zeitreihen im gleichen Zeitraum, allerdings stehen die multivariaten Zeitreihen im Verhältnis zueinander, wodurch die eigentliche Aussagekraft der Zeitreihe von mehreren Parametern abhängt. (Kreiß, 2006)

Die Observierte Zeitreihe lässt sich in verschiedene Komponenten aufteilen (vgl. Abbildung 1).

- ein Trend beschreibt die langfristige Veränderung des Mittleren Niveaus, damit zeichnet sich bei der Trend Analyse das Lineare Fallen oder Steigen der Zeitreihe ab.
- Saisonalitäten beschreiben das wiederholende Auftreten von ähnlichen Mustern, entlang der Zeitreihe.
- Außerdem können Zeitreihen durch zufällige Ereignisse geprägt worden sein. Wodurch ein unstrukturiertes Muster entsteht, welches als Rauschen bezeichnet wird und kaum bis gar nicht vorhersagbar ist.

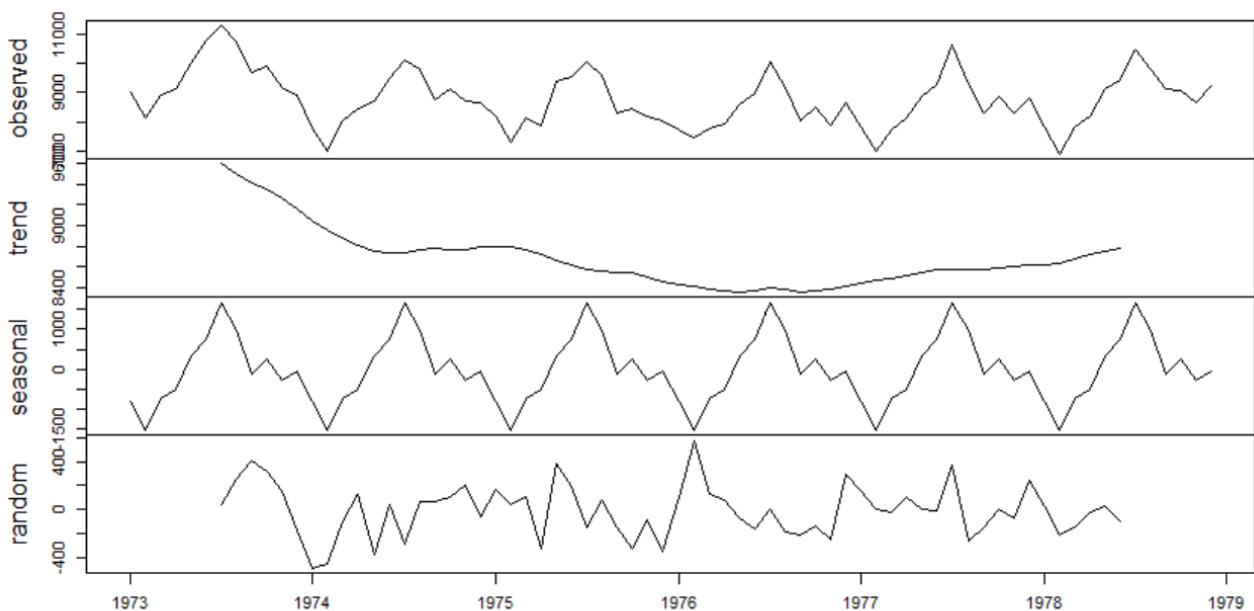


Abbildung 1: Zeitreihen Komponenten (Novustat, 2019)

### 3. Methodik

Für die Analyse der ausgewählten Zeitreihe, wird sich an dem DASC-PM v1.0 orientiert. Der DASC-PM ist ein Vorgehensmodell für Data-Science Projekte, welches 2020 von einer Arbeitsgruppe der Nordakademie entworfen wurde. (Neuhaus, 2020)

Um zu überprüfen, inwiefern sich der ausgewählte Datensatz für die bevorstehende Bachelorarbeit eignet, werden alle Punkte welche im Vorgehensmodell unter den Schlüsselpunkt „Daten“ beschrieben werden analysiert und auf den Datensatz angewendet. Dieser Bereich stellt den ersten Arbeitsschritt im Vorgehensmodell dar und dient damit als Vorarbeit für die bevorstehende Analyse. In der Abbildung 2 ist zu erkennen wie der Schlüsselbereich „Daten“ im Vorgehensmodell aufgebaut ist. Die Überprüfung der Datenqualität fällt dabei unter den Punkt der Ursprungsdatenquelle, da die Bewertung der Qualität bereits mit dem Beschaffung und Verwaltungsaufwand geprüft wird. Außerdem wird eine Explorative Datenanalyse durchgeführt welche als Vorbereitung für die bevorstehende Aufbereitung des Datensatzes dient. (Neuhaus, 2020, S. 26)

Weiterhin wird das Datenmanagement und die Analytischen Datenquellen betrachtet. Das Daten Management beschreibt dabei die Notwendigkeit, auf die Daten im Laufe des Projektes zugreifen zu können, sowie die Betrachtung des Datenschutzes. Da es sich bei den Datensatz um Betriebsinterne Daten handelt und die Verfügbarkeit durch die Betriebsinterne IT-Infrastruktur gegeben ist, wird dieser Punkt im Folgenden nicht weiter behandelt.

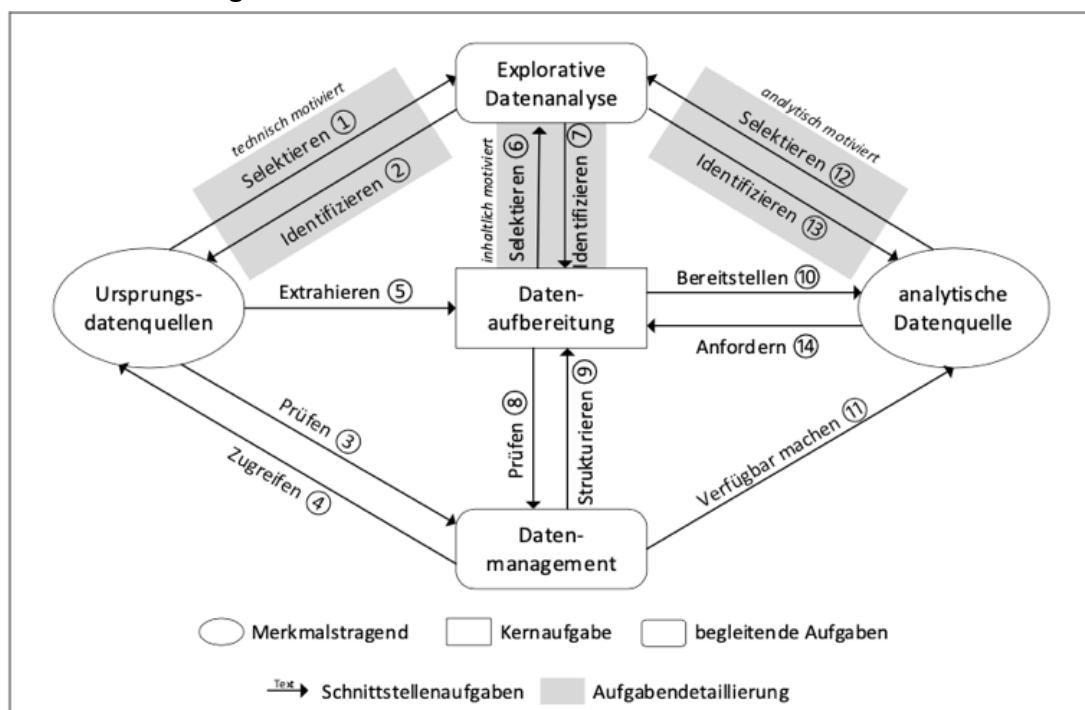


Abbildung 2: DASC-PM Daten (Neuhaus, 2020, S. 26)

Aus dem Vorgehensmodell heraus kann die nicht Erfüllung eines Bereiches, zum Abbruch des Projektes führen. Damit ist Ziel der nachfolgenden Kapitel einen umfangreichen Einblick in die Daten zu gewährleisten und die Projektphase erfolgreich abzuschließen.

Sollte die Durchführbarkeit des Projektes mit dem vorliegenden Datensatz möglich sein, können weiteren Schritte hinsichtlich der Aufbereitung des Datensatzes definiert werden. Ist die Durchführbarkeit des Projektes nicht gegeben, muss das Ziel des Projektes geändert werden oder die Ursprungsdatenquelle angepasst werden.

## 4. Projektdomäne

Da das Domänenwissen im vorgestellten Vorgehensmodell eine übergeordnete Rolle spielt und übergreifend in jeder Phase des Projektes betrachtet werden sollte, werden im Folgenden die Hintergründe zu den gesammelten Daten und dessen Zusammenhänge erklärt. Sowie das Projektziel im Kontext zur Praxis erläutert. (Neuhaus, 2020, S. 10)

Wie beschrieben ist das Ziel der Analyse, die Bestellzahlen welchen täglich im Lager einlaufen, mithilfe eines geeigneten Analyseverfahren vorherzusagen. Um beurteilen zu können welche Daten benötigt werden und welche Aussagekraft diese besitzen, wird im Folgenden der Bestellprozess sowie der Weg zum Erfassungszeitpunkt der erklärt.

Durch die Durchführung des Dualen Studiums bei den in Hamburg ansässigen Drogeriehändler Budnikowsky beziehen sich die weiteren Beschreibungen auf den dort standardisierten Bestellprozess. Die Lieferketten aller Bundikowsky Filialen laufen über das gleiche Logistikzentrum.

Der Bestellauftrag einer Filiale wird automatisch anhand der aktuellen Lagerbestände, der Filiale generiert, anschließend kann der Auftrag außerdem von befugten Mitarbeiter der Filiale manuell angepasst werden. Sobald die Bestellung zusammengestellt und bestätigt worden ist, wird diese an das Logistikzentrum weitergeleitet. Die Frequenz in welcher eine Filiale im Lager bestellt ist von dessen Größer sowie dessen Standort abhängig. Die Bestellhäufigkeit einer Filiale pro Woche ist dabei gleichbleibend, allerdings kann die Bestellhäufigkeit unter den Filialen zwischen eins und fünf variieren.

Sobald alle Bestellungen eingegangen sind, kann mit der Kommissionierung der Ware im Lager begonnen werden. Bei der Kommissionierung wird aus einer großen Mengen an Waren eine spezifische Auftragsabhängige Teilmenge zusammengestellt. Anschließend wird die Kommissionierte Ware für den Versand bereitgestellt.

Nachdem Bestelleingang, wird die Ware innerhalb der nächsten 12-24 Stunden verarbeitet und versendet. Alle Filialen werden dabei in Abhängigkeit der zugeordneten Priorität in Batche eingeteilt. Ein Batch beschreibt dabei eine Teilmenge der gesamten Auftragslast.

Der Batch und die dazugehörigen Aufträge durchlaufen zuerst den gesamten Lagerprozess, bevor der nächste Batch gestartet wird. Dies hat den Vorteil das die Aufträge Priorisiert werden können und die Kapazitäten im Lager gleichmäßig verteilt werden.

Pro Tag werden alle Aufträge in drei bis vier Batche aufgeteilt, die Kommissionierung erfolgt dabei, in Abhängigkeit von der aktuellen Bestelllage, in zwei oder drei Schichtsystem.

Für das Projekt ist unter der Betrachtung der Domäne weiterhin wichtig zu wissen, dass die Kommissionierung der Ware in 3 unterschiedlichen Lagerabteilen durchgeführt, da die verschiedenen Warentypen sich für unterschiedlichen Lagervarianten unterschiedlich gut eignen.

Mithilfe einer Zielgenauen Vorhersage der Bestellzahlen kann die Mitarbeiterplanung angepasst werden. Die Mitarbeiterplanung der nächsten Woche erfolgt am Donnerstag der Vorwoche. Damit eine effiziente Planung anhand der Zeitreihenvorhersage möglich ist, sollte die Zeitreihe Batch und Lagerweise vorhergesagt werden.

Bei 3 unterschiedlichen Lagerbereichen und 4 Batche pro Tag besteht damit die Notwendigkeit 12 Zeitschritte pro Tag vorherzusagen mit einer Vorlaufzeit von 9 Tagen.

Um das Problem zu approximieren wird im folgenden nur die Zeitreihe eines Lagerbereiches analysiert und die Bestellzahlen pro Batch summiert. Sodass pro Tag nur eine Vorhersage für ein Lagerbereich notwendig ist. Sollten die Ergebnisse zufriedenstellend sein kann mit den anderen Bereichen fortgefahrene werden. Da der Prozess gleichbleibend ist, ändert sich von der Anwendung des erstellten Analyseverfahrens, nur die Datengrundlage.

## 5. Datensatz

Im Folgenden werden die Grundlegenden Eigenschaften des Datensatzes erklärt und aufgezeigt, dies ermöglicht im weiteren Verlauf die Korrekte Umsetzung der weiteren Bearbeitungsschritte.

Der zu Analysierende Datensatz umfasst 2093 Zeilen mit jeweils 3 Spalten (vgl. Abbildung 3).

- Die erste Spalte beschreibt das Datum an welchen der Datensatz erfasst worden ist, die Zeitreihe reicht dabei von Anfang 2015 bis hin zum Zeitpunkt des Datenabzuges im November 2021.
- In der zweiten Spalte befinden sich die Auftragsdaten des jeweiligen Tages.
- Die dritte Spalte beschreibt den Lagerbereich in welchen die Aufträge abgearbeitet werden. Wie in der Abbildung 4 zu erkennen ist, ist der Datensatz bereits selektiert und auf einen spezifischen Lagerbereich gefiltert, sodass dieser im weiteren Verlauf verwendet werden kann

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2093 entries, 0 to 7361
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Datum        2093 non-null   object 
 1   VPE_Gesamt   2093 non-null   int64  
 2   Lager        2093 non-null   object 
dtypes: int64(1), object(2)
memory usage: 65.4+ KB
```

Abbildung 3: Daten Info (Eigene Darstellung)

	Datum	VPE_Gesamt	Lager
0	2015-02-01	83685	DPS
1	2015-03-01	83810	DPS
2	2015-05-01	74101	DPS
3	2015-06-01	56912	DPS
4	2015-07-01	58630	DPS
...	...	...	...
2088	2021-01-11	49484	DPS
2089	2021-02-11	81549	DPS
2090	2021-03-11	76804	DPS
2091	2021-04-11	82104	DPS
2092	2021-05-11	17347	DPS

2093 rows × 3 columns

Abbildung 4: Daten Ausschnitt (Eigene Darstellung)

## 6. Ursprungsdatenquelle

Die Analyse der Ursprungsdatenquelle umfasst die Betrachtung von Faktoren, welche das uneingeschränkte Arbeiten hinsichtlich des Datenzugriffs im Projekt ermöglichen. Vor allem der Beschaffungs- und Verwaltungsaufwand spielen dabei eine übergeordnete Rolle. Außerdem wird die Datenqualität der Rohdaten auch im Punkt der Ursprungsdatenquelle geprüft.

Der Beschaffungs- und Verwaltungsaufwand, der für den vorliegenden Datensatz benötigt wird, ist gering, da der Prozess bereits automatisiert ist und die Daten nachdem sie erfasst worden sind redundanzfrei auf eine Datenbank abgelegt werden. Bei der Datenbank handelt es sich um eine Microsoft Datenbank, welche im klassischen relationalen SQL-Format strukturiert ist. Außerdem ist die Datenbank im Laufe des Projektes, für alle Projektbeteiligten frei zugänglich, wodurch ein sauberer Umgang mit den Daten während des Projektes gewährleistet ist.

Häufig wird bei der ersten Betrachtung der Ursprungsdatenquelle sowie des Rohdatensatzes von der Qualität eines Datensatzes gesprochen. Die Qualität eines Datensatzes, ist dabei ein nicht klar definierter Begriff und kann nicht anhand eines Maßstabes gemessen werden. Meist wird von qualitativ hochwertigen Daten gesprochen, sobald die Daten vollständig, aktuell und über ein großen Bezug zum vorliegenden Projekt verfügen. (Gitzel, 2016)

Eine weitere Möglichkeit ist es die Datenqualität anhand verschiedener Parameter zu prüfen, eine beispielhafte Auflistung von infrage kommender Parametern ist in Abbildung 5 zusehen. Im Folgenden wird der Datensatz anhand ausgewählter Parametern aus der Abbildung 5 verglichen und geprüft. Dabei ist entscheidend inwiefern die aufgelisteten Bewertungskriterien auf den vorliegenden Datensatz zutreffen oder auffindbar sind. Außerdem wird geprüft, ob der betrachtete Datensatz, eine für die Aufgabenstellung essentielle Eigenschaft nicht besitzt und inwiefern das Fehlen der Eigenschaft die Qualität der Daten in einem Maß beeinflusst das der Misserfolg des Projektes begünstigt wird.

Die Qualitätsprüfung des Datensatzes verfolgt dabei nicht den Anspruch eine Vollumfängliche Analyse durchzuführen, sondern die Prüfung von Eigenschaften, welche für die spezifische Aufgabenstellung vorhanden sein müssen. Damit kann zwar kein Anspruch der Vollständigkeit erhoben werden, allerdings kann dennoch mit der Prüfung der relevantesten Parameter ein unvorhersehbares Scheitern, bedingt durch die Qualität und Form des Rohdatensatzes, frühzeitig auf ein Minimum reduziert werden.

Infolge werden die geprüften Parameter vorgestellt und geprüft. Teilweise benötigt die Prüfung der Parameter ein tiefen Einblick in die Daten und erfolgen demnach erst mit Explorativen Datenanalyse. (Helfert, 2001)

Aktualität	Allgemeingültigkeit	Alter	Anderungshäufigkeit	Aufbereitungsgrad
Bedeutung	Benutzbarkeit	Bestätigungsgrad	Bestimmtheit	Detailliertheit
Effizienz	Eindeutigkeit	Fehlerfreiheit	Flexibilität	Ganzheit
Geltungsdauer	Genauigkeit	Glaubwürdigkeit	Gültigkeit	Handhabbarkeit
Integrität	Informationsgrad	Klarheit	Kompaktheit	Kompression
Konsistenz	Konstanz	Korrektheit	Neutralität	Objektivität
Operationalität	Performance	Portabilität	Präzision	Problemadäquatheit
Prognosegehalt	Prüfbarkeit	Quantifizierbarkeit	Rechtzeitigkeit	Relevanz
Reliabilität	Richtigkeit	Robustheit	Seltenheit	Sicherheit
Signifikanz	Speicherbedarf	Standardisierungsgrad	Subjektdäquatheit	Testbarkeit
Umfang	Unabhängigkeit	Überprüfbarkeit	Übertragbarkeit	Validität
Verdichtungsgrad	Verfügbarkeit	Verfügungsmacht	Verknüpfbarkeit	Verlässlichkeit
Verschlüsselungsgrad	Verständlichkeit	Vertrauenswürdigkeit	Verwendungsbereitschaft	Vollständigkeit
Wahrheitsgehalt	Wahrscheinlichkeit	Wartungsfreundlichkeit	Wiederverwendbarkeit	Wirkungsdauer
Zeitadäquatanz	Zeithorizont	Zeitoptimal	Zuverlässlichkeit	Zuverlässlichkeit

Abbildung 5: Qualitätsmerkmale Daten (Helfert, 2001)

## **Konsistenz**

Die Datenkonsistenz beschreibt die Korrektheit und Vollständigkeit der gespeicherten Daten, dass bedeutet das die vorliegenden Daten eine widerspruchsfreie und vollständige Abbildung aller relevanten Aspekte der Realität darstellen. (Harris, 2013)

Da es sich bei den vorliegenden Datensatz um intern gesammelte Daten handelt kann es lediglich durch eine falsche Bestellung einer Filiale oder durch einen Fehler beim Datentransfer zu Inkonsistenten Daten kommen.

Sollte eine Filiale eine falsche Bestellung aufgeben und der Auftrag nicht zu deren Zufriedenheit abgearbeitet werden, entspricht dieser Datensatz trotzdem den realen Umständen, da ein solcher Fehler erst nach dem eigentlichen Kommissionier Prozess erkannt wird und somit die Ware die gleichen Ressourcen beansprucht.

Das Ziel der Analyse ist die Mitarbeiterplanung entlang des Logistikprozesses zu optimieren und die Bestellzahlen vorherzusagen, damit handelt es sich auch um relevante Daten sollte eine Bestellung einer Filiale fehlerhaft sein oder falsche Artikel beinhalten, da diese Artikel den gleichen Prozess sowie den gleichen Aufwand im Lager bedeuten. Die Vollständigkeit bei einer Zeitreihe ist gegeben, wenn es zu jedem relevanten Zeitstempel auch ein erfasster Datenpunkt gibt. Die Vollständigkeit wird damit im weiteren Verlauf der explorativen Datenanalyse geprüft. Weiterhin ist bereits in Abbildung 3 zu erkennen, dass sich im Datensatz keine Nullwerte enthalten sind.

## **Redundanzfreiheit**

Sollte die gleiche Information mehrfach abgelegt worden sein, wird von einer Dublette gesprochen und der Datensatz ist nicht mehr redundanzfrei. Dubletten sollten vermieden werden, da diese die Ergebnisse des Analyseverfahrens verzehren können. (Helfert, 2001)

Da die Daten täglich erfasst worden sind und für den Arbeitsprozess benötigt werden wurde die Zeitreihe bereits auf Fehler und Dubletten geprüft, jedoch sollten bei der folgenden Explorativen Datenanalyse mehrfach gespeicherte Zeitschritte auffallen und die in der Datenaufbereitung entfernt werden.

## **Verständlichkeit**

Die Daten sind verständlich sobald diese im richtigen Datentyp vorliegen und alle Beschreibung sowie Überschriften verständlich sind, zwingenderweise sollten dabei die Begrifflichkeiten und Strukturen mit der Vorstellung des Fachbereiches übereinstimmen (van der Aalst, 2016).

Die Verständlichkeit des Datensatzes ist gegeben wie in Abbildung 4 zu erkennen ist. Außerdem können die Überschriften geändert werden, sollte es zu Missverständnissen im Projektverlauf kommen. Allerdings ist in Abbildung 3 zu erkennen, dass die Spalte Datum, nach dem Transfer von der Datenbank, nur im Datentyp String vorliegt, wodurch eine weitere Verarbeitung in der Analyse nur eingeschränkt möglich ist. Damit ist die Transformation des Strings in ein geeignetes Zeitformat, im Prozess der Datenaufbereitung zu berücksichtigen.

## **Zuverlässigkeit**

Die Datenquellen sollten zu jedem Zeitpunkt nachvollziehbar sein. Die Zuverlässigkeit der Daten, reicht dabei von der Aggregierung über die Analyse bis hin zur Löschung. Wichtig dabei ist, dass die Datenquellen bekannt sind und zuverlässig sind. (van der Aalst, 2016)

Dadurch, dass der Prozess hinter der Datenaggregierung bekannt ist, kann davon ausgegangen werden, dass die erfassten Daten zuverlässig sind. Allerdings ist es nicht möglich, sollte es zu fehlerhaften Werten kommen, diese nachzuvollziehen und die Ursache hinter dem Messfehler zu finden. Damit ist dieser Aspekt nur teilweise erfüllt.

## 7. Explorative Datenanalyse

Das Ziel der Explorativen Datenanalyse ist es ein besseres inhaltliches Verständnis über den vorliegenden Datensatz aufzubauen. Mithilfe dessen kann geklärt werden ob die Daten in ausreichender Menge zur Verfügung stehen und inwiefern die Qualität und Aussagekraft der Daten zur Beantwortung der Fragestellung genügt. Die Explorative Datenanalyse besteht dabei aus der Datenvisualisierung und der Anwendungen von statistischen Methoden (Pastorello, 2014). Auftretende Probleme und Auffälligkeiten werden dabei entsprechend dokumentiert, sodass diese in der folgenden Datenaufbereitung behandelt werden können. Im Folgenden wird der vorliegende Datensatz visualisiert und untersucht.

Wie bereits vorgestellt umfasst der Datensatz 2093 Zeilen und 3 Spalten, in welchen Informationen über den Zeitpunkt, Bestellmenge und Lagerbereich enthalten sind. Um ein erster Überblick über die Daten zu gewinnen werden im ersten Schritt verschiedene Kennzahlen aus der deskriptiven Statistik ausgewertet, welche in Abbildung 6 dargestellt sind.

Zuerkennen ist, dass der Durchschnitt und der Median nah beinaderliegen, was vermuten lässt, dass die Verschiedenen Bestellvolumina entlang der Zeitreihe Normalverteilt sind.

Außerdem lässt sich durch das errechnete Quartil erkennen, dass ein Großteil der erfassten Daten nah am Mittelwert liegen, wodurch die Zeitreihe eine schwache Streuung aufweist.

Trotz der schwachen Streuung ist zu erkennen, dass der Größte und kleinste Wert weit vom errechneten Durchschnitt entfernt liegen, resultierend daraus kann es notwendig sein die erkannten Ausreißer im Schritt der Datenaufbereitung zu entfernen.

VPE_Gesamt	
<b>count</b>	2093.000000
<b>mean</b>	56518.604396
<b>std</b>	15008.320062
<b>min</b>	1365.000000
<b>25%</b>	48218.000000
<b>50%</b>	56573.000000
<b>75%</b>	66108.000000
<b>max</b>	129499.000000

Abbildung 6: Daten Kennzahlen (Eigene Darstellung)

Im nächsten Schritt werden die Daten visualisiert, dafür werden die gemessenen Werte entlang der Zeitreihe in einem Histogramm und Liniendiagramm dargestellt.

Das Histogramm teilt die Summe der täglich Bestellten Ware in Klassen auf, welche an der X-Achse erkennbar sind. Mithilfe der Frequenz der jeweiligen Klassen im Datensatz, welche an der Y-Achse erkennbar ist, lässt sich eine Aussage über die Verteilung der unterschiedlichen Bestell Volumina täglich treffen (vgl. Abbildung 8).

Das Liniendiagramm plottet alle verfügbaren Datenpunkte entlang des verfügbaren Zeitraumes, mithilfe dessen können Eigenschaften von Zeitreihen wie Trends, Saisonalitäten oder Rauschen erkannt werden, welche im Kapitel 2 vorgestellt worden sind. (vgl. Abbildung 7).

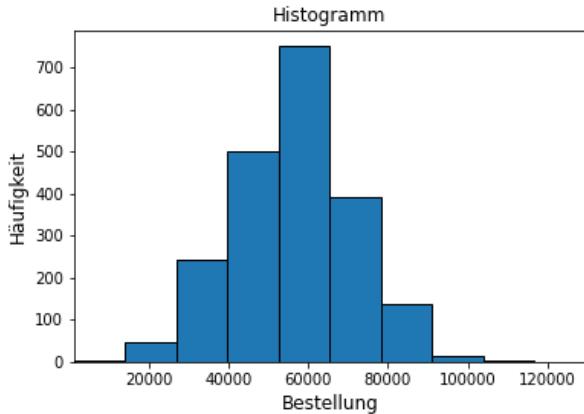


Abbildung 7: Histogramm gesamter Datensatz  
(Eigene Darstellung)

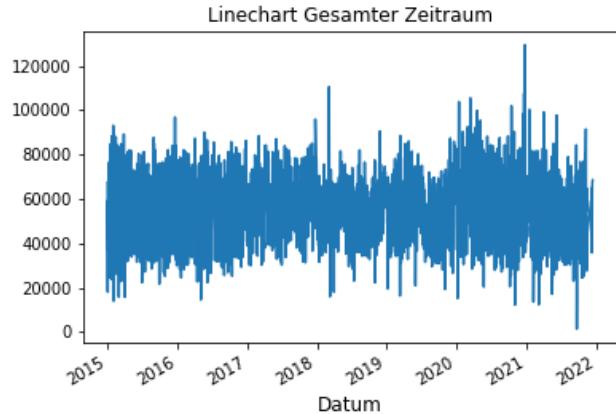


Abbildung 8 Liniendiagramm gesamter Datensatz  
(Eigene Darstellung)

Das Histogramm lässt erkennen, dass das Bestellvolumen betrachtet auf den gesamten Zeitraum annähernd normal verteilt ist und es nur wenige Ausreißer nach unten und oben gibt, was eine gute Grundlage für die bevorstehende Analyse darstellt. Sollten die Daten schwanken und zu Extremwerten tendieren, besteht die Gefahr das keine Korrelation unter den einzelnen Werten erkennbar ist und die Schwankung der Zeitreihe nur von äußereren, nicht betrachteten Faktoren abhängig ist, damit wäre die Vorhersage kaum bis gar nicht möglich.

Im Liniendiagramm in Abbildung 7 lassen sich kaum bis keine Mustere erkennen, außerdem hat es den Anschein, dass die Datenpunkte zufällig verteilen sind, wodurch eine Vorhersage nur schwer bis gar nicht möglich wäre, dies kann aber auch eine Folge des größtmöglichen Betrachtungszeitraums sein. (Wilke, 2020)

Für ein detailliertere Untersuchung wurde in der Abbildung 9, ein Zeitraum von 100 Tagen im Liniendiagramm dargestellt. In der Abbildung lässt sich erkennen das ein Rauschen zwar anteilig vorhanden ist, allerdings lassen sind auch Muster und Trends im Datensatz erkennen.

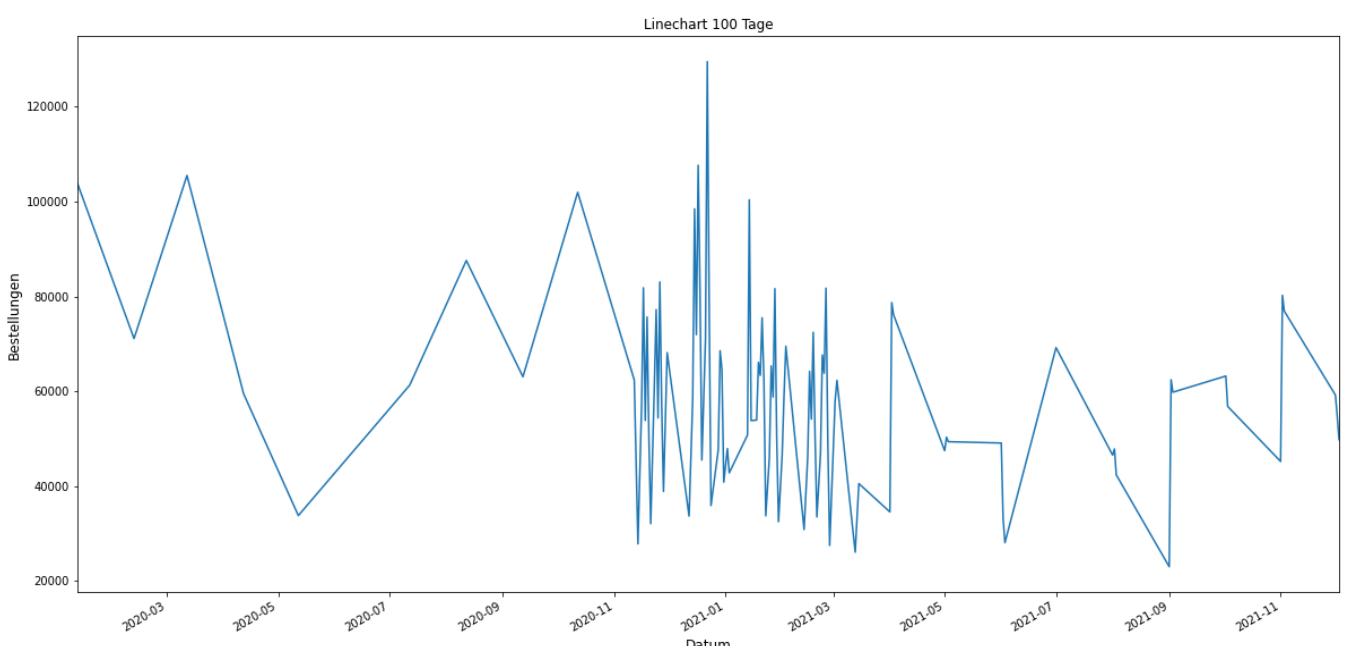


Abbildung 9 Liniendiagramm 100 Tage (Eigene Darstellung)

Damit kann es in Abhängigkeit des gewählten Analyseverfahrens sinnvoll sein die Zeitreihe, im Schritt der Datenaufbereitung, zu glätten. Die Glättung von Zeitreihen kann die Fähigkeit des Modells auf Ausreißer und extreme Schwankung der Zeitreihe zu reagieren allerdings beinträchtigen (Nielsen, 2020). Im nächsten Schritt werden die Ausreißer welche sich in den Abbildung 7 und Abbildung 8 gezeigt haben, genauer untersucht. Dafür wurden die Daten im Boxplots dargestellt. Zum einen wurden die Daten in Abbildung 10 anhand der Wochentage an welchen sie erfasst worden sind, in verschiedene Boxplots unterteilt. Zum anderen wurden die Daten in Abbildung 12 anhand der Kalenderwoche in Box-Plots unterteilt.

Zusätzlich wurde für die Untersuchung von Saisonalitäten das Durchschnittliche Bestellvolumen an spezifischen Wochentagen und Monaten in Heat-Maps dargestellt.

Die Aufteilung nach Wochentagen lässt erkennen dass die verschiedenen Wochentage ein spezifisches Verhalten besitzen. Besonders auffällig ist es, dass es an Donnerstagen zu Ausreißern nach unten kommt. Allerdings liegt der Durchschnitt größtenteils über den anderen Wochentagen. Besonders wird das im Vergleich zu Heatmap deutlich (vgl. Abbildung 11). Im Vergleich zu den ausgewerteten Liniendiagramm lassen die einzelnen Wochentage deutlich mehr Muster erkennen. Besonders deutlich wird das bei der Betrachtung der Unterschiede zwischen den Samstag und Montag. Damit lassen sich innerhalb der einzelnen Wochentage Trends aufzeigen.

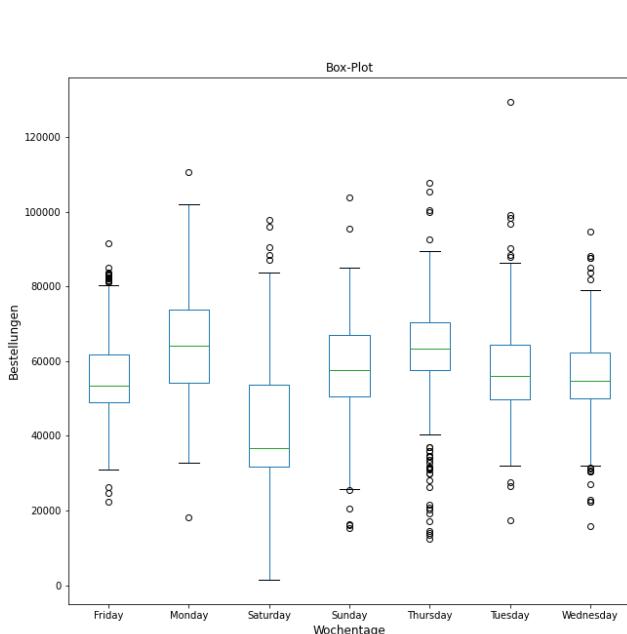


Abbildung 10 Box-Plot Wochentage  
(Eigene Darstellung)

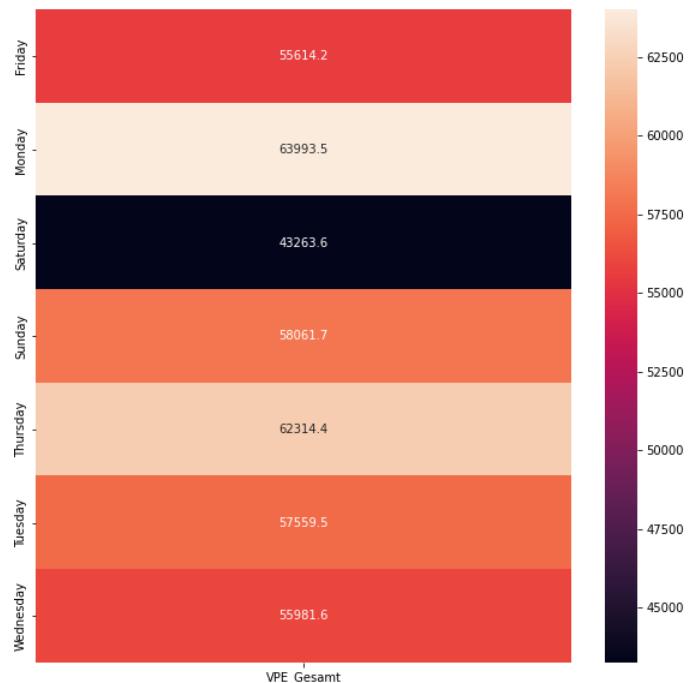


Abbildung 11 Heat-Map Wochentage  
(Eigene Darstellung)

Die Betrachtung der Box-Plots in Abbildung 12 lässt erkennen dass es innerhalb der Monate auch zu Saisonalitäten kommt, da sowohl mit Betrachtung des Medians als auch mit Betrachtung des oberen Quartils eine vier Wöchiger Trend erkennbar ist.

Weiterhin lassen sich Saisonalitäten innerhalb eines Jahres erkennen (vgl. Abbildung 13). Besonders auffällig sind dabei die letzten Monate jedes Jahres, was zurückzuführen ist auf das Weihnachtsgeschäft. Ähnlich wie der Anstieg gegen März erklärbar durch das Ostergeschäft ist. Fraglich ist jedoch, ob diese Saisonalität innerhalb der Zeitreihe, bei einer geplanten täglichen Vorhersage, ein positiver Effekt auf das Analyseergebnis hat.

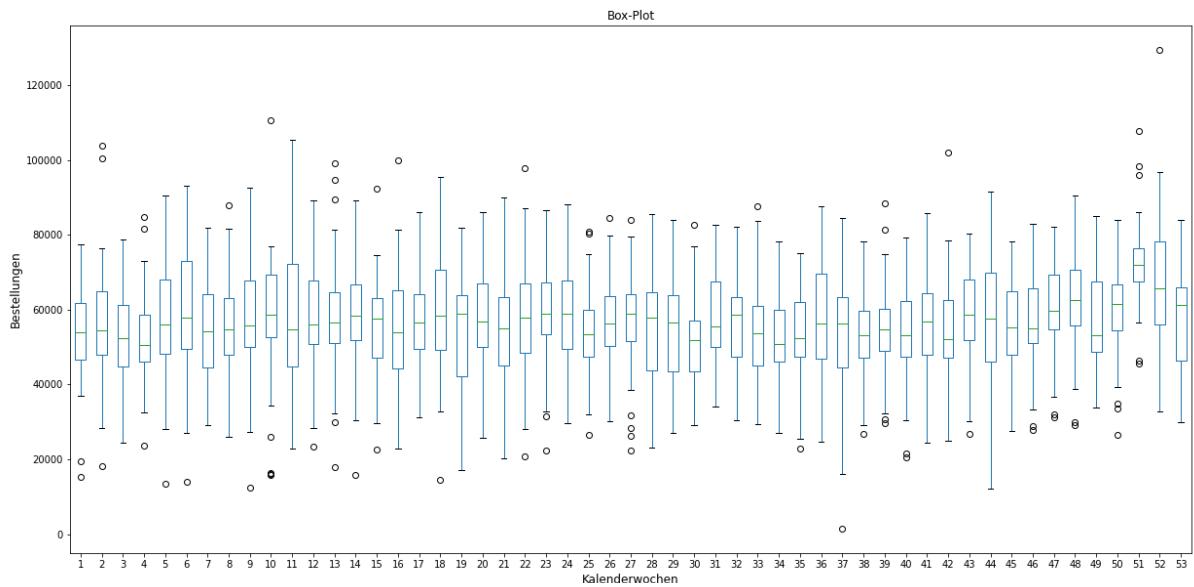


Abbildung 12: Box-Plot Wochen (Eigene Darstellung)



Abbildung 13 Heat-Map Monate (Eigene Darstellung)

## 8. Datenaufbereitung

Das Grundlegende Ziel der Datenaufbereitung ist es, die Daten in eine Form zubringen in welche sich diese, für ein ausgewähltes Analyseverfahren eignen. Weiterhin werden mithilfe der Ergebnisse aus der Explorativen Datenanalyse und den Erkenntnissen der Qualitätsbewertung Änderung am Datensatz vorgenommen, um zum einen die Qualität der Daten zu verbessern und zum anderen zu verhindern das Unstimmigkeiten im Datensatz für eine Verzerrung des Analyseergebnisses sorgen. Das Ziel des Kapitels ist es damit, mithilfe der gewonnenen Erkenntnisse aus der Explorativen Datenanalyse sowie aus der Untersuchung verschiedenere Qualitätskriterien, Schritte zu definieren die es gilt im Prozessschritt der Datenaufbereitung zu automatisieren und durchzuführen. Die Automatisierung des Prozesses setzt den Grundstein für die darauffolgende Analyse, sowie die Integration des fertigen Modells in den Livebetrieb.

Die Qualitätsprüfung hat gezeigt das viele Kriterien wie die Prüfung der Vollständigkeit, Korrektheit, Redundanzfreiheit oder Konsistenz bereits sichergestellt werden können, sobald im Schritt der Datenaufbereitung alle Dubletten entfernt worden sind und die Zeitreihe dahingehend geprüft wird ob es Zeitstempel gibt denen kein Messwert zugeordnet werden kann.

Andere Faktoren die in der Qualitätsprüfung genannt worden sind, wie der Wahrheitsgehalt, Zuverlässigkeit und die Validität der Daten, werden bereits darüber sichergestellt, dass es sich um interne Daten handelt und das der Prozess der Datenaggregierung bekannt ist. Außerdem kann durch die interne IT-Infrastruktur garantiert werden, dass es zu keinen Datenverlust beim Daten Transfer kommt. Dennoch sollte in der Datenaufbereitung geprüft werden ob Daten vorhanden sind, die nicht der Wahrheit entsprechen können oder extreme Ausreißer darstellen, sodass diese im Laufe des Prozesses entfernt werden können.

Die Explorative Datenanalyse hat gezeigt das sich im Datensatz viele Ausreißer befinden und das es Phasenweise in der Zeitreihe zu zufälligen Ereignissen kommt, was sich wie in Abbildung 9 zusehen als Rauschen darstellt. Schlussfolgernd daraus besteht die Möglichkeit, in der Datenaufbereitung sowohl die Ausreißer zu entfernen als auch die Zeitreihe zu glätten um den rauschenden Effekt zu minimieren. (Nielsen, 2020)

Inwiefern diese Schritte notwendig ist, sollte in Abhängigkeit des gewählten Analyseverfahrens entschieden werden, da sowohl das Rauschen als auch die Ausreißer ein Großteil des Datensatzes darstellen.

Mit der gesamten Bereinigung des Rauschen und der Ausreißer, verliert das fertige Model die Fähigkeit auf ein solches Rauschen oder entsprechende Ausreißer reagieren zu können. Damit kann zwar ein gutes Ergebnis auf den Testdatensatz erwartet werden, jedoch würde die Performance im Live Betrieb stark leiden. (Nielsen, 2020)

Weiterhin konnte bei der Explorativen Datenanalyse festgestellt werden, dass im kurzen Zeitraum kaum Saisonalitäten und Trends erkennbar waren. Allerdings konnten die Heat Maps welche die Bestellung der Wochentage und Monate untersuchten (vgl. Abbildung 11,13) deutlich darstellen, dass es im Laufe des Jahres und abhängig vom Wochentag oder Woche zu Saisonalitäten kommt. Für die Errechnung des nächsten Zeitschrittes im Analyseverfahrens, kann damit die Betrachtung der erfassten Bestellzahlen der letzten Jahre zum nächsten Zeitschritt, ein deutlich besseres Ergebnis erbringen als die Betrachtung der zuletzt erfasst Bestellzahlen. Damit sollte eine Analyse dahingehend möglich sein.

## 9. Fazit

Die vorliegende Transferleistung hatte als Ziel gehabt einen Datensatz, welcher Bestellzahlen beinhaltet, auf dessen Qualität zu prüfen und außerdem mithilfe einer Explorativen Datenanalyse Einblicke in den Datensatz zu gewähren. Damit ist das Ziel der Arbeit erfüllt worden, indem der betrachtete Datensatz anhand diverser Qualitätskriterien bewertet worden ist, außerdem wurden beschreibende Kennzahlen der Deskriptiven Statistik ermittelt und der Datensatz visualisiert.

Die Ergebnisse entsprechen insofern den Anforderungen, dass der Datensatz formal nach dem Vorgehensmodell auf die darin enthaltenen Qualitätsmerkmale geprüft worden ist.

Zwar konnte aufgezeigt werden, dass die vorliegenden Daten eine hohe Qualität aufweisen, welche mithilfe der Datenaufbereitung verbessert werden kann. Allerdings kann mithilfe der Ergebnisse, keine Aussage darüber getroffen werden, inwiefern die zukünftigen Ergebnisse des ausgewählten Analyseverfahrens ausfallen werden. Damit bleibt fraglich, in wie fern eine Vorhersage der Zeitreihe möglich ist und ob die Daten die notwendige Aussagekraft liefern können.

Weiterhin sei gesagt, dass die Transferleistung insofern das Ziel verfehlt hat, da die Qualitätsprüfung von multivarianten Zeitreihen das Hauptthema war. Allerdings wurde der Datensatz aufgeteilt und als mehrere Univariate Zeitreihen untersucht. Die Beschreibung und Deutung einer einzelnen Zeitreihe erfolgte im Laufe der Arbeit.

Der Grund für die Untersuchung der Zeitreihe, als univariate Zeitreihen und nicht als Darstellung einer multivariaten Zeitreihe, ist zu erklären durch das mangelnde Wissen bei Beginn der Arbeit.

Zu Beginn der Arbeit wurde davon ausgegangen, dass es sich im vorliegenden Datensatz um eine Multivariate Zeitreihe handelt, da an jedem Tag die eingelaufenen Bestellzahlen von mehreren Lagerabschnitten dokumentiert werden, damit sind zu jeden Zeitstempel mehrere Datenpunkte vorhanden, woraus sich die Schlussfolgerung ergibt, dass es sich um eine multivariate Zeitreihe handelt.

Mit der Recherche und weiteren Untersuchung von Zeitreihen, wurde klar, dass es bei multivarianten Zeitreihen essentiell ist, dass die Aussagekraft der Zeitreihe von mehreren beobachteten Parametern abhängig ist. Sollte dies nicht der Fall sein, sind es lediglich Univariate Zeitreihen die gemeinsam untersucht werden, was die Untersuchung komplexer gestaltet, allerdings kein positiven Effekt auf die Ergebnisse mit sich bringt.

Da es sich bei den betrachteten Daten um unabhängige Zeitreihen handelt, liegen mehrere Univariate Zeitreihen vor, wodurch die Aussagekraft bei der Betrachtung der einzelnen Zeitreihen nicht eingeschränkt ist.

Damit wurde mehrere unabhängige univariate Zeitreihen fälschlicherweise als Multivariate Zeitreihe angenommen, wodurch sich der Schwerpunkt im Laufe der Arbeit veränderte. Dennoch konnten die Daten untersucht und Ergebnisse erzielt werden, indem nötige Schritte zur Datenaufbereitung festgelegt worden sind.

## 10. Literaturverzeichnis

Cisco Systems. (2018). *Cisco Global Cloud*. Cisco Systems.

Gartner. (2015). Gartner Business Intelligence & Analytics Summit 2015. München: Gartner Summits.

Gitzel, R. (2016). *Data Quality in Time Series Data*. Ladenburg: Proceedings of CBI.

Harris, H. M. (2013). *Analyzing the Analyzers: An Introspective Survey of Data Scientists and Their Work*. O'Reilly.

Helfert, M. H. (2001). *Datenqualitätsmanagement*. St. Gallen: Institut für Wirtschaftsinformatik.

Kreiß, J.-P. (2006). *Einführung in die Zeitreihenanalyse*. Berlin: Springer-Verlag Berlin Heidelberg.

Neuhaus, U. M. (2020). *DASC-PM v1.0 - Ein Vorgehensmodell für Data-Science Projekte*. Hamburg.

Nielsen, A. (2020). *Practical Time Series Analysis*. Gravenseein: O'Reilly.

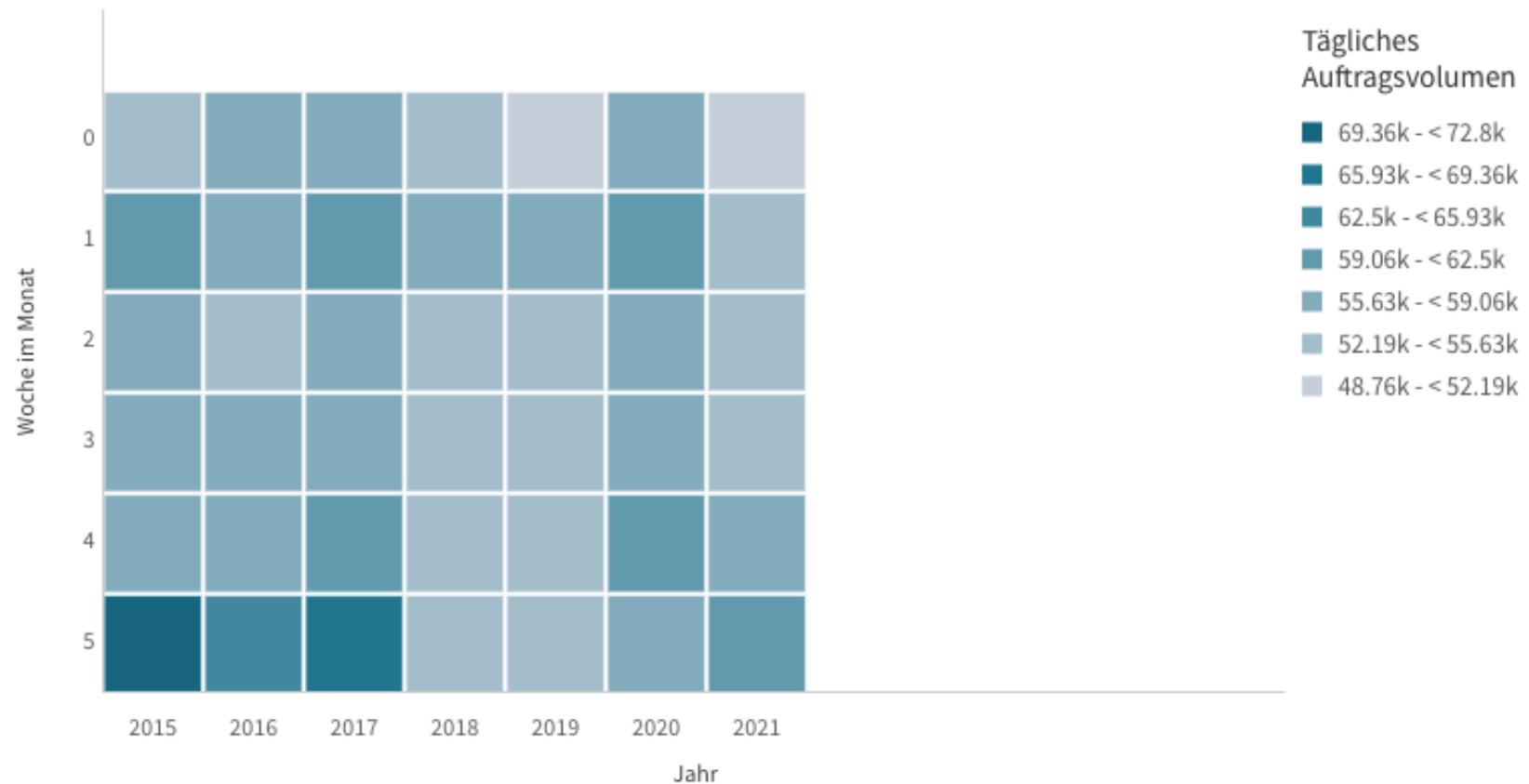
Novustat. (April 2019). *novustat.com*. Abgerufen am 10. 11 2021 von  
<https://novustat.com/statistik-blog/zeitreihenanalyse-fuer-dummies-in-r.html>

Pastorello, G. (2014). Observational Data Patterns for Time Series Data Quality Assessment. *10th IEEE International Conference on e-Science*. Guaruja, Brazil: AmenFlux.

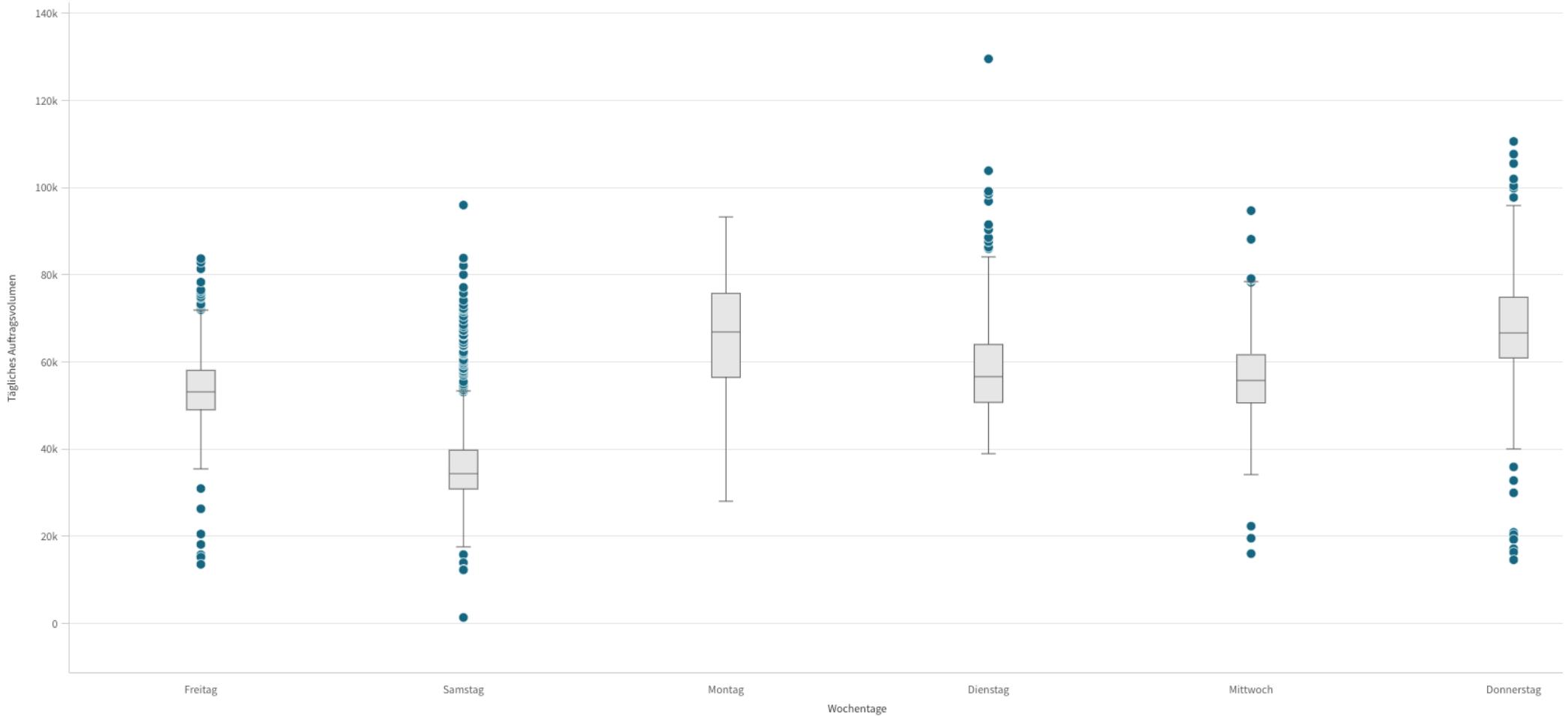
van der Aalst, W. (2016). *Data science in action*. Berlin: Springer.

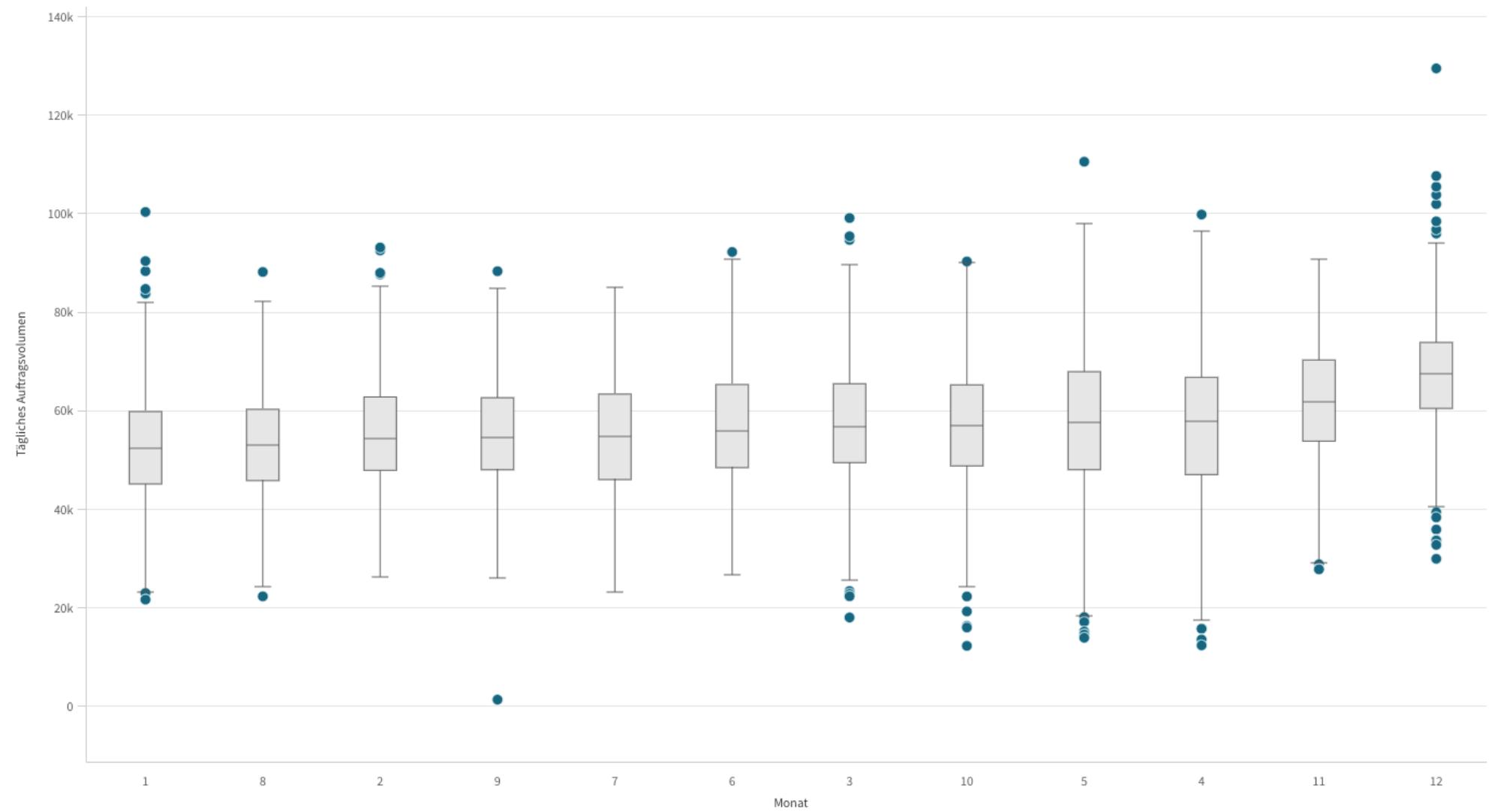
Wilke, C. O. (2020). *Datenvisualisierung – Grundlagen und Praxis*. Berlin: O'Reilly.

Anhang 7  
Heatmap, Bestellverhalten in Abhängigkeit  
der Woche im Monat

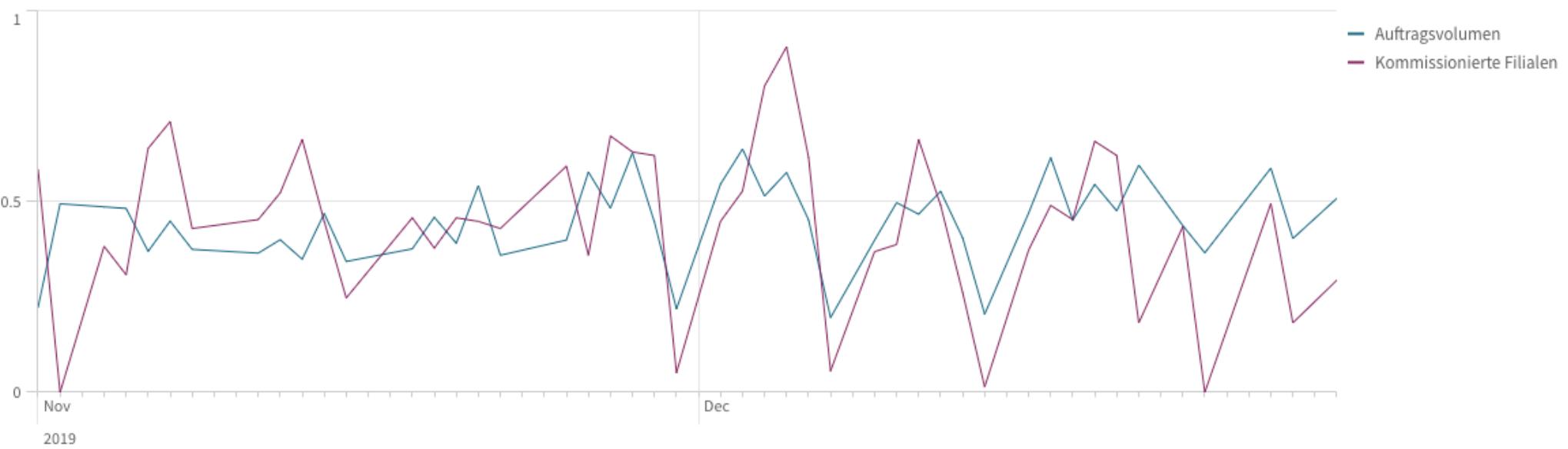
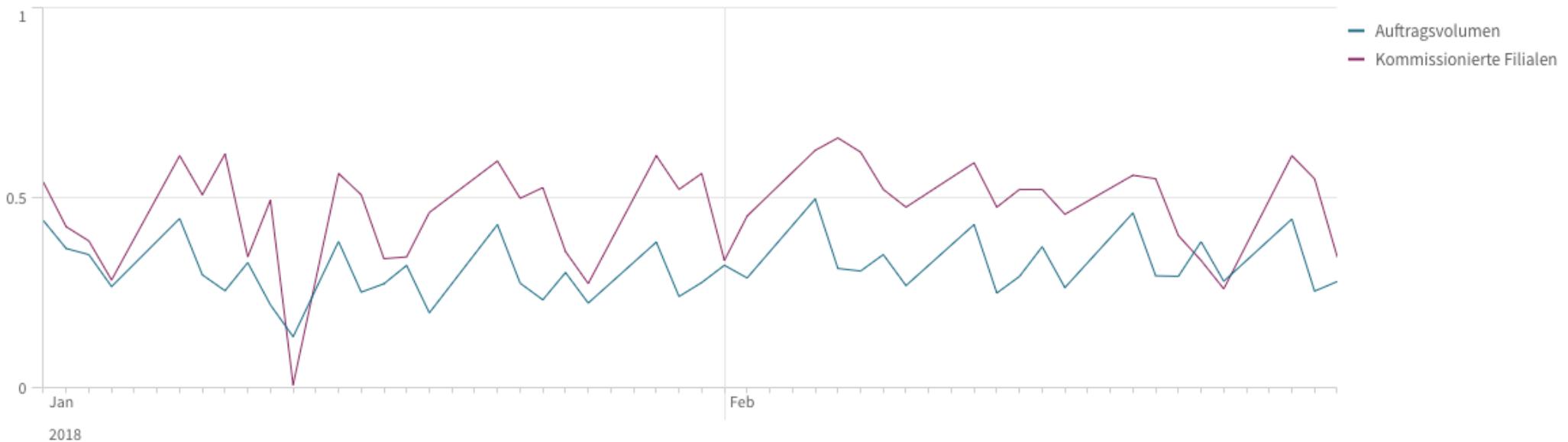


Anhang 8  
Boxplot-Auftragsdaten

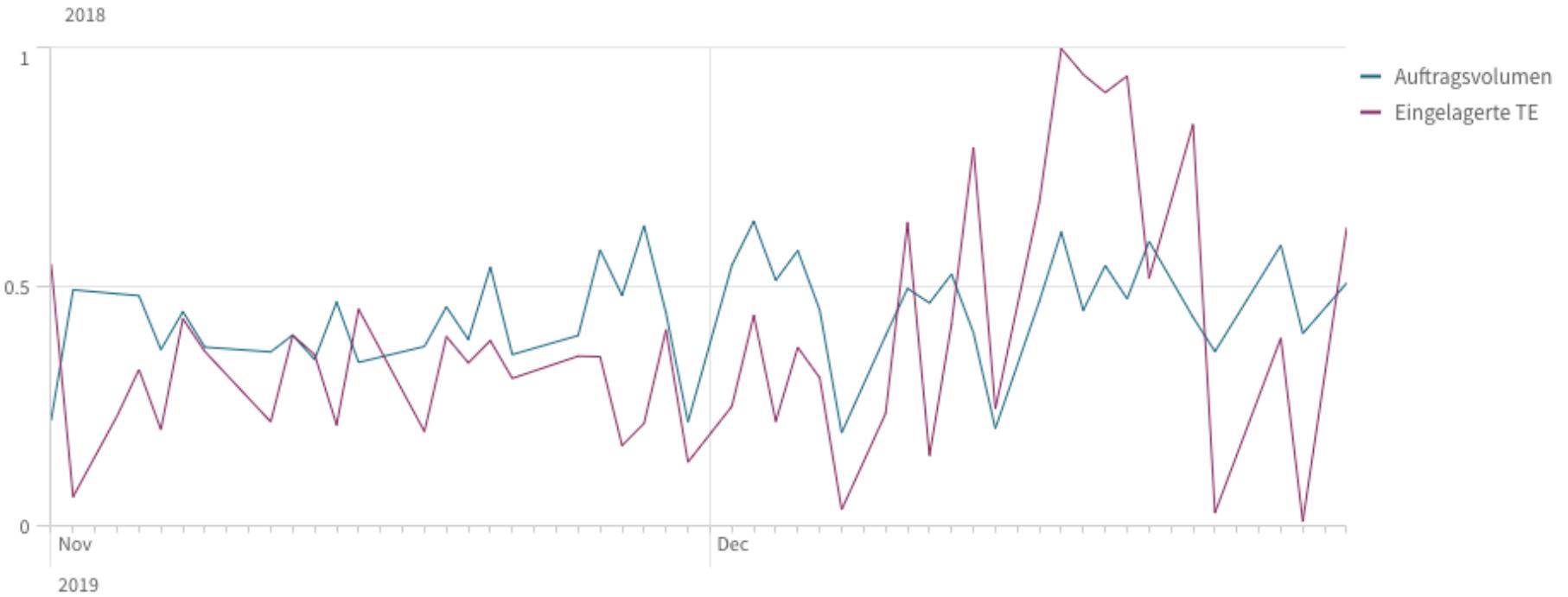
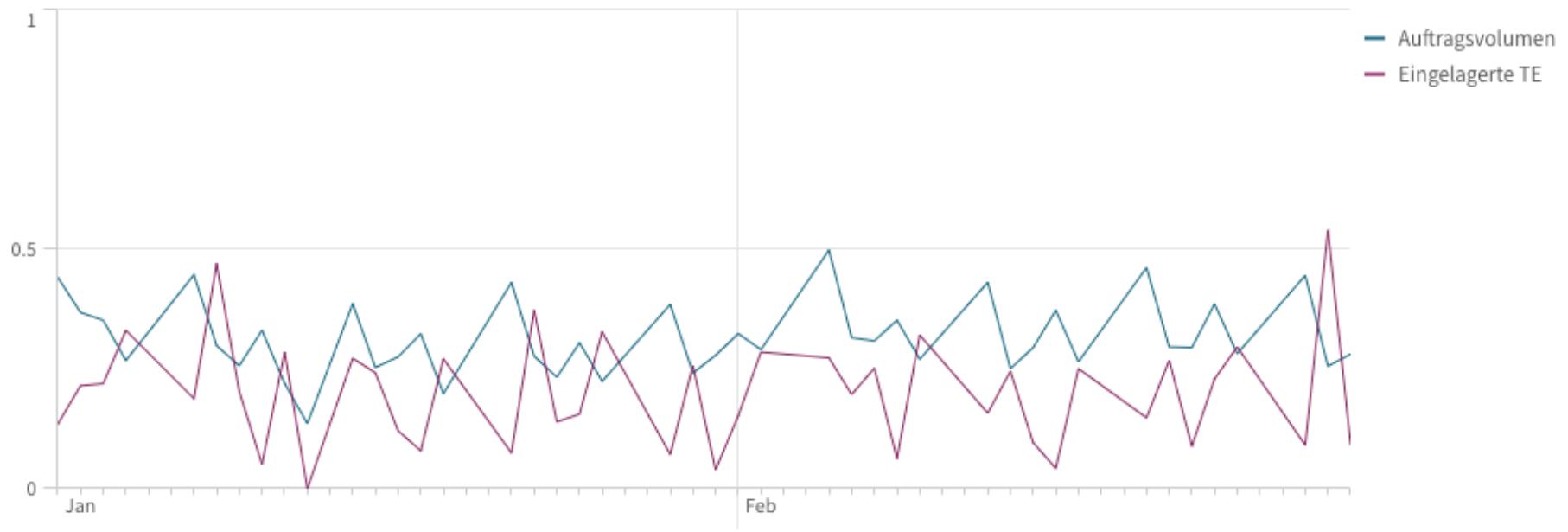




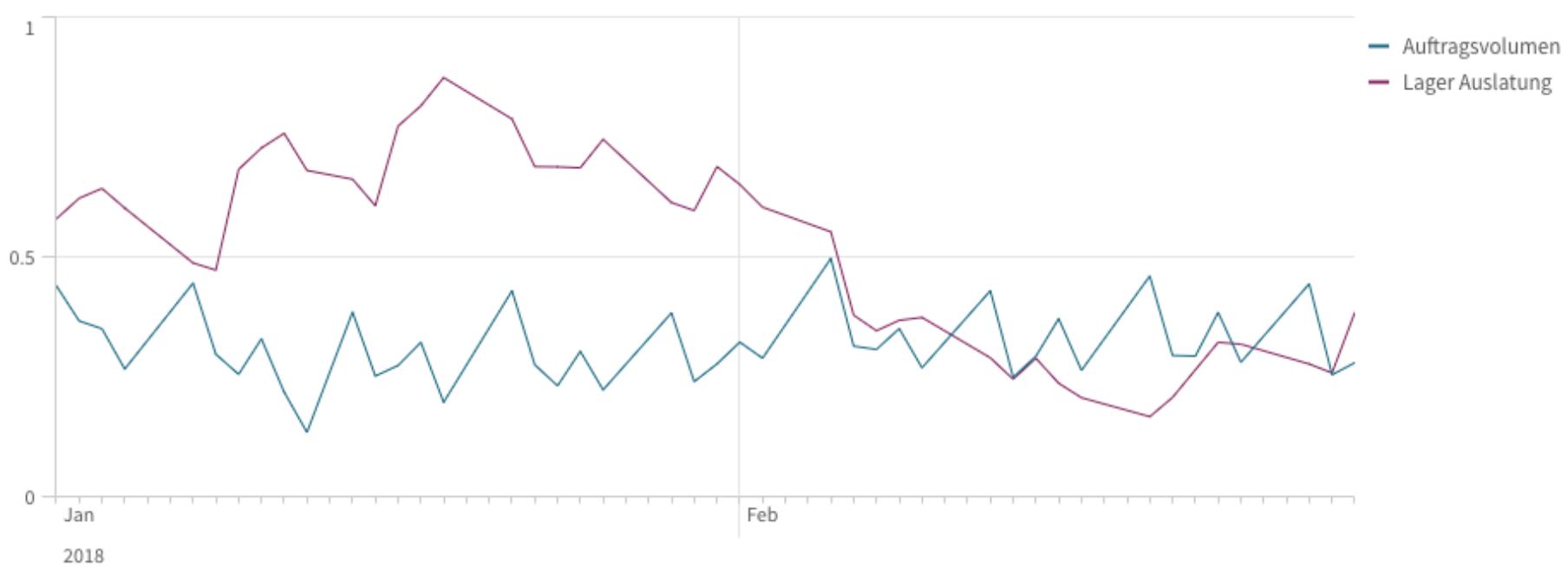
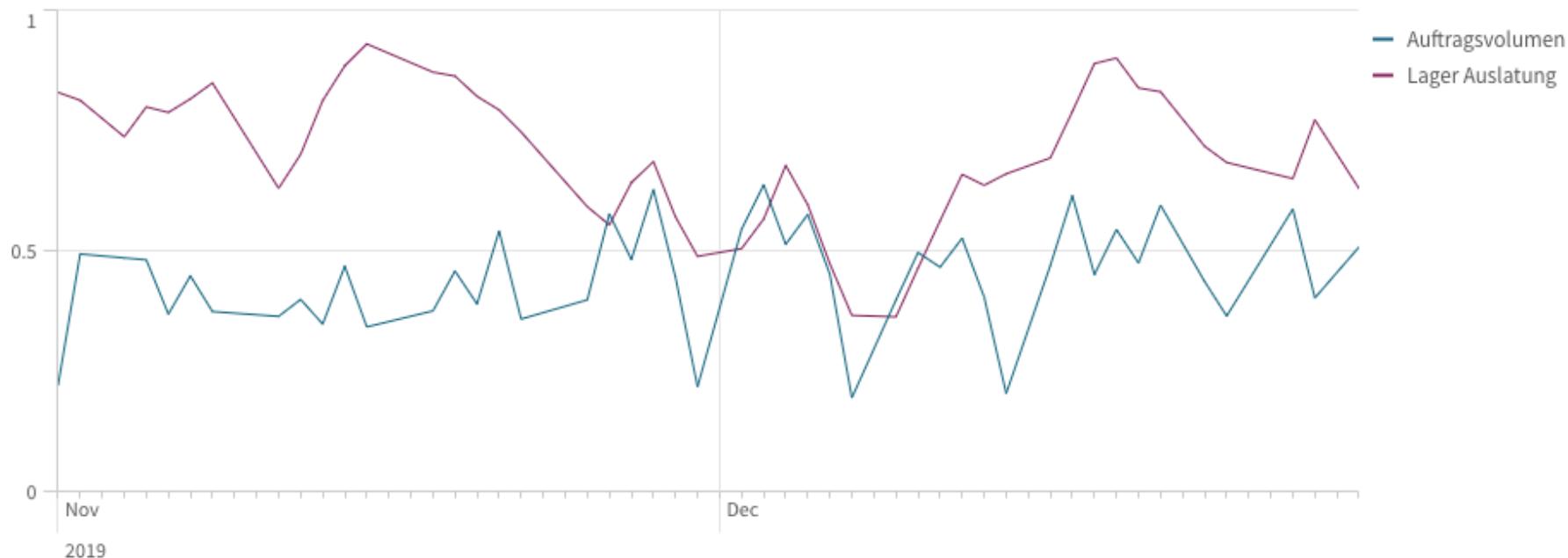
Anhang 9  
Liniendiagramm, Kommissionierte Filialen



Anhang 10  
Liniendiagramm Eingelagerte TE



Anhang 11  
Liniendiagramm Lagerkapazitäten



Anhang 12  
Modell Evaluation,  
Ausreißer über gesamten Datensatz entfernt



## Bestellzahlen Vorhersage Alle Daten Reporting 2 DPS

Step	RMSE	MAE	MAPE
1	8830	5771	0.24
2	8796	5812	0.23
3	8991	5836	0.24
4	8595	5758	0.24
5	9035	6048	0.25
6	8792	5903	0.25
7	8810	6132	0.25

Wochentag	RMSE	MAE	MAPE
Montag	8622	7070	0.16
Dienstag	7860	4782	0.08
Mittwoch	5647	4209	0.07
Donnerstag	14511	9254	0.22
Freitag	8139	5535	0.14
Samstag	7174	4895	0.9

Wochentag	Mean_r	Std_r	Q50_r
Montag	47296	5240	46707
Dienstag	63380	9148	63042
Mittwoch	62338	5137	62745
Donnerstag	71396	13739	71608
Freitag	49473	7884	50298
Samstag	31459	6891	32974

Mean_p	Std_p	Q50_p
52511	4574	52465
62362	3475	61374
61488	3586	60026
71024	3607	71649
50225	3350	50771
31674	4652	30984

Anhang 13  
Modell Evaluation,  
Chronologische Sequenz von Zeitschritten als Feature

Modelle mit der chronologischen Abfolge von Zeitschritten als Trainingsgrundlage

**Features: 50 Zeitschritte vor Prognosezeitraum**

Architekturen	RMSE	MAE	MAPE
CNN	10182	7863	0.16
LSTM	17414	14035	0.27
LSTM-Stacked	33258	26339	0.51
BI-LSTM	55435	53744	1.0
CNN-LSTM	10350	8219	0.17
ED-LSTM	11906	9860	0.21
ED-LSTM-CNN	14440	12033	0.24
ED-ConvLSTM	9864	7759	0.16

**Features: 100 Zeitschritte vor Prognosezeitraum**

Architekturen	RMSE	MAE	MAPE
CNN	10954	8598	0.17
LSTM	25059	18985	0.37
LSTM-Stacked	90965	61873	1.19
BI-LSTM	55457	53684	1.0
CNN-LSTM	10350	8219	0.17
ED-LSTM	12997	10493	0.22
ED-LSTM-CNN	15032	12273	0.26
ED-ConvLSTM	11199	9176	0.19

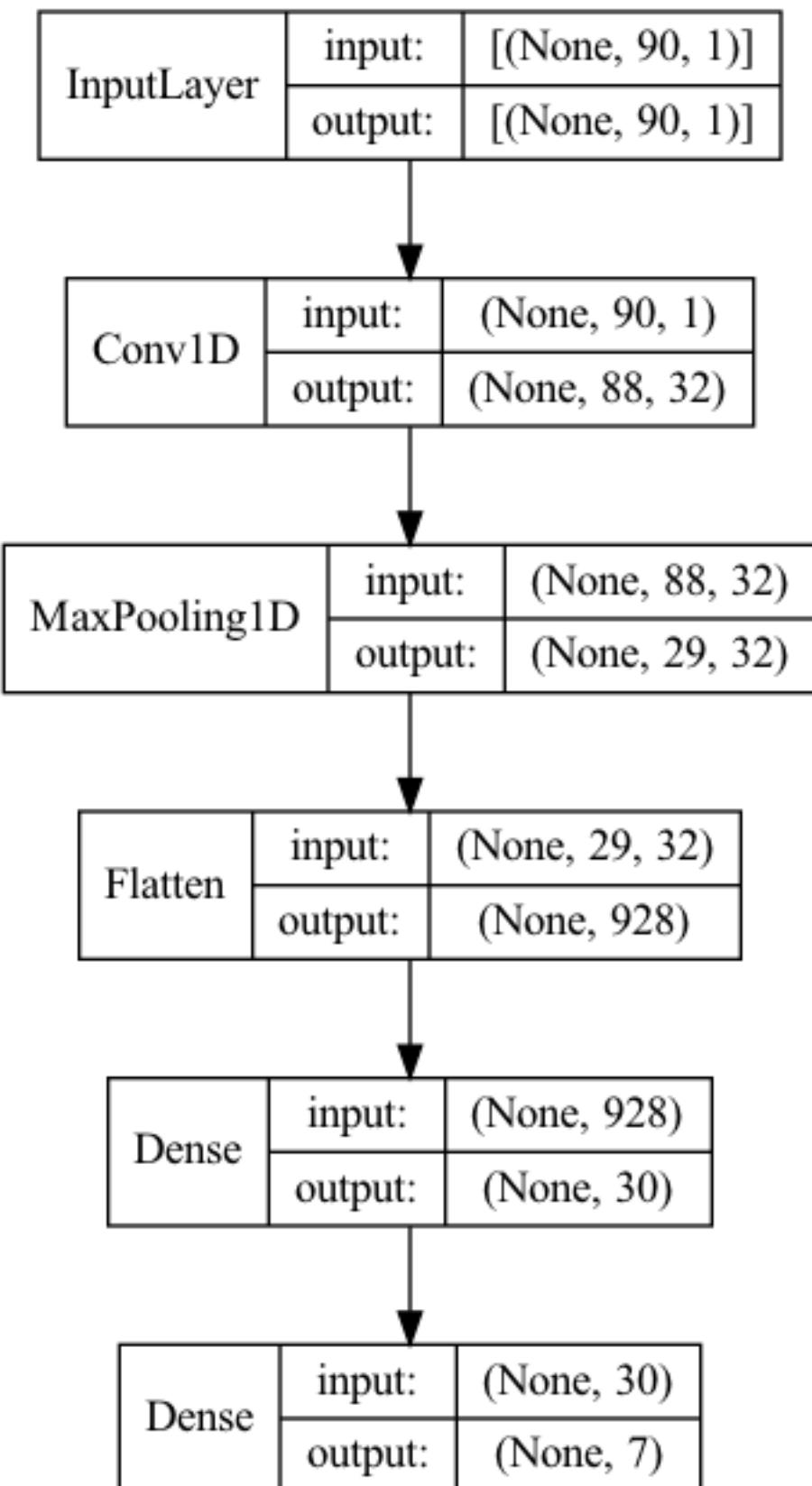
**Features: 200 Zeitschritte vor Prognosezeitraum**

Architekturen	RMSE	MAE	MAPE
CNN	10796	8856	0.18
LSTM	NaN	NaN	NaN
LSTM-Stacked	NaN	NaN	NaN
BI-LSTM	NaN	NaN	NaN
CNN-LSTM	12421	9927	0.22
ED-LSTM	12961	11020	0.23
ED-LSTM-CNN	17043	13836	0.3
ED-ConvLSTM	11682	9527	0.19

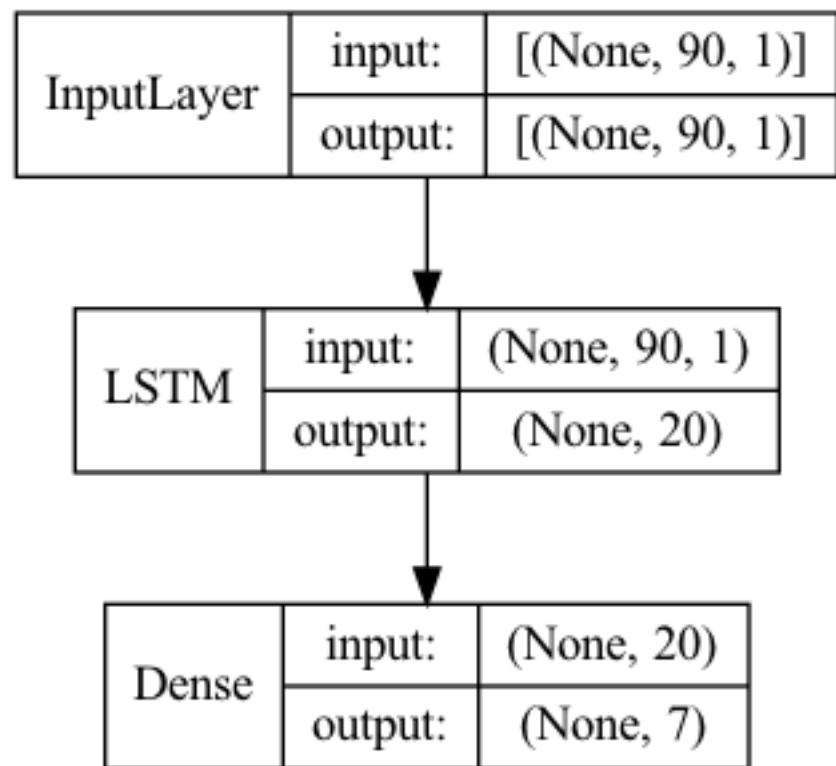
**Features: 400 Zeitschritte vor Prognosezeitraum**

Architekturen	RMSE	MAE	MAPE
CNN	11682	9527	0.19
LSTM	NaN	NaN	NaN
LSTM-Stacked	NaN	NaN	NaN
BI-LSTM	NaN	NaN	NaN
CNN-LSTM	55676	53840	1.0
ED-LSTM	12008	9821	0.2
ED-LSTM-CNN	17411	14180	0.3
ED-ConvLSTM	11769	9274	0.2

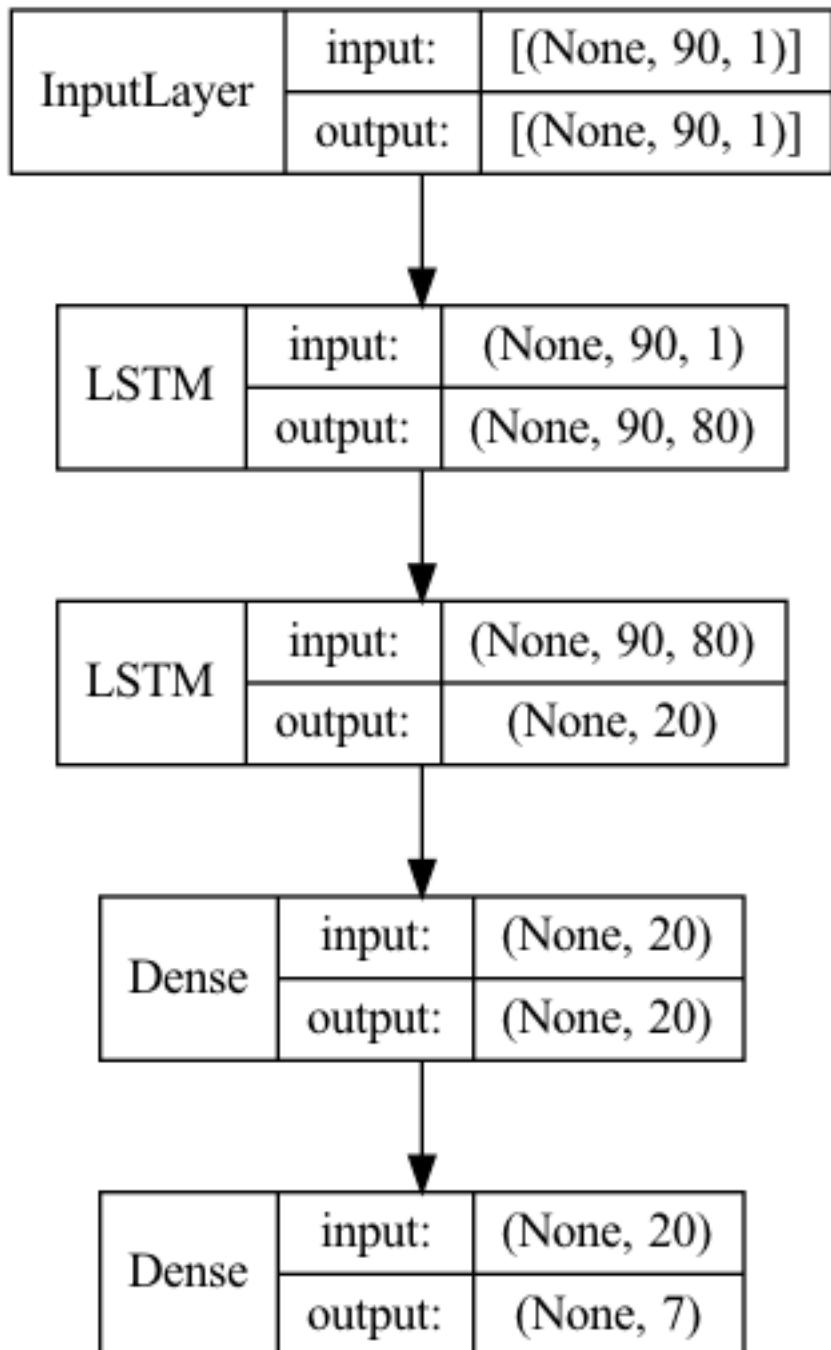
Anhang 14  
Aufbau Sequenzielles CNN



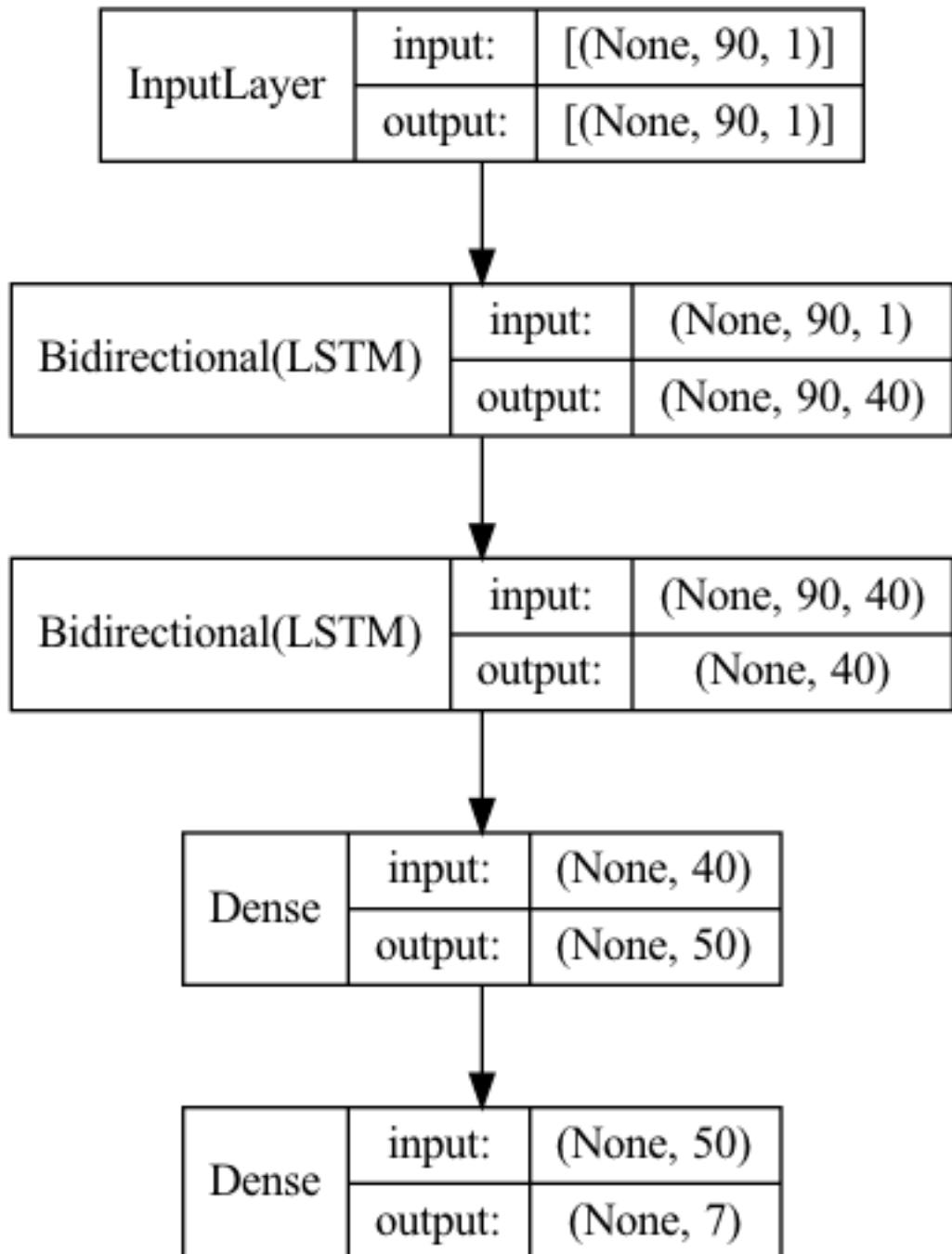
Anhang 15  
Aufbau Sequenzielles LSTM



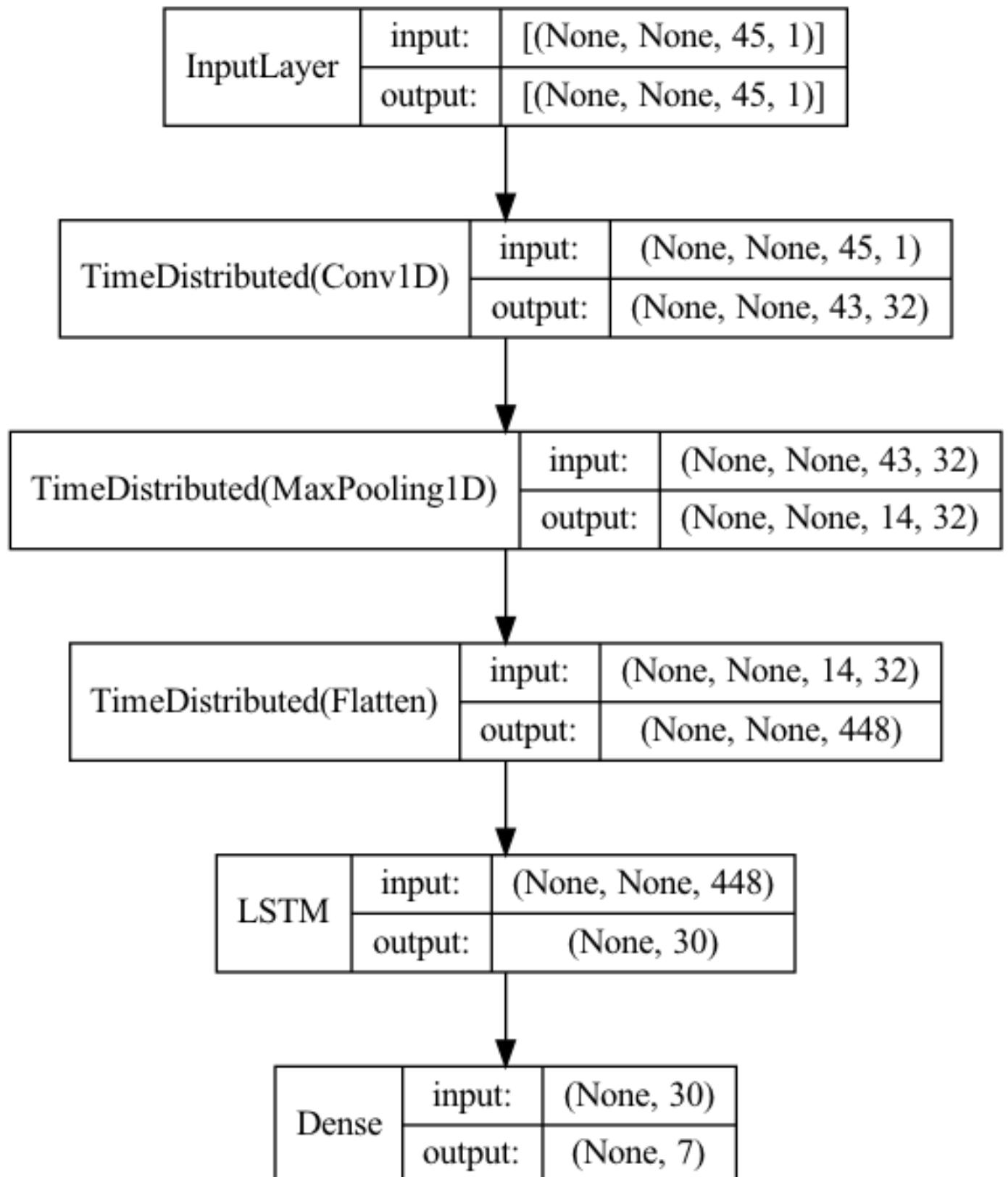
Anhang 16  
Aufbau Sequenzielles Mehrschichtiges-LSTM



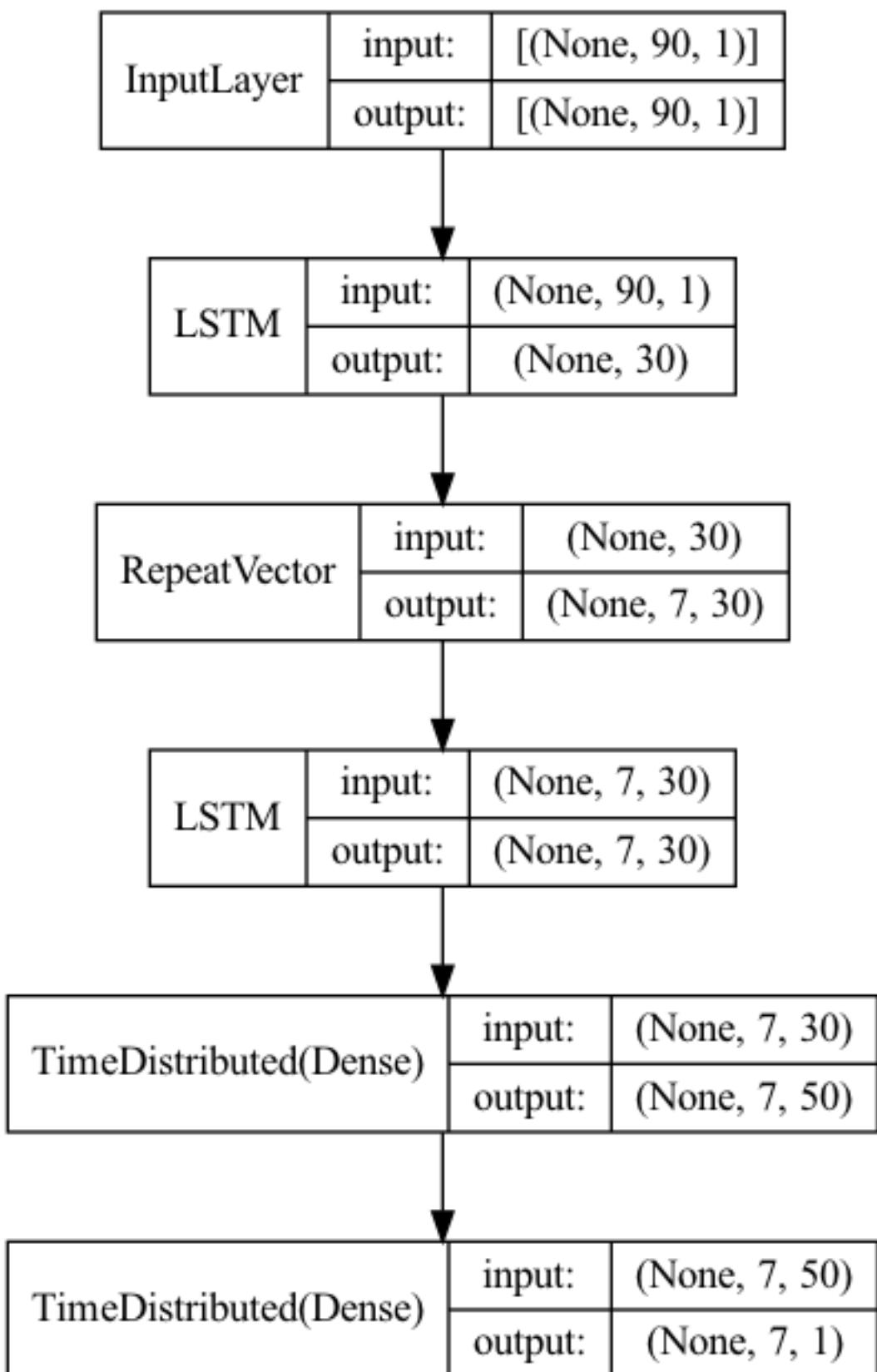
Anhang 17  
Aufbau Sequenzielles Bidirektionales-LSTM



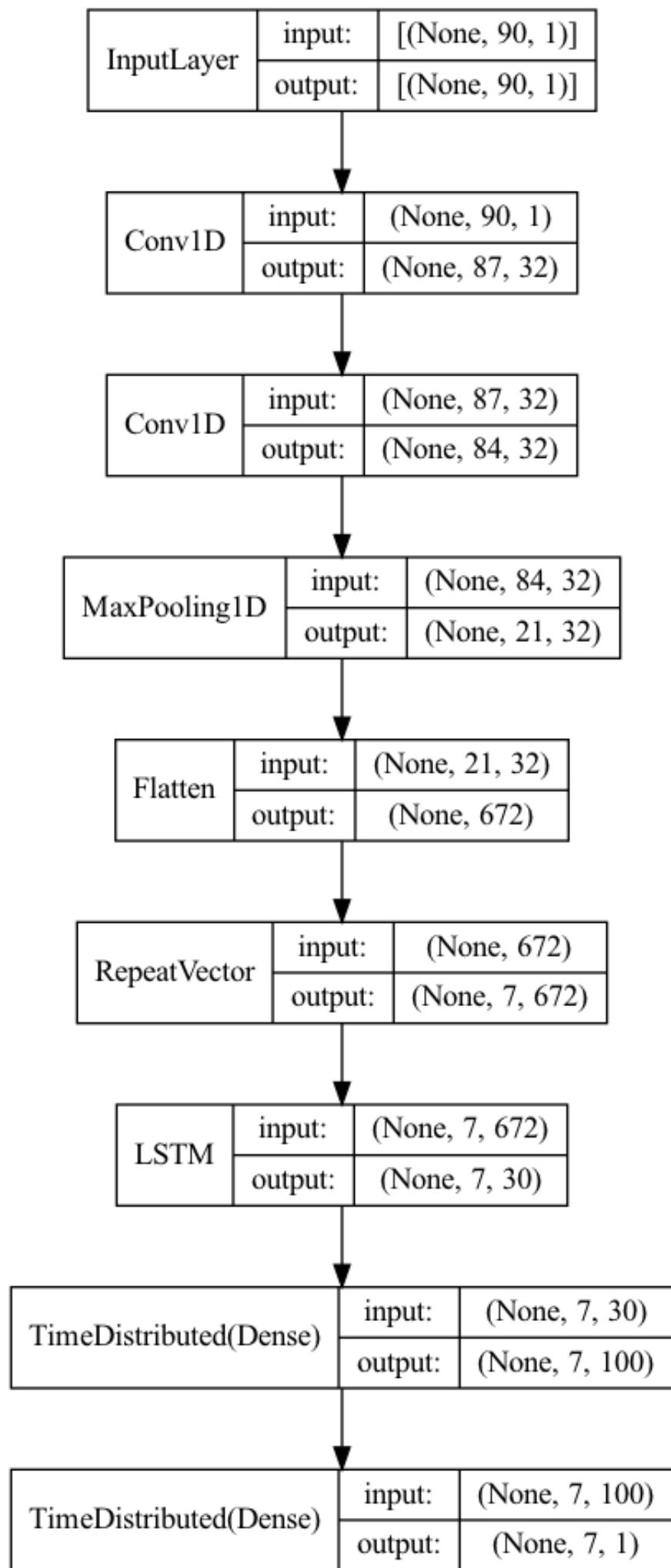
Anhang 18  
Aufbau Sequenzielles CNN-LSTM



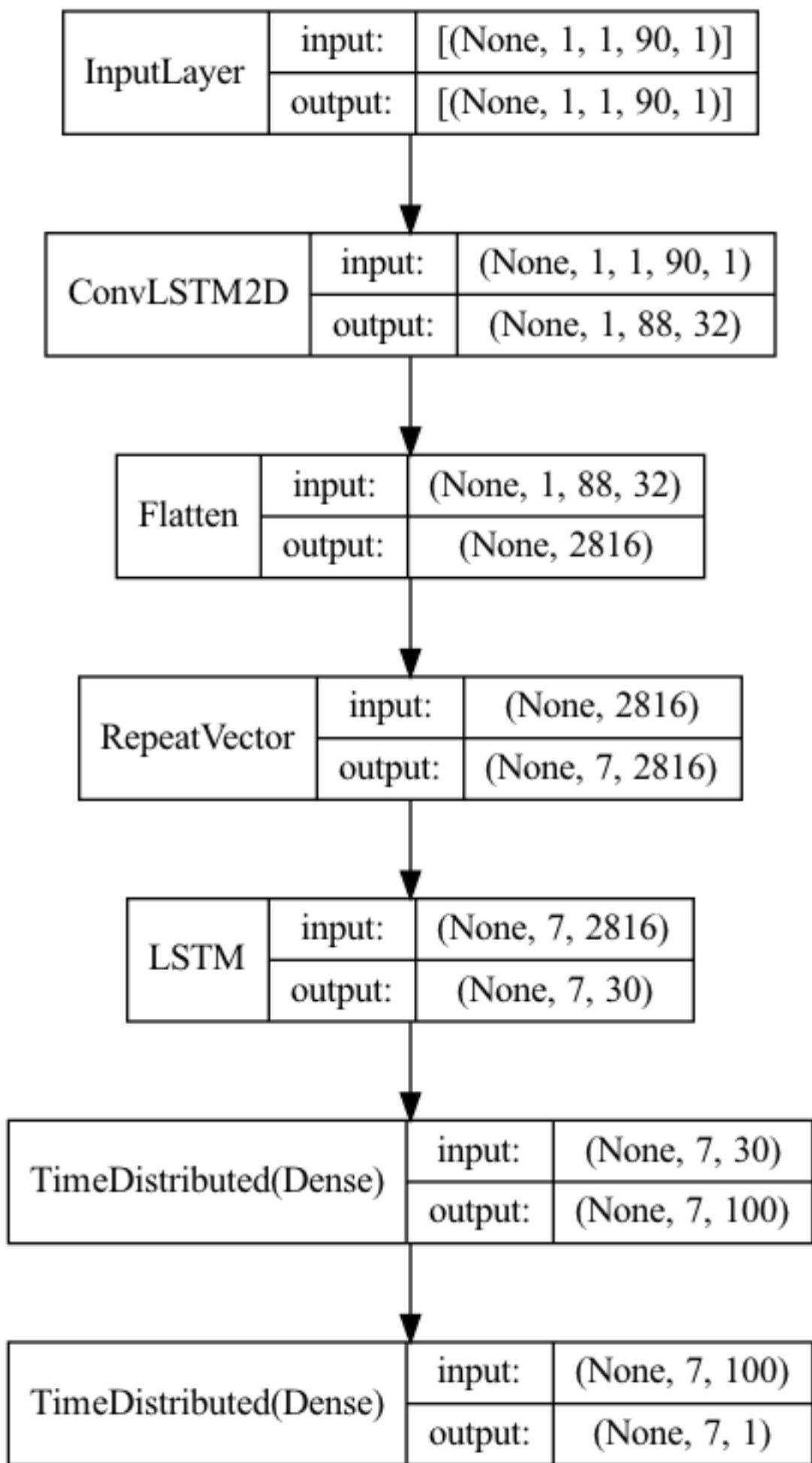
Anhang 19  
Aufbau Sequenzielles Encoder-Decoder-LSTM



Anhang 20  
Aufbau Sequenzielles Encoder-Decoder-LSTM-CNN



Anhang 21  
Aufbau Sequenzielles Encoder-Decoder-ConvLstm



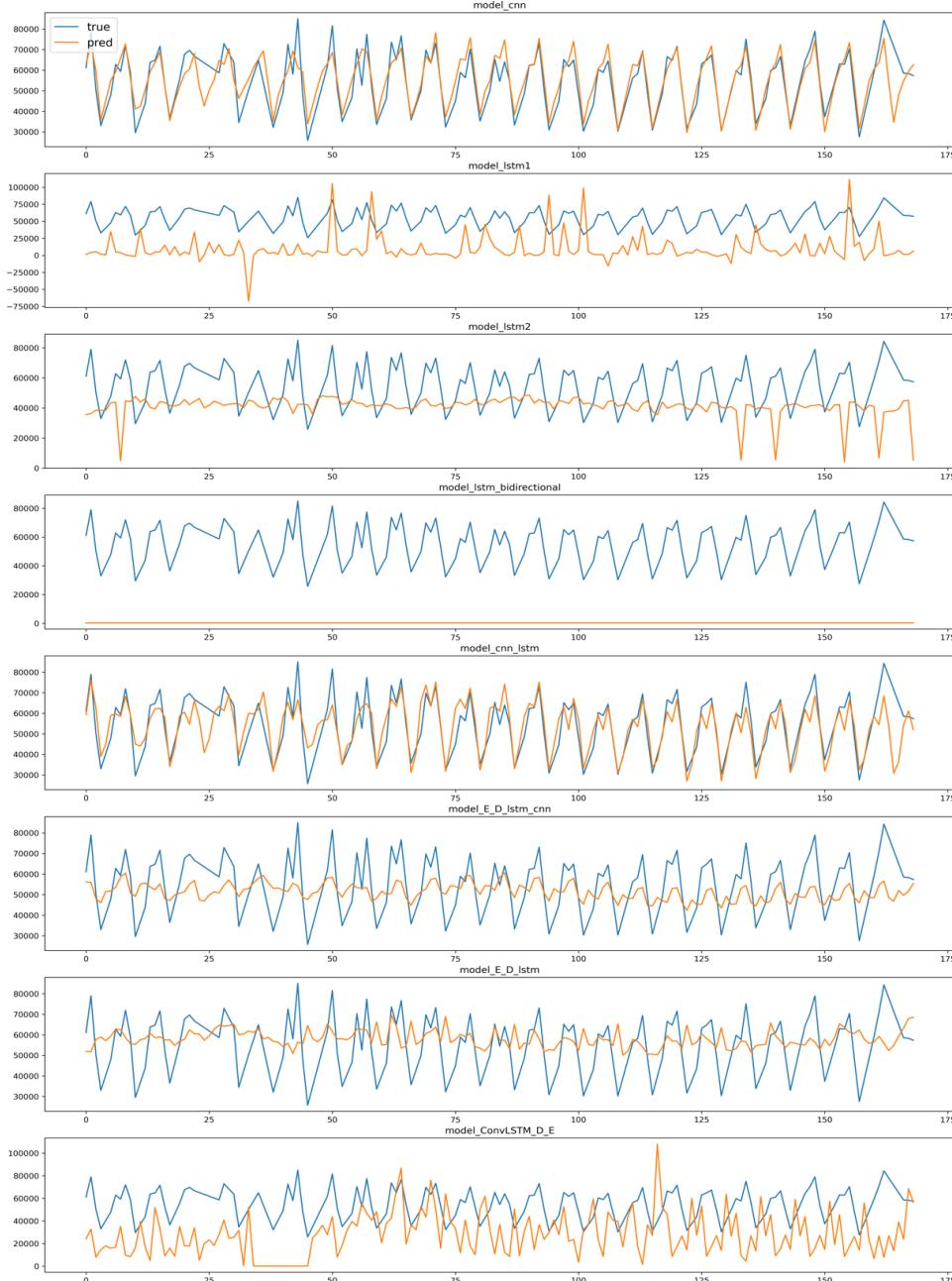
Anhang 22  
Ergebnisse Sequenzielle Modelle  
Trainingsgrundlage: Historische Auftragsdaten

# Erste Eingabesequenz, Sequenzielle Modelle

## Fehlermetriken

Architekturen	RMSE	MAE	MAPE
CNN	8223	6007	0.11
LSTM	24346	18722	0.36
LSTM-Stacked	16101	13356	0.24
BI-LSTM	41811	39542	0.72
CNN-LSTM	8290	6989	0.14
ED-LSTM	14917	12046	0.25
ED-LSTM-CNN	11408	9347	0.2
ED-ConvLSTM	10588	8477	0.16

## Prognosen Evaluierungsdatensatz



## Optimierte Hyperparameter Konfiguration

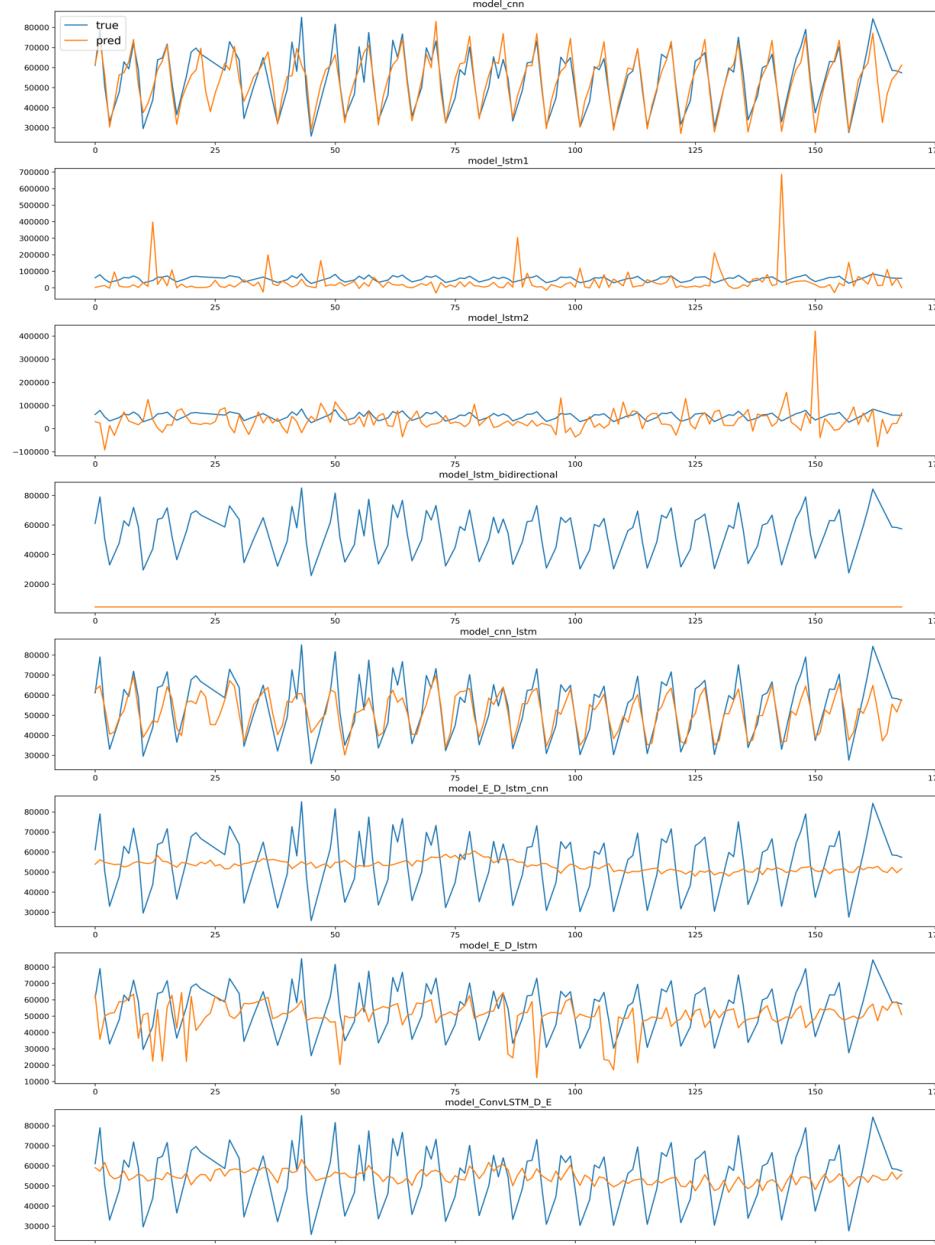
Architekturen	parameters
CNN	{'epochs': 60, 'batch_size': 128, 'filters': 16, 'kernel_size': 5, 'pool_size': 1, 'loss': 'huber', 'optimizer': 'adam'}
CNN_LSTM	{'filters': 64, 'lstm_units': 200, 'loss': 'huber', 'kernel_size': 4, 'dense_units': 200, 'epochs': 90, 'batch_size': 256, 'optimizer': 'adam'}
ED_ConvLSTM	{'filters': 64, 'kernel_size': 5, 'pool_size': 2, 'lstm_units': 110, 'loss': 'mse', 'epochs': 70, 'batch_size': 128, 'optimizer': 'adam'}

## Zweite Eingabesequenz, Sequenzielle Modelle

### Fehlermetriken

Architekturen	RMSE	MAE	MAPE
CNN	8562	6533	0.12
LSTM	20835	17269	0.3
LSTM-Stacked	14076	11789	0.24
BI-LSTM	41807	39537	0.72
CNN-LSTM	9301	7122	0.14
ED-LSTM	13681	11565	0.24
ED-LSTM-CNN	12083	9964	0.21
ED-ConvLSTM	11245	9160	0.19

### Prognosen Evaluierungsdatensatz



### Optimierte Hyperparameter Konfiguration

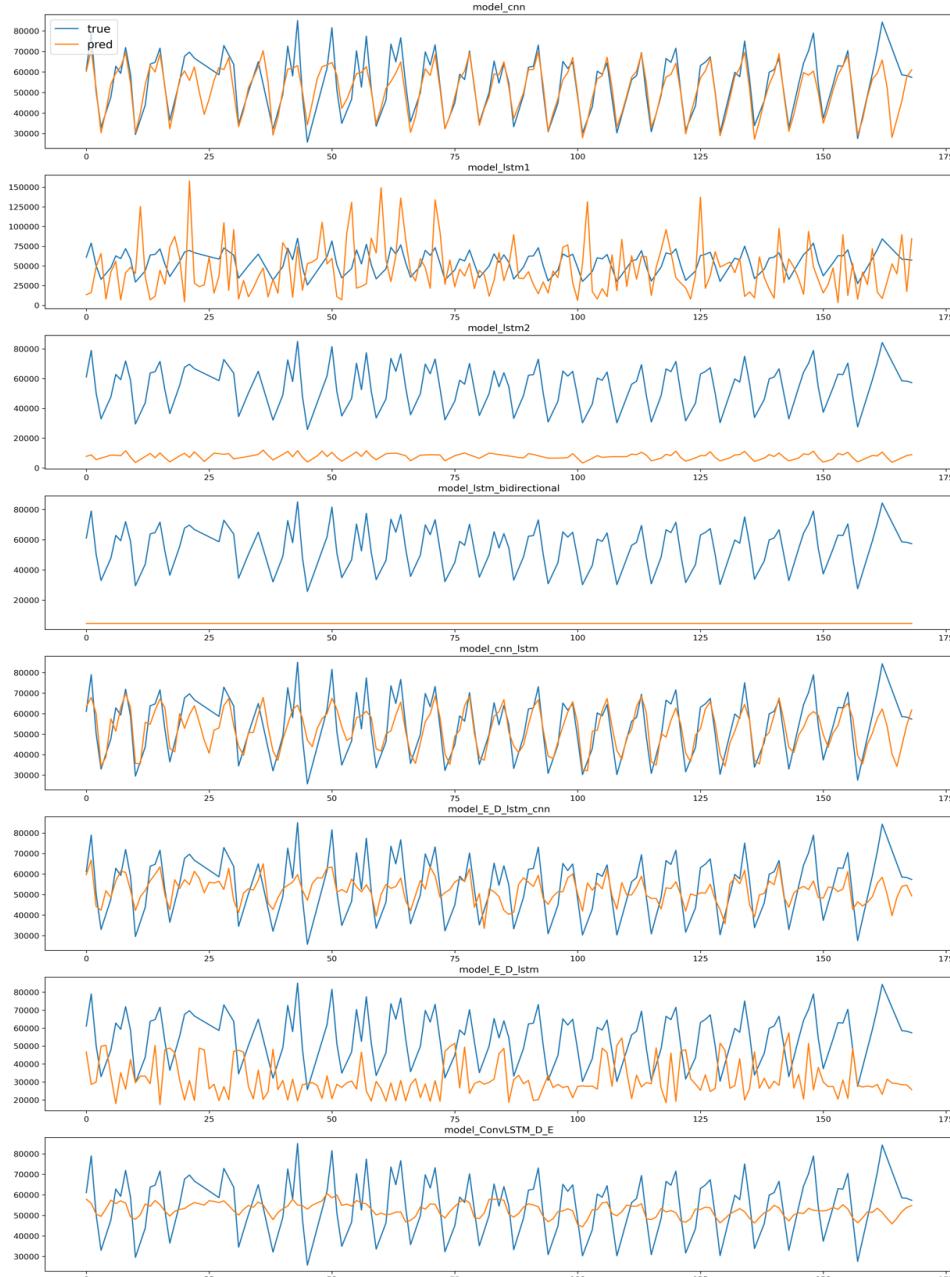
Architekturen	parameters
CNN	{'epochs': 70, 'batch_size': 32, 'filters': 8, 'kernel_size': 3, 'pool_size': 1, 'loss': 'mse', 'optimizer': 'adam'}
CNN_LSTM	{'filters': 128, 'lstm_units': 250, 'loss': 'mse', 'kernel_size': 5, 'dense_units': 210, 'epochs': 70, 'batch_size': 256, 'optimizer': 'adam'}
ED_ConvLSTM	{'filters': 128, 'kernel_size': 6, 'pool_size': 2, 'lstm_units': 170, 'loss': 'huber', 'epochs': 80, 'batch_size': 256, 'optimizer': 'adam'}

## Dritte Eingabesequenz, Sequenzielle Modelle

### Fehlermetriken

Architekturen	RMSE	MAE	MAPE
CNN	8362	5784	0.11
LSTM	14463	12135	0.24
LSTM-Stacked	19073	15224	0.3
BI-LSTM	52188	53489	0.99
CNN-LSTM	8315	6011	0.11
ED-LSTM	12083	16741	0.35
ED-LSTM-CNN	11845	9479	0.19
ED-ConvLSTM	10799	8754	0.17

### Prognosen Evaluierungsdatensatz



### Optimierte Hyperparameter Konfiguration

Architekturen	parameters
CNN	{'epochs': 70, 'batch_size': 64, 'filters': 32, 'kernel_size': 2, 'pool_size': 1, 'loss': 'mse', 'optimizer': 'adam'}
CNN_LSTM	{'filters': 64, 'lstm_units': 150, 'loss': 'mse', 'kernel_size': 5, 'dense_units': 150, 'epochs': 100, 'batch_size': 256, 'optimizer': 'adam'}
ED_ConvLSTM	{'filters': 256, 'kernel_size': 4, 'pool_size': 2, 'lstm_units': 150, 'loss': 'huber', 'epochs': 100, 'batch_size': 256, 'optimizer': 'adam'}

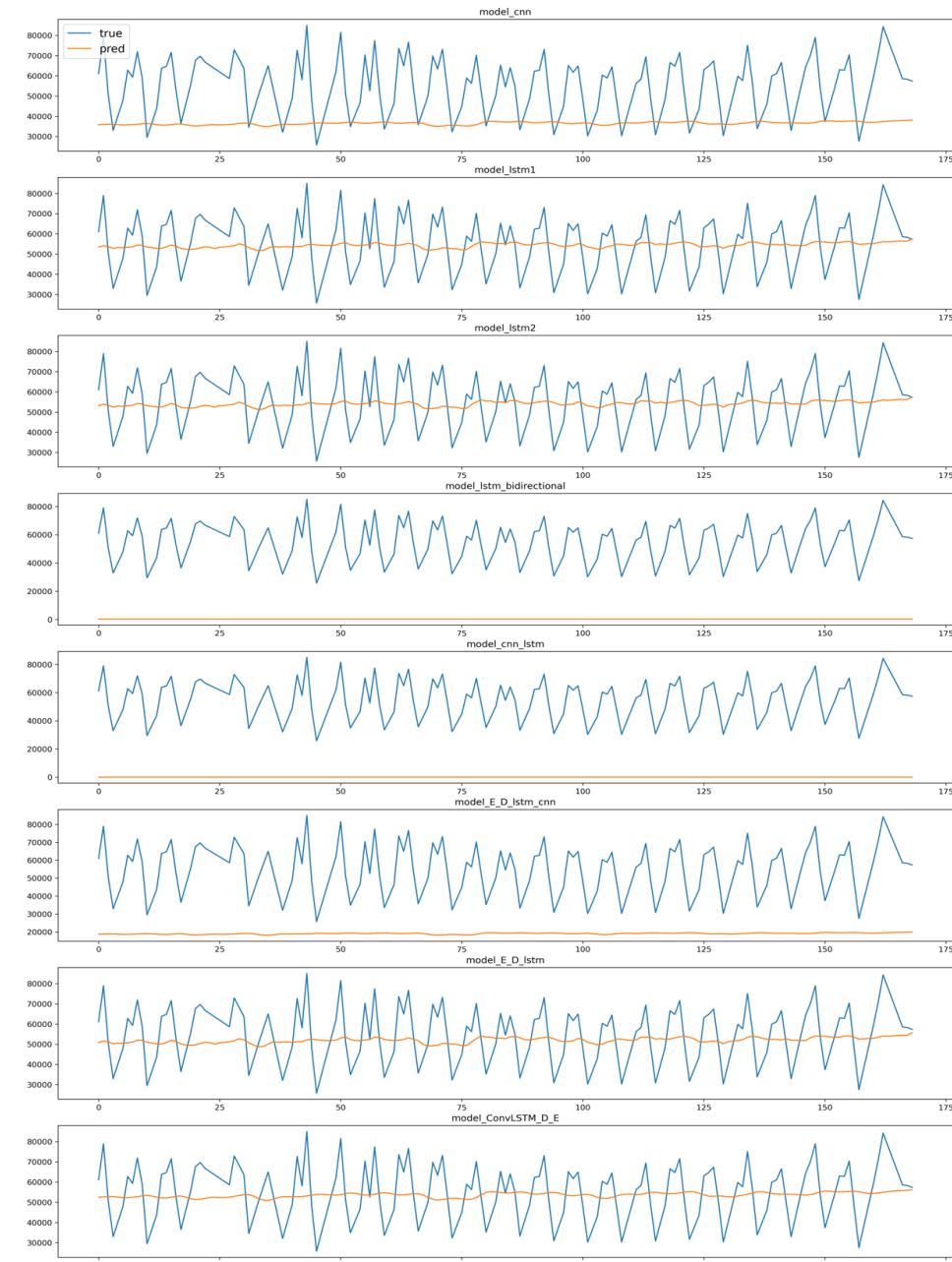
Anhang 23  
Ergebnisse Sequenzielle Modelle  
Trainingsgrundlage: Weitere Datensätze

## Vierte Eingabesequenz, Sequenzielle Modelle

### Fehlermetriken

Architekturen	RMSE	MAE	MAPE
CNN	22148	18537	0.31
LSTM	13668	11572	0.25
LSTM-Stacked	13628	11552	0.24
BI-LSTM	55180	53481	0.99
CNN-LSTM	55492	53803	1.0
ED-LSTM	19141	15019	0.3
ED-LSTM-CNN	13693	11609	0.24
ED-ConvLSTM	13650	11580	0.24

### Prognosen Evaluierungsdatensatz



### Optimierte Hyperparameter Konfiguration

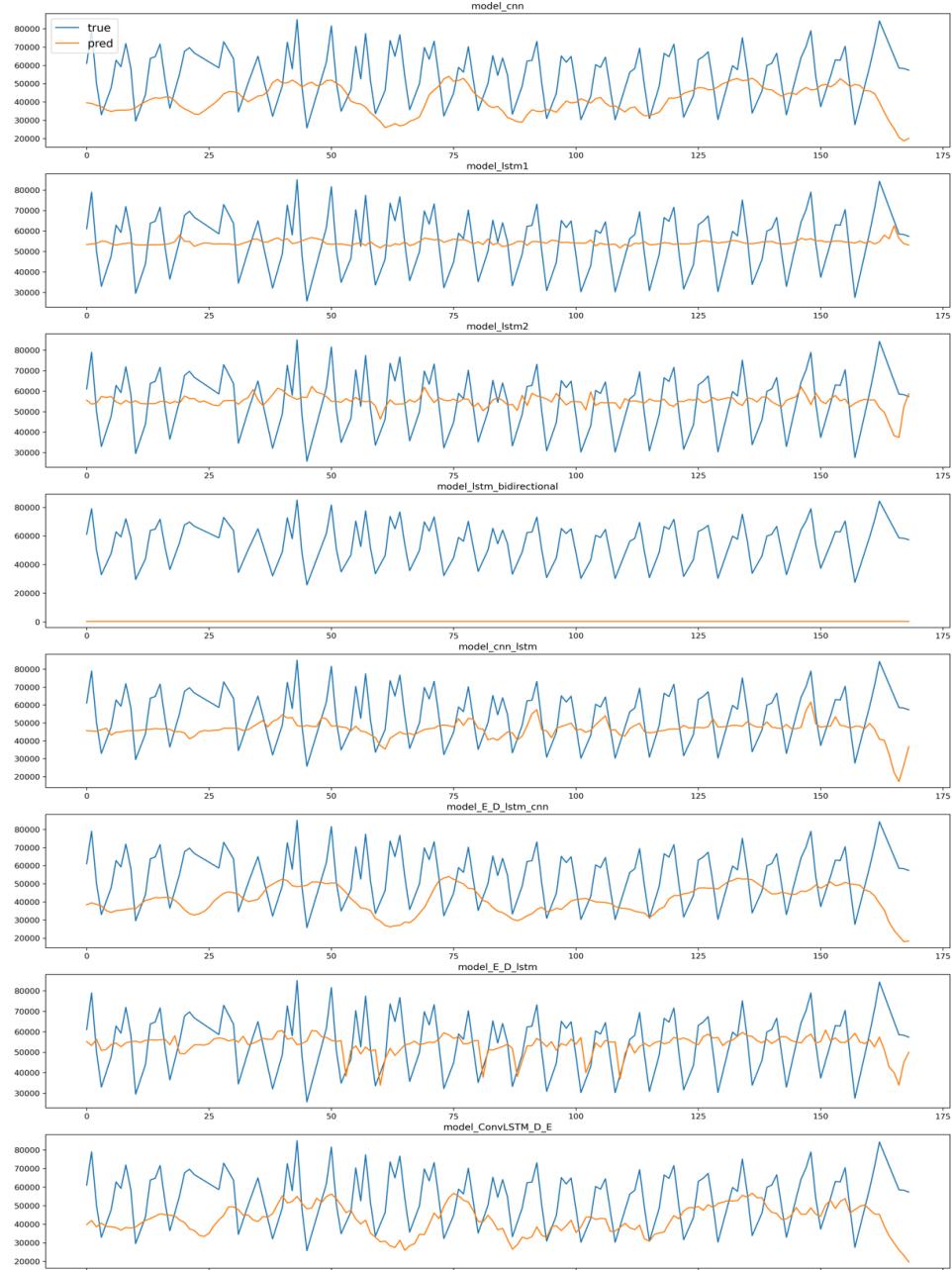
Architekturen	parameters
LSTM_Stacked	{lstm_units1 : 30, lstm_units2 : 30, optimizer : 'adam', loss : 'mse', epochs : 10, batch_size : 32, 'optimizer': 'adam'}
LSTM	{lstm_units : 45, optimizer : 'adam', loss : 'mse', epochs : 10, batch_size : 32, 'optimizer': 'adam'}
ED_ConvLSTM	{filters : 128, kernel_size : 2, pool_size : 1 lstm_units : 90, loss : 'huber', epochs : 70, batch_size : 256, 'optimizer': 'adam'}

## Fünfte Eingabesequenz, Sequenzielle Modelle

### Fehlermetriken

Architekturen	RMSE	MAE	MAPE
CNN	19774	16376	0.29
LSTM	13934	11798	0.25
LSTM-Stacked	13593	11493	0.24
BI-LSTM	55189	53490	0.99
CNN-LSTM	20384	16671	0.3
ED-LSTM	14566	12227	0.26
ED-LSTM-CNN	20158	16458	0.29
ED-ConvLSTM	15259	12661	0.24

### Prognosen Evaluierungsdatensatz



### Optimierte Hyperparameter Konfiguration

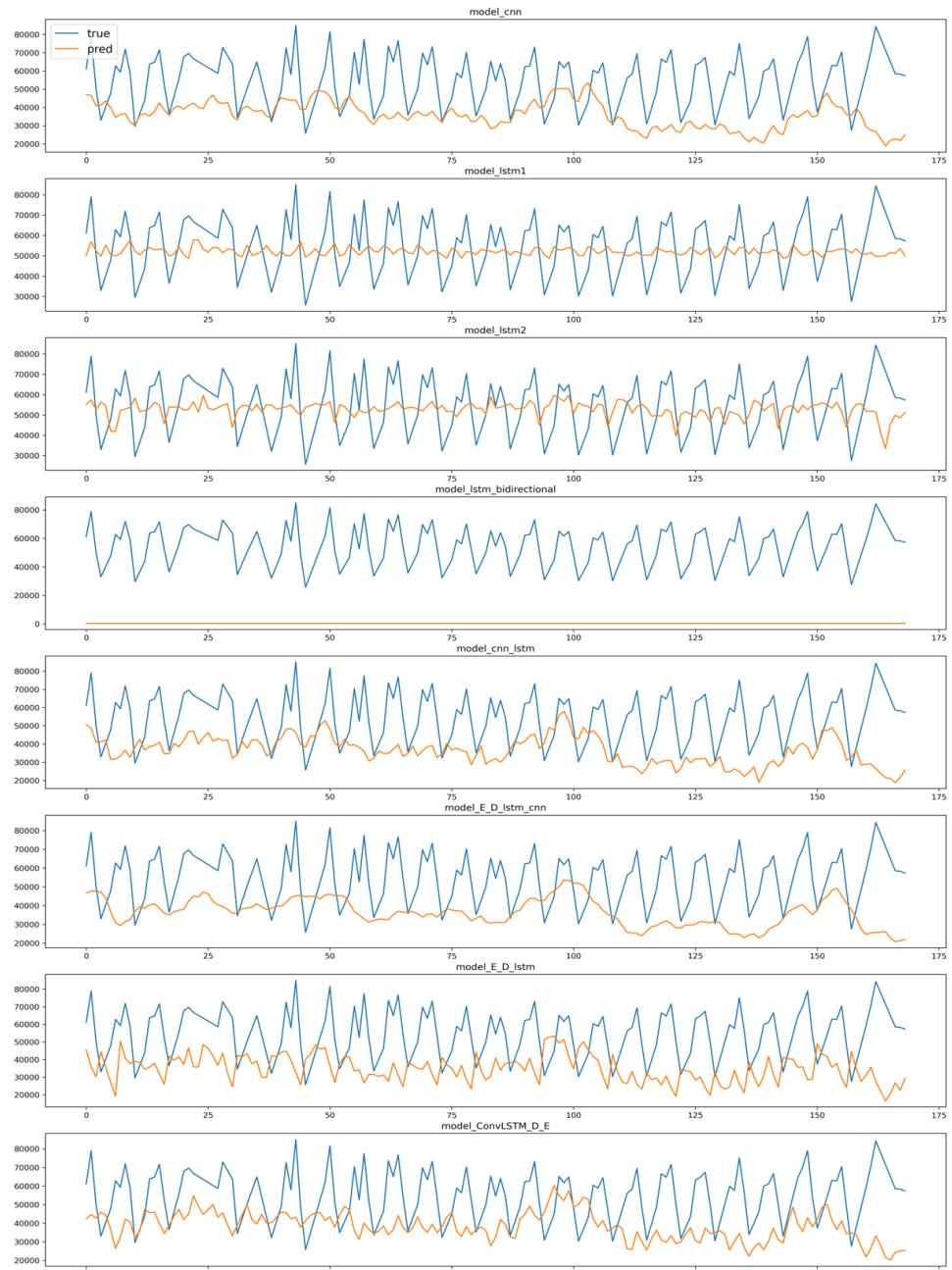
Architekturen	parameters
LSTM_Stacked	{lstm_units1 : 60, lstm_units2 : 40, optimizer : 'adam', loss : 'mse', epochs : 60, batch_size : 256, 'optimizer': 'adam'}
LSTM	{lstm_units : 40, optimizer : 'adam', loss : 'mse', epochs : 40, batch_size : 128, 'optimizer': 'adam'}
ED_LSTM	{lstm_units : 30, optimizer : 'adam', loss : 'huber', epochs : 60, batch_size : 128, 'optimizer': 'adam'}

## Sechste Eingabesequenz, Sequenzielle Modelle

### Fehlermetriken

Architekturen	RMSE	MAE	MAPE
CNN	19812	16179	0.29
LSTM	13621	11492	0.24
LSTM-Stacked	13916	11807	0.24
BI-LSTM	55190	53492	0.99
CNN-LSTM	20384	16671	0.3
ED-LSTM	20469	16706	0.29
ED-LSTM-CNN	13974	11773	0.25
ED-ConvLSTM	15259	12661	0.24

### Prognosen Evaluierungsdatensatz



### Optimierte Hyperparameter Konfiguration

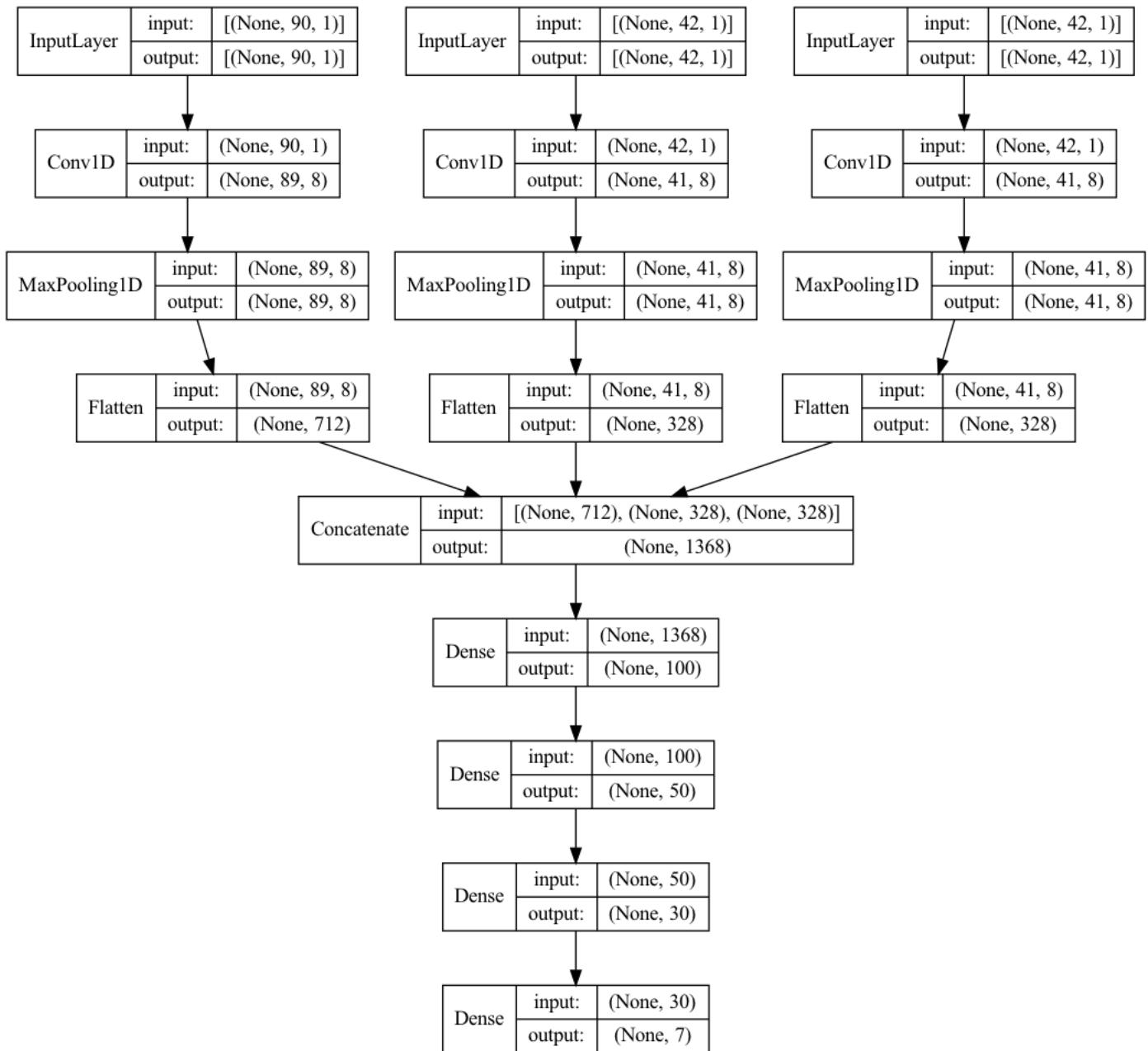
Architekturen	parameters
LSTM	{lstm_units : 30, optimizer : 'adam', loss : 'mse', epochs : 60, batch_size : 256, 'optimizer': 'adam'}
LSTM_Stacked	{lstm_units1 : 50, lstm_units2 : 40, optimizer : 'adam', loss : 'huber', epochs : 50, batch_size : 256, 'optimizer': 'adam'}
ED_LSTM_CNN	{filters: 128, lstm_units: 110, 'loss':'huber', 'kernel_size': 3, dense_units: 100, 'epochs': 60, 'batch_size': 128, 'optimizer': 'adam'}

Anhang 24  
Darstellung Optimierungsprozess

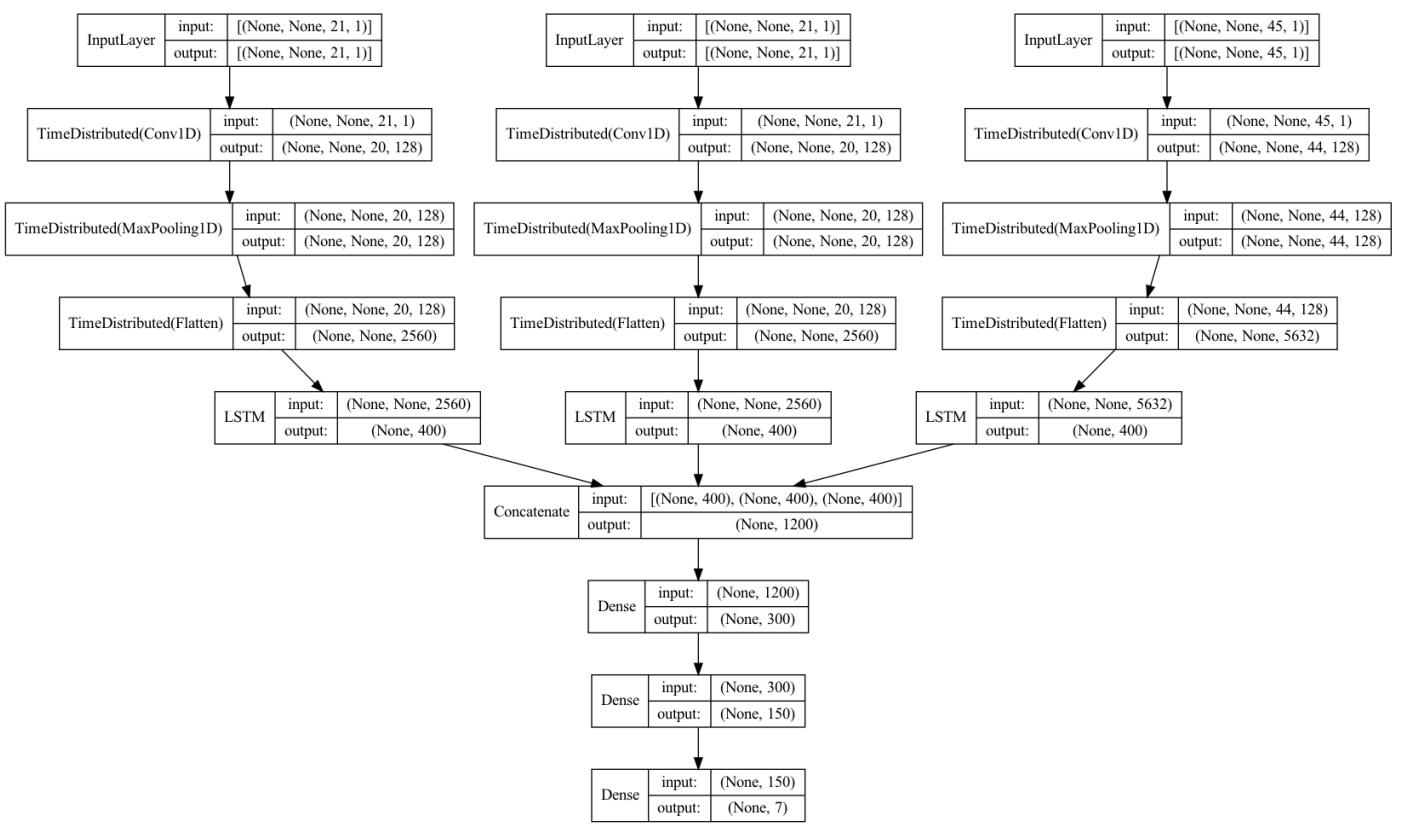
## Optimierung Random-Search: Erste Eingabesequenz / CNN-Modell

Iteration	Modell	RMSE	MAE	MAPE	Hyperparameter
38	CNN1	8223	5921	0,1	{'epochs': 60, 'batch_size': 128, 'filters': 16, 'kernel_size': 5, 'pool_size': 1, 'loss': 'huber'}
25	CNN1	8298	5935	0,11	{'epochs': 70, 'batch_size': 32, 'filters': 8, 'kernel_size': 3, 'pool_size': 1, 'loss': 'mse'}
41	CNN1	8250	5975	0,11	{'epochs': 70, 'batch_size': 64, 'filters': 32, 'kernel_size': 2, 'pool_size': 1, 'loss': 'mse'}
18	CNN1	8223	5996	0,11	{'epochs': 60, 'batch_size': 128, 'filters': 32, 'kernel_size': 5, 'pool_size': 1, 'loss': 'huber'}
5	CNN1	9142	6809	0,13	{'epochs': 60, 'batch_size': 32, 'filters': 16, 'kernel_size': 5, 'pool_size': 1, 'loss': 'huber'}
10	CNN1	9396	7076	0,14	{'epochs': 50, 'batch_size': 64, 'filters': 16, 'kernel_size': 3, 'pool_size': 1, 'loss': 'mse'}
0	CNN1	9468	7083	0,13	{'epochs': 100, 'batch_size': 32, 'filters': 16, 'kernel_size': 5, 'pool_size': 1, 'loss': 'huber'}
20	CNN1	9431	7166	0,14	{'epochs': 90, 'batch_size': 64, 'filters': 8, 'kernel_size': 3, 'pool_size': 2, 'loss': 'huber'}
11	CNN1	9357	7175	0,14	{'epochs': 80, 'batch_size': 64, 'filters': 8, 'kernel_size': 5, 'pool_size': 2, 'loss': 'huber'}
19	CNN1	9477	7183	0,14	{'epochs': 100, 'batch_size': 32, 'filters': 8, 'kernel_size': 4, 'pool_size': 1, 'loss': 'huber'}
35	CNN1	9291	7237	0,15	{'epochs': 100, 'batch_size': 128, 'filters': 32, 'kernel_size': 2, 'pool_size': 2, 'loss': 'huber'}
44	CNN1	9300	7327	0,15	{'epochs': 50, 'batch_size': 64, 'filters': 32, 'kernel_size': 4, 'pool_size': 3, 'loss': 'mae'}
42	CNN1	9485	7416	0,15	{'epochs': 90, 'batch_size': 64, 'filters': 32, 'kernel_size': 1, 'pool_size': 2, 'loss': 'huber'}
32	CNN1	9757	7705	0,16	{'epochs': 90, 'batch_size': 32, 'filters': 4, 'kernel_size': 1, 'pool_size': 2, 'loss': 'mse'}
27	CNN1	9763	7634	0,15	{'epochs': 80, 'batch_size': 128, 'filters': 16, 'kernel_size': 3, 'pool_size': 2, 'loss': 'huber'}
43	CNN1	9772	7693	0,16	{'epochs': 80, 'batch_size': 32, 'filters': 16, 'kernel_size': 4, 'pool_size': 4, 'loss': 'mse'}
16	CNN1	10115	7974	0,16	{'epochs': 90, 'batch_size': 32, 'filters': 16, 'kernel_size': 3, 'pool_size': 5, 'loss': 'mae'}
9	CNN1	10264	8366	0,17	{'epochs': 90, 'batch_size': 128, 'filters': 32, 'kernel_size': 5, 'pool_size': 5, 'loss': 'huber'}
7	CNN1	10310	8175	0,17	{'epochs': 90, 'batch_size': 128, 'filters': 16, 'kernel_size': 4, 'pool_size': 5, 'loss': 'mae'}
14	CNN1	10561	8419	0,18	{'epochs': 100, 'batch_size': 32, 'filters': 4, 'kernel_size': 4, 'pool_size': 3, 'loss': 'huber'}
8	CNN1	10766	8522	0,18	{'epochs': 60, 'batch_size': 64, 'filters': 8, 'kernel_size': 5, 'pool_size': 4, 'loss': 'mae'}
40	CNN1	10788	8819	0,19	{'epochs': 70, 'batch_size': 128, 'filters': 8, 'kernel_size': 4, 'pool_size': 3, 'loss': 'mse'}
36	CNN1	11146	8768	0,16	{'epochs': 90, 'batch_size': 32, 'filters': 16, 'kernel_size': 4, 'pool_size': 5, 'loss': 'mse'}
15	CNN1	11310	8706	0,15	{'epochs': 100, 'batch_size': 64, 'filters': 32, 'kernel_size': 1, 'pool_size': 3, 'loss': 'mae'}
46	CNN1	11349	8679	0,16	{'epochs': 80, 'batch_size': 128, 'filters': 8, 'kernel_size': 3, 'pool_size': 3, 'loss': 'huber'}
3	CNN1	11354	8760	0,15	{'epochs': 50, 'batch_size': 128, 'filters': 32, 'kernel_size': 3, 'pool_size': 4, 'loss': 'mae'}
21	CNN1	11643	9570	0,19	{'epochs': 70, 'batch_size': 32, 'filters': 8, 'kernel_size': 3, 'pool_size': 5, 'loss': 'mse'}
47	CNN1	11877	9330	0,17	{'epochs': 100, 'batch_size': 64, 'filters': 8, 'kernel_size': 5, 'pool_size': 5, 'loss': 'mse'}
24	CNN1	11912	9531	0,18	{'epochs': 100, 'batch_size': 128, 'filters': 8, 'kernel_size': 3, 'pool_size': 3, 'loss': 'huber'}
28	CNN1	12026	9906	0,19	{'epochs': 90, 'batch_size': 128, 'filters': 16, 'kernel_size': 3, 'pool_size': 5, 'loss': 'huber'}
49	CNN1	12449	10434	0,22	{'epochs': 100, 'batch_size': 128, 'filters': 8, 'kernel_size': 2, 'pool_size': 4, 'loss': 'huber'}
34	CNN1	12480	9852	0,18	{'epochs': 70, 'batch_size': 64, 'filters': 8, 'kernel_size': 2, 'pool_size': 4, 'loss': 'mse'}
37	CNN1	12749	10544	0,21	{'epochs': 90, 'batch_size': 128, 'filters': 4, 'kernel_size': 5, 'pool_size': 5, 'loss': 'mae'}
12	CNN1	12811	10658	0,23	{'epochs': 70, 'batch_size': 32, 'filters': 4, 'kernel_size': 2, 'pool_size': 5, 'loss': 'mse'}
23	CNN1	12916	10819	0,23	{'epochs': 100, 'batch_size': 64, 'filters': 32, 'kernel_size': 4, 'pool_size': 6, 'loss': 'mae'}
30	CNN1	13140	10956	0,24	{'epochs': 90, 'batch_size': 64, 'filters': 16, 'kernel_size': 4, 'pool_size': 6, 'loss': 'huber'}
31	CNN1	13148	11085	0,23	{'epochs': 80, 'batch_size': 32, 'filters': 8, 'kernel_size': 4, 'pool_size': 6, 'loss': 'mae'}
33	CNN1	13244	11168	0,24	{'epochs': 70, 'batch_size': 64, 'filters': 16, 'kernel_size': 1, 'pool_size': 5, 'loss': 'huber'}
17	CNN1	13279	11111	0,24	{'epochs': 80, 'batch_size': 64, 'filters': 4, 'kernel_size': 1, 'pool_size': 5, 'loss': 'mae'}
13	CNN1	13289	11148	0,24	{'epochs': 50, 'batch_size': 64, 'filters': 8, 'kernel_size': 1, 'pool_size': 5, 'loss': 'mse'}
2	CNN1	13359	11377	0,24	{'epochs': 100, 'batch_size': 128, 'filters': 4, 'kernel_size': 4, 'pool_size': 4, 'loss': 'huber'}
45	CNN1	13448	11307	0,24	{'epochs': 100, 'batch_size': 128, 'filters': 32, 'kernel_size': 2, 'pool_size': 6, 'loss': 'mse'}

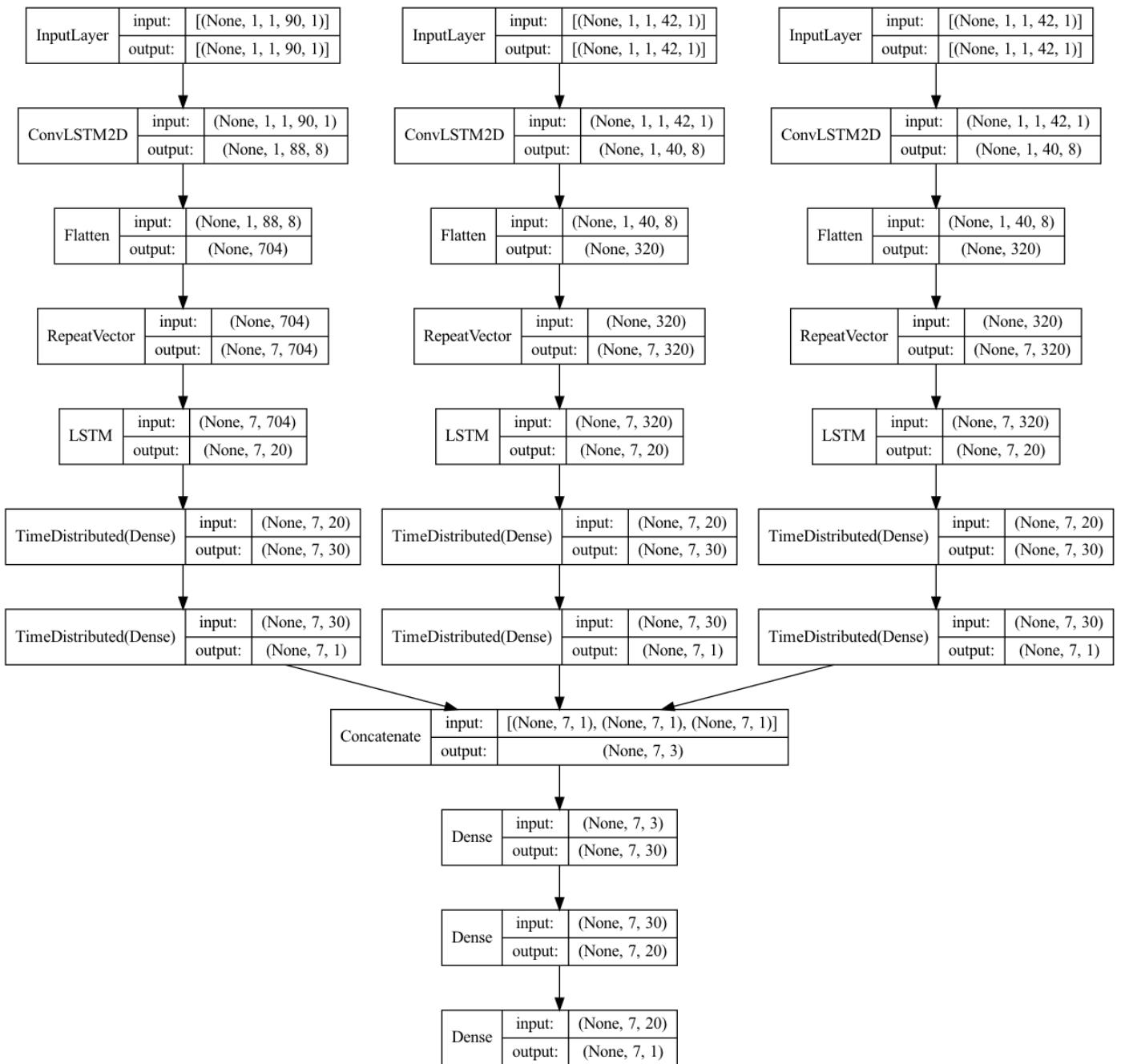
Anhang 25  
Aufbau Multi-Head-3-CNN



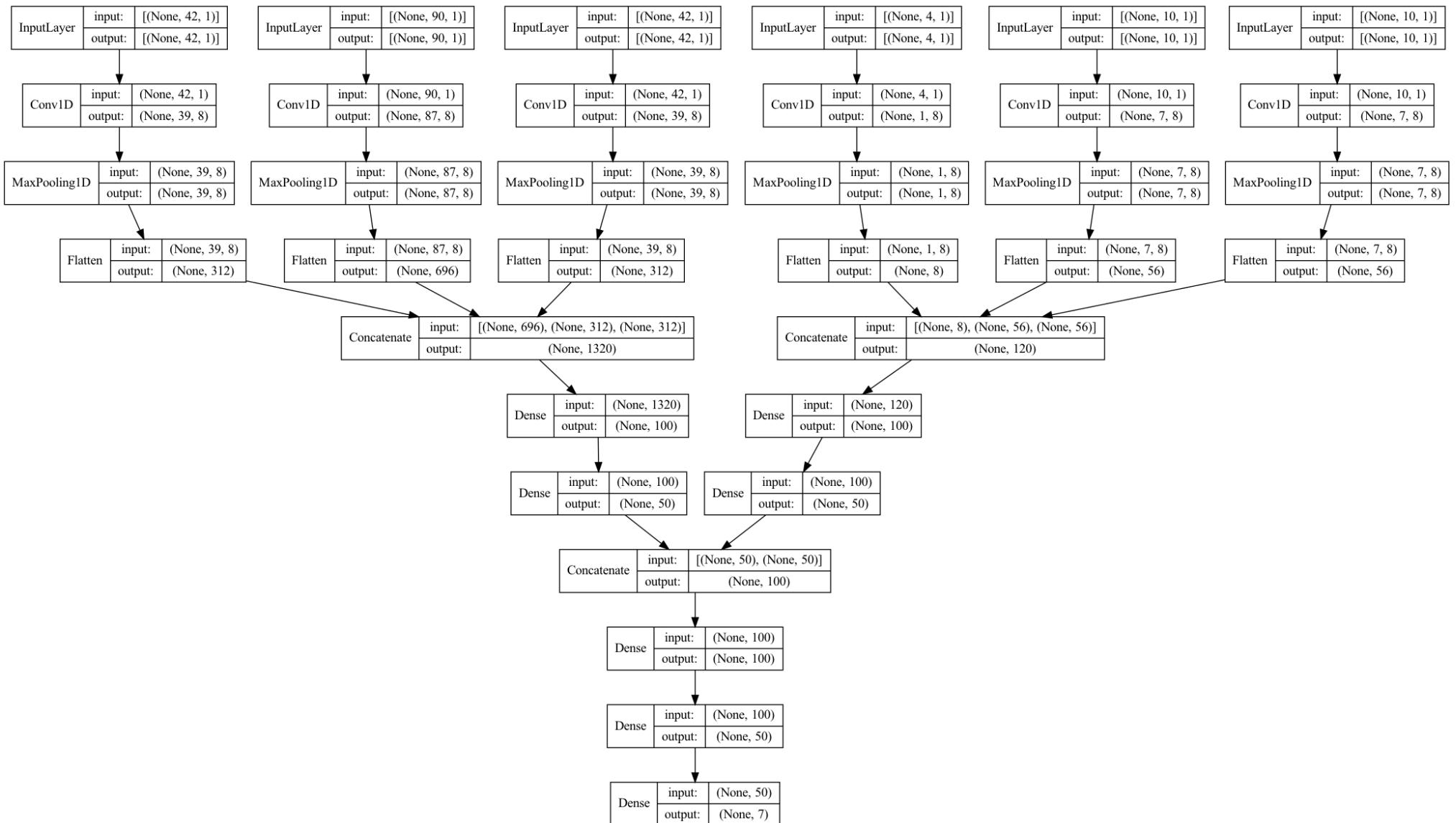
Anhang 26  
Aufbau Mutli-Head-3 CNN-LSTM



Anhang 27  
Aufbau Multi-Head-3-ConvLSTM



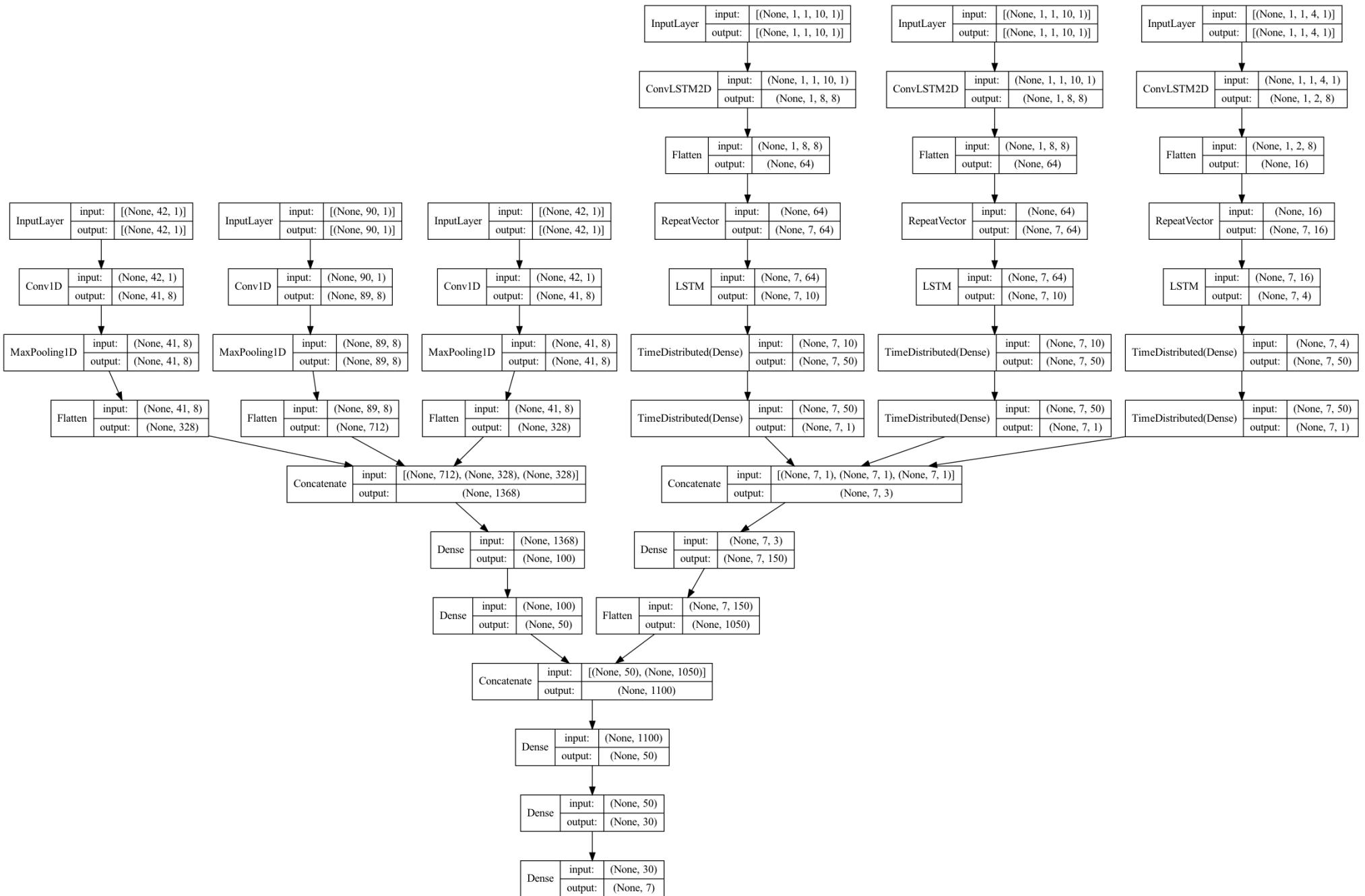
Anhang 28  
Aufbau Multi-Head-5-CNN



Anhang 29  
Aufbau Mutli-Head-5-ConvLSTM



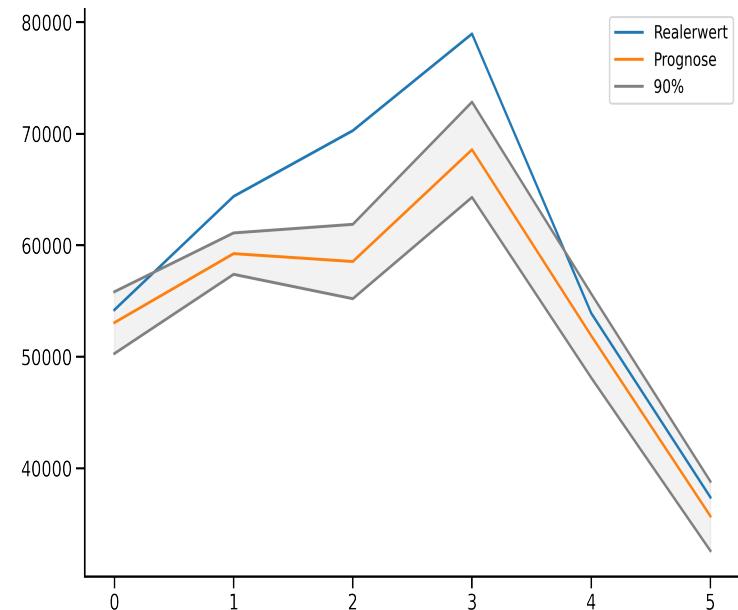
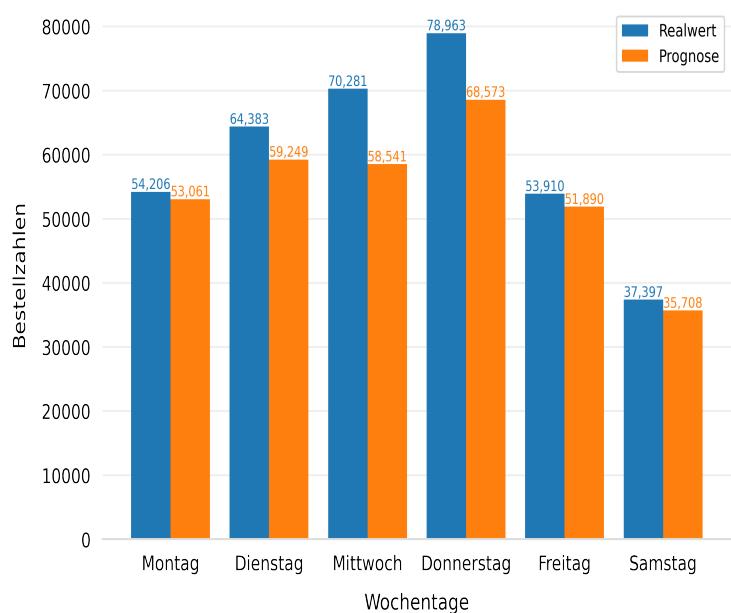
Anhang 30  
Aufbau Mutli-Head-5-CNN-ConvLSTM



Anhang 31  
Report Evaluierungsdatensatz Pro-Label

2021-08-30 - 2021-09-04

Datum	Wochentag	Realwert	Prognose	Lower	Upper
2021-08-30	Montag	54206	53061	50288	55832
2021-08-31	Dienstag	64383	59249	57391	61106
2021-09-01	Mittwoch	70281	58541	55205	61876
2021-09-02	Donnerstag	78963	68573	64298	72848
2021-09-03	Freitag	53910	51890	48123	55657
2021-09-04	Samstag	37397	35708	32597	38817

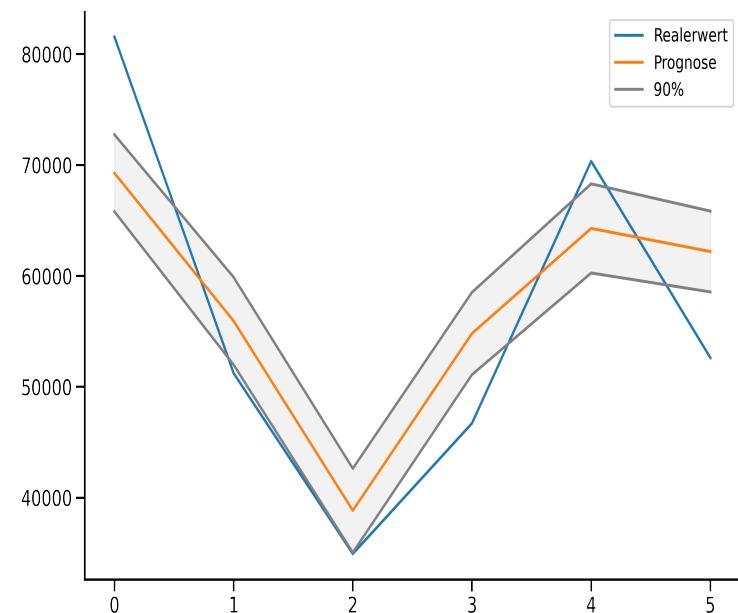
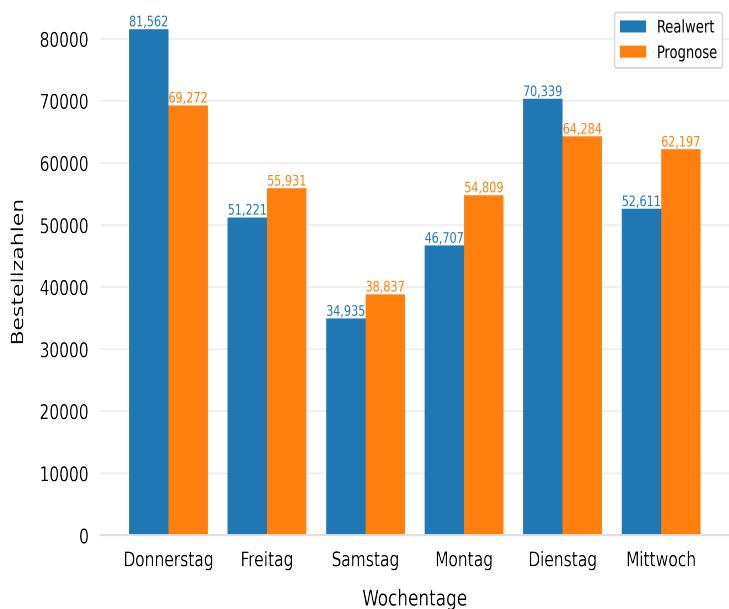


Kennzahl	Realwert	Prognose
MEAN	59857	54504
STD	14601	10945
Q0.5	59294	55801

Errors	Prognose
MSE	6836
MAE	5353
MAPE	0.08

2021-05-27 - 2021-06-02

Datum	Wochentag	Realwert	Prognose	Lower	Upper
2021-05-27	Donnerstag	81562	69272	65800	72742
2021-05-28	Freitag	51221	55931	51981	59879
2021-05-29	Samstag	34935	38837	35047	42626
2021-05-31	Montag	46707	54809	51095	58523
2021-06-01	Dienstag	70339	64284	60262	68304
2021-06-02	Mittwoch	52611	62197	58550	65842

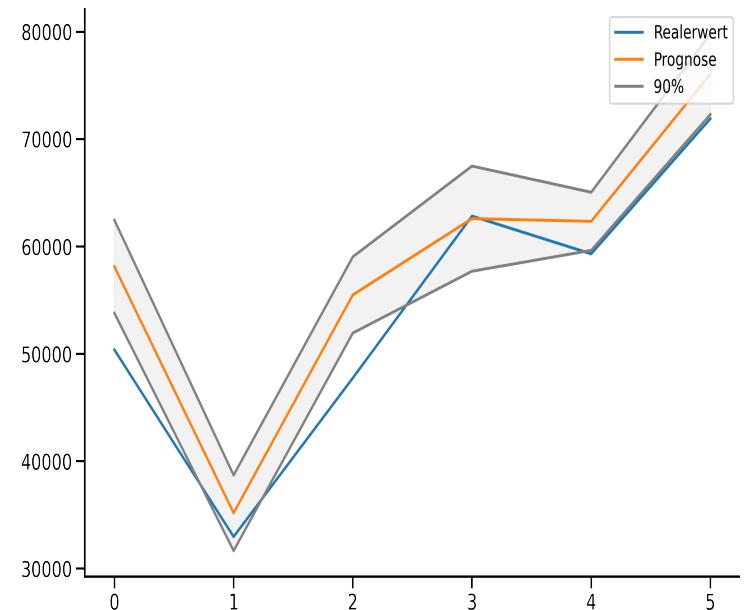
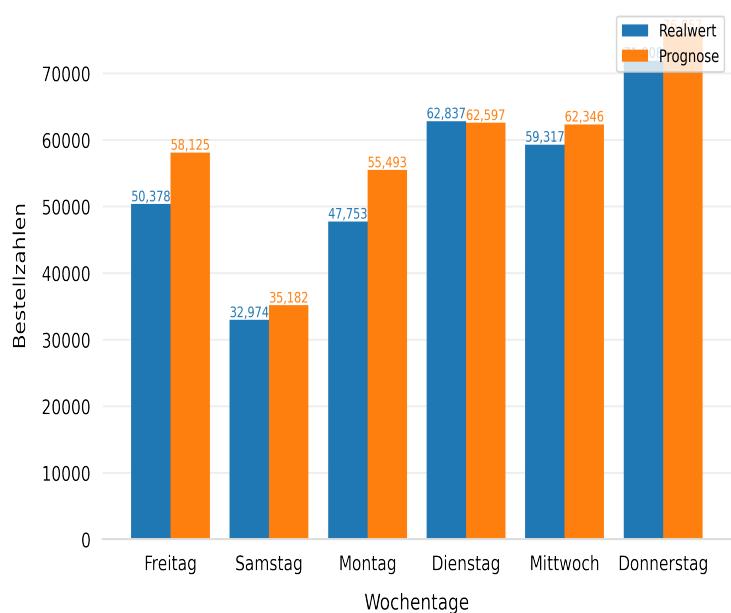


Kennzahl	Realwert	Prognose
MEAN	56229	57555
STD	16871	10627
Q0.5	51916	59064

Errors	Prognose
MSE	7986
MAE	7441
MAPE	0.13

2021-04-09 - 2021-04-15

Datum	Wochentag	Realwert	Prognose	Lower	Upper
2021-04-09	Freitag	50378	58125	53794	62456
2021-04-10	Samstag	32974	35182	31661	38701
2021-04-12	Montag	47753	55493	51942	59043
2021-04-13	Dienstag	62837	62597	57699	67494
2021-04-14	Mittwoch	59317	62346	59637	65054
2021-04-15	Donnerstag	71908	76057	72303	79810



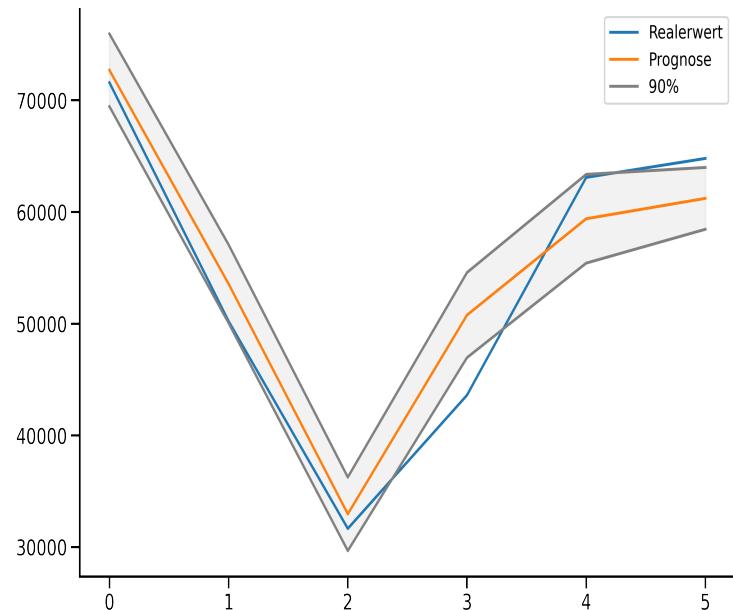
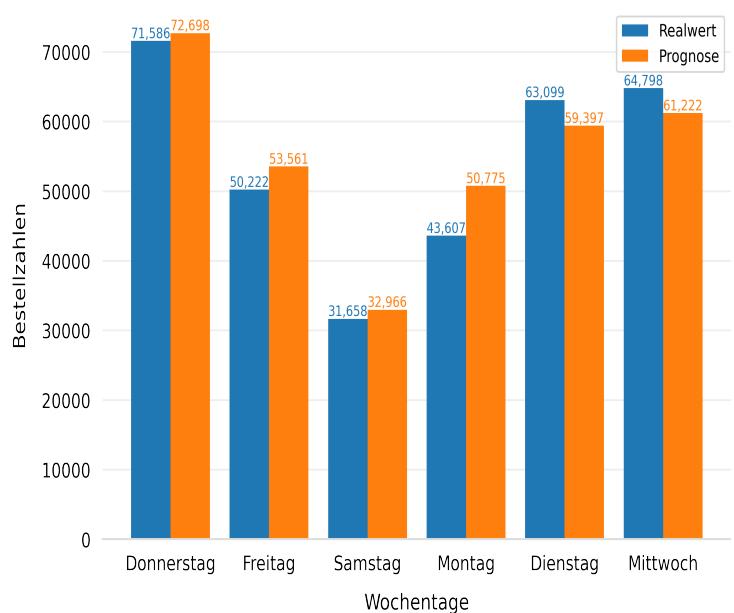
Kennzahl	Realwert	Prognose
MEAN	54194	58300
STD	13573	13360
Q0.5	54848	60236

Errors	Prognose
MSE	5021
MAE	4186
MAPE	0.08

## Bestellzahlen Vorhersage Reporting DPS

2021-08-05 - 2021-08-11

Datum	Wochentag	Realwert	Prognose	Lower	Upper
2021-08-05	Donnerstag	71586	72698	69443	75952
2021-08-06	Freitag	50222	53561	50050	57071
2021-08-07	Samstag	31658	32966	29672	36259
2021-08-09	Montag	43607	50775	46964	54584
2021-08-10	Dienstag	63099	59397	55425	63369
2021-08-11	Mittwoch	64798	61222	58454	63990



Kennzahl	Realwert	Prognose
MEAN	54162	55103
STD	15033	13241
Q0.5	56660	56479

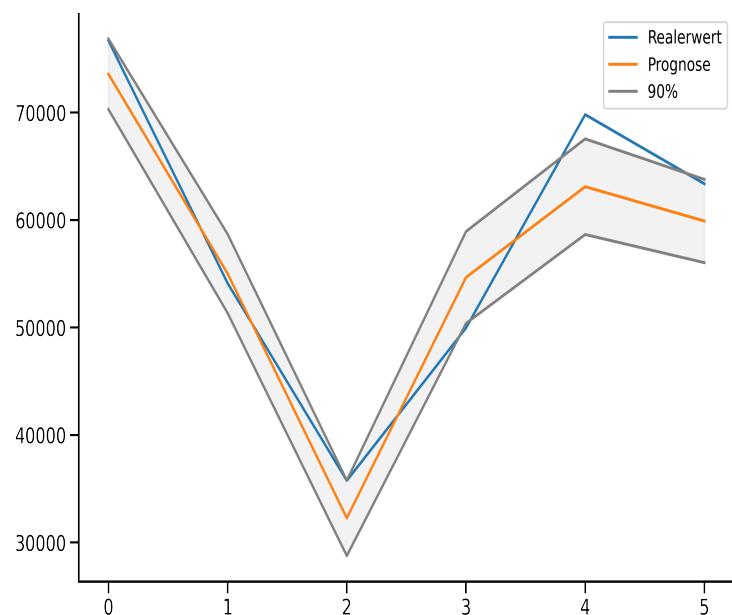
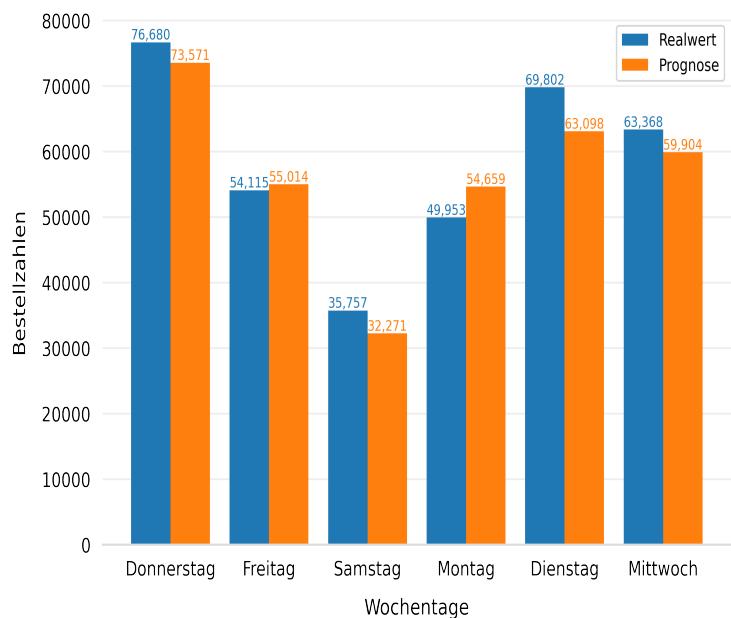
Errors	Prognose
MSE	3915
MAE	3368
MAPE	0.07



## Bestellzahlen Vorhersage Reporting DPS

2021-06-10 - 2021-06-16

Datum	Wochentag	Realwert	Prognose	Lower	Upper
2021-06-10	Donnerstag	76680	73571	70290	76851
2021-06-11	Freitag	54115	55014	51342	58686
2021-06-12	Samstag	35757	32271	28766	35776
2021-06-14	Montag	49953	54659	50387	58931
2021-06-15	Dienstag	69802	63098	58649	67545
2021-06-16	Mittwoch	63368	59904	56031	63776



Kennzahl	Realwert	Prognose
MEAN	58279	56420
STD	14773	13705
Q0.5	58742	57459

Errors	Prognose
MSE	4117
MAE	3728
MAPE	0.07

Anhang 32  
Report Evaluierungsdatensatz Gesamt



## Bestellzahlen Vorhersage Alle Daten Reporting 2 DPS

Step	RMSE	MAE	MAPE
1	7363	5065	0.09
2	7434	5137	0.09
3	7493	5125	0.09
4	7486	4951	0.09
5	7560	5158	0.09
6	7423	4959	0.09
7	7517	5003	0.09

Wochentag	RMSE	MAE	MAPE
Montag	5964	5120	0.11
Dienstag	5902	5009	0.08
Mittwoch	5288	4168	0.07
Donnerstag	7758	6329	0.09
Freitag	4285	3695	0.07
Samstag	2922	2510	0.08

Wochentag	Mean_r	Std_r	Q50_r
Montag	48743	4318	48580
Dienstag	63230	5272	63042
Mittwoch	62291	5039	62322
Donnerstag	72892	6240	71758
Freitag	52066	4011	51221
Samstag	32580	2912	32974

Mean_p	Std_p	Q50_p
53107	2635	54026
59569	2236	60068
61836	1563	61838
73066	2967	73170
54437	2179	54428
32298	2685	32235

Anhang 33  
Ergebnisse Testdatensatz

## Evaluation Testdatensatz CNN-LSTM

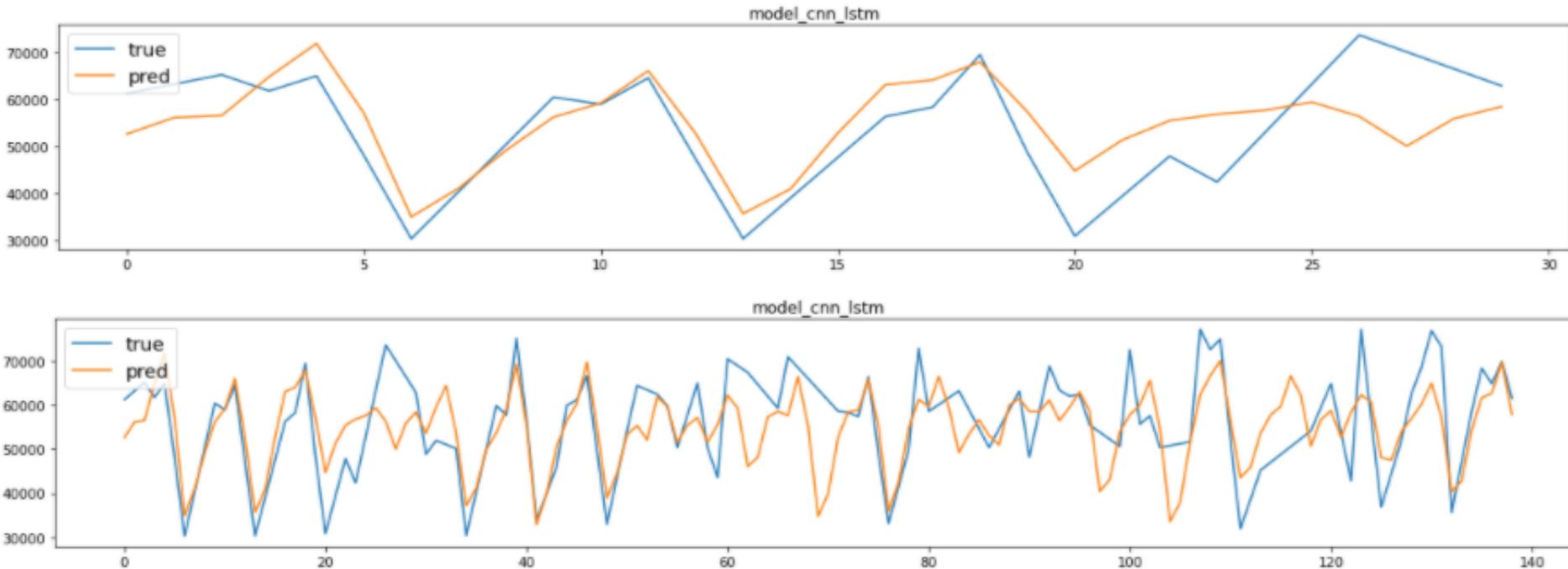
Fehlermetriken

Architekturen	RMSE	MAE	MAPE
CNN-LSTM	8401	6318	0,11

Hyperparameter Konfiguration

Architekturen	parameters
CNN-LSTM	{'epochs': 60, 'batch_size': 256, 'filters': 128, 'kernel_size': 2, 'pool_size': 1, 'loss': 'mae', 'dense1': 300, 'dense2': 150, 'lstm_units': 400, 'optimizer': 'adam'}

Prognosen Testdatensatz



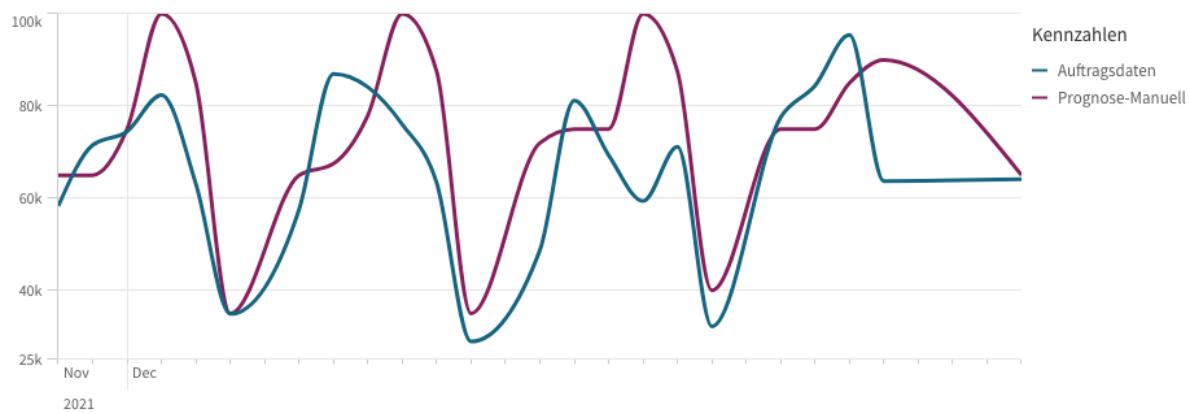
Anhang 34  
Prognosen Dezember-2021

## Vergleich Prognosen Dezember 2021

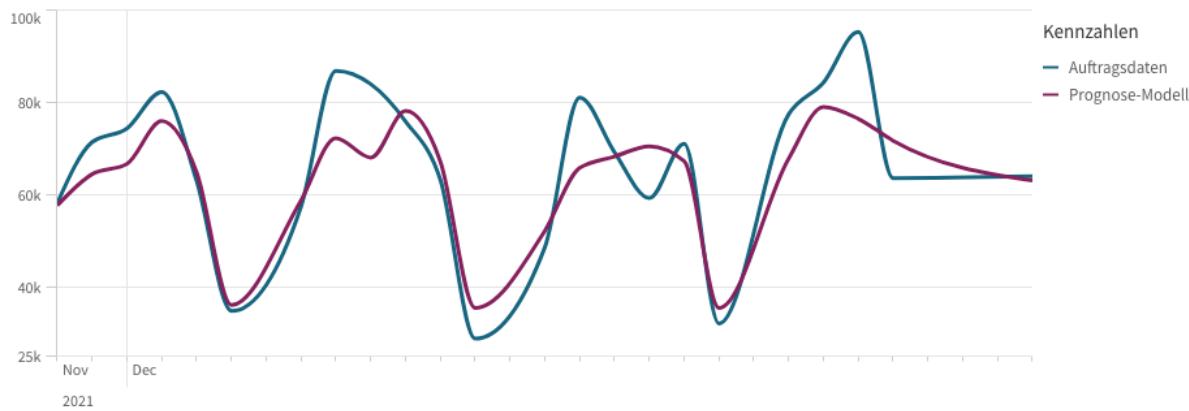
Datum	Prognose_Manuell	Prognose_Modell	Echter Wert
29.11.21	65000	57780	58307
30.11.21	65000	64582	71510
01.12.21	75000	66809	74488
02.12.21	100000	76086	82411
03.12.21	85000	65048	63107
04.12.21	35000	36221	34918
06.12.21	65000	58920	57592
07.12.21	67500	72386	86951
08.12.21	78000	68162	84102
09.12.21	100000	78258	76021
10.12.21	87500	67314	63343
11.12.21	35000	35571	28919
13.12.21	72000	52299	48722
14.12.21	75000	65921	81210
15.12.21	75000	68399	69328
16.12.21	100000	70637	59388
17.12.21	87500	67408	71186
18.12.21	40000	35560	32203
20.12.21	75000	67955	77557
21.12.21	75000	79148	84437
22.12.21	85000	76588	95392
23.12.21	90000	71821	63733
27.12.21	65000	63185	64125

Prognose	MAE	RMSE	MAPE
Manuell	12602	16185	0,2
Modell	6534	8406	0,1

Manuelle-Prognose Dezember\_2021



Modell-Prognose Dezember\_2021



Anhang 35  
Bereitstellungs-Report



## Bestellzahlen Vorhersage Reporting DPS

2022-02-01 - 2022-02-08

Datum	Wochentag	Prognose	Lower	Upper
2022-02-01	Dienstag	58646.0	53350.01	63942.43
2022-02-02	Mittwoch	61938.0	57981.01	65894.92
2022-02-03	Donnerstag	72639.0	68815.02	76463.9
2022-02-04	Freitag	63044.0	59396.84	66691.39
2022-02-05	Samstag	39518.0	33674.04	45361.54
2022-02-07	Montag	44353.0	40649.29	48056.04
2022-02-08	Dienstag	55438.0	50906.5	59969.79

