Lehrstuhl für Informationstechnologie

Masterarbeit

Evaluierung der Eignung von neuronalen Netzen zum Forecasting logistischer Zeitreihen - Ein Beispiel aus der österreichischen Lebensmittelindustrie

Jürgen Kotzbeck, BSc

Mai 2022

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 08.05.2022

Unterschrift Verfasser/in
Jürgen Kotzbeck

Abstract

The forecasting of future sales volumes is critical to be successful in many industries. Due to an ever increasing complexity in the time series under consideration, traditional forecasting methods are reaching limits in their performance. Therefore, new approaches are needed to continue this vital task.

The goal of this master thesis is to evaluate the use of neural networks to forecast logistic time series, specifically the determination of primary demands in material requirements planning. The operational processes and weekly customer demands of an Austrian food company will form the basis of this approach. The main objective is to decrease inventory while still maintaining the minimum delivery capability desired by the company.

To evaluate the feasibility of using neural networks to determine the primary demands, the effects of the forecast on inventory and delivery capability were compared to the results of the current dispatch planning. The effects of the forecasts were mesured for the year 2021. A simulation of material requirement planning was used to calculate the inventory.

The results of the evaluation show potential for the use of neural networks in material requirements planning, both to reduce total inventory and in relation to the different disposition procedures used by the company.

Kurzfassung

Die Erstellung von Prognosen zur Bestimmung zukünftiger Absatzmengen stellt in vielen Industriezweigen eine erfolgskritische Aktivität dar. Aufgrund der zunehmenden Komplexität in den betrachteten Zeitreihen stoßen klassische Prognosemethoden oftmals an ihre Leistungsgrenzen, weshalb alternative Ansätze benötigt werden.

Im Rahmen dieser Masterarbeit soll die Eignung neuronaler Netze für die Prognose logistischer Zeitreihen evaluiert werden. Den Prognosegegenstand stellt hierbei der Primärbedarf in der Materialbedarfsplanung dar. Die Prognose des Primärbedarfs erfolgt je Produkt und anhand der wöchentlichen Kundenbedarfe. Ziel der Prognose ist die Minimierung des Lagerbestandes unter Berücksichtigung einer durch das betrachtete Unternehmen vorgegebenen Mindestanforderung an die Lieferfähigkeit. Die Problemstellung wird am Beispiel eines Unternehmens der österreichischen Lebensmittelindustrie bearbeitet.

Für die Evaluierung werden die Auswirkungen der durch den Einsatz von neuronalen Netzen erstellten Prognosen auf den Gesamtlagerbestand und die Lieferfähigkeit je Produkt analysiert. Diese Ergebnisse werden mit dem Gesamtlagerbestand und den Lieferfähigkeiten basierend auf der aktuellen Dispositionsplanung verglichen. Die Berechnungen werden für das Kalenderjahr 2021 durchgeführt. Für die Ermittlung des Lagerbestandes wird eine Simulation der Materialbedarfsplanung verwendet.

Die Ergebnisse der Evaluierung zeigen Potenziale beim Einsatz neuronaler Netze in der Materialbedarfsplanung sowohl zur Reduzierung des Gesamtbestandes als auch in Bezug auf die unterschiedlichen Dispositionsverfahren im betrachteten Unternehmen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

FOP Fixed Order Period

FOQ Fixe Order Quantity

MRP Material Requirement Planning

MRP II Material Ressourece Planning

SCM Supply Chain Management

1 Einleitung

Die Erstellung und die Interpretation von Prognosen stellt in vielen Industriezweigen eine erfolgskritische Aktivität dar. Vor allem im Supply Chain Management (SCM) werden Prognosen oft als treibender Faktor für Planungs- und Entscheidungsprozesse angesehen [Avci 2019].
Die Beantwortung von Fragen wie *"Welche Produkte werden in welchen Mengen in den nächsten Wochen nachgefragt?"* bestimmt direkt die benötigte Art und Menge der Produkte zum Zeitpunkt und Ort des Verkaufs und beeinflusst damit indirekt die Ressourcen- und Kapazitätsplanung innerhalb der gesamten Lieferkette vom Rohstoffeinkauf bis zur Produktverteilung an den Endkunden [Patàk und Vlckova 2014].

Unternehmen, welche die Fähigkeit besitzen, die für sie relevanten Zeitreihen zu interpretieren und daraus Vorhersagen über die weiteren Verläufe abzuleiten, verschaffen sich Vorteile in der Entscheidungsfindung dieser sowohl langfristigen, strategischen als auch kurzfristigen, operativen Fragestellungen.

Zuverlässige Prognosen bieten die Möglichkeit, frühzeitig innerbetriebliche Prozesse gezielt auf die aktuellen und zukünftigen Anforderungen anzupassen und sich somit Wettbewerbsvorteile gegenüber der Konkurrenz zu sichern.

Um diese Potentiale zu nutzen, wurden bereits unterschiedlichste qualitative und quantitative Ansätze zur Interpretation und Prognose von Zeitreihen entwickelt [Veiga et al. 2010], wobei das Autoregressive- (AR) und Autoregressive-Moveing-Average Modelle (ARMA) als eine der wichtigsten Vertreter in der Prognose von univariaten Zeitreihen angesehen werden können [Koller 2014].

Die zunehmende Volatilität der Kundenbedarfe und immer kürzer werdende Produktlebenszyklen in vielen Industriezweigen haben allerdings zur Folge, dass die betrachteten Zeitreihen kürzer werden und der Anteil des nicht-prognostizierbaren Fehlers immer höher wird. Zusätzlich haben nicht-stationäre Komponenten wie stochastischer bzw. deterministischer Trend und Saisonalität und deren Handhabung einen wesentlichen Einfluss auf die Prognosequalität. Diese Faktoren führen dazu, dass univariate, lineare Zeitreihenmodelle, wie die erwähnten AR und ARMA Modelle, an ihre Leistungsgrenzen stoßen [Koller 2014].

Ein weiteres Festhalten an diesen Modellen und die damit einhergehende, abnehmende Prognosequalität birgt die Gefahr, Fehlentscheidungen in der innerbetrieblichen Prozessgestaltung zu treffen. Als Konsequenz können sich steigende Lagerbestände oder fehlerhafte Ressourcen-Planungen entlang der gesamten Wertschöpfungskette ergeben.

Die Lebensmittelindustrie kann als typisches Beispiel jener Industriezweige angesehen werden, die sowohl mit der zunehmenden Volatilität in den Absatzmengen als auch mit immer kürzer werdenden Produktlebenszyklen konfrontiert werden. Steigende Wiederbeschaffungszeiten für die einzelnen Komponenten und ein immer größer werdendes Produktsortiment stellen in dieser Industrie zusätzliche Herausforderungen dar, durch die die Erstellung von akkuraten Prognosen für diese Industrie mithilfe von statistischen, linearen Modellen limitiert ist [Avci 2019].

Sowohl theoretische als auch praktische Arbeiten zeigen, dass die Erstellung von Prognosen mithilfe von neuronalen Netzen ein vielversprechendes Themengebiet darstellt, um die Prognosequalität in volatilen Zeitreihen gegenüber den linearen, statistischen Modellen zu erhöhen [Koller 2014].
In den meisten dieser Arbeiten liegt der Fokus allerdings nicht auf einem bestimmten Industriezweig und dessen Problemstellungen, sondern auf der Anwendung neuronaler Netze für spezifische Charakteristika von Zeitreihen.

Daher werden im Rahmen dieser Arbeit die Erkenntnisse aus der einschlägigen Literatur für die spezifischen Zeitreihen anhand eines Praxisbeispiels der Lebensmittelindustrie analysiert und bezüglich der Einbindung der Vorhersagen in die logistischen Planungsprozesse evaluiert.

1.1 Zielsetzung

Basierend auf der in Abschnitt 1 erläuterten Problemstellung, besteht das Ziel der Arbeit in der Evaluierung der Eignung und Einsetzbarkeit von neuronalen Netzen für die Prognose logistischer Zeitreihen anhand eines Praxisbeispiels der österreichischen Lebensmittelindustrie.

Der Fokus liegt hierbei auf der Anwendung der Prognoseergebnisse für das Teilgebiet der Produktionsplanung und der damit einhergehenden Bestimmung der Produktionslosgrößen und Produktionszyklen.

Als Prognosegegenstand werden die wöchentlichen Absatzzahlen bis zum Stichtag am 31.01.2021 der einzelnen Produkte verwendet, wobei der Fokus auf den zuvor mittels einer ABC/XYZ-Analyse klassifizierten A/Z-Produkten liegt.

In Abhängigkeit der Einstellungen an den Materialstämmen im Enterprise-Resource-Planning (ERP) System und der damit einhergehenden unterschiedlichen Zeiträume der Produktionszyklen, erfolgt die Bestimmung des Prognosezeitraums dynamisch je Produkt. Ebenfalls wird für jedes Produkt ein separates Modell erstellt.

Um eine korrekte Auswertung der Modelle zu garantieren, werden die vorhandenen Daten in Trainings-, Validierungs- und Testset geteilt, wobei das Testset erstmalig bei der abschließenden Modellevaluierung herangezogen wird.
Die Teilung erfolgt auf zwei unterschiedliche Arten. Einerseits wird eine zeitliche Trennung zwischen den Sets durchgeführt. Somit werden, wie in Abbildung 1.1 ersichtlich, alle verfügbaren Daten ab dem 01.01.2021 ausschließlich für das Testset verwendet. Andererseits erfolgt eine Teilung innerhalb aller verfügbaren Daten bis zum Stichtag am 30.12.2020. Diese Daten werden im Verhältnis 50/50 auf das Trainings- und Validierungsset aufgeteilt.

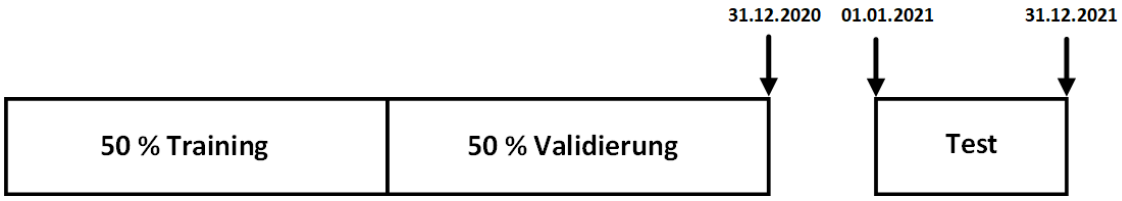


Abbildung 1.1: Teilung Trainings-, Validierungs- und Testset

Zusätzliche Einschränkungen in der Definition der Zielfunktion für die Problemstellung, bilden die spezifischen Anforderungen der Lebensmittelindustrie wie beispielsweise die Berücksichtigung des Mindesthaltbarkeitsdatums bzw. der maximalen Lagerzeit und die

damit einhergehende Vermeidung zu hoher Lagerbestände.

Basierend auf der Problemstellung und der Zielsetzung soll im Rahmen dieser Arbeit folgende Hauptforschungsfrage betrachtet werden:

Hauptforschungsfrage

Welche Auswirkungen auf den Lagerbestand von Fertigteilen zeigen sich durch die Bestimmung der Planprimärbedarfe mithilfe von neuronalen Netzen gegenüber der aktuellen Materialbedarfsplanung?

Diese Hauptforschungsfrage kann anhand der im Ist-Prozess durchgeführten Verfahren der Materialbedarfsplanung weiter spezifiziert werden, wodruch sich die nachfolgende Forschungsunterfrage ergibt:

Forschungsunterfrage 1

Welche Einflüsse auf die Potentiale der Anwendung von neuronalen Netzen gegenüber der aktuellen Planung ergeben sich durch die unterschiedlichen Verfahren der Materialbedarfsplanung?

Neben den Arten der Materialbedarfsplanung, lassen sich auch Unterschiede aufgrund der Nachfrageschwankungen der spezifischen Produkte in Hinsicht auf die Potentiale in der Anwendung von neuronalen Netzen vermuten. Dieses Thema soll anhand folgender Forschungsunterfrage behandelt werden:

Forschungsunterfrage 2

Welche Auswirkungen auf die Potentiale der Anwendung von neuronalen Netzen gegenüber der aktuellen Planung ergeben sich durch die Volatlität der betrachteten Zeitreihe?

2 Ist-Prozess der Produktionsplanung

Ziel des nachfolgenden Kapitels ist es, den Ist-Prozess der Produktionsplanung des betrachteten Unternehmens detailliert zu beschreiben und zu analysieren.

Da die erhobenen Ist-Prozesse Parallelen zum allgemeinen Ablauf des Manufacturing-Resource-Planning (MRP II) aufweisen, wird dieser Ablauf als Referenzmodell im Rahmen des gesamten Kapitels und der Arbeit herangezogen.
Diese Verknüpfung der theoretischen und praktischen Prozessabläufe soll der Schaffung eines einheitlichen Verständnisses der verwendeten logistischen Terminologien und zur Einordnung der Teilprozesse in das Supply-Chain-Management dienen.

2.1 Gliederung des Ist-Prozesses anhand des Manufacturing-Resource-Planning

Analog zum allgemeinen Ablauf des Manufacturing-Resource-Planning kann der erhobene Ist-Prozess der Produktionsplanung in die drei Teilbereiche Langfristplanung, Mittelfristplanung und Kurzfristplanung gegliedert werden.
Der Ablauf der MRP II Ist-Planung und die Zuordnung der einzelnen Planungsschritte zu den jeweiligen zeitlichen Teilbereichen sind in Abbildung 2.1 ersichtlich.
Die grau hinterlegen Prozessschritte beschreiben alle nötigen Planungsaktivitäten im Gesamtablauf. Die in die Planung eingehenden bzw. aus den einzelnen Prozessschritten resultierenden Informationen und Datenstrukturen sind weiß dargestellt.

Im Falle des Ist-Prozesses gliedert sich die Langfristplanung in die beiden Hauptteile der Absatz- und Programmplanung. Deren Ergebnisse beeinflussen in weiterer Folge direkt die Grobplanung der Ressourcen und Kapazitäten für das resultierende Produktionsprogramm.
Als Zeitraum für die Absatzplanung wird in der Regel eine Periode von einem Jahr betrachtet, wobei die aus der Planung resultierende Absatzvorschau aufgrund von Erfahrungswerten je Produkt festgelegt und zeitlich in monatliche Subperioden unterteilt wird. Die zeitliche Kombination aus Jahr und Monat wird als Geschäftsperiode bezeichnet.

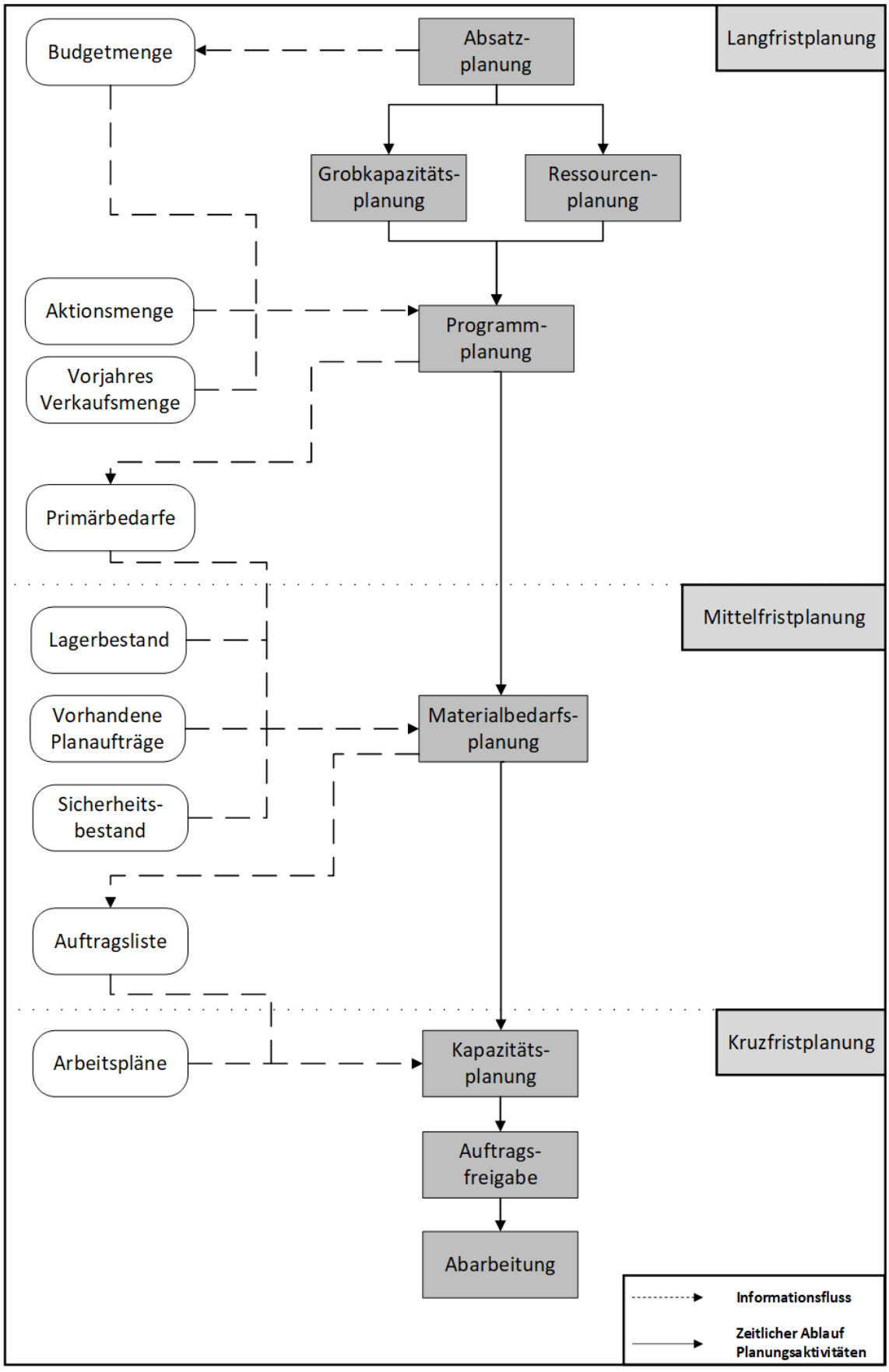


Abbildung 2.1: Überblick MRP II Planung nach [Jodlbauer 2008]

Die in der Absatzplanung monatlich festgelegten Bedarfsmengen werden im nachfolgenden Schritt der Programmplanung als Budgetmenge berücksichtigt. Zusammen mit den Verkaufszahlen des Vorjahres und bereits bekannter Aktionsmengen in der betrachteten Geschäftsperiode werden mithilfe der Budgetmenge die Primärbedarfe für jedes dispositiv relevante Produkt erstellt. Dem Disponenten wird für diese Schritte ein Dispositions-Cockpit zur Verfügung gestellt, dessen Aufbau und Funktionsweise in Abschnitt 2.1.1 dargestellt sind.

Basierend auf den in der Programmplanung bestimmten Primärbedarfen wird in der Mittelfristplanung des MRP II Verfahrens die Materialbedarfsplanung durchgeführt.
Zentrale Aufgabe der Materialbedarfsplanung ist hierbei die Sicherstellung der Materialverfügbarkeit sowohl für die eigen- als auch für die fremdgefertigten Produkte im Produktionsprogramm.
Unter Berücksichtigung des aktuellen Lagerbestandes, des definierten Sicherheitsbestandes und bereits vorhandener Planaufträge aus Vorperioden wird aus diesem Grund im Rahmen der Materialbedarfsrechnung der Nettobedarf je Produkt bestimmt und es werden Beschaffungsvorschläge in Form von Planaufträgen erzeugt.
Der Ablauf der Berechnung sowie die Erhebung der benötigten Komponenten und Zeiträume werden ab Abschnitt 2.1.2 betrachtet.

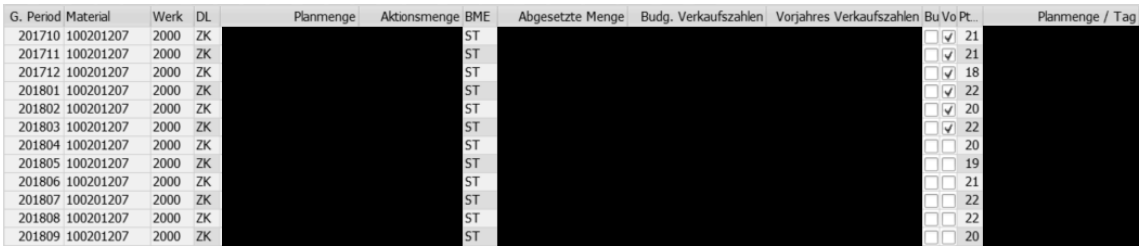
Den Abschluss des MRP II Ist-Prozesses bildet die Kurzfrist- oder auch Feinplanung. Basierend auf den in der Materialbedarfsrechnung bestimmten Planaufträgen werden im Rahmen der Kurzfristplanung Fertigungsaufträge erzeugt. Für diese erfolgen auf Grundlage der den Produkten zugeordneten Arbeitspläne und verfügbaren Kapazitäten die Festlegung und Freigabe der genauen Produktionsreihenfolge sowie die abschließende Erzeugung der Produkte.

Da der Fokus der Arbeit auf der Evaluierung der Potentiale von Forecasts für die Schritte der Langfrist- bzw. Mittelfristplanung liegt, wird auf die genauen Aktivitäten der Kurzfristplanung innerhalb der nachfolgenden Kapitel nicht näher eingegangen. Informationen zu diesen Planungsaktivitäten liefern beispielsweise die Inhalte von [Jodlbauer 2008] oder [Heiserich et al. 2011].

2.1.1 Programmplanung im Ist-Prozess

Die Programmplanung im Ist-Prozess erfolgt immer rollierend in Drei-Monats-Schritten, somit wird beispielsweise im Januar 2021 die Planung für die Monate Februar, März und April erstellt. Im Februar können die bereits erstellten Planungen für März und April angepasst werden. Zusätzlich erfolgt die Planung für Mai.

Zur Unterstützung der Durchführung der Programmplanung im Ist-Prozess, wird dem Disponenten ein eigengefertigtes Cockpit zur Planung der Primärbedarfe zur Verfügung gestellt. Dieses ist in Abbildung 2.2 ersichtlich.



G. Period	Material	Werk	DL	Planmenge	Aktionsmenge	BME	Abgesetzte Menge	Budg. Verkaufszahlen	Vorjahres Verkaufszahlen	Bu	Vo	Pt..	Planmenge / Tag
201710	100201207	2000	ZK			ST				☐	☑	21	
201711	100201207	2000	ZK			ST				☐	☑	21	
201712	100201207	2000	ZK			ST				☐	☑	18	
201801	100201207	2000	ZK			ST				☐	☑	22	
201802	100201207	2000	ZK			ST				☐	☑	20	
201803	100201207	2000	ZK			ST				☐	☑	22	
201804	100201207	2000	ZK			ST				☐	☐	20	
201805	100201207	2000	ZK			ST				☐	☐	19	
201806	100201207	2000	ZK			ST				☐	☐	21	
201807	100201207	2000	ZK			ST				☐	☐	22	
201808	100201207	2000	ZK			ST				☐	☐	22	
201809	100201207	2000	ZK			ST				☐	☐	20	

Abbildung 2.2: Cockpit Dispositionsplanung

Über das Cockpit werden dem Disponenten sowohl die relevanten Materialinformationen, wie beispielsweise das dem Produkt zugeordnete Dispositionsverfahren, als auch alle zur Bedarfsplanung benötigten Mengenfelder in übersichtlicher Form angezeigt.
Neben der über die Absatzplanung definierten Budgetmenge stellen die aktuelle Aktionsmenge und die Absatzmenge des Vorjahres die Haupteinflussfaktoren auf die Bedarfsplanung je Geschäftsperiode dar.

Zur Bestimmung der Planmenge stehen dem Disponenten drei Möglichkeiten zur Verfügung:

- Bestimmung der Planmenge rein auf Grundlage der Budgetmenge
- Bestimmung der Planmenge rein auf Grundlage der Vorjahresverkaufsmenge
- eigenständige Bestimmung der Budgetmenge auf Basis der Informationen aus Budget-, Aktions- und Vorjahresverkaufsmenge.

Die Auswahl der reinen Budget- oder Vorjahresverkaufsmenge und deren Übernahme in die gesamte Planmenge werden hierbei über eine Checkbox im Cockpit durchgeführt. Für die manuelle Bestimmung der Planmenge wird der erhobene Wert händisch in das dazu vorgesehene Feld eingetragen.

Im Anschluss an die Fixierung der gesamten Planmenge für die jeweilige Geschäftsperiode, erfolgt auf deren Basis eine automatische Ermittlung der Tagesbedarfe. Grundlage

für die Berechnung der Tagesbedarfe sind die Werktage innerhalb der betrachteten Geschäftsperiode. Diese werden über einen im SAP-ERP-System definierten Fabrikkalender je Werk ermittelt.

Die Planmenge pro Tag kann mithilfe der Formel (2.1) berechnet werden, wobei immer auf ganze Zahlen aufgerundet wird:

$$x_{Plan_T} = \left\lceil \frac{x_{Plan}}{t_{Arbeit}} \right\rceil \qquad t_{Arbeit} \geq 1 \tag{2.1}$$

mit
x_{Plan_T} .... Planmenge pro Tag
x_{Plan} .... gesamte Planmenge
t_{Arbeit} .... Arbeitstage in der Geschäftsperiode

Dieser ermittelte Wert stellt den Bruttobedarf für die nachfolgende Materialbedarfsplanung dar. Die dazu benötigten Primärbedarfe werden automatisch durch das System erzeugt.

2.1.2 Theoretische Materialbedarfsplanung

Die Materialbedarfsplanung (Material-Requirements-Planning, MRP) stellt die zentrale Funktion der mittelfristigen Produktionsplanung dar. Ziel der Planung ist es, auf Basis der in der Programmplanung ermittelten Bedarfe, die Materialverfügbarkeit in jenem Maß sicherzustellen, dass sowohl alle geplanten Produktionsvorgänge als auch Kundenaufträge termin- und mengengerecht abgewickelt werden können [Kappauf et al. 2017].

Das Ergebnis der Materialbedarfsplanung sind Beschaffungsvorschläge (Planaufträge) für den festgelegten Bedarfszeitpunkt. Die Bestimmung des Zeitpunktes und des Zeithorizontes der Mengenplanung ist hierbei abhängig von den spezifischen Rahmenbedingungen des betrachteten Produktes. Die Mindestanforderungen für den zeitlichen Vorlauf zur termingerechten Bereitstellung ergeben sich für fremdbeschaffte Produkte aus der Wiederbeschaffungszeit der jeweiligen Komponenten bzw. für eigengefertigte Produkte anhand der benötigten Vorplanungszeit der Produktionsplanung und der Fertigungsdauer[Jodlbauer 2008]. Die detaillierte Berechnung der Horizonte (Planübergangszeiten) ist in Abschnitt 2.1.3.2 ersichtlich.

Zur Durchführung der Mengenplanung werden in der Praxis je nach Art der vorhandenen Informationen im Wesentlichen zwei unterschiedliche Dispositionsverfahren herangezogen.

Die plangesteuerte oder auch deterministische Disposition, leitet ihre Bedarfe aus dem zuvor definierten Produktionsprogramm bzw. aus der Stücklistenstruktur des jeweils betrachteten Produktes ab. Haupteinflussfaktoren für die Bedarfe der plangesteuerten Disposition sind bereits bekannte Kundenaufträge, die im Zuge der Programmplanung ermittelten Primärbedarfe, Materialreservierungen für Fertigungsaufträge und die Stücklisten zur Ermittlung der benötigten Komponenten [Zsifkovits 2012].
Der Gesamtprozess zur Berechnung des Nettobedarfs und die Ermittlung der dazu benötigten Einflussfaktoren werden ab Abschnitt 2.1.2.3 im Detail behandelt.

Im Gegensatz zur plangesteuerten Disposition besteht bei der verbrauchsgesteuerten Disposition (auch stochastischen Disposition genannt) kein Bezug zum definierten Produktionsplan. Somit werden in der Nettobedarfsrechnung der verbrauchsgesteuerten Disposition nicht die Primär- oder Sekundärbedarfe des betrachteten Produktes herangezogen. Je nach Verfahren erfolgt die Bedarfsermittlung anhand einer Unterschreitung eines festgelegten Meldepunktes im Lagerbestand oder auf Basis von errechneten Prognosebedarfen anhand vergangener Verbräuche [Zsifkovits 2012].
Das Bestellpunkt- und Bestellrhythmusverfahren sowie die stochastische Disposition zählen zu den häufigsten praktischen Auslegungen der verbrauchsgesteuerten Disposition, auf deren Ausführungen im Rahmen dieser Arbeit nicht weiter eingegangen wird. Einzelheiten zu diesen Verfahren sind in [Schönsleben 2007] oder [Gulyássy et al. 2014] ersichtlich.

2.1.2.1 Materialbedarfsplanung im SAP-ERP-System

Das Dispositionsverfahren wird im SAP-ERP-System anhand eines Dispositionsmerkmals bestimmt und über die Materialstammdaten für jedes Produkt einzeln festgelegt.

Basierend auf den Verfahren der plangesteuerten Disposition, stellt SAP-ERP in der Standardausführung hierzu folgende Verfahren zur Verfügung [Kappauf et al. 2017]:

- plangesteuerte Disposition mit Stücklistenauflösung
- plangesteuerte Disposition ohne Stücklistenauflösung
- Leitteileplanung

Die plangesteuerte Disposition mit Stücklistenauflösung referenziert hierbei auf den in Abschnitt 2.1.2 behandelten Standardprozess der Materialbedarfsrechnung, unter Berücksichtigung aller Komponenten, die zur Fertigung des spezifischen Produktes benötigt werden.
In der plangesteuerten Disposition ohne Stücklistenauflösung erfolgt eine reine Analyse der Nettobedarfe der Fertigteile, Komponenten werden nicht berücksichtigt.
Durch das Verfahren der Leitteileplanung wird in SAP eine bevorzugte Planung für jene

Produkte bereitgestellt, welche die Wertschöpfungskette im Unternehmen in hohem Maße beeinflussen. Ziel dieser Planung ist es, Engpässe für kostenintensive Produkte zu vermeiden. Aus diesem Grund stehen dem Disponenten für die als Leitteile gekennzeichneten Produkte erweiterte Planungsmöglichkeiten, bezogen auf das Produktionsprogramm, zur Verfügung.

Zur Feinsteuerung der einzelnen Dispositionsverfahren ermöglicht SAP-ERP weitere Konfigurationsmöglichkeiten innerhalb der unterschiedlichen Dispositionsmerkmale. Die Berücksichtigung des Fixierungshorizontes mithilfe von Fixierungsarten stellt hierbei die relevanteste Konfigurationsmöglichkeit im Rahmen dieser Arbeit dar.
Unter einem Fixierungshorizont wird jener Zeitraum verstanden, innerhalb dessen die Materialbedarfsplanung keine Beschaffungsvorschläge anlegt oder löscht. Dieser wird je Produkt am Materialstamm gepflegt. Ziel der Definition eines Fixierungshorizontes ist es, kurzfristige, maschinelle Mengenänderungen vor Produktionsbeginn zu verhindern und somit ein gewisses Ausmaß an Stabilität in der Produktionsplanung zu erreichen.

Die Art der Berücksichtigung des Fixierungshorizontes wird anhand der Fixierungsart festgelegt. Insgesamt stehen im SAP-Standard vier Fixierungsarten zur Verfügung, die sich anhand der automatischen Fixierung der Elemente bzw. der Erstellung neuer Elemente im Fixierungszeitraum unterscheiden. Abbildung 2.3 zeigt hierzu die jeweiligen Eigenschaften der Fixierungsarten.

	Neue Beschaffungsvorschläge am Ende des Fixierungshorizonts	**Keine Beschaffungsvorschläge für Unterdeckung im Fixierungshorizont**
Automatische Fixierung der Beschaffungsvorschläge im Fixierungshorizont	Fixierungsart 1	Fixierungsart 2
Keine Fixierung der Beschaffungsvorschläge im Fixierungshorizont	Fixierungsart 3	Fixierungsart 4

Abbildung 2.3: Fixierungsarten nach [Dickersbach und Keller 2014]

Das Verhalten der verschiedenen Fixierungsarten ist in Abbildung 2.4 anhand einer Unterdeckung beispielhaft dargestellt. Im Beispiel steht ein Planauftrag von fünf Stück einem Bedarf von zehn Stück innerhalb des Fixierungshorizonts gegenüber.

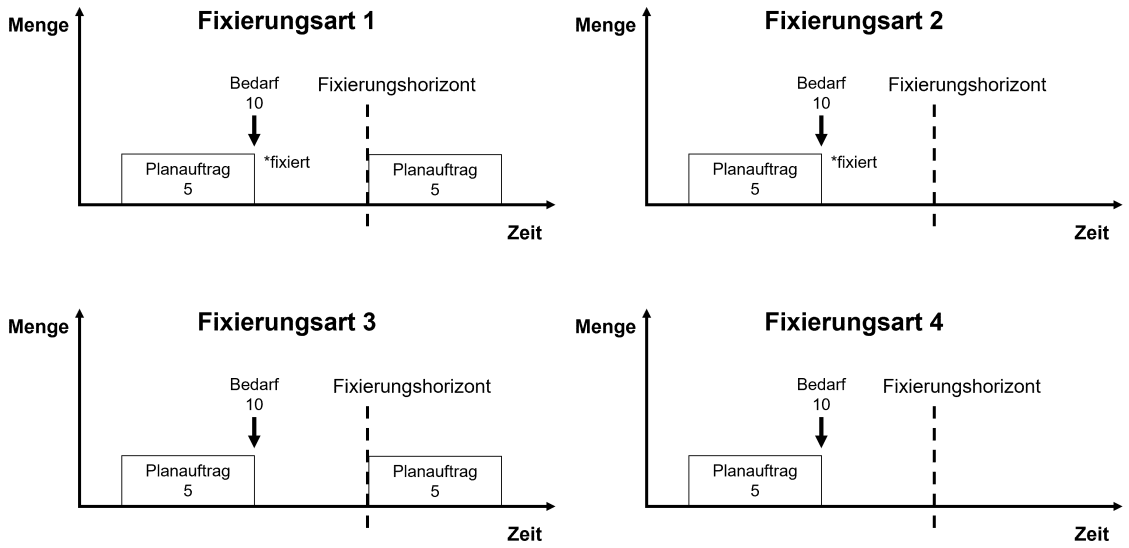


Abbildung 2.4: Einfluss Fixierungsarten nach [Dickersbach und Keller 2014]

Innerhalb des Fixierungshorizontes werden für keine Fixierungsart Beschaffungsvorschläge generiert. Im Falle der Fixierungsarten 1 und 3 werden diese automatisch am Ende des Horizontes erstellt. Für die Fixierungsarten 2 und 4 erfolgt keine automatische Erzeugung eines Beschaffungsvorschlages.

Unterschiede zwischen Fixierungsart 1 und 3 würden sich im Falle einer Überdeckung ergeben. Würde der Bedarf im Fixierungshorizont zwei Stück anstatt von zehn Stück betragen, würde folgendes Verhalten eintreten:
Für Produkte mit Fixierungsart 3 würde der Planauftrag innerhalb des Fixierungshorizontes maschinell auf zwei Stück reduziert werden. In Fixierungsart 1 wird keine maschinelle Anpassung durchgeführt. Eingriffe sind nur manuell erlaubt.

2.1.2.2 Materialbedarfsplanung im Ist-Prozess

Die Auswertung der für die Disposition relevanten Produkte zeigt, dass im Falle des Ist-Prozesses rein die plangesteuerte Disposition mit Stücklistenauflösung herangezogen wird. Hierzu werden die beiden *Dispositionsmerkmale PD - plangesteuerte Disposition* und *P3 - plangesteuert mit Fixierungsart 3* verwendet.

Der Anteil der Merkmale in den für die Disposition relevanten Produkte ist in Abbildung 2.5 ersichtlich.

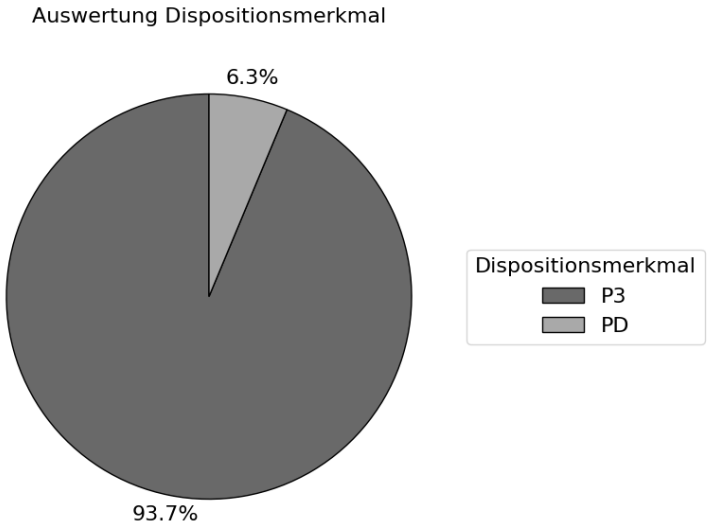


Abbildung 2.5: Auswertung Dispomerkmal

Unterschiede in den beiden Merkmalen zeigen sich anhand der in Abschnitt 2.1.2.1 beschriebenen Fixierungsart.
Bezogen auf das Dispositionsmerkmal P3 mit Fixierungsart 3 bedeutet dies folgendes Verhalten: Entsteht aufgrund der Beschaffungsvorschläge im Fixierungshorizont gegenüber der bekannten Bedarfe eine Überdeckung, so werden die Beschaffungsvorschläge anteilsmäßig gelöscht. Im Falle einer Unterdeckung werden Beschaffungsvorschläge über die Differenzmenge automatisch am Ende des Fixierungshorizonts erstellt.

Für das Dispositionsmerkmal PD erfolgt keine automatische Fixierung der Beschaffungsvorschläge, sodass der Fixierungshorizont für dieses Kennzeichen und die betroffenen Produkte nicht berücksichtigt wird.

2.1.2.3 Theoretischer Ablauf und Einflussgrößen der Materialbedarfsplanung

Da, wie in Abschnitt 2.1.2.2 beschrieben, im Ist-Prozess rein die plangesteuerte Disposition angewandt wird, liegt der Fokus der weiteren Abschnitte rein auf den Planungsschritten dieses Dispositionsverfahrens. Für die Verfahren der verbrauchsgesteuerten Disposition wird auf die Beiträge von [Gudehus 2011] und [Arnold et al. 2008] verwiesen.

Im Falle der Materialbedarfsplanung der plangesteuerten Disposition, gliedert sich der Ablauf im Wesentlichen in vier Schritte, die in Abbildung 2.6 dargestellt sind.

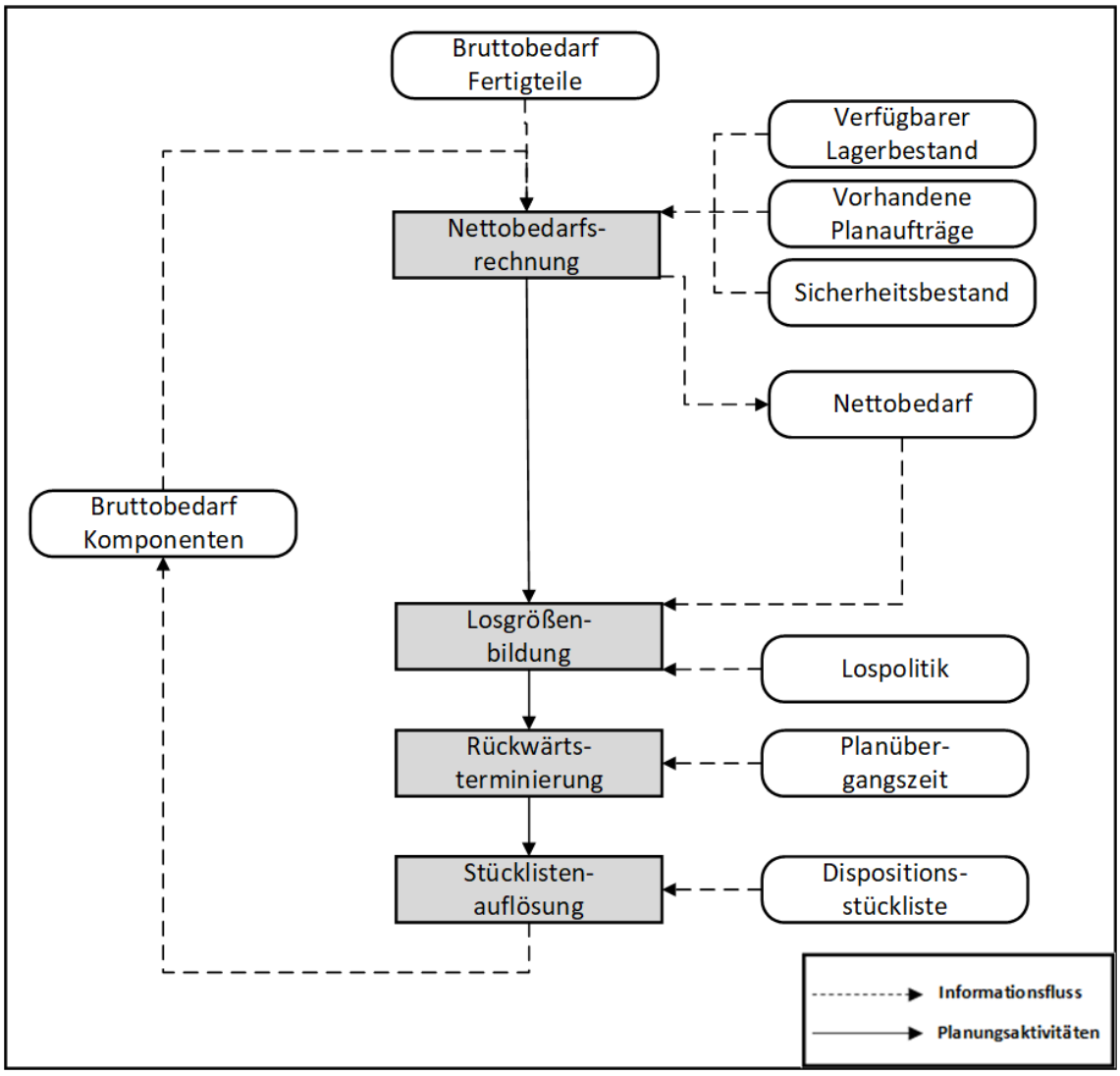


Abbildung 2.6: Überblick MRP Schritte nach [Jodlbauer 2008]

Ausgehend von den in der Programmplanung ermittelten Bruttobedarfen für das jeweilige Produkt wird unter Berücksichtigung des verfügbaren Lagerbestands, bereits vorhandener Planaufträge und des für das Produkt festgelegten Sicherheitsbestandes, der Nettobedarf für die jeweiligen Planungsperioden ermittelt.

Allgemein berechnet sich der Nettobedarf zum Bedarfszeitpunkt t_{Bedarf} anhand folgender Formel:

$$x_{Netto}(t_{Bedarf}) = x_{Brutto}(t_{Bedarf}) - x_{Verfügbar}(t_{Bedarf}) + x_{Sicherheit}(t_{Bedarf}) - \sum_{i=0}^{t_{Bedarf}} x_{WE_{Plan}}(i) + \sum_{i=0}^{t_{Bedarf}} x_{WA_{Plan}}(i) \quad (2.2)$$

mit
x_{Netto} .... Nettobedarf zum Bedarfszeitpunkt t
x_{Brutto} .... Bruttobedarf zum Bedarfszeitpunkt t
$x_{Verfügbar}$ .... verfügbarer Lagerbestand zum Bedarfszeitpunkt t
$x_{Sicherheit}$ .... Sicherheitsbestand in der Periode t
$x_{WE_{plan}}$ .... geplante Wareneingänge in der Periode t
$x_{WA_{plan}}$ .... geplante Warenausgänge in der Periode t

Ist der berechnete Nettobedarf für die Planungsperiode größer null, so wird der definierte Sicherheitsbestand für das Produkt aufgrund der Deckung des Bruttobedarfs unterschritten, sodass für diese Periode ein Planauftrag erzeugt werden muss. Im Falle eines negativen Nettobedarfs oder eines Nettobedarfs gleich null erfolgt keine Erzeugung eines Planauftrages. Die Ermittlung des Lagerbestandes zum Stichtag und des Sicherheitsbestandes des jeweiligen Materials für den Ist-Prozess wird in den Abschnitten 2.1.2.5.1 und 2.1.2.5.5 aufgezeigt.

Im Zuge der Losgrößenbildung wird anhand eines positiven Nettobedarfs die Bedarfsmenge des Planauftrages ermittelt. Basis zur Bestimmung der Losgröße stellt die Lospolitik des betrachteten Produktes dar. Generell kann hierbei zwischen statischen, periodischen und optimierenden Verfahren unterschieden werden, wobei die Wahl des Verfahrens einen direkten Einfluss auf die Frequenz der Produktion des Produktes hat. Eine detaillierte Aufarbeitung der einzelnen Verfahren und der verwendeten Lospolitiken im Ist-Prozess wird in den Abschnitten 2.1.2.6 und 2.1.2.7 vorgenommen.

Aufgrund der Durchführung der Nettobedarfsrechnung und Losgrößenbildung sind sowohl der Bedarfszeitpunkt als auch die Bedarfsmenge für das betrachtete Produkt bekannt. Mithilfe des Prozessschrittes der Rückwärtsterminierung kann anschließend die Bestimmung des Freigabetermins durchgeführt werden. Dieser stellt den spätestmöglichen Zeitpunkt dar, an dem das Produkt produziert bzw. bestellt werden muss, damit es zum Bedarfszeitpunkt zur Verfügung steht. Basis zur Ermittlung des Freigabetermins bildet die Planübergangszeit. Diese beinhaltet den gesamten Zeitraum zur Lieferung und Bearbeitung des Produktes.

Der Freigabetermin kann mithilfe folgender Formel berechnet werden:

$$t_{Freigabe} = t_{Bedarf} - t_{Plan} \tag{2.3}$$

mit
$t_{Freigabe}$ .... Freigabetermin
t_{Bedarf} .... Bedarfstermin
t_{Plan} .... Planübergangszeit

Zu berücksichtigen ist hierbei, dass die Planübergangszeit in Werktagen berechnet wird und sich am Fabrikkalender des betrachteten Werks orientiert. Die detaillierte Berechnung für die Planübergangszeit im Ist-Prozess ist in Abschnitt 2.1.3.2 ersichtlich.

Den Abschluss des Prozesses der Materialbedarfsplanung bildet die Stücklistenauflösung. Für diesen Zweck wird eine Dispositionsstückliste gebildet, in der genau eine Position für jedes zu planende Produkt existiert. Verbrauchsgesteuerte Produkte und Schüttguter werden nicht berücksichtigt.
Für die Darstellung der Dispositionsstückliste wird diese in einzelne Ebenen unterteilt. Die Ebene null (Dispostufe 0) referenziert hierbei auf die Fertigprodukte. Das relevanteste Kriterium der Dispositionsstückliste ist, dass Produkte höherer Stufen nur in Produkte niedrigerer Stufen eingehen [Jodlbauer 2008]. Somit beinhaltet die Dispostufe 1 nur jene Produkte, die direkt für das Fertigteil benötigt werden. Die Produkte der Dispostufe 2 können sowohl auf die Produkte der Stufe 0 als auch die der Stufe 1 referenzieren.

Abbildung 2.7 zeigt einen beispielhaften Aufbau einer Dispositionsstückliste bzw. deren Erstellung aus den Stücklisten zweier Fertigprodukte.

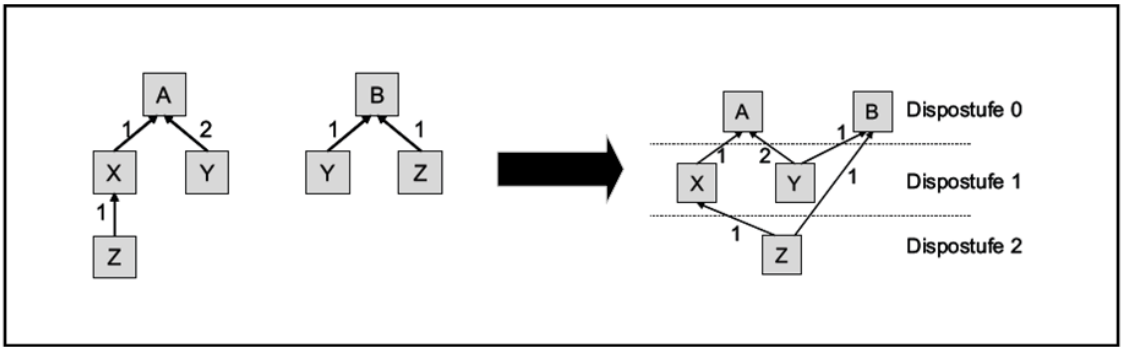


Abbildung 2.7: Dispositionsstückliste nach [Jodlbauer 2008]

Der Bruttobedarf der Produkte der Dispositionsstufe 1 ergibt sich aus den Nettobedarfen der Produkte der Dispositionsstufe 0 und der Berechnung der Anzahl der benötigten Produkte anhand der Stücklistenauflösung. Für das Produkt Y in Abbildung 2.7 errechnet

sich der Bruttobedarf somit aus dem dopplelten Nettobedarf des Fertigteils A und dem einfachen Nettobedarf des Fertigteils B. Der Nettobedarf des Produktes Y kann anschließend analog zu den Produkten A und B über den MRP-Prozess berechnet werden. Der Bedarfszeitpunkt für die Berechnung der Dispositionsstufe 1 entspricht dem Produktionsstart der Dispositionsstufe 0.

Da der Fokus der Arbeit rein auf der Evaluierung der Auswirkungen von Prognosen auf die Produkte der Dispositionsstufe 0 liegt, wird in weiterer Folge nicht genauer auf die Verfahren der Stücklistenauflösung im Ist-Prozess eingegangen. Theoretische Grundlagen zu diesem Thema sind in [Jodlbauer 2008] nachzulesen. Die SAP spezifischen Verfahren sind in [Dickersbach und Keller 2014] dargestellt.

2.1.2.4 Materialbedarfsplanung im Ist-Prozess

Die Materialbedarfsplanung im Ist-Prozess wird analog zum theoretischen Ablauf nach [Jodlbauer 2008] in Abbildung 2.6 durchgeführt. Alle weiteren Abschnitte mit Bezug auf den Ist-Prozess der Materialbedarfsplanung referenzieren somit auf diese Vorgehensweise.

2.1.2.5 Bestände in der Materialbedarfsplanung

Die Kenntnis der exakten Bestandssituation innerhalb des Unternehmens hat einen wesentlichen Einfluss auf die Qualität der Ergebnisse der Materialbedarfsrechnung. Zusätzlich zu den rein physisch vorhandenen Lagerbestand werden auch Reservierungen für Produktions- oder Kundenaufträge, Wareneingänge in Form von Bestellungen oder Fertigungsaufträgen und der Sicherheitsbestand je Produkt in der Berechnung berücksichtigt. Neben der reinen mengenmäßigen Lagerbestandsermittlung ist somit die Bestimmung des für die Disposition verfügbaren Bestandes entscheidend für die exakte Ermittlung des Nettobedarfs der Produkte.

2.1.2.5.1 Theoretische Bestimmung des physischen Lagerbestandes

Der physische Lagerbestand beschreibt den Gesamtbestand eines Produktes innerhalb eines Unternehmens, unabhängig von dessen Verfügbarkeit für die Produktion, zur Deckung von Kundenaufträgen oder sonstiger Prozessschritte innerhalb der Wertschöpfungskette.

Die Ermittlung des physischen Lagerbestandes eines Produktes kann mithilfe des Lager-Durchlaufdiagramms nach [Nyhuis und Wiendahl 2013] erfolgen. Grundidee dieser Darstellungsform ist es, die voneinander unabhängigen Teilprozesse des Warenein- und Warenausgangs getrennt über den betrachteten Zeitraum darzustellen. Die jeweiligen Mengen werden kumuliert über die Zeit bis zum Stichtag aufgetragen. Zu beachten ist, dass im

Gegensatz zur Lagerabgangskurve die Lagerzugangskurve aufgrund des Anfangsbestandes im Untersuchungszeitraum nicht im Koordinatenursprung beginnt.
Mithilfe des Diagramms lässt sich der Lagerbestand nun fortlaufend aus dem senkrechten Abstand zwischen den Kurven des Wareneinn- und Warenausgangs berechnen.

Abbildung 2.8 zeigt eine beispielhafte Darstellung eines Lager-Durchlaufdiagramms für ein Produkt.

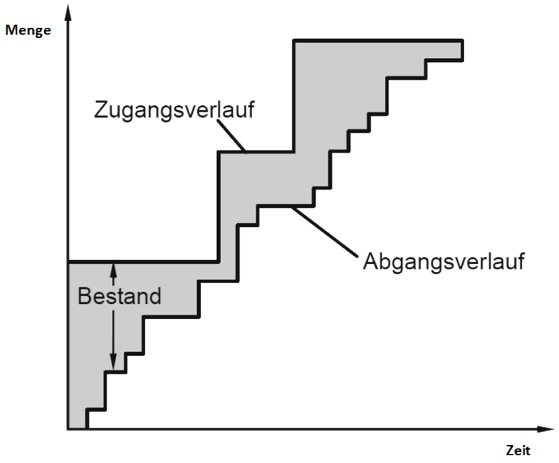


Abbildung 2.8: Lager Durchlaufdiagramm nach [Nyhuis und Wiendahl 2013]

Formal lässt sich der physische Lagerbestand zum Stichtag t mithilfe folgender Formel berechnen:

$$x_{Physisch}(t) = x_{Anfang} + \sum_{i=0}^{t-1} x_{WE}(i) - \sum_{i=0}^{t-1} x_{WA}(i) \tag{2.4}$$

mit
$x_{physisch}$ .... physischer Lagerbestand
x_{Anfang} .... Anfangsbestand
x_{WE} .... Wareneingangsmenge
x_{WA} .... Warenausgangsmenge

2.1.2.5.2 Bestimmung des physischen Lagerbestandes im Ist-Prozess

Die Ermittlung des physische Lagerbestandes eines Produktes im Ist-Prozess erfolgt mithilfe der Auswertung aller zu diesem Produkt zugeordneten Materialbelege im SAP-ERP-System. Der Materialbeleg stellt das zentrale Protokoll zum Nachweise für Warenbewegungen innerhalb des Systems dar und ist die Ausgangsbasis zur Fortschreibung des Bestandes [Kappauf et al. 2017].

Aufgebaut ist der Materialbeleg aus einem Belegkopf und mindestens einer Belegposition. Diese Elemente des Belegs enthalten das Buchungsdatum, das Produkt und die bewegte Menge, sowie Angaben über das betroffene Werk und die Informationen ob es sich bei der Bewegung um einen Wareneinn- oder Warenausgang handelt. Die Informationen des Materialbelegs ermöglichen somit die Berechnung des physischen Lagerbestandes anhand des Durchlaufdiagramms der Lagerhaltung und der Gegenüberstellung der Wareneinn- und Warenausgänge bis zum betrachteten Stichtag.

2.1.2.5.3 Bestandsklassifikation in SAP-ERP-System

Neben der reinen Ermittlung des physischen Lagerbestandes, spielt vor allem die Art des Lagerbestandes und dessen Verfügbarkeit eine zentrale Rolle in der Bestandsführung und Disposition.
Zur Klassifikation des Bestandes werden hierzu im SAP-ERP-System Bestandsarten herangezogen, die eine eindeutige Steuerung der Verwendbarkeit des Bestandes für die Disposition ermöglichen.
Im SAP-Standard sind hierbei die drei nachfolgenden Bestandsarten verfügbar [Kappauf et al. 2017]:

- **Frei verwendbarer Bestand (F)**
  Produkte im freien Bestand besitzen keine Einschränkungen in ihrer Verwendung und sind sowohl für den Verbrauch als auch für die Disposition voll verfügbar. Diese Bestandsart wird für Produkte herangezogen, die keine Qualitätsprüfung durchlaufen müssen bzw. für welche die Prüfung bereits positiv abgeschlossen wurde.

- **Qualitätsprüfbestand (Q)**
  Diese Bestandsart referenziert auf Produkte, die sich aktuell in einem Qualitätsprüfprozess befinden. Die Produkte sind dispositiv verfügbar. Die Entnahme der Produkte zum Verbrauch ist in der Regel allerdings nicht möglich.

- **Gesperrter Bestand (S)**
  Produkte im gesperrten Bestand sind weder dispositiv noch für den Verbrauch verfügbar. Als typisches Beispiel können hier Produkte zugeordnet werden, die im Zuge der Qualitätsprüfung aussortiert und noch nicht verschrottet wurden.

2.1.2.5.4 Bestimmung der Verfügbarkeit des Lagerbestandes im Ist-Prozess

Die in Abschnitt 2.1.2.5.3 beschriebenen SAP-Standard-Einstellungen für die Bestandsarten werden in gleicher Form im Ist-Prozess verwendet. Zusätzlich erfolgt eine Berücksichtigung des Umlagerbestands (U) in der Dispositionsrechnung.
Der Umlagerbestand beschreibt die Menge, die zwischen zwei internen Werken verschoben wird. Zum Stichtag wurde die Menge im abgebenden Werk bereits entnommen, im empfangenden Werk allerdings noch nicht eingebucht. Für das empfangende Werk wird dieser Bestand in der Disposition als erwarteter Wareneingang und im abgebenden Werk als Warenausgang berücksichtigt.

Somit gilt für den verfügbaren Bestand zum Stichtag t und in Abhängigkeit der Bestandsart B des physischen Bestands folgende Formel:

$$x_{Verfügbar}(t) = \sum_{B \in \{F,Q,U\}} x_{Physisch}(t, B) \tag{2.5}$$

2.1.2.5.5 Theoretische Bestimmung des Sicherheitsbestandes

Der Sicherheitsbestand beschreibt jenen Bestandspuffer, der zum Auffangen von Vorhersagefehlern in der Bedarfsberechnung und Abweichungen in der Wiederbeschaffungszeit definiert wird. Ziel der Festlegung eines Sicherheitsbestandes ist es Out-of-Stock Situationen und damit einhergehende Produktionsstillstände oder Lieferunfähigkeiten bestmöglich zu vermeiden.

In Abhängigkeit der Art des betrachteten Artikels wird in Schönsleben 2007 zwischen drei wesentlichen Formen in der Bestimmung des Sicherheitsabstandes unterschieden:

- **Festlegung einer festen Größe**
  Für diesen statischen Ansatz wird unabhängig von Verbrauchsschwankungen eine fixe Sicherheitsbestandshöhe definiert.

- **Berechnung auf Basis von Prognosen**
  In diesem Ansatz wird der Sicherheitsbestand auf Grundlage von Prognosen für einen festgelegten Zeitraum, im Anschluss an den Bedarfstermin des Produktes, berechnet. Soll der Wareneingang für ein Produkt somit z.B. am 1.3.2021 erfolgen und ist für das Produkt ein Sicherheitsbestand von fünf Werktagen definiert, so wird dieser aufgrund von Prognosewerten bis zum 8.3.2021 berechnet.

- **Berechnung auf Basis Vergangenheitsdaten**
  Analog zur Berechnung basierend auf Prognosen, wird in diesem Ansatz ein Zeitraum definiert, in dem Schwankungen im Bedarf gedeckt werden sollen.

Grundlage der Berechnung stellt hierbei allerdings kein Forecast, sondern der Verbrauch aus vergangenen Perioden dar. Typische Ansätze zur Bestimmung des Sicherheitsbestandes sind die Berechnung des durchschnittlichen Tagesbedarfs auf Basis vergangener Wochen oder Monate.

Auf die Einzelheiten der theoretischen Berechnung des Sicherheitsbestandes wird im Rahmen dieser Arbeit nicht genauer eingegangen. Details zu diesem Thema sind beispielsweise in [Schönsleben 2007] oder [Gudehus 2011] einzusehen.

2.1.2.5.6 Bestimmung des Sicherheitsbestandes im SAP-ERP

Für die Bestimmung des Sicherheitsbestandes unterstützt das SAP-ERP-System zwei Varianten. In der statischen Festlegung des Sicherheitsbestandes wird analog zur theoretischen Festlegung eines fixen Sicherheitsbestandes über die Stammdaten des Produktes ein absoluter Wert definiert, der den verfügbaren Lagerbestand für die Dispositionsrechnung reduziert [Dickersbach und Keller 2014].

Alternativ zum statischen Ansatz kann mithilfe von Reichweitenprofilen eine dynamische Bestimmung des Sicherheitsbestandes je Produkt realisiert werden, wobei dieses Verfahren auf die theoretische Berechnung auf Basis von Vergangenheitswerten referenziert. Die Profile definieren den Zeitraum des Sicherheitsbestands im Anschluss an den Bedarfszeitpunkt der Nettobedarfsplanung. Somit passt sich der dynamische Sicherheitsbestand automatisch den Bedarfsänderungen an.
Durch die Definition von Minimal- bzw. Maximalreichweiten kann zusätzlich eine mengenmäßige Begrenzung realisiert werden.

Die Berechnung des dynamischen Sicherheitsbestandes zum Stichtag t erfolgt mithilfe folgender Formel:

$$x_{Sicherheit}(t) = \overline{x_{Bedarf_T}} * t_{Reichweite_A} \tag{2.6}$$

mit
$x_{Sicherheit}$ .... Sicherheitsbestand
$\overline{x_{Bedarf_T}}$ .... Durchschnittlicher Tagesbedarf
$t_{Reichweite_A}$ .... Reichweite in Arbeitstagen

Der durchschnittliche Bedarf wird folgendermaßen berechnet:

$$\overline{x_{Bedarf_T}} = \frac{1}{n_{SB}} * \sum_{i=t_{SB}}^{t-1} x_{WA}[i] \tag{2.7}$$

mit
$\overline{x_{Bedarf_T}}$ .... durchschnittlicher Tagesbedarf
n_{SB} .... Anzahl der zu berücksichtigen vergangenen Arbeitstage
t_{SB} .... Zeitpunkt für Beginn der Berechnung des Sicherheitsbestandes
x_{WA} .... Warenausgangsmenge

Der Zeitraum für die Berechnung des durchschnittlichen Tagesbedarfs bzw. der Zeitpunkt für den Beginn der Berechnung des Sicherheitsbestandes (t_{SB}), kann über das Customizing im jeweiligen Reichweitenprofil definiert werden. Abbildung 2.9 zeigt hierzu ein Beispiel.

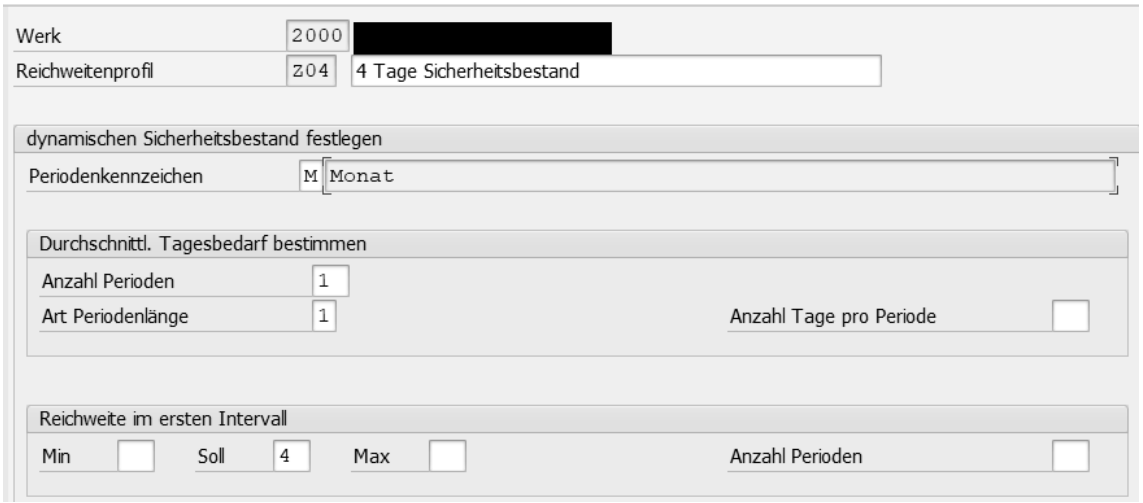


Abbildung 2.9: Beispiel Reichweitenprofil

Über das Periodenkennzeichen wird festgelegt, ob sich die Perioden für den Sicherheitsbestand auf Monate oder Wochen beziehen. Die Anzahl der Perioden gibt den Zeitraum für die Berechnung an. Die Art der Periodenlänge bestimmt zusätzlich, ob es sich um Wochen- oder Arbeitstage handelt.
Bezogen auf das Beispiel in Abbildung 2.9, wird für die Berechnung des durchschnittlichen Tagesbedarfs damit ein Monat auf Basis der Arbeitstage herangezogen. Die Soll-Reichweite wird in Arbeitstagen angegeben und beträgt im Beispiel vier Tage.

Formal ergibt sich der Zeitraum für die Berücksichtigung vergangener Bedarfe zur Berechnung des Sicherheitsbestandes aus folgender Formel:

$$t_{SB} = t - n_{SB} \tag{2.8}$$

mit
t_{SB} .... Beginn Berechnung Sicherheitsbestand
t .... Stichtag der Berechnung
n_{SB} .... Anzahl der zu berücksichtigenden vergangenen Arbeitstage

2.1.2.5.7 Bestimmung des Sicherheitsbestandes im Ist-Prozess

Im Ist-Prozess wird rein die dynamische Bestimmung des Sicherheitsbestandes verwendet. Die Reichweite wird hierbei spezifisch für jedes Produkt und jedes Werk anhand der Reichweitenprofile und auf Basis von Erfahrungswerten definiert und am Materialstamm hinterlegt.

Abbildung 2.10 zeigt die Auswertung der gepflegten Sicherheitsbestände mit Stand März 2022.

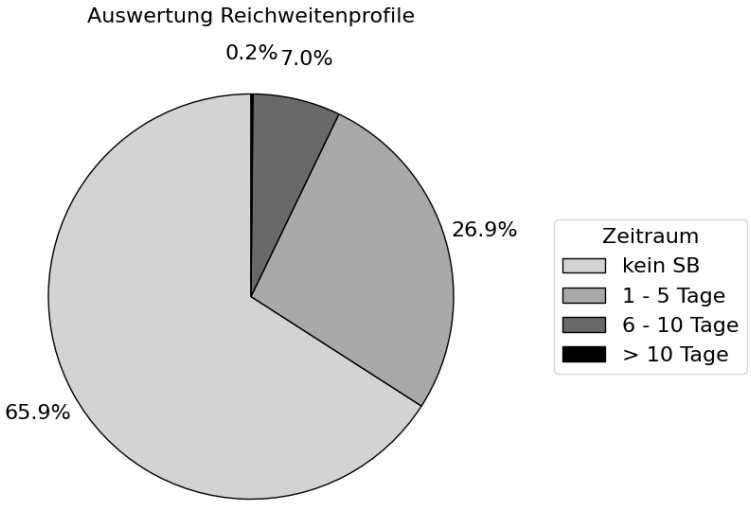


Abbildung 2.10: Auswertung Sicherheitsbestand

Es zeigt sich, dass für über 65 Prozent der dispositiv relevanten Produkte kein Sicherheitsbestand gepflegt ist. Dies kann einerseits durch die der Anforderungen der Lebensmittelindustrie bezüglich der maximalen Lagerdauer begründet werden. Um Ausschüsse aufgrund von abgelaufenen Mindesthaltbarkeitsdaten zu vermeiden, wird für Produkte, die nicht regelmäßig verbraucht werden, kein Sicherheitsbestand gepflegt.
Andererseits beinhaltet dieser Prozentsatz auch jene Produkte, die im Zuge der in Ab-

schnitt 4.2 durchgeführten ABC-Analyse als A-Teile klassifiziert werden.

Details zu den Auswirkungen des fehlenden Sicherheitsbestandes auf die Materialbedarfsplanung werden in Abschnitt 8.4 diskutiert.

2.1.2.6 Theoretische Lospolitiken in der Materialbedarfsplanung

Mithilfe der Lospolitiken können im Prozessschritt der Losgrößenbildung der Materialbedarfsplanung Anzahl und Menge der benötigten Planaufträge anhand des errechneten Nettobedarfs je Produkt bestimmt werden. Allgemein kann zwischen statischen, periodischen und optimierenden Losgrößenverfahren unterschieden werden.

Typische Vertreter der statischen Losgrößenverfahren (Fixed-Order-Quantity, FOQ) stellen die exakte (EX) und feste (FX) Losgrößenberechnung dar. Für diese Verfahren wird in den erzeugten Planaufträgen eine fixe Losgröße von x Einheiten vorgegeben. Wenn der Nettobedarf eine Losgröße übersteigt, so wird ein Vielfaches der vorgegebenen Losgröße gebildet [Zsifkovits 2012].

Im Gegensatz zu den statischen Verfahren mit fixer Losgröße entspricht die Anzahl der Einheiten in den Planaufträgen mit periodischer Losgröße (Fixed-Order-Period, FOP) dem errechneten Nettobedarf der betrachteten Periode. Je nach den Konfigurationsmöglichkeiten des verwendeten ERP-Systems können allerdings Mindestlosgrößen für eine Periode definiert werden. Die Periodenanzahl und Periodendauer wird für jedes Produkt einzeln bestimmt [Jodlbauer 2008].

Optimierende Losgrößenverfahren berücksichtigen in der Berechnung der Anzahl der Einheiten je Planauftrag sowohl losgrößenfixe Kosten, wie beispielsweise Rüstkosten für Maschinen oder Bestellkosten, als auch variable Kosten, wie Kapitalbindungskosten in der Lagerhaltung. Diese Verfahren versuchen basierend auf der gegebenen Kostenstruktur die optimale Losgröße festzulegen. Typische Vertreter sind das Verfahren nach Groff oder das Verfahren nach Silver-Meal [Dickersbach und Keller 2014].
Letztere werden im Rahmen der Arbeit nicht genauer betrachtet und können in [Jodlbauer 2008] nachgelesen werden.

2.1.2.7 Lospolitiken im Ist-Prozess

Bezogen auf den Ist-Prozess wird die Lospolitik anhand eines Kennzeichens der Dispositionslosgröße im Materialstamm, spezifisch für jedes Produkt definiert.
SAP-ERP bietet hierbei in der Standardausführung analog zu den im Abschnitt 2.1.2.6 definierten theoretischen Grundlagen die drei Gruppen der statischen, periodischen und

optimierenden Verfahren an.
Die Auswertung in Abbildung 2.11 für den Ist-Prozess zeigt, dass für die dispositiv relevanten Produkte zwei Verfahren verwendet werden.

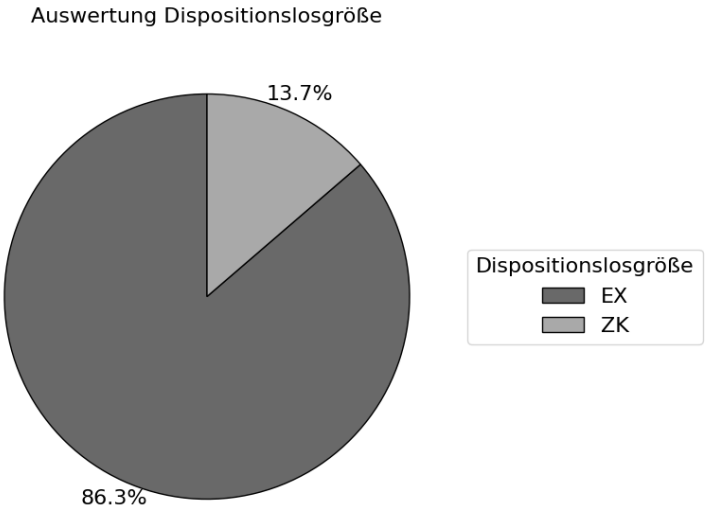


Abbildung 2.11: Auswertung Dispositionslosgröße

Für 86.3 Prozent der Produkte wird das statische Verfahren der exakten Losgrößenberechnung (*EX*) herangezogen. Bei den restlichen 13.7 Prozent wird ein Periodenlosgrößenverfahren angewendet, das mithilfe des Kennzeichens *ZK* definiert wird.
Die Unterschiede in der Berechnung der Periodenlänge und der Periodenlosgröße je Verfahren, werden in Abschnitt 2.1.3 aufgezeigt.

2.1.3 Berechnung der Losgröße und der Periodendauer im Ist-Prozess

Bezogen auf das exakte Losgrößenverfahren (*EX*) wird die Losgröße, wie in Abbildung 2.12 ersichtlich, mithilfe des Rundungswertes bzw. der Mindestlosgröße am Materialstamm berechnet.

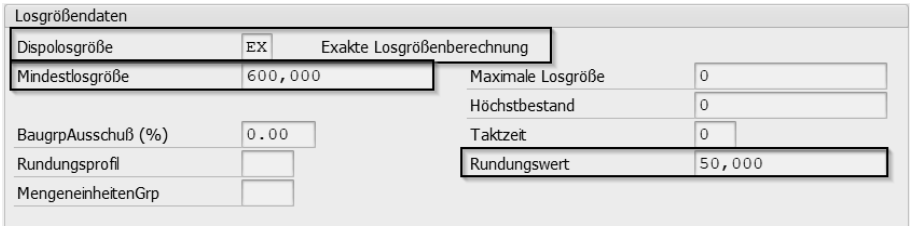


Abbildung 2.12: Losgrößendaten Materialstamm

Die Losgröße wird anhand des errechneten Nettobedarfs in der Periode bestimmt. Falls

für das Produkt ein Rundungswert gepflegt ist, wird die berechnete Losgröße mithilfe dieses Wertes aufgerundet. So wird ein Nettobedarf von beispielsweise 14 525 Stück bei einem Rundungswert von 50000 auf 50000 Stück aufgerundet (vgl. Abbildung 2.12),. Bei einer Überschreitung von 50000 Stück, z.B. 55000 Stück, wird auf ein Vielfaches des Rundungswertes, in diesem Fall 100000 Stück, gerundet.
Unterschreitet die neu berechnete Losgröße eine gepflegte Mindestlosgröße, so wird der Wert der Mindestlosgröße als Gesamtlosgröße übernommen. Zum Beispiel beträgt die gesamte Losgröße somit 600000 Stück.

Formal wird damit folgende Formel verwendet:

$$x_{Los} = max(x_{Los_{min}}, \left\lceil \frac{x_{Netto}}{x_{Runden}} \right\rceil * x_{Runden}) \qquad x_{Runden} \geq 1 \tag{2.9}$$

mit
x_{Los} .... Losgröße
$x_{Los_{min}}$ .... Mindestlosgröße
x_{Netto} .... Nettobedarf
x_{Runden} .... Rundungswert

Bezogen auf die Periodendauer wird im exakten Losgrößenverfahren eine rollierende Planung auf täglicher Basis durchgeführt. Damit entspricht die Periodendauer dem Stichtag t der Berechnung.

$$t_{Periode} = t \tag{2.10}$$

mit
$t_{Periode}$ .... Periodendauer in Tagen
t .... Stichtag der Berechnung

Die Berechnung der Gesamtlosgröße wird für das Periodenlosgrößenverfahren *ZK* analog zur Formel (2.9) durchgeführt. Unterschiede ergeben sich in der Berechnung der Periodendauer.

Die Periodendauer wird anhand eines Planungskalenders bestimmt, der spezifisch für jedes Produkt am Materialstamm definiert ist. Abbildung 2.13 zeigt ein Beispiel eines Planungskalenders.

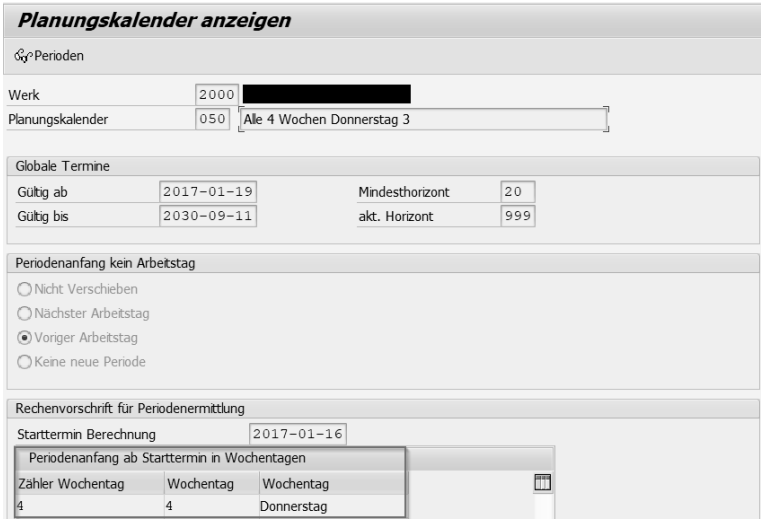


Abbildung 2.13: Beispiel Planungskaldender

Über die Felder *„Zähler Wochentag“* und *„Wochentag“* wird die Periodizität des Beschaffungszeitpunktes festgelegt.
Für Produkte, die dem Planungskalender im Beispiel zugeordnet sind, können Planaufträge somit im Rhythmus von vier Wochen, jeweils am Donnerstag, erstellt werden.
Für die Periodendauer ergibt sich folgende Berechnung:

$$t_{Periode} = n_{Woche} * 7 \tag{2.11}$$

mit
$t_{Periode}$ .... Periodendauer in Tagen
n_{Woche} .... Zähler Wochentag

2.1.3.1 Theoretische Bestimmung der Planübergangszeit in der Materialbedarfsplanung

Im Anschluss an die durch die unterschiedlichen Politiken durchgeführte Losgrößenbildung, wird im Rahmen des Prozesses der Materialbedarfsplanung der Schritt der Rückwärtsterminierung zur Ermittlung des Freigabetermins, durchlaufen . Basis für die Terminierung bildet die Planübergangszeit.

Wie in Abschnitt 2.1.2.3 beschrieben, berechnet sich der Freigabetermin mithilfe folgender Formel:

$$t_{Freigabe} = t_{Bedarf} - t_{Plan} \tag{2.12}$$

mit
$t_{Freigabe}$ .... Freigabetermin
t_{Bedarf} .... Bedarfstermin
t_{Plan} .... Planübergangszeit

Für die Bestimmung der Planübergangszeit wird im Allgemeinen zwischen den Verfahren der konstanten und der variablen Planübergangszeit unterschieden. Im Falle der variablen Übergangszeit wird die Berechnung in Abhängigkeit der Auftragsgröße, der aktuellen Auslastung und der Bestandssituation durchgeführt [Jodlbauer 2008]. Diese Größen werden als Vorgabezeit pro Einheit in der Berechnung berücksichtigt, sodass sich für die dynamische Planübergangszeit folgende Formel ergibt:

$$t_{Plan} = t_0 + X * t_{Vorgabe}(x) \tag{2.13}$$

mit
t_{Plan} .... Planübergangszeit
t_0 .... fixer Summand (z.B. Rüstzeiten+ Vor- und Nachlaufzeit)
X .... Losgröße
$t_{Vorgabe}$ ....Vorgabezeit pro Einheit

Die konstante Planübergangszeit wird alternativ mithilfe einer losgrößenunabhängigen Eigenfertigungszeit berechnet:

$$t_{Plan} = t_0 + t_{Eigenfertigung} \tag{2.14}$$

mit
t_{Plan} .... Planübergangszeit
t_0 .... fixer Summand (z.B. Rüstzeiten +Vor- und Nachlaufzeit)
$t_{Eigenfertigung}$ .... Losgrößenunabhängigen Eigenfertigungszeit

2.1.3.2 Bestimmung der Planübergangszeit im Ist-Prozess

Im Ist-Prozess wird für die Bestimmung der Planübergangszeit das Verfahren der konstanten Übergangszeit herangezogen.

Anstelle des fixen Summanden t_0 gemäß Formel (2.14) wird in der Ist-Planung der Fixierungshorizont herangezogen. Dieser wird spezifisch für jedes Produkt am Materialstamm und in Arbeitstagen gepflegt. Die losgrößenunabhängige Eigenfertigungszeit im Ist-Prozess wird analog zum Fixierungshorizont je Produkt definiert, wodurch sich für die Planübergangszeit im Ist-Prozess folgende Formel ergibt:

$$t_{Plan} = t_{Fixierung} + t_{Eigenfertigung} \tag{2.15}$$

mit
t_{Plan} .... Planüberganszeit
$t_{Fixierung}$ .... Fixierungshorizont
$t_{Eigenfertigung}$ .... losgrößenunabhängigen Eigenfertigungszeit

Abbildung 2.14 zeigt eine Auswertung des Fixierungshorizontes und der Eigenfertigungszeit in den Dispositionssichten 1 und 2 am Materialstamm.

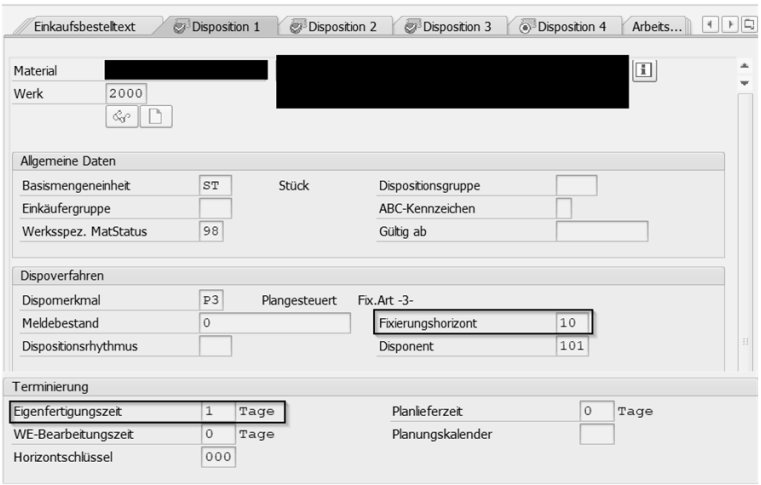


Abbildung 2.14: Beispiel Planübergangszeit

2.1.3.3 Durchführung der Materialbedarfsplanung im Ist-Prozess

Der nachfolgende Algorithmus 1 zeigt den gesamten Ist-Prozess der Materialbedarfsplanung und die darin benötigten Berechnungen. Dieser dient als Übersicht aller in Kapitel 2 beschriebenen Teilprozesse. Die Inputdaten entsprechen den vorhandenen Informationen auf Basis der Materialbelege, Materialstammdaten oder der Customizing-Einstellung im Ist-Prozess.

Algorithm 1: Ist-Materialbedarfsrechnung, Teil 1

Input:

Mengen	**Zeiten**
$x_{Plan_T}[\quad]$ = tägliche Planprimärbedarfe	t = Stichtag der Berechnung
$x_{WE}[\quad]$ = durchgeführte Wareneingänge	$t_{Fixierung}$ = Fixierungshorizont (Arbeitstage)
$x_{WA}[\quad]$ = durchgeführte Warenausgänge	$t_{Eigenfertigung}$ = Eigenfertigungszeit (AT)
$x_{WE_{Plan}}[\quad]$ = geplante Wareneingänge	$t_{Periode}$ = Periodendauer (AT)
$x_{WA_{Plan}}[\quad]$ = geplante Warenausgänge	n_{SB} = Periodenlänge Sicherheitsbestand (SB)
$x_{Anfang} = Anfangsbestand$	$t_{Reichweite_A}$ = Reichweite SB (AT)

Output:
x_{Los} = Ermittelte Losgröße

1 **Berechnung des Enddatums der Planungsperiode (Abschnitt 2.1.3)**

2 $$t_{Ende_P} = \begin{cases} t & \text{Fixed-Order-Quantity (FOQ) Politik} \\ t + t_{Periode} & \text{Fixed-Order-Period (FOP) Politik} \end{cases}$$

3 **Berechnung des Bruttobedarfs (Abschnitt 2.1.1)**

4 $$x_{Brutto}(t) \leftarrow \sum_{i=t}^{t_{Ende_P}} x_{Plan_T}[i]$$

5 **Berechnung des physischen Bestandes (Abschnitt 2.1.2.5.1)**

6 $$x_{Physisch}(t) \leftarrow x_{Anfang} + \sum_{i=0}^{t-1} x_{WE}[i] - \sum_{i=0}^{t-1} x_{WA}[i]$$

7 **Berechnung des verfügbaren Bestandes (Abschnitt 2.1.2.5.3)**

8 $$x_{Verfügbar}(t) \leftarrow \sum_{B \in \{F,Q,U\}} x_{Physisch}(t, B)$$

Algorithm 2: Ist-Materialbedarfsrechnung, Teil 2

Berechnung des Sicherheitsbestandes (Abschnitt 2.1.2.5.5)

$t_{SB} \leftarrow t - n_{SB}$

$\overline{x_{Bedarf_T}} \leftarrow \frac{1}{n_{SB}} * \sum_{i=t_{SB}}^{t-1} x_{WA}[i]$

$x_{Sicherheit}(t) \leftarrow \overline{x_{Bedarf_T}} * t_{Reichweite_A}$

Berechnung des Nettobedarfs (Abschnitt 2.1.2)

$x_{Netto}(t) \leftarrow$

$$x_{Brutto}(t) - x_{Verfügbar}(t) + x_{Sicherheit}(t) - \sum_{i=0}^{t_{Ende_P}} x_{WE_{Plan}}[i] + \sum_{i=0}^{t_{Ende_P}} x_{WA_{Plan}}[i]$$

Berechnung der Planübergangszeit (Abschnitt 2.1.3.2)

$t_{Plan} \leftarrow t_{Fixierung} + t_{Eigenfertigung}$

Berechnung des Freigabedatums (Abschnitt 2.1.3.2)

$t_{Freigabe} \leftarrow t - t_{Plan}$

Berechnung der Losgröße zum Freigabedatum (Abschnitt 2.1.2.7)

$$x_{Los}(t_{Freigabe}) = \begin{cases} max(x_{Los_{min}}, \left\lceil \frac{x_{Netto}}{x_{Runden}} \right\rceil * x_{Runden}) & \text{wenn } x_{Netto} > 0 \\ 0 & \text{sonst} \end{cases}$$

3 Einbindung des Forecasts

In Kapitel 2 wurden der Ist-Prozess der Produktionsplanung und die dazugehörigen theoretischen Grundlagen mithilfe des MPR II Prozesses nach [Jodlbauer 2008] erläutert. Aufbauend auf diesen Informationen wird in den nachfolgenden Abschnitten auf die unterschiedlichen Verfahren zur Erstellung von Prognosen und deren Einbindung in den Gesamtprozess eingegangen.
Zur Schaffung eines einheitlichen, theoretischen Verständnisses werden zu Beginn des Kapitels zusätzlich der Aufbau und die für diese Arbeit relevanten Eigenschaften von Zeitreihen betrachtet. Diese Informationen bilden die Grundlage für die Erläuterung der einzelnen Schritte der Prognoseerstellung und der darin verwendeten Modelle.

3.1 Theoretische Grundlagen Zeitreihen

Unter einer Zeitreihe wird die Repräsentation zeitlich geordneter reeller Werte einer spezifischen Problemstellung verstanden, die über einen diskreten Zeitraum aufgezeichnet oder verarbeitet werden. Beispielsweise stellen die wöchentlichen Kundenbedarfe eines Produktes innerhalb des vergangenen Jahres eine typische Zeitreihe im Rahmen dieser Arbeit dar.
Die einzelnen Werte der Zeitreihe können als Zufallsvariablen aufgefasst werden.
Sind die Zeitabstände zwischen diesen Zufallsvariablen in einem geeigneten Messsystem ohne Einschränkungen gleich eins, so kann folgendes grundlegendes Modell definiert werden:

$$X = (X_t : t \in T) \qquad T = \mathbb{N} \text{ oder } \mathbb{Z} \tag{3.1}$$

mit Zufallsvariablen $X_t, t \in T$ [Kreiss und Neuhaus 2006].

Um die Entwicklung einer Zeitreihe einschätzen zu können, ist es nötig, deren Gesetzmäßigkeiten zu identifizieren. Eine Zerlegung der Zeitreihe in ihre unterschiedlichen Bestandteile bietet die Möglichkeit, diese Informationen für weitere Analysen zur Verfügung zu stellen [Bourier 2010]. In den Abschnitten 3.1.1 und 3.1.1.1 werden hierzu die einzelnen Komponenten und deren Zusammensetzung aufgezeigt.

3.1.1 Komponenten einer Zeitreihe

Die Komponenten einer Zeitreihe beschreiben die unterschiedlichen Einflussgrößen, die auf die einzelnen Werte einwirken und deren weiteren Verlauf bestimmen.

Den Grundbestandteil bildet die glatte Komponente, die betriebswirtschaftlich als Trend (T) bezeichnet wird. Dieser beschreibt die langfristige Ausrichtung der Zeitreihe, um die sich die einzelnen Werte im Zeitverlauf bewegen. Die Einflussgrößen auf den Trend ändern sich in der Regel nur langsam, sodass sich ein glatter Kurvenverlauf ergibt. Abbildung 3.1 zeigt eine qualitative Darstellung einer Trendkomponente für einen Absatzverlauf im Zeitraum von acht Zeiteinheiten.

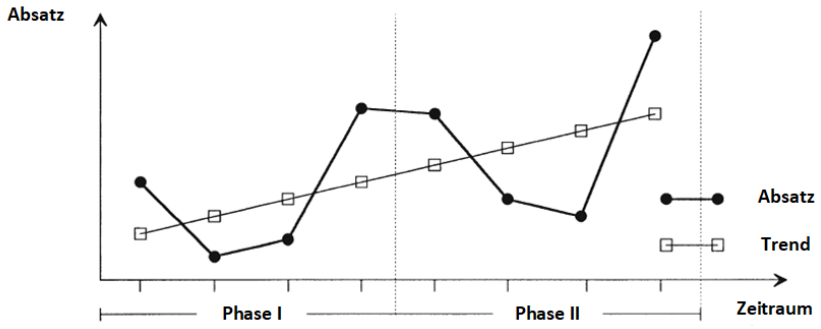


Abbildung 3.1: Trend Komponente nach [Bourier 2010]

Die saisonale oder zyklische Komponente S beschreibt sich regelmäßig wiederholende Schwankungen über einzelne Perioden. Als typisches Beispiel auf Basis von Lebensmitteln kann hier der Verkauf von Eis betrachtet werden. So ist zu erwarten, dass der Konsum in den Sommermonaten höher ausfällt als in den kälteren Wintermonaten, wodurch sich die Verkaufszahlen zyklisch erhöhen und wieder senken.
Allgemein wird eine Schwankungsphase durch die Dauer der Phase, die Anzahl der Abschnitte in der Phase und die Abweichungen vom Trend in den einzelnen Abschnitten beschrieben. Zu beachten ist, dass je nach Granularität der Daten auch mehrere zyklische Komponenten in der Zeitreihe enthalten sein können. Bei einer tageweisen Erhebung der Daten können beispielsweise sowohl jährliche und monatliche als auch wöchentliche Zyklen enthalten sein.

Analog zur Trend-Komponente zeigt Abbildung 3.2 den qualitativen Einfluss der Saison-Komponente am Beispiel der Absatzentwicklung für acht Zeiteinheiten. Die schwarzen Rechtecke repräsentieren jenen Absatz, der sich bei einer reinen Betrachtung von Trend- und Saison-Komponente ergeben hätte.
Wie in Abbildung 3.2 ersichtlich ist, ergibt sich trotz der Berücksichtigung der Trend- und Saison-Komponente eine Differenz zu den realen Werten.

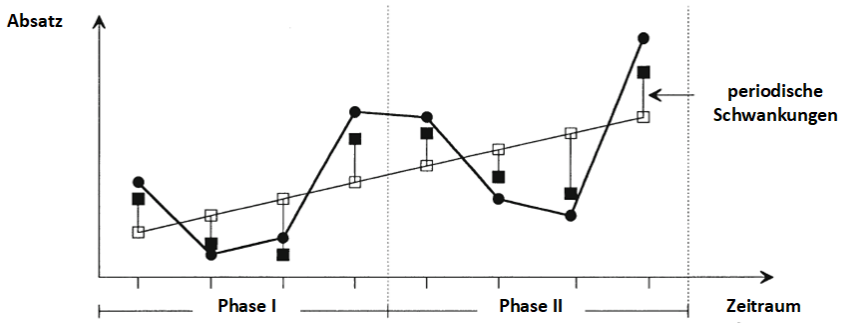


Abbildung 3.2: Saison Komponente nach [Bourier 2010]

Verantwortlich hierfür ist die Rest-Komponente R der Zeitreihe, die meist unbekannte Größen darstellt, die unregelmäßig in ihrer Intensität und Richtung einwirken.
Typische Beispiele hierfür sind Zusatznachfragen aufgrund von unbekannten Werbeaktionen durch die Kunden oder Produktionsausfälle bedingt durch Streik oder gesetzlicher Einschränkungen.

Am Beispiel der Absatzentwicklung wird die Rest-Komponente in Abbildung 3.3 durch die senkrechten Linien zwischen den realen Werten und den schwarzen Rechtecken dargestellt, die Trend und Saison repräsentieren.

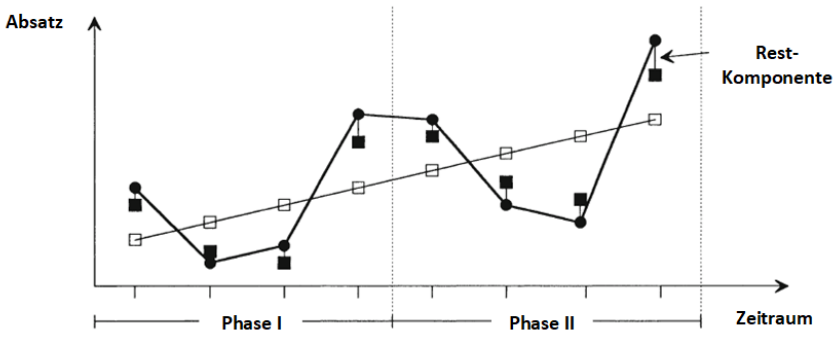


Abbildung 3.3: Rest Komponente nach [Bourier 2010]

Für die Berechnung der einzelnen Komponenten, stehen unterschiedliche Verfahren zur Verfügung. Bezogen auf die Trend- und Saisonkomponente werden vor allem die Methode der kleinsten Fehlerquadrate, das Periodogrammverfahren und das Verfahren nach Loess verwendet.
Das Verfahren nach Loess wird im Rahmen dieser Arbeit zur Analyse der Zeitreihen herangezogen. Für dessen Herleitung wird auf den Beitrag von [Cleveland et al. 1990] verwiesen.

3.1.1.1 Zusammensetzung der Komponenten einer Zeitreihe

Auf Basis der in Abschnitt 3.1.1 dargestellten Komponenten der Zeitreihe kann ein einzelner Zeitreihenwert x_t als Funktion von Trend T, Saison S und Rest R anhand der nachfolgenden Formel dargestellt werden.

$$x_t = f(T_t, S_t, R_t) \qquad t = (1 \dots n) \tag{3.2}$$

Die Zusammensetzung der Komponente erfolgt im einfachsten Fall rein additiv oder multiplikativ. Die Entscheidung über die Art der Zusammensetzung wird empirisch ermittelt.

Für den multiplikativen Fall kann somit folgende Formel definiert werden.

$$x_t = T_t * S_t * R_t \qquad t = 1 \dots n \tag{3.3}$$

Ansonsten erfolgt die Zusammensetzung der Komponenten additiv anhand Formel (3.4):

$$x_t = T_t + S_t + R_t \qquad t = 1 \dots n \tag{3.4}$$

3.1.2 Eigenschaften von Zeitreihen

Neben den einzelnen Komponenten, deren Ermittlung und der additiven oder multiplikativen Zusammensetzung stellt sich in der Analyse von Zeitreihen die Frage, in welcher Relation die einzelnen Werte zueinander stehen und welche Eigenschaften diese aufweisen. Vor allem für die statistischen Modelle zur Prognoseerstellung, die in Abschnitt 6.1 beschrieben werden, sind die nachfolgend aufgezählten Eigenschaften von Zeitreihen eine Grundvoraussetzung in der Verwendbarkeit der Modelle bzw. in der Parameterauswahl. Im Folgenden werden aus diesen Gründen die für diese Arbeit relevanten Eigenschaften der Stationarität, Autokovarianz, Autokorrelation und partiellen Autokorrelation beschrieben. Für weiterführende Informationen zu den Eigenschaften von Zeitreihen wird auf die Beiträge von [Kreiss und Neuhaus 2006] sowie [Nielsen 2019] verwiesen.

3.1.2.1 Stationarität von Zeitreihen

Allgemein kann zur Erklärung der Stationarität eines Prozesses bzw. einer Zeitreihe die in [Nielsen 2019] verwendete allgemeine Definition herangezogen werden:

> **Definition:**
> Ein Prozess ist stationär, wenn für alle möglichen Zeitverschiebungen h die Verteilung $y_t, y_{t+1}, ..., y_{t+h}$ nicht von t abhängig ist [Nielsen 2019].

Vereinfacht bedeutet das, dass ein stochastische Prozess stationär ist, wenn der Mittelwert und die Varianz der betrachteten Zeitreihe über die Zeit konstant sind und die Kovarianz zwischen zwei Zeitpunkten nur von der Länge der Zeitverschiebung abhängt, nicht aber vom Zeitpunkt, an dem gemessen wird [Kreiss und Neuhaus 2006].

Viele der in der praktischen Zeitreihenprognose verwendeten, statistischen Modelle setzen in ihrer Anwendbarkeit voraus, dass die betrachtete Zeitreihe stationär ist. Aus diesem Grund wurden unterschiedliche statistische Tests zur Feststellung der Stationarität von Zeitreihen entwickelt. Der Augmented-Dickey-Fuller-Test und der Kwiatkowski–Phillips–Schmidt–Shin-Test (KPSS) stellen hierbei zwei weitverbreitete Anwendungsverfahren dar [Nielsen 2019].
Da im Rahmen dieser Arbeit auf eine detaillierte, mathematische Erklärung dieser Verfahren verzichtet wird, wird hierzu auf die Beiträge von [Dickey und Fuller 1979] und [Kokoszka und Young 2016] verwiesen.

Falls ihm Rahmen der Tests festgestellt wird, dass die betrachtete Zeitreihe nicht stationär ist, gibt es unterschiedliche Verfahren um die Stationarität herzustellen. Diese werden in Abschnitt 5.5 der Datenaufbereitung genauer betrachtet.

3.1.2.2 Empirische Autokorrelation

Die empirische Autokorrelation beschreibt die Korrelation einer Zeitreihe mit einer zeitlich verschobenen Kopie von sich selbst. Die Autokorrelation für eine Verschiebung von zwei Perioden (Lags) misst damit die Korrelation zwischen allen Beobachtungen der Originalzeitreihe und derselben Zeitreihe, die um zwei Perioden nach vorne verschoben ist [Hillier und Liebermann 2014].
Damit stellt die Berechnung der Autokorrelation ein Hilfsmittel zum besseren Verständnis der Eigenschaften der jeweils betrachteten Zeitreihe dar. Abbildung 3.4 zeigt ein Beispiel für eine Zeitreihe mit saisonalem Verhalten anhand der monatlichen Absatzzahlen im Zeitraum zwischen 01.01.2018 und 31.12.2021.
Diese saisonalen Gesetzmäßigkeiten sind auch in der Darstellung der Autokorrelation

(rechtes Bild) ersichtlich mit maximalen, positiven Korrelationen bei einer Verschiebung von 12 bzw. 24 Monaten.

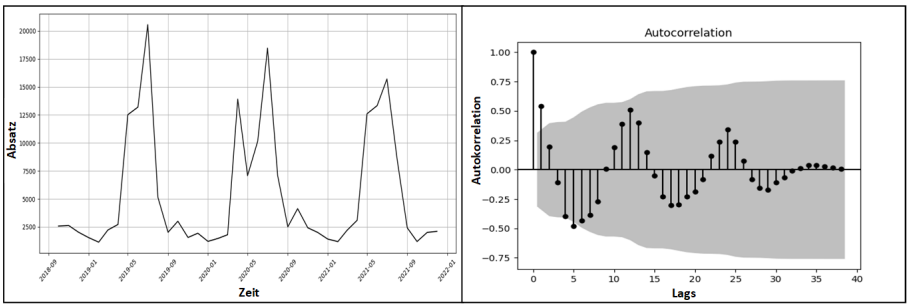


Abbildung 3.4: Beispiel Autokorrelation

Bei einer Zunahme der Komplexität der betrachteten Zeitreihe ist die optische Bestimmung der Gesetzmäßigkeiten oftmals nicht mehr möglich. Die Berechnung der Autokorrelation stellt hierbei ein Hilfsmittel dar, um diese Eigenschaften trotz zunehmender Volatilität zu bestimmen.

Die Werte der Autokorrelation bewegen sich im Bereich von +1 bis -1, wobei diese folgendermaßen interpretiert werden können:

- +1 - starke positive Korrelation
- 0 - keine Korrelation
- -1 - starke negative Korrelation

Allgemein gilt, dass alle Zeitreihen bei einem Lag von 0 eine Autokorrelation von 1 aufweisen. Der grau hinterlegte Bereich in der Darstellung der Autokorrelation stellt das Konfidenzintervall dar. Werte, die diesen Bereich überschreiten, stellen eine signifikate Autokorrelation dar, da die Nullhypothese (H_0: Es besteht keine Autokorrelation) abgelehnt wird. Die Berechnung des Konfidenzintervalls erfolgt im Beispiel mithilfe der Formel nach Bartlett. Für deren Herleitung wird auf den Beitrag von [Francq und Zakoïan 2009] verwiesen.

Anwendung finden die Informationen der Autokorrelation vor allem in der Parameterauswahl zur Berechnung der Prognosen mithilfe von Moving-Average-Modellen (MA(q)). Die Vorgehensweise zur Bestimmung der Parameter wird in Abschnitt 6.1.2.1 genauer betrachtet.

Für die formale Berechnung der Autokorrelation wird auf die Inhalte von [Kreiss und Neuhaus 2006] verwiesen.

3.1.2.3 Empirische partielle Autokorrelation

Die partielle Autokorrelation einer Zeitreihe beschreibt analog zur Autokorrelation die Korrelation zwischen zwei Punkten derselben Zeitreihe zu einem früheren Zeitpunkt. Bei partiellen Autokorrelation werden jedoch die linearen Abhängigkeiten kürzerer Intervalle nicht berücksichtigt. [Hillier und Liebermann 2014].

Abbildung 3.5 zeigt analog zur Autokorrelation ein Beispiel der partiellen Autokorrelation anhand der monatlichen Absatzzahlen im Bereich von 01.01.2018 bis 31.12.2021 derselben Zeitreihe wie in Abschnitt 3.1.2.2.

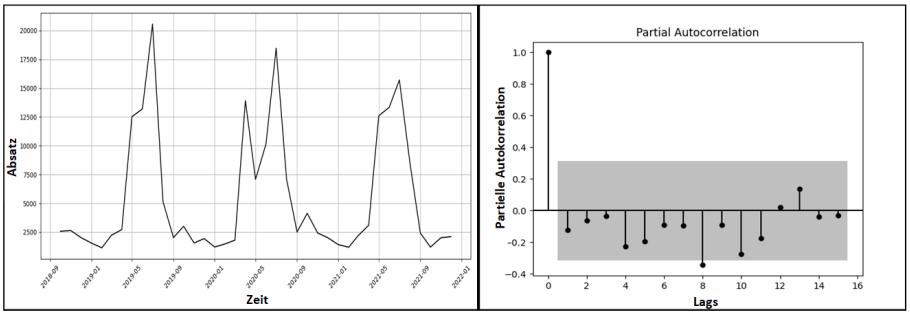


Abbildung 3.5: Beispiel partielle Autokorrelation

Die Entfernung der linearen Abhängigkeiten zwischen den einzelnen Verschiebungen führt dazu, dass die erste positive Korrelation im Beispiel nach einer Verschiebung von 13 Lags und damit ungefähr nach einem Jahr auftritt. Allgemein zeigt die partielle Autokorrelation die eindeutigen Informationen für die spezifische Zeitreihe mit einer Verschiebung von 0 zu einer Zeitreihe mit einer Verschiebung h, ohne den Einfluss redundanter Korrelationen früherer Zeitverschiebungen.

Die Informationen der partiellen Autokorrelation werden vor allem zur Bestimmung der Parameter der autoregressiven Modelle (AR(p)) verwendet. Das genaue Vorgehen wird hierzu in Abschnitt 6.1.1.1 betrachtet.

Analog zur Autokorrelation beschreibt der graue Bereich das Signifikanzniveau und die Werte bewegen sich im Bereich +1 bis -1. Für die formale Berechnung der partiellen Autokorrelation wird auf die Inhalte von Kreiss und Neuhaus 2006 verwiesen.

3.2 Einbindung des Forecasts in den MRP II Ist-Prozess

Die in Abschnitt 3.1 behandelten Eigenschaften und Komponenten von Zeitreihen stellen die Grundvoraussetzung für die Erstellung von Prognosen und die Anwendung der dazu benötigen Modelle dar. Vorab stellt sich allerdings die Frage, wo in Bezug auf dem in Abschnitt 2.1 beschriebenen Gesamtprozess der Produktionsplanung eine Prognose angewandt werden kann.
Wird der Ausschnitt der Langfrist- und Mittelfristplanung in Abbildung 3.6 betrachtet, so zeigt sich, dass drei mögliche Anwendungspunkte zur Verfügung stehen.

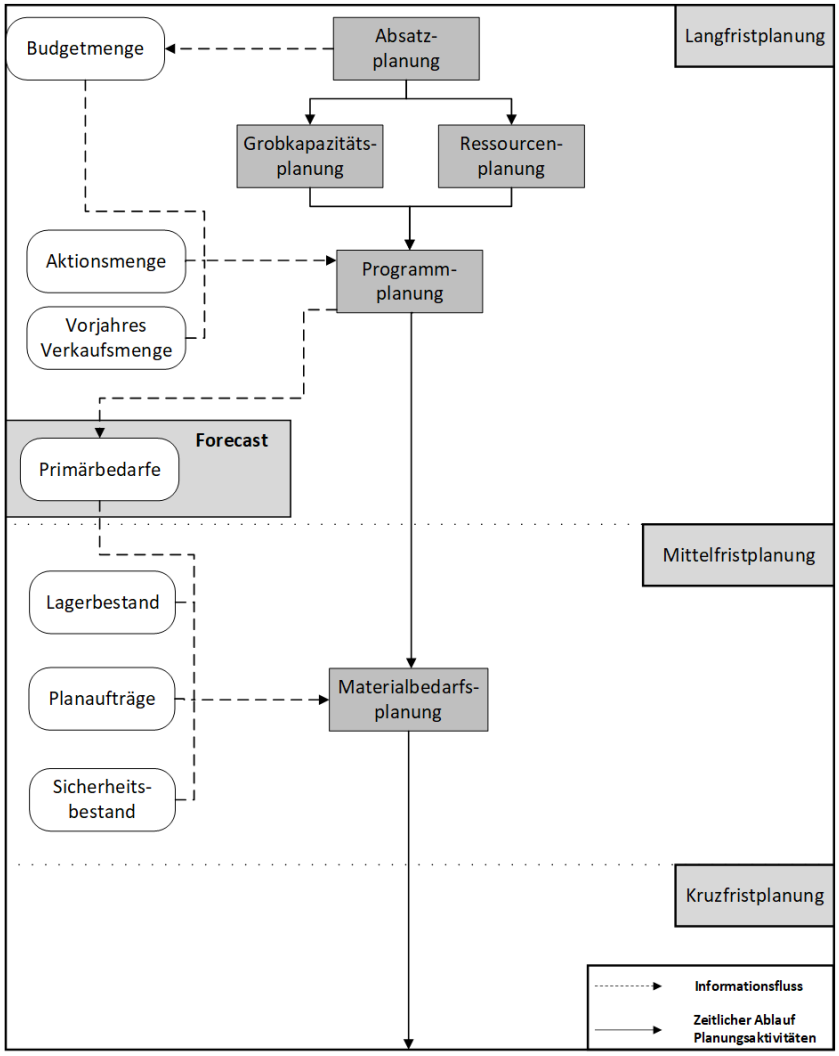


Abbildung 3.6: Einbindung des Forecasts in den MRP II Prozess nach [Jodlbauer 2008]

Einerseits kann eine langfristige Prognose im Bereich von einem Jahr zur Unterstützung der Absatzplanung und Bestimmung der daraus resultierenden Budgetmenge für das jeweilige Produkt herangezogen werden.
Andererseits stellt die Planung der Primärbedarfe im Zuge der Programmplanung den zweiten Anwendungsfall dar. Prognosen im Zeitraum von drei Monaten können dem Disponenten als zusätzliche Information neben der Budgetmenge aus der Absatzplanung, der Verkaufsmenge des Vorjahres und der aktuellen Aktionsmenge zur Bestimmung der Planmenge je Geschäftsperiode zur Verfügung gestellt werden. Der tägliche Plan- bzw. Primärbedarf des betrachteten Produktes würde sich damit weiterhin analog zu der in Abschnitt 2.1.1 Formel (2.1) berechnen:

$$x_{Plan_T} = \frac{x_{Plan}}{t_{Arbeit}} \tag{3.5}$$

mit
x_{Plan_T} .... Planmenge pro Tag
x_{Plan} .... Gesamte Planmenge
t_{Arbeit} .... Arbeitstage in der Geschäftsperiode

Durch die Planungshorizonte von einem Jahr für die Absatzplanung bzw. von drei Monaten für die Programmplanung stellen diese beiden Eingriffspunkte einen langfristigen Ansatz im Hinblick auf den Prognosehorizont dar. Zudem würde weiterhin eine gleichmäßige Verteilung der Bedarfe aufgrund der Ergebnisse der Programmplanung durchgeführt werden.
Auf Basis der Informationen in den durchgeführten Workshops wurde aus diesem Grund im Rahmen dieser Arbeit ein dritter möglicher Ansatz zur Berechnung der Prognosehorizonte herangezogen.
Dieser stellt eine Alternative zur statischen gleichmäßigen Verteilung der Primärbedarfe über eine gesamte Geschäftsperiode von einem Monat dar. In Abhängigkeit des Dispositionsverfahrens und der Planübergangszeit in Form des Fixierungshorizontes und der statischen Eigenfertigungszeit wird in diesem Ansatz der Planungshorizont dynamisch je Produkt bestimmt.
Die Prognosen werden auf Basis wöchentlicher anstatt von monatlicher oder jährlicher Absatzzahlen durchgeführt. Diese Granularität der Prognosen ist für den aktuellen operativen Ablauf ausreichend, sodass keine Prognose von Tagesbedarfen benötigt wird.

3.3 Berechnung der Prognosehorizonte im Ist-Prozess

Wie in Abschnitt 3.2 beschrieben wurde, berechnet sich der Prognosehorizont im Rahmen dieser Arbeit anhand des Dispositionsverfahrens des jeweiligen Produktes.
Für ein Produkt mit der exakten Losgrößenpolitik (EX) ergibt sich die in Abbildung 3.7 ersichtliche Berechnung des Prognosezeitraums.

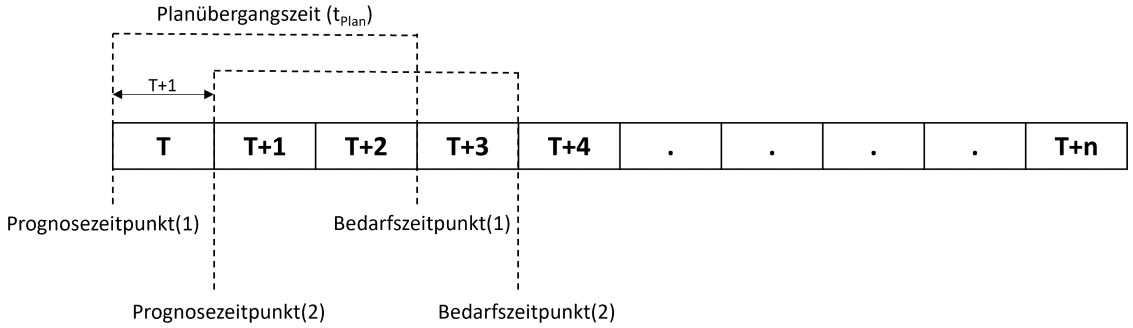


Abbildung 3.7: Prognosezeitraum exaktes Losgrößenverfahren

In dieser Berechnungsform wird der Nettobedarf täglich für einen zukünftigen Bedarfstermin berechnet. Der Prognosezeitraum zwischen dem Tag der Berechnung des zukünftigen Bedarfs (Prognosezeitpunkt 1) und dem Tag des geplanten Wareneingangs der gefertigten Produkte (Bedarfszeitpunkt 1) wird hierbei durch die Planübergangszeit t_{Plan} bestimmt. Dieser Zeitraum stellt den letztmöglichen Anpassungspunkt des Primärbedarfs vor dem Beginn der Produktion dar. Kurzfristigere Mengenänderungen sollen, laut Informationen aus den durchgeführten Workshops sowohl in der im Rahmen dieser Arbeit durchgeführten Simulation als auch in der täglichen operativen Produktionsplanung vermieden werden. Somit ergibt sich für Produkte mit exakter Losgröße folgende Formel zur Berechnung des Prognosezeitraums:

$$t_{Prognose_{EX}} = t_{Plan} = t_{Fixierung} + t_{Eigenfertigung} \tag{3.6}$$

mit
$t_{Prognose_{EX}}$ .... Prognosezeitraum exakte Losgrößenpolitik
t_{Plan} .... Planüberganszeit
$t_{Fixierung}$ .... Fixierungshorizont
$t_{Eigenfertigung}$ .... Losgrößenunabhängigen Eigenfertigungszeit

Für Produkte, die anhand des Periodenlosgrößenverfahren (ZK) berechnet werden, ergibt sich für die Berechnung des Prognosezeitraums die in Abbildung 3.8 dargestellte Berechnungsgrundlage.

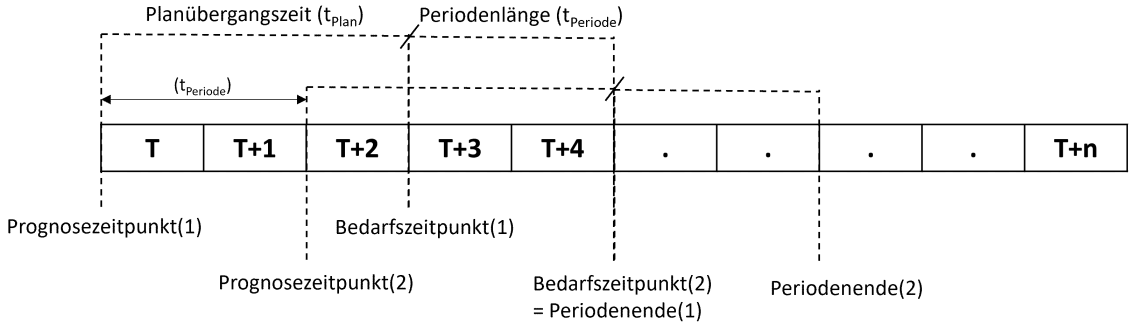


Abbildung 3.8: Prognosezeitraum Periodenlosgrößenverfahren

Im Gegensatz zu Fertigteilen mit exakter Losgröße verlängert sich in diesem Fall der Prognosezeitraum um die spezifische Periodenlänge des jeweiligen Produktes. Diese errechnet sich, wie in Abschnitt 2.1.3.2 beschrieben, anhand des dem Produkt zugeordneten Planungskalenders. Bezogen auf Abbildung 3.8 ergibt sich folgendes Vorgehen:
Analog zur exakten Losgröße wird die Bedarfsberechnung (Prognosezeitpunkt 1) am letztmöglichen Tag der Mengenänderung vor der Produktion (Bedarfszeitpunkt 1) durchgeführt. Die Differenz zwischen Prognosezeitpunkt 1 und Bedarfszeitpunkt 1 entspricht der Planübergangszeit des jeweiligen Produktes. Zusätzlich muss im Falle des Periodenlosgrößenverfahrens der Bedarf innerhalb der Periode berücksichtigt werden,in der keine Produktion durchgeführt wird. Dieser Zeitraum zwischen Bedarfszeitpunkt 1 und Periodenende 1 entspricht der gesamten Periodenlänge des Produktes.

Der Prognosezeitraum für Produkte mit Periodenlosgröße ZK errechnet sich somit anhand folgender Formel:

$$t_{Prognose_{ZK}} = t_{Plan} + t_{Periode} = t_{Fixierung} + t_{Eigenfertigung} + n_{Woche} * 7 \qquad (3.7)$$

mit
$t_{Prognose_{ZK}}$ .... Prognosezeitraum Periodenlosgröße
t_{Plan} .... Planüberganszeit
$t_{Periode}$ .... Periodendauer in Tagen
$t_{Fixierung}$ .... Fixierungshorizont
$t_{Eigenfertigung}$ .... losgrößenunabhängigen Eigenfertigungszeit
n_{Woche} .... Anzahl der Wochen zwischen zwei Perioden

Im Unterschied zum exakten Losgrößenverfahren, wird die Bedarfsrechnung für Produkte mit Periodenlosgröße nicht täglich durchgeführt. Der zeitliche Abstand zwischen Progno-

sezeitpunkt 1 und Prognosezeitpunkt 2 beträgt somit nicht $T + 1$, sondern entspricht der Periodenlänge $t_{Periode}$.

3.3.1 Theoretische Grundlagen der Prognoseerstellung

Bei der Erstellung von Prognosen kann prinzipiell zwischen zwei unterschiedlichen Ansätzen unterschieden werden.
Qualitative Verfahren basieren auf Schätzungen von meist mehreren Personen und versuchen auf Basis dieser Erkenntnisse die zukünftigen Bedarfe vorherzusagen. Typische Vertreter dieser Verfahren sind Expertenschätzungen oder die Delphi-Methode. Diese werden vor allem dann angewandt, wenn keine aussagekräftigen Vergangenheitsdaten für den jeweiligen Prognosegegenstand zur Durchführung von quantitativen Prognosen verfügbar sind.
Quantitative Verfahren basieren hingegen auf der Überlegung, dass zukünftige Verläufe aufgrund von Vergangenheitsdaten abgeleitet werden können und damit einem gewissen Muster folgen [Heiserich et al. 2011]. In diese Kategorie fallen die statistischen Modelle und neuronalen Netze, die in Kapitel 6 betrachtet werden.

Da im Rahmen dieser Arbeit der Fokus auf der Evaluierung der Auswirkungen von quantitativen Prognosen auf den Produktionsplanungsprozess liegt, wird in den weiterführenden Abschnitten nicht näher auf die unterschiedlichen qualitativen Verfahren eingegangen. Für diese Verfahren wird auf die Beiträge von [Heiserich et al. 2011] und [Gudehus 2011] verwiesen.

3.3.1.1 Quantitative Verfahren zur Prognoseerstellung

Basierend auf der verfügbaren und verwendeten Datenmenge, kann in der Erstellung von quantitativen Prognosen zwischen zwei unterschiedlichen Vorgehensweisen unterschieden werden. Univariate Zeitreihenmodelle (Singulärprognosen) berücksichtigen ausschließlich die Vergangenheitsdaten des jeweiligen Prognosegegenstandes. So werden beispielsweise für die Vorhersage von Kundenbedarfen für die nächsten drei Wochen nur die Kundenbedarfe aus der Vergangenheit herangezogen. Weitere Einflussfaktoren werden in der univariaten Zeitreihenprognose nicht berücksichtigt. Autoregressive-Integrated-Moving-Average-Modelle (ARIMA) stellen typische Vertreter von univariaten Zeitreihenmodellen dar. Diese werden in Abschnitt 6.1 beschrieben.
Sind für einen Zeitpunkt weitere Informationen verfügbar und sollen diese auch in der Erstellung der Prognose berücksichtigt werden, so wird ein multivariater Modellansatz verwendet. Die Abschätzung der Relevanz von alternativen Informationen auf den Prognosegegenstand kann durch eine Korrelationsanalyse und die Erstellung einer Korrelationsmatrix vorgenommen werden. In den statistischen Modellansätzen wird das Vector-

Autoregression-Modell (VAR) typischerweise zur Prognose multivariater Zeitreihen herangezogen. Da auf diesen Modellansatz im Rahmen dieser Arbeit nicht genauer eingegangen wird, wird hierzu auf die Beträge von [Kreiss und Neuhaus 2006] und [Nielsen 2019] verwiesen.

Neuronale Netze bieten je nach Art der Implementierung die Möglichkeit sowohl univairate als auch multivariate Prognosen zu erstellen. Die unterschiedlichen Modellarten werden in Abschnitt 6.2 aufgezeigt.

3.3.1.2 Einfluss des Prognosehorizonts auf die Prognoseerstellung

Neben der Entscheidung der Datenmenge und der damit einhergehenden Verwendung von univariaten oder multivariaten Modellansätzen existieren auch Verfahren in der Prognoseerstellung in Bezug auf den Prognosehorizont. Falls für die Problemstellung eine Vorhersage benötigt wird, die länger als eine Zeiteinheit (Single-Step) ausfällt, so können unterschiedliche Verfahren für die Prognose längerer Zeiträume (Multi-Step-Prognose) angewandt werden. Laut [Bontempi et al. 2013] kann hierzu zwischen den folgenden Basis-Strategien unterschieden werden:

- Rekursive Multi-Step-Strategie
- Direkte Multi-Step-Strategie
- Multiple-Output-Strategie

3.3.1.2.1 Rekursive Multi-Step-Strategie

Die rekursive Multi-Step-Strategie basiert auf der Erstellung eines Single-Step-Models *f*, das anhand folgender Formel trainiert wird:

$$y_{t+1} = f(y_t, ..., y_{t-n+1)}) \tag{3.8}$$

für
$t = n \dots N - 1$

Anschließend wird das Modell rekursiv für die Prognose der weiteren Zeitpunkte verwendet und der Prognosewert y_{t+1} entspricht dem Wert y_t für die nächste Vorhersage. Alle weiteren Inputwerte werden ebenfalls um einen Zeitpunkt nach vorne verschoben.

Bei der Anwendung des rekursiven Ansatzes ist zu berücksichtigen, dass durch die Verwendung bereits vorhergesagter Werte ($y_t = y_{t+1}$) für den nächsten Output, die Prognosefehler vergangener Perioden auch die Qualität zukünftiger Werte beeinflussen. Damit wird dieser Ansatz zunehmend empfindlicher gegenüber Prognosefehlern, je länger der benötigte Prognosezeitraum $t_{Prognose}$ ist [Bontempi et al. 2013].

3.3.1.2.2 Direkte Multi-Step-Strategie

Die direkte Multi-Step-Strategie trainiert H voneinander unabhängige Single-Step-Modelle f_h anhand der Formel 3.9 und verbindet die einzelnen Prognosen anschließend zu einer gesamten Vorhersage. H entspricht der Länge des benötigten Prognosehorizontes $t_{Prognose}$.

$$y_{t+h} = f_h(y_t, ..., y_{t-n+1}) \tag{3.9}$$

für
$t = n \dots N - H$
$h = 1 \dots t_{Prognose}$

Im Gegensatz zur rekursiven Multi-Step-Strategie werden in der direkten Multi-Step-Strategie keine bereits prognostizierten Werte verwendet. Dadurch wird das Risiko einer Fehleranhäufung verringert. Allerdings ergeben sich in diesem Ansatz Nachteile bezogen auf die Anzahl der benötigten Modelle. Das Trainieren von H unabhängigen Modellen führt zu einer Zunahme der Rechenzeit in Abhängigkeit der Länge des benötigten Prognosezeitraums, wodurch dieser Ansatz rechenintensiv werden kann. Durch die voneinander unabhängige Vorhersage von H Prognosewerten werden zudem die Eigenschaften der Zeitreihe, wie Trend und Saison, nur schwer erkannt [Bontempi et al. 2013].

3.3.1.2.3 Multiple-Output-Strategie

Sowohl die rekursive als auch die direkte Multi-Step-Strategie zeigen ihre Nachteile vor allem bei der Zunahme der Länge des Prognosezeitraums. Begründet liegen diese Probleme vor allem in der Verwendung von Single-Step-Prognosen zur Vorhersage des gesamten Prognosehorizontes [Bontempi et al. 2013].
Die Multiple-Output-Strategie bietet eine Alternative zu diesen Ansätzen. Anstatt der Vorhersage einzelner Werte wird in dieser Strategie der gesamte Prognosezeitraum in einem Schritt vorhergesagt.

Für die Prognose wird ein Modell F erstellt und die Vorhersage wird anhand der Formel (3.10) berechnet:

$$[y_{t+H}, ..., y_{t+1}] = F(y_t, ..., y_{t-n+1}) \tag{3.10}$$

für
$t = n \dots N - H$
$H = 1 \dots t_{Prognose}$

Der Unterschied zur direkten Multi-Step-Strategie besteht darin, dass im Zuge der Multiple-Output-Strategie ein einzelnes Modell im Trainingsprozess erstellt wird. Diese Vorgehensweise soll dazu beitragen, dass im Gegensatz zu den H-Modellen in der direkten Multi-Step-Strategie Abhängigkeiten innerhalb der betrachteten Zeitreihe besser erkannt werden. Analog zur direkten Multi-Step-Strategie zeigt die Multiple-Ouptput-Strategie die gleichen Vorteile gegenüber der rekursiven Multi-Step-Strategie bezüglich der Fehlerfortpflanzung [Bontempi et al. 2013].
Für eine detaillierte Herleitung der einzelnen Vorgehensweisen wird auf die Beiträge von Bontempi et al. 2013 und Kline 2004 verwiesen.

3.3.2 Prognoseerstellung im Ist-Prozess

Bezogen auf den Ist-Prozess, wird mit Ausnahme in den dafür vorgesehenen univariaten, statistischen Modellen ein multivariater Ansatz zur Erstellung der Prognose verwendet. Basierend auf dem in Abschnitt 2.1.1 beschriebenen Cockpit der Dispositionsplanung stehen folgende vergangene Informationen je Periode zur Verfügung:

- abgesetzte Menge der aktuellen Periode
- abgesetzte Menge der Periode im Vorjahr
- Aktionsmenge in der Periode
- Budgetmenge in der Periode

Zusätzlich werden den verwendeten neuronalen Netzen auch zeitliche Daten in folgender Form zur Verfügung gestellt:

- Monat
- Woche
- Arbeitstage in der jeweiligen Woche

Wird basierend auf der Berechnung der Länge des Prognosehorizonts eine Multi-Step-Strategie für die Prognose benötigt, so wird im Rahmen dieser Arbeit die in Abschnitt 3.3.1.2.3 beschriebene Multiple-Ouput-Strategie herangezogen. Begründet liegt diese Wahl in den Vorteilen dieses Ansatzes bezüglich der Verwendung eines Modells je Produkt und der Vermeidung der Fehlerfortpflanzung im Gegensatz zum rekursiven Ansatz.

Der Gesamtprozess zur Entwicklung einer Prognose lässt sich für die Problemstellung der Arbeit auf vier wesentliche Schritte eingrenzen. Diese sind in Abbildung 3.9 ersichtlich und beinhalten die Hauptpunkte Datenanalyse, Datenaufbereitung, Modellentwicklung und Modellevaluierung. Zwischen den beiden Schritten der Modellentwicklung und der Modellevaluierung besteht hierbei ein wechselseitiger Zusammenhang. Dieser bezieht sich vor allem auf die Entwicklung der neuronalen Netze und deren Hyperparameter-Tuning im Rahmen des Trainingsprozesses.

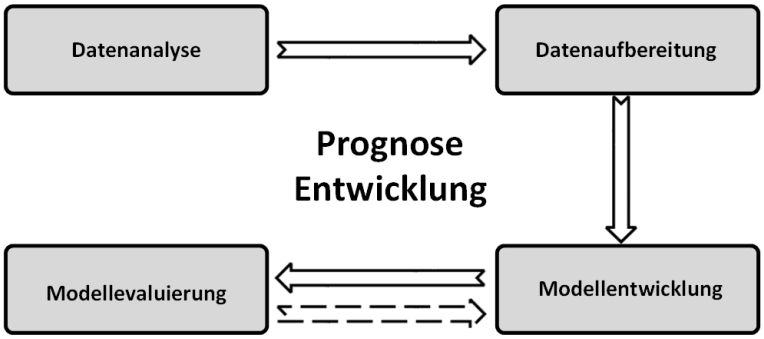


Abbildung 3.9: Überblick Ablauf Forecast

Alle weiteren theoretischen und auf den Ist-Prozess bezogenen Kapitel orientieren sich an diesen vier Schritten zur Erstellung der Prognosen.

4 Datenanalyse

Der Teilprozess der Datenanalyse dient als vorbereitender Schritt im Rahmen der Entwicklung von Prognosen. Dessen Ziel ist es, die für die Prognose relevanten Informationen aus der Gesamtheit der verfügbaren Systemdaten abzuleiten und in strukturierter Form zur Verfügung zu stellen.
Die nachfolgenden Abschnitte zeigen aus diesem Grund die im Rahmen dieser Arbeit verwendeten Verfahren zur Klassifikation von Artikeln. Diese ermöglichen eine Einschränkung bezüglich der für die Prognoseerstellung relevanten Produkte.

4.1 Theoretische Grundlagen der ABC-Analyse

Die ABC-Analyse ist eines der am weitesten verbreiteten Verfahren zur Klassifikation innerhalb von logistischen Problemstellungen. Das Verfahren findet Anwendung z.B. in der Klassifikation von Produkten basierend auf deren Lagerkosten oder Häufigkeit im Verbrauch. Es wird aber auch zur strategischen Auswahl von Kunden oder Lieferanten verwendet [Zsifkovits 2012].
Die Berechnung der ABC-Analyse erfolgt üblicherweise in vier Schritten. Im Hinblick auf die in Abschnitt 4.2 durchgeführte ABC-Analyse für den Ist-Prozess werden diese anhand einer Artikelklassifikation, bezogen auf die Häufigkeit des Verbrauchs, beschrieben.

1. Ermittlung der Anzahl der Kundenbestellungen je Produkt
2. Sortierung der Verbrauchswerte in absteigender Reihenfolge
3. Berechnung des prozentualen Anteils des Verbrauchs je Produkt auf Basis des Gesamtverbrauchs
4. Kumulierung des prozentualen Anteils für jedes Produkt

Die Einteilung in die Klassen A, B oder C kann anschließend auf Basis von Wert- oder Mengengrenzen durchgeführt werden.

Bei einer wertmäßigen Unterteilung, erfolgt die Klassifikation laut [Heiserich et al. 2011] anhand des berechneten kumulierten Anteils K und folgender Bereiche:

- A-Teile: $0 \leq K \leq 0.8$
- B-Teile: $0.8 < K \leq 0.95$
- C-Teil: $0.95 < K \leq 1.0$

Bezogen auf das Beispiel der Kundenbestellungen, ist aufgrund der mengenmäßigen Unterteilung damit folgende Aussage für die A klassifizierten Teile möglich:

- X Prozent der Produkte verursachen 80 Prozent der Bestellungen.

Analog kann für die B und C klassifizierten Produkte vorgegangen werden.

Die mengenmäßige Klassifizierung ermöglicht genau die umgekehrte Aussage. Soll festgestellt werden, welcher Anteil des Gesamtverbrauchs durch einen fixen prozentualen Teil an Produkten (A_P) erfolgt, so werden laut [Zsifkovits 2012] folgende Bereiche empfohlen:

- A-Teile: $0 \leq A_P \leq 0.2$
- B-Teile: $0.2 < A_P \leq 0.4$
- C-Teile: $0.4 < A_P \leq 1.0$

4.2 ABC-Analyse im Ist-Prozess

Bezogen auf den Ist-Prozess wird die ABC-Analyse anhand der Häufigkeit der Kundenaufträge des jeweiligen Produktes bestimmt. Berechnungsgrundlage bilden hierbei die einzelnen Auftragspositionen in den Belegen des SAP-ERP-Systems. Die Bestellmenge der Produkte wird für die ABC-Analyse nicht berücksichtigt.

Abbildung 4.1 zeigt hierzu ein Beispiel eines Kundenauftrags. Anhand der Kombination aus Auftragsnummer (2793 im Beispiel) und Positionsnummer (10) erfolgt die eindeutige Zuordnung zu den bestellten Produkten. Ein Kundenauftrag kann hierbei auch mehrere Positionen mit unterschiedlichen Produkten enthalten. Für die Berechnung der Häufigkeit der Bestellungen je Produkt wird daher die produktspezifische Anzahl an Positionseinträgen summiert.

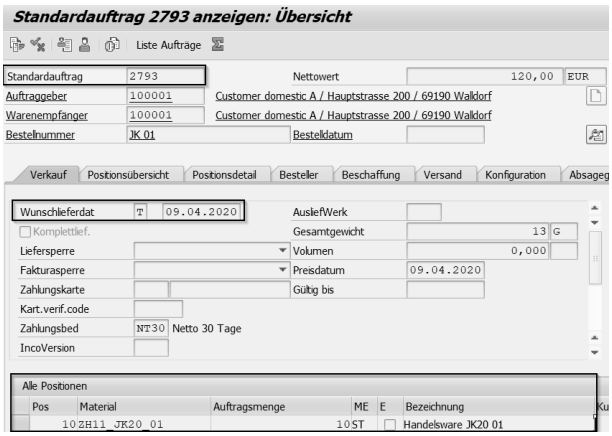


Abbildung 4.1: Beispiel Kundenauftrag

Für den Betrachtungszeitraum der ABC-Analyse wurde die Zeitspanne zwischen 01.01.2019 und 31.12.2021 bezogen auf das Wunschlieferdatum der Kunden im SAP-Beleg herangezogen. Zusätzlich wurden alle Produkte aus der Analyse ausgeschlossen, die zum Stichtag am 01.03.2022, für die Löschung vorgemerkt waren.
Die Zuteilung der Produkte zu den einzelnen Klassen der ABC-Analyse wurde wertmäßig anhand der in Abschnitt 4.1 beschriebenen Grenzen nach [Heiserich et al. 2011] durchgeführt.

Basierend auf diesen Rahmenbedingungen, wurden 1289 Produkte klassifiziert. 237 dieser Produkte und damit knapp 18 Prozent sind für 80 Prozent der Auftragspositionen verantwortlich. Diese Produkte wurden als A-Teile klassifiziert. Weitere 250 Produkte wurden der B-Kategorie zugeordnet. Damit entstanden im Zeitraum zwischen 01.01.2019 und 31.12.2021 durch 487 Produkte 95 Prozent der Auftragspositionen.

Das Ergebnis der ABC-Analyse ist in Abbildung 4.2 ersichtlich.

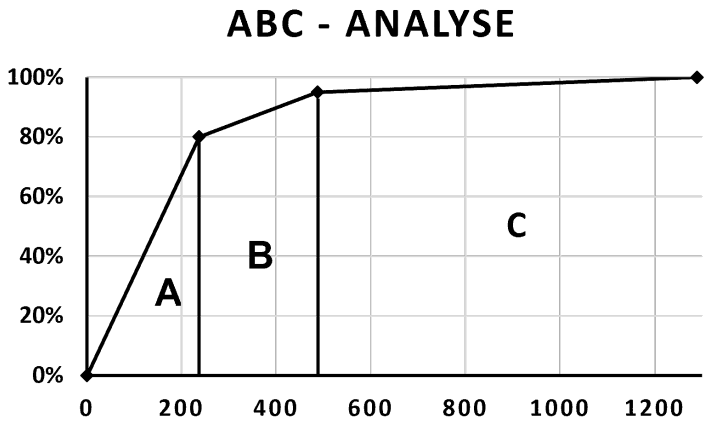


Abbildung 4.2: ABC Analyse Ist-Prozess

4.3 Theoretische Grundlagen XYZ-Analyse

Neben der ABC-Analyse stellt die XYZ-Analyse ein weiteres Instrument der Artikelklassifikation dar. Sie bietet die Möglichkeit der Einteilung der einzelnen Produkten anhand der Verbrauchsschwankungen [Zsifkovits 2012].

Die einzelnen Klassen können laut [Arnold et al. 2008] folgendermaßen interpretiert werden:

- X-Teile: regelmäßiger Verbrauch
- Y-Teile: leicht schwankender oder trendmäßig steigender bzw. fallender Verbrauch
- Z-Teile: unregelmäßiger Verbrauch

Die Zuordnung von Produkten zu den Klassen erfolgt anhand der Ermittlung des Variationskoeffizienten je Produkt. Formel (4.1) zeigt dessen formale Berechnung:

$$v = \frac{\sigma_X}{\overline{X}} \tag{4.1}$$

mit
v .... Variationskoeffizient
σ_X .... empirische Standardabweichung der Menge X
$\overline{X}$ .... arithmetischer Mittelwert der Menge X

Für die Zuordnung zu den Gruppen X, Y, und Z empfiehlt [Zsifkovits 2012] folgende Wertebereiche:

- Klasse X: $0 \leq v \leq 0.3$
- Klasse Y: $0.3 < v \leq 0.7$
- Klasse Z: $v > 0.7$

4.4 XYZ-Analyse im Ist-Prozess

Analog zur ABC-Analyse in Abschnitt 4.2, wurde die XYZ-Analysen im Ist-Prozess auf Basis der Auftragspositionen in den Kundenaufträgen durchgeführt. Die für die Berechnung des Variationskoeffizienten benötigten Verkaufsmengen wurden hierbei je Kalenderwoche summiert. Die Ergebnisse der Analyse beziehen sich somit auf die wöchentlichen Bedarfsschwankungen des jeweiligen Produktes. Der Fokus der XYZ-Analyse liegt auf den in Abschnitt 4.2 klassifizierten A-Produkten.

201 der 237 A-klassifizierten Produkte wurden auf Basis der in Abschnitt 4.3 definierten Wertebereiche, der Kategorie Z zugeordnet. Damit unterliegen knapp 85 Prozent der A-klassifiziereten Teile starken, unregelmäßigen Bedarfsschwankungen bezogen auf die wöchentlichen Kundenbedarfe.

Das gesamte Ergebnis der XYZ-Analyse ist in Abbildung 4.3 ersichtlich.

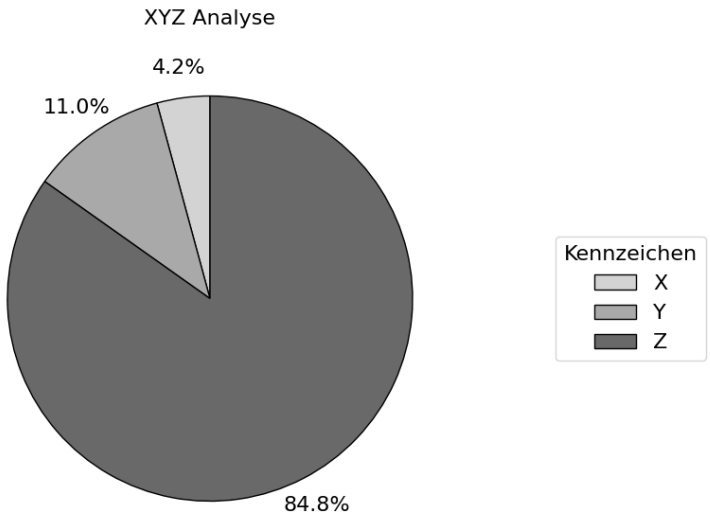


Abbildung 4.3: XYZ Analyse Ist-Prozess

Die im Rahmen dieser Arbeit betrachtete Problemstellung weist somit die gleichen Herausforderungen bezüglich der Volatilität und Prognostizierbarkeit der Kundenbedarfe auf, die auch in den Beiträgen von [Patàk und Vlckova 2014] sowie [Veiga et al. 2010] allgemein für die Lebensmittelindustrie beschrieben wurden. Der Fokus der weiteren Abschnitte liegt daher auf die Anwendung der Prognosen für die A/Z-klassifizierten Produkte und der Erstellung von Modellen zur Unterstützung der operativen Materialbedarfsplanung.

5 Datenaufbereitung

Die Tätigkeiten innerhalb der Datenaufbereitung beinhalten die vorbereitenden Schritte, die zur Sicherstellung der Anwendbarkeit für die in Kapitel 6 beschriebenen Modelle benötigt werden.

5.1 Theoretische Grundlagen Datenaufteilung

In der Anwendung von statistischen Modellen und neuronalen Netzen ist es nötig, eine Teilung der verfügbaren Daten in unterschiedliche Sets durchzuführen. Diese dienen der Sicherstellung einer korrekten Auswertung der Modelle und zur Feststellung deren Performance auf die jeweilige Problemstellung.

Allgemein wird hierzu zwischen drei Sets unterschieden:

- **Trainingsset**
  Das Trainingsset wird zum Erlernen der spezifischen Muster und Zusammenhänge der Problemstellung herangezogen. Die Modelle versuchen hierbei, die zugrunde liegende Zielfunktion der Problemstellung zu minimieren bzw. zu maximieren.
- **Validierungsset**
  Mithilfe des Validierungssets wird die Performance der erstellten Modelle getestet. Durch Parameteranpassungen an den Modellen kann das Validierungsset mehrmalig zur Beurteilung der Performance verwendet werden.
- **Testset**
  Das Testset wird zur Messung der Performance des finalen Modells herangezogen. Im Gegensatz zum Validierungsset wird dieses einmalig verwendet.

Die Verteilung der verfügbaren Daten auf die Sets kann rein prozentual, z.B. im Verhältnis 70/20/10 auf Trainings-, Validierungs- und Testset, oder mithilfe unterschiedlicher Verfahren, z.B. der Kreuzvalidierung, durchgeführt werden.
Da im Rahmen dieser Arbeit eine rein prozentuale Teilung verwendet wird, wird für weitere Verfahren auf die Beiträge von [Xu und Goodacre 2018] sowie [Cerqueira et al. 2020] verwiesen.

5.2 Datenaufteilung im Ist-Prozess

Unabhängig von der Art der Teilung muss in Bezug auf Zeitreihen berücksichtigt werden, dass die zeitliche Ordnung ein wesentliches Kriterium darstellt. Somit ist in Bezug auf die Problemstellung der Arbeit eine strikte Teilung der Daten anhand der Zeit erforderlich. Ein Mischen der Daten, wie es in vielen Machine-Learning-Problemstellungen üblich ist, darf nicht erfolgen.

Wie in Abschnitt 1.1 beschrieben, erfolgt die Teilung im Ist-Prozess daher in zwei Schritten. Einerseits wird eine zeitliche Trennung zwischen den Sets durchgeführt. Somit werden alle verfügbaren Daten ab dem 01.01.2021 ausschließlich für das Testset verwendet. Andererseits erfolgt die zweite Teilung innerhalb aller verfügbaren Daten bis zum Stichtag am 30.12.2020. Diese Daten werden im Verhältnis 50/50 auf das Trainings- und Validierungsset aufgeteilt.

Abbildung 5.1 zeigt die Aufteilung auf die einzelnen Sets.

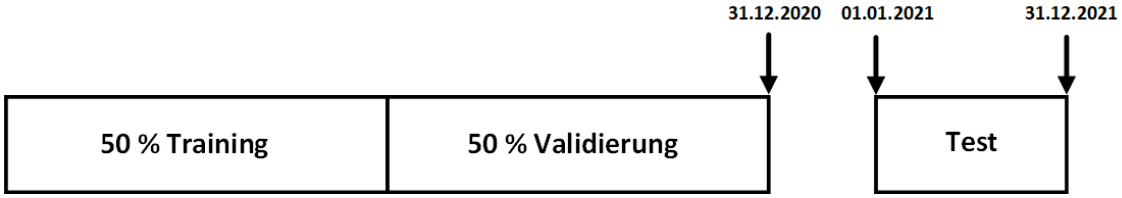


Abbildung 5.1: Teilung Trainings-, Validierungs- und Testset

Innerhalb des Trainings- und des Validierungssets werden die Daten gemäß Abbildung 5.2 geteilt. Als Input (grau) werden immer die letzten 52 Wochendaten herangezogen. Die Output-Länge (schwarz) wird dynamisch je Produkt berechnet. Die Zeitverschiebung L wird hierbei, wie in Abschnitt 3.3 beschrieben, in Abhängigkeit des jeweiligen Dispositionsverfahrens berechnet und entspricht dem Prognosehorizont $t_{Prognose}$.

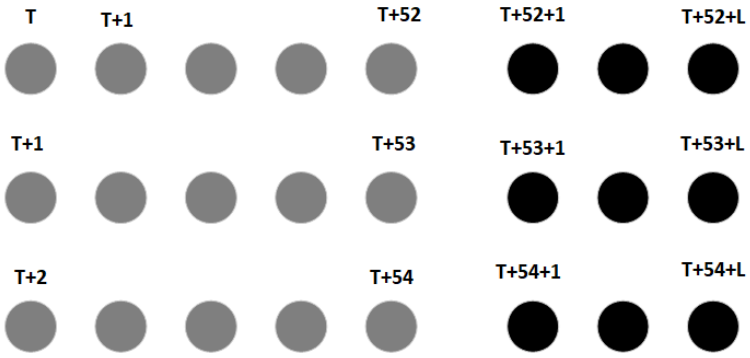


Abbildung 5.2: Teilung Trainings- und Validierungsdaten

5.3 Theoretische Grundlagen der Datenskalierung

Die Skalierung der Input Daten ist einer der relevantesten Vorbereitungsschritte im Hinblick auf die in Abschnitt 6.2 beschriebenen neuronalen Netze. Mit wenigen Ausnahmen ist die Funktionstüchtigkeit der Machine-Learning-Algorithmen eingeschränkt, wenn die eingegebenen numerischen Attribute unterschiedliche Skalierungen aufweisen [Géron 2017]. Im Falle der betrachteten Problemstellung würden diese Effekte vor allem bei multivariaten Prognosen auftreten. Werden z.B. die Kalenderwochen zu den jeweiligen Verkaufsmengen mitgegeben, so bewegen sich diese im Bereich zwischen 1 und 52. Die Verkaufsmenge selbst kann allerdings mehrere Tausend Stück betragen. Typische Vorgehensweisen zur Skalierung der Inputdaten sind die Minimum-Maximum-Skalierung (Min-Max) oder die Standardisierung der Daten.

Die Min-Max-Skalierung berechnet die skalierten Werte anhand der Formel (5.1):

$$x_{Skaliert} = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{5.1}$$

mit
X .... Menge aller zu skalierenden Werte
x .... aktueller Wert der Menge X

Durch die Subtraktion des minimalen Wertes des Datensets X vom jeweiligen Datenpunkt und die anschließende Division mittels der Differenz zwischen Minimum und Maximum werden die Werte in diesem Ansatz in einen fixen Wertebereich zwischen 0 und 1 skaliert.

Die Standardisierung skaliert die einzelnen Werte indem im ersten Schritt der Mittelwert der Daten von den einzelnen Datenpunkten subtrahiert wird. Dieser Schritt der „Mittelwertzentrierung“ hat zur Folge, dass der Mittelwert der transformierten Daten gleich 0 ist. Anschließend wird dieses Ergebnis durch die Standardabweichung der Ausgangsdaten dividiert, womit die Varianz der resultierenden Verteilung gleich 1 ist.

Formal berechnet sich die Standardisierung anhand folgender Formel:

$$x_{Skaliert} = \frac{(x - \overline{X})}{\sigma_X} \tag{5.2}$$

mit
x .... aktueller Wert der Menge X
$\overline{X}$ .... Mittelwert der Menge X
σ_X .... Standardabweichung der Menge X

Im Gegensatz zur Min-Max Skalierung, werden im Zuge der Standardisierung die Werte nicht in einen fixen Bereich zwischen 0 und 1 skaliert. Allerdings ist dieser Ansatz stabiler gegenüber Ausreißern in den betrachteten Daten. In der Min-Max-Skalierung führt ein Ausreißer nach oben dazu, dass die restlichen Werte nahe von 0 skaliert werden. Umgekehrt führt ein Ausreißer nach unten zu einer Skalierung nahe 1 für den restlichen Datensatz.

5.4 Datenskalierung im Ist-Prozess

Basierend auf den Forschungsarbeiten von [Panigrahi und Behera 2013] sowie [Bhanja und Das 2018] wird im Rahmen dieser Arbeit die Min-Max-Skalierung verwendet. Diese zeigt in den genannten Arbeiten sowohl in der Single-Step als auch Multi-Step-Prognose bessere Ergebnisse als die Standardisierung. Weiter Ansätze zur Skalierung werden im Rahmen dieser Arbeit nicht untersucht. Hierzu wird ebenfalls auf die Beiträge von [Panigrahi und Behera 2013] sowie [Bhanja und Das 2018] verwiesen.

5.5 Theoretische Grundlagen der Datenaufbereitung für statistische Modelle

Wird im Zuge der Bestimmung der Eigenschaften der betrachteten Zeitreihe festgestellt, dass Letztere nicht stationär ist, so können die Verfahren der Datentransformation dazu beitragen, diesen Zustand herzustellen. Unterschieden wird hierbei zwischen varianzstabilisierenden und mittelwertstabilisierenden Transformationen. Diese Verfahren können sowohl einzeln als auch zusammen auf die betrachtete Zeitreihe angewandt werden [Nielsen 2019].

5.5.1 Theoretische Grundlagen der varianzstabilisierenden Transformation

Innerhalb der varianzstabilisierenden Verfahren, werden die logarithmische, die Quadratwurzel- und die Box-Cox-Transformation häufig verwendet[Nielsen 2019]. Die transformierten Werte w_t werden anhand folgender Formeln berechnet:

Logarithmische-Transformation:

$$w_t = \log(x_t) \qquad t = 1, 2, \ldots, n \tag{5.3}$$

Quadratwurzel-Transformation:

$$w_t = \sqrt{x_t} \qquad t = 1, 2, \ldots, n \tag{5.4}$$

Die Box-Cox-Transformation stellt einen Ansatz dar, der sowohl die logarithmische als auch die Quadratwurzel-Transformation berücksichtigt. Der transformierte Wert wird hierbei anhand folgender Formel berechnet:

$$w_t = \begin{cases} x_t^{\lambda} & wenn \lambda \neq 0 \\ \ln(x_t) & \text{sonst} \end{cases} \tag{5.5}$$

Somit entspricht ein Wert von λ gleich 0.5 einer Quadratwurzel-Transformation und ein Wert von 0 einer logarithmischen Transformation. Auf die Bestimmung von λ für die Box-Cox-Transformation, wird im Rahmen dieser Arbeit nicht genauer eingegangen. Hierzu wird auf den Beitrag von [Proietti und Lütkepohl 2013] verwiesen.

5.5.2 Theoretische Grundlagen der mittelwertstabilisierenden Transformation

Weist eine Zeitreihe einen Trend auf, so kann dieser mithilfe der einfachen Differenzbildung entfernt werden. Hierzu wird der Wert zum Zeitpunkt $t-1$ vom Wert zum Zeitpunkt t subtrahiert. Die formale Berechnung ist in Formel (5.6) ersichtlich:

$$w_t = x_t - x_{t-1} \qquad t = 2, 3, \ldots, n \tag{5.6}$$

Um die Stationarität einer Zeitreihe herzustellen, kann es aufgrund komplexer Trendstrukturen nötig sein, den Schritt der einfachen Differenzbildung öfter als einmal durchzuführen. Nach [Nielsen 2019] wird von mehr als drei Wiederholungen allerdings abgeraten. Konnte bis zu diesem Zeitpunkt die Stationarität nicht erreicht werden, so wird die Verwendung alternativer Verfahren empfohlen.

5.6 Datenaufbereitung für statistische Modelle im Ist-Prozess

Für die Datentransformation im Ist-Prozess wird bezogen auf die varianzstabilisierenden Verfahren die Box-Cox-Transformation herangezogen. Hierzu wird das Python Paket „scipy.stats.boxcox“ verwendet, das den Parameter λ anhand der Maximierung der Log-Likelihood-Funktion berechnet. Für die mittelwertstabilisierende Transformation wird das in Abschnitt 5.5.2 beschriebene Verfahren der Differenzierung verwendet.

6 Modellentwicklung

Im Rahmen dieses Kapitels werden unterschiedliche Verfahren zur Prognose von Zeitreihen betrachtet. Abschnitt 6.1 zeigt hierzu die im Rahmen dieser Arbeit relevanten statistischen Modelle. Ab Abschnitt 6.2 werden die unterschiedlichen Architekturen von neuronalen Netzen beschrieben. Die für die Problemstellung spezifischen Anpassungen der Modelle werden direkt im Anschluss an den theoretischen Input betrachtet. Falls keine Anpassung benötigt wird werden die anhand der einschlägigen Literatur beschriebenen Modelle auch zur Prognose im Ist-Prozess herangezogen.

6.1 Statistische Modelle

Im Zuge dieses Abschnittes werden unterschiedliche lineare Modelle zur Prognose von Zeitreihen betrachtet. Bezogen auf univariate Zeitreihen werden im Rahmen dieser Arbeit das autoregressive Modell (Abschnitt 6.1.1), das Moving-Average-Modell (Abschnitt 6.1.2) und deren Kombination im Autoregressive-Integrated-Moving-Average-Modell (Abschnitt 6.1.3) beschrieben. Im multivariaten Fall werden nur die in Abschnitt 6.2 betrachteten neuronalen Netze verwendet. Statistische Modelle für multivariate Prognosen, wie z.B. das vektorautoregressive Modell (VAR), werden im Rahmen dieser Arbeit nicht analysiert. Für diese Modelle wird auf die Inhalte von [Kreiss und Neuhaus 2006] oder [Nielsen 2019] verwiesen.

6.1.1 Theoretische Grundlagen der autoregressiven Modelle (AR)

Autoregressive Modelle (AR) basieren auf der Annahme, dass die zukünftigen Werte einer Zeitreihe mithilfe deren Verhalten in der Vergangenheit vorhergesagt werden können. Für die Vorhersage des Wertes zum Zeitpunkt t, wird eine lineare Kombination aus den Werten der Vergangenheit verwendet. Wie viele Werte der Vergangenheit (Lags) für die Prognose berücksichtigt werden, wird über den Wert p des Modells bestimmt. Dieser wird als Ordnung (Order) des AR-Prozesses bezeichnet.

Allgemein kann ein AR(p)-Modell anhand folgender Formel beschrieben werden:

$$y_t = c + \delta_1 * y_{t-1} + \delta_2 * y_{t-2} + \cdots + \delta_p * y_{t-p} + \epsilon_t \tag{6.1}$$

mit
c .... Konstante
δ .... Koeffizienten des AR Modells
ϵ .... Weißes Rauschen (White Noise)

Unter weißem Rauschen wird eine Zeitreihe mit unkorrelierten, reellen Zufallsvariablen ϵ_t mit $E[\epsilon_t] = 0$ und $Var[\epsilon_t] = E[|\epsilon_t|^2] = \sigma_\epsilon^2$ verstanden.

6.1.1.1 Bestimmung der Ordnung für das AR Modell

Die visuelle Abschätzung der Ordnung eines AR(p)-Modells kann mithilfe der in Abschnitt 3.1.2.3 beschriebenen partiellen Autokorrelation durchgeführt werden. Allgemein gilt für die partielle Autokorrelation π in einer AR(p)-Zeitreihe mit n Datenpunkten und zu einer Zeitverschiebung h: $\pi(h) = 0$ für $h > p$. Zusätzlich kann für die Werte der empirischen partiellen Autokorrelation $\hat{\pi}$ gezeigt werden, dass diese für $h > p$ annähernd normalverteilt sind. Deshalb kann für die Ordnung p das größte Lag h gewählt werden, das nicht im Konfidenzintervall $\pm 1.96/\sqrt{n}$ liegt . Das kürzere Intervall $\pm 1.96/\sqrt{n}$ wird verwendet, da für die Bestimmung des Konfidenzintervalls anhand der Formel $\overline{X} \pm z * \frac{\sigma_X}{\sqrt{n}}$ die benötigten Werte der Autokorrelation zur Berechnung des Mittelwerts $\overline{X}$ und der Standardabweichung σ_X in der Regel nicht bekannt sind ($z_{0.95} = 1.96$) [Kreiss und Neuhaus 2006].

Abbildung 6.1 zeigt hierzu ein Beispiel einer AR(2)-Zeitreihe der Länge $n = 200$ und den 95 Prozent Konfidenzintervall $\pm 1.96/\sqrt{n}$.

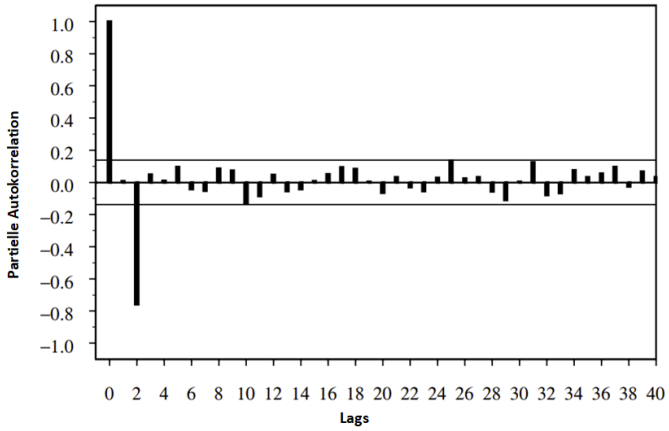


Abbildung 6.1: Partielle Autokorrelation AR(2)-Prozess nach [Kreiss und Neuhaus 2006]

Bezogen auf dieses Beispiel wird für die Ordnung $p = 2$ gewählt.

6.1.1.2 Parameterauswahl für das AR Modell

Für die Anwendbarkeit des AR-Modells wird vorausgesetzt, dass die betrachtete Zeitreihe stationär ist. Damit wird, wie in Abschnitt 3.1.2.1 beschrieben, davon ausgegangen, dass der Mittelwert und die Varianz über die Zeit konstant sind.
Die Restriktion des AR Modells auf stationäre Zeitreihen, führt zu Einschränkungen bezüglich des Wertebereichs für den Parameter δ.

Wird hierzu der einfachste Fall eines AR(1)-Modells, gemäß Formel (6.2) betrachtet, führt die Bedingung der Stationarität zu folgender Einschränkung:

$$y_t = c + \delta_1 * y_{t-1} + \epsilon_t \qquad \text{mit } -1 < \delta_1 < 1 \tag{6.2}$$

Für die mathematische Herleitung dieser Bedingung wird auf den Beitrag von [Nielsen 2019] verwiesen. Allgemein führt eine Zunahme der Ordnung p zu komplexeren Bedingungen für den Parameter δ.

Die Bestimmung der Parameter ist für das AR-Modell nicht eindeutig [Kreiss und Neuhaus 2006]. Bei einer bekannten Ordnung p und unter Berücksichtigung der Einschränkungen werden die Werte für δ geschätzt. Bezogen auf das AR-Modell werden hierzu am häufigsten die Yule-Walker-Gleichungen, die Methode der kleinsten Quadrate und der Maximum-Likelihood-Ansatz verwendet. Für deren Berechnung wird auf den Beitrag von [Kreiss und Neuhaus 2006] verwiesen.

6.1.2 Theoretische Grundlagen der Moving-Average-Modelle (MA)

Anstatt die Werte vergangener Zeitpunkte zur Prognose der Zukunft heranzuziehen, wird im das Moving-Average-Modell (MA) der Fehler (Error) vergangener Prognosen verwendet. Allgemein kann das MA(q)-Modell durch Formel (6.3) beschrieben werden. Der Parameter q stellt hierbei die Ordnung (Order) des Modells dar.

$$y_t = c + \epsilon_t + \theta_1 * \epsilon_{t-1} + \theta_2 * \epsilon_{t-2} + \cdots + \theta_q * \epsilon_{t-q} \tag{6.3}$$

mit
c .... Konstante
θ .... Koeffizienten des MA Modells
ϵ_t .... weißes Rauschen (White Noise)

6.1.2.1 Bestimmung der Ordnung für das MA-Modell

Analog zum AR(p)-Modell kann die Definition der Ordnung für das MA(q)-Modell visuell abgeschätzt werden. Allerdings wird für das MA-Modell die Autokorrelation aus Abschnitt 3.1.2.2 für die Definition der Ordnung herangezogen. Bezogen auf ein MA(q)-Modell gilt für die Autokorrelation $\rho = 0$ für eine Zeitverschiebung $h > q$. Für die Bestimmung der Ordnung p wird damit die größte Zeitverschiebung h gewählt, für welche die empirische Autokorrelation $\hat{\rho}$ nicht im Konfidenzintervall liegt [Kreiss und Neuhaus 2006].

Abbildung 6.2 zeigt eine MA-Zeitreihe der Länge $n = 300$ und mit dem Konfidenzintervall $\pm 1.96/\sqrt{n}$.

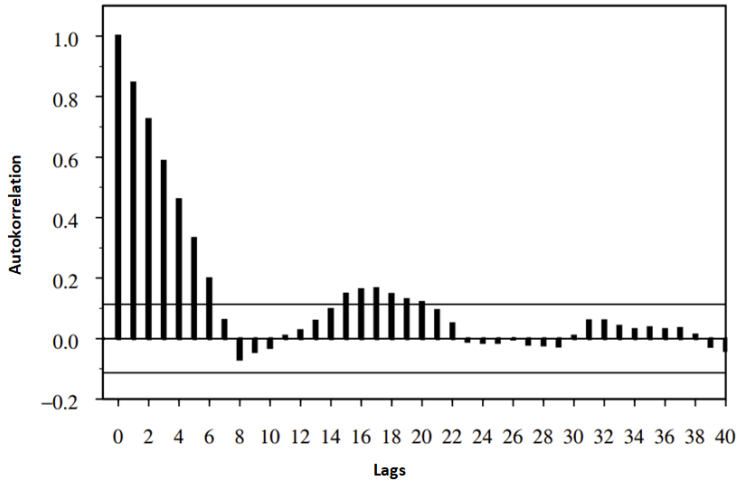


Abbildung 6.2: Autokorrelation MA-Prozess nach [Kreiss und Neuhaus 2006]

Bezogen auf die beschriebene Vorgehensweise würde in diesem Fall $p = 20$ gewählt werden.

6.1.2.2 Parameterauswahl für das MA-Modell

Die Schätzung der Parameter θ für das MA(q)-Modell erfolgt analog zu den in Abschnitt 6.1.1.2 erwähnten Verfahren der Methode der kleinsten Quadrate und des Maximum-Likelihood-Ansatzes.

6.1.3 Theoretische Grundlagen der Autoregressive-Integrated-Moving-Average-Modelle (ARIMA)

Das Autoregressive-Integrated-Moving-Average-Modell (ARIMA) stellt eine Kombination aus den zuvor betrachteten AR- und MA-Modell dar. Der „Integreted “ Teil bezieht sich auf das in Abschnitt 5.5.2 beschriebene Verfahren der Differenzbildung. Die Anzahl der Differenzbildungen für die betrachtete Zeitreihe wird anhand des Parameters d bestimmt.

Dieser wird als Grad der Differenzierung bezeichnet, sodass dem ARIMA-Modell folgende Parameter zur Verfügung stehen:

- p = Ordnung des AR Prozesses
- d = Grad der Differenzierung
- q = Ordnung des MA Prozesses

Formal kann ein ARIMA(p,d,q)-Modell durch Formel (6.4) beschrieben werden:

$$y'_t = c + \delta_1 * y'_{t-1} + \cdots + \delta_p * y'_{t-p} + \theta_1 * \epsilon_{t-1} + \cdots + \theta_q * \epsilon t - q + \epsilon_t \tag{6.4}$$

mit
y'.... mit Ordnung d differenzierte Zeitreihe y
c .... Konstante
δ .... Koeffizienten AR Teil
θ .... Koeffizienten MA Teil
ϵ_t .... weißes Rauschen (White Noise)

6.1.3.1 Bestimmung der Ordnung im ARIMA-Modell

Im Fall einer ARIMA(p,d,q)-Zeitreihe können die einzelnen Ordnungen nicht visuell wie bei den in Abschnitt 6.1.1 und 6.1.2 beschriebenen AR(p)- und MA(q)-Zeitreihen bestimmt werden. Bezogen auf die Autokorrelation ρ ist diese für eine ARIMA-Zeitreihe nicht gleich 0, sondern klingt geometrisch ab.
Alternativ zu visuellen Annäherung der Ordnungen, kann eine Modellauswahl mithilfe des Akaike-Informationskriterium (AIC) durchgeführt werden. Das AIC wird für unterschiedliche Modell mit verschiedenen Ordnungen p und q berechnet. Gewählt wird jenes Modell, welches das AIC minimiert. Die Berechnung des Informationskriterium erfolgt anhand Formel (6.5):

$$AIC(p,q) = -\frac{2}{n} * \ln\left(L_n(\hat{\delta}, \hat{\theta}, \hat{\sigma_\epsilon^2})\right) + 2 * \frac{p+q+1}{n} \tag{6.5}$$

mit
n .... Anzahl der Datenpunkte in der Zeitreihe
p .... Ordnung AR-Teil
q .... Ordnung MA-Teil
δ .... Koeffizienten AR-Teil
θ .... Koeffizienten MA-Teil
ϵ_t .... Weißes Rauschen (White Noise)
$L_n(\hat{\delta}, \hat{\theta}, \hat{\sigma_\epsilon^2})$ .... Maximum-Likelihood Funktion

6.1.3.2 Parameterauswahl im ARIMA Modell

Für die Parameterauswahl im ARIMA(p,d,q)-Modell können dieselben Ansätze wie für die AR(p)- und MA(q)-Modelle herangezogen werden.

6.2 Neuronale Netze

Die in Abschnitt 6.1 beschriebenen statistischen Modelle setzen alle einen linearen Zusammenhang zwischen den einzelnen Werten der betrachteten Zeitreihe voraus. Viele Problemstellungen unterliegen allerdings nicht linearen Zusammenhängen, womit diese Modelle an ihre Grenzen stoßen [Nielsen 2019].
Neuronale Netze ermöglichen sowohl die Modellierung linearer als auch nicht linearer Zusammenhänge. Zusätzlich stellt es für deren Anwendbarkeit keine Voraussetzung dar, dass die betrachteten Zeitreihen stationär sind [Aladag et al. 2009].

Bezogen auf die Prognose von Zeitreihen stellen das Recurrent-Neural-Network (Abschnitt 6.2.4), das Convolutional-Neural-Network (Abschnitt 6.2.6) und deren Kombinationen (Abschnitt 6.2.7), die häufigsten angewandten Verfahren dar. In Abschnitt 6.2.8 wird mit dem Transformer-Modell ein alternativer Ansatz zu diesen Standard-Netzwerken aufgezeigt. Dieser Netzwerktyp wird auch für die Problemstellung dieser Arbeit verwendet und mit den Standard-Architekturen verglichen. Bevor auf die einzelnen Netzwerk-Architekturen genauer eingegangen wird, wird in Abschnitt 6.2.1 ein allgemeiner Überblick zum Aufbau und zur Funktionsweise von neuronalen Netzen gegeben.

6.2.1 Theoretische Grundlagen neuronaler Netze

Nach der Idee der Funktionsweise eines Gehirns, kann ein neuronales Netzwerk als eine Menge von Knoten beschrieben werden, die in unterschiedliche Schichten (Layer) unterteilt werden. Ein Knoten repräsentiert ein Neuron. In Abbildung 6.3 (links) ist ein beispielhafter Aufbau eines neuronalen Netzwerkes mit zwei Layern dargestellt.
Der erste Layer entspricht dem Input-Layer. In Bezug auf die Prognose von Zeitreihen werden diesem Layer die bekannten Werte der Vergangenheit übergeben, die für die Vorhersage benötigt werden. Über den letzten Layer, den Output-Layer, werden der prognostizierte Wert bzw. die prognostizierten Werte zurückgegeben. Dieser Layer kann somit ebenfalls aus einem oder mehreren Neuronen bestehen. Wie in Abbildung 6.3 (rechts) ersichtlich, können zwischen Input- und Output-Layer weitere Schichten in Form von Hidden-Layern hinzugefügt werden. Diese ermöglichen die Erstellung komplexer Architekturen zum Erlernen der Zusammenhänge für die spezifischen Problemstellungen.

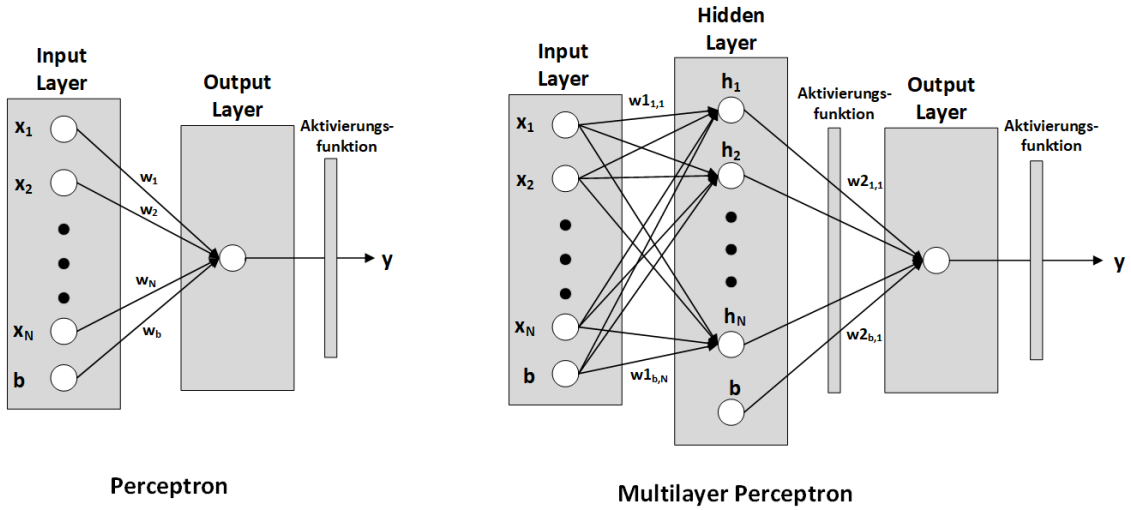


Abbildung 6.3: Beispiel Feed Forward Netzwerk nach [Dhaheri et al. 2017]

Die einzelnen Layer sind durch Kanten miteinander verbunden, wobei jeder Kante ein Gewicht zugeordnet wird. Der Einfluss eines Neurons auf den nachfolgenden Layer wird anschließend über die Größe des Gewichts der Kante und einer für den Layer definierten Aktivierungsfunktion ϕ bestimmt.

Ist jedes Neuron eines Layers mit allen Neuronen des vorherigen Layers verbunden, wird das Netzwerk als *fully connected* bzw. der Layer als *Dense-Layer* bezeichnet. Fließen die Informationen, wie in Abbildung 6.3, zusätzlich nur in eine Richtung (vom Input zum Output) und enthält das Netzwerk keine Zyklen, entsteht ein *Feed-Forward-Netzwerk* .

Der Output j eines Fully-Connected-Layers wird anhand Formel (6.6) berechnet:

$$y_j = \phi(\sum_{i=1}^{n} w_{i,j} * h_i + b) \tag{6.6}$$

h_i .... Werte des Hidden-Layers
$w_{i,j}$ .... Kantengewicht zwischen Hidden-Neuron i und Output-Neuron j
b .... Bias Vector
ϕ .... Aktivierungsfunktion

Die Berechnung der Werte des Hidden-Layer erfolgt auf Basis von Formel (6.7):

$$h_j = \phi(\sum_{i=1}^{n} w_{i,j} * x_i + b) \tag{6.7}$$

mit
x_i .... Inputwerte zum Zeipuntk i
$w_{i,j}$ .... Kantengewicht zwischen Input-Neuron i und Hidden-Neuron j
b_1 .... Bias Vector

In beiden Fällen wird hierbei auf Abbildung 6.3 (rechts) verwiesen.

Typische Aktivierungsfunktionen ϕ stellen beispielsweise die *ReLU-Funktion* oder die *Sigmoid-Funktion* dar. Auf eine genaue Erklärung der unterschiedlichen Aktivierungsfunktionen wird im Rahmen dieser Arbeit nicht eingegangen und auf den Beitrag von [Géron 2017] verwiesen.

6.2.2 Training neuronaler Netze

Der nachfolgende Abschnitt zeigt einen Überblick der theoretischen Grundlagen des Trainingsprozesses eines neuronalen Netzwerkes. Dieser wird hierbei anhand des in Abbildung 6.3 ersichtlichen Perceptrons in einfacher Form erklärt und dient als Übersicht der benötigten Komponenten für die im Rahmen dieser Arbeit angewandten Modelle.
Für komplexere Architekturen wird der *Backpropagation-Algorithmus* verwendet, der in seiner Grundform auf dem Gradientenverfahren zur Anpassung der Kantengewichte basiert. Algorithmen wie beispielsweise der *Adam-Algorithmus* oder der *AdaGrad-Algorithmus* stellen Alternativen zum reinen Gradientenverfahren dar und helfen den Trainingsprozess komplexer Strukturen zu beschleunigen. Für die theoretischen Grundlagen des Backpropagation-Algorithmus sowie die unterschiedlichen Optimierungen, wird auf die Beiträge von [Géron 2017] bzw. [Kingma und Ba 2014] und [Duchi et al. 2011] verwiesen.

Das Training von neuronalen Netzen erfolgt über mehrere Zyklen (Epochen). In jeder Epoche werden anhand einer Menge von Trainingsbeispielen P die Outputwerte y_{pred_p} des Netzwerks mit den tatsächlichen Werten y_{true_p} je Trainingsbeispiel p verglichen und der Fehler (Error, E_p) des jeweiligen Trainingsbeispiels berechnet [Kriesel 2007].

Der gesamte Fehler E zur Anpassung der Kantengewichte ergibt sich durch die Summe der Fehler je Trainingsbspiel E_p. Die Anzahl der Trainingsbeispiele für die Berechnung des gesamten Fehlers wird über den Parameter der Batch-Size definiert. Die Werte der Batch-Size bewegen sich hierbei zwischen 1 und der Anzahl aller Trainingsbeispiele in der Menge P. Innerhalb einer Epoche werden alle Trainingsbeispiele der Menge P betrach-

tet. Werden aufgrund der Batch-Size mehrere Batches definiert, so erfolgt ebenfalls eine mehrmalige Anpassung der Kantengewichte innerhalb einer Epoche. Als Fehler für eine Epoche wird beispielsweise der durchschnittliche Fehler aller Batches angenommen.

Bezogen auf Zeitreihen kann ein einzelnes Trainingsbeispiel p beispielsweise folgendermaßen aufgebaut sein: Als Input i werden die letzten 52 verfügbaren Wochen verwendet. Die Ouputlänge bestimmt sich über den definierten Prognosezeitraum und beträgt beispielsweise drei Wochen. Für ein Trainingsbeispiel p ergibt sich somit die Kombination aus Input i (52 Wochen) und Output y_{pred} (3 Wochen).

Die Berechnung des Fehlers E_p je Trainingsbeispiel erfolgt mithilfe einer für das neuronale Netz definierten Zielfunktion. Typische Zielfunktionen stellen hierzu beispielsweise die mittlere quadratische Abweichung (mean squared error) oder der Huber-Loss dar, für welche die Abweichung zwischen y_{true} und y_{pred} im Zuge des Trainingsprozesses minimiert werden soll. Für eine detaillierte Beschreibung der unterschiedlichen Zielfunktionen wird auf den Beitrag von [Géron 2017] verwiesen.

Die Outputwerte des neuronalen Netzwerks hängen von den Kantengewichten zwischen den einzelnen Layern ab. Damit nimmt sowohl der gesamte Fehler E als auch der Fehler je Trainingsbeispiel E_p zu oder ab, je nachdem wie die Kantengewichte verändert werden. Die Anpassung eines einzelnen Kantengewichts w zur Minimierung des gesamten Fehlers E kann mithilfe der Delta-Regel durchführt werden. Diese ist in den Formeln (6.8) und (6.9) ersichtlich.

$$w_{i,j}^{\text{neu}} = w_{i,j} + \Delta w_{i,j} \tag{6.8}$$

mit

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}} \tag{6.9}$$

$w_{i,j}^{\text{neu}}$ .... neues Kantengewicht zwischen Input-Neuron i und Output-Neuron j
$w_{i,j}$ .... aktuelles Kantengewicht zwischen Input-Neuron i und Output-Neuron j
η .... Lernrate
E .... gesamter Error aufgrund der verwendeten Zielfunktion

Die Lernrate η gibt an wie stark die Gewichtung einer Kante aufgrund des Fehlers angepasst wird. Die Anzahl an Epochen kann für den Trainingsprozess entweder fix vorgegeben werden oder es wird, wie in Abschnitt 6.2.3.3 beschrieben, eine Bedingung definiert, die das Training beendet.

6.2.3 Training von neuronalen Netzen im Rahmen der Arbeit

Wie in Abschnitt 6.2.2 anhand des Perceptrons beschrieben wurde, ist es für die Anwendung neuronaler Netzwerken nötig, eine Zielfunktion, eine Lernrate und gegebenenfalls eine Abbruchbedingung für das Training zu definieren. Die nachfolgenden Abschnitte zeigen hierzu die im Rahmen dieser Arbeit definierten Vorgehensweisen. Als Backpropagation-Algorithmus wird der *Adam-Algorithmus* verwendet. Die Batch-Size beträgt für alle verwendeten Modelle 1.

6.2.3.1 Zielfunktion der angewandten Modelle

Bei der Definition der Zielfunktion der im Rahmen dieser Arbeit verwendeten Modelle, müssen die unterschiedlichen wirtschaftlichen Auswirkungen bei Über- bzw. Unterbestand (Out-of-Stock) berücksichtigt werden. Während die Erzeugung von Überbestand aufgrund von zu hohen Prognosewerten neben den zunehmenden Kapitalbindungskosten auch den Handlingsaufwand innerhalb des Lagers erhöht und das Risiko einer zu langen Lagerdauer bei Lebensmittel birgt, können im Falle des Unterbestandes Opportunitätskosten und Aufwände aufgrund von Sonderproduktionen oder Sonderlieferungen entstehen.

Symmetrische Zielfunktionen wie die mittlere quadratische Abweichung berücksichtigen in der Berechnung des Errors E Abweichungen nach oben und unten in gleicher Form. Zur Analyse der Auswirkungen in Bezug auf den Über- und Unterbestand wird im Rahmen dieser Arbeit daher eine asymmetrische Zielfunktionen verwendet. Diese orientiert sich am Beitrag von [Toth 2015] und ist in Formel (6.10) ersichtlich.

$$L(p,\alpha) = \begin{cases} 2 * \alpha * |E|^p & \text{wenn } E \leq 0 \\ 2 * [\alpha + (1 - 2 * \alpha)] * |E|^p & \text{sonst} \end{cases} \tag{6.10}$$

mit
E .... Error aufgrund der Zielfunktion
α .... Parameter für Über- und Unterbestand
p .... Exponent für Gewichtung des Errors

Abbildung 6.4 zeigt die Funktion für unterschiedliche Parameter p und α.

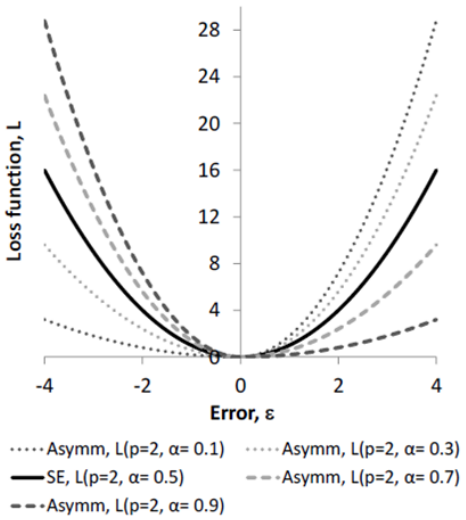


Abbildung 6.4: Beispiel Zielfunktion nach [Toth 2015]

Bei einem Wert von $\alpha < 0.5$ wird eine Überschätzung (Error $E > 0$) stärker gewichtet, bei einem Wert von $\alpha > 0.5$ wird eine Unterschätzung ($E < 0$) stärker berücksichtigt. Der Parameter p ist ein positiver Integer, der es ermöglicht, größere Abweichungen stärker zu berücksichtigen. Werte von $p = 2$ und $\alpha = 0.5$ entsprechen somit der symmetrischen Zielfunktion der quadratischen Abweichung [Toth 2015]. Basierend auf den Ergebnissen des Validierungssets werden im Rahmen dieser Arbeit die Werte $p = 2$ und $\alpha = 0.7$ verwendet.

Formel (6.11) zeigt die formale Berechnung des Errors E im Rahmen dieser Arbeit:

$$E = \sum_{t=0}^{n} y_{pred}(t) - y_{true}(t) * (1 - r) \tag{6.11}$$

mit
E .... Error der Vorhersage
y_{pred} .... Vorhersage des neuronalen Netzes zum Zeitpunkt t
y_{true} .... Wahrer Wert zum Zeitpunkt t
r .... Parameter zur Konfiguration der Lieferfähigkeit bzw. des Sicherheitbestandes
n .... Anzahl der Wochen im Prognoszeitraum

In der Berechnung werden die Abweichungen über den gesamten Prognosezeitraum summiert. Anhand des Parameters r kann der Zielwert y_{true} so konfiguriert werden, dass das neuronale Netz die Kantengewichte aufgrund einer geforderten Lieferfähigkeit ($r > 0$) oder anhand eines dynamischen Sicherheitsbestandes ($r < 0$) anpasst.

Da die Berechnung des dynamischen Sicherheitsbestandes im Ist-Prozess (Abschnitt 2.1.2.5.6)

anhand der Kundenbedarfe der letzten x Tage durchgeführt wird, stellt dieses Vorgehen nur eine näherungsweise Abbildung der realen Berechnung dar.

Die Lieferfähigkeit wird im Rahmen dieser Arbeit durch Formel 6.12 berechnet:

$$L = \frac{n_{oos}}{n_A} * 100 \tag{6.12}$$

mit
L .... Lieferfähigkeit in Prozent
n_{oos} .... Anzahl der Arbeitstage an denen nicht geliefert werden konnte
n_A .... Anzahl der Arbeitstage im Betrachtungszeitraum mit Kundenbedarf

Analog zum dynamischen Sicherheitsbestand bildet auch hier die Anzahl der Tage die Basis in der Berechnung der Lieferfähigkeit. Daher stellt auch die Formel (6.11) nur eine näherungsweise Lösung dar.

Auf Basis der durchgeführten Workshops wurde im Rahmen dieser Arbeit eine Lieferfähigkeit von mindestens 90 Prozent vereinbart, womit ein Wert von $r = 0.10$ für die angewandten Modelle verwendet wird.

6.2.3.2 Lernrate der angewandten Modelle

In Anlehnung an den Beitrag von [Vaswani et al. 2017] und in Hinblick auf das Transformer-Modell (Abschnitt 6.2.10) wird die Lernrate, die dem Adam-Optimizer übergeben wird, durch Formel (6.13) berechnet:

$$l_{rate} = d_{model}^{-0.5} * min(epoche^{-0.5}, epoche * warmup^{-1.5}) \tag{6.13}$$

mit
l_{rate} .... aktuelle Lernrate
d_{model} .... Dimension des Inputs
$epoche$ .... aktuelle Epoche
$warmup$ .... Anzahl der Warmup Epochen

Innerhalb der ersten Epochen (Warmup) wird die Lernrate linear erhöht und anschließend wieder proportional zur inversen Quadratwurzel der aktuellen Epoche reduziert. Der Parameter d_{model} entspricht der Dimension des Inputs. Werden beispielsweise die letzten 52 Wochen als Input übergeben, wird $d_{model} = 52$ angenommen. Basierend auf den Ergebnissen der Validierungsdaten wird für Modelle im Rahmen dieser Arbeit $warmup = 50$ verwendet.

6.2.3.3 Vermeidung von Overfitting

Durch die Vielzahl der verfügbaren Parameter und deren Anpassungsmöglichkeiten stellen neuronale Netze flexible Architekturen zur Lösung komplexer Problem dar. Diese Flexibilität birgt allerdings das Risiko, dass im Zuge des Trainingsprozesses die verwendeten Trainingsdaten auswendig gelernt werden. Die betroffenen Modelle erzielen einen geringen Fehler bei der Vorhersage auf Basis der Trainingsdaten. Werden die Modelle an unbekannten Daten, z.B. am Validierungsset, angewandt, weisen diese allerdings eine schlechte Performance auf. Abbildung 6.5 zeigt hierzu einen beispielhaften Verlauf des Errors der Trainings- und Validierungsdaten.

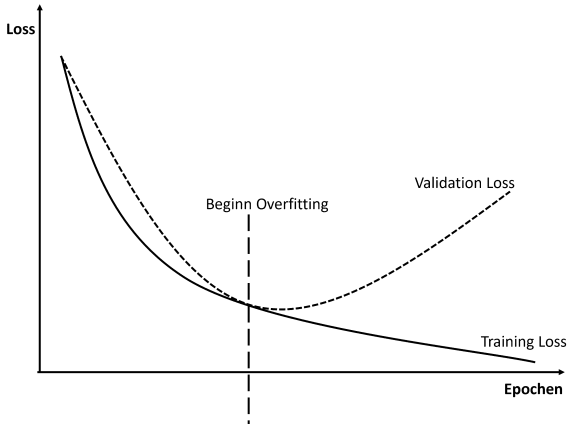


Abbildung 6.5: Beispiel Overfitting nach [Thakur 2020]

Dieser Effekt wird als Overfitting bezeichnet und kann mithilfe von unterschiedlichen Regulierungsverfahren, beispielsweise Dropout-Layern, reduziert werden. Zusätzlich zu den Dropout-Layern wird im Rahmen dieser Arbeit das Training frühzeitig beendet, falls innerhalb von 200 Epochen keine Verbesserung des Validierungsfehlers entsteht. Der Wert von 200 Epochen wurde hierbei aufgrund von Erfahrungswerten innerhalb der Erstellung und Auswertung der unterschiedlichen Modellansätze definiert.

Für die Prognose der zukünftigen Zeitreihenwerte wird jenes Modell verwendet, das innerhalb des Trainingsprozesses den geringsten Validierungsfehler aufweist. Die theoretischen Grundlagen der Dropout-Layer werden im Rahmen dieser Arbeit nicht genauer betrachtet. Hierzu wird auf den Beitrag von [Géron 2017] verwiesen.

6.2.4 Theoretische Grundlagen rekurrente neuronale Netzwerke (RNN)

In Abschnitt 6.2.1 wurde der allgemeine Aufbau eines Feed-Forward-Netzwerkes dargestellt. In dieser Netzwerkarchitektur fließen die Informationen ausschließlich vorwärts, vom Input-Layer zum Output-Layer. Dieses Verhalten führt allgemein dazu, dass reine

Feed-Forward-Netzwerke nur den aktuellen Input einer Epoche betrachten und Inputwerte aus früheren Zeitpunkten nicht berücksichtigt werden. Rekurrente neuronale Netzwerke stellen im Gegensatz Architekturen dar, die den Output früherer Zeitpunkte aufgrund einer Rückkoppelung berücksichtigen. Damit zeigt dieser Netzwerktyp vor allem in der Handhabung von Sequenzen, wie den betrachteten Zeitreihendaten, Vorteile gegenüber reinen Feed-Forward-Architekturen.

Der allgemeine Aufbau eines RNNs ist in Abbildung 6.6 ersichtlich. Im einfachsten Fall (linker Teil der Abbildung), besteht das Netzwerk aus einem Neuron. Dieses erhält im ersten Durchlauf einen Input x und erzeugt einen Output y. In der nächsten Epoche erhält das Neuron wiederum einen Input x und den Output des früheren Schrittes. Bezogen auf einen Zeitpunkt t besteht der Input eines RNN-Neurons im einfachsten Fall somit aus den Werten x_t und y_{t-1}. Abbildung 6.6 (rechts) zeigt dieses Verhalten über mehrere Zeitschritte.

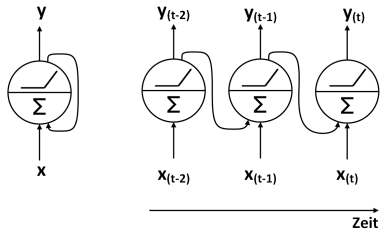


Abbildung 6.6: Aufbau RNN nach [Géron 2017]

Allgemein kann der Ouput Y_t eines RNNs mit Formel (6.14) berechnet werden:

$$Y_t = \phi(X_t * W_x + Y_{t-1} * W_y + b) \tag{6.14}$$

mit
Y_t .... Output Matrix zum Zeitpunkt t
X_t .... Input Matrix zum Zeitpunkt t
W_x .... Gewichtsmatrix der Inputs zum Zeitpunkt t
W_y .... Gewichtsmatrix der Outputs zum Zeitpunkt t-1
b .... Bias Vektor
ϕ .... Aktivierungsfunktion

Für den Traininingsprozess und zur Anpassung der Kantengewichte eines RNNs wird eine adaptierte Form des Backpropagation-Algorithmus verwendet. Diese wird als *Backpropagation-Through-Time (BPTT)* bezeichnet und berücksichtigt die zeitliche Abhängigkeit der einzelnen Neuronen. Für die Beschreibung dieses Algorithmus wird auf den Beitrag von [Géron 2017] verwiesen.

6.2.4.1 Theoretische Grundlagen Input-Output-Sequenzen RNN

Basierend auf der Anzahl der Input- und Outputwerte, gibt es unterschiedliche Möglichkeiten für die Architektur von RNNs. Abbildung 6.7 zeigt vier beispielhafte Vorgehensweisen. Die einzelnen Ansätze sind nicht auf eine Problemstellung begrenzt bzw. können gleiche Problemstellungen mit mehreren Architekturen gelöst werden.

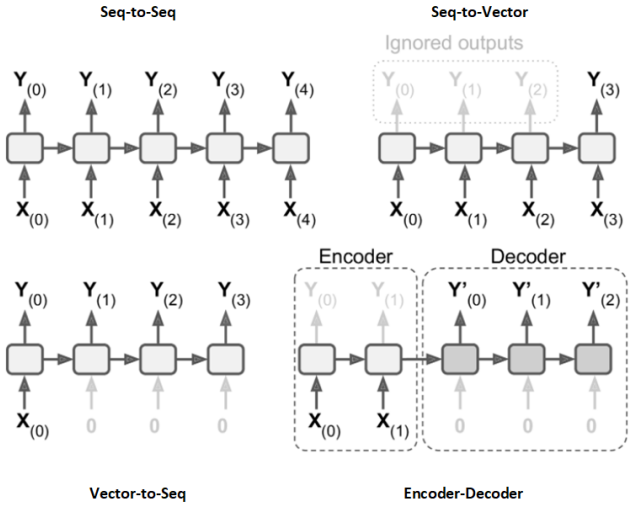


Abbildung 6.7: Input-Output-Sequenzen nach [Géron 2017]

Wird dem neuronalen Netz eine Sequenz von Input-Vektoren x übergeben und erzeugt das neuronale Netz wiederum eine Sequenz von Output-Vektoren y, wird diese Netzwerk-Architektur als *Sequence-to-Sequence-Network (Seq-to-Seq)* bezeichnet. Ein beispielhafter Aufbau eines Seq-to-Seq-Netzwerkes ist in Abbildung 6.7 (oben links) ersichtlich. Typische Anwendungsgebiete dieses Netzwerktyps sind Zeitreihenprognosen oder Übersetzungen.

Wird anstelle einer Sequenz ein einzelner Vektor als Output zurückgegeben und wird weiterhin eine Sequenz als Input übergeben, entsteht ein *Sequence-to-Vector (Seq-to-Vec)* Netzwerk. Diese Architektur ist in Abbildung 6.7 (rechts oben) abgebildet. Analog zur Seq-to-Seq-Architektur wird dieser Netzwerktyp in der Zeitreihenprognose verwendet.

Eine weitere Möglichkeit in der Architektur von RNN-Netzwerken ist die Übergabe eines einzelnen Vektors als Input und die Rückgabe einer Sequenz von Vektoren als Output. Dieses *Vector-to-Sequence (Vec-to-Seq)* Netzwerk wird zum Beispiel in der Bilderkennung verwendet. Der Aufbau eines Vec-to-Seq-Netzwerkes ist in Abbildung 6.7 (links unten) dargestellt.

Der in Abbildung 6.7 (rechts unten) dargestellte *Encoder-Decoder-Aufbau* stellt eine Kombination eines Seq-to-Vec- und eines Vec-to-Seq-Modells dar. Analog zum Seq-to-Seq-Aufbau wird dieser Ansatz zum Beispiel für die Übersetzung von Sätzen verwendet. Das

Transfomer-Modell (Abschnitt 6.2.9) zeigt hierzu eine beispielhafte Anwendung dieser Architektur.

6.2.5 Theoretische Grundlagen des Long-Short-Ter-Memory-Netzwerk (LSTM)

In der in Abschnitt 6.2.4 beschriebenen Implementierung weisen RNN-Netzwerke Probleme in der Handhabung längerer Sequenzen auf. Vor allem bei der Prognose von Zeitreihen kann dies in Bezug auf die Erkennung von saisonalen Schwankungen zu Problemen führen. Im Trainingsprozess entsteht bei der in Abschnitt 6.2.4 beschriebenen Implementierung zudem das Problem instabiler Gradienten im Rahmen der Berechnung der neuen Kantengewichte durch den Backpropagation-Algorithmus.

Long-Short-Term-Memory-Zellen (LSTM) stellen einen Ansatz dar, um diese Probleme zu lösen. Der Aufbau einer LSTM-Zelle ist in Abbildung 6.8 ersichtlich.

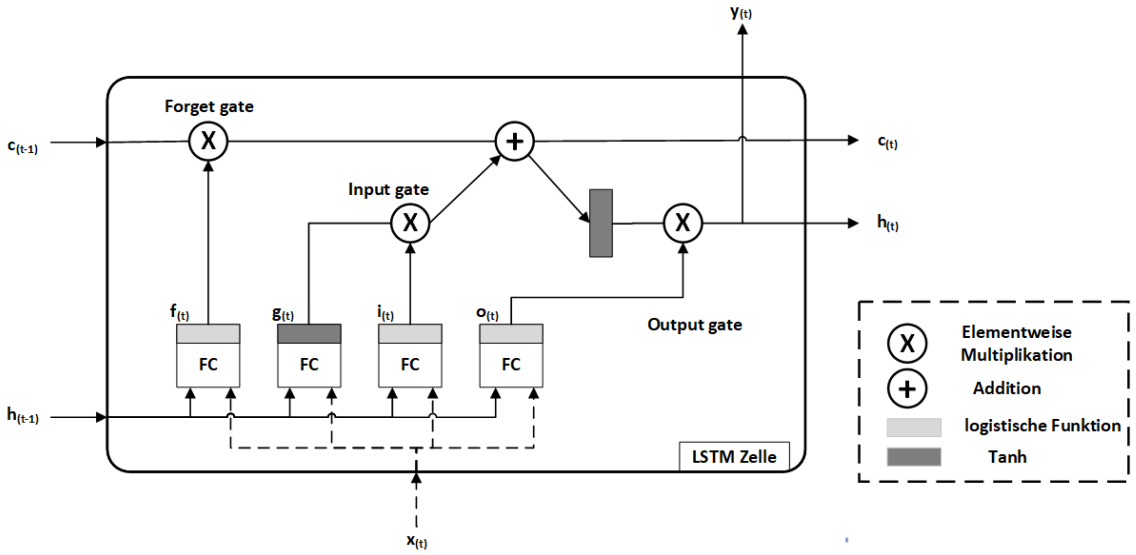


Abbildung 6.8: Aufbau LSTM nach [Géron 2017]

Ziel einer LSTM-Zelle ist es festzustellen, welche bereits gelernten Informationen im Langzeitzustand (long term state) c_t gespeichert werden sollen, welche Informationen verworfen werden können und wie bereits bekannte Informationen der Vergangenheit mit neuen Informationen verknüpft werden sollen. Hierzu verwendet die LSTM-Zelle drei einzelne Komponenten, die als *Gates* bezeichnet werden. Nachfolgend wird das Verhalten eines LSTM anhand Abbildung 6.8 beschrieben.

Formell wird der Langzeitzustand c_t mit Formel (6.15) berechnet:

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \tag{6.15}$$

mit
c_t .... Langzeitzustand zum Zeitpunkt t
f_t .... Forget-Gate zum Zeitpunkt t
c_{t-1} .... Langzeitzustand zum Zeitpunkt t-1
i_t .... Kontroll-Vektor Input-Gate
g_t .... Analyse-Vektor Input-Gate

Das Forget-Gate (f_t) bestimmt in Abhängigkeit des aktuellen Inputs x_t und des Kurzzeitzustandes des vorhergehenden Zeitpunktes h_{t-1}, welche Informationen im Langzeitzustand c_t verworfen werden sollen. Das Forget-Gate wird durch Formel (6.16) berechnet:

$$f_t = \phi(W_{xf}^T * x_t + W_{hf}^T * h_{t-1} + b_f) \tag{6.16}$$

mit
f_t .... Forget-Gate zum Zeitpunkt t
x_t .... Input zum Zeitpunkt t
h_{t-1} ... Kurzzeitzustand zum Zeitpunkt t-1
W_{xf} .... Gewichts-Matrix für den Input Vektor x_t
W_{hf} .... Gewichts-Matrix für den Kurzzeitzustand h_{t-1}
b_f .... Bias Vektor des Forget-Gates

Als Aktivierungsfunktion ϕ wird die logistische Funktion verwendet, deren Outputwerte sich im Bereich zwischen 0 und 1 bewegen. Werte von 0 im Forget-Gate f_t führen durch die elementweise Multiplikation (Hadamard-Produkt) des Gates mit dem Langzeitzustand des früheren Zeitpunktes c_{t-1} dazu, dass diese Einträge in der Berechnung des Outputs nicht berücksichtigt (vergessen) werden.

Über das Input-Gate wird anhand der aktuellen Inputwerte x_t und des Kurzzeitzustandes des vorherigen Zeitpunktes h_{t-1} entschieden, welche neuen Informationen zum Langzeitzustand c_t hinzugefügt werden sollen. Hierzu werden die beiden Vektoren i_t und g_t durch Formel (6.17) und Formel (6.18) berechnet:

$$i_t = \phi(W_{xi}^T * x_t + W_{hi}^T * h_{t-1} + b_i) \tag{6.17}$$

mit
i_t .... Kontroll-Vektor Input-Gate zum Zeitpunkt t
x_t .... Input zum Zeitpunkt t
h_{t-1} ... Kurzzeitzustand zum Zeitpunkt t-1
W_{xi} .... Gewichts-Matrix für den Input-Vektor x_t
W_{hi} .... Gewichts-Matrix für den Kurzzeitzustand h_{t-1}
b_i .... Bias-Vektor des Inpu-Gates

$$g_t = tanh(W_{xg}^T * x_t + W_{hg}^T * h_{t-1} + b_g) \tag{6.18}$$

mit
g_t .... Analyse-Vektor Input-Gate zum Zeitpunkt t
x_t .... Input zum Zeitpunkt t
h_{t-1} ... Kurzzeitzustand zum Zeitpunkt t-1
W_{xg} .... Gewichts-Matrix für den Input-Vektor x_t
W_{hg} .... Gewichts-Matrix für den Kurzzeitzustand h_{t-1}
b_g .... Bias-Vektor des Input-Gates

Der Vektor g_t hat die Aufgabe den aktuellen Input x_t und den Kurzzeitzustand des vorherigen Zeitpunktes h_{t-1} zu analysieren. Als Aktivierungsfunktion wird hierzu in der Grundform des LSTM eine Tangens-Hyperbolicus-Funktion verwendet. Analog zum Forget-Gate f_t wird über den Vektor i_t und der elementweisen Multiplikation mit g_t gesteuert, welche Werte von g_t zum Langzeitzustand c_t addiert werden.

Über das Output-Gate o_t wird abschließend bestimmt, welche Werte des Langzeitzustandes c_t als Output y_t und als neuer Kurzzeitzustand h_t zurückgegeben werden. Das Output-Gate verwendet hierzu die gleiche Logik wie der Vektor i_t des Input-Gates und des Forget-Gates f_t, mit der logistischen Funktion als Aktivierungsfunktion. Formal wird das Output-Gate anhand Formel (6.19) berechnet:

$$o(t) = \phi(W_{xo}^T * x(t) + W_{ho}^T * h(t-1) + b_o) \tag{6.19}$$

mit
o_t .... Output-Gate zum Zeitpunkt t
x_t .... Input zum Zeitpunkt t
h_{t-1} ... Kurzzeitzustand zum Zeitpunkt t-1
W_{xo} .... Gewichts-Matrix für den Input-Vektor x_t

W_{ho} .... Gewichts-Matrix für den Kurzzeitzustand h_{t-1}
b_o .... Bias-Vektor des Output-Gates

Der Output y_t und der neue Kurzzeitzustand h_t berechnen sich abschließend durch Formel (6.20):

$$y(t) = h(t) = o(t) \otimes tanh(c(t)) \tag{6.20}$$

mit
y_t .... Ouput zum Zeitpunkt t
h_t ... Kurzzeitzustand zum Zeitpunkt t
c_t .... Langzeitzustand zum Zeitpunkt t
o_t .... Ouptut-Gate zum Zeitpunkt t

In Bezug auf die Handhabung langer Sequenzen zeigen LSTM-Zellen einen klaren Vorteil gegenüber der Standard-RNN-Architektur. Je länger die übergebenen Sequenzen werden, umso schwieriger ist es allerdings auch für eine LSTM Zelle Langzeitzusammenhänge zu erkennen. Aus diesem Grund werden in der praktischen Prognose von Zeitreihen Ansätze verwendet, welche die betrachtete Sequenz verkürzen, bevor diese als Input in die LSTM-Zelle übergeben werden [Géron 2017].

Eine Möglichkeit die Input-Sequenz zu verkürzen, ist eine kombinierte Architektur eines LSTM- und eines Convolutional (CNN)- bzw. Pooling-Layers. Die Funktionsweise des Convolutional- bzw. Pooling-Layers werden in Abschnitt 6.2.6 beschrieben.

6.2.6 Theoretische Grundlagen des Convolutional Neural Networks (CNN)

Die nachfolgenden Abschnitte zeigen die im Rahmen dieser Arbeit relevanten Inhalte des Convolutional Layers und des Pooling-Layers, welche in Convolutional-Neural-Network-Architekturen für Zeitreihen typischerweise verwendet werden. Für alternative Anwendungen wie beispielsweise der Bilderkennung wird auf den Beitrag von [Géron 2017] verwiesen.

6.2.6.1 Theoretische Grundlagen des Convolutional Layers

Mithilfe des Convolutional-Layers wird versucht, die für die Problemstellung relevanten Informationen aus den Inputdaten zu extrahieren. Wie in Abbildung 6.9 ersichtlich ist, wird hierzu eine Filter-Matrix (Kernel) über die Inputdaten bewegt.
Für Sequenzen wie Zeitreihendaten wird der Filter in eine Richtung (1D CNN) und entlang der Zeit verschoben. Die Spaltenanzahl (Features) des Filters entspricht hierbei der

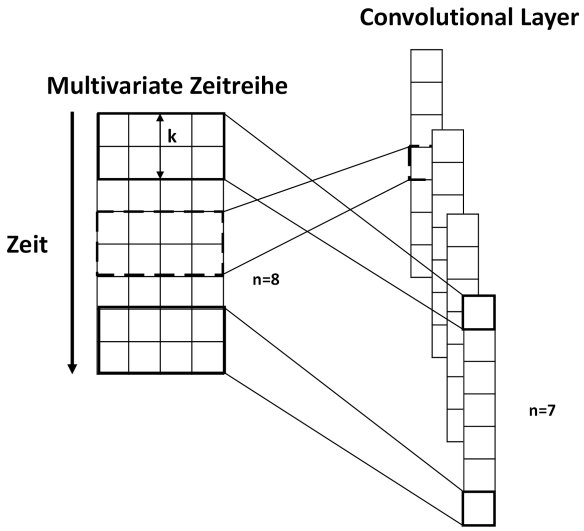


Abbildung 6.9: Beispiel 1D CNN

Spaltenanzahl der Inputdaten. Über den Parameter der *Kernel-Size* kann die Zeilenanzahl des Filters bestimmt werden. Bezogen auf Zeitreihendaten ermöglicht dieser Parameter die Angabe der Anzahl an Zeiteinheiten, die pro Berechnung zu berücksichtigten sind.

Die Schrittweite des Filters wird über den *Stride-Parameter* bestimmt. Bei einem Wert von 1 wird jede Woche des Inputs in der Berechnung mindestens einmal berücksichtigt.

Mithilfe des *Padding-Parameter* wird zu Beginn der Berechnung des CNNs die Startposition des Filters für die Bewegung über die Inputdaten festgelegt. Abbildung 6.9 zeigt hierzu das *Valid-Padding* (kein Padding) mit Kernel-Size $k = 2$. Wie in Abbildung 6.9 ersichtlich ist, bewegt sich der Filter des CNN im Falle des Valid-Padding immer innerhalb der Inputdaten.

Die formale Berechnung eines Wertes des Convolutional-Layers für das Valid-Padding zeigt Formel (6.21):

$$c_t = \phi(\sum_{t=0}^{k-1} x_t * h_t^T) \tag{6.21}$$

mit
c_t .... Wert Convolutional-Layer zum Zeitpunkt t
$x(i)$ .... i-te Zeile der Input-Matrix
$h(i)$ .... i-te Zeile der Kernel-Matrix
k .... Kernel-Size
ϕ .... Aktivierungsfunktion

Zu beachten ist, dass im Falle des Valid-Paddings der Output in Abhängigkeit der Größe der Kerne-Size und des Stride-Parameters verkürzt wird. Die Berechnung der Outputlänge ist in Formel (6.22) ersichtlich:

$$n_{Output} = \frac{n_{Input} - k}{s} + 1 \tag{6.22}$$

mit
n_{Output} .... Anzahl der Zeilen des Convolutional-Layer
n_{Input} .... Anzahl der Zeilen des Input-Layer
k .... Kernel-Size
s .... Stride

Für die Berechnung des Outputwertes für den Zeitpunkt t werden bei der Verwendung des Valid-Paddings zusätzlich Inputwerte des Zeitpunkts größer t berücksichtigt. Bei der Berechnung von Zeitreihen würde dieses Vorgehen einem Blick in die Zukunft entsprechen.

Abbildung 6.10 zeigt mit dem *Causal-Padding* eine alternative Vorgehensweise für die Bewegung des Filters über die Inputdaten.

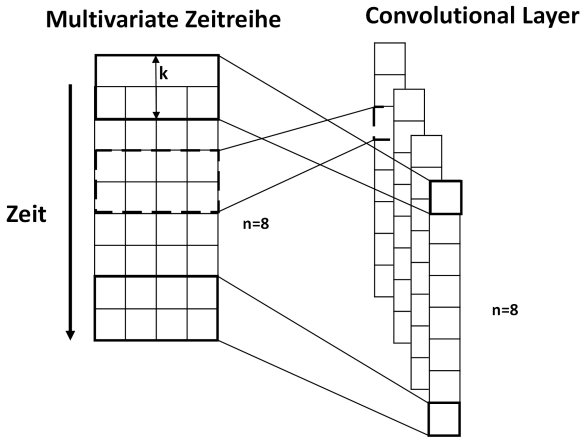


Abbildung 6.10: Causal Padding

Für das Causal-Padding wird der Filter in jener Form über die Inputdaten verschoben, dass die letzte Zeile des Filters dem aktuellen Zeitpunkt entspricht, für den die Berechnung des Convolutional-Layers durchgeführt wird. Dieses Vorgehen führt dazu, dass zu Beginn für einen Teil der Zeilen des Filters keine Zeitreihendaten vorhanden sind, wodurch diese Werte gleich 0 angenommen werden.

Die formale Berechnung des Causal-Paddings ist in Formel (6.23) ersichtlich:

$$c_t = \phi(\sum_{i=0}^{k-1} x_{t-i} * h_i^T)$$
$$x_{t-i} = \begin{cases} 0 & \text{wenn } t-i < 0 \\ x_{t-i} & \text{sonst} \end{cases} \tag{6.23}$$

c_t .... Wert Convolutional-Layer zum Zeitpunkt t
$x(i)$ .... i-te Zeile der Input-Matrix
$h(i)$ .... i-te Zeile der Kernel-Matrix
k .... Kernel-Size
ϕ .... Aktivierungsfunktion

6.2.6.2 Theoretische Grundlagen Pooling Layer

Die Reduzierung der Anzahl der betrachteten Daten wird mithilfe eines Pooling-Layers realisiert. Hierbei wird zwischen lokalem und globalem Pooling unterschieden.

Für das lokale Pooling wird über den Parameter der *Pool Size* festgelegt, um welchen Faktor die Inputdaten des Pooling-Layers verkürzt werden. Die Länge des Outputs berechnet sich hierbei anhand Formel (6.24):

$$n_{Output} = (n_{Input} - p + 1) \tag{6.24}$$

mit
n_{Output} .... Länge des Outputs
n_{Input} .... Länge des Inputs
p .... Pool Size

Die Richtung des Poolings kann über den Parameter *Data-Format* festgelegt werden. Mithilfe des Data-Format-Parameters *Channels-First* wird die Anzahl der Spalten der übergebenen Inputdaten reduziert. Der Wert *Channels-Last* reduziert hingegen die Anzahl der Zeilen der Input-Matrix. Abbildung 6.11 zeigt das Verhalten der unterschiedlichen Data-Format Varianten bei einer Pool-Size von 3.

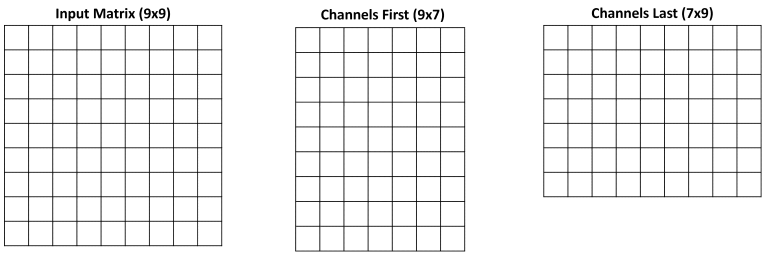


Abbildung 6.11: Beispiel Pooling

Für die Auswahl der Werte, die weiterhin verwendet werden sollen, werden im einfachsten Fall zwei unterschiedliche Ansätze verwendet. Mithilfe des Max-Poolings wird der maximale Wert in den Output übernommen. Das Average-Pooling berücksichtigt den Mittelwert. Abbildung 6.12 zeigt hierzu ein Beispiel.

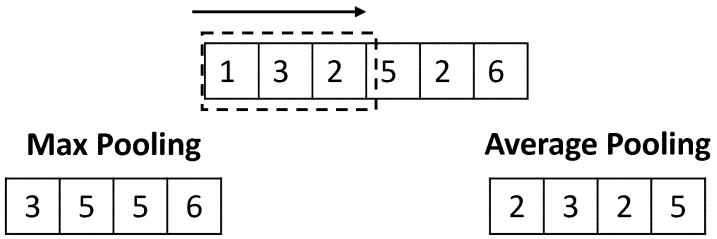


Abbildung 6.12: Beispiel Max- und Average-Pooling

Wird das globale Pooling verwendet, wird ein maximaler (Max-Pooling) oder durchschnittlicher Wert (Average-Pooling) je Zeile oder Spalte berechnet. Die Wahl der Zeile oder Spalte erfolgt analog über den Data-Format-Parameter. Bezogen auf das Beispiel in Abbildung 6.11 würde im Falle des Data-Formats *Channels-First* ein 9×1 Vektor und im Falle des Data-Formats *Channels-Last* ein 1×9 Vektor entstehen.

6.2.7 Kombinierte Architekturen im Rahmen der Arbeit

Für die Prognose des Primärbedarfs wird im Rahmen dieser Arbeit eine kombinierte Architektur von CNN und LSTM verwendet. Das verwendete Netzwerks ist in Abbildung 6.13 qualitativ dargestellt. Für eine übersichtliche Darstellung wurden die Dimensionen der einzelnen Vektoren und Matrizen gegenüber den realen Modellen reduziert.

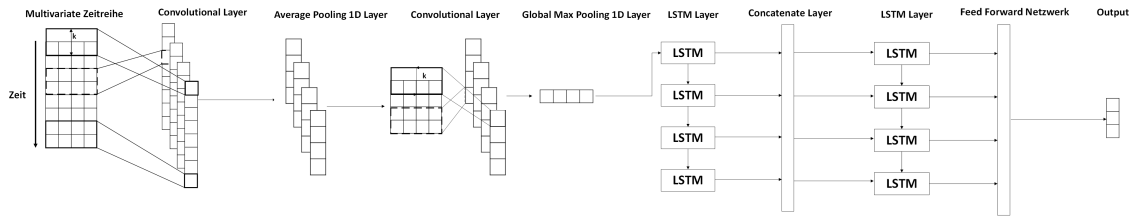


Abbildung 6.13: Beispiel CNN-LSTM

Basierend auf dem in Abschnitt 2.1.1 beschriebenen Cockpit der Dispositionsplanung stehen folgende Informationen je Woche zur Verfügung, wobei dem Modell die letzten 52 Wochen als Input übergeben werden:

| Jahr | Monat | Woche | Arbeitstage | Abgesetzte Menge | Budget Menge | Menge des Vorjahres | Aktionsmenge |

Mithilfe von zwei CNN-Layern werden diese Inputdaten bezüglich der relevanten Informationen und Zusammenhänge untersucht. Zwischen den beiden CNN-Layern befindet sich ein Average-Pooling-Layer, der die Dimension der Matrix und damit die Anzahl der Zeitpunkte reduziert. Hyperparameter wie Kernel-Size und Pooling-Size werden für jedes betrachtete Produkt separat und anhand der Ergebnisse am Trainings- und Validierungsset definiert.

Über einen Global-Max-Pooling-Layer werden die Informationen der CNN-Layer an ein LSTM übergeben. Die Dimension des Outputs des LSTM-Layers entspricht der Größe des Prognosehorizonts des jeweiligen Produktes.

Begründet liegt die Definition der Outputgröße in der Funktion des nachfolgenden Concatenate-Layers. Über diesen werden die bekannten Werte für den Prognosehorizont, wie die Budgetmenge oder die abgesetzte Menge des Vorjahres, mit den Outputwerten des ersten LSTM verbunden. Der Input des zweiten LSTM-Layers enthält somit folgende Werte für jeden Zeitpunkt des Prognosehorizonts:

Wert LSTM 1| Budget Menge | Menge des Vorjahres | Aktionsmenge |

Die Prognose der Absatzmenge wird im Anschluss an das zweite LSTM anhand eines Feed-Forward-Layers (Time Distributed) mit einer Sigmoid-Funktion als Aktivierungsfunktion durchgeführt.

6.2.7.1 Parameter des CNN-LSTM Modells

Tabelle 6.1 zeigt einen Überblick der wichtigsten verwendeten Hyperparameter im CNN-LSTM-Modell. Dropout-Layer wurden in der Tabelle nicht berücksichtigt.

Hyperparameter	**Wert**
Input 1	
Input-Size	52 × 8
CNN 1	
Anzahl Filter	256
Aktivierungsfunktion	ReLU
Kernel-Size	Länge Prognosehorizont * 2
Padding	Causal
Max-Pooling-Layer	
Pool-Size	2
Data-Format	Channels-Last
CNN 2	
Anzahl Filter	256
Aktivierungsfunktion	ReLU
Kernel-Size	Länge Prognosehorizont
Padding	Causal
Global-Average-Pooling-Layer	
Data-Format	Channels-Last
LSTM 1	
Anzahl Neuronen	Länge Prognosehorizont
Aktivierungsfunktion	ReLU
Input 2	
Input-Size	Länge Prognosehorizont × 3
LSTM 2	
Anzahl Neuronen	Länge Prognosehorizont
Aktivierungsfunktion	ReLU
Dense-Layer	
Anzahl Neuronen	1 (Time Distributed)
Aktivierungsfunktion	Sigmoid

Tabelle 6.1: Hyperparameter CNN-LSTM-Modell

6.2.8 Alternative Modellarchitekturen zur Erstellung von Prognosen

Analog zu den in den Abschnitten 6.2.4 bis 6.2.6 beschriebenen Standard-Architekturen zur Prognose von Zeitreihen basieren auch Standard-Ansätze für die Problemstellung zur maschinellen Verarbeitung und Interpretation von Sprachen (Natural-Language-Processing, NLP) auf unterschiedlichen Kombinationen von RNNs und CNNs, meistens in einer Encoder-Decoder-Anordnung [Géron 2017].

Im Beitrag von [Vaswani et al. 2017] wurde ein alternativer Ansatz zur Lösung der NLP-Probleme vorgestellt. Die Transformer-Architektur, welche in Abbildung 6.14 ersichtlich ist.

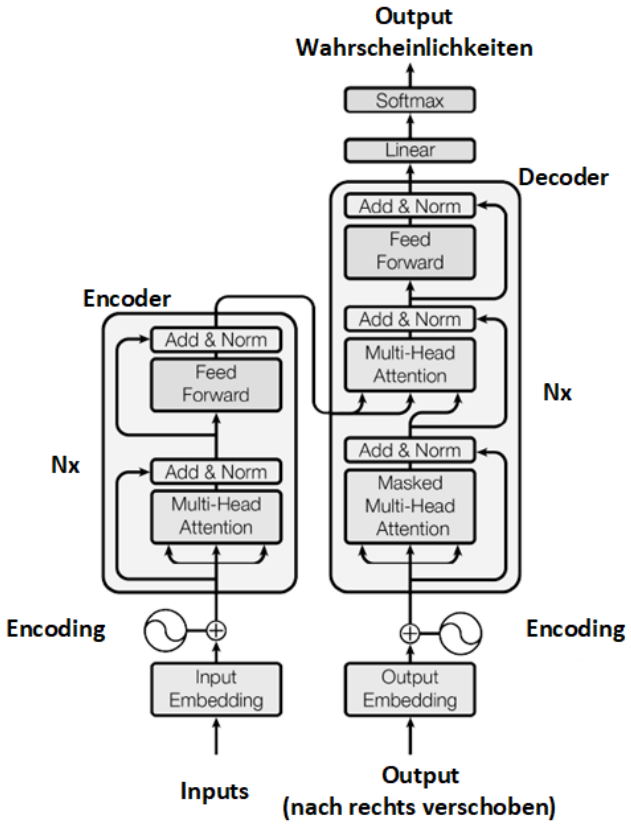


Abbildung 6.14: Architektur Transformer nach [Vaswani et al. 2017]

Bezogen auf die NLP-Problemstellung zeigt dieser Ansatz sowohl Verbesserungen in der Qualität der Ergebnisse als auch im Hinblick auf die Rechenzeit und Parallelisierbarkeit der Modelle. [Vaswani et al. 2017]

Erste Beiträge zur Prognose von Zeitreihen mithilfe von Transformern wurden beispielsweise in [Li et al. 2019] oder [Lim et al. 2021] veröffentlicht. Beide Forschungsarbeiten zei-

gen, dass die Ergebnisse der Standard-CNN- und RNN-Architekturen auch mithilfe von Transformern erreicht werden können. Abschnitt 6.2.10 beinhaltet die Beschreibung eines adaptierten Aufbaus eines Transformers für die spezifische Problemstellung der Lebensmittelindustrie. Die theoretischen Hintergründe in der Architektur werden in Abschnitt 6.2.9 und anhand der Abbildung 6.14 behandelt.

6.2.9 Theoretische Grundlagen des Transformer-Modells

Wie in Abbildung 6.14 ersichtlich ist, besteht die Standard-Architektur des Transformers aus einem Encoder (links) und einem Decoder (rechts).
Sowohl der Encoder als auch der Decoder sind aus einzelnen Modulen aufgebaut, die mehrfach angewandt werden können. Die Anzahl der Module im Encoder und Decoder wird hierbei durch den Parameter Nx bestimmt. Im Beitrag von [Vaswani et al. 2017] wird für beide Teile $N = 6$ angenommen.

Das Transformer-Modell enthält im Gegensatz zu RNNs und CNNs keine Informationen über die zueinander relative bzw. absolute Position eines Inputs in der gesamten Sequenz. Sowohl für NLP-Problemstellungen als auch für die Prognose von Zeitreihen stellen die Position eines Wortes in einem Satz bzw. der Zeitpunkt eines Wertes und deren Abstand zueinander ein entscheidendes Kriterium dar. Über das Positional-Encoding wird aus diesem Grund neben dem Input des Encoders auch der Input des Decoders um diese Informationen erweitert. Für NLP-Problemstellungen wird im Rahmen dieser Arbeit nicht genauer auf die Vorgehensweise des Positional-Encodings eingegangen. Hierzu wird auf die Beiträge von [Vaswani et al. 2017] bzw. [Géron 2017] verwiesen. Die Anwendung für die Zeitreihen im Rahmen dieser Arbeit ist in Abschnitt 6.2.10 beschrieben.

Im Falle des Encoders besteht jedes der Module aus zwei Sub-Layern. Der Input des Encoders fließt zuerst in einen Multi-Head-Attention-Layer. Allgemein formuliert hilft dieser Layer, die Bedeutung der Zusammenhänge zwischen den einzelnen Wörtern der Input-Sequenz zu erkennen. Bezogen auf das Beispiel in Abbildung 6.15 und dem Satz „They welcomed the Queen of the United Kingdom“ würde damit erwartet werden, dass in Bezug auf das Wort „Queen“ der Encoder seine Aufmerksamkeit auf die Wörter „United“ und „Kingdom“ legt und nicht auf die Wörter „They“ oder „welcomed“ [Géron 2017].

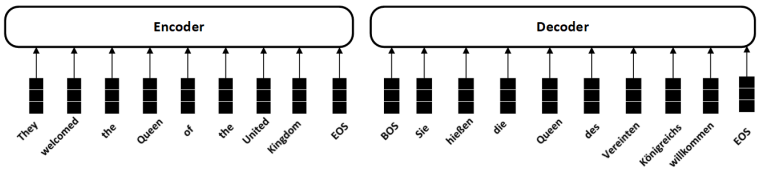


Abbildung 6.15: Encoder, Decoder Input Transformer

Diese Art der Multi-Head-Attention wird als Self-Attention bezeichnet. Die formale Funktionsweise der Self-Attention und der Multi-Head-Attention wird in Abschnitt 6.2.9.1 beschrieben.

Den zweiten Teil des Encoders bildet ein Feed-Forward-Netzwerk, mit dessen Hilfe eine lineare Transformation für jede Position des Outputs der Multi-Head-Attention durchgeführt wird. Der Aufbau des Feed-Forward-Netzwerkes wird in Abschnitt 6.2.9.2 genauer betrachtet.

Sowohl am Ende des Multi-Head-Attention-Layers als auch am Ende des Feed-Forward-Netzwerkes wird der Output des jeweiligen Layers mit dem dazugehörigen Input addiert und anschließend normalisiert. Dieser Schritt wird als Residual-Dropout bezeichnet. [He et al. 2016] zeigen in ihrem Beitrag, dass dieser Ansatz vor allem in komplexen Architekturen zu einer Verbesserung der Trainingsergebnisse der Netzwerke führt. Für vertiefende Informationen wird ebenfalls auf den Beitrag von [He et al. 2016] verwiesen. Die theoretischen Hintergründe zur Layer-Normalisierung werden von [Géron 2017] erläutert.

Der Decoder des Transformers wird wiederholt durchlaufen und besteht aus den gleichen beiden Sublayern wie der Encoder. Zusätzlich wird zu Beginn ein weiterer Multi-Head-Attention-Layer für den Input des Decoders hinzugefügt.
In jedem Durchlauf des Decoders wird ein Outputwert erzeugt. Die bekannten Outputwerte bilden den Input des Decoders für den nächsten Durchlauf. Damit verwendet der Decoder für die Erzeugung des Outputs zum Zeitpunkt $t+1$ alle bisherigen Outputwerte bis zum Zeitpunkt t.

Bezogen auf das Beispiel in Abbildung 6.15 ergibt sich folgendes Vorgehen:
Zu Beginn wurde noch kein Output durch den Decoder erzeugt. Als Input wird dem Decoder ein Wert übergeben, der den Beginn der Output-Sequenz kennzeichnet. (BOS = Begin of Sequence). Der Decoder erzeugt im ersten Durchlauf das Wort „Die“. Dieser Wert wird für den nächsten Durchgang zum Input hinzugefügt. Damit erhält der erste Multi-Head-Attention-Layer im zweiten Durchgang die Werte „BOS“ und „Die“ als Input. Dieser Prozess wird solange wiederholt, bis der Decoder als Output das Ende der Sequenz ausgibt (EOS = End of Sequence) .

Der zweite Muti-Head-Attention-Layer des Decoders berücksichtigt einerseits die Informationen des Outputs des Encoders und andererseits die Informationen aus dem Output des ersten Mulit-Head-Attention-Layers im Decoder. Somit wird in diesem Schritt die Verknüpfung zwischen den Informationen des ursprünglichen Inputs (englischer Satz) und des bisherigen Outputs (deutsche Übersetzung) hergestellt.

Analog zum Encoder wird anschließend ein Feed-Forward Netzwerk durchlaufen. Dessen Output dient als Input für einen Dense-Layer mit linearer Aktivierungsfunktion. Die Anzahl der Neuronen für den Dense-Layer wird hierbei in Abhängigkeit der verfügbaren Wörter zur Übersetzung der NLP-Problemstellung bestimmt. Stehen zur Lösung des Problems beispielsweise 10.000 verschiedenen Wörter zur Verfügung, so beträgt die Größe des Dense-Layers 10.000. Über einen Softmax-Layer werden diesen 10000 Werten Wahrscheinlichkeiten zugeordnet. Der finale Outputwert entspricht abschließend jenem Wort mit der höchsten Wahrscheinlichkeit. Die Zuordnung erfolgt über den Index des Outputs des Softmax-Layers und den Index der Wörter des verfügbaren Wortschatzes für die Übersetzung.

6.2.9.1 Funktionsweise der Multi-Head-Attention

Zur Erläuterung der Multi-Head-Attention wird im nachfolgenden Abschnitt 6.2.9.1.1 zuerst die Funktionsweise der Scaled-Dot-Product-Attention betrachtet, auf der dieser Ansatz basiert.

6.2.9.1.1 Scaled-Dot-Product-Attention

Ausgangspunkt zur Berechnung der Scaled-Dot-Product-Attention bilden die drei Matrizen $Query(Q)$, $Key(K)$ und $Value(V)$. Der allgemeine Ablauf ist in Abbildung 6.16 ersichtlich.

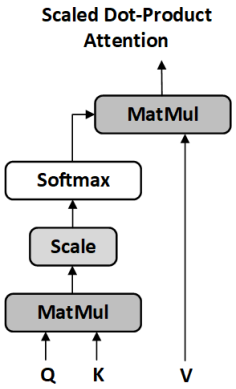


Abbildung 6.16: Scaled-Dot-Product-Attention nach [Vaswani et al. 2017]

Die Query-Matrix beinhaltet diejenigen Werte, *für die* die Attention berechnet werden soll. Die Key-Matrix beinhalten hingegen jene Werte, *anhand derer* die Attention berechnet wird. Die Attention-Werte zwischen diesen beiden Matrizen können mithilfe der Formel (6.25) berechnet werden:

$$a = softmax(\frac{Q * K^T}{\sqrt{d_K}}) \tag{6.25}$$

mit
K ..... Key Matrix
Q .... Query Matrix
a .... Attention Werte zwischen Matrix Q und Matrix K
d_K .... Spaltenanzahl der Key Matrix

Der Faktor d_K entspricht der Spaltenanzahl der Key-Matrix und wird zur Skalierung der Ergebnisse des Skalarproduktes zwischen diesen Matrizen herangezogen. Begründet wird diese Skalierung im Beitrag von [Vaswani et al. 2017] durch die Annahme stabilerer Gradienten für große Werte von d_K.

Abschließend werden für die skalierten Werte durch die Softmax-Funktion Wahrscheinlichkeitswerte zwischen 0 und 1 berechnet. Größeren Werten des Skalarproduktes wird hierbei eine höhere Wahrscheinlichkeit zugeordnet als niedrigeren Werten.

Mithilfe der Scaled-Dot-Product-Attention wird die Bedeutung der Werte von K für die Werte von Q anhand der Bedingung berechnet, dass das Skalarprodukt zwischen zwei Vektoren maximal ist, wenn diese parallel zueinander verlaufen. Die Berechnung des Skalarproduktes durch Formel (6.26) soll diese Aussage verdeutlichen:

$$\vec{a} * \vec{b} = |\vec{a}| * |\vec{b}| * cos\phi \tag{6.26}$$

Verlaufen die beiden Vektoren $\vec{a}$ und $\vec{b}$ parallel zueinander, so beträgt der Winkel ϕ gleich 0°. Damit ergibt sich führ $cos\phi = 1$. In diesem Fall wird das Skalarprodukt zwischen den zwei Vektoren durch die Multiplikation der Beträge berechnet und ist für die beiden Vektoren maximal.

Die berechneten Attention-Werte für die Matrizen Q und K werden abschließend mit der Value-Matrix V multipliziert. Sowohl im Beitrag von [Vaswani et al. 2017] als auch im Rahmen dieser Arbeit entspricht die Value-Matrix V der Key-Matrix K. Der erzeugte Output enthält somit die anhand der Attention für die Query-Matrix Q gewichteten Werte der Key-Matrix.

Die gesamte Berechnung ist in Formel (6.27) ersichtlich:

$$Attention(Q, K, V) = softmax(\frac{Q * K^T}{\sqrt{d_K}}) * V \tag{6.27}$$

mit
K ..... Key-Matrix
Q .... Query-Matrix
V .... Value-Matrix
a .... Attention Werte zwischen Matrix Q und Matrix K
d_K .... Spaltenanzahl der Key-Matrix

Wird wie im Fall des Encoders und der ersten Multi-Head-Attention im Decoder für Q, K und V dieselbe Matrix übergeben, wird diese Berechnung als Self-Attention bezeichnet.

6.2.9.1.2 Multi-Head-Attention

Wie in Abbildung 6.17 ersichtlich ist, wird im Zuge der Multi-Head-Attention die Scaled-Dot-Product-Attention h-mal mithilfe der Matrizen Q, K und V durchgeführt.

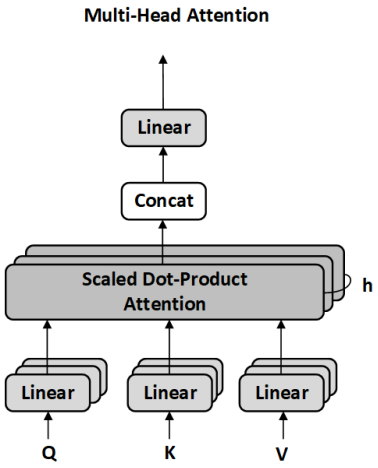


Abbildung 6.17: Multi-Head-Attention nach [Vaswani et al. 2017]

Der Parameter h beschreibt die Anzahl an Heads des Multi-Head-Attention-Layers, wobei im Beitrag von [Vaswani et al. 2017] $h = 8$ angenommen wird. Zusätzlich durchlaufen sowohl Q und K als auch V einen linearen Layer je Head, bevor die Attention berechnet wird. Die Kalkulation der einzelnen Scaled-Dot-Product-Attentions wird parallel durchgeführt und die Ergebnisse werden über einen Concatenate-Layer miteinander verbunden. Abbildung 6.18 zeigt hierzu ein Beispiel der Verknüpfung von acht Heads.

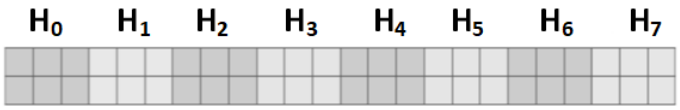


Abbildung 6.18: Verknüpfung Attention Werte

Um zu verhindern, dass die Dimension des Outputs der Multi-Head-Attention nicht mit der Input-Dimension übereinstimmt, wird diese im letzten linearen Layer wieder auf die Dimension der ursprünglichen Value-Matrix V reduziert.

Allgemein kann die Multi-Head-Attention anhand der Formel (6.28) und (6.29) berechnet werden.

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h) * W^O \tag{6.28}$$

$$head_i = Attention(Q * W_i^Q, K * W_i^K, V * W_i^V) \tag{6.29}$$

mit
K ..... Key Matrix
Q .... Query Matrix
V .... Value Matrix
W^Q, W^K, W^V, W^O .... Gewichts Matrizen

6.2.9.2 Funktionsweise Feed-Forward-Netzwerk

Im Anschluss an die Multi-Head-Attention wird sowohl im Encoder als auch im Decoder ein Feed-Forward-Netzwerk durchlaufen. Dieses wird auf jede Zeile der Output-Matrix separat angewandt (Time Distributed). Bezogen auf den Beitrag von [Vaswani et al. 2017] besteht das Feed-Forward Netzwerk aus zwei Dense-Layern. Der erste Layer verwendet eine ReLU-Aktivierungsfunktion. Dem zweiten Layer ist keine Aktivierungsfunktion zugeordnet. Formal wird somit die Berechnung gemäß Formel (6.30) durchgeführt:

$$FNN(x) = \max(0, X * W_1 + b_1) * W_2 + b_2 \tag{6.30}$$

mit
X .... Input-Matrix
$W1$, $W2$ .... Gewichts-Matrizen
$b1$, $b2$ .... Bias-Vektoren

Alternativ zu den Dense-Layern können auch zwei Convolutional-Layer mit Kernel-Size 1

verwendet werden. Analog zu den Dense-Layern wird dem ersten CNN-Layer eine ReLU-Aktivierungsfunktion zugeordnet. Der zweite Layer enthält keine Aktivierungsfunktion [Vaswani et al. 2017].

6.2.10 Transformer im Rahmen der Arbeit

Die im Rahmen dieser Arbeit angewandte Transformer-Architektur basiert auf der Idee des Beitrags von [Vaswani et al. 2017] und der beschriebenen Funktionsweise laut Abschnitt 6.2.9. Der Aufbau ist in Abbildung 6.19 ersichtlich.

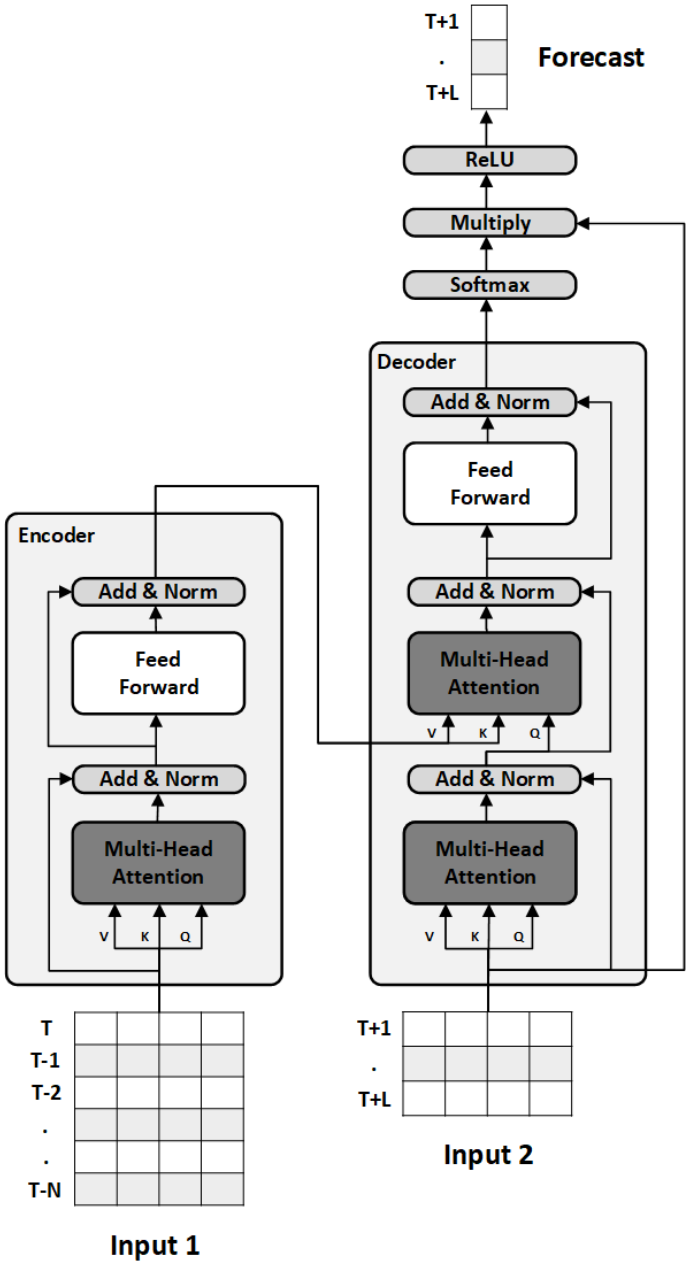


Abbildung 6.19: Aufbau Transformer Forecast

In den Abschnitten 6.2.10.1 bis 6.2.10.3 werden daher nur die Unterschiede zur Standard-Architektur beschrieben. Die Funktionsweisen der Multi-Head-Attention, des Feed-Forward-Netzwerkes und des Residual-Dropouts entsprechen den Vorgehensweisen nach [Vaswani et al. 2017].

6.2.10.1 Positional-Encoding im Transformer-Modell

Das Positional-Encoding und damit die Zuordnung zeitlicher Abstände zwischen den Inputwerten, werden durch das Hinzufügen der Daten Jahr, Monat, Woche und Anzahl der Arbeitstage in der jeweiligen Woche erreicht. Damit wird dem Transformer sowohl im Encoder als auch im Decoder eine Matrix mit nachfolgenden Daten übergeben:

| Jahr | Monat | Woche | Arbeitstage | **Abgesetzte Menge** | Budget Menge | Menge des Vorjahres | Aktionsmenge |

Analog zum CNN-LSTM-Modell (Abschnitt 6.2.7) entspricht die Inputgröße des Encoders den Einträgen der letzten 52 Wochen. Die Inputgröße des Decoders wird durch die Größe des Prognosezeitraums je Produkt bestimmt.

6.2.10.2 Aufbereitung der Inputdaten im Decoder

Der Input des Decoders entspricht den bekannten Werten für den gesuchten Prognosezeitraum. Wird beispielsweise eine Prognose für die Kalenderwochen 49, 50 und 51 erstellt, so entspricht der Input des Decoders den bekannten Werten dieser Wochen.

Diese Vorgehensweise führt dazu, dass ohne weitere Anpassungen dem Decoder im Trainingsprozess die gesuchten Absatzmengen übergeben werden.

| Jahr | Monat | Woche | Arbeitstage | **Abgesetzte Menge** | Budget Menge | Menge des Vorjahres | Aktionsmenge |

Die Absatzmengen des Decoders werden aus diesem Grund durch einen naiven Forecast der letzten bekannten Woche ersetzt. Im beschriebenen Beispiel enthält die Spalte der Absatzmenge für die Wochen 49, 50, und 51 somit jeweils den bekannten Wert der Woche 48.

| Jahr | Monat | Woche | Arbeitstage | **Naiver Forecast** | Budget Menge | Menge des Vorjahres | Aktionsmenge |

Neben der Verwendung eines naiven Forecasts ermöglicht diese Architektur auch das Hinzufügen alternativer Prognosen, z.B. anhand der in Abschnitt 6.1.3 beschriebenen ARIMA-Modelle. Diese Ansätze wurden im Rahmen dieser Arbeit nicht untersucht und bieten Anlass für weiterer Forschungsarbeiten.

6.2.10.3 Output des angewandten Transformers

Im Unterschied zur Standard-Architektur werden die Werte des Inputs des Decoders elementweise mit den Outputwerten der Softmax-Funktion gemäß Abbildung 6.19 multipliziert. Dadurch werden die Inputwerte des Decoders und damit die bekannten Werte für den Prognosezeitraum, mit der Wahrscheinlichkeit multipliziert, dass die zukünftige Absatzmenge diesem Wert entspricht. Die Spalten Jahr, Monat, Woche und Arbeitstag werden in der Berechnung der Softmax-Funktion nicht berücksichtigt und erhalten eine Wahrscheinlichkeit von 0. Dieser Effekt wird durch eine Masking-Matrix im Softmax-Layer des verwendeten Python-Keras-Pakets erreicht.

Die Absatzmenge wird abschließend über einen Dense-Layer (Time Distributed) mit einer ReLU-Aktivierungsfunktion berechnet.

6.2.10.4 Parameter des angewandten Transformers Modells

Die Tabellen 6.2 und 6.3 zeigen einen Überblick der wichtigsten verwendeten Hyperparameter im Transformer-Modell. Dropout-Layer wurden in der Tabelle nicht berücksichtigt.

Hyperparameter im Encoder:

Hyperparameter	**Wert**
Anzahl Module Encoder (Nx)	1
Input	
Input-Size	52×8
Multi-Head Attention	
Anzahl Heads	10
CNN 1	
Anzahl Filter	10
Aktivierungsfunktion	ReLU
Kernel-Size	1
Padding	Causal
CNN 2	
Anzahl Filter	8
Kernel-Size	1
Padding	Causal

Tabelle 6.2: Hyperparameter Encoder Transformer-Modell

Hyperparameter im Decoder:

Hyperparameter	**Wert**
Anzahl Module Decoder (Nx)	1
Input	
Input-Size	52×8
Multi-Head Attention 1	
Anzahl Heads	10
Multi-Head Attention 2	
Anzahl Heads	10
CNN 1	
Anzahl Filter	10
Aktivierungsfunktion	ReLU
Kernel-Size	1
Padding	Causal
CNN 2	
Anzahl Filter	8
Kernel-Size	1
Padding	Causal
Dense Layer	
Anzahl Neuronen	1 (Time Distributed)
Aktivierungsfunktion	ReLU

Tabelle 6.3: Hyperparameter Decoder Transformer Modell

7 Modellevaluierung

Das nachfolgende Kapitel zeigt die im Rahmen dieser Arbeit angewandten Verfahren der Modellevaluierung. Hierzu werden in den Abschnitten 7.1 und 7.2 die Vorgehensweisen zur Beurteilung der logistischen Kriterien aufgezeigt. Abschnitt 7.3 zeigt das Vorgehen zum Vergleich der angewandten Modelle und der aktuellen Planung.

7.1 Auswertung der Lieferfähigkeit

Die Lieferfähigkeit wird im Rahmen dieser Arbeit anhand Formel (7.1) und für jedes Produkt einzeln berechnet. Hierbei wird die Anzahl der Arbeitstage, an denen ein Kundenauftrag aufgrund eines zu geringen Lagerbestandes nicht erfüllt werden kann n_{OOS} (OOS = Out of Stock), der Gesamtzahl an Arbeitstagen mit Kundenbedarf n_A gegenübergestellt.

$$L = \frac{n_{oos}}{n_A} * 100 \tag{7.1}$$

mit
L .... Lieferfähigkeit in Prozent
n_{oos} .... Anzahl der Arbeitstage an denen nicht geliefert werden konnte
n_A .... Anzahl der Arbeitstage im Betrachtungszeitraum mit Kundenbedarf

Für den Vergleich der unterschiedlichen Planungsmethoden werden die berechneten Lieferfähigkeiten der einzelnen Produkte kumuliert in einem Histogramm dargestellt.
Die Beurteilung erfolgt anhand der Anzahl an Produkten, welche die geforderte Lieferfähigkeit von 90 Prozent unterschreiten und der allgemeinen Verteilung der Lieferfähigkeiten im Histogramm für die angewandten Modelle.

7.2 Auswertung des Lagerbestandes

Die Auswertung der Veränderung des Lagerbestandes wird durch den Vergleich des mittleren Gesamtbestandes je Prognosemodell bzw. der aktuellen Planung durchgeführt. Formel (7.2) zeigt die Berechnung des durchschnittlichen Lagerbestandes im Rahmen dieser Arbeit.

$$\overline{x} = \frac{1}{n_A} * \sum_{t=1}^{n_A} \lceil x_t^p \rceil \tag{7.2}$$

mit
$\overline{x}$ .... mittlerer Lagerbestand
x_t .... Lagerbestand zum Zeitpunkt t
n_A .... Anzahl der Arbeitstage mit Kundenbedarf im Betrachtungszeitraum
p .... ganzzahliger Exponent

Der Exponent p wird im Zuge der Simulation (Kapitel 8) und in Zusammenspiel mit der im nachfolgenden Abschnitt 7.3 beschriebenen Evaluierungsmethodik dazu verwendet, um höhere Lagerbestände sukzessive stärker zu bewerten.

7.3 Auswertung der unterschiedlichen Planungsmethoden

Um sicherzustellen, dass eine Verbesserung des mittleren Lagerbestandes des Prognosemodells A nicht durch einzelne Prognosefehler des Modells B verursacht wird, erfolgt im Rahmen der Evaluierung der Modelle eine sukzessive Anpassung des Modells B ohne dessen Lieferfähigkeit zu beeinflussen. Die Anpassung wird anhand folgender Schritte durchgeführt:

1. Berechnung der Differenz zwischen den prognostizierten und abgesetzten Wochenmengen (in Paletten) pro Produkt
2. Berechnung der maximal möglichen Anpassung (in Paletten) pro Monat für jedes Produkt
3. Berechnung der maximal möglichen Monatsanpassung aller Produkte
4. Reduzierung der Monatsplanung jenes Produktes mit der maximalen Anpassungsmöglichkeit

Diese Schritte werden so lange wiederholt, bis der Mittelwert des Lagerbestandes für das Modell B geringer ist als jener für das Modell A. In der in Kapitel 8 beschriebenen Simulation, wird dieser Prozess auf Basis von Formel (7.2) für mehrere Exponenten p

durchgeführt.

Formal können die Schritte je Produkt anhand der Formeln (7.3) bis (7.5) berechnet werden.

Berechnung der Differenz zwischen den prognostizierten und abgesetzten Wochenmengen (in Paletten) pro Produkt

$$x_{Diff_{Prod,W}} = x_{Prognose_{Prod,W}} - x_{real_{Prod,W}} \tag{7.3}$$

mit
$x_{Diff_{Prod,W}}$ .... Differenz des Produktes *Prod* in der Woche *W*
$x_{Prognose_{Prod,W}}$ .... Prognose des Modells für das Produkt *Prod* in der Woche *W*
$x_{real_{Prod,W}}$ .... reale Absatzmenge des Produktes *Prod* in der Woche *W*

Berechnung der maximal möglichen Anpassung (in Paletten) pro Monat für jedes Produkt

$$min_{Diff_{Prod,Mon}} = min(\frac{x_{Diff_{Prod,W}}}{n_W} * n_{W,Mon}) \tag{7.4}$$

mit
$min_{Diff_{Prod,Mon}}$ .... maximal mögliche Anpassungen des Produktes *Prod* im Monat *Mon*
$x_{Diff_{Prod,W}}$ .... Differenz des Produktes *Prod* in der Woche *W*
n_W .... Anzahl der Arbeitstage in der Woche *W*
$n_{W,Mon}$ .... Anzahl der Arbeitstage der Woche *W* die den Monat *Mon* betreffen

Berechnung der maximal möglichen Monatsanpassung aller Produkte

$$max_{Diff} = max(x_{Diff_{Prod,Monat}}) \tag{7.5}$$

mit
max_{Dif} .... maximal mögliche Anpassungsmenge
$x_{Diff_{Prod,W}}$ .... Differenz des Produktes *Prod* in der Woche *W*

8 Simulation und Ergebnisevaluierung

Die nachfolgenden Abschnitte des Kapitels 8 zeigen die Ergebnisse der Masterarbeit. Zur Beantwortung der Forschungsfragen gemäß Abschnitt 1.1 wurden im Rahmen dieser Arbeit vier wesentliche Schritte durchlaufen, die in Abbildung 8.1 ersichtlich sind und nachfolgend beschrieben werden.

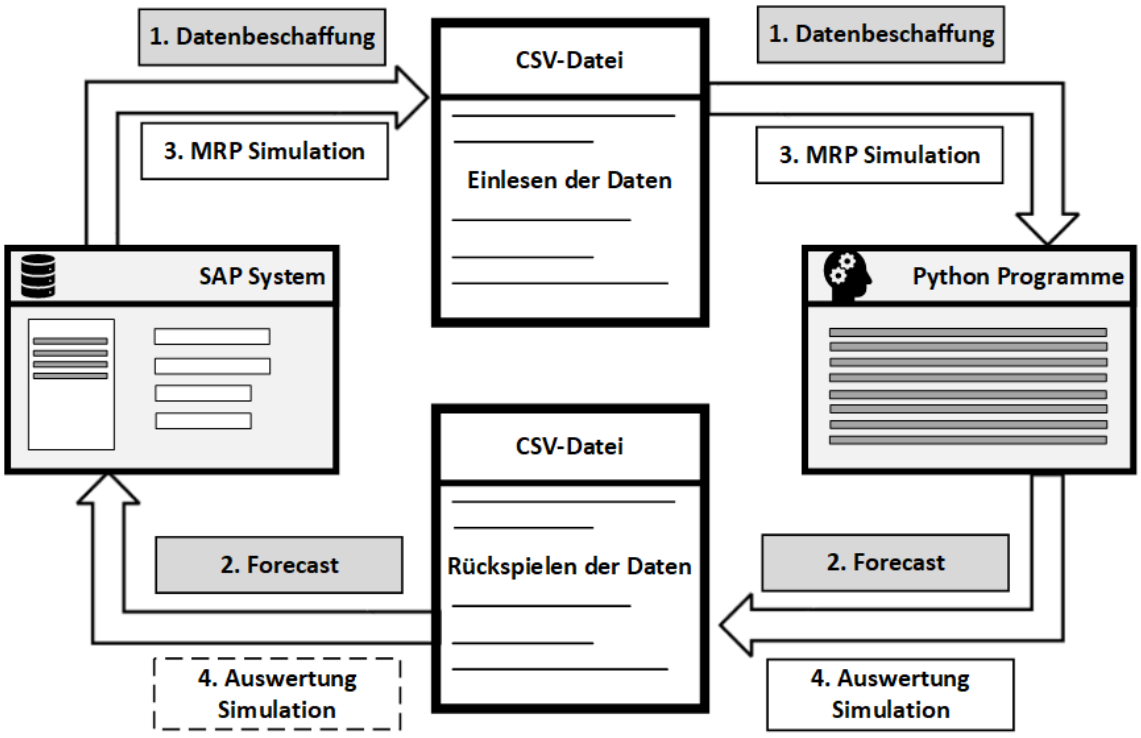


Abbildung 8.1: Übersicht Ablauf Simulation

Für die unterschiedlichen Aufgaben der Datenbeschaffung (1. Datenbeschaffung), wurden im SAP-ERP-System des betrachteten Unternehmens verschiedene Klassen und Methoden implementiert. Diese Methoden ermöglichen sowohl eine Auswertung der für die Problemstellung relevanten Materialstammdaten als auch eine Analyse der unterschiedlichen Materialbelege wie Kundenaufträge, Wareneinn- und Warenausgänge. Für eine einfache Bedienbarkeit wurden die für die Datenbeschaffung benötigten Methoden in ein eigenständiges Programm mit unterschiedlichen Selektionsbedingungen zusammengefasst.

Die Selektionsmaske des Programms ist in Abbildung 8.2 ersichtlich.

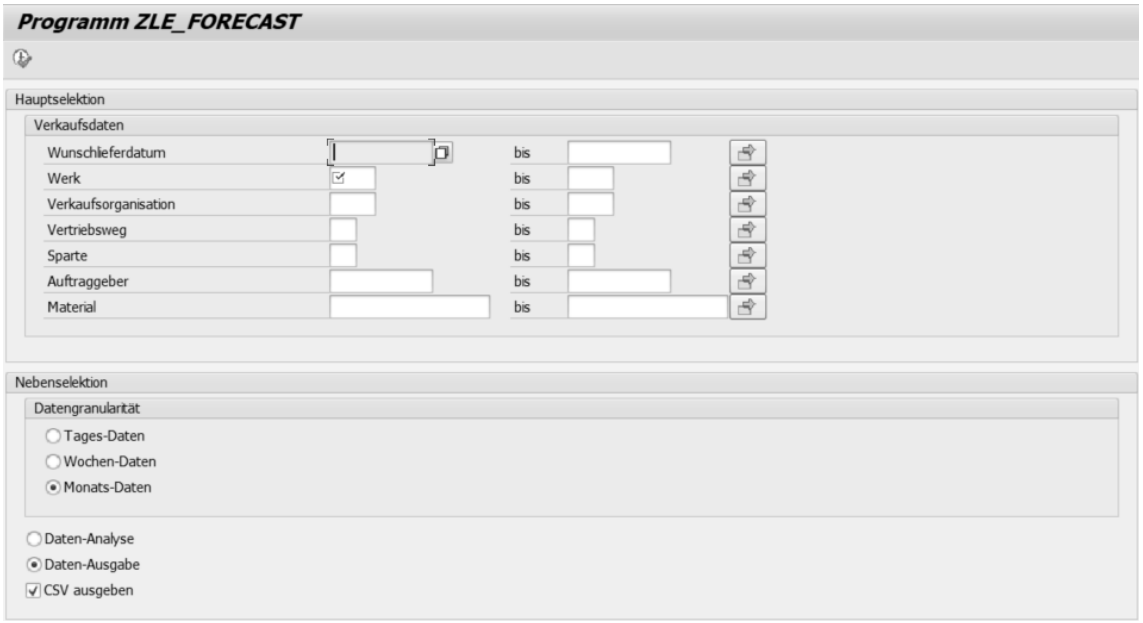


Abbildung 8.2: Programm Datenbeschaffung SAP

Die *Hauptselektion* des Programms ermöglicht es dem jeweiligen User, Einschränkungen für Detailanalysen vorzunehmen. Die Eingabe des Werks, für das die Analyse durchgeführt werden soll, bzw. die Eingabe des Start- und Enddatums der Analyse sind obligatorisch. Wird einem Parameter, z.B. dem Material, kein Wert übergeben, so erfolgt die Auswertung auf alle verfügbaren Daten im jeweiligen Werk.

Über die *Nebenselektion* kann zwischen einer *Daten-Analyse* und einer *Daten-Ausgabe* unterschieden werden. Im Rahmen der Daten-Analyse werden alle benötigten Daten für die in den Abschnitten 4.2 und 4.4 beschriebenen ABC und XYZ-Analysen je Produkt aufbereitet. Der Parameter der *Datengranularität* gibt an, ob die für die XYZ-Analyse benötigten Mengen der Kundenbedarfe tageweise, wochenweise oder monatsweise summiert werden. Im Rahmen dieser Arbeit wurden die abgesetzten Mengen je Woche und Produkt summiert.

Die Selektion der Daten-Ausgabe ermöglicht die Aufbereitung aller Daten, die zur Erstellung der Prognosen benötigt werden und als Input für die einzelnen Prognosemodelle dienen. Analog zur ABC/XYZ-Analyse kann über den Parameter der Datengranularität festgelegt werden, auf welcher Basis die unterschiedlichen Mengen summiert werden. Für die Analyse im Rahmen dieser Arbeit wurden alle Mengen wieder je Woche und Produkt summiert.

Sowohl die Ergebnisse der Daten-Analyse als auch die Daten-Ausgabe können über einen Download einer csv-Datei lokal am jeweiligen Computer abgelegt werden. Der Download kann über eine Check-Box (*CSV ausgeben*) in der Nebenselektion gesteuert werden.

Die im Rahmen der Daten-Analyse erstellten csv-Dateien dienen als Basis für den zweiten Schritt der Prognoseerstellung (2. Forecast). In diesem werden die Prognosen der für die Problemstellung relevanten A/Z-Produkte erstellt. Für jedes Produkt und Verfahren wird hierbei ein eigenes Modell implementiert. Folgende Modelle wurden im Rahmen dieser Arbeit evaluiert:

- ARIMA-Modell vgl. Abschnitt 6.1.3
- CNN-LSTM-Modell vgl. Abschnitt 6.2.7
- Transfomer-Modell vgl. Abschnitt 6.2.10

Die Ergebnisse der Prognosen der einzelnen Modelle werden lokal in einer csv-Datei gespeichert und im dritten Schritt als Input für die Simulation der Materialbedarfsplanung herangezogen (3. MRP-Simulation).

Analog zum Schritt der Datenbeschaffung, erfolgt die MRP-Simulation im SAP-ERP-System des betrachteten Unternehmens und mithilfe von eigenentwickelten Klassen und Methoden. Die Berechnung orientiert sich hierbei an den Beschreibungen des Ist-Prozesses nach Kapitel 2 und der zusammengefassten MRP-Berechnung nach Algorithmus 1. Für die abschließende Auswertung in Python, werden die Ergebnisse der MRP-Simulation lokal als csv-Datei gespeichert und anschließend in die jeweiligen Programme importiert.

Im Rahmen der Arbeit wurde kein abschließender Import der Auswertungsergebnisse in SAP-ERP durchgeführt (4. Auswertung Simulation gestrichelt). Dieser Schritt kann für eine mögliche Anwendung der Modelle im operativen Tagesgeschäft zukünftig angedacht werden.

Die nachfolgenden Abschnitte 8.1 bis 8.4 zeigen die Ergebnisse der einzelnen Prognosemodelle.

8.1 Auswertung des Gesamtergebnisses

In diesem Abschnitt werden die zur Beantwortung der Hauptforschungsfrage durchgeführten Auswertungen beschrieben.

Hauptforschungsfrage

Welche Auswirkungen auf den Lagerbestand von Fertigteilen zeigen sich durch die Bestimmung der Planprimärbedarfe mithilfe von neuronalen Netzen gegenüber der aktuellen Materialbedarfsplanung?

Zur Beantwortung der Hauptforschungsfrage und der beiden Forschungsunterfragen wird folgende Vorgehensweise angewandt:

1. Vergleich der Verteilung der Lieferfähigkeit mithilfe einer Darstellung als Histogramm
2. Vergleich des mittleren Gesamtbestands für das Jahr 2021
3. Vergleich des mittleren Gesamtbestands für das Jahr 2021 mithilfe von Formel (7.2) und Exponenten von 1-10

Alle Auswertungen beziehen sich auf die analysierten A/Z-Produkte ohne fehlerhafte Stammdaten (123 Produkte). Die Auswirkung fehlender Stammdaten wird in Abschnitt 8.4 beschrieben.

Die reale Dispositionsplanung des betrachteten Unternehmens wird mit Bezug auf das in Abschnitt 2.1.1 beschriebene Dispositionscockpit in allen Auswertungen als *Cockpit-Planung* bezeichnet. In dieser Planung sind keine nachträglichen Änderungen im Zuge der Produktionsfeinplanung berücksichtigt.

8.1.1 Auswertung der Lieferfähigkeit

Abbildung 8.3 zeigt die Verteilung der Lieferfähigkeit für die gesamten A/Z-klassifizierten Produkte. Die Lieferfähigkeit wird hierbei anhand Formel 7.1 berechnet. Auf der x-Achse ist die prozentuale Lieferfähigkeit aufgetragen. Die y-Achse zeigt den kumulierten prozentualen Anteil jener Produkte, die eine bestimmte Lieferfähigkeit der x-Achse unterschreiten bzw. genau erfüllen. Der schwarze Balken stellt mit 89 Prozent die Grenze zur definierten Mindestanforderung von 90 Prozent Lieferfähigkeit dar.

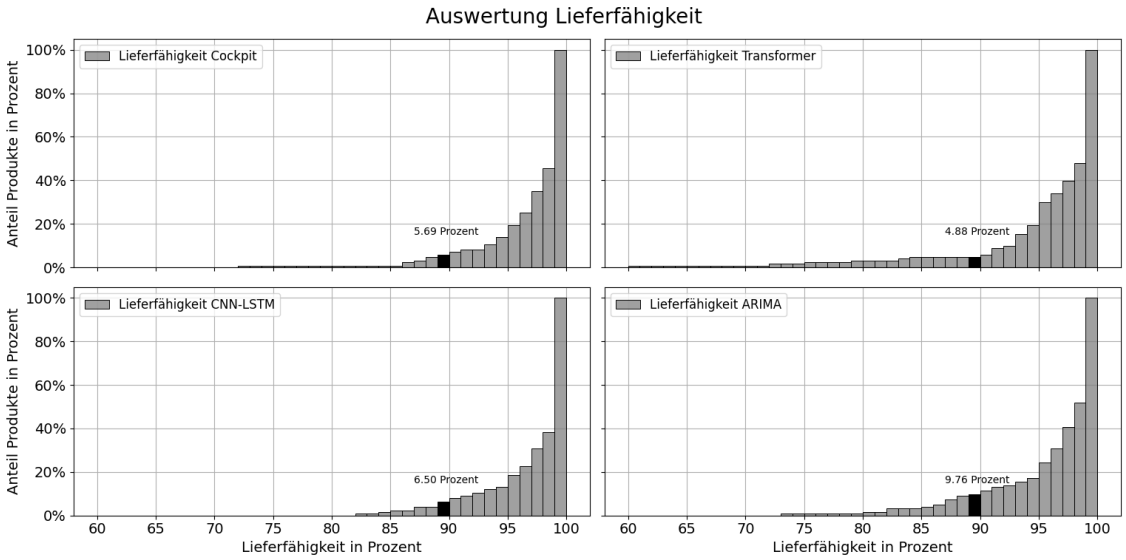


Abbildung 8.3: Lieferfähigkeit gesamt

Die Verteilung der Lieferfähigkeit in Abbildung 8.3 zeigt, dass das Transformer-Modell (Verteilung rechts oben) die geforderte Lieferfähigkeit von 90 Prozent für 4.88 Prozent der betrachteten Produkte nicht erfüllt. Im Vergleich zur aktuellen Cockpit-Planung (Verteilung links oben) kann der Anteil jener Produkte, welche die Mindestandforderung nicht erfüllen, somit um 0.81 Prozentpunkte gesenkt werden. Bezogen auf die 123 analysierten Produkte entspricht diese Senkung einem Wert von einem Produkt. In Bezug auf die Einhaltung der Mindestanforderung der Lieferfähigkeit sind für den Vergleich des Transformer-Modells mit der aktuellen Cockpit-Planung somit keine Vor- bzw. Nachteile gegeben.

Wie in Abbildung 8.3 ersichtlich ist, entspricht die minimale Lieferfähigkeit eines Produktes des Transformer-Modells (Verteilung rechts oben) knapp 60 Prozent. Jene Produkte, welche die Lieferfähigkeit von 90 Prozent unterschreiten, weisen im Vergleich zwischen Transformer-Modell und Cockpit-Planung generell für das Transformer-Modell geringere Lieferfähigkeiten auf. Diese Unterschreitung kann auf Produkte mit Dispositionsverfahren ZK (Fixed-Order-Period) und Produktionszyklen über zwei Wochen und Sicherheitsbe-

ständen unter fünf Tagen zurückgeführt werden. Dadurch, dass im Zuge der Simulation die festgelegten Perioden fix eingehalten werden, entstehen für diese Produkte längere Zeiträume ohne Lagerbestand bei einer frühzeitigen Out-of-Stock Situation. Das Transformer-Modell zeigt in Bezug auf die Lieferfähigkeit für diese spezifische Produktgruppe Nachteile gegenüber den alternativen Planungsmethoden bzw. der Cockpit-Planung. Insgesamt entspricht diese Gruppe 4.87 Prozent der analysierten Produkte

Für das CNN-LSTM-Modell (Verteilung links unten) und das ARIMA-Modell (Verteilung rechts unten) liegen die Anteile jener Produkte, die das Mindestkriterium nicht erfüllen bei 6.50 Prozent bzw. 9.76 Prozent. Diese Prozentsätze entsprechen in Bezug auf das Transformer-Modell einem Zuwachs von zwei bzw. sechs Produkten. Das CNN-LSTM stellt jenes Modell mit den meisten Produkten dar, welche eine 100-prozentige Lieferfähigkeit besitzen. Die Auswirkung dieser Lieferfähigkeit auf den Lagerbestand wird in Abschnitt 8.1.2 betrachtet. Bezogen auf die Lieferfähigkeit sind im Vergleich der einzelnen Modelle keine weiteren wesentlichen Unterschiede ersichtlich.

8.1.2 Auswertung des Lagerbestandes

Für die Auswertung der Veränderung des Lagerbestandes wird im ersten Schritt die prozentuale Abweichung der Ergebnisse der Prognosemodelle in Bezug auf die Cockpit-Planung analysiert. Verglichen wird hierbei der mittlere Lagerbestand je Modell, bezogen auf die täglichen Lagerbestände im Geschäftsjahr 2021. Das Ergebnis ist in Tabelle 8.1 ersichtlich.

Transformer vs. Cockpit	**ARIMA vs. Cockpit**	**CNN-LSTM vs. Cockpit**
-0.81 % (68 Paletten)	+0.57 % (40 Paletten)	+7.45 % (502 Paletten)

Tabelle 8.1: Abweichung Mittelwert Modelle vs. Cockpit (Gesamt)

Das Transformer-Modell stellt mit 68 Paletten (-0.81 Prozent) hierbei das einzige Modell dar, das eine Reduzierung des durchschnittlichen Lagerbestandes pro Tag gegenüber der Cockpit-Planung erzielt. In Bezug auf den Gesamtbestand ist diese Reduzierung allerdings nicht wesentlich, sodass kein Vor- bzw. Nachteil durch das Transformer-Modell entsteht.

Die Auswirkung der vermehrten 100-prozentigen Lieferfähigkeit ist für das CNN-LSTM-Modell im durchschnittlichen Lagerbestand ersichtlich. Dieser wird um 502 Paletten (7.45 Prozent) erhöht. Das ARIMA-Modell zeigt gegenüber der Cockpit-Planung eine Erhöhung des durchschnittlichen Lagerbestandes um 40 Paletten (0.57 Prozent).

Um sicherzustellen, dass die Reduzierung des Lagerbestandes durch das Transformer-Modell nicht aufgrund einzelner Überplanungen in der Cockpit-Planung entsteht, erfolgt

eine zusätzliche Auswertung des Lagerbestandes anhand des beschriebenen Prozesses in Abschnitt 7.3. Dieses Vorgehen wird insgesamt zehn Mal wiederholt, wobei der mittlere Lagerbestand mithilfe der Formel (7.2) berechnet wird. In jedem Durchgang wird der Exponent p um 1 erhöht. Damit werden höhere Lagerbestände sukzessive stärker gewichtet und in der Beurteilung berücksichtigt.

Das Ergebnis dieser Auswertung ist in Abbildung 8.4 ersichtlich. Jede Linie in der Abbildung zeigt den Vergleich zwischen zwei Planungsmethoden, Planung A und Planung B. Die Planung B wird sukzessive anhand des beschriebenen Prozesses in Abschnitt 7.3 angepasst. Die Planung A wird konstant gehalten.

Die x-Achse zeigt den Exponenten p des jeweiligen Durchgangs. Auf der y-Achse ist die Anzahl der benötigten Anpassungen aufgetragen, die durchgeführt wurden, damit der berechnete Mittelwert der Planung B zumindest den Mittelwert der Planung A entspricht.

In der oberen Grafik werden die im Rahmen dieser Arbeit erstellten Prognosemodelle (Planung A) mit der Cockpit-Planung (Planung B) verglichen. Die untere Grafik zeigt den Vergleich zwischen dem Transformer-Modell (Planung A) und dem ARIMA- bzw. CNN-LSTM-Modell (Planung B).

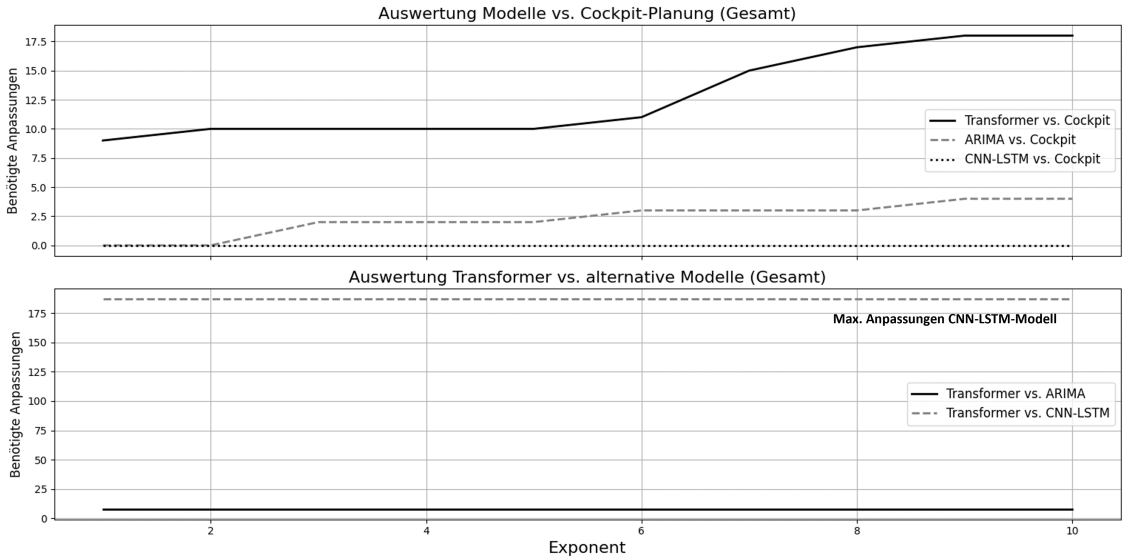


Abbildung 8.4: Auswertung Lagerbestand gesamt

Die Auswertung des Lagerbestandes zeigt, dass das Transformer-Modell jenen Ansatz darstellt, für welchen die Cockpit-Planung am häufigsten angepasst werden muss, um gleiche Ergebnisse in Bezug auf den mittleren Lagerbestand zu erzielen. Bei zunehmender Gewichtung der höheren Lagerbestände muss die Anzahl der Anpassungen weiter erhöht

werden. Damit tendiert das Transformer-Modell allgemein zu geringeren Lagerbeständen im Vergleich zur Cockpit-Planung. Die Einsparungspotenziale liegen, in Tabelle 8.1 beschrieben, im Bereich von knapp einem Prozent. Bezogen auf die Auswertung des Gesamtbestandes kann die Verminderung des Lagerbestandes je nach Exponent auf 9 bis 18 Überplanungen des Cockpits gegenüber dem Transformer-Modell zurückgeführt werden. Die Anpassungen einer Monatsplanung liegen im Bereich von 203 bis 69 Paletten.

Die Auswertung der ARIMA-Modelle kann analog zu den Transformer-Modellen interpretiert werden. Allerdings ist in Bezug auf die Cockpit-Planung eine geringere Anzahl an Anpassungen nötig. Für das CNN-LSTM wird keine Anpassung der Cockpit-Planung durchgeführt. Der mittlere Lagerbestand liegt für dieses Modell immer über den Ergebnissen der Cockpit-Planung.

Beim Vergleich des Transformer-Modells mit dem ARIMA-Modell zeigt sich, dass unabhängig vom Exponenten p zur Berechnung des Mittelwertes die Anzahl der Anpassungen konstant bleibt. Damit erzielt das ARIMA-Modell durch das Umplanen von acht Einträgen gleiche Ergebnisse wie das Transformer-Modell. Die Anpassungen für eine Prognose des ARIMA-Modells liegen zwischen 470 und 170 Paletten für eine Monatsplanung eines Produktes.

Analog zum Vergleich mit der Cockpit-Planung konnten mithilfe des CNN-LSTM-Modells auch die Ergebnisse des Transformer-Modells nie erreicht werden. Abbildung 8.4 zeigt hierzu eine konstante Linie bei 187 Anpassungen. Dieser Wert entspricht den maximal möglichen Anpassungen des CNN-LSTM, ohne die Lieferfähigkeit zu reduzieren.

Die Ergebnisse der Evaluierung zeigen, dass durch die Verwendung von neuronalen Netzen zur Prognose der Primärbedarfe in der Materialbedarfsplanung keine wesentliche Reduzierung des Lagerbestandes gegenüber der aktuellen Cockpit-Planung erzielt werden kann. Das Transformer Modell stellt allerdings einen Ansatz dar, der äquivalente Ergebnisse zur Cockpit-Planung liefert.

Durch diesen Modellansatz konnten für das Kalenderjahr 2021 zwar nur geringfügige Einsparungen im Bereich von einem Prozent und bezogen auf den Gesamtlagerbestand erreicht werden. Dadurch, dass das Transformer-Modell auch keine wesentlichen Veränderungen in der Lieferfähigkeit zeigt, bietet dieser Ansatz das Potential für eine automatisierte bzw. unterstützende Planung.

Bezogen auf die Hauptforschungsfrage zeigen sich durch die Anwendung von neuronalen Netzen zur Bestimmung der Planprimärbedarfe somit keine wesentlichen Potentiale in

Bezug auf die Senkung des Lagerbestandes. Durch das Transformer-Modell wurde allerdings ein Ansatz evaluiert, der aufgrund der Möglichkeit einer automatisierten Planung, Potentiale bezüglich der Reduzierung der innerbetrieblichen Aufwände zur Erstellung der Prognosen zeigt.

8.2 Auswertung auf Basis der Dispositionsverfahren

Der nachfolgende Abschnitt zeigt die Auswertungen des Lagerbestandes und der Lieferfähigkeit unter Berücksichtigung des Dispositionsverfahrens. Damit wird die Beantwortung folgender Forschungsunterfrage ermöglicht:

Forschungsunterfrage 1

Welche Einflüsse auf die Potentiale der Anwendung von neuronalen Netzen gegenüber der aktuellen Planung ergeben sich durch die unterschiedlichen Verfahren der Materialbedarfsplanung?

Die Vorgehensweisen der einzelnen Auswertungen sind analog zu Abschnitt 8.1.

8.2.1 Auswertung der Produkte mit fixer Losgröße (EX)

Für die Produkte mit Dispositionsverfahren *EX* wird die Losgröße der Primärbedarfe anhand des Algorithmus 1 in Abschnitt 2.1.3.3 ermittelt. Die Berechnung erfolgt hierbei täglich. Die genaue Ermittlung des Prognosehorizontes für Produkte mit Dispositionsverfahren *EX* wird in Abschnitt 3.3 beschrieben.

8.2.1.1 Auswertung der Lieferfähigkeit

Wie in Abbildung 8.5 ersichtlich, wird für die Produkte mit Dispositionsverfahren EX die minimale Lieferfähigkeit von 90 Prozent von allen Planungsmethoden, mit Ausnahme des ARIMA-Modells, zu 100 Prozent erfüllt. Für das ARIMA-Modell konnte die Lieferfähigkeit von 90 Prozent für ein Produkt nicht erreicht werden und liegt für dieses Produkt bei 87 Prozent. Im Vergleich zwischen dem Transformer-Modell (Verteilung rechts oben) und der Cockpit-Planung (Verteilung links oben) sind für die Produkte mit Dispositionsverfahren EX keine wesentlichen Unterschiede ersichtlich. Analog zur Analyse des Gesamtbestandes stellt das CNN-LSTM-Modell (Verteilung links unten) jenes Verfahren mit den meisten Produkten dar, die eine 100-prozentige Lieferfähigkeit aufweisen.

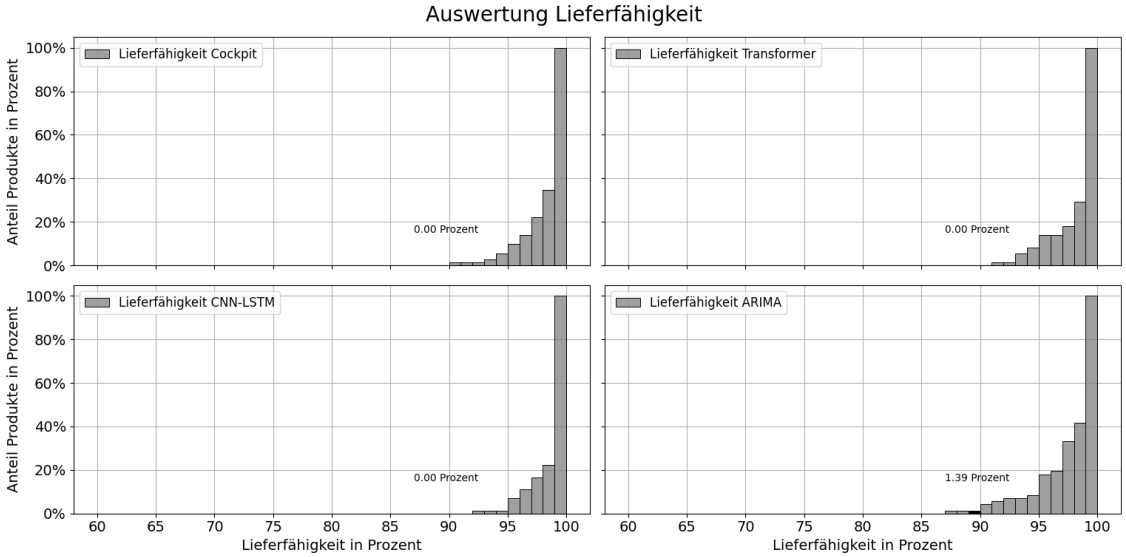


Abbildung 8.5: Lieferfähigkeit Dispositionsverfahren EX

8.2.1.2 Auswertung des Lagerbestandes

Bezogen auf die prozentuale Abweichung des durchschnittlichen Lagerbestandes pro Tag, stellt das Transformer-Modell mit durchschnittlich 32 Paletten (1.22 Prozent) den Modellansatz mit den größten Einsparungspotential gegenüber der Cockpit-Planung dar. Analog zur Beurteilung im Abschnitt 8.1.2, ist somit keine wesentliche Veränderung in Bezug auf die Senkung des Lagerbestandes und durch den Einsatz des Transformer-Modells zu erkennen.

Das Ergebnis des ARIMA-Modells zeigt in Bezug auf den durchschnittlichen Lagerbestand keinen wesentlichen Unterschied zum Transformer-Modell. Die Reduzierung des Lagerbestandes in Bezug auf die Cockpit-Planung liegt bei 11 Paletten (0.49 Prozent).

Das CNN-LSTM-Modell zeigt in Bezug auf die Cockpit-Planung eine wesentliche Erhöhung des durchschnittlichen Lagerbestandes um 353 Paletten (15.75 Prozent). Die Ergebnisse sind in Tabelle 8.2 ersichtlich.

Transformer vs. Cockpit	**ARIMA vs. Cockpit**	**CNN-LSTM vs. Cockpit**
-1.22 % (32 Paletten)	-0.49 % (11 Paletten)	+15.75 % (353 Paletten)

Tabelle 8.2: Abweichung Mittelwert zur Cockpit-Planung (EX)

Analog zur Evaluierung des Gesamtbestandes wird für die Produkte mit Dispositionsverfahren EX die Anzahl der Anpassungen berechnet, die benötigt werden, damit z.B. die Cockpit-Planung die gleichen Ergebnisse erzielt wie das Transformer-Modell. Die Berech-

nung wird wieder auf Basis des Prozesses laut Abschnitt 7.3 und für Exponenten von 1 bis 10 durchgeführt. Das Ergebnis der Evaluierung ist in Abbildung 8.6 dargestellt.

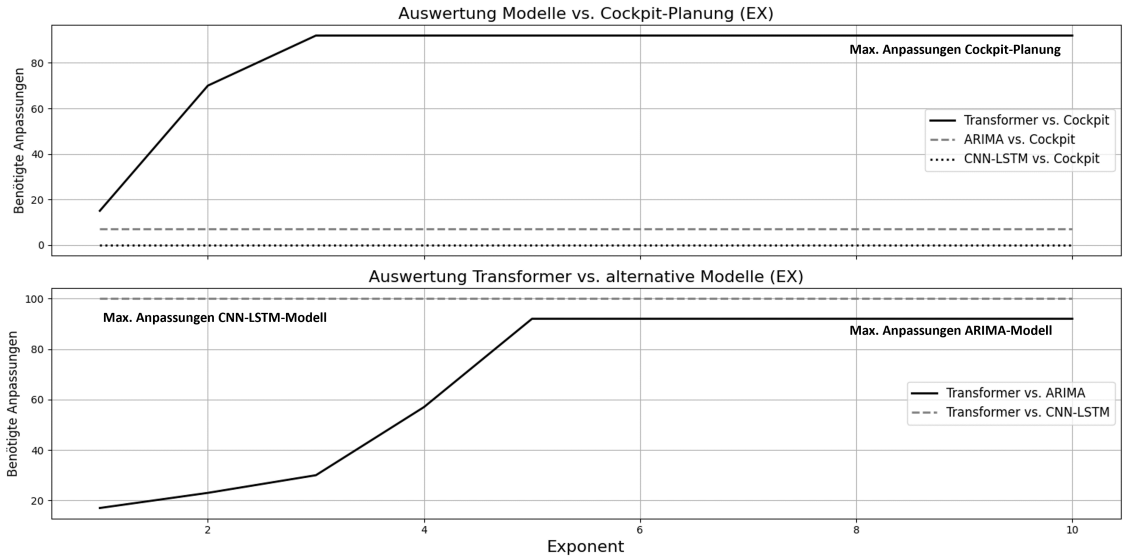


Abbildung 8.6: Auswertung Lagerbestand Dispositionsverfahren EX

Im Vergleich zwischen Transformer-Modell und Cockpit-Planung (Abbildung oben) ist ersichtlich, dass ab einem Exponenten von 3 konstant 92 Anpassungen der Cockpit-Planung benötigt werden. Diese 92 Anpassungen entsprechen der maximal möglichen Anzahl an Reduzierungen, die für die Cockpit-Planung durchgeführt werden kann, ohne die Lieferfähigkeit zu senken. Damit erreicht die Cockpit-Planung bezogen auf den durchschnittlichen Lagerbestand und ab einem Exponenten von 3 nie die Ergebnisse des Transformer-Modells.

Die Unterschiede zum Ergebnis in Abschnitt 8.1.2 können dadurch begründet werden, dass für die Produkte mit Dispositionsverfahren EX geringere Mengen für die Reduzierung zur Verfügung stehen. Zum Ausgleich der prozentualen Abweichung müssen somit mehrere Iterationen für das Verfahren gemäß Abschnitt 7.3 durchgeführt werden. Ab einem Exponenten von 3 kann dieser Unterschied nicht mehr ausgeglichen werden, womit das Transformer-Modell generell zu geringen Lagerbeständen in dem in Tabelle 8.2 dargestellten Bereich tendiert.

Im Vergleich zwischen ARIMA-Modell und Cockpit-Planung müssen in der Cockpit-Planung, unabhängig vom Exponenten, acht Monatsplanungen angepasst werden. Damit ist die Reduzierung des durchschnittlichen Lagerbestandes von 0.49 Prozent auf diese acht Überplanungen zurückzuführen. Für das CNN-LSTM-Modell sind keine Anpassung der Cockpit-Planung nötig. Der durchschnittliche Lagerbestand der Cockpit-Planung ist in diesem Vergleich für jeden Exponenten geringer.

Im Vergleich zwischen Transformer-Modell, dem ARIMA-Modell und dem CNN-LSTM-Modell, gemäß Abbildung 8.1.2 (unten), ist ersichtlich, dass das CNN-LSTM-Modell nie das Ergebnis des Transformer-Modells erreicht und konstant die maximale Anzahl an Reduzierungen (100) durchgeführt wird. Für das ARIMA-Modell tritt dieser Fall ab einem Exponenten von 5 und 92 Anpassungen ein.

8.2.2 Auswertung der Produkte mit fixer Periode (ZK)

Analog zu den Produkten mit Dispositionsverfahren *EX* werden die Primärbedarfe für die Produkte mit Dispositionsverfahren *ZK* anhand des Algorithmus 1 in Abschnitt 2.1.3.3 berechnet. Die Primärbedarfe der Produkte werden allerdings nicht täglich, sondern in Abhängigkeit einer festen Periode berechnet. Dieser Zeitraum wird über einen Planungkalender für jedes Produkt einzeln festgelegt. Die genau Berechnung des Prognosehorizontes für Produkte mit Dispositionsverfahren *ZK* ist in Abschnitt 3.3 ersichtlich.

8.2.2.1 Auswertung der Lieferfähigkeit

Der Vergleich der Lieferfähigkeiten für die Produkte mit Dispositionsverfahren ZK ist in Abbildung 8.1 ersichtlich. Mit 11.76 Prozent stellt das Transformer-Modell das Prognoseverfahren mit dem geringsten Anteil jener Produkte dar, welche die geforderte Lieferfähigkeit von 90 Prozent nicht erfüllen. Wie in Abschnitt 8.1.1 beschrieben, werden die geringen Lieferfähigkeiten des Transformer-Modells (minimal 60 Prozent) durch jene Produkte mit Produktionszyklen größer als zwei Wochen und geringen Sicherheitsbeständen verursacht.

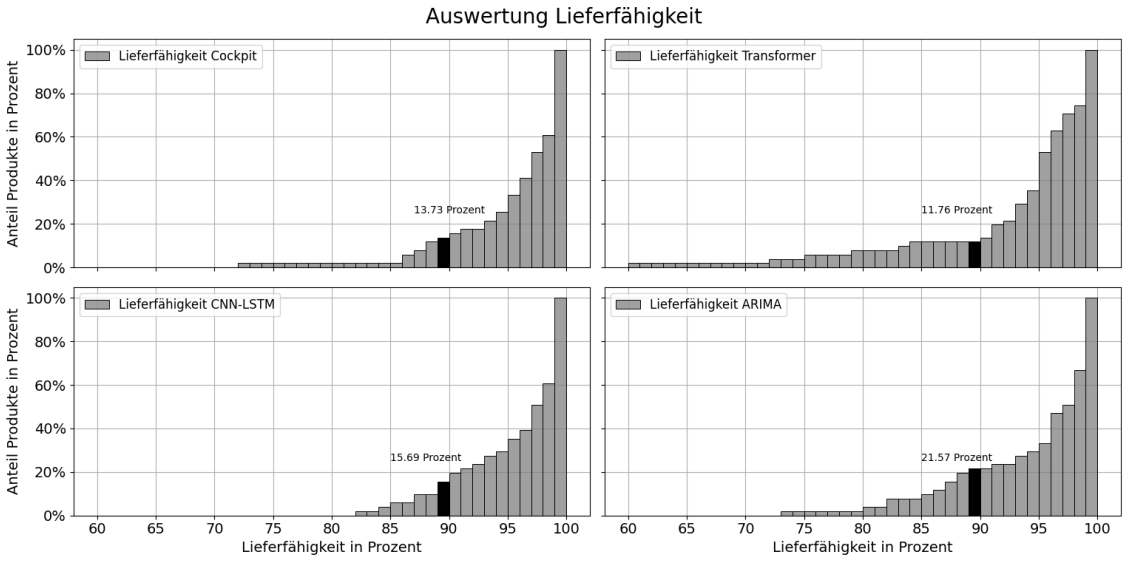


Abbildung 8.7: Lieferfähigkeit Dispositionsverfahren ZK

Der Anteil jener Produkte der Cockpit-Planung, welche die minimale Lieferfähigkeit nicht erreichen, ist um 1.97 Prozentpunkte höher als für das Transformer-Modell. Dieser Wert entspricht einer Differenz von einem Produkt.

In Bezug auf die Mindestanforderung der Lieferfähigkeit ist damit kein wesentlicher Vor- bzw. Nachteil im Vergleich zwischen Transformer-Modell und Cockpit-Planung gegeben. Unterschiede zeigen sich ab einer Lieferfähigkeit von 95 Prozent. Die Cockpit-Planung erzielt im Vergleich zum Transformer-Modell einen größeren Anteil an Produkten mit einer Lieferfähigkeit über diesem Prozentsatz. Der Anteil an Produkten mit einer Lieferfähigkeit größer als 95 Prozent beträgt bei der Cockpit-Planung 34 Produkte (66.67 Prozent). Beim Transformer-Modell liegen 24 Produkte (47.05 Prozent) über diesem Wert.

Im Vergleich zwischen den Ergebnissen des CNN-LSTM-Modells (Verteilung links unten) und der Cockpit-Planung sind keine wesentlichen Unterschiede ersichtlich. Das ARIMA-Modell stellt mit 21.57 Prozent das Prognoseverfahren mit dem größten Anteil an Produkten dar, welche die Lieferfähigkeit von 90 Prozent nicht erfüllen. Dieser Prozentsatz entspricht im Vergleich zum Transformer-Modell fünf zusätzlichen Produkten.

8.2.2.2 Auswertung des Lagerbestandes

Die Auswertung des durchschnittlichen Lagerbestandes pro Tag zeigt, dass analog zur Evaluierung des Gesamtbestandes in Abschnitt 8.1.2 das Transformer-Modell der einzige Modellansatz ist, durch den in Bezug auf die Cockpit-Planung eine Reduzierung erzielt werden kann. Diese liegt bei durchschnittliche 36 Paletten (0.62 Prozent), sodass keine wesentliche Veränderung des Lagerbestandes aufgrund des Transformer-Modells erzielt wird. Sowohl für das CNN-LSTM-Modell als auch für das ARIMA-Modell wird der durchschnittliche Lagerbestand gegenüber der Cockpit-Planung erhöht. Die Ergebnisse sind in Tabelle 8.3 ersichtlich.

Transformer vs. Cockpit	**ARIMA vs. Cockpit**	**CNN-LSTM vs. Cockpit**
-0.62 % (36 Paletten)	+1.07 % (51 Paletten)	+3.28 % (149 Paletten)

Tabelle 8.3: Abweichung Mittelwert Modelle vs. Cockpit (ZK)

Die Auswertung des durchschnittlichen Lagerbestandes gemäß Abbildung 8.8 (oben) zeigt, dass das Transformer-Modell in Bezug auf die Cockpit-Planung den Ansatz darstellt, für den die meisten Anpassungen benötigt werden. Die zunehmende Anzahl der benötigten Anpassungen bei höheren Werten für den Exponenten p zeigt, dass das Transformer-Modell generell zu geringeren Beständen in dem in Tabelle 8.3 beschriebenen Bereich neigt. Die Mengen der monatlichen Anpassungen bewegen sich im Bereich von 203 bis

108 Paletten.

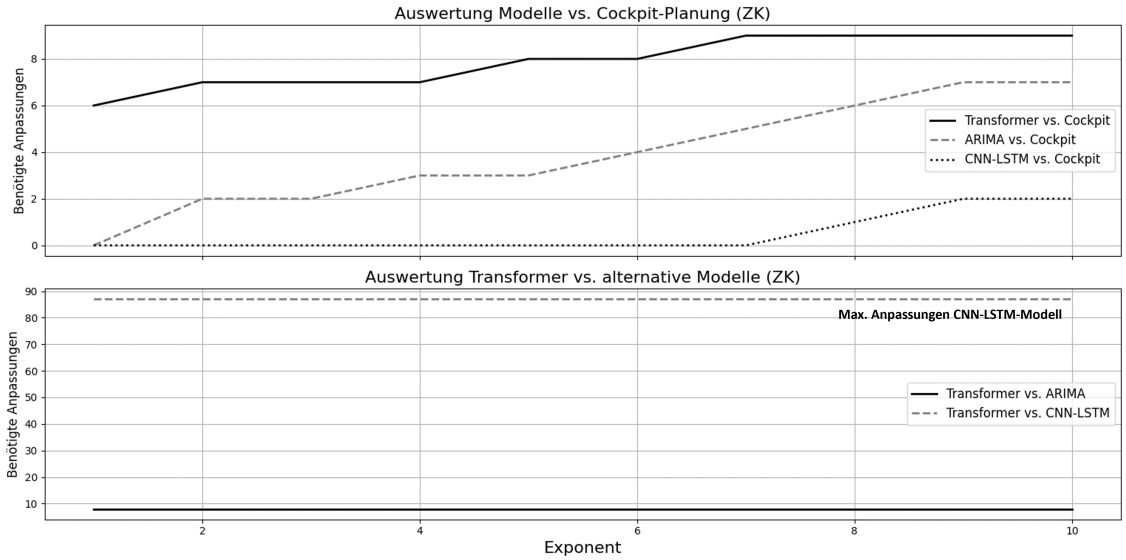


Abbildung 8.8: Auswertung Lagerbestand Dispositionsverfahren ZK

Der Vergleich zwischen ARIMA-Modell und Cockpit-Planung kann analog zum Transformer-Modell beurteilt werden. Allerdings wird für das ARIMA-Modell eine geringere Anzahl an Anpassungen benötigt. In Bezug auf das CNN-LSTM-Modell stellt die Evaluierung der Produkte nach Dispositionsverfahren ZK den einzigen Ansatz im Rahmen dieser Arbeit dar, für den eine Anpassung an der Cockpit-Planung durchgeführt werden muss. Diese erfolgt ab einem Exponenten von 8 und führt zu maximal zwei Reduzierungen der Monatsplanung.

Der Vergleich zwischen Transformer- und CNN-LSTM-Modell gemäß Abbildung 8.8 (unten) zeigt, dass konstant 87 Anpassungen durchgeführt werden. Dieser Wert entspricht der maximalen Anzahl an Reduzierungen ohne die Lieferfähigkeit zu beeinflussen. Damit kann das Ergebnis des Transformer-Modells durch das CNN-LSTM-Modell nicht erreicht werden.

Bezogen auf das ARIMA-Modell werden im Vergleich zum Transformer-Modell konstant acht Anpassungen durchgeführt. Damit ist der Unterschied im durchschnittlichen Mittelwert auf acht Überplanungen des ARIMA-Modells zurückzuführen, die sich im Bereich zwischen 470 und 170 Paletten bewegen.

Zur Beantwortung der Forschungsunterfrage 1 kann die gleiche Argumentation wie zur Beantwortung der Hauptforschungsfrage herangezogen werden. In Bezug auf den Lagerbestand kann durch die Prognose der Primärbedarfe mithilfe von neuronalen Netzen keine

wesentliche Reduzierung zur aktuellen Cockpit-Planung erzielt werden. Die Ergebnisse des Transformer-Modells für die Produkte mit Dispositionsverfahren EX und ZK zeigen allerdings das Potential einer automatischen bzw. unterstützenden Planung und damit einer Reduzierung der innerbetrieblichen Aufwände zur Erstellung der Prognosen. Unter Berücksichtigung der Ergebnisse in Abschnitt 8.2.1 sind diese Potentiale vor allem für die Produkte mit Dispositionsverfahren EX gebenden.

8.3 Auswertung auf Basis des Variationskoeffizienten

Neben der Evaluierung der Prognoseergebnisse aufgrund des verwendeten Dispositionsverfahrens, werden die im Rahmen dieser Arbeit angewandten Modell zusätzlich auf deren Potentiale bezüglich des Variationskoeffizienten der betrachteten Zeitreihen untersucht. Die genaue Formulierung der Forschungsfrage lautet:

Forschungsunterfrage 2

Welche Auswirkungen auf die Potentiale der Anwendung von neuronalen Netzen gegenüber der aktuellen Planung ergeben sich durch die Volatlität der betrachteten Zeitreihe?

Für die Auswertung werden die analysierten Produkte in zwei Gruppen geteilt. Alle Produkte mit einem Variationskoeffizienten ≤ 1.5 fallen hierbei in die Gruppe 1. Die restlichen Produkte werden der Gruppe 2 zugeordnet. Die Wahl der Grenze des Variationskoeffizienten erfolgte auf Basis annähend gleicher Gruppengrößen.

8.3.1 Auswertung der Produkte der Gruppe 1

Gruppe 1 enthält alle Produkte mit einem Variationskoeffizienten ≤ 1.5. Der Variationskoeffizient wurde wie in Abschnitt 4.4 beschrieben, anhand der wöchentlichen Kundenbedarfe je Produkt berechnet. Insgesamt wurden 58 der 123 analysierten Produkte dieser Kategorie zugeordnet.

8.3.1.1 Auswertung der Lieferfähigkeit

Bezogen auf die Lieferfähigkeit der Produkte der Gruppe 1 stellt das Transformer-Modell jenes Prognoseverfahren mit dem geringsten Anteil an Produkten dar, welche die Mindestanforderung von 90 Prozent nicht erfüllen. Im Vergleich mit der Cockpit-Planung beträgt der Unterschied zwei Produkte (3.45 Prozentpunkte). Allgemein zeigt der Vergleich dieser Verfahren keine wesentlichen Abweichungen, sodass diese Prognoseverfahren in Bezug auf die Lieferfähigkeit als gleichwertig angesehen werden.

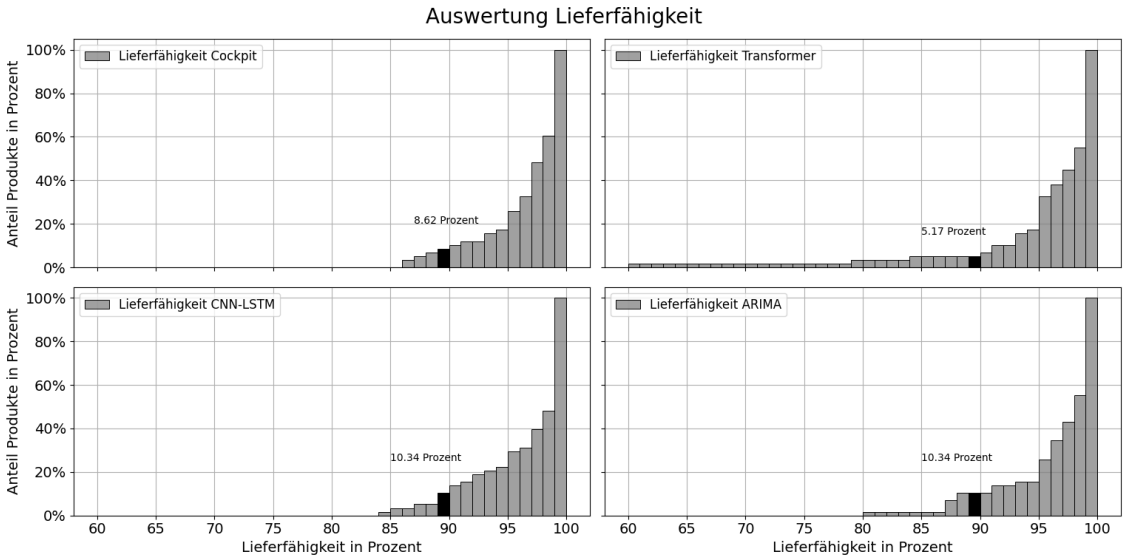


Abbildung 8.9: Lieferfähigkeit Gruppe 1

Im Vergleich zum Transformer-Modell wird die Anzahl der Produkte, welche die minimale Lieferfähigkeit nicht erfüllen, sowohl durch das CNN-LSTM-Modell als auch durch das ARIMA-Modell verdoppelt.

8.3.1.2 Auswertung des Lagerbestandes

Die Auswertung des durchschnittlichen Lagerbestandes zeigt, dass in Bezug auf die Cockpit-Planung nur durch das Transformer-Modell eine Reduzierung erreicht werden kann. Mit 54 Paletten (1.11 Prozent) ist diese Reduzierung in Bezug auf den Gesamtbestand allerdings nicht wesentlich, womit die Cockpit-Planung und das Transformer-Modell als gleichwertig angesehen werden können.

Die gleiche Beurteilung in Bezug auf die Cockpit-Planung, kann für das ARIMA-Modell herangezogen werden, das den durchschnittlichen Bestand um 51 Paletten (1.09 Prozent) pro Tag erhöht. Mit einer Erhöhung von 280 Paletten (4.75 Prozent), erzielt das CNN-LSTM-Modell das schlechteste Ergebnis in Bezug auf den durchschnittlichen Lagerbestand. Die Auswertung ist in Tabelle 8.5 ersichtlich.

Transformer vs. Cockpit	**ARIMA vs. Cockpit**	**CNN-LSTM vs. Cockpit**
-1.11 % (54 Paletten)	+1.09 % (51 Paletten)	+4.74 % (280 Paletten)

Tabelle 8.4: Abweichung Mittelwert Modelle vs. Cockpit (Gruppe 1)

Abbildung 8.10 (oben) zeigt die benötigten Anpassungen der Cockpit-Planung im Vergleich zu den Prognosemodellen. Im Vergleich zwischen Transformer-Modell und Cockpit-Planung wird mit einem zunehmenden Exponenten p die Anzahl der benötigten Anpassungen erhöht. Somit tendiert das Transformer-Modell allgemein zu geringeren Lagerbeständen. Die Anpassungen bewegen sich im Bereich von 203 bis 108 Paletten.

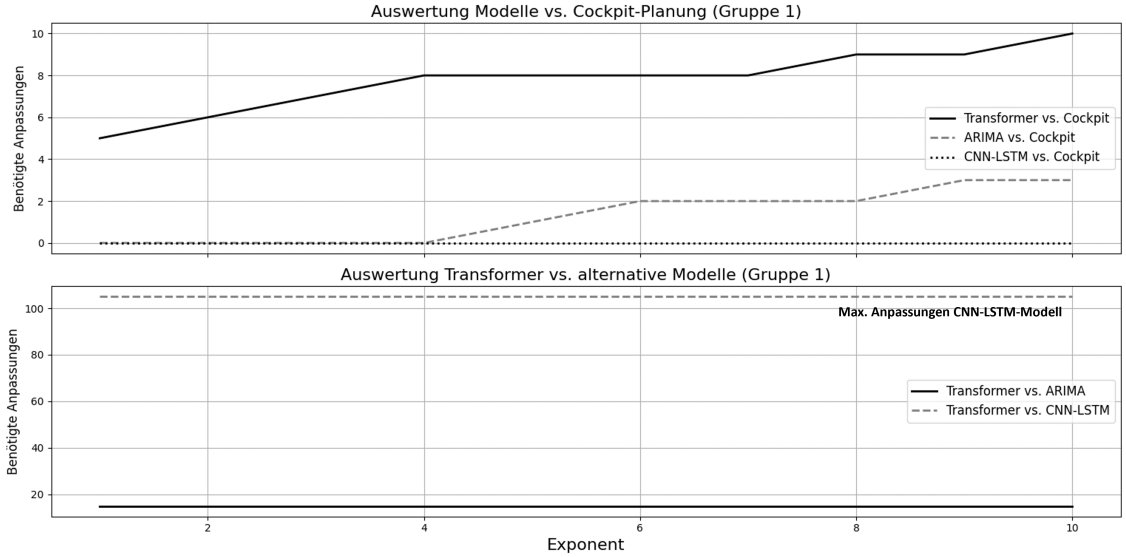


Abbildung 8.10: Auswertung Lagerbestand Gruppe 1

Beim CNN-LSTM wurden keine Anpassungen der Cockpit-Planung benötigt. Die Ergebnisse des ARIMA-Modells erfordern maximal drei Anpassungen beginnend ab einem Exponenten von 5. Damit stellt das Transformer-Modell jenes Prognoseverfahren dar, das in Bezug auf die Senkung des Lagerbestandes das größte Potential aufweist.

Dieses Ergebnis ist auch im direkten Vergleich mit dem ARIMA-Modell, Abbildung 8.10 (unten), ersichtlich. Unabhängig vom Exponenten müssen für das ARIMA-Modell konstant 16 Anpassungen im Bereich von 470 bis 87 Paletten durchgeführt werden. Das CNN-LSTM-Modell kann das Ergebnis des Transformer-Modells ohne eine Reduzierung der Lieferfähigkeit nicht erreichen.

8.3.2 Auswertung der Produkte der Gruppe 2

Gruppe 2 enthält alle Produkte mit einem Variationskoeffizienten größer 1.5. Insgesamt wurden 65 Produkte dieser Gruppe zugeordnet.

8.3.2.1 Auswertung der Lieferfähigkeit

Die Auswertung der Lieferfähigkeit für die Gruppe 2 ist in Abbildung 8.11 dargestellt. Der Vergleich zwischen Transformer-Modell (Verteilung rechts oben) und Cockpit-Planung (Verteilung links oben) zeigt, dass diese Evaluierung der einzige durchgeführte Ansatz ist, bei dem der Anteil der Produkte, welche die Lieferfähigkeit von 90 Prozent nicht erfüllen, für die Cockpit-Planung geringer ist. Mit 1.54 Prozentpunkten, oder einem Produkt, ist dieser Unterschied allerdings nicht wesentlich in der Beurteilung der Vor- und Nachteile zwischen den Modellen.

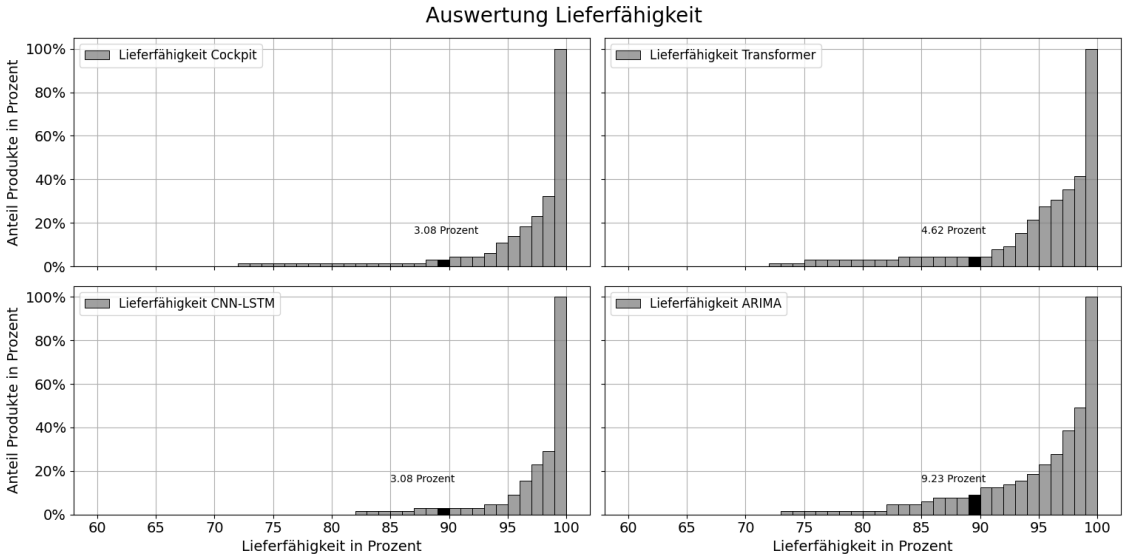


Abbildung 8.11: Lieferfähigkeit Gruppe 2

Die Lieferfähigkeiten des CNN-LSTM (Verteilung links unten) zeigen keinen wesentlichen Unterschied zur Cockpit-Planung. Das ARIMA-Modell stellt mit sechs Produkten jenes Verfahren dar, für das die Lieferfähigkeit von 90 Prozent am häufigsten nicht erreicht werden kann. Im Vergleich zur Cockpit-Planung entspricht dieser Wert einer Differenz von vier Produkten.

8.3.2.2 Auswertung des Lagerbestandes

Die Auswertungen der prozentualen Abweichung des durchschnittlichen Lagerbestandes gemäß Tabelle 8.5 zeigt, dass für die Produkte der Gruppe 2 sowohl durch das Transformer-Modell als auch durch das ARIMA-Modell eine Reduzierung des durchschnittlichen Lagerbestandes erreicht werden kann. Mit 14 Paletten (2.00 Prozent) für das Transformer-Modell und 11 Paletten (1.57 Prozent) für das ARIMA-Modell sind diese Reduzierungen in Bezug auf die Cockpit-Planung allerdings nicht wesentlich.
Die Evaluierung des CNN-LSTM-Modells zeigt eine Erhöhung des durchschnittlichen Lagerbestandes um 72 Paletten (10.30 Prozent). Im Vergleich mit den Produkten der Gruppe 1 werden durch die Produkte der Gruppe 2 allgemein geringere Palettenmengen erzeugt.

Transformer vs. Cockpit	**ARIMA vs. Cockpit**	**CNN-LSTM vs. Cockpit**
-2.00 % (14 Paletten)	-1.57 % (11 Paletten)	+10.30 % (72 Paletten)

Tabelle 8.5: Abweichung Mittelwert Modelle vs. Cockpit (Gruppe 2)

Die Auswertung auf Basis der benötigten Anpassungen für einen Exponenten p zeigt, dass die Cockpit-Planung sowohl die Ergebnisse des Transformer-Modells als auch die Ergebnisse des ARIMA-Modells ohne eine Reduzierung der Lieferfähigkeiten nicht erreicht. Unabhängig vom Exponenten p werden für die Cockpit-Planung konstant 80 Anpassungen und damit die maximal mögliche Anzahl an Reduzierungen benötigt. Für das CNN-LSTM-Modell muss keine Anpassung der Cockpit-Planung durchgeführt werden. Die Auswertung ist in Abbildung 8.12 (oben) ersichtlich.

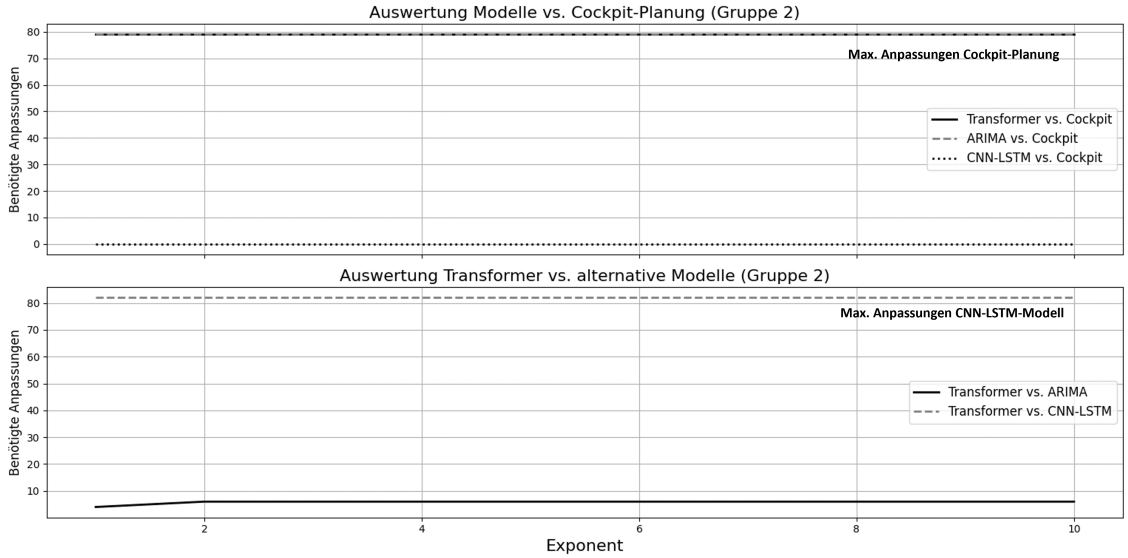


Abbildung 8.12: Auswertung Gruppe 2

Der Vergleich zwischen Transformer-, ARIMA- und CNN-LSTM-Modell, Abbildung 8.12 (unten) zeigt, dass die Abweichung zwischen Transformer- und ARIMA-Modell auf maximal sieben Anpassungen zurückgeführt werden kann. Diese bewegen sich im Bereich von 21 bis 10 Paletten, womit diese Modelle für die Produkte der Gruppe 2 und in Bezug auf den durchschnittlichen Lagerbestand als gleichwertig beurteilt werden können. Für das CNN-LSTM-Modell wurde unabhängig vom Exponenten p, die maximal mögliche Anzahl an Anpassungen (82) durchgeführt. Das Ergebniss des Transformer-Modells bezüglich des durchschnittlichen Lagerbestands konnte durch das CNN-LSTM Modell nicht ohne eine Reduzierung der Lieferfähigkeit erreicht werden.

Basierend auf den Ergebnissen in Abschnitt 8.3 kann für die Forschungsunterfrage 2 das gleiche Ergebnis wie für die Hauptforschungsfrage und die Forschungsunterfrage 1 abgeleitet werden. Die Verwendung von neuronalen Netzen zur Prognose der Primärbedarfe zeigt unabhängig vom Variationskoeffizienten der betrachteten Zeitreihen keine wesentlichen Potentiale zur Senkung des Lagerbestandes.

Das Transformer-Modell is allerdings ein Prognoseverfahren, das auch in Bezug auf den Variationskoeffizienten das Potential einer automatisierten bzw. unterstützenden Planung in Bezug auf die aktuelle Cockpit-Planung aufweist, ohne die Lieferfähigkeit zu reduzieren oder den Lagerbestand zu erhöhen. Damit bietet dieses Modell auch innerhalb der Forschungsunterfrage 2 die Möglichkeit der Reduzierung des innerbetrieblichen Aufwandes zur Erstellung der Prognosen der Primärbedarfe.

8.4 Einfluss der Stammdatenqualität auf die Prognoseergebnisse

Der nachfolgende Abschnitt zeigt die Auswirkungen fehlerhafter Stammdaten für jene Produkte, die aufgrund ihrer Häufigkeit im Verbrauch als A Teile klassifiziert wurden. Als Produkte mit fehlerhaften Stammdaten wurden jene Fertigteile klassifiziert, die mit dem Dispositionsverfahren EX (fixe Losgröße) geplant werden und keine Losgröße am Materialstamm enthalten oder für die keine Berechnung des Sicherheitsbestandes hinterlegt wurde.

8.4.1 Auswertung des Lagerbestandes

Der Einfluss der Stammdatenqualität auf die Lieferfähigkeit ist in Abbildung 8.13 ersichtlich. Unabhängig vom Prognoseverfahren stellt die Stammdatenqualität ein wesentliches Kriterium in der Anwendung der Prognosen für die Materialbedarfsplanung dar.

In Bezug auf die Cockpit-Planung kann die geforderte Lieferfähigkeit von 90 Prozent für 83.33 Prozent der analysierten Produkte nicht erreicht werden. Für das Transformer- bzw. CNN-LSTM-Modell erfüllen 75 Prozent der analysierten Produkte dieses Mindestkriterium nicht. Mit 66.67 Prozent zeigt das ARIMA-Modell den geringsten Anteil der Produkte mit einer Lieferfähigkeit unter 90 Prozent.

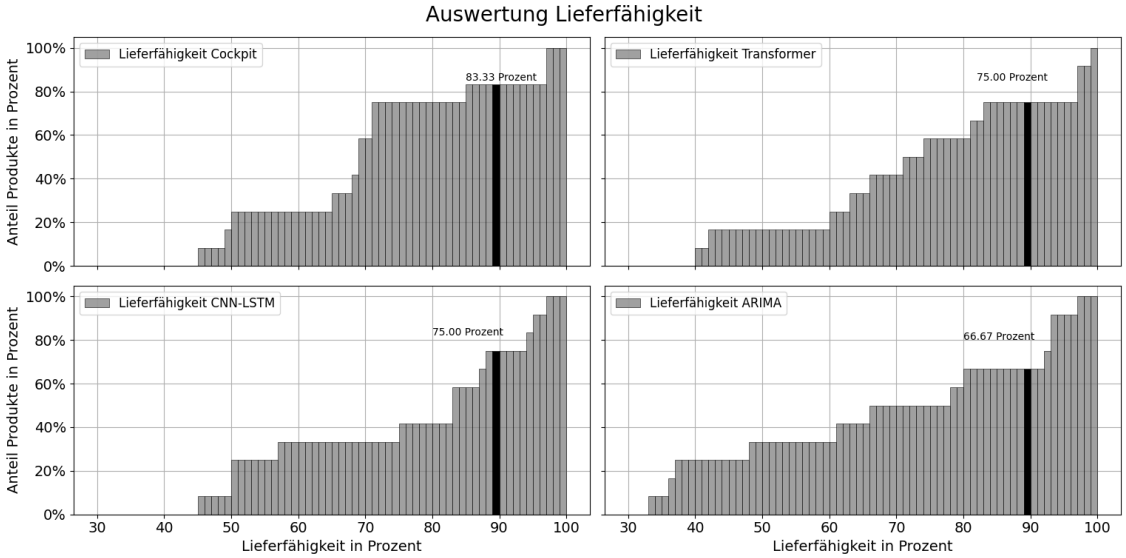


Abbildung 8.13: Auswertung Lieferfähigkeit Stammdaten

Die Analyse der tatsächlichen Lieferfähigkeit zeigt einen Wert von 96.10 Prozent. Um diese Lieferfähigkeit zu erreichen, musste die Produktionsmenge durch die Feinplanung manuell angepasst werden oder es wurden Planaufträge unabhängig von den Ergebnissen der Materialbedarfsplanung manuell erzeugt. Die unvollständigen Stammdaten führen daher zu regelmäßigen Zusatzaufwänden in der Planung der Primärbedarfe.

Wie in Abschnitt 2.1.2.5.7 beschrieben wurde, ist bezogen auf alle verfügbaren Produkte und unabhängig von der ABC-XYZ-Klassifizierung, für 65.9 Prozent der Produkte kein Sicherheitsbestand gepflegt. Die Auswirkungen auf B- bzw. C-klassifizierte Produkte wurden im Rahmen dieser Arbeit nicht untersucht und bieten das Potential weiterer Analysen. Alternativ zu einer rein plangesteuerten Disposition der Produkte können im Rahmen dieser Forschungen beispielsweise die Auswirkungen von verbrauchsgesteuerten Verfahren untersucht werden.

Die Auswirkungen auf den Lagerbestand wurden für die Produkte mit fehlerhaften Stammdaten nicht analysiert. Durch die hohe Schwankungsbreite der Lieferfähigkeiten unter 90 Prozent zwischen den einzelnen Modellen und der niedrigen Lagerbestände aufgrund dieser geringen Lieferfähigkeiten, ist eine Beurteilung basierend auf dem durchschnittlichen Lagerbestand für die Produkte mit fehlerhaften Stammdaten nicht aussagekräftig.

9 Zusammenfassung und Ausblick

Die Ergebnisse der Evaluierung zeigen, dass durch die Verwendung von neuronalen Netzen zur Prognose der Primärbedarfe in der Materialbedarfsplanung keine wesentliche Reduzierung des durchschnittlichen Lagerbestandes gegenüber der aktuellen Dispositionsplanung erzielt werden kann.

Das Transformer-Modell zeigt allerdings das Potential einer unterstützenden bzw. automatisierten Planung der A/Z-klassifizierten Produkte, ohne dass eine Reduzierung der Lieferfähigkeit bzw. eine Erhöhung des täglichen Lagerbestandes entstehen. In diesen Kategorien konnten durch das Transformer-Modell geringfügige Verbesserungen im Bereich von einem Prozent für das Kalenderjahr 2021 erzielt werden.

Bezogen auf die verwendeten Dispositionsverfahren in der Materialbedarfsplanung ist dieses Potential vor allem für jene Produkte gegeben, die mit einer fixen Losgröße geplant werden. Im Vergleich zur aktuellen Dispositionsplanung werden für diese Produkte durch das Transformer-Modell kontinuierlich geringere Lagerbestände erzeugt.

In der Beurteilung mit Bezug auf den Variationskoeffizienten der betrachteten Zeitreihen zeigt das Transformer Modell sowohl für die definierte Gruppe 1 (Variationskoeffizient $\leq$ 1.5) als auch für die Gruppe 2 (Variationskoeffizient $>$ 1.5) Potentiale für eine unterstützende bzw. automatisierte Planung. Vor allem in der Gruppe 1 kann für das Kalenderjahr 2021 der Anteil jener Produkte, welche die Mindestanforderung von 90 Prozent Lieferfähigkeit unterschreiten, gegenüber der aktuellen Dispositionsplanung gesenkt werden.

Basierend auf den Ergebnissen der Evaluierung bietet das im Rahmen dieser Arbeit angewandte Transformer-Modell das Potential für weitere Forschungsarbeiten. Die Erweiterung der Inputdaten um zusätzliche saisonale Zeitpunkte bietet beispielsweise die Möglichkeit einer Verbesserung der Prognosergebnisse. Zusätzlich kann der naive Forecast im Decoder durch alternative Prognoseverfahren ersetzt werden um die Qualität der Ergebnisse zu erhöhen. Eine detaillierte Analyse der Auswirkungen der einzelnen Modelle auf die maximalen Lagerbestände stellt zudem einen weiteren möglichen Ansatz für zukünftige Forschungsarbeiten dar.

Literaturverzeichnis

Buchquellen

Aladag, C. H., Egrioglu, E. & Kadilar, C. (2009). Forecasting nonlinear time series with a hybrid methodology. *Applied Mathematics Letters*, *22*(9), 1467–1470. https://doi.org/10.1016/j.aml.2009.02.006

Arnold, D., Isermann, H., Kuhn, A., Tempelmeier, H. & Furmans, K. (2008). *Handbuch Logistik* -. Springer Berlin Heidelberg

Avci, O. (2019). *A study to examine the importance of forecast accuracy to supply chain performance: A case study at a company from the FMCG industry* (Diss.). https://doi.org/10.13140/RG.2.2.16814.95048

Bhanja, S. & Das, A. (2018). Impact of Data Normalization on Deep Neural Network for Time Series Forecasting

Bontempi, G., Ben Taieb, S. & Le Borgne, Y.-A. (2013). Machine Learning Strategies for Time Series Forecasting. https://doi.org/10.1007/978-3-642-36318-4_3

Bourier, G. (2010). *Statistik-Übungen - Beschreibende Statistik - Wahrscheinlichkeitsrechnung - Schließende Statistik*. Springer-Verlag

Cerqueira, V., Torgo, L. & Mozetic, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Mach. Learn.*, *109*, 1997–2028

Cleveland, R. B., Cleveland, W. S., McRae, J. E. & Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion). *Journal of Official Statistics*, *6*, 3–73

Dhaheri, K., Woon, W. & Aung, Z. (2017). Wind Speed Forecasting Using Statistical and Machine Learning Methods: A Case Study in the UAE, 107–120. https://doi.org/10.1007/978-3-319-71643-5_10

Dickersbach, J. T. & Keller, G. (2014). *Produktionsplanung und -steuerung mit SAP ERP - Ihr umfassendes Handbuch zu SAP PP - 5. Auflage*. Rheinwerk Verlag GmbH

Dickey, D. & Fuller, W. (1979). Distribution of the Estimators for Autoregressive Time Series With a Unit Root. *JASA. Journal of the American Statistical Association*, *74*. https://doi.org/10.2307/2286348

Duchi, J., Hazan, E. & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*, 2121–2159

Francq, C. & Zakoïan, J.-M. (2009). Bartlett's formula for a general class of Nonlinear Processes. *Journal of Time Series Analysis*, *30*(4), 449–465. https://doi.org/10.1111/j.1467-9892.2009.00623.x

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media

Gudehus, T. (2011). *Dynamische Disposition - Strategien, Algorithmen und Werkzeuge zur optimalen Auftrags-, Bestands- und Fertigungsdisposition*. Springer Berlin Heidelberg

Gulyássy, F., Hoppe, M., Köhler, O. & Vithayathil, B. (2014). *Disposition mit SAP - Funktionen und Customizing in SAP ERP und SAP SCM (SAP APO)*. Rheinwerk Verlag GmbH

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Heiserich, O.-E., Helbig, K. & Ullmann, W. (2011). *Logistik - Eine praxisorientierte Einführung*. Springer-Verlag

Hillier, F. S. & Liebermann, G. J. (2014). *Operations Research - Einführung*. Walter de Gruyter GmbH & Co KG

Jodlbauer, H. (2008). *Produktionsoptimierung - Wertschaffende sowie kundenorientierte Planung und Steuerung*. Springer Vienna

Kappauf, J., Koch, M. & Lauterbach, B. (2017). *Logistik mit SAP - Der umfassende Einstieg. Inkl. Einführung in SAP S/4HANA. Ausführlicher Überblick über Funktionen und Prozesse. Zusammenhänge und Unterschiede der SAP-Logistiklösungen verstehen. Mit anschaulichen Praxisbeispielen*. Rheinwerk Verlag GmbH

Kingma, D. & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*

Kline, D. (2004). Methods for Multi-Step Time Series Forecasting with Neural Networks. https://doi.org/10.4018/978-1-59140-176-6.ch012

Kokoszka, P. & Young, G. (2016). KPSS test for functional time series. *Statistics*, *50*, 1–17. https://doi.org/10.1080/02331888.2015.1128937

Koller, W. (2014). Prognose makroökonomischer Zeitreihen: Ein Vergleich linearer Modelle mit neuronalen Netzen. https://doi.org/10.3726/978-3-653-03344-1

Kreiss, J.-P. & Neuhaus, G. (2006). *Einführung in die Zeitreihenanalyse* -. Springer Berlin Heidelberg

Kriesel, D. (2007). *Ein kleiner Überblick über Neuronale Netze*. erh%5C%22%7Ba%7Dltlich%20auf%20http://www.dkriesel.com

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X. & Yan, X. (2019). Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting

Lim, B., Arık, S., Loeff, N. & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*. https://doi.org/10.1016/j.ijforecast.2021.03.012

Nielsen, A. (2019). *Practical Time Series Analysis: Prediction with Statistics and Machine Learning*. O'Reilly Media. https://books.google.at/books?id=odCwDwAAQBAJ

Nyhuis, P. & Wiendahl, H.-P. (2013). *Logistische Kennlinien - Grundlagen, Werkzeuge und Anwendungen*. Springer-Verlag

Panigrahi, S. S. & Behera, H. S. (2013). Effect of Normalization Techniques on Univariate Time Series Forecasting using Evolutionary Higher Order Neural Network

Patàk, M. & Vlckova, V. (2014). Demand Planning Specifics In Food Industry Enterprises. https://doi.org/10.3846/bm.2012.150

Proietti, T. & Lütkepohl, H. (2013). Does the Box–Cox transformation help in forecasting macroeconomic time series? *International Journal of Forecasting*, *29*(1), 88–99. https://doi.org/https://doi.org/10.1016/j.ijforecast.2012.06.001

Schönsleben, P. (2007). *Integrales Logistikmanagement - Operations and Supply Chain Management in umfassenden Wertschöpfungsnetzwerken*. Springer-Verlag

Thakur, A. (2020). *Approaching (almost) any machine learning problem*. Thakur, Abhishek (Independently published)

Toth, E. (2015). Estimation of flood warning runoff thresholds in ungauged basins with asymmetric error functions. *Hydrology and Earth System Sciences Discussions*, *12*, 6011–6041. https://doi.org/10.5194/hessd-12-6011-2015

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Hrsg.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. https : / / proceedings . neurips . cc / paper / 2017 / file / 3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Veiga, C., Veiga, C. & Duclós, L. (2010). THE ACCURACY OF DEMAND FORECAST MODELS AS A CRITICAL FACTOR IN THE FINANCIAL PERFORMANCE OF THE FOOD INDUSTRY. *Future Studies Research Journal: Trends and Strategies*, *2*, 81–104

Xu, Y. & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating

the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, *2*. https://doi.org/10.1007/s41664-018-0068-2

Zsifkovits, H. E. (2012). *Logistik*. UTB GmbH