Vorhersagen von räumlich korrelierten epidemiologischen Zeitreihen mittels Methoden der Statistik und des maschinellen Lernens

Forecasting spatially correlated epidemiological time series using statistical and machine learning methods

Master-Thesis von Fabian Hammes aus Enkirch

Tag der Einreichung:

1. Gutachten: Prof. Dr. techn. Johannes Fürnkranz
2. Gutachten: Dr. Eneldo Loza Mencía

TECHNISCHE UNIVERSITÄT DARMSTADT

Knowledge Engineering Group
Fachbereich Informatik

Vorhersagen von räumlich korrelierten epidemiologischen Zeitreihen mittels Methoden der Statistik und des maschinellen Lernens
Forecasting spatially correlated epidemiological time series using statistical and machine learning methods

Vorgelegte Master-Thesis von Fabian Hammes aus Enkirch

1. Gutachten: Prof. Dr. techn. Johannes Fürnkranz
2. Gutachten: Dr. Eneldo Loza Mencía

Tag der Einreichung:

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, Fabian Hammes, die vorliegende Master-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

English translation for information purposes only:

Thesis Statement pursuant to § 22 paragraph 7 and § 23 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I, Fabian Hammes, have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.

I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.

In the submitted thesis the written copies and the electronic version for archiving are pursuant to § 23 paragraph 7 of APB identical in content.

For a thesis of the Department of Architecture, the submitted electronic version corresponds to the presented model and the submitted architectural plans.

Darmstadt, den 09.05.2019

(Fabian Hammes)

Zusammenfassung

Die folgende Arbeit beschäftigt sich mit der Vorhersage von Zeitreihen der Fallzahlen von Krankheiten in Frankfurt am Main. Zu Beginn werden diesbezüglich einige renommierte Vorhersage-Modelle für Zeitreihen vorgestellt. Des Weiteren werden einige Verfahren erläutert, mit deren Hilfe, durch Anpassung oder Zerlegung der Zeitreihe, eine genauere Vorhersage ermöglicht wird. Daraufhin werden bereits existierende Strategien zur Evaluation präsentiert, mit welchen bestimmt werden kann, welches Modell sich am besten für die Vorhersage einer Zeitreihe eignet. Anschließend wird eine Strategie ausgearbeitet, welche zusätzlich ein passendes Bearbeitungs-Verfahren für die gegebene Zeitreihe auswählt, um den Fehler innerhalb der Vorhersage zu minimieren. In einem weiteren Schritt werden einige Vorhersage-Modelle erweitert, sodass diese ebenfalls epidemiologische Informationen aus der regionalen Umgebung von Frankfurt am Main verarbeiten können. Abschließend wird zu den Krankheiten Campylobacter-Enteritis und Keuchhusten eine ausführliche Analyse durchgeführt. Verglichen werden die normalen Vorhersagen der vorgestellten Modelle mit denen der ausgearbeiteten Strategien inklusive, sowohl auch ohne, der zusätzlichen Informationen aus der Umgebung. Die resultierenden Ergebnisse werden anschließend diskutiert und bewertet.

Abstract

The following work deals with the forecast of time series of counts from disease-cases in Frankfurt am Main. At the beginning, some state-of-the-art models to forecast a time series are presented. In addition, some methods are explained with the help of which a more accurate prediction can be made by transforming or decomposing the time series. After that, existing evaluation strategies are presented to determine which model is most appropriate for predicting a time series. A strategy is then developed, which additionally selects a suitable transformation and decomposition method for the given time series in order to minimize the error within the prediction. In a further step, some prediction models will be extended so that they can also process epidemiological information from the regional surroundings of Frankfurt am Main. Finally, a detailed analysis of the diseases Campylobacter enteritis and whooping cough will be performed. The normal predictions of the presented models will be compared with those of the elaborated strategies, including and excluding the additional information from the surroundings of Frankfurt am Main. The results will then be discussed and evaluated.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Jede Woche erkranken Menschen an den unterschiedlichsten Krankheiten. Krankenhäuser und Ärzte sind daher mit einem Grundvorrat von Medikamenten gegen die regional üblichen Krankheiten ausgestattet. Doch was passiert, wenn eine ansteckende Krankheit untypisch viele Krankheitsfälle in einer bestimmten Region aufweist? In diesem Fall würde von einer Epidemie, oder auch Seuche, gesprochen werden. Das Robert Koch-Institut (RKI), ein Bundesinstitut für Infektionskrankheiten und nicht übertragbare Krankheiten, setzt eine Epidemie mit einem starken Ausbruch der Krankheit gleich [Kiehl 2015]. Eine Epidemie muss eingedämmt werden, da sich die Krankheit ansonsten immer weiter ausbreiten würde. Um dies realisieren zu können, muss eine überdurchschnittliche Ausbreitung frühzeitig erkannt werden. Falls eine Krankheit jedoch stark ausbrechen sollte und zu spät erkannt werden würde, kann es dazu kommen, dass die zuvor genannten Grundvorräte nicht mehr ausreichen. In dem folgenden Unterkapitel wird erläutert, wie die vorliegende Arbeit dabei helfen soll, diesen Fall vermeiden zu können.

1.1 Thematik der Arbeit

Diese Arbeit ist ein Teilprojekt aus einem Projekt[1] des Fachbereichs Knowledge Engineering der TU Darmstadt in Kooperation mit dem Gesundheitsamt Frankfurt, dem RKI und einigen weiteren Partnern aus dem Gesundheitswesen. Das Ziel besteht darin, ein vermehrtes Auftreten von Infektionserkrankungen frühzeitig regional erkennen zu können. Diesbezüglich soll ein Frühwarnsystem, welches vom RKI entwickelt wird, durch Methoden des maschinellen Lernens unterstützt werden. Mit Hilfe von Datenanalyse-Methoden sollen die Möglichkeiten zur Frühwarnung verbessert werden. Die Aufgabe der vorliegenden Arbeit besteht darin, die Fallzahl von Krankheiten in Frankfurt am Main (FFM) bestmöglich vorherzusagen, um zum Beispiel frühzeitig auf etwaige Abweichungen reagieren zu können. Ebenfalls kann diese Vorhersage wiederum für weitere Teilprojekte verwendet werden. Die Basis der Vorhersage beruht auf den vergangenen Fallzahlen der Krankheit in FFM und der regionalen Umgebung.

Um eine bestmögliche Vorhersage zu produzieren, werden als Erstes einige Vorhersage-Modelle ausgewählt, welche den höchsten Erfolg versprechen und am besten auf die Aufgabe zugeschnitten sind. Ergänzend werden Verfahren ausgearbeitet, welche die Zeitreihen im Vorhinein anpassen, sodass die Ergebnisse der Vorhersage optimiert werden. Nachdem dieser Grundbaustein gelegt ist, wird eine Evaluations-Strategie entwickelt, welche aus den gesammelten Verfahren und Modellen die Vorhersagen erstellen, bewerten und die Beste auswählen soll. Diese Strategie wird darauf spezialisiert, die Fallzahlen einer Krankheit der nächsten Woche zu bestimmen. Außerdem wird überprüft, ob die Krankheitsfallzahlen der Land- und Stadtkreise in der Umgebung von FFM möglicherweise Korrelationen zu den Fallzahlen von FFM aufweisen, welche die Vorhersage verbessern könnten. Die Vermutung ist, dass Steigungen in den Zeitreihen der Nachbar-Kreise ein Indiz dafür sind, dass die Fallzahlen in FFM ebenfalls ansteigen. Der Grund dafür besteht darin, dass die Krankheiten sich regional ausbreiten. Wie sich in der Auswertung zeigen wird, haben diese zusätzlichen Informationen einen sehr positiven Effekt auf die Ergebnisse der Vorhersage.

[1] Projektbeschreibung: `https://innovationsfonds.g-ba.de/projekte/versorgungsforschung/eseg-erkennung-und-steuerung-epidemiologischer-gefahrenlagen.151`

1.2 Aufbau

Im zweiten Kapitel werden zunächst alle nötigen Grundlagen erläutert, welche zum Verständnis dieser Arbeit nötig sind. Des Weiteren werden Methoden und Modelle nähergebracht, welche zur Vorhersage sowie deren Optimierung verwendet werden. Abschließend werden in diesem Kapitel einige Strategien zur Evaluation der Ergebnisse einer Vorhersage beschrieben. Mit dem Wissen der Grundlagen werden im nächsten Kapitel einige Verfahren zur Vorhersage optimiert. Daraufhin wird eine Vorgehensweise ausgearbeitet, mit der die bestmögliche Vorhersage des nächsten Zeitpunktes für eine gegebene Zeitreihe erstellt werden soll. Das Kapitel Auswertung beinhaltet eine ausführliche Analyse der Ergebnisse der zuvor erarbeiteten Algorithmen und Strategien. Abschließend folgt das Resümee, in dem auf diese Arbeit eingegangen wird.

2 Theoretische Grundlagen

In diesem Kapitel werden alle Methoden erläutert und Grundlagen vermittelt, welche zum Verständis der folgenden Kapitel notwendig sind. Es wird auf die zur Verfügung stehenden Daten, das benötigte Wissen von Zeitreihen und dessen Vorhersage sowie auf Evaluations-Strategien eingegangen.

2.1 Datengrundlage

Die Arbeit beschäftigt sich mit der Analyse von Fallzahlen meldepflichtiger Krankheiten. Die zugehörigen Daten werden vom Robert Koch-Institut im sogenannten Infektionsepidemiologischen Jahrbuch[2] bereitgestellt. Diese Daten werden bundesweit einheitlich erfasst und können in der frei zugänglichen Web-Applikation *SurvStat*[3] des RKI abgefragt werden. Das RKI unterscheidet zwischen fünf Falldefinitionskategorien, welche jeweils unterschiedliche Evidenztypen erfüllen. Die Evidenztypen teilen sich auf in das klinische Bild, den labordiagnostischen Nachweis und die epidemiologische Bestätigung. Das *klinische Bild* beschreibt die Symptome und klinischen Zeichen, welche für die jeweilige Krankheit erfüllt sein müssen. Der *labordiagnostische Nachweis* ist erfüllt, wenn vorher definierte Materialien und Labormethoden für den Erregernachweis verwendet wurden. Die *epidemiologischen Bestätigung* setzt ein erfülltes klinisches Bild voraus. Eine solche Bestätigung liegt vor, wenn das Bild mit einem labordiagnostisch nachgewiesenen Fall in einen epidemiologischen Zusammenhang gebracht wurde. Da eine gewisse Zeit vergeht, bis einem Fall seine endgültige Falldefinitionskategorie zugewiesen wurde, werden in dieser Arbeit, wenn nicht anders erwähnt, die Gesamtzahl der Fälle betrachtet. Aufgrund dieser Tatsache werden die Falldefinitionskategorie hier nicht weiter erläutert.
Um die Daten ausführlich analysieren zu können, wird zusätzlich zu einem Fall der Ort des Auftretens, das Alter und Geschlecht des Patienten aufgezeichnet. Des Weiteren werden das Jahr sowie die Kalenderwoche, in welcher das Gesundheitsamt offiziell Kenntnis von dem Fall erlangt, notiert. Der Ort des Auftretens wird als Kombination von Bundesland und zugehörigem Landkreis (LK) oder Stadtkreis (SK) dargestellt. Mit diesen Informationen können die Fälle zusätzlich nach Geschlecht, Alters-Intervallen und Ort gruppiert werden. Zum Vergleich der Fallzahlen von verschiedenen Gruppen wird der Wert *Inzidenz* genutzt. Dieser Wert erstellt eine Relation zwischen der Größe der betrachteten Menge an Menschen und Anzahl der Fallzahlen in dieser Menge. Die Inzidenz wird in Fallzahlen pro 100.000 Personen der gegebenen Gruppe angegeben.

Krankheiten

Die Arbeit beschäftigt sich im Speziellen mit zwei bestimmten Krankheiten. Die folgenden Informationen zu den Erkrankungen stammen aus zwei Artikeln des RKI. Diese werden jeweils am Anfang des betreffenden Abschnittes referenziert.
Die erste Krankheit, Campylobacter-Enteritis[4] (Campylobacter), ist eine Durchfallerkrankung, gekoppelt mit Fieber und Kopfschmerzen. Campylobacter ist eine bakterielle Erkrankung, welche in den europäischen Ländern vermehrt in der warmen Jahreszeit auftritt. In Deutschland sind überwiegend Kinder

[2] RKI Infektionsepidemiologisches Jahrbuch (abgerufen am 01.04.2019): https://www.rki.de/DE/Content/Infekt/Jahrbuch/jahrbuch_node.html
[3] Robert Koch-Institut: SurvStat@RKI 2.0, https://survstat.rki.de
[4] RKI-Ratgeber Campylobacter-Enteritis (abgerufen am 01.04.2019): https://www.rki.de/DE/Content/InfAZ/C/Campylobacter/Campylobacter.html?nn=2386228

Tabelle 1: Beispiel-Daten von Campylobacter-Enteritis in Frankfurt am Main mit unterschiedlich gefilterten Gruppen an Menschen. Die Altersgruppe ist in Intervallen angegeben, wobei 20..24 bedeutet, dass nach dem Alter von 20 bis 24 gefiltert ist.

Campylobacter-Enteritis						
Bundesland	Kreis	Kalenderwoche	Geschlecht	Altersgruppe	Fallzahl	Inzidenz
Hessen	SK Frankfurt	2015-KW29	-	-	14	1,91
Hessen	SK Frankfurt	2015-KW29	-	20..24	5	11,55
Hessen	SK Frankfurt	2015-KW29	m	20..24	3	14,20

unter 5 Jahren und Erwachsene zwischen 20 und 29 Jahren betroffen. Mit ca. 60.000 bis 70.000 übermittelten Fällen pro Jahr ist Campylobacter die häufigste bakterielle meldepflichtige Krankheit in Deutschland. Die typischen Infektionswege sind das Baden in kontaminierten Oberflächengewässern oder der Verzehr von kontaminiertem Trinkwasser, nicht pasteurisierter Milch (Rohmilch) oder unzureichend gegartem Fleisch, welches mit Campylobacter infiziert ist. Da im Einzelhandel verkauftes Hähnchenfleisch oft kontaminiert ist, sollte immer dafür gesorgt werden, dass dieses Fleisch ausreichend gegart wurde. Die Übertragung von Mensch zu Mensch spielt bei Erwachsenen eine eher kleine Rolle. Kleinkinder sind jedoch anfällig bezüglich einer direkten Übertragung.
Bei der zweiten Krankheit handelt es sich um Keuchhusten[5], welcher auch unter dem lateinischen Namen Pertussis bekannt ist. Keuchhusten ist das ganze Jahr über vertreten, wobei die Inzidenz im Frühling und im Herbst etwas höher ist. Seit Frühjahr 2013 ist Keuchhusten eine bundesweit meldepflichtige Krankheit. Es gab in den vergangenen Jahren einige Erkrankungswellen, welche hauptsächlich Kinder von 4 bis 18 Jahren betroffen haben. Säuglinge haben ebenfalls eine sehr hohe Krankheitslast. Mittlerweile sind auch die Fallzahlen der Erwachsenen angestiegen, wodurch Empfehlungen für Impfungen ausgesprochen wurden. Keuchhusten wird mittels Tröpfcheninfektion übertragen, welche durch Husten, Reden oder Niesen in einem Abstand von einem Meter erfolgen kann.

Datenbeispiel

In Tabelle 1 sind als Beispiel drei Datenpunkte von Campylobacter dargestellt. Diese Datenpunkte beschreiben den gleichen Ort und die gleiche Zeit. Der Unterschied liegt in der Gruppe von Menschen, welche bei dem jeweiligen Datenpunkt betrachtet wird. In der Tabelle wird die betrachtete Gruppe in dem Datenpunkt von oben nach unten immer kleiner. Die Inzidenz kann daher ansteigen, obwohl die Fallzahlen immer geringer werden. Aus dem Beispiel lässt sich schließen, dass Männer im Alter von 20 bis 24 Jahren in dieser Woche, im Vergleich zu ganz Frankfurt, mit einer höheren Wahrscheinlichkeit an Campylobacter erkrankt sind.

2.2 Vorverarbeitung von Zeitreihen

Bei einer Zeitreihe handelt es sich um eine bestimmte Charakteristik, welche relativ zur Zeit dargestellt wird. Diese wird in festgelegten Abständen, sogenannten *Zeitperioden*, ermittelt. Es werden verschiedene Varianten verwendet, um den Wert einer Zeitperiode zu ermitteln. In dieser Arbeit werden Zeitreihen

[5] RKI-Ratgeber Keuchhusten (abgerufen am 01.04.2019):
`https://www.rki.de/DE/Content/Infekt/EpidBull/Merkblaetter/Ratgeber_Pertussis.html`

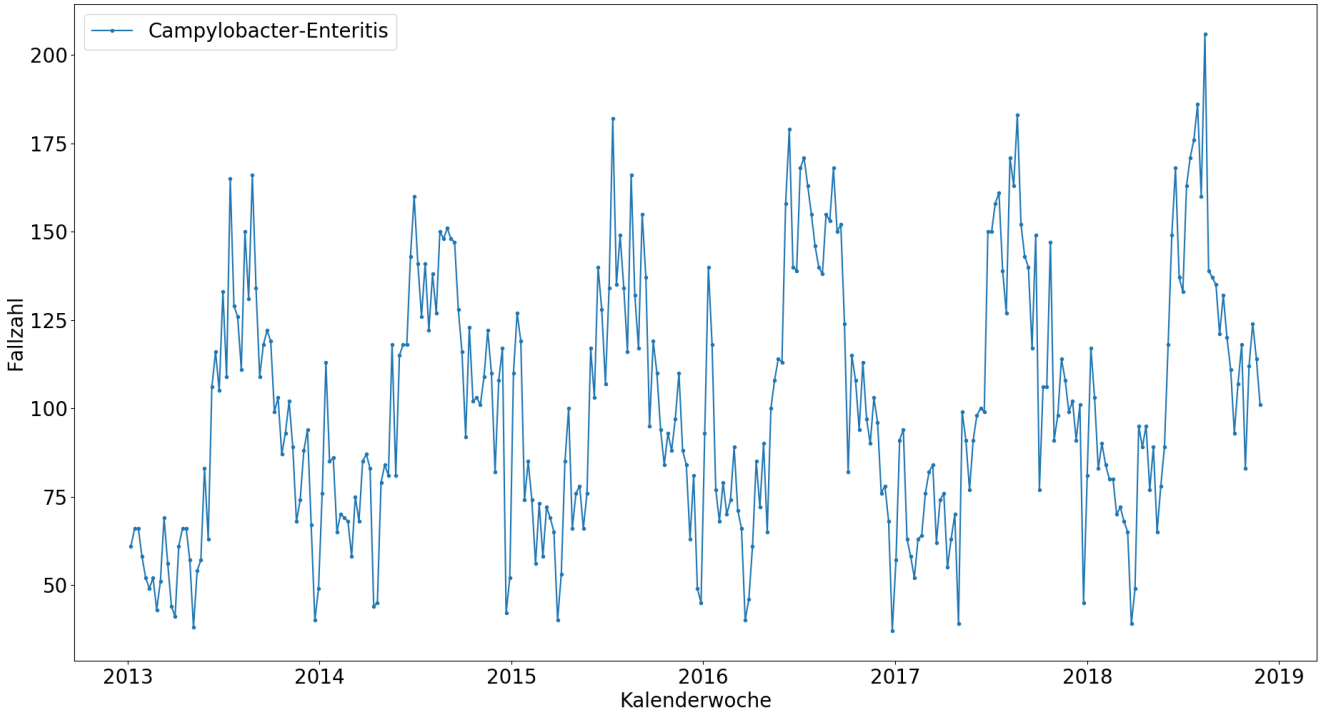

Abbildung 1: Beispiel einer Zeitreihe. Daten: Fallzahlen von Campylobacter-Enteritis in Hessen.

betrachtet, in welchen die Werte mittels Addition aller Fallzahlen einer Krankheit in einem Zeitraum von einer Kalenderwoche berechnet werden. Andere mögliche Varianten wären der Minimal-, Maximal- oder Mittelwert aller Werte in diesem Zeitraum. Diese Varianten werden jedoch hauptsächlich bei der Darstellung von Messwerten, wie zum Beispiel einer Geschwindigkeit oder Temperatur, genutzt.

Muster in Zeitreihen

In Zeitreihen können mehrere verschiedene Muster auftreten, welche es ermöglichen, eine genauere Vorhersage zu treffen. Die drei wichtigsten Muster sind der *Trend*, die *Saisonalität* und der *Zyklus*.

Der Trend beschreibt ein langfristiges Steigen oder Fallen des Wertes der Zeitreihe. Dieser kann linear aber auch abhängig von der Zeit sein.

Es wird von Saisonalität gesprochen, wenn die Zeitreihe ein immer wiederkehrendes Muster aufweist, welches in einem Zeitraum mit konstanter Länge auftritt. Dieser Zeitraum nennt sich *Saison* und hat wie vorher erwähnt eine feste Länge bzw. Frequenz, wie zum Beispiel wöchentlich oder jährlich. Bei der Betrachtung von Krankheiten ist oft eine sich jährlich wiederholende Saisonalität zu erkennen. Zum Beispiel tritt die Krankheit Campylobacter im Sommer immer öfter auf als im Winter.

Ein Zyklus steigt und fällt ebenfalls wie die Saisonalität. Der Unterschied liegt darin, dass es keine feste Frequenz gibt, wodurch aufeinander folgende Zyklen verschieden lange Zeiträume betreffen.

In Abbildung 1 ist der Verlauf der Krankheitsfälle von Campylobacter in Hessen dargestellt. Es ist eine klare Saisonalität zu erkennen, welche sich durch einen Anstieg der Fallzahl im Sommer und einen Abfall im Winter widerspiegelt. Darüber hinaus ist über die Jahre ein leichter Trend in positiver Richtung zu erkennen.

2.2.1 Anpassung einer Zeitreihe

Mit Hilfe von Anpassungen können Fehler und Ausreißer in den Daten bereinigt werden, welche ansonsten mit in die Vorhersage einfließen würden. Die Daten werden ebenfalls so angepasst, dass sie für die Auswertung miteinander verglichen werden können.
Eine sehr bekannte Anpassung von Zeitreihen ist die Kalender-Anpassung [Hyndman und Athanasopoulos 2018]. Bei monatlichen oder jährlichen Daten muss berücksichtigt werden, dass die Monate bzw. Jahre verschieden viele Tage beinhalten. Durch umwandeln der Werte in eine durchschnittliche Anzahl pro Tag des betrachteten Zeitraums, würden die Abweichungen behoben werden. Wöchentliche Daten haben jedoch immer die gleiche Anzahl an Tagen, weshalb bei den zur Verfügung stehenden Daten keine Kalender-Anpassung vorgenommen werden muss.
Um Zeitreihen mit unterschiedlich großen betrachteten Gruppen miteinander vergleichen zu können, muss der Wert der Zeitreihe relativ zur Größe der Gruppe und nicht absolut sein. Die Kalender-Anpassung erstellt einen relativen Bezug zur Zeit, wodurch ohne Probleme ein Februar à 28 Tagen mit einem Januar à 31 Tagen verglichen werden kann. Da bereits die Inzidenz in den Daten gegeben ist, muss keine extra Spalte in den Daten hinzugefügt werden, da diese relativ zu der betrachteten Menschengruppe ist. Die betrachteten Werte lassen sich folglich unabhängig von der Menschengruppe und der Kalenderwoche vergleichen. Bei einem Vergleich von Datenpunkten aus verschiedenen Orten sollte immer berücksichtigt werden, dass wahrscheinlich verschiedene regionale Effekte auf die Daten einwirken. Mögliche Einflüsse wären Wetter, Ferien oder kontaminiertes Wasser in einer bestimmten Region.
Es wird davon ausgegangen, dass in den betrachteten Daten keine falschen Informationen vorhanden sind. Ausreißer werden nicht entfernt, da diese eine mögliche Epidemie der Krankheit darstellen könnten. Dies wäre eine sehr wichtige Information, welche auf jeden Fall berücksichtigt werden muss. Falls in einer Woche keine neuen Fälle der Krankheit eingegangen sind, wird diese nicht in die Daten eingebunden. Daher müssen im Vorfeld die fehlenden Wochen mit Nullwerten aufgefüllt werden. Zusätzlich werden die Daten je nach Anwendungsfall nach Alter, Geschlecht und Ort gefiltert.

2.2.2 Transformieren einer Zeitreihe

Transformationen können die Muster einer Zeitreihe vereinfachen, wodurch die Vorhersage dieser präziser wird. Jedes Vorhersage-Modell hat verschiedene Vor- und Nachteile. In bestimmten Fällen ist es daher von Vorteil, eine Zeitreihe vor der Vorhersage zu transformieren und dies im Nachhinein wieder rückgängig zu machen. Dadurch können die Vorteile eines Modells besser ausgeschöpft werden und gegebenfalls sogar die Nachteile minimiert werden. In den folgenden Abschnitten werden die Logarithmus-, Exponential- und Box-Cox-Transformation erläutert. Hierzu wird die Original-Zeitreihe mit $y_1, ..., y_T$ und die transformierte Zeitreihe mit $w_1, ..., w_T$ abgebildet, wobei T für die Anzahl der Datenpunkte in der Zeitreihe steht.

Logarithmus-Transformation

Eine Logarithmus-Transformation (Log-Transformation) ist sinnvoll, wenn sich die Varianz der Zeitreihe mit einem ansteigenden Wert ebenfalls vergrößert. Die Transformation stabilisiert die Varianz der Zeitreihe, wodurch die Ergebnisse der Vorhersage sich drastisch verbessern können. Eine stabile Varianz

liegt vor, wenn die Varianz nicht von dem Mittelwert abhängt. Falls die Log-Transformation die Varianz jedoch nicht stabilisieren kann, leidet die Präzision der Vorhersage darunter. Geprüft wurde diese Aussage in einer Arbeit von Helmut Lütkepohl und Fang Xu [Lütkepohl und Xu 2009]. Bei einer Log-Transformation handelt es sich um eine Punkt-Transformation. Damit ist gemeint, dass jeder Datenpunkt unabhängig von dem Rest der Zeitreihe transformiert wird. Die Transformation und Rück-Transformation werden nach den folgenden Gleichungen umgesetzt. In diesen wird der natürliche Logarithmus verwendet, jedoch kann die Basis des Logarithmus beliebig gewählt werden, solange die Rück-Transformation dementsprechend angepasst wird.

$$w_t = log_e(y_t) \qquad y_t = e^{w_t} \qquad \text{mit} \quad 1 \leq t \leq T \tag{1}$$

Ein großes Problem dieser Transformation stellt dar, dass sie nicht mit Nullwerten oder negativen Werten umgehen kann, da der Definitionsbereich des Logarithmus den echt positiv reellen Zahlen entspricht. Deshalb müssen Null- und Negativwerte gesondert behandelt, oder es müssen im Vorhinein alle Nullwerte bereinigt werden. Die Log-Transformation bietet zwei entscheidende Vorteile. Einerseits wird die Vorhersage der Original-Zeitreihe auf positive Werte beschränkt, da der Wertebereich der Rück-Transformation den echt positiv reellen Zahlen entspricht. Auf der anderen Seite sind die transformierten Daten interpretierbar. Bei einer Log-Transformation zur Basis 10 würde eine Erhöhung der transformierten Zeitreihe um 1 einer Multiplikation um 10 in der Original-Zeitreihe entsprechen.

Exponential-Transformation

Eine Alternative zu der Log-Transformation ist die Exponential-Transformation. Die Quadratwurzel-Transformation oder kubische Wurzel-Transformation fallen in diese Kategorie, da die Rück-Transformation der Exponential-Transformation einer Wurzelfunktion entspricht. In den folgenden Gleichungen werden die Transformation und Rück-Transformation dargestellt:

$$w_t = y_t^p \qquad y_t = w_t^{1/p} \qquad \text{mit} \quad 1 \leq t \leq T \tag{2}$$

Eine Exponential-Transformation sollte lediglich auf nicht negativen reellen Zahlen angewandt werden, da ansonsten die Information des Vorzeichens verloren geht. Ebenfalls sind die transformierten Werte nicht einfach zu interpretieren. Die Ergebnisse von Vorhersagen werden mittels dieser Transformation nur leicht verändert, jedoch hat sie einen großen Einfluss auf die Vorhersage-Intervalle, welche im Abschnitt 2.3 genauer erläutert werden.

Box-Cox-Transformation

Die Box-Cox-Transformation kombiniert die Log- und Exponential-Transformation, indem der Parameter λ bestimmt, welche der beiden Transformationen genutzt wird. In dieser Arbeit wird jedoch eine erweiterte Form verwendet, in der die Parameter λ_1 und λ_2 eingesetzt werden. Die Angepasste-Box-Cox-Transformation (ABC-Transformation) ist wie folgt definiert, wobei diese für jeden Zeitpunkt $1 \leq t \leq T$ gilt:

$$w_t = \begin{cases} log_e(y_t + \lambda_2) & wenn\ \lambda_1 = 0; \\ \frac{(y_t+\lambda_2)^{\lambda_1} - 1}{\lambda_1} & sonst. \end{cases} \tag{3}$$

Der Parameter λ_1 hat die gleiche Funktion wie λ in der normalen Box-Cox-Transformation. Falls λ_1 gleich null ist, wird eine Log-Transformation mit natürlichem Logarithmus angewandt. Andernfalls wird eine Exponential-Transformation mit einer einfachen Skalierung verwendet. Der Parameter λ_2 ist eine Konstante, welche vor der Transformation auf die Werte von y addiert wird. Mit Hilfe der Verschiebung der Zeitreihe in y-Achsen-Richtung kann die Zeitreihe so angepasst werden, dass die Log- bzw. Exponential-Transformation ohne Komplikationen durchgeführt werden kann. Wie in den vorherigen Abschnitten beschrieben, ist der Definitionsbereich der Log-Transformation auf die echt positiv reellen Zahlen und der Definitionsbereich der Exponential-Transformation auf die nicht negativen reellen Zahlen beschränkt. In der folgenden Gleichung ist mit denselben Variablenbezeichnungen die Rück-Transformation dargestellt, welche ebenfalls für jeden Zeitpunkt $1 \leq t \leq T$ gilt:

$$y_t = \begin{cases} e^{w_t} - \lambda_2 & wenn\ \lambda_1 = 0; \\ (\lambda_1 w_t + 1)^{1/\lambda_1} - \lambda_2 & sonst. \end{cases} \tag{4}$$

2.2.3 Dekomposition einer Zeitreihe

Eine Zeitreihe lässt sich in die vorher genannten Muster zerlegen. Es wird unterschieden zwischen der additiven und multiplikativen Dekomposition. In dieser Arbeit werden lediglich Verfahren der additiven Dekomposition verwendet. Der Grund besteht darin, dass durch die Kombination einer Logarithmus-Transformation und einer additiven Dekomposition die gleichen Resultate erzielt werden, wie mit einer multiplikativen Dekomposition. Durch die Logarithmus-Transformation wechseln, wie in der folgenden Gleichung gezeigt wird, die multiplikativen Veränderungen für jeden Zeitpunkt $1 \leq t \leq T$ zu additiven Veränderungen [Hyndman und Athanasopoulos 2018].

$$y_t = T_t \cdot S_t \cdot R_t \quad \text{ist äquivalent zu} \quad log(y_t) = log(T_t) + log(S_t) + log(R_t) \tag{5}$$

In den folgenden Abschnitten wird zwischen Verfahren der Dekomposition unterschieden, welche die Zeitreihe in zwei oder drei Komponenten zerlegen. Die Zerlegung der Ursprungs-Zeitreihe y zum Zeitpunkt t in drei Komponenten lässt sich durch Addition des Trends T, der Saisonalität S und des Rests R darstellen. Falls die Zeitreihe einen Zyklus beinhaltet, ist dieser normalerweise im Trend verschlüsselt. In Abbildung 2 ist ein Beispiel einer Zerlegung der Zeitreihe in drei Komponenten dargestellt.

$$y_t = T_t + S_t + R_t \qquad \text{mit} \quad 1 \leq t \leq T \tag{6}$$

Die Zerlegung in zwei Komponenten bezeichnet eine *Glättung* der Zeitreihe, womit die Varianz bzw. das Rauschen der Werte rausgefiltert wird. Das Resultat einer Glättung ist in Abbildung 3 dargestellt. In

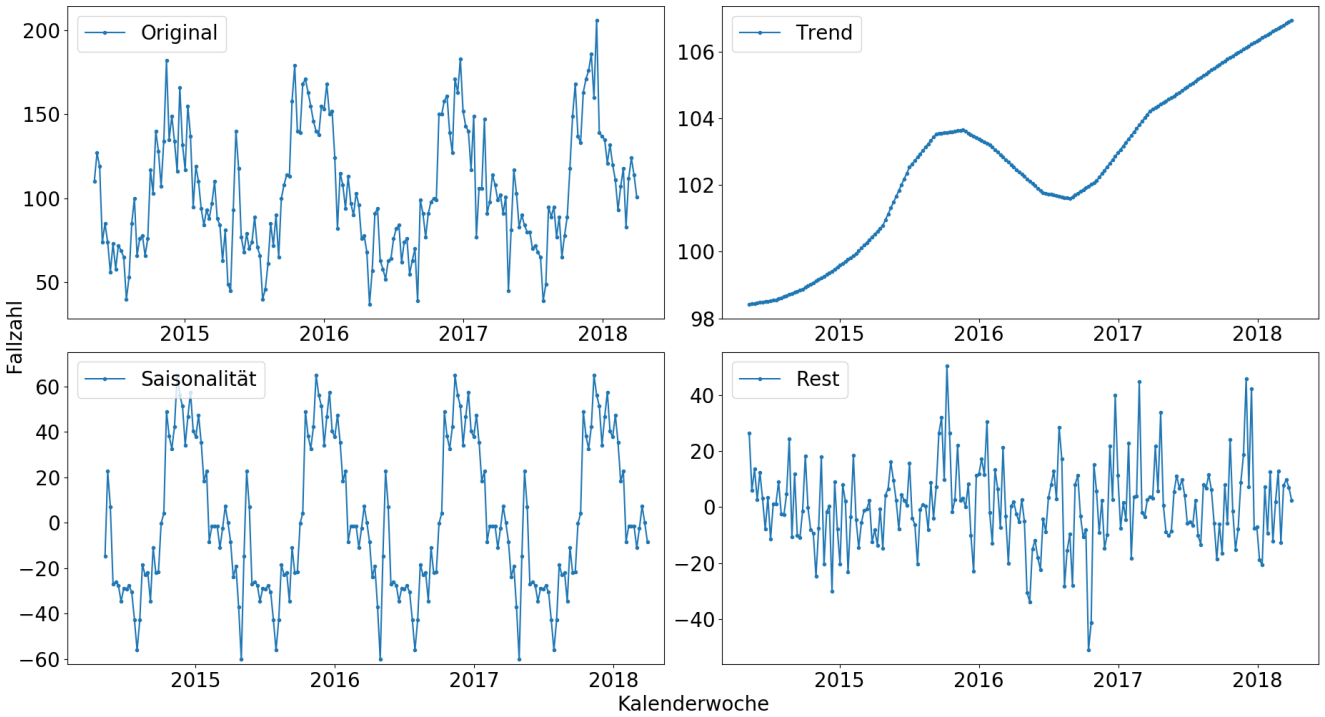

Abbildung 2: Beispiel einer mit der STL Dekomposition zerlegten Zeitreihe. Daten: Fallzahlen von Campylobacter-Enteritis in Hessen.

dieser sind die Original-Zeitreihe y, die geglättete Zeitreihe G und der Rest R abgebildet. In der geglätteten Zeitreihe können Trend, Zyklus und Saisonalität verschlüsselt sein. Der Rest hingegen realisiert das Rauschen der Zeitreihe sowie alle Muster, welche nicht durch das Verfahren erfasst wurden.

$$y_t = G_t + R_t \qquad \text{mit} \quad 1 \leq t \leq T \tag{7}$$

Moving Averages

Eine der einfachsten und bekanntesten Glättungen ist die Methode *Moving Averages* (MA) [Hyndman und Athanasopoulos 2018]. Bei dieser Methode werden zur Berechnung eines Wertes auch die *Nachbarn* des Wertes einbezogen. Nachbarn sind Datenpunkte in der Zeitreihe, welche zeitlich direkt vor oder nach dem Datenpunkt liegen. Der Bereich der Nachbarn, welcher für die Berechnung eines bestimmten Datenpunktes genutzt wird, lautet *Fenster*. Die Berechnung der Glättung G der Zeitreihe y ist wie folgt definiert, wobei m die Größe, w_s der Start und w_e das Ende des Fensters sind.

$$G_t = \frac{1}{m} \sum_{j=w_s}^{w_e} y_{t+j} \qquad \text{mit} \quad (1 - w_s) \leq t \leq (T - w_e) \tag{8}$$

Der Start des Fensters liegt im Normalfall in der Vergangenheit und ist daher eine negative ganze Zahl. Das Ende des Fensters kann ein Datenpunkt in der Zukunft oder der betrachtete Datenpunkt selbst sein, falls zur Glättung keine Informationen aus der Zukunft verwendet werden sollen. Wenn das Fenster bei einem Datenpunkt am Anfang oder Ende der Zeitreihe nicht komplett gefüllt werden kann, ist der Wert

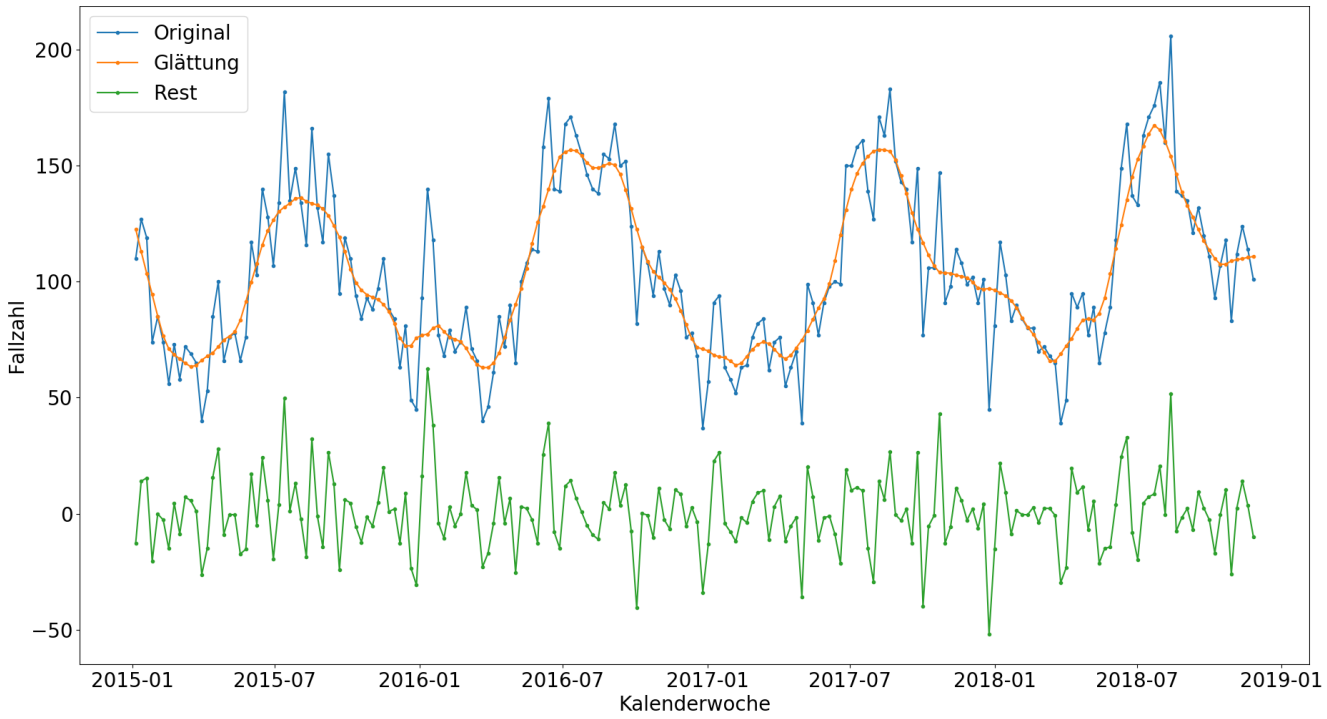

Abbildung 3: Beispiel einer mit dem LOWESS Algorithmus geglätteten Zeitreihe. Daten: Fallzahlen von Campylobacter-Enteritis in Hessen.

in der geglätteten Zeitreihe nicht definiert. In Tabelle 2 ist ein einfaches Beispiel einer Glättung mittels MA dargestellt, wobei das Fenster einen Zeitpunkt vor dem jeweiligen Datenpunkt startet und einen danach endet.

Locally Weighted Estimated Scatterplot Smoothing

Eine weitere, komplexere Glättung ist bekannt als *Locally Weighted Estimated Scatterplot Smoothing* (LOWESS). Diese Methode ist eine nicht parametrische Strategie, für welche kein Vorwissen oder irgendwelche Vermutungen zur Verteilung der Daten angegeben werden müssen. Im Folgenden wird lediglich die Grundidee von LOWESS übermittelt, da eine detailreiche Darstellung über den Rahmen der Arbeit hinaus gehen würde. Falls jedoch genauere Informationen benötigt werden, können diese in den Arbeiten von William Cleveland nachgelesen werden [Cleveland 1979; Cleveland und Loader 1996].

Für die Berechnung der Glättung eines Datenpunktes wird, wie bei MA, ein lokales Fenster um den Datenpunkt betrachtet. Hierfür muss anfangs festgelegt werden, wie groß dieses Fenster sein soll. Ebenfalls werden zu Beginn die Nachbarschaftsgewichte jedes Datenpunktes mit einer Gewichtsfunktion bestimmt. Diese Gewichte erzeugen den Effekt, dass weiter entfernte Punkte einen kleineren Einfluss auf die Glättung haben werden. Anschließend wird in jedem Fenster eine Lineare Regression mit einer Kleinsten-Quadrate-Schätzung durchgeführt, welche später in Abschnitt 2.3.2 genau erläutert wird. Da die Fenster nicht viele Werte beinhalten, sind die Schätzungen der Linearen Regression sehr anfällig bezüglich Ausreißern. Daher werden im letzten Schritt des LOWESS-Verfahrens Robustheitsgewichte bestimmt, mit denen die Ausreißer ein kleineres Gewicht zugewiesen bekommen sollen. Danach werden die Nachbarschaftsgewichte und die Robustheitsgewichte jedes Datenpunktes miteinander multipliziert.

Tabelle 2: Beispiel einer Glättung mit MA mit $w_s = -1$ und $w_e = 1$. Die Zeitreihe besteht aus willkürlichen Daten.

Moving Averages			
Kalenderwoche	Original	Glättung	Rest
1	57	-	-
2	70	65	5
3	75	74	1
4	77	72	5
5	64	65.3	-1.3
6	55	53	2
7	40	-	-

Abschließend wird mit den neuen Gewichten erneut eine Lineare Regression in den Fenstern durchgeführt.

Der letzte Schritt kann beliebig oft durchgeführt werden, bis die gewünschte Glättung erreicht ist. Die Zeitreihe wird jedoch bei einer gewissen Anzahl an Iterationen kaum eine weitere Veränderung wahrnehmen. Wie schon in der Erklärung der Glättung erwähnt, ist in Abbildung 3 ein Beispiel des LOWESS Verfahrens abgebildet.

Seasonal-Trend Decomposition Procedure Based on Loess

Eine sehr verbreitete Dekomposition in drei Komponenten ist bekannt als *Seasonal-Trend Decomposition Procedure Based on Loess* oder kurz STL Dekomposition. Ein Kriterium für die Wahl dieses Verfahrens ist, dass es beliebig lange Saisons bearbeiten kann. Andere Verfahren, wie zum Beispiel X11- oder SEATS-Dekomposition, besitzen vordefinierte Längen, wie 4 oder 12, für eine Saison [Hyndman und Athanasopoulos 2018]. Aus diesem Grund könnten diese Verfahren nicht mit wöchentlichen Daten arbeiten, wodurch sie auf die in dieser Arbeit verwendeten Daten nicht anwendbar sind. Zusätzlich ist das STL-Verfahren sehr robust bezüglich Ausreißern. Ausgehend davon werden die Ausreißer im Rest R abgebildet, was es ermöglicht, die Ausreißer in dieser Komponente genauer zu analysieren. Im Folgenden wird erneut lediglich vereinfacht das Grundprinzip der STL Dekomposition erläutert, da eine umfangreiche Beschreibung über die Reichweite dieser Arbeit hinaus gehen würde. Das Verfahren wurde von William Cleveland und weiteren Personen detailreich ausgearbeitet und in einer wissenschaftlichen Ausarbeitung festgehalten [Cleveland u. a. 1990].

Das STL-Verfahren besteht aus zwei rekursiven Schleifen, wobei sich die eine Schleife in der anderen befindet. Aus diesem Grund werden diese auch innere und äußere Schleife genannt. Eine sehr abstrakte Darstellung des Verfahrens ist in Algorithmus 1 dargestellt. In der inneren Schleife wird in jeder Iteration durch Anwendung von MA-Glättung, LOESS und weiteren Glättungen die Trend- und Saisonalitäts-Komponente geglättet, auf Grundlage dessen die Genauigkeit der Schätzung der Komponenten erhöht wird. Zusätzlich werden zunehmend mit jedem Durchlauf der Trend aus der Saisonalitäts-Komponente und die Saisonalität aus der Trend-Komponente minimiert. Nachdem die innere Schleife die angegebene Anzahl an Iterationen durchlaufen hat, werden in der äußeren Schleife Gewichte berechnet, welche die Robustheit gegenüber Ausreißern sicherstellen. Anschließend startet der Prozess der

Algorithm 1 Abstrakte Darstellung einer STL-Dekomposition der Zeitreihe Y

```
→ Initialisiere Trend T = 0
→ Initialisiere Rest R = 0
for i = 1 to n_o do
    → Berechne den Rest
    → Berechne die Robustheitsgewichte
    for j = 1 to n_i do
        → Entferne den Trend von Y
        → Glättung jeder individuellen Saison von Y (LOESS, MA und weitere Verfahren)
        → Entfernung des Trends jeder individuellen Saison
        → Entfernen der Saisonalität von Y
        → Glättung des Trends (LOESS)
    end for
end for
```

inneren Schleife erneut. In dem neuen Durchlauf werden allerdings die neu berechneten Gewichte der äußeren Schleife integriert, wodurch die Berechnung der Komponenten präziser wird. Die Anzahl der Iterationen der inneren n_i und der äußeren Schleife n_o müssen im Vorhinein festgelegt werden. Ein zusätzlicher Parameter legt fest, wie groß das Fenster sein soll, welches zur Berechnung eines Saisonalitäts-Wertes betrachtet wird. Die Fenstergröße und ein zusätzlicher Parameter bestimmen, wie sehr sich die Saisonalitäts-Komponente über die Zeit verändert. Wenn das Fenster sehr groß gewählt wird, weisen die aufeinanderfolgenden Saisons nahezu keine Veränderung auf. In Abbildung 2 ist das Ergebnis einer STL Dekomposition mit einer hohen Fenstergröße dargestellt.

2.3 Vorhersage von Zeitreihen

Das Ziel der Vorhersage einer Zeitreihe ist es, den nächsten Zeitpunkt der *Zielvariable* möglichst genau zu bestimmen. Unter Zielvariable versteht sich der Wert, welcher in der Vorhersage bestimmt werden soll. In dieser Arbeit wird, je nach Anwendungsfall, entweder die Anzahl der Fallzahlen oder die Inzidenz als Zielvariable verwendet. Bei der Vorhersage wird unterschieden zwischen Modellen, welche mit einen oder mehreren *Inputs* arbeiten. Modelle, welche nur einen Input verwenden, nutzen lediglich eine Information pro Zeitpunkt, um den oder die nächsten Zeitpunkte vorherzusagen. Im Normalfall handelt es sich bei dieser Information um die vergangene Zeitreihe, für welche die Vorhersage getroffen werden soll. Falls mehrere Informationen vorhanden sind, wie zum Beispiel zusätzliche Wetterdaten oder Informationen über Ferien und Feiertage, können Modelle mit mehreren Inputs zusätzlich diese Informationen für die Vorhersage nutzen. Wenn die zusätzlichen Informationen Korrelationen zu der Zielvariable aufweisen, sollte die Vorhersage präziser werden. Es existieren zusätzlich auch Modelle, welche mehrere Zielvariablen vorhersagen. Diese werden jedoch in dieser Arbeit nicht benötigt.

Das Ergebnis der Vorhersage kann auf zwei verschiedene Arten angegeben werden. Die erste Variante ist die *Punkt-Vorhersage*, welche den wahrscheinlichsten Wert für den nächsten Zeitpunkt angibt. Die Alternative sind *Vorhersage-Intervalle* bzw. *Konfidenz-Intervalle*, welche ein Intervall von Werten darstellen, in dem der nächste Zeitpunkt zu einer bestimmten Wahrscheinlichkeit liegt. In Abbildung 4 wird die Kombination der beiden Varianten in einem willkürlichen Beispiel dargestellt. Es handelt sich um eine 10-Jahres-Vorhersage, welche in der Punkt-Vorhersage einen linearen Anstieg zeigt. Um die Punkt-Vorhersage sind zwei Vorhersage-Intervalle mit den Wahrscheinlichkeiten von 80 und 95 Prozent

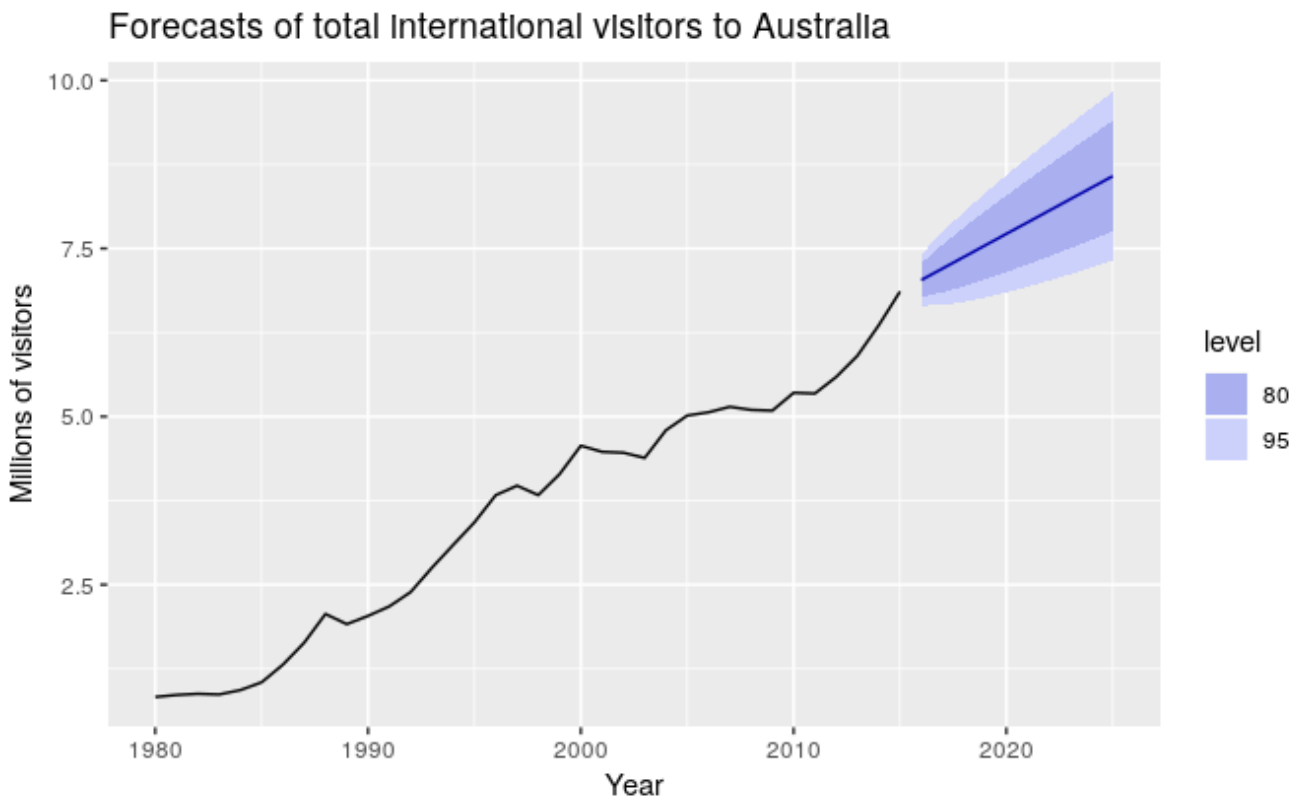

Abbildung 4: Vorhersage von 10 Jahren der gesamten Anzahl an internationalen Besuchern in Australien. Zur Vorhersage wurde das Drift Modell genutzt. Darstellung der Punkt-Vorhersage und der Vorhersage-Intervalle der Wahrscheinlichkeiten 80 und 95 Prozent. Quelle: [Hyndman und Athanasopoulos 2018]

eingezeichnet. Diese kennzeichnen den Bereich, in dem der Wert der Vorhersage zur gegebenen Wahrscheinlichkeit liegen wird. Klar erkennbar ist, dass, je weiter die Vorhersage in der Zukunft liegt, desto breiter werden die Intervalle. Dies ist darin begründet, dass die Vorhersagen immer ungenauer ausfallen, da der Fehler der vorherigen Vorhersagen in die Berechnung der nächsten einfließt.

In dieser Arbeit werden zur Vorhersage statistische Modelle sowie Modelle aus dem Gebiet des maschinellen Lernens verwendet. Es gibt sehr simple Modelle, welche lediglich bestimmte Werte aus den gegebenen Informationen abrufen oder mit einfachen mathematischen Gleichungen zu berechnen sind. Komplexere Modelle müssen jedoch erst trainiert werden, welches als Fitting, Training oder Lernen eines Modells bezeichnet wird. Während dieser Phase lernt das Modell, mit Hilfe der gegebenen Informationen der Inputs und den dazugehörigen Zielvariable des Trainings-Datensatzes, zum Beispiel Parameter, Gewichte oder Entscheidungsbäume. Wie der Trainings-Datensatz generiert wird, ist im späteren Abschnitt 2.4.2 beschrieben. Das in der Lern-Phase trainierte Modell ermöglicht es, die Zielvariable eines Datenpunktes, mittels gegebenen Input dieses Punktes, zu bestimmen. Da die Vorhersage nicht unbedingt mit dem observierten Wert übereinstimmt, wird im Folgenden die Vorhersage mit $\hat{y}$ und die Original-Zeitreihe mit y beschrieben.

2.3.1 Standard Modelle

Die folgenden Modelle sind sehr einfach gehalten. Trotzdem produzieren sie teils sehr gute Vorhersagen. Sie dienen daher als guter Vergleich, um herauszufinden, ob ein komplexeres Modell sich überhaupt rentiert. Ein großer Vorteil der hier genannten Standard Modelle besteht darin, dass die Vorhersage mit sehr wenig bis gar keinem Rechenaufwand verbunden ist. Die Ergebnisse sind in diesem Fall schnell abrufbar.

Naïve Modell

Bei diesem Modell ist keine Berechnung notwendig. Die Vorhersage der Zeitreihe y wird mit

$$\hat{y}_{T+h} = y_T \tag{9}$$

bestimmt, wobei T der Gesamtanzahl der Datenpunkte in y entspricht und h den Zeitpunkt bestimmt, welcher vorhergesagt werden soll. Dieses einfache Modell erzeugt überraschend gute Ergebnisse in Zeitreihen von Finanzen, solange h nicht zu groß gewählt ist.

Seasonal Naïve Modell

Das Seasonal Naïve Modell (S-Naive) beinhaltet ebenfalls lediglich eine Abfrage eines Wertes der gegebenen Zeitreihe y. Die Vorhersage wird mit

$$\hat{y}_{T+h} = y_{T-m+(h \bmod m)} \tag{10}$$

bestimmt, wobei m die Anzahl der Datenpunkte einer Saison entspricht. Das bedeutet, dass immer der letzte bekannte Wert der gleichen Position in der Saison vorhergesagt wird. Hierfür muss im Vorhinein die Länge einer Saison definiert sein.

Average Modell

Zur Vorhersage werden bei diesem Modell alle Werte der gegebenen Zeitreihe y für die Berechnung verwendet.

$$\hat{y}_{T+h} = \sum_{j=1}^{T} y_j \tag{11}$$

Die Ergebnisse des Average Modells weichen schnell von den beobachteten Werten ab, falls Zyklen, Saisonalitäten oder Trends in der Zeitreihe vorhanden sind. Dieses Modell kann sich für das Vorhersagen von einer Zeitreihe, welche um eine Konstante rauscht, eignen.

Drift Modell

Das Drift Modell ist das letzte in dieser Arbeit vorgestellte Standard Modell. Dieses Modell erweitert das vorher genannte Naïve Modell mit einer Abbildung des Trends in der Vorhersage. Die Umsetzung wird mit

$$\hat{y}_{T+h} = y_T + h\frac{y_T - y_1}{T-1} \tag{12}$$

dargestellt, wobei der Bruch den Trend bzw. die Steigung zwischen dem ersten und letzten Datenpunkt der gegebenen Zeitreihe y darstellt. Dadurch wird der Trend in der Vorhersage berücksichtigt, indem er durchgehend weitergeführt wird.

2.3.2 Lineare Regression

Das Modell der Linearen Regression [Hyndman und Athanasopoulos 2018] kann mit einem oder auch mehreren Inputs umgehen. In dieser Arbeit wird ein Modell mit einem Input *einfache Lineare Regression* und mit mehr als einem Input *multiple Lineare Regression* genannt. Es existieren mehrere Bezeichnungen in verschiedenen Literaturen. Zum Beispiel werden die Bezeichnungen univariable Lineare Regression und multivariable Lineare Regression verwendet [Schneider, Hommel und Blettner 2010]. Die Lineare Regression kann lediglich numerische Inputs verwenden. Falls ein möglicher Input nicht numerisch ist, kann dieser zum Beispiel mit sogenannten *Dummy Variablen* codiert werden, welche jeweils eine 0 oder 1 annehmen können. Eine binäre Information wie das Geschlecht kann mit einer Dummy Variable dargestellt werden, indem die Variable 0 für männlich und 1 für weiblich annimmt. Falls jedoch ein kategorisches Attribut mit mehr als zwei möglichen Werten dargestellt werden soll, muss dieses mit einer Kodierung mehrerer Dummy Variablen dargestellt werden. Die Gleichung der multiplen Linearen Regression lautet

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \ldots + \beta_k x_{k,t} + \epsilon_t \qquad \text{mit} \quad 1 \leq t \leq T, \tag{13}$$

wobei y die Zielvariable bzw. unabhängige Variable ist, $x_1,\ldots,$ x_k die Vorhersage- bzw. abhängigen Variablen, β_0 der y-Achsen-Abstand ist und $\beta_1,\ldots,$ β_k die Gewichte der abhängigen Variablen sind. In ϵ_t ist die Abweichung zwischen der Vorhersage und der tatsächlichen Zielvariable verschlüsselt, welcher als Vorhersage-Fehler bezeichnet wird. Für $k = 1$ entspricht die Gleichung der einfachen Linearen Regression, welche daher eine besondere Form der multiplen Linearen Regression ist.

Um eine Vorhersage des nächsten Zeitpunktes zu treffen, müssen zunächst die Gewichte geschätzt werden. Zur Schätzung der Gewichte werden die Daten der Vorhersage-Variablen x und der Zielvariable y genutzt. Das Ziel der Schätzung ist es, die Gewichte so zu bestimmen, dass der Gesamtfehler aller Vorhersage minimiert wird. Der Gesamtfehler ist in folgender Gleichung abgebildet:

$$\sum_{t=1}^{T} \epsilon_t^2 = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 x_{1,t} - \beta_2 x_{2,t} - \ldots - \beta_k x_{k,t})^2 \tag{14}$$

Diese Methode nennt sich *Kleinste-Quadrate-Schätzung* bzw. in Englisch *Least Square Estimation* (LSE). Das Suchen der besten Gewichte wird in diesem Modell als das Training bezeichnet. Zur Vorhersage werden dem gelernten Modell die abhängigen Variablen des gewünschten Zeitpunktes übergeben und das Modell berechnet die Vorhersage mit den geschätzten Gewichten.

2.3.3 Random Forest Regression

Die Technik *Random Forest* kann für Klassifikation und Regression verwendet werden [Breiman 2001]. Da die Fallzahlen bzw. Inzidenzen vorhergesagt werden müssen, wird die Variante der Regression verwendet. Die Anzahl der Inputs ist bei Random Forest nicht begrenzt und wird daher in dieser Arbeit analog zur Linearen Regression mit einfache und multiple Random Forest Regression bezeichnet. Die Grundidee des Random Forest ist es, eine Großzahl an Entscheidungsbäumen zu generieren. Zur Generierung eines Baumes wird ein *Sample* bzw. eine *Stichprobe* aus den Trainings-Daten verwendet. Eine bekannte Methode zur Erstellung eines Samples ist das *Bootstrap-Verfahren*, welches aus den Trainings-Daten zufällig Datenpunkte zieht und diese wieder zurücklegt. Dadurch entstehen Samples, welche bestimmte Datenpunkte mehrfach oder auch gar nicht beinhalten können. Nachdem eine gegebene Anzahl von Samples erstellt wurde, werden aus diesen die Entscheidungsbäume gelernt.
Zur Vorhersage wird der Input des gesuchten Zeitpunktes jedem Entscheidungsbaum übergeben. Nachdem jeder Baum eine Vorhersage mit Hilfe des gegebenen Inputs bestimmt hat, wird ein Mittelwert aller Ergebnisse gebildet. Dieser Wert ist die endgültige Vorhersage des Random Forest.

2.3.4 Exponential Smoothing

Das einfachste Modell des *Exponential Smoothing* (ES) nennt sich *Simple Exponential Smoothing* (SES) [Hyndman und Athanasopoulos 2018; Holt 2004]. Beim SES werden zur Berechnung der Vorhersage die vergangenen observierten Datenpunkte verwendet. Je weiter der Datenpunkt in der Vergangenheit liegt, desto kleiner ist das Gewicht dieses Datenpunktes. Die Gewichte werden mittels eines sogenannten *Smoothing Parameters* α berechnet, für welchen $0 \leq \alpha \leq 1$ gilt. Die Vorhersage des nächsten Datenpunktes mit SES ist folgendermaßen definiert:

$$\hat{y}_{T+1} = \sum_{i=0}^{T-1} \alpha(1-\alpha)^i y_{T-i} \tag{15}$$

Um den Rechenaufwand zu minimieren, lässt sich diese Gleichung umstellen, sodass nicht jedes Mal alle vergangenen Datenpunkte mit einem Gewicht multipliziert werden müssen. Die Gleichung wird in eine rekursive Form gebracht, welche wie folgt beschrieben ist:

$$\hat{y}_{T+1} = \alpha y_T + (1-\alpha)\hat{y}_T \tag{16}$$

In dieser Form ist einfach zu erkennen, welchen Einfluss α in der Berechnung hat. Je näher α an 1 liegt, desto mehr fließt der letzte observierte Wert in die Vorhersage ein. Andernfalls hat die letzte Vorhersage einen größeren Einfluss auf die Vorhersage.

In SES wird die Vorhersage mit einer Gleichung und einem zugehörigen Smoothing Parameter abgebildet. Das komplexere Modell *Holt-Winters* baut auf dieser Idee auf und nutzt drei dieser Gleichungen, um eine Vorhersage zu treffen. Diese Gleichungen stellen das Level l, den Trend b und die Saisonalität s dar. Es wird unterschieden zwischen einigen Variationen von ES-Modellen. Der Trend kann additiv oder gedämpft additiv in die Vorhersage einfließen. Bei einer Dämpfung würde ein zusätzlicher Parameter dafür sorgen, dass der Trend nicht proportional zur Zeit, sondern ebenfalls mit der Zeit abschwächen würde. Die Saisonalität hingegen kann additiv oder multiplikativ in die Vorhersage einbezogen werden. Beim additiven Holt-Winters fließen der Trend und die Saisonalität additiv in die Vorhersage ein. Die Gleichungen dieses Modells müssen iterativ für alle Zeitpunkte $1 \leq t \leq T$ angewandt werden:

$$\begin{aligned} l_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \\ s_t &= \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \\ \hat{y}_{T+h} &= l_t + hb_t + s_{t-m+(h \bmod m)} \end{aligned} \tag{17}$$

Die Smoothing Parameter der jeweiligen Gleichung sind mit α, β und γ dargestellt und für diese gilt $0 \leq \alpha, \beta, \gamma \leq 1$. In der Gleichung der Vorhersage wird der Trend proportional fortgesetzt und als Saisonalitäts-Wert wird der letzte bekannte Wert an dieser Position der Saison mit der Länge m verwendet. Die Smoothing Parameter werden hier auch *Hyperparameter* des Modells genannt, da diese nicht während des Verfahrens, sondern vor der Durchführung festgelegt werden müssen. Diese müssen daher speziell für die betrachteten Daten gewählt werden, sodass die Vorhersage in der bestmöglichen Genauigkeit resultiert.
Alle weiteren ES-Variationen sind in Abbildung 5 aufgelistet, wobei in der Mitte das additive Holt-Winters Modell und oben links das SES Modell wiederzufinden sind. Es ist darauf zu achten, dass in der abgebildeten Tabelle β^* anstatt β als Smoothing Parameter für den Trend verwendet wird. Die Wahl des richtigen Modells ist von dem gegebenen Datensatz abhängig, da jede Variation seine Vor- und Nachteile hat. Zum Beispiel sollte kein Modell mit der Saisonalitäts Smoothing Gleichung verwendet werden, wenn der Datensatz gar keine Saisonalität aufweist.

2.3.5 Autoregressive Integrated Moving Averages

Das *Autoregressive Integrated Moving Averages* (ARIMA) Modell ist ein sehr renommiertes Verfahren zur Vorhersage von Zeitreihen [Hyndman und Athanasopoulos 2018]. Es wird unterschieden zwischen dem *Non-Seasonal ARIMA* (nARIMA) und *Seasonal ARIMA* (sARIMA) Modell. Das nARIMA Modell besteht aus drei unterschiedlichen Komponenten. Die erste ist ein Autoregressives (AR) Modell der Ordnung p, welches quasi eine multiple Lineare Regression ist, wobei die Vorhersage-Variablen den p letzten Werten der Zeitreihe entspricht.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_k y_{t-p} + \epsilon_t \qquad \text{mit} \quad (1+p) \leq t \leq T \tag{18}$$

Die Gewichte ϕ müssen einige Bedingungen erfüllen, welche hier nicht weiter erläutert werden, jedoch im Werk von Hyndman nachgeschlagen werden können [Hyndman und Athanasopoulos 2018].

Trend	Seasonal		
	N	**A**	**M**
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{A_d}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$

Abbildung 5: Auflistung aller möglichen Modell-Variationen des Exponential Smoothing, wobei N für nicht vorhanden, M für multiplikativ, A für additiv und A_d für gedämpft additiv steht. Die Variable k entspricht dem Ergebnis einer ganzahligen Division von $(h-1)/m$. Quelle: [Hyndman und Athanasopoulos 2018]

Die zweite Komponente ist ein Moving Average Modell der Ordnung q, welches ebenfalls mit einer Regression dargestellt wird und nicht mit der MA Glättung zu verwechseln ist. Diesmal handelt es sich bei den Vorhersage-Variablen um die vergangenen Vorhersage-Fehler ϵ.

$$y_t = c + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \ldots + \theta_k\epsilon_{t-q} \qquad \text{mit} \quad (1+q) \leq t \leq T \tag{19}$$

Der Vorhersage-Fehler ϵ_t kann während der Vorhersage des Datenpunktes y_t unmöglich bekannt sein. Deshalb wird dieser Fehler mit weißem Rauschen dargestellt, welches hier eine normalverteilte Variable ist.

Die letzte Komponente bestimmt, wie oft die Zeitreihe vor der Anwendung des AR Modells differenziert werden soll. Die Differenzierung einer Zeitreihe ist identisch mit der Ableitung in der Mathematik. In einer Differenzierung erster Ordnung wird die Steigung zwischen aufeinander folgenden Datenpunkten gemessen und diese in einer neuen Zeitreihe dargestellt. Der Parameter d des nARIMA Modells legt fest, wie oft die Zeitreihe im Vorhinein differenziert werden muss. Die Differenzierung muss so oft durchgeführt werden, bis die Zeitreihe stationär ist. Stationäre Zeitreihen haben laut Definition keinen Trend oder Saisonalität in sich verschlüsselt. Sie können jedoch einen Zyklus beinhalten, da diese keine Regelmäßigkeit darstellen. Es gibt mehrere Testverfahren, welche die Zeitreihe auf diese Eigenschaft untersuchen. Da der Parameter d manuell angegeben wird, muss im Vorhinein getestet werden, wie oft die Zeitreihe differenziert werden soll.

Um eine Vorhersage treffen zu können, muss folgende Gleichung nach y_t aufgelöst werden:

$$AR(p) = MA(q), \tag{20}$$

wobei das AR Modell die um d differenzierte Zeitreihe verwendet. Wenn $d > 0$ gilt, dann muss abschließend die Differenzierung rückgängig gemacht werden. Dieses Modell wird als $nARIMA(p,d,q)$ bezeichnet. Das sARIMA Modell hingegen besteht aus zwei AR und MA Modellen und wird mit $sARIMA(p,d,q)(P,D,Q,m)$ beschrieben, wobei m die Länge einer Saison angibt. Die Gleichung für sARIMA ist folgendermaßen definiert:

$$AR(p)\,AR(P) = MA(q)\,MA(Q) \tag{21}$$

Der zusätzliche Teil in diesem Modell betrachtet die Werte der Zeitreihe sowie Vorhersage-Fehler, welche vor genau einer Saison aufgezeichnet wurden. Dieser Teil bildet die Saisonalität des Modells ab.
Die Grundidee der ARIMA Modelle wurde hiermit übermittelt. Für detailreiche Erklärungen und Formeln kann das Buch von Rob J. Hyndman und George Athanasopoulos zur Recherche verwendet werden [Hyndman und Athanasopoulos 2018].

2.4 Evaluation der Vorhersage

Da jedes Vorhersage-Modell seine Vor- und Nachteile hat, muss entschieden werden, welches Modell sich am besten für die Vorhersage des gegebenen Datensatzes eignet. Ebenfalls besitzen einige Modelle, wie bei Holt-Winters erwähnt wurde, Hyperparameter, welche vor dem Start der Lern-Phase festgelegt werden müssen. Da diese manuell festgelegt werden, müssen sie auf den gegebenen Datensatz optimiert werden. Daher müssen alle möglichen Modelle bzw. Varianten von Modellen auf einem Test-Datensatz evaluiert werden, um herauszufinden, welches der Modelle für die Vorhersage genutzt werden soll.
Im nächsten Abschnitt werden Fehlermaße erläutert, mit denen der Fehler bzw. die Genauigkeit des Modells bestimmt wird. Der darauffolgende Abschnitt beschreibt Verfahren zur Erstellung von Trainings- und Test-Datensätze, auf welche die Maße angewandt werden.

2.4.1 Fehlermaße

Fehlermaße werden verwendet, um den Fehler einer Vorhersage zu berechnen. Über diesen Fehler lassen sich verschiedene Modelle miteinander vergleichen. In den folgenden Abschnitten wird auf drei verschiedenen Arten von Fehlermaßen eingegangen. Es handelt sich um den *skalierungsabhängigen*, den *prozentualen* und den *skalierten Fehler* [Hyndman und Koehler 2006; Chen und Yang 2004]. Für die folgenden Berechnungen wird der Vorhersage-Fehler e benötigt, welcher mit

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h} \tag{22}$$

berechnet wird, wobei die Vorhersage $\hat{y}_{T+h}$ mit einem Modell bestimmt wird, welches lediglich mit den Datenpunkten bis zum Zeitpunkt T gelernt wird. Der Test-Datensatz ist somit gegeben durch eine Menge mit Datenpunkten y_t zum Zeitpunkt $t > T$.

Skalierungsabhängige Fehler

Wie schon der Name sagt, sind diese Fehler abhängig von der Skalierung der Daten. Ein sehr simples Maß ist der *Mean Absolute Error* (MAE), welcher mit

$$MAE = \frac{\sum_{t=1}^{T} |e_t|}{T}, \tag{23}$$

berechnet wird, wobei T die Anzahl der vorhandenen Vorhersage-Fehler angibt. Ein großer Vorteil des MAE ist es, dass er sehr gut zu interpretieren ist, da er in der Skalierung der Daten angegeben wird. Dies bedeutet aber auch, dass ein Fehler von zum Beispiel 30 sehr gut ist, falls es sich um Werte im Millionenbereich handelt. In einem begrenzten Bereich von 0 bis 100 wäre dieser jedoch sehr schlecht. Daher sollten skalierungsabhängige Fehler nicht als Vergleich der Ergebnisse von unterschiedlich skalierten Zeitreihen verwendet werden.
Eine Alternative zu dem MAE ist der *Root Mean Squared Error* (RMSE). Dieser ist ähnlich zu dem MAE und wird wie folgt berechnet:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T} e_t^2}{T}} \tag{24}$$

Der RMSE wird öfter genutzt als der MAE, obwohl er schwieriger zu interpretieren ist. Der Grund dafür besteht darin, dass das Minimieren des MAE zu einer Vorhersage des Median-Wertes führt und das Minimieren des RMSE wiederum zur Vorhersage des Mittelwertes [Hyndman und Athanasopoulos 2018]. Im Normalfall ist es das Ziel, den Mittelwert vorherzusagen, wodurch der RMSE in den Vordergrund rückt.
Weitere skalierungsabhängige Fehler wären zum Beispiel der *Mean Square Error* oder der *Median Absolute Error*. Diese werden aber in dieser Arbeit nicht weiter behandelt.

Prozentuale Fehler

Der prozentuale Fehler p wird aus dem tatsächlichen Wert y und dem Vorhersage-Fehler e folgendermaßen berechnet:

$$p_t = 100 e_t / y_t \qquad \text{mit} \quad 1 \leq t \leq T \tag{25}$$

Aus diesem Fehler können mehrere Fehlermaße gebildet werden. Ein Beispiel ist der *Mean Absolute Percentage Error*, welcher analog zum MAE berechnet wird. Es existieren noch einige weitere Maße, welche hier nicht weiter erläutert werden. Ein Vorteil der prozentualen Maße ist, dass diese nicht abhängig von der Skalierung der Zeitreihe sind, wodurch Vergleiche zwischen Ergebnissen von Zeitreihen mit verschiedenen Skalierungen durchgeführt werden können. Allerdings weisen sie auch einige Probleme auf. Einerseits sind die prozentualen Fehler für Nullwerte nicht definiert und sie resultieren in extremen Werten, falls y sehr nahe an null liegt. Ebenfalls sind einige der existierenden Maße nicht symmetrisch, was in diesem Fall bedeutet, dass positive Fehler größer als negative Fehler gewichtet werden. Deshalb wurden symmetrische Maße eingeführt, welche dieses Problem kompensieren. Nichtsdestotrotz werden dadurch nicht alle Probleme der prozentualen Fehler gelöst, weshalb solche nicht in dieser Arbeit verwendet werden.

Skalierte Fehler

Als Alternative zu dem prozentualen Fehler kann der skalierte Fehler eingesetzt werden, um Ergebnisse der Vorhersagen von Zeitreihen mit unterschiedlichen Skalierungen zu vergleichen [Hyndman und Koehler 2006; Franses 2016]. Dieser ist wie folgt definiert:

$$q_t = \frac{e_t}{\frac{1}{T-1}\sum_{i=2}^{T}|y_i - y_{i-1}|} \tag{26}$$

Der Nenner berechnet die durchschnittliche absolute Abweichung zwischen zeitlich aufeinander folgenden Datenpunkten der tatsächlichen Zeitreihe y, wobei T der Anzahl der Datenpunkte in y entspricht. Zu beachten ist, dass der Zähler lediglich einmal berechnet werden muss, da dieser Zähler unabhängig von der Zeit t ist. Da Zähler sowie Nenner in der Skalierung der Zeitreihe liegen, wird diese durch die Division im skalierten Fehler q entfernt. Deshalb sind auf q aufbauende Fehlermaße für den Vergleich der Ergebnisse einer Vorhersage von Zeitreihen unterschiedlicher Skalierung verwendbar. Es gibt lediglich einen möglichen Fall, bei welchem der skalierte Fehler q nicht definiert ist. Dieser Fall würde auftreten, wenn alle Datenpunkte der Zeitreihe y den gleichen Wert beinhalten. In dieser Arbeit wird als skaliertes Fehlermaß der *Mean Absolute Scaled Error* (MASE) eingesetzt, welcher mit

$$MASE = \frac{\sum_{t=1}^{T}|q_t|}{T} \tag{27}$$

berechnet wird. Nach einer genaueren Betrachtung des Nenners von q wird klar, dass es sich um den MAE einer Vorhersage des Trainings-Datensatzes y mittels Naïve Modells handelt. Denn es gilt, dass der Vorhersage-Fehler des Naïve Modells und die Abweichung von zwei aufeinander folgenden Datenpunkten identisch sind, solange das Modell in jedem Zeitpunkt den Datenpunkt $t-1$ kennt. Deshalb ist der MASE ein Faktor, welcher beschreibt, um wie viel niedriger oder höher der Fehler e des verwendeten Modells zu dem Fehler des Naïve Modells auf den Trainings-Daten ist. Zusätzlich dazu, dass der MASE nicht abhängig von der Skalierung ist, hat er den Vorteil, dass er äußerst robust bezüglich Ausreißern ist. Analog zum MASE können auch der *Root Mean Squared Scaled Error* und der *Median Absolute Scaled Error* definiert werden, jedoch werden diese in der vorliegenden Arbeit nicht für die Evaluation verwendet.

2.4.2 Evaluations-Strategien

Um die Genauigkeit eines Modells zu prüfen, wird der Vorhersage-Fehler des Modells auf einem Datensatz inklusive der Zielvariable getestet. Dieser Datensatz darf nicht in das Training des Modells einfließen. Ansonsten hätte das Modell die Zielvariable schon in der Lern-Phase gesehen, was bei einer tatsächlichen Vorhersage nicht der Fall ist. Die Genauigkeit wäre somit in positive Richtung verfälscht, da der Vorhersage-Fehler geringer werden würde. Deswegen muss strikt dafür gesorgt werden, dass der *Trainings-Datensatz* und *Test-Datensatz* zwei voneinander getrennte Mengen sind.
Eine sehr simple, aber effektive Strategie ist es, den Original-Datensatz an einer festgelegten Stelle aufzuteilen. Die Test-Daten müssen mindestens so viele Datenpunkte beinhalten, wie in späterer Anwendung vorhergesagt werden sollen. Falls damit noch nicht 20% des Original-Datensatzes abgedeckt sind, kann

Abbildung 6: Beispiel einer Aufteilung eines Datensatzes in Trainings- und Test-Datensatz. Quelle: [Hyndman und Athanasopoulos 2018]

der Test-Datensatz auf ca. 20% vergrößert werden. Vorzugsweise sollten, wie in Abbildung 6 dargestellt, die zeitlich neuesten Daten verwendet werden, da diese am ehesten der Vorhersage des nächsten Zeitpunktes entsprechen, wodurch die Genauigkeit am aussagekräftigsten wird. Sobald beide Datensätze erstellt wurden, wird mit Hilfe des Trainings-Datensatzes das Modell gelernt. Anschließend wird für jeden Datenpunkt des Test-Datensatzes eine Vorhersage bestimmt. Im letzten Schritt wird mit einem ausgewählten Maß der Fehler zwischen tatsächlicher Zielvariable des Test-Datensatzes und der Vorhersage des Modells berechnet.
Eine anspruchsvolle Alternative hingegen ist eine Evaluations-Strategie, welche in einem Fachbuch als *Time Series Cross-Validation* bezeichnet wird [Hyndman und Athanasopoulos 2018]. In dieser Strategie werden mehrere Varianten von Trainings- und Test-Datensätzen erstellt. Ab einem festgelegten Zeitpunkt in dem Original-Datensatz wird nach jedem Datenpunkt ein Schnitt durchgeführt, welcher eine Variante von Trainings- und Test-Datensatz darstellt. Es muss ebenfalls festgelegt werden, wie viele Datenpunkte ein Test-Datensatz beinhaltet. Dies sollte, wenn möglich, davon abhängig gemacht werden, welche Anzahl Datenpunkten das Modell in späterer Anwendung vorhersagen soll. In Abbildung 7 wird eine Vielzahl von erstellten Varianten veranschaulicht, welche mit einem Test-Datensatz der Größe 1 erstellt wurden. Die grau dargestellten Punkte sind Datenpunkte, welche in dieser Variation nicht berücksichtigt werden, da für die Vorhersage lediglich Daten aus der Vergangenheit zur Verfügung stehen sollen.
Nachdem alle Varianten erstellt wurden, muss jeweils ein Modell mit dem Trainings-Datensatz gelernt werden. Mit diesem Modell werden alle Datenpunkte des jeweiligen Test-Datensatzes vorhergesagt, woraufhin der Fehler der Vorhersage mit einem festgelegten Maß berechnet wird. Als Gesamtfehler des Modells wird der Durchschnitt aller Fehler der Varianten gebildet.

2.5 Verwandte Arbeiten

Bevor im nächsten Kapitel erklärt wird, wie in dieser Arbeit die Vorhersage der Krankheitsfallzahlen optimiert wurde, wird hier zunächst auf wissenschaftliche Arbeiten eingegangen, welche sich mit dem gleichen oder einem ähnlichen Thema befasst haben.
Es existieren sehr viele veröffentlichte wissenschaftliche Arbeiten, welche sich mit dem Thema der Vorhersage von Zeitreihen beschäftigen. In den meisten Arbeiten werden ausgewählte Daten analysiert [zB.: Sarkar und Chatterjee 2017]. Zur Vorhersage werden einige Modelle betrachtet und die Ergebnisse dieser anschließend miteinander verglichen. In diesen Arbeiten werden die Modelle manuell auf den Datensatz angepasst und das Beste ermittelt. Da die Modelle in diesen Fällen auf die jeweiligen Datensätze zugeschnitten wurden, müssten diese, im Falle einer Datenänderung oder der Betrachtung neuer Datensätze, erneut angepasst werden.
Bei der vorliegenden Arbeit wurde jedoch darauf geachtet, dass das Ergebnis auf jegliche Zeitreihen von Krankheiten in FFM anwendbar ist. Der Grund dafür besteht darin, dass die spätere Anwendung, ohne

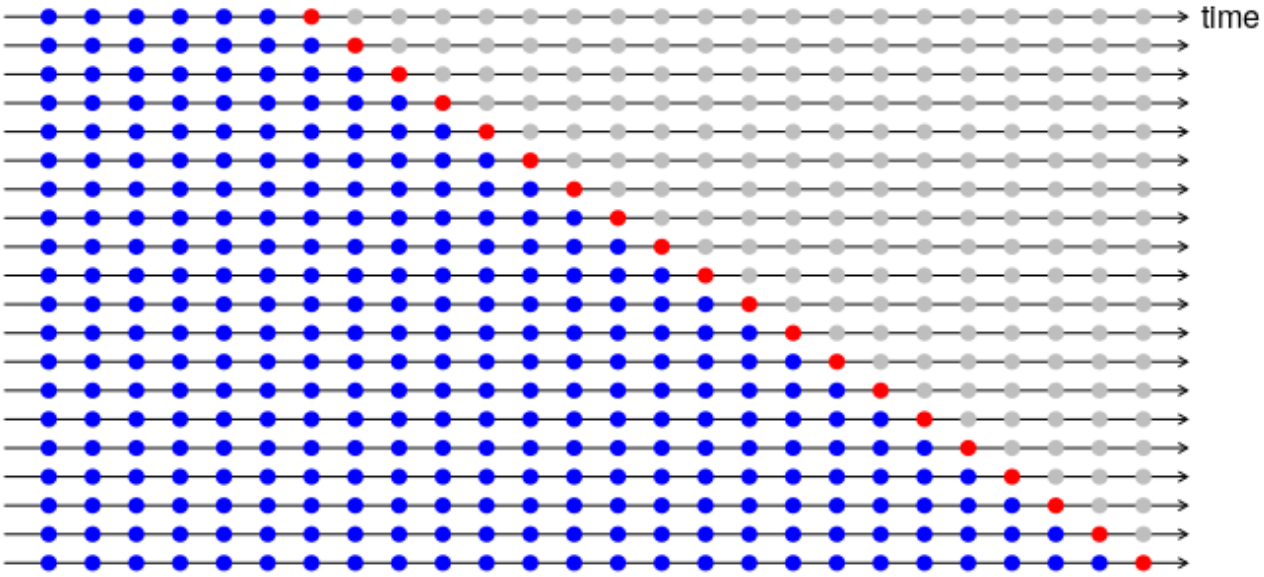

Abbildung 7: Beispiel einer Time Series Cross-Validation eines Datensatzes. In blau sind die Trainings-Datensätze, in rot die Test-Datensätze und in grau die verworfenen Datenpunkte für diese Variante abgebildet. Quelle: [Hyndman und Athanasopoulos 2018]

größeres Knowhow in Zeitreihen-Analyse, möglich sein soll. Zusätzlich kann sich der Verlauf der Krankheitsfälle über die Zeit stark verändern. Aus diesem Grund ist es erforderlich, dass die Anwendung mit solchen Veränderungen umgehen kann. Alternativ müssten die Parameter der spezifizierten Verfahren regelmäßig angepasst werden. Dies wurde ähnlich in dem *Automatic Statistician Projekt* [Steinruecken u. a. 2018] umgesetzt. Das Ziel dieser Ausarbeitung bestand darin, gegebene Daten automatisch zu bearbeiten, Vorhersagen zu erstellen und aus diesen Berichte zu generieren. Diese sollen, ohne viel Wissen im Bereich Informatik sowie Statistik, interpretierbar sein. Der Unterschied zu diesem Projekt besteht darin, dass die Vorhersage durch ein Modell erstellt wird, welches basierend auf den Daten individuell entwickelt wird. In der genannten Arbeit wurde für die Modell-Erstellung eine Sprache aus Gaußschen Prozessen entwickelt, aus welcher ein Modell gebaut wird, welches die vorhandene Zeitreihe bestmöglich annähert.

In der vorliegenden Arbeit wird mit unterschiedlichen und vordefinierten Modellen gearbeitet, aus denen die Besten gewählt werden. Der Vorteil ist, dass in dieser zusätzlich Verfahren zur Transformation und Dekomposition verwendet werden. Diese Vorgehensweise begünstigt eine weitere Optimierung der Vorhersage.

3 Vorhersage von epidemiologischen Zeitreihen

Um eine ideale Vorhersage treffen zu können, müssen zunächst die Daten auf die Aufgabe vorbereitet werden. Da einige der vorgestellten Vorhersage-Modelle Hyperparameter besitzen, welche im Vorhinein gewählt werden müssen, werden im zweiten Unterkapitel die Vorgehensweisen zur Bestimmung dieser Parameter verdeutlicht. Anschließend wird das Kernstück, der Vorhersage-Evaluator, der vorliegenden Arbeit beschrieben. Dieses bestimmt, mittels der zur Verfügung stehenden Modellen zur Vorhersage und Transformation, die bestmögliche Vorhersage der Zeitreihe. Im letzten Unterkapitel wird der Evaluator erweitert, sodass den Modellen mit mehreren Inputs zusätzliche Informationen übergeben werden.

3.1 Aufbereitung der Daten des RKI

Aufgrund der Tatsache, dass das Robert Koch-Institut die Daten äußerst gepflegt aufzeichnet, benötigen diese nahezu keine Aufbereitung. Wie bereits in Kapitel 2 erwähnt, werden in dieser Arbeit teilweise Datensätze verwendet, in denen Wochen ohne einen Krankheitsfall nicht vorhanden sind. Jeder Datensatz wird automatisch nach fehlenden Kalenderwochen überprüft, um daraus resultierende Probleme zu vermeiden. Die fehlenden Zeitpunkte werden anschließend mit Nullwerten aufgefüllt.

3.1.1 Generierung der Haupt-Zeitreihe

Die Daten vom RKI sind für eine Vorhersage nicht direkt verwertbar. Der Grund besteht darin, dass diese nach einigen Kriterien, wie in Tabelle 1 auf Seite 10 zu sehen ist, gefiltert sind. Die Daten wurden jedoch mit Absicht in dieser Form angefordert, da auf diese Weise zahlreiche Zeitreihen mit unterschiedlichen Kriterien erstellt werden können. Da das Ziel der Arbeit darin liegt, die Fallzahlen von FFM vorherzusagen, wird die Zeitreihe der Fallzahlen aller Bewohner in FFM als Grundbaustein benötigt. Diese wird in dieser Arbeit als *Haupt-Zeitreihe* bezeichnet.

Zur Erstellung dieser Zeitreihe werden alle Daten, welche nicht *Hessen* und *SK Frankfurt am Main* als Attribut besitzen, verworfen. Die resultierenden Datenpunkte betreffen lediglich FFM, jedoch sind diese weiterhin nach der Kalenderwoche, dem Alter und dem Geschlecht klassifiziert. Im Anschluss müssen daher alle Daten zu einer Kalenderwoche gruppiert bzw. aggregiert werden. Da die Fallzahlen absolute Zahlen sind, können diese einfach addiert werden und als Ergebnis für die jeweilige Woche übernommen werden. Die Berechnung der Inzidenz fällt etwas komplexer aus, da es sich um eine relative Angabe zur Größe der betrachteten Gruppe handelt. Aus der Fallzahl und Inzidenz eines Datenpunktes lässt sich die Größe der zugehörigen betrachteten Gruppe berechnen. Wenn also mehrere Datenpunkte aggregiert werden sollen, muss die Größe der darausfolgenden Gruppe aufsummiert werden, um damit die neue Inzidenz berechnen zu können. In der folgenden Gleichung ist die Berechnung der Inzidenz für eine bestimmte Woche definiert:

$$Inzidenz_{aggr,t} = \frac{\sum_{i=1}^{N} Fallzahl_{i,t}}{\sum_{i=1}^{N} \frac{Fallzahl_{i,t}}{Inzidenz_{i,t}}} \qquad \text{mit} \quad 1 \leq t \leq T \tag{28}$$

wobei N die Anzahl der Datenpunkte zum Zeitpunkt t darstellt, welche aggregiert werden sollen. Der Zähler berechnet die Summe der Fallzahlen und der Nenner die Summe der Größe der betrachteten Gruppen. Da der Bruch in der Summe des Nenners bereits in der Skalierung pro 100.000 ist, muss der Nenner für die Berechnung der Inzidenz nicht mehr skaliert werden.
Erst nachdem zu jedem Zeitpunkt alle Daten aggregiert wurden, erfolgt die Prüfung auf fehlende Wochen und die etwaige Füllung der Lücken mit Nullwerten. Hiermit wäre die Zeitreihe, für welche in den folgenden Unterkapiteln eine Vorhersage getroffen werden soll, fertig generiert.

3.1.2 Zusätzliche Datenauswertung

In Abschnitt 3.4 wird die Vorhersage erweitert, indem die Modelle zusätzliche Informationen geliefert bekommen. Diese sollten auf irgendeine Weise Korrelationen zu der Zeitreihe von FFM aufweisen, da die zusätzliche Information ansonsten keinen Mehrwert für die Vorhersage schaffen würde. Daher werden in dieser Arbeit Zeitreihen der Land- und Stadtkreise von Frankfurts Nachbarn generiert, welche in einer gegebenen *Reisezeit* mit einem Auto oder öffentlichen Verkehrsmittel erreichbar sind. Je größer diese Reisezeit gewählt wird, desto mehr Zeitreihen werden erzeugt. Die Vermutung besteht darin, dass mögliche Veränderungen in der Zeitreihe von FFM mit Hilfe der Daten aller Nachbarn früher erkannt werden können. Aufgrund der kurzen Reisezeit zwischen den Orten sollten Abhängigkeiten bzw. Korrelationen zwischen den Zeitreihen erkennbar sein, welche von den Vorhersage-Modellen ausgenutzt werden können. In Abbildung 10 auf Seite 40 sind zwei Koordinatensysteme dargestellt, welche jeweils die Zeitreihen von FFM und der Nachbar-Kreise beinhalten. Die Zeitreihe der Nachbarn bildet sich in diesem Beispiel aus allen Kreise, welche innerhalb von 60 Minuten erreichbar sind. In der oberen Darstellung ist die Fallzahl und in der unteren die Inzidenz abgebildet. Die Fallzahlen der Nachbar-Kreise, weisen einen kontinuierlich höheren Wert auf. Dies ist darin zu begründen, dass die betrachtete Gruppe ca. um das 6-fache größer ist. An der Inzidenz ist ersichtlich, dass die Ansteckrate der beiden Gruppen in Relation zur Einwohnerzahl in etwa gleich hoch ist.

3.2 Einstellung der Vorhersage-Modelle

Die Vorhersage-Modelle werden dahingehend erweitert, dass die Hyperparameter automatisch auf den Datensatz zugeschnitten werden. Dies gewährleistet, dass die einzelnen Modelle nicht per Hand auf die Zeitreihen optimiert werden müssen. Umgesetzt wird dies zum Teil mit eigenen Implementierungen und auch mit vorgefertigten Verfahren des R-Pakets *forecast*[6].

3.2.1 Regressions-Modelle

Die Lineare Regression besitzt grundsätzlich keine Hyperparameter. In der Random Forest Regression hingegen, muss die Anzahl an Bäumen im Vorhinein festgelegt werden. Diese ist in dieser Arbeit auf einen bestimmten Wert festgelegt, auf welchen in dem Kapitel der Auswertung eingegangen wird. Zusätzlich müssen zu Beginn die Vorhersage-Variablen der Modelle ausgewählt werden. Da in der ersten Phase dieser Arbeit lediglich die Zeitreihe von FFM zur Vorhersage verwendet wird, müssen die Vorhersage-Variablen mit Informationen aus dieser gefüllt werden. Deshalb werden, analog zur Glei-

[6] https://cran.r-project.org/web/packages/forecast/forecast.pdf

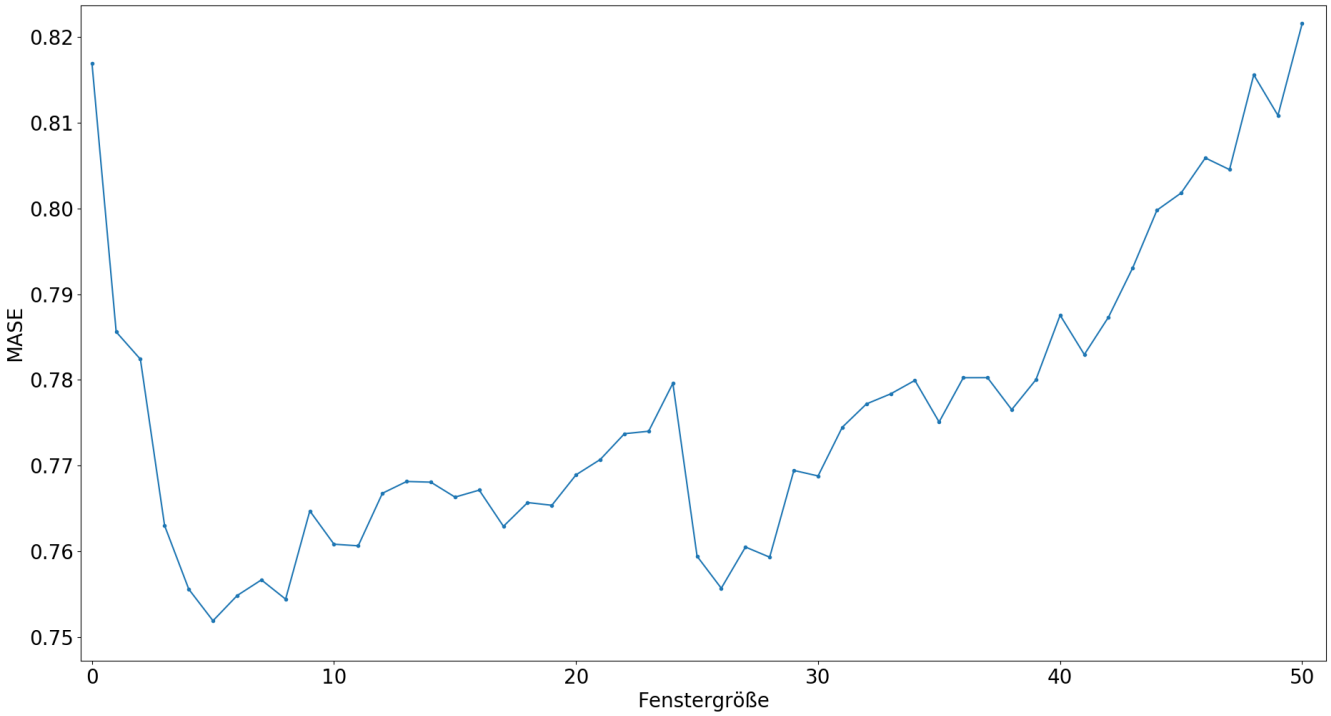

Abbildung 8: Entwicklung des MASE bei zunehmender Fenstergröße.

chung des Autoregressiven Modells auf Seite 23, die zuletzt observierten Werte als Vorhersage-Variablen genutzt.

Der Versuch die beste Anzahl der Vorhersage-Variablen anzunähern ist gescheitert, da der Fehler mit der Zunahme an Variablen nicht monoton steigt bzw. fällt. Daher wird eine Vielzahl an verschiedenen Modellen erstellt, welche alle mit in die Evaluation einfließen. In Abbildung 8 ist die Entwicklung des MASE im Bezug zur Fenstergröße dargestellt. Dieses Fenster beschreibt, wie viele vergangene Werte der Zeitreihe mit in die Regression einfließen. Es ist klar zu erkennen, dass einige kleine, aber auch ein großes lokales Minima vorhanden sind, in welchem sich das Optimierungsverfahren festsetzen könnte.

3.2.2 Exponential Smoothing

In der Evaluation werden zwei verschiedene Varianten zur Suche des besten ES-Modells verwendet. Als Grundlage wird jeweils die übergebene Zeitreihe bis Zeitpunkt t verwendet. Gesucht werden in der ersten Variante die Hyperparameter α, β und γ eines Holt-Winters Modells (ES-HW). Hierzu wird die Parameter-Kombination ermittelt, wodurch der Vorhersage-Fehler auf der gegebenen Zeitreihe minimal wird. Als Fehlermaß wird der MASE genutzt. Um alle Vorhersage-Fehler zu bestimmen zu können, muss zu jedem Zeitpunkt t und jeder Hyperparameter-Kombination das Level, der Trend und die Saisonalität berechnet werden. Aus diesen drei Komponenten lassen sich die Vorhersage und der zugehörige Fehler ermitteln. Da sich das Modell in den ersten Vorhersagen anfänglich auf den Datensatz einpendeln muss, werden die Vorhersage-Fehler der ersten Saison nicht in die Berechnung des MASE mit einbezogen. Die erste Saison wird demzufolge allein als Training verwendet. Alle darauffolgenden Datenpunkte werden zur Berechnung des MASE und zur weiteren Optimierung des Modells genutzt.

Die zweite Variante greift auf das oben genannte R-Paket zurück (ES-R). Diese ist nicht auf ein bestimmtes ES-Modell beschränkt, sondern schätzt auf Basis der gegebenen Zeitreihe ab, welches Modell am

Tabelle 3: Alle Kombinationen der Parameter d und D, welche in dieser Arbeit in der sARIMA Variante verwendet werden.

Parameter	Variante 1	Variante 2	Variante 3	Variante 4	Variante 5	Variante 6
d	0	1	0	1	2	2
D	0	0	1	1	0	1

besten für die Vorhersage geeignet wäre. Zur Auswahl stehen alle in Kapitel 2 genannten Varianten des Exponential Smoothing. Zusätzlich werden die Hyperparameter automatisch mit einem Optimierungsverfahren des Pakets bestimmt.

Werden die Varianten objektiv miteinander verglichen, sollte die Erste in der Qualität der Vorhersage keine Vorteile aufweisen. Dieser Verdacht begründet sich darin, dass sich die zweite Variante theoretisch noch besser auf den Datensatz optimieren kann. In Kapitel 4 zeigt sich jedoch in der Auswertung, dass die Holt-Winters Variante Vorteile birgt, welche auf den ersten Blick nicht sichtbar sind.

3.2.3 ARIMA

Für das ARIMA Modell werden in dieser Arbeit drei Varianten zur Auswahl der Modell-Parameter verwendet. Zunächst werden jedoch die Parameter d und ggf. D bestimmt, da diese Wahl unabhängig von den drei Varianten ist. Analog zum Exponential Smoothing wird auch hier zur Bestimmung der Parameter die komplette Zeitreihe verwendet.

Wie bereits in den Grundlagen erwähnt wurde, muss zur Bestimmung des Parameters d geprüft werden, wie oft die Zeitreihe differenziert werden muss, damit sie stationär wird. Zwei bekannte Verfahren, welche eine Zeitreihe auf Stationarität testen, sind der *Augmented Dickey-Fuller* Test und der *Kwiatkowski-Phillips-Schmidt-Shin-Test* (KPSS-Test). In dieser Arbeit wird der KPSS-Test angewandt, welcher als Signifikanztest aufgebaut ist. Bei diesem Test wird die Stationarität der Zeitreihe als Nullhypothese angenommen. Um d zu bestimmen, wird abwechselnd die Zeitreihe auf Stationarität getestet und, sofern die Nullhypothese verworfen wurde, differenziert. Sobald die Hypothese der Stationarität nicht mehr verworfen wird, nimmt d als Wert die Anzahl der benötigten Differenzierungen an.

Falls jedoch aufgrund eines sARIMA Modells zusätzlich der Parameter D benötigt wird, muss eine andere Strategie verfolgt werden. Hierzu wird zusätzlich überprüft, ob eine *saisonale Differenzierung* der Zeitreihe Stationarität hervorruft. Eine solche Differenzierung berechnet die Abweichung zum Wert an der gleichen Position der letzten Saison. In Tabelle 3 sind alle Werte-Kombinationen der Parameter abgebildet, welche in der vorliegenden Reihenfolge getestet werden. Der Wert des jeweiligen Parameters spiegelt die Ordnung der normalen bzw. saisonalen Differenzierung wider, welche durchgeführt werden, bevor der KPSS-Test durchgeführt wird. Sobald der Test die Nullhypothese nicht mehr verwirft, werden die Parameter d und D dieser Kombination für die Modellbildung genutzt.

In den folgenden Erklärungen der Varianten müssen ARIMA Modelle mit verschiedenen Parametern verglichen werden. Hierfür wird das *Akaike's Information Criterion* (AIC) verwendet, welches das Modell bezüglich Qualität und Komplexität bewertet. Es muss darauf geachtet werden, dass der AIC nur im Vergleich von Modellen verwendet wird, welche mit der identischen Zeitreihe gelernt wurden. Dies bedeutet auch, dass ARIMA Modelle, welche verschiedene Werte für d oder D nutzen, nicht mit dem AIC

Tabelle 4: Modell-Konfigurationen, welche bei dem Hyndman-Khandakar Algorithmus als Erstes geprüft werden

Initial-Modell	p	q	P	Q
Nr. 1	0	0	0	0
Nr. 2	0	1	0	1
Nr. 3	2	0	2	0
Nr. 4	2	2	1	1
Nr. 5	4	0	0	0
Nr. 6	4	0	2	0

verglichen werden dürfen. Der AIC wird daher für die Wahl der übrigen Parameter des ARIMA bzw. sARIMA Modells verwendet.

Variante 1: nARIMA Modell

Die erste Variante ist auf das nARIMA(p,d,q) Modell beschränkt. Deshalb müssen lediglich die Parameter p und q bestimmt werden, wobei diese in der vorliegenden Arbeit auf $0 \leq p \leq 4$ und $0 \leq q \leq 2$ eingeschränkt werden. Die Vorgehensweise ist sehr simpel gehalten, da die Berechnung eines Modells wenig Zeit beansprucht. Es wird zu jeder Parameter-Kombination ein Modell erstellt und der zugehörige AIC ermittelt. Das Modell, welches den niedrigsten AIC aufweist, wird für die Vorhersage ausgewählt.

Variante 2: sARIMA Modell

In dieser Variante wird nach einem passenden sARIMA$(p,d,q)(P,D,Q,m)$ Modell gesucht. Da die Berechnung für ein sARIMA Modell um einiges länger dauert, wird in diesem Fall der *Hyndman-Khandakar Algorithmus* [Hyndman und Khandakar 2008] implementiert, welcher sich rekursiv an das beste Modell annähert. Die Parameter sind in dieser Arbeit auf $0 \leq p,q \leq 4$ und $0 \leq P,Q \leq 2$ beschränkt. Die Idee des Algorithmus besteht darin, dass zu Beginn der AIC von sechs Modellen mit vordefinierten Parametern ermittelt wird. Diese Parameter sind in Tabelle 4 zu finden. Für den nächsten Schritt wird das Modell mit dem besten AIC als Grundlage verwendet. Aus diesem werden einige neue Modellvariationen erstellt, für welche erneut der AIC berechnet wird. Dieser Vorgang wird wiederholt, bis keine weitere Verbesserung beim AIC auszumachen ist und das Modell zur Vorhersage somit feststeht.

Bei der Erstellung der Modellvariationen werden maximal 12 neue Modelle generiert, für welche der AIC berechnet werden muss. Die Änderung in den Variationen betreffen entweder einen der vier Parameter, oder jeweils die zwei Parameter des saisonalen bzw. nicht-saisonalen Teils. Die betroffenen Parameter werden je um eins inkrementiert sowie dekrementiert. Falls einer der Parameter dadurch außerhalb seines Wertebereichs liegen sollte, wird diese Variation verworfen.

Variante 3: (s)ARIMA Modell

Die dritte Variante verwendet ebenfalls das oben genannte R-Paket (ARIMA-R). Ein Unterschied zu den vorherigen Varianten liegt darin, dass diese Methode automatisch die Parameter d und D bestimmt. Ebenfalls ist diese nicht auf ARIMA oder sARIMA beschränkt. Das Verfahren bestimmt das Modell und alle zugehörigen Parameter automatisch. Dadurch ist diese Funktion sehr flexibel und auf viele unterschied-

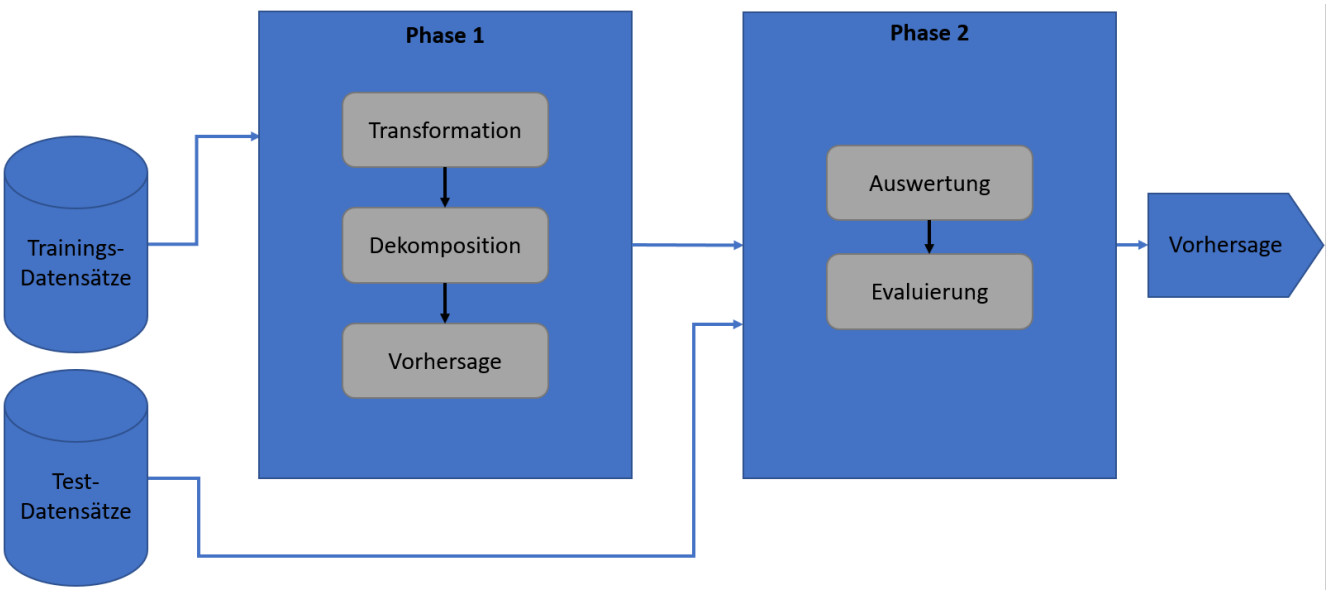

Abbildung 9: Abstaktions-Darstellung des Vorhersage-Evaluators.

liche Zeitreihen mit verschiedenen Mustern anwendbar. Aufgrund dessen müssen jedoch zusätzlich viele Abschätzungen und Annahmen getroffen werden, damit die Laufzeit der Variante nicht zu groß ist.

3.3 Vorhersage-Evaluator

In diesem Unterkapitel wird der Vorhersage-Evaluator beschrieben. Dieser Algorithmus kombiniert alle Verfahren, Modelle und Strategien, welche in dieser Arbeit bisher erläutert wurden. Das Ziel von diesem besteht darin, zu einer gegebenen Zeitreihe die bestmögliche Vorhersage des nächsten Zeitpunktes zu bestimmen. Der Evaluator besteht aus zwei Phasen. In in der ersten Phase werden die Vorhersage-Ergebnisse erstellt. Die zweite Phase beinhaltet die Auswertung sowie Evaluation der Ergebnisse und liefert letztendlich die Vorhersage der gegebenen Zeitreihe. In Abbildung 9 ist der Aufbau des Evaluators abstrahiert dargestellt. Die erste Phase wird mehrfach aufgerufen, da mehrere Paare an Trainings- und Test-Datensätzen bearbeitet werden. Die genaue Beschreibung der Erstellung von Datensätzen sowie der Phasen wird in den nächsten Abschnitten erläutert.

3.3.1 Vorhersage der Zeitreihen und deren Variationen

Bevor die Variationen der Zeitreihen erstellt werden können, wird eine Zeitreihe benötigt, aus der diese hervorgehen. Aus diesem Grund muss die gegebene Zeitreihe als erstes in Trainings- und Testdatensatz aufgeteilt werden. Hierbei fließt die Grundidee der Time Series Cross-Validation mit ein. Es wird zunächst, wie in Abschnitt 2.4.2, eine Menge von Zeitreihen inklusive einem Test-Datenpunkt erstellt. In dieser Arbeit wird die Anzahl der Trainings- und Testdatensätze auf 100 Paare festgelegt. Für eine Evaluation eines Modells müssten also 100 Modelle trainiert werden, welche jeweils einen Zeitpunkt vorhersagen. Bevor jedoch von einer Evaluation gesprochen werden kann, sind noch einige Vorarbeiten notwendig. Im Folgenden wird erläutert, wie mit einem Paar von Trainings- und Test-Datensatz in der ersten Phase des Evaluators umgegangen wird, wobei der Datenpunkt zum Testen vorerst nicht verwendet wird. Dieser kommt in der Evaluation der Vorhersage-Ergebnisse zum Einsatz.

Im nächsten Schritt der Abstraktion in Abbildung 9, also am Anfang der ersten Phase, werden die Variationen der Zeitreihe eines Trainings-Datensatzes erstellt. Diese können mit zwei unterschiedlichen Arten von Verfahren erstellt werden. Die Zeitreihe kann entweder transformiert, oder mit einer Dekomposition zerlegt werden. Diese Verfahren haben wiederum variable Parameter, wodurch eine Vielzahl von Variationen mit einem einzelnen Verfahren erstellt werden kann. Zusätzlich können die Verfahren kombiniert werden, indem die Zeitreihe zuerst transformiert und daraufhin zerlegt wird. Demzufolge entsteht eine große Anzahl an Kombinationen von Transformations- und Dekompositions-Verfahren sowie die daraus resultierenden Variationen der Zeitreihe des Trainings-Datensatzes. Die Original-Zeitreihe zählt ebenfalls als eine Variation, da in den Kombinationen zusätzlich akzeptiert wird, dass eins oder sogar keins der Verfahren angewandt wird.

Diese Kombinationen werden auf jeden vorhandenen Trainings-Datensatz angewandt. Somit besitzt jeder Datensatz Variationen der gleichen Art. Es ist zu beachten, dass eine Variation aufgrund der Dekomposition aus bis zu drei Zeitreihen bestehen kann.

Die letzte Aufgabe der ersten Phase besteht darin, eine Vorhersage für alle Zeitreihen-Variationen zu ermitteln. In den Grundlagen wurden mehrere Vorhersage-Modelle vorgestellt. Zusätzlich wurden einige dieser, im letzten Unterkapitel 3.2, um ein paar Varianten erweitert. Hier werden nun all diese Modelle dazu verwendet, den nächsten Zeitpunkt einer Zeitreihen-Variation zu bestimmen. Wie bereits erwähnt, kann eine Variation aus mehreren Zeitreihen bestehen. In diesem Fall wird für jede dieser Zeitreihen eine Vorhersage durchgeführt.

Mit dem Abschluss der ersten Phase des Evaluators liegen für jede Kombination von Transformations- und Dekompositions-Verfahren zu jedem Test-Datenpunkt mehrere Vorhersagen vor. Diese Vorhersagen bestehen aus den Ergebnissen von allen vorhandenen Modellen.

3.3.2 Evaluation der Ergebnisse

Nachdem alle Vorhersagen erstellt wurden, muss herausgefunden werden, welche Kombination von Transformations- und Dekompositions-Verfahren mit zugehörigem Vorhersage-Modell das höchste Potential zu einer guten Vorhersage hat. Bevor hier eine endgültige Entscheidung getroffen werden kann, müssen erst andere Aspekte in Augenschein genommen werden.

Aktuell ist die Vorhersage einer Variation, bei welcher ein Dekompositions-Verfahren genutzt wurde, auf mehrere vorhergesagte Zeitreihen aufgeteilt. Diese Zeitreihen wurden aber ebenfalls von allen Vorhersage-Modellen erstellt. Hinsichtlich dieses Aspektes muss entschieden werden, welche der Vorhersagen aufsummiert werden, um die Ursprungs-Zeitreihe zu erhalten. Intuitiv würde der Vorhersage-Fehler der Modelle jeder einzelnen Dekompositions-Komponenten betrachtet, und anschließend die Vorhersagen mit dem niedrigsten Fehler kombiniert werden. Bei diesem Ansatz wird jedoch nicht beachtet, dass der Vorhersage-Fehler von zwei Vorhersagen unterschiedlicher Komponenten, unabhängig der Größe des Fehlers, in der Summe dieser nahezu verschwinden kann. Deshalb werden alle Zusammenstellungen von Vorhersage-Modellen in der Addition der Dekompositions-Komponenten in Betracht gezogen.

Bevor die letztendliche Evaluation durchgeführt werden kann, müssen die transformierten Zeitreihen wieder in die Original-Skalierung zurückgebracht werden. Darunter fallen auch wieder zusammengefüg-

te Dekompositions-Komponenten, welche vor der Dekomposition transformiert werden. Die Gleichungen der Rück-Transformationen wurden in den Grundlagen ausgearbeitet.

Nachdem all diese Schritte durchgeführt wurden, besitzt jede Kombination von Transformations- und Dekompositions-Verfahren eine Menge von Vorhersagen der Haupt-Zeitreihe. Aus dieser Menge an Vorhersagen unterschiedlicher Modelle, sowie verschiedener Modellkombination der Zusammenführung einer möglichen Dekomposition, wird jeweils die beste Vorhersage ermittelt. Zum Vergleich können die Fehlermaße MAE, RMSE und MASE verwendet werden. Die skalierungsabhängigen Fehlermaße können hier angewandt werden, da sich alle Vorhersagen auf die gleiche Zeitreihe beziehen. Wie sich in dem Kapitel der Auswertung zeigen wird, bestätigt sich die Aussage in den Grundlagen, dass die Minimierung des MAE und des RMSE zu verschiedenen Ergebnissen führt. Die Wahl des Fehlermaßes bestimmt, ob Verfahren ausgewählt werden, welche Ausreißer innerhalb der Daten mehr oder weniger in das Modell einbeziehen. Dies kann erreicht werden, da die Minimierung des MAE zur Vorhersage des Median-Wertes und die Minimierung des RMSE zur Vorhersage des Mittelwertes führt. Falls in den Daten überwiegend Ausreißer in positiver Richtung vorhanden sind, würde dies in der Vorhersage des Mittelwertes, im Vergleich zum Median-Wert, auffallen. Aufgrund der Tatsache, dass es sich hier um zwei interessante Fälle handelt, werden diese abschließend in Kapitel 4, der Auswertung, betrachtet und miteinander verglichen.

Abschließend muss die Kombination von Transformations- und Dekompositions-Verfahren ermittelt werden, deren Vorhersage den niedrigsten Fehler aufweist. Hierzu wird erneut das im vorherigen Schritt gewählte Fehlermaß verwendet, da der Einsatz zwei unterschiedlicher Maße zu einem schlechteren Ergebnis führen würde. Der Grund hierfür liegt darin, dass ein Modell, welches bezüglich des zweiten Fehlermaßes das Beste gewesen wäre, möglicherweise bereits im ersten Schritt verworfen wurde. Als Ergebnis wird die beste Vorhersage für alle Test-Datenpunkte bezüglich des Fehlermaßes angegeben. Die zugehörigen Modelle, sowie die Kombination von Transformations- und Dekompositions-Verfahren, werden anschließend für die Vorhersage des nächsten Zeitpunktes verwendet.

3.4 Erweiterung um zusätzliche Inputs

Bisher betrachtet der Evaluator lediglich die Zeitreihe aus FFM. In diesem Unterkapitel wird dieser mit weiteren Funktionalitäten ausgestattet, sodass in den Modellen mit mehreren Inputs zusätzliche Informationen verarbeitet werden können. Mehrere Inputs besitzen lediglich die beiden Regressions-Modelle. Die anderen vorgestellten Vorhersage-Modelle verarbeiten für die Vorhersage maximal eine Information pro Zeitpunkt. Wie bereits in Abschnitt 3.1.2 erwähnt, werden die Nachbar-Kreise von FFM verwendet, um zusätzliche Informationen bereitzustellen. Im folgenden Abschnitt wird erläutert, wie diese Informationen generiert werden.

3.4.1 Generierung der Informationen

Das Ziel dieses Abschnittes besteht darin, Informationen zu generieren, welche in irgendeiner Weise Korrelationen zu der Haupt-Zeitreihe aufweisen. Diese zusätzlichen Informationen werden für die Vorhersage im Evaluator den Regressions-Modellen übergeben. In Abbildung 9 würden die zusätzlichen Informationen, ähnlich wie die Trainings-Datensätze, von außerhalb in die erste Phase einfließen. Der

Unterschied liegt jedoch darin, dass diese direkt in die Vorhersage übertragen werden, anstatt vorher noch die Transformation und Dekomposition zu durchlaufen.
Die Zeitreihe der Nachbar-Kreise wird mittels Angabe der Reisezeit erstellt, welche ein variabler Parameter ist. Mit diesem können einige unterschiedliche Zeitreihen gebildet werden, wodurch die Wahrscheinlichkeit erhöht wird, dass eine dieser Korrelationen zu der von FFM aufweist. Die erste mögliche Variante einer zusätzlichen Informationen wäre damit schon geschaffen. In dieser würde den Regressions-Modellen zusätzlich der Wert des letzten Datenpunktes der Nachbar-Zeitreihe übergeben werden. Alternativ können auch mehrere Datenpunkte der Nachbarn übergeben werden.
Eine weitere Idee wäre, die Steigung bzw. Differenzen der Nachbar-Zeitreihe zu berechnen und diese als zusätzliche Information zu übermitteln. Diese könnten normal oder relativ berechnet werden. Die normale Variante wird, so wie in den ARIMA Modellen, mit der Differenz zwischen aufeinander folgenden Datenpunkten dargestellt. In der relativen Variante hingegen werden Divisionen statt Subtraktionen verwendet. Hiermit wird das Ergebnis skalierungsunabhängig, weshalb keine Probleme durch die unterschiedlichen Skalierungen entstehen sollten. Die relative Differenzierung besitzt jedoch einen Nachteil. Dieser besteht darin, dass sie nicht definiert ist, falls der letzte Wert gleich null ist. Daher wird in diesem Fall die normale Differenz angenommen, was durch eine Erhöhung des Nullwertes um eins erreicht wird. Da die Zeitreihen in den wenigsten Fällen auf null herabfallen, kann über diese Abweichung hinweggesehen werden. Zusätzlich handelt es sich lediglich um eine Zusatzinformation, welche zur Verbesserung der Vorhersage beitragen soll. Den Regressions-Modellen würde in dieser Variante, zur Vorhersage des nächsten Zeitpunktes, die Differenz zwischen den letzten beiden Datenpunkten der Nachbar-Zeitreihe übergeben werden.
Die zuvor erläuterte Idee kann noch weiter vertieft werden. Hierzu werden nicht mehr alleinig die Daten der Nachbarn betrachtet, sondern es werden Bezüge zwischen der Haupt- und Nachbar-Zeitreihe hergestellt und aus diesen Informationen generiert. Da es sich bei den Zeitreihen um unterschiedliche Skalierungen handelt, werden hier die Werte der Inzidenz zur Hand genommen. In Abbildung 10 sind im unteren Koordinatensystem Inzidenz der beiden Zeitreihen dargestellt. Die Vermutung ist, dass eine Abweichung von Nachbar- zu Haupt-Zeitreihe ein Indiz für eine kommende Änderung in der Zeitreihe von FFM darstellen könnte. Daher wird in dieser Variante die normale Abweichung zwischen der Nachbar- und Haupt-Zeitreihe berechnet, indem die Differenz der Werte des gleichen Zeitpunktes gebildet wird. In den Regressions-Modellen wird zur Vorhersage zusätzlich die Abweichung des letzten Zeitpunktes als Vorhersage-Variable verwendet.
All diese Informationen können zusätzlich, einzeln oder gebündelt, dem Evaluator übergeben werden. Die Regressions-Modelle werden daraufhin mit dementsprechenden Vorhersage-Variablen erweitert. Die Erweiterung des Evaluators ist nicht auf diese Informationen begrenzt. Es wird jede Art von numerischen Daten akzeptiert, solange für jeden Zeitpunkt der Test-Daten ein zugehöriger Wert vorhanden ist.

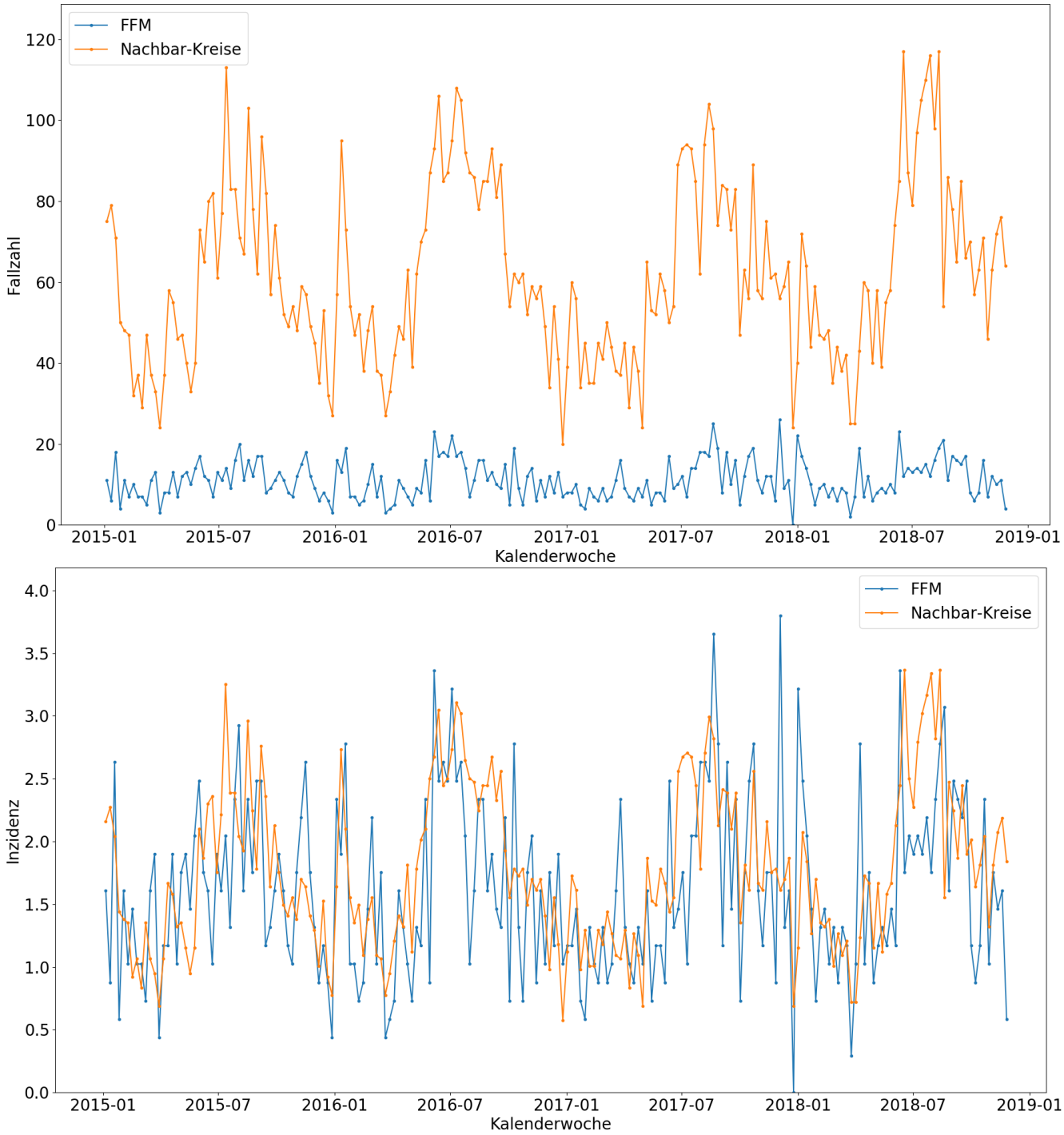

Abbildung 10: Beispieldaten der Krankheit Campylobacter-Enteritis von FFM und der Nachbar-Kreise von FFM, welche in 60 Minuten zu erreichen sind. Die Obere Grafik zeigt die absolute Fallzahl und die untere die zur Einwohnerzahl relative Inzidenz an.

3.5 Ausblick: Langzeit-Simulation von epidemiologischen Zeitreihen

Das bisherige Ziel bestand immer darin, den nächsten Zeitpunkt bestmöglich zu bestimmen. In diesem Unterkapitel wird die Idee für eine zusätzliche Erweiterung vermittelt, welche von diesem Ziel abweicht. Diese Idee wurde in dieser Arbeit nicht weiterverfolgt, da die Bearbeitung dieser über die vorgegebene verfügbare Zeit hinaus gegangen wäre. Trotzdem wird die Ergänzung in diesem Abschnitt festgehalten.
Bisher wurden lediglich die Punkt-Vorhersagen der Modelle betrachtet. In der folgenden Erweiterung werden erstmalig die Vorhersage-Intervalle verwendet. Das Ziel besteht darin, eine gegebene Anzahl an Zeitpunkten zu simulieren bzw. vorherzusagen. In diesem Fall soll nicht wie bisher eine Punkt-Vorhersage, sondern ein Bereich bestimmt werden, in dem der Wert zu einer bestimmten Wahrscheinlichkeit liegen wird. Um dies zu realisieren, müssen, wie bereits zuvor erwähnt, die Vorhersage-Intervalle bzw. die Verteilungsfunktion, welche sich hinter diesen verbirgt, verwendet werden. Wie in dem Kapitel der Grundlagen erwähnt wurde, gibt die Punkt-Vorhersage den Wert an, der nach dieser Verteilung am wahrscheinlichsten auftreten würde.
Zur Bestimmung der genannten Wahrscheinlichkeiten wird ein Verfahren der Stochastik angewandt. Dieses ist unter anderem bekannt als *Monte-Carlo-Simulation* (MC-Simulation) [Raychaudhuri 2008]. In der MC-Simulation wird ein bestimmtes Zufallsexperiment extrem oft wiederholt. Aus den Ergebnissen dieser können zum Beispiel Zustände geschätzt oder Verteilungsfunktionen ermittelt werden. In dieser Arbeit würde das Zufallsexperiment der Vorhersage der nächsten Zeitpunkte entsprechen. Es müsste zunächst festgelegt werden, wie oft die Vorhersage durchgeführt werden soll. Aus den daraus resultierenden Ergebnissen werden Bereiche bestimmt, in denen die Vorhersage zu einer bestimmten Wahrscheinlichkeit liegen wird. Diese Bereiche werden Konfidenzintervalle genannt. Würde also die Punkt-Vorhersage in dem Zufallsexperiment verwendet werden, würde es sich nicht mehr um ein Zufallsexperiment handeln, da die Vorhersage in jedem Durchlauf die gleiche wäre.
In dieser Arbeit wurden einige Vorhersage-Modelle vorgestellt. Da die Standard-Modelle, im Vergleich zu den übrigen Modellen, keine Verteilungsfunktion angeben können, würde dies in diesem Fall zu Problemen führen. Sie könnten jedoch mit einem normalverteilten Rauschen ausgestattet werden. Hierzu müssten detailliertere Ideen ausgearbeitet werden. In den Iterationen der MC-Simulation dürften die Vorhersage-Modelle nicht den wahrscheinlichsten Wert angeben, sondern müssten einen Zufallswert mit der Verteilungsfunktion bestimmen. Dieser Zufallswert wird dann für die Erstellung des nächsten Modells zur Vorhersage des nächsten Zeitpunktes verwendet. Mit dieser Vorgehensweise wird eine große Menge an unterschiedlichen Zeitreihen erstellt, welche für die Bestimmung der Konfidenzintervalle eingesetzt werden.
Die Simulation würde zum Beispiel mit der besten Kombination an Transformation und Dekomposition aus dem Vorhersage-Evaluator erstellt werden. Hierbei muss beachtet werden, dass, im Falle einer Dekomposition, die Simulation für alle Komponenten durchgeführt werden muss.

Tabelle 5: Alle Parameterkonfigurationen der Transformations- und Dekompositions-Verfahren, welche in der Auswertung verwendet wurden.

Transformation		Dekomposition	
Verfahren	Parameter	Verfahren	Parameter
Original	-	Original	-
ABC-Transformation	0.0	MA-Glättung	3
ABC-Transformation	0.5	MA-Glättung	6
ABC-Transformation	1.5	MA-Glättung	8
ABC-Transformation	2.0	LOWESS-Glättung	5
		LOWESS-Glättung	10
		LOWESS-Glättung	20
		STL-Dekomposition	7
		STL-Dekomposition	13
		STL-Dekomposition	20
		STL-Dekomposition	200

4 Auswertung

In der Auswertung wird überprüft, ob der Evaluator, im Vergleich zu einer normalen Vorhersage ohne Transformation oder Dekomposition, eine Verbesserung in der Vorhersage hervorruft. Hierzu wurde ein Experiment durchgeführt. In diesem wurde zunächst die Haupt-Zeitreihe ohne Transformations- bzw. Dekompositions-Verfahren oder zusätzliche Inputs vorhergesagt. Daraufhin wurden die resultierenden Ergebnisse als Basis zum Vergleich verwendet. Verglichen wurden diese mit den Vorhersagen, welche mit zusätzlichen Transformations- und Dekompositions-Verfahren sowie mit zusätzlichen Inputs erzeugt wurden. Abschließend werden die Ergebnisse diskutiert und bewertet.

4.1 Vorhersage

Bevor das Experiment durchgeführt werden konnte, musste die Menge an Vorhersage-Modellen sowie Transformations- und Dekompositions-Verfahren festgelegt werden. Die Vorhersage-Modelle sind auf 109 unterschiedliche Modelle festgelegt worden. Darunter befinden sich allein 100 Regressions-Modelle, da diese jeweils mit Fenstergrößen von 1 bis 50 aufgenommen wurden. Zusätzlich sind die vier Standard-Modelle, drei Varianten von ARIMA sowie zwei von Exponential Smoothing vertreten. Von den Random Forest Regressions Modellen verwenden alle eine Anzahl von 10 Bäumen. Dies ist damit begründet, dass eine Änderung des RMSE bei einer größeren Anzahl, von zum Beispiel 100 Bäumen, nicht existent ist. Bei einer Verwendung von 10.000 Bäumen sinkt der RMSE um ca. 3%, jedoch steigt die Laufzeit auch enorm an. Dadurch wäre das Modell nicht mehr verwendbar.
In Tabelle 5 sind alle Parameterkonfigurationen der Transformations- und Dekompositions-Verfahren aufgelistet, welche in dieser Auswertung genutzt wurden. Die unterschiedlichen Varianten der ABC-Transformation verwenden alle $\lambda_2 = 1$. Auf diese Weise werden etwaige Probleme mit Nullwerten verhindert. Der angegebene Wert in der Tabelle bezieht sich auf λ_1. Es darf nicht vergessen werden, dass im Schritt der Transformation und Dekomposition jeweils auch die Möglichkeit besteht, dass kein Verfahren angewandt wird. Dafür ist in der Tabelle jeweils *Original* notiert. Die Parameterkonfiguarationen wurden manuell ausgewählt. Das Kriterium der Auswahl war einerseits, die Anzahl der Konfigurationen so gering wie möglich zu halten. Folglich sind einige Konfigurationen direkt verworfen worden, da es ab einer gewissen Größe der Parameter nahezu keine Veränderungen in den Ergebnissen gibt. Andererseits müssen jedoch alle Konfigurationen, die unterschiedliche Ergebnisse liefern, abgedeckt sein.

4.1.1 Vorhersage der Haupt-Zeitreihe

Der erste Schritt des Experiments war es, die Haupt-Zeitreihe normal vorherzusagen. Normal bedeutet in diesem Fall, dass keine Transformations- bzw. Dekompositions-Verfahren verwendet wurden. Ebenfalls wurden keine zusätzlichen Inputs eingesetzt. Die Vorhersagen sind also nach dem Prinzip des Vorhersage-Evaluators erstellt worden, wobei die Schritte der Transformation und Dekomposition gezielt ausgelassen wurden. Daher wurde für jeden der 100 Test-Datenpunkte eine Vorhersage jedes Modells erstellt.
Zunächst wird die Zeitreihe von Campylobacter-Enteritis und anschließend die von Keuchhusten in FFM betrachtet. Die Ergebnisse sind in den meisten Fällen in Tabellenform und grafisch dargestellt. In den Tabellen sind von den Regressions-Modellen jeweils das beste Modell bezüglich MAE sowie RMSE angegeben. Es ist zu jedem dargestellten Modell der MAE, RMSE und MASE in der zugehörigen Spalte eingetragen. Zusätzlich sind die besten Modelle eines Fehlermaßes in der Tabelle fett markiert.

Campylobacter-Enteritis

Die Daten von Campylobacter werden schon seit Anfang 2001 erfasst. Daher werden in diesem Abschnitt zwei verschiedene Szenarien betrachtet. Im ersten Fall wurden die Daten zwischen September 2014 und November 2018 übergeben. Der Grund für die Wahl des genannten Zeitraumes war, dass für die Berechnung des ersten Test-Datenpunktes ein wenig mehr als zwei Saisons für das Training des Modells zur Verfügung stehen. In diesem speziellen Fall beinhaltet der kleinste Trainings-Datensatz 123 Datenpunkte.
Im zweiten Szenario wurden die kompletten Daten, welche bisher erfasst wurden, verwendet. Der kleinste Trainings-Datensatz enthält in diesem Fall 834 Datenpunkte. Die Vermutung war die, dass die Ergebnisse der Vorhersage im zweiten Szenario besser werden, da 6- bis 7-Mal so viele Trainings-Daten vorhanden sind.
Als Zusatz wurde überprüft, ob die Ergebnisse variieren, falls die Inzidenz statt der Fallzahl vorhergesagt wird. Für diesen Fall wurde der kleine Zeitraum aus dem ersten Szenario verwendet.
Aufgrund der langen Laufzeit der zweiten Variante von ARIMA, welche auf das saisonale Modell beschränkt ist, wurden für diese Variante lediglich ein Mal die Modell-Parameter bestimmt. Diese wurden für den kleinsten Trainings-Datensatz ermittelt und daraufhin für jedes der 100 Trainings eingesetzt. Die Genauigkeit dieser Variante wird daher nicht bestmöglich sein, jedoch war es Laufzeit-technisch nicht anders umsetzbar.

Szenario 1: Verwendung der Daten von 2014-2018

Die Ergebnisse des ersten Szenarios sind in Abbildung 11 und in Tabelle 6 dargestellt. Unter Betrachtung des MAE bzw. MASE ist die R-Paket Variante des Exponential Smoothing das optimale Modell. Wird jedoch der RMSE als Maß verwendet, liefert die Random Forest Regression die besten Ergebnisse. Da es zwei bestmögliche Vorhersage-Modelle gibt, sind in dem Koordinatensystem zwei Vorhersagen eingezeichnet.
Der absolute Unterschied des MAE der beiden Vorhersagen liegt unter einem Zehntel. Im Vergleich der grafisch dargestellten Zeitreihen ist jedoch ein großer Unterschied zu erkennen. Die Vorhersage des ES-Modells, welche nach dem MASE das beste Modell ist, ähnelt einer Glättung bzw. dem Durchschnitt der Haupt-Zeitreihe. Das Regressions-Modell hingegen versucht zusätzlich das Rauschen der Original-Zeitreihe abzubilden.
Es ist vom Anwendungsfall abhängig, welche der Vorhersagen genutzt werden sollte. Falls eine grobe Abschätzung für die nächste Woche, ohne große Beeinflussung durch Ausreißer, gewünscht ist, fällt die richtige Wahl auf das ES-Modell. Das Regressions-Modell sollte verwendet werden, wenn eine etwas realitätsgetreuere Vorhersage oder eine Suche nach möglichen zukünftigen Anstiegen gefordert ist. Zum Bespiel hätte das Regressions-Modell im Mai 2017 frühzeitig einen kleineren Anstieg erkannt. Die größeren positiven Ausreißer konnten jedoch von keinem der beiden Modelle erfasst werden.

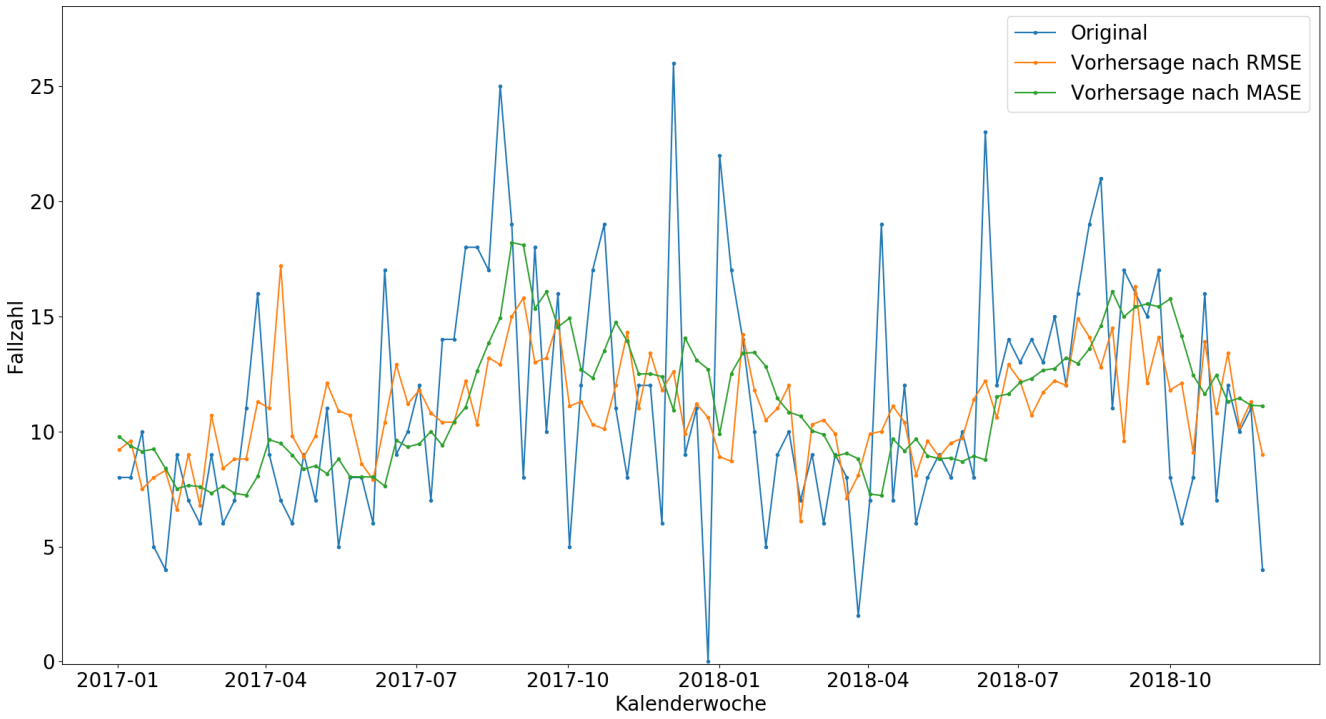

Abbildung 11: Vorhersage des besten Modells im Vorhersage-Evaluator ohne Transformation und Dekomposition. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Campylobacter-Enteritis 2014-2018, Modell-RMSE: Random Forest Regression mit einer Fenstergröße von 31, Modell-MASE: Holt-Winters in R-Paket Variante

Tabelle 6: Ergebnisse des Vorhersage-Evaluators ohne Transformation und Dekomposition. Daten: Fallzahlen von Campylobacter-Enteritis 2014-2018

Fallzahl: Campylobacter-Enteritis				
Vorhersage-Modell	Variante	MAE	RMSE	MASE
Average	-	4.0973	5.0787	0.8705
Naive	-	4.6700	6.3048	0.9921
Seasonal-Naive	-	4.5600	5.9548	0.9688
Drift	-	4.6896	6.3237	0.9963
Exponential Smoothing	Holt-Winters	3.6915	4.8553	0.7842
Exponential Smoothing	**R-Paket**	**3.5387**	4.8849	**0.7518**
ARIMA	ARIMA	3.7453	4.9218	0.7957
ARIMA	sARIMA	4.0531	5.1660	0.8611
ARIMA	R-Paket	3.7381	4.8819	0.7941
Lineare Regression	Window=5	3.6551	4.8088	0.7765
Lineare Regression	Window=6	3.6542	4.8353	0.7763
Random Forest Regression	**Window=31**	3.5750	**4.6628**	0.7595
Random Forest Regression	Window=45	3.5490	4.7026	0.7540

Szenario 2: Verwendung der Daten von 2001-2018

Das zweite Szenario unterscheidet sich lediglich in der Anzahl der Trainings-Datenpunkte. Es wurden die gleichen Test-Datenpunkte wie im vorherigen Szenario vorhergesagt. Die Ergebnisse zu diesem sind in Abbildung 12 und in Tabelle 7 dargestellt. Hier ist es ebenfalls ein ES-Modell, welches die besten Ergebnisse produziert. In diesem Fall ist es jedoch die Holt-Winters Variante, welche sogar bezüglich MAE, RMSE und MASE im vorderen Bereich liegt. Im Vergleich zum ersten Szenario ist klar erkennbar, dass bei fast jedem Modell der MAE um ca. zwei Zehntel besser ist. Der RMSE wiederum zeigt relativ gesehen eine kleinere Verbesserung.

Wie bereits in Abschnitt 3.2.2 kurz erwähnt wurde, ist die R-Paket Variante des ES-Modells nicht zwingend besser als die Holt-Winters Variante. In diesem Szenario ist dies klar erkennbar. Im Vergleich des RMSE der beiden Varianten ist die Holt-Winters Variante eindeutig im Vorteil. Daher ist anzunehmen, dass die Holt-Winters Variante die zusätzlichen Trainings-Daten besser verarbeitet. Die R-Paket Variante des Exponential Smoothing zeigt im zweiten Szenario nahezu keine Verbesserung. Um dies zu begründen, muss diese Variante genauer betrachtet werden. Eine Vermutung ist, dass diese Variante die zusätzlichen Daten nicht vollständig mit in die Berechnung einfließen lässt, um die Laufzeit möglichst gering zu halten.

Unter Betrachtung der grafischen Darstellung der Vorhersage ist in diesem Szenario zu erkennen, dass dieses Modell die Saisonlität der Daten gut wiedergibt. Anders, als von dem ES-Modell im ersten Beispiel, wird in diesem Fall das Rauschen deutlich besser angenähert. Werden die Ergebnisse jedoch mit der Vorhersage der Random Forest Regression im erstem Szenario verglichen, sind das Rauschen bzw. die Ausreißer schlechter angenähert.

Im Großen und Ganzen ist jedoch eine eindeutige Verbesserung in nahezu jedem Modell zu erkennen. Der Fehler sinkt im Schnitt um ungefähr 5 Prozent. Es ist nur auffällig, dass der RMSE der Random Forest Regression schlechter als im ersten Szenario ist. Dies hängt jedoch damit zusammen, dass dieses Verfahren einen Zufalls-Aspekt beinhaltet, wodurch der Fehler nach jeder Ausführung ein anderer ist. Durch diese Variation scheint der Fehler in diesem Szenario schlechter zu sein, obwohl es sich lediglich um einen Zufall handelt. Nach einer weiteren Ausführung waren die Werte ebenfalls besser als im ersten Szenario. Diese Ergebnisse sollen zeigen, dass es sich in der Random Forest Regression um keinen konstanten Fehler handelt. Abschließend ist zu sagen, dass, falls die Laufzeit zweitrangig ist, immer alle Daten verwendet werden sollten, um das bestmögliche Ergebnis erzielen zu können.

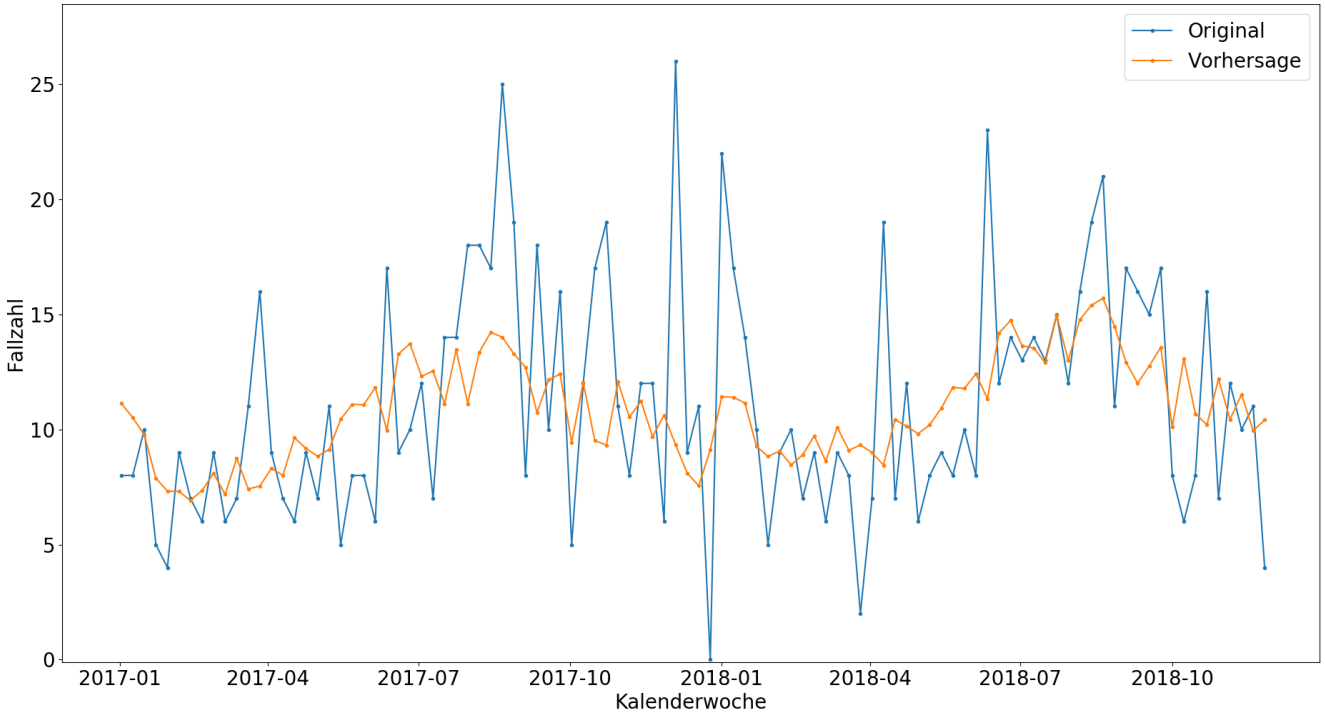

Abbildung 12: Vorhersage des besten Modells im Vorhersage-Evaluator ohne Transformation und Dekomposition. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Campylobacter-Enteritis 2001-2018, Modell: Exponential Smoothing in Holt-Winters Variante

Tabelle 7: Ergebnisse des Vorhersage-Evaluators ohne Transformation und Dekomposition. Daten: Fallzahlen von Campylobacter-Enteritis 2001-2018

Fallzahl: Campylobacter-Enteritis				
Vorhersage-Modell	Variante	MAE	RMSE	MASE
Average	-	4.0265	5.3737	0.8554
Naive	-	4.6700	6.3048	0.9921
Seasonal-Naive	-	4.5600	5.9548	0.9688
Drift	-	4.6728	6.3083	0.9927
Exponential Smoothing	**Holt-Winters**	**3.4716**	**4.5810**	**0.7375**
Exponential Smoothing	R-Paket	3.4989	4.8735	0.7433
ARIMA	ARIMA	3.5418	4.8744	0.7524
ARIMA	sARIMA	3.7182	4.8249	0.7899
ARIMA	R-Paket	3.7743	4.9036	0.8018
Lineare Regression	Window=35	3.4906	4.8802	0.7416
Lineare Regression	Window=42	3.5381	4.8041	0.7517
Random Forest Regression	Window=15	3.4800	4.7400	0.7393
Random Forest Regression	Window=24	3.5640	4.7072	0.7572

Tabelle 8: Ergebnisse des Vorhersage-Evaluators ohne Transformation und Dekomposition. Daten: Inzidenz von Campylobacter-Enteritis 2014-2018

Inzidenz: Campylobacter-Enteritis				
Vorhersage-Modell	Variante	MAE	RMSE	MASE
Average	-	0.5988	0.7422	0.8705
Naive	-	0.6825	0.9214	0.9921
Seasonal-Naive	-	0.6664	0.8703	0.9688
Drift	-	0.6854	0.9242	0.9963
Exponential Smoothing	Holt-Winters	0.5390	0.7088	0.7836
Exponential Smoothing	**R-Paket**	**0.5215**	0.7166	**0.7581**
ARIMA	ARIMA	0.5504	0.7263	0.8000
ARIMA	sARIMA	0.5719	0.7598	0.8314
ARIMA	R-Paket	0.5460	0.7141	0.7938
Lineare Regression	Window=5	0.5342	0.7028	0.7765
Lineare Regression	Window=6	0.5341	0.7067	0.7763
Random Forest Regression	**Window=33**	0.5406	**0.6800**	0.7858
Random Forest Regression	Window=48	0.5361	0.7020	0.7793

Vorhersage der Inzidenz der Daten von 2014-2018

Da die Skalierung der Inzidenz eine andere ist, wie die der Fallzahlen, wird zum Vergleich der Ergebnisse der MASE verwendet. Die Vermutung ist, dass die Ergebnisse des ersten Szenarios gleich mit denen in Tabelle 8 sind. Dies ist damit zu begründen, dass die Zeitreihe in jedem Zeitpunkt gleich skaliert wurde, wodurch die Struktur unberührt bleibt. Im Vergleich der Ergebnisse spiegelt sich diese Vermutung wider. Aufgrund dieser Tatsache wird in den folgenden Abschnitten die Fallzahl analysiert. Die ARIMA Variante, welche auf das sARIMA Modell beschränkt ist, zeigt alleinig einen größeren Unterschied. Diese Variante wird jedoch ohnehin nicht für die folgenden Auswertungen verwendet, da die Laufzeit zu hoch ist.

Keuchhusten

Wie bereits in Kapitel 2 angesprochen, werden die Daten zu Keuchhusten erst seit April 2013 aufgezeichnet. Aufgrund dieser Tatsache werden die Daten nicht in zwei Szenarien aufgeteilt. In Abbildung 13 und in Tabelle 9 sind die Ergebnisse, welche unter Verwendung aller zur Verfügung stehenden Daten erstellt wurden, dargestellt. Sehr auffällig ist, dass nahezu alle Modelle schlechtere Ergebnisse als das Average Modell aufweisen. Der Grund hierfür liegt darin, dass die Daten kaum vorhersagbare Muster beinhalten und eigentlich nur ein Grundrauschen um den Wert drei darstellen. Da es sich bei dem Rauschen scheinbar um einen normalverteilten Wert handelt, erweist sich hier das Average Modell als sehr geeignet. Am besten schneidet jedoch die Lineare Regression ab, wessen Vorhersage auch in dem Koordinatensystem dargestellt ist.

Im Vergleich mit den Ergebnissen von Campylobacter muss erneut der MASE verwendet werden, da es sich um unterschiedliche Zeitreihen handelt. Es ist eindeutig zu erkennen, dass die Ergebnisse der Vorhersage von Campylobacter um einiges besser sind. Die Vermutung ist, dass die Zeitreihe von Campylobacter mehr Muster aufweist, welche vorhergesagt werden können.

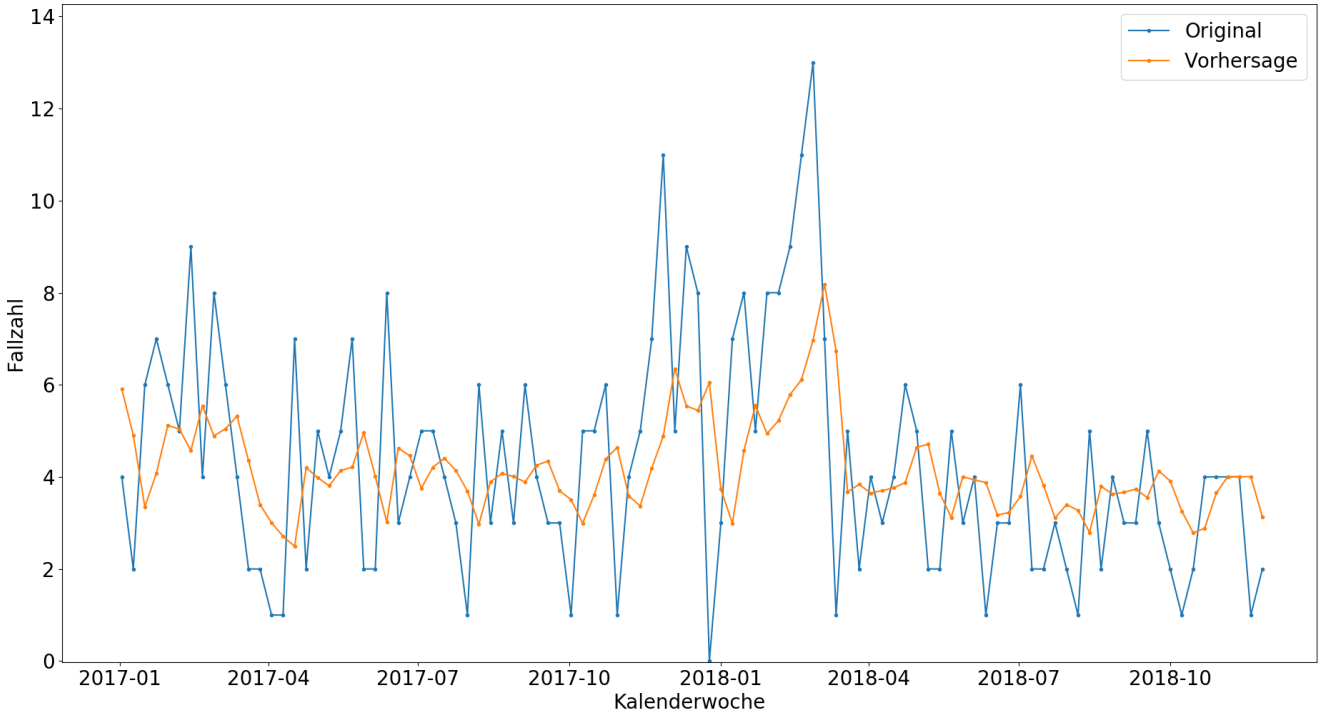

Abbildung 13: Vorhersage des besten Modells im Vorhersage-Evaluator ohne Transformation und Dekomposition. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Keuchhusten 2013-2018, Modell: Lineare Regression mit einer Fenstergröße von 2

Tabelle 9: Ergebnisse des Vorhersage-Evaluators ohne Transformation und Dekomposition. Daten: Fallzahlen von Keuchhusten 2013-2018

Fallzahl: Keuchhusten				
Vorhersage-Modell	Variante	MAE	RMSE	MASE
Average	-	1.9611	2.5429	0.8825
Naive	-	2.2500	2.8583	1.0125
Seasonal-Naive	-	2.7100	3.5426	1.2195
Drift	-	2.2556	2.8650	1.0150
Exponential Smoothing	Holt-Winters	2.2614	2.7474	1.0176
Exponential Smoothing	R-Paket	2.0503	2.5599	0.9226
ARIMA	ARIMA	2.0132	2.4801	0.9059
ARIMA	sARIMA	1.9800	2.4185	0.8910
ARIMA	R-Paket	2.0212	2.4850	0.9095
Lineare Regression	**Window=2**	**1.8835**	**2.3562**	**0.8476**
Random Forest Regression	Window=1	2.0047	2.5193	0.9021
Random Forest Regression	Window=4	1.9777	2.5469	0.8900

4.1.2 Vorhersage-Evaluator

Im nächsten Schritt des Experiments kam die eigentliche Funktion des Vorhersage-Evaluators zum Einsatz. Hier wurden nun die Vorgänge der Transformation und Dekomposition, mit den in Tabelle 5 aufgelisteten Verfahren, durchgeführt. Aus diesen entstehen insgesamt 55 Kombinationen. In diesen Kombinationen müssen jeweils bis zu drei Mal Vorhersagen für 100 Test-Datenpunkte erstellt werden, wobei für jeden Datenpunkt jeweils ein neues Modell erstellt werden muss. Zusätzlich wird für jedes der 109 aufgeführten Modelle eine Vorhersage erzeugt. Daraus folgend wurden in dieser Konfiguration des Evaluators ca. 1,4 Millionen Modelle erstellt, welche jeweils einen Punkt vorhergesagt haben. Dies beanspruchte ungefähr 80 Prozent der Laufzeit. Die restliche Laufzeit wurde dafür verwendet, die zerlegten Zeitreihen bestmöglich zusammenzusetzen. Der Grund dafür besteht darin, dass, wie in Abschnitt 3.3.2 festgelegt wurde, alle möglichen Kombinationen in der Zusammensetzung getestet werden müssen. Daraus resultiert bei einer Zusammensetzung von drei Komponenten die Suche der besten Kombination aus ca. 1,3 Millionen Ergebnissen.

Campylobacter-Enteritis

Die Effektivität des Vorhersage-Evaluators wurde bei Campylobacter mit den Daten des ersten Szenarios, demnach von 2014-2018, getestet. Um die Daten des kompletten Zeitraums nutzen zu können, ist vorrangig eine Laufzeitoptimierung des Evaluators zwingend erforderlich. Diese wird in Kapitel 5 weitergehend angesprochen. Andernfalls würde die Durchführung für den Betrieb zu viel Zeit beanspruchen.
In Tabelle 10 sind die Ergebnisse zu jeder Kombination von Transformations- und Dekompositions-Verfahren sowie die verwendeten Vorhersage-Modelle der zugehörigen Dekompositions-Komponenten dargestellt. Bei der Wahl der Modelle wurde in diesem Fall die Zusammenführung der Komponenten mit dem niedrigsten RMSE gesucht. Die Tabelle ist in fünf Abschnitte aufgeteilt, welche jeweils die 11 unterschiedlichen Dekompositionen zu einer bestimmten Transformation darstellen. In jedem Abschnitt ist das beste Modell nach dem RMSE sowie nach dem MAE fett hervorgehoben. Hiermit ist der Vergleich der unterschiedlichen Transformationen einfacher zu überblicken. Es ist zu beachten, dass die Kombination, welche in diesen Ergebnissen den besten MAE aufweist, nicht zwingend die beste bezüglich des MAE ist. Der Grund dafür ist, dass die Zusammenführung unter Betrachtung des RMSE durchgeführt wurde.
Die Tabelle der Ergebnisse von Campylobacter ist absichtlich nicht verkürzt dargestellt, um die folgende detailreichere Analyse besser zu unterstützen. Die beste Kombination, bezüglich RMSE sowie MAE, befindet sich direkt im ersten Abschnitt, in dem keine Transformation angewandt wurde. Als Dekomposition wurde das STL-Verfahren mit einer Fenstergröße von 13 genutzt. Zur Vorhersage des Trends wurde das Average Modell, der Saisonalität das ES-Modell mit der R-Paket Variante und des Rests die Lineare Regression mit einer Fenstergröße von 3 verwendet. Die Zeitreihe der Vorhersage ist zusammen mit der Haupt-Zeitreihe in Abbildung 14 dargestellt. Wenn diese mit den Ergebnissen aus dem vorherigen Abschnitt verglichen werden, ist klar zu erkennen, dass sich diese näher an der Haupt-Zeitreihe anliegt. Es ist aber ebenfalls eindeutig, dass die großen Ausreißer in keinster Art und Weise erkannt werden konnten. Dies könnte damit zu begründen sein, dass diese keinen bestimmten Mustern folgen und die Zeitreihe quasi willkürlich für einen einzigen Zeitpunkt in positiver Richtung ausschlägt. Im Vergleich des Fehlers des besten Modells aus diesem sowie dem letzten Abschnitt zeigt sich, dass der Vorhersage-Evaluator den RMSE um ca. 6% und den MAE um ca. 15% senken konnte. Die überzeugen-

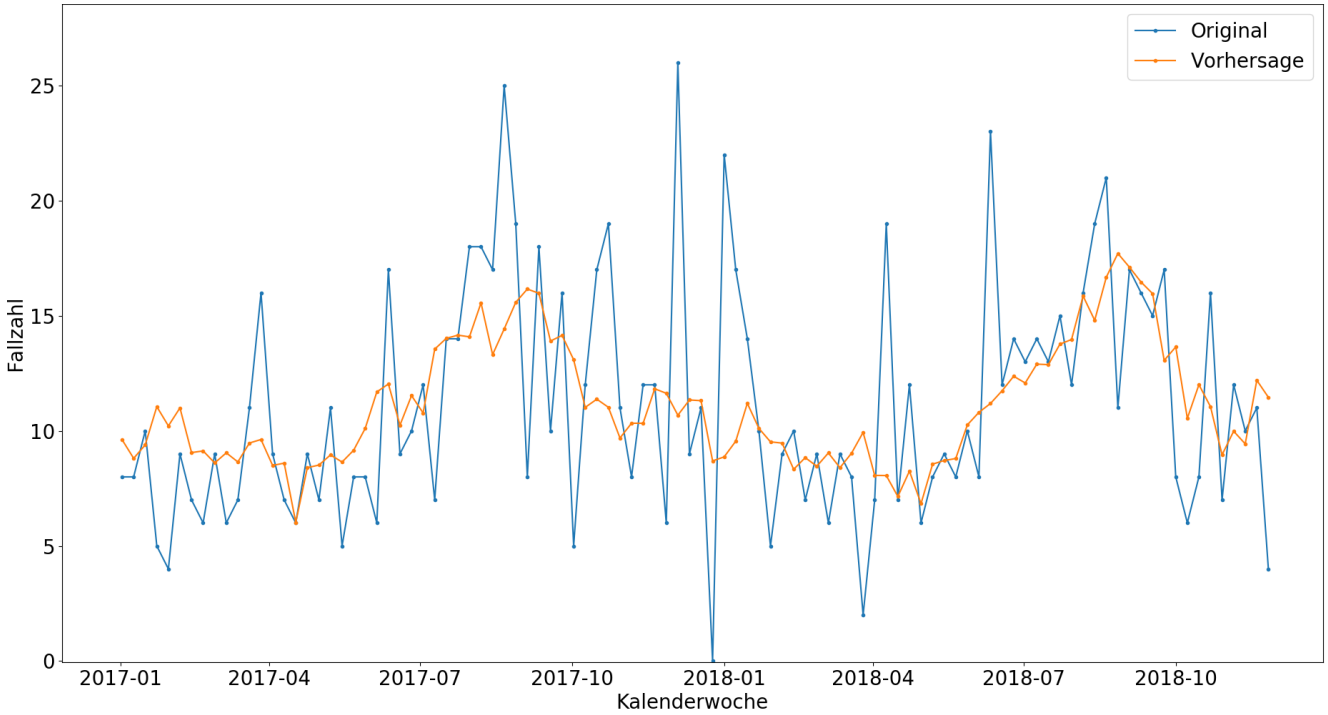

Abbildung 14: Vorhersage des besten Modells im Vorhersage-Evaluator. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Campylobacter-Enteritis 2014-2018. Dekomposition: STL-13 (Trend-Modell: Average, Saison-Modell: Exponential Smoothin R-Paket Variante, Rest-Modell: Lineare Regression Fenstergröße=3)

de Verbesserung des MAE spiegelt sich an der extremeren Annäherung an die Haupt-Zeitreihe wider. Die etwas kleinere Verbesserung des RMSE lässt sich damit begründen, dass die großen Ausschläge und resultierenden Fehler im RMSE stärker einfließen, da der Fehler quadratisch in die Berechnung eingeht.

Anschließend werden die Ergebnisse der Kombinationen von Transformationen und Dekompositionen analysiert. Die am schlechtesten abschneidende Dekomposition ist das LOWESS-Verfahren mit einer Fenstergröße von 5. Dieses schneidet in 4 von 5 Fällen schlechter ab als eine normale Vorhersage mit dem Average-Modell. In den Ergebnissen, welche eine ABC-Transformation mit $\lambda_1 = 2$ beeinhalten, gibt es nur eine Kombination, die um einen Bruchteil besser ist als die Ergebnisse aus dem letzten Abschnitt. Daraus schließt sich, dass eine solche Transformation sich definitiv negativ auf die Ergebnisse dieser Zeitreihe auswirkt. Die größten Verbesserungen gab es, wenn keine Transformation, oder eine ABC-Transformation mit $\lambda_1 = 0,5$ durchgeführt wurde.

Unter genauerer Betrachtung der Dekompositions-Verfahren lässt sich ein eindeutiger Trend ableiten. In allen Abschnitten, welche eine klare Verbesserung aufweisen, ist in der besten Kombination die STL-Dekomposition verwendet worden. Bei genauerer Untersuchung der Vorhersage-Modelle ist auffällig, dass der Trend des STL-Verfahrens immer mit dem Average Modell vorhergesagt wird. In den restlichen Komponenten herrscht eine große Variation der Vorhersage-Modelle.

Abschließend lässt sich sagen, dass es eindeutig bessere und schlechtere Verfahren bzw. Parameter-Konfigurationen in der Transformation und Dekomposition gibt. Jedoch sollte dies mit Vorsicht zu betrachten sein, da diese Konfigurationen womöglich einfach nur auf die Daten von Campylobacter perfekt zugeschnitten sind. Für andere Daten sind möglicherweise andere Konfigurationen besser geeig-

net. Daher werden im nächsten Abschnitt die Ergebnisse des Vorhersage-Evaluators unter Verwendung der Daten von Keuchhusten analysiert.

Tabelle 10: Ergebnisse des Vorhersage-Evaluators. Daten: Fallzahlen von Campylobacter-Enteritis 2014-2018

Transformation	Dekomposition	Trend	Saison	Rest	MAE	RMSE	MASE
Original	Original	-	-	RF-40	3.5090	4.5815	0.7455
Original	MA-3	RF-11	-	Average	3.4665	4.6041	0.7365
Original	MA-6	RF-42	-	RF-43	3.6303	4.6904	0.7712
Original	MA-8	RF-41	-	LR-2	3.3845	4.7812	0.7190
Original	LOWESS-5	Average	-	LR-1	4.0758	5.0560	0.8659
Original	LOWESS-10	RF-49	-	RF-27	3.6594	4.8315	0.7774
Original	LOWESS-20	S-Naive	-	ES-HW	3.5894	4.7583	0.7626
Original	STL-7	Average	ES-R	RF-12	3.2784	4.4292	0.6965
Original	**STL-13**	**Average**	**ES-R**	**LR-3**	**3.0929**	**4.4046**	**0.6571**
Original	STL-20	Average	LR-30	LR-3	3.2989	4.4249	0.7008
Original	STL-200	Average	nARIMA	LR-3	3.3467	4.4493	0.7110
ABC-0.0	Original	-	-	ES-HW	3.5844	4.8590	0.7615
ABC-0.0	MA-3	RF-18	-	Average	3.4042	4.6624	0.7232
ABC-0.0	MA-6	RF-49	-	ARIMA-R	3.6457	4.8057	0.7745
ABC-0.0	MA-8	S-Naive	-	ES-R	3.6881	4.8786	0.7835
ABC-0.0	LOWESS-5	Average	-	LR-1	4.0004	5.1788	0.8499
ABC-0.0	LOWESS-10	S-Naive	-	LR-3	3.5610	4.7586	0.7565
ABC-0.0	LOWESS-20	S-Naive	-	ARIMA-R	3.3982	4.5542	0.7219
ABC-0.0	STL-7	Average	RF-48	RF-7	3.6082	4.7101	0.7666
ABC-0.0	STL-13	Average	RF-43	RF-6	3.5740	4.6503	0.7593
ABC-0	**STL-20**	**Average**	**RF-42**	**ARIMA-R**	**3.3618**	**4.4148**	**0.7142**
ABC-0	STL-200	Average	RF-8	Average	3.4789	4.5551	0.7391
ABC-0.5	Original	-	-	RF-37	3.5201	4.6577	0.7478
ABC-0.5	MA-3	RF-41	-	Average	3.4385	4.5879	0.7305
ABC-0.5	MA-6	RF-41	-	RF-46	3.4463	4.5462	0.7321
ABC-0.5	MA-8	RF-50	-	RF-19	3.4178	4.5674	0.7261
ABC-0.5	LOWESS-5	Average	-	LR-1	4.0149	5.0925	0.8529
ABC-0.5	LOWESS-10	S-Naive	-	LR-3	3.6029	4.8262	0.7654
ABC-0.5	LOWESS-20	S-Naive	-	ARIMA-R	3.5725	4.7286	0.7590
ABC-0.5	**STL-7**	**Average**	**ES-R**	**LR-3**	**3.1360**	4.4568	**0.6662**
ABC-0.5	**STL-13**	**Average**	**RF-48**	**ES-HW**	3.3769	**4.4075**	0.7174
ABC-0.5	STL-20	Average	RF-50	ES-HW	3.4101	4.4351	0.7245
ABC-0.5	STL-200	Average	LR-28	LR-4	3.2960	4.4271	0.7002
ABC-1.5	Original	-	-	RF-32	3.6432	4.7180	0.7740

LR=Lineare Regression, RF=Random Forest Regression Fortsetzung nächste Seite

Tabelle 10: Ergebnisse des Vorhersage-Evaluators. Daten: Fallzahlen von Campylobacter-Enteritis 2014-2018

Transformation	Dekomposition	Trend	Saison	Rest	MAE	RMSE	MASE
ABC-1.5	MA-3	RF-44	-	ES-R	3.6272	4.7725	0.7706
ABC-1.5	MA-6	RF-42	-	LR-3	3.5226	4.6723	0.7484
ABC-1.5	MA-8	RF-40	-	LR-2	3.4481	4.7659	0.7325
ABC-1.5	LOWESS-5	Average	-	LR-1	4.1659	5.0674	0.8850
ABC-1.5	LOWESS-10	S-Naive	-	RF-49	3.7375	4.7761	0.7940
ABC-1.5	LOWESS-20	S-Naive	-	RF-12	3.4349	4.5850	0.7297
ABC-1.5	STL-7	Average	RF-48	ES-HW	3.6525	4.5499	0.7760
ABC-1.5	**STL-13**	**Average**	**LR-11**	**ES-HW**	**3.3755**	4.5437	**0.7171**
ABC-1.5	STL-20	Average	LR-24	LR-3	3.6199	4.5816	0.7690
ABC-1.5	**STL-200**	**Average**	**nARIMA**	**LR-3**	3.4774	**4.5261**	0.7388
ABC-2.0	Original	-	-	ES-HW	3.9442	5.0082	0.8379
ABC-2.0	MA-3	RF-12	-	Average	3.7302	4.7335	0.7925
ABC-2.0	MA-6	RF-30	-	RF-42	3.6566	4.7099	0.7768
ABC-2.0	MA-8	RF-47	-	RF-10	3.7271	4.7791	0.7918
ABC-2.0	LOWESS-5	RF-46	-	RF-50	3.6996	4.8067	0.7860
ABC-2.0	LOWESS-10	RF-45	-	RF-22	3.7179	4.9491	0.7899
ABC-2.0	**LOWESS-20**	**S-Naive**	-	**RF-18**	**3.4147**	4.5924	**0.7254**
ABC-2.0	STL-7	Average	LR-8	ES-HW	3.7394	4.7652	0.7944
ABC-2.0	**STL-13**	**Average**	**LR-8**	**ES-HW**	3.6075	**4.5490**	0.7664
ABC-2.0	STL-20	Average	LR-7	LR-3	3.5881	4.6325	0.7623
ABC-2.0	STL-200	Average	LR-7	ES-HW	3.6594	4.7386	0.7774

LR=Lineare Regression, RF=Random Forest Regression

Tabelle 11: Ergebnisse des Vorhersage-Evaluators. Daten: Fallzahlen von Keuchhusten 2013-2018

Fallzahl: Keuchhusten

Transformation	Dekomposition	Trend	Saison	Rest	MAE	RMSE	MASE
Original	Original	-	-	LR-2	1.8835	2.3562	0.8476
Original	STL-20	S-Naive	LR-1	RF-19	1.8282	2.3201	0.8227
ABC-0.0	STL-20	S-Naive	LR-1	LR-10	1.8439	2.3998	0.8298
ABC-0.5	STL-200	S-Naive	ARIMA-R	ARIMA-R	1.8880	2.3375	0.8496
ABC-1.5	**STL-200**	**S-Naive**	**Naive**	**LR-9**	**1.8181**	**2.2849**	**0.8181**
ABC-2.0	STL-200	S-Naive	Drift	LR-9	1.8539	2.2888	0.8343

Keuchhusten

In der Anwendung des Vorhersage-Evaluators wurden erneut die kompletten Daten von Keuchhusten, also von 2013-2018, verwendet. Die Analyse dieser Krankheit wird nicht so detailliert aufgeführt wie die von Campylobacter. Im Folgenden wird überprüft, ob der Vorhersage-Evaluator bei Keuchhusten die gleichen Ausmaße an Verbesserungen aufzeigt wie bei Campylobacter. Die Vermutung war, dass die Verbesserung nicht so gut wie bei Campylobacter abschneiden wird, da bereits eine einfache Vorhersage, ohne Transformation und Dekomposition, nicht sonderlich gut ausgefallen ist.

Die Ergebnisse der Anwendung des Evaluators sind in Abbildung 15 und in Tabelle 11 veranschaulicht. Diesmal ist die Tabelle kompakter dargestellt, indem zu jeder Transformation lediglich die beste Dekomposition angegeben ist. Zusätzlich ist das beste Ergebnis der normalen Vorhersage, ohne Dekomposition und Transformation, aus dem letzten Abschnitt vorangestellt. Klar zu erkennen ist, dass die Verwendung von Transformations- und Dekompositions-Verfahren eine Verbesserung im Fehler aufweist. Diese liegt im RMSE und im MAE bei ungefähr 5%. Die Verbesserung des RMSE, auf dem in der Ausführung der Fokus lag, ist identisch zu der bei Campylobacter. Der MAE hingegen weist im Vergleich eine um einiges geringere Verbesserung auf. Dies bestätigt die zuvor angesprochene Vermutung, dass der Evaluator bei Keuchhusten weniger Auswirkungen auf das Ergebnis haben wird. Aufgrund mangelnder Muster in der Haupt-Zeitreihe scheint die Vorhersage nicht sehr viel Optimierungs-Potential zu besitzen. Es existieren zwei Möglichkeiten, welche dies begründen könnten. Entweder, die in dieser Arbeit verwendeten Herangehensweisen sind für diese Art von Zeitreihe nicht optimal, oder die Daten folgen keinen vorhersehbaren Mustern.

Wenn die grafisch dargestellten Ergebnisse mit denen aus Abbildung 13 auf Seite 49 verglichen werden, scheinen sich die hier gezeigten Ergebnisse besser an das Rauschen anzunähern.

Ungeachtet dessen, hat der Evaluator das Ergebnis der Vorhersage verbessern können und das, obwohl die Zeitreihe von Keuchhusten sehr zufällig zu sein scheint. Eindeutig zu sehen ist jedoch, dass in der Zeitreihe ein Trend und eine Saisonalität verschlüsselt sind, welche sich mit dem STL-Verfahren abtrennen sowie separat vorhersagen lassen.

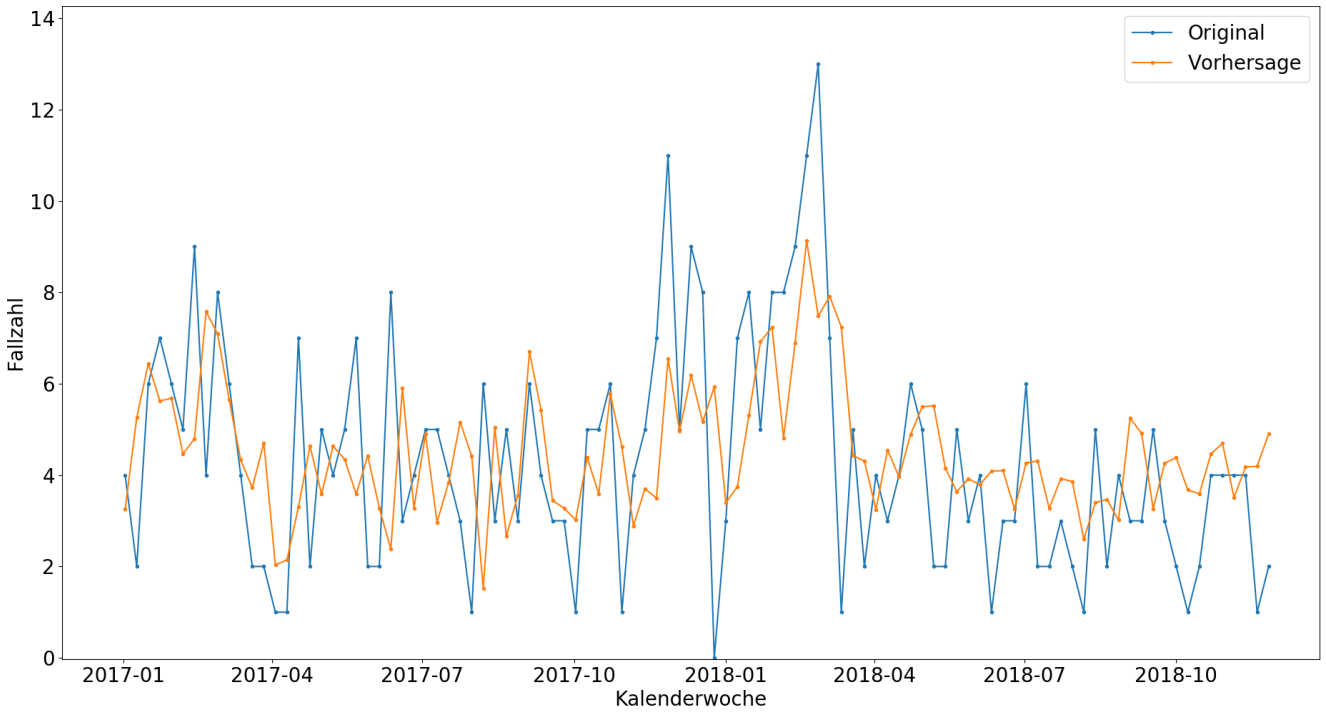

Abbildung 15: Vorhersage des besten Modells im Vorhersage-Evaluator. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Keuchhusten 2013-2018, Dekomposition: STL-200 (Trend-Modell: Seasonal-Naive, Saison-Modell: Naive, Rest-Modell: Lineare Regression Fenstergröße=9)

4.1.3 Erweiterung um zusätzliche Inputs

In dem letzten Versuch des Experiments wurden die Regressions-Modelle, Lineare Regression (LR) und Random Forest Regression (RF), um die zusätzlichen Inputs erweitert. Die übrigen Vorhersage-Modelle bieten nicht die Struktur, diese mit zusätzlichen Informationen aufzustocken. Zunächst wird überprüft, ob die zusätzlichen Informationen in einer normalen Vorhersage, ohne Dekomposition sowie Transformation, Verbesserungen aufweisen. Im nächsten Schritt werden zusätzlich die Dekomposition sowie Transformation durchgeführt. Aufgrund der Tatsache, dass es, wie im Unterkapitel 3.4 beschrieben, einige unterschiedliche Varianten von zusätzlichen Informationen gibt, werden in diesem Schritt lediglich ausgewählte Kombinationen von Dekompositionen und Transformationen angewandt. Ausgewählt wurden jene, die im letzten Abschnitt die besten Ergebnisse geliefert haben. Es wird im gesamten Abschnitt das Fehlermaß RMSE zur Bestimmung der besten Kombinationen verwendet.

In dem Experiment wurden vier Varianten an zusätzlichen Informationen geprüft. Jede dieser wurde mit zwei unterschiedlichen Nachbar-Zeitreihen erstellt. Diese unterscheiden sich in der Reisezeit von 30 bzw. 60 Minuten. Die erste Variante beinhaltet lediglich die Verwendung der einfachen Nachbar-Zeitreihe. In der Vorhersage wird der Wert des letzten Zeitpunktes dieser mitberücksichtigt. Die zweite Variante verwendet den letzten Wert der relativen Steigung der Nachbar-Zeitreihe als zusätzliche Information. In der dritten Variante wird die normale Abweichung zwischen der Haupt- und Nachbar-Zeitreihe verwendet. Zur Berechnung dieser wurde aufgrund der unterschiedlichen Skalierung die Inzidenz betrachtet. Die letzte Variante beinhaltet keine Bestimmung neuer Werte. Diese verwendet die Kombination der Werte

aus den anderen drei Varianten. Es werden also in dieser Variante zur Vorhersage drei zusätzliche Werte an die Regressions-Modelle übergeben.

Campylobacter-Enteritis

Als Erstes wird erneut Campylobacter-Enteritis untersucht. Die Ergebnisse werden anschließend mit denen aus den letzten beiden Abschnitten verglichen. Bei dem verwendeten Zeitraum handelt es sich ein weiteres Mal um 2014-2018, wobei im ersten Teil kurz auf die Nutzung des kompletten Zeitraums eingegangen wird.

Vorhersage ohne Transformations- und Dekompositions-Verfahren

Wie bereits erwähnt, wird zunächst keine Transformation und Dekomposition angewandt. Die Ergebnisse hierzu sind in Tabelle 12 dargestellt. Die ersten beiden Zeilen zeigen die besten Vorhersagen, bezüglich MAE bzw. RMSE, aus dem ersten Teil des Experiments. In diesem wurde der gleiche Zeitraum betrachtet und ebenfalls keine Transformation und Dekomposition durchgeführt. Eindeutig zu erkennen ist, dass die Random Forest Regression, unter Betrachtung des RMSE, in jeder Kombination die besten Ergebnisse erzielte. Die Fehler der Varianten 1 sowie 4 heben sich im MAE klar von den anderen beiden ab. In Abbildung 16 sind die Ergebnisse der ersten Variante mit einer Reisezeit von 30 Minuten dargestellt. Diese war die beste Vorhersage und sie zeigt Verbesserungen von ungefähr 10% im MAE sowie 6% im RMSE auf. Der RMSE ist hier schon besser als im letzten Abschnitt, in welchem die Transformationen und Dekompositionen im Vorhersage-Evaluator verwendet wurden. In den letzten beiden Zeilen der Tabelle sind zusätzlich die besten Vorhersagen der jeweiligen Reisezeit mittels Linearer Regression abgebildet. Diese sind zusätzlich aufgelistet, da das Lineare Regressions Modell in erneuten Durchläufen immer den gleichen Fehler aufweist. Dieser Fall soll zeigen, dass die zusätzlichen Inputs in einer eindeutigen Verbesserung der Vorhersage resultieren und dies nicht auf mögliche Varianzen im Fehler der Random Forest Regression zurückzuführen ist.
Aufgrund der Tatsache, dass hier noch keine Transformationen und Dekompositionen genutzt werden, wurde hier ebenfalls der komplette Zeitraum von 2001-2018 getestet. Die Ergebnisse dieses Szenarios zeigten jedoch, im Vergleich zu der Variante ohne zusätzlicher Informationen, keinerlei Verbesserung. Die Vermutung war, dass die Varianz des Fehlers vom Random Forest Modell dazu beigetragen hat, dass die Auswertung des kleinen Zeitraums ein guter, und die des großen ein vergleichsweiser schlechter Durchlauf war. Da die Verbesserung durch die zusätzliche Information jedoch ziemlich groß ausgefallen ist, handelte es sich unwahrscheinlich um einen Zufall. Weitere Tests mit dem kompletten Zeitraum zeigten eine geringfügige Verbesserung. Der Fehler war dennoch lediglich um ca. 3% niedriger als die Vergleichswerte aus dem ersten Teil des Experiments. Daraus geht hervor, dass die Ergebnisse mit dem kleineren Zeitraum weitaus bessere Werte aufweisen.
Diese Ergebnisse spiegeln die Erkenntnis aus Abschnitt 4.1.1 wider. In diesem wurde Campylobacter ebenfalls mit dem kompletten Zeitraum als Training vorhergesagt. Die Ergebnisse der Regressions-Modelle hatten in diesem Test auch nahezu keine Verbesserung im RMSE. Den ganzen Zeitraum zu verwenden lohnt sich hier daher nur, wenn zusätzlich alle übrigen Vorhersage-Modelle untersucht werden.

Tabelle 12: Ergebnisse des Vorhersage-Evaluators ohne Dekomposition und Dekomposition jedoch inklusive zusätzlicher Inputs. Daten: Fallzahlen von Campylobacter-Enteritis 2014-2018

Fallzahl: Campylobacter-Enteritis					
Input	Reisezeit	Vorhersage-Modell	MAE	RMSE	MASE
-	-	ES-R	**3.5387**	4.8849	**0.7518**
-	-	RF-31	3.5750	**4.6628**	0.7595
Variante 1	60	RF-49	3.3930	4.5802	0.7208
Variante 2	60	RF-45	3.6430	4.5307	0.7739
Variante 3	60	RF-11	3.5220	4.5300	0.7482
Variante 4	60	RF-48	3.3230	4.4828	0.7060
Variante 1	**30**	**RF-42**	**3.2160**	**4.3961**	**0.6832**
Variante 2	30	RF-37	3.7480	4.6630	0.7962
Variante 3	30	RF-33	3.6270	4.5823	0.7705
Variante 4	30	RF-33	3.3550	4.4670	0.7128
Variante 4	60	LR-2	3.3774	4.6462	0.7175
Variante 4	30	LR-6	3.5425	4.5912	0.7526

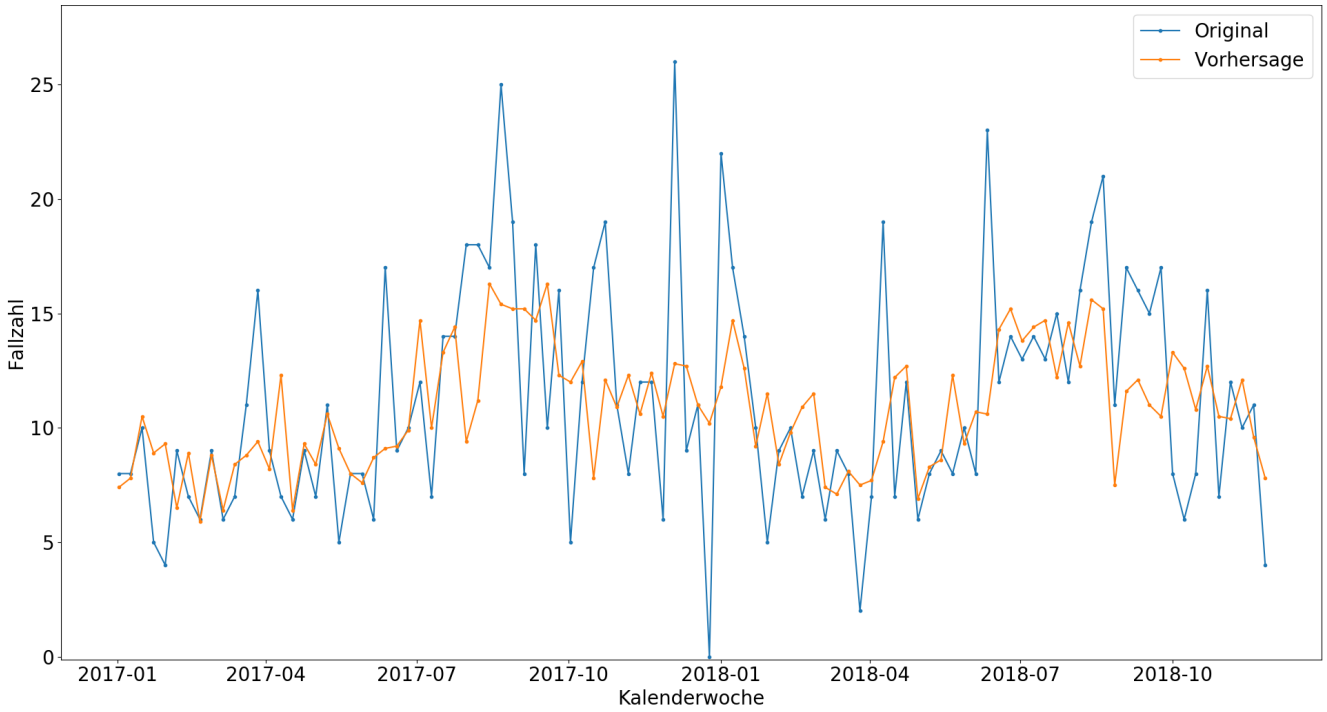

Abbildung 16: Vorhersage des besten Modells im Vorhersage-Evaluator ohne Dekomposition und Dekomposition jedoch inklusive zusätzlicher Inputs. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Campylobacter-Enteritis, 2014-2018 Modell: Random Forest Regression mit einer Fenstergröße von 42, Zusätzliche Information: Variante 1 mit einer Reisezeit von 30 Minuten

Vorhersage mit Transformations- und Dekompositions-Verfahren

Im folgenden Schritt werden zusätzlich ausgewählte Transformationen und Dekompositionen hinzugefügt. Die Auswahl wurde auf Basis der Ergebnisse aus dem letzten Abschnitt getroffen. Als Dekomposition wird lediglich die STL-Dekomposition verwendet, da die anderen Verfahren die Vorhersage bei Campylobacter verschlechtert haben. Es hatte sich herausgestellt, dass die einzige Transformation, welche positive Ergebnisse erbrachte, die ABC-Transformation mit $\lambda_1 = 0,5$ war. Zusätzlich wird in diesem Test die Original-Zeitreihe an die Dekomposition übertragen. In Tabelle 13 sind die Ergebnisse aufgelistet. Dargestellt ist hier, zu jeder Variante von zusätzlichen Informationen sowie Reisezeit, lediglich die beste Kombination an Transformation und Dekomposition wie auch die zugehörigen Vorhersage-Modelle der Dekompositions-Komponenten. In den ersten beiden Zeilen sind ebenfalls die besten Vorhersagen der vorherigen Teile des Experiments abgebildet.

Die Ergebnisse zeigen, dass der RMSE, im Vergleich zu den Ergebnissen der letzten beiden Abschnitte, in jeder Variante, sowie bei 30 und 60 Minuten Reisezeit, eine Verbesserung aufweist. Diese beträgt im besten Fall knapp 7% zu den Ergebnissen des Vorhersage-Evaluator, welcher bereits eine hohe Fehlerminimierung realisierte. Die beste Variante hat im Vergleich zu der normalen Vorhersage, ohne Transformation und Dekomposition, einen um ca. 12% besseren RMSE und um ca. 13% besseren MAE. Der MAE ist in diesem Fall schlechter, als bei der Vorhersage vom Vorhersage-Evaluators ohne zusätzliche Inputs. Da hier jedoch nach dem besten RMSE gesucht wurde, hat dies eher wenig Relevanz. Zweifellos wäre es besser, wenn das Modell zufälligerweise in beiden Maßen das Beste wäre, aber dieser Aspekt steht in diesem Fall nicht im Vordergrund.

Äußerst auffällig ist, dass die Random Forest Regression durch die zusätzlichen Informationen an Priorität gewonnen hat. In den Ergebnissen vom Vorhersage-Evaluator im letzten Abschnitt, welche in Tabelle 10 auf Seite 53 dargestellt sind, wurde in den fünf besten Vorhersagen à drei Vorhersage-Modelle lediglich zwei Mal die Random Forest Regression verwendet. In diesem Abschnitt hingegen ist in jedem Ergebnis mindestens für eine Komponente die Random Forest Regression vertreten. Ebenfalls wurde in jeder Variante die Original-Zeitreihe an die Dekomposition übertragen. In der Reisezeit hat sich jedoch kein klarer Favorit gekennzeichnet. Die teilweise besseren Fehler können sich auch auf die Varianz des Fehlers der Random Forest Regression zurückführen lassen.

Die Kombination des Vorhersage-Evaluators und der zusätzlichen Informationen zeigte ohne Ausnahme eindeutige Verbesserungen auf. Im folgenden Abschnitt wird an der Krankheit Keuchhusten überprüft, ob sich die gleichen Verbesserungen aufzeigen.

Tabelle 13: Ergebnisse des Vorhersage-Evaluators inklusive zusätzlicher Inputs. Daten: Fallzahlen von Campylobacter-Enteritis 2014-2018

Fallzahl: Campylobacter-Enteritis									
Input	Reisezeit	Transf.	Dekomp.	Trend	Saison	Rest	MAE	RMSE	MASE
-	-	Original	Original	-	-	RF-31	3.5750	4.6628	0.7595
-	-	Original	STL-13	Average	ES-R	LR-3	3.0929	4.4046	0.6571
V1	60	Original	STL-20	Average	RF-46	ES-HW	3.2537	4.3120	0.6912
V2	60	Original	STL-20	Average	RF-43	RF-9	3.4955	4.3759	0.7426
V3	60	Original	STL-20	Average	RF-45	RF-24	3.3981	4.3484	0.7219
V4	60	Original	STL-20	Average	RF-6	ARIMA-R	3.3257	4.2542	0.7065
V1	30	Original	STL-13	Average	LR-5	RF-10	3.3445	4.3462	0.7105
V2	30	Original	STL-20	Average	ES-R	RF-7	3.2976	4.3792	0.7006
V3	**30**	**Original**	**STL-13**	**Average**	**ES-R**	**RF-9**	**3.1316**	**4.1256**	**0.6653**
V4	30	Original	STL-7	Average	RF-2	ES-HW	3.3345	4.2050	0.7084

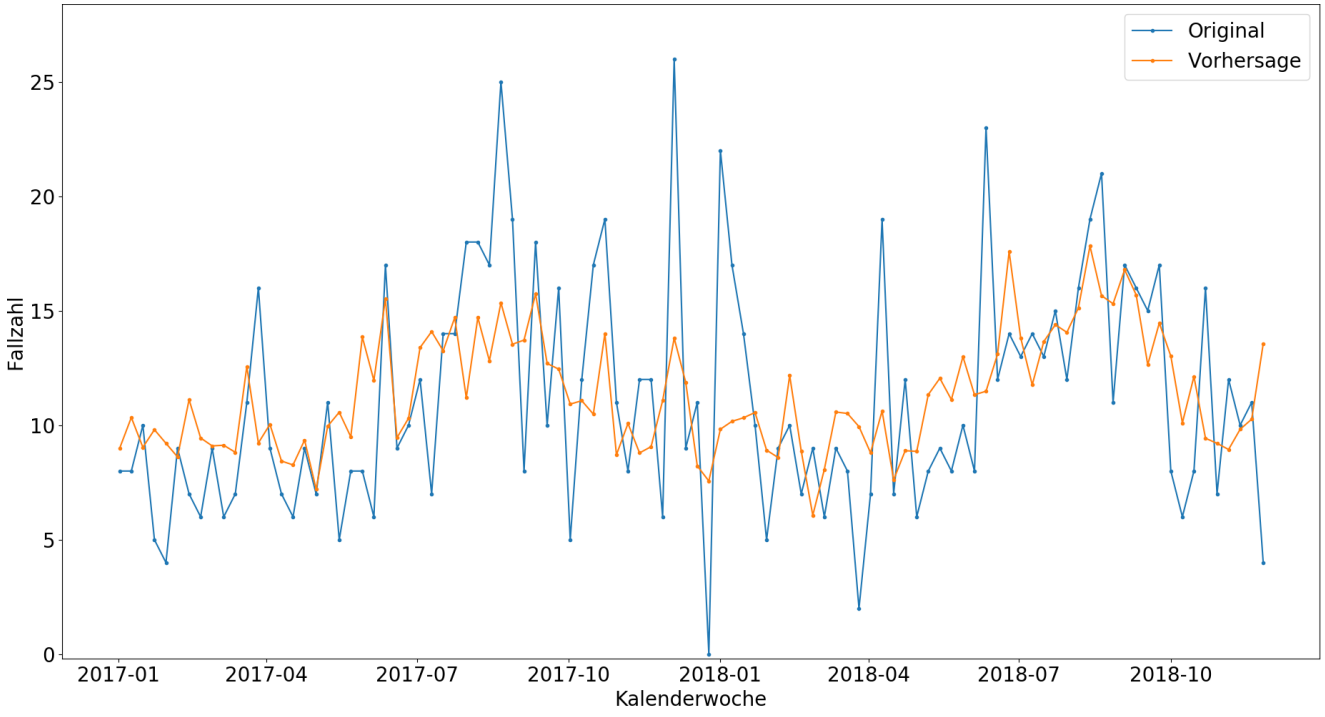

Abbildung 17: Vorhersage des besten Modells im Vorhersage-Evaluator inklusive zusätzlicher Inputs. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Campylobacter-Enteritis 2001-2018, Dekomposition: STL-13 (Trend-Modell: Average, Saison-Modell: Exponential Smooting mit R-Paket, Rest-Modell: Random Forest Regression Fenstergröße=9)

Im letzten Schritt des Experiments wurde Keuchhusten mit Hilfe der zusätzlichen Informationen der Nachbar-Zeitreihe vorhergesagt. Genau wie bei Campylobacter wird die Vorhersage zunächst ohne Transformationen und Dekompositionen durchgeführt. Anschließend werden jene, welche in Abschnitt 4.1.2 die Besten waren, im Evaluator zusammen mit den zusätzlichen Informationen geprüft. Der betrachtete Zeitraum ist, wie er bereits im ersten Teil des Experiments definiert, auf 2013-2018 festgelegt.

Vorhersage ohne Transformations- und Dekompositions-Verfahren

Die Ergebnisse von dem ersten Test sind in Tabelle 14 abgebildet. In dieser ist klar erkennbar, dass die Variante 1 und 3 denselben Effekt auf die Vorhersage haben, wodurch die Fehler dieser identisch sind. Bei der Linearen Regression scheint es daher irrelevant zu sein, ob dem Modell die Haupt- sowie Nachbar-Zeitreihe oder die Haupt-Zeitreihe sowie Abweichung zur Nachbar-Zeitreihe übergeben wird. Das Ergebnis ist in beiden Fällen dasselbe. Bei Campylobacter hingegen zeigte sich bei der Random Forest Regression zwischen den Varianten ein großer Unterschied im Fehler, welcher nicht auf die Varianz des Fehlers zurückzuführen ist.

Die besten Ergebnisse erzielte erneut, wie bei Campylobacter, die erste Variante mit einer Reisezeit von 30 Minuten. Im Vergleich zu der besten Vorhersage aus dem ersten Teil des Experiments, welche zusätzlich in der ersten Zeile der Tabelle dargestellt ist, zeigt sich eine eindeutige Verbesserung. Der RMSE verringerte sich um ungefähr 5% und der MAE um knapp 8%. Diese Ergebnisse sind, genauso wie bei Campylobacter, eine Spur besser als die des Vorhersage-Evaluators.

Auffällig ist hier, dass in jeder Variante die Lineare Regression die besten Ergebnisse aufweist. Jedoch ist dies nicht sehr verwunderlich, da diese bereits im ersten Teil des Experiments den kleinsten Fehler produzierte. Die Lineare Regression zeigte dort bereits eine deutliche Überlegenheit gegenüber den restlichen Vorhersage-Modellen.

In Abbildung 18 ist die Vorhersage mit dem niedrigsten RMSE dargestellt. Diese passt sich, im Vergleich zu der Vorhersage aus dem ersten Teil des Experiments, viel besser an die Varianz der Original-Zeitreihe an. Im ersten Teil ähnelte die Vorhersage noch eher einer Glättung mit kaum wahrnehmbaren Ausschlägen in positiver oder negativer Richtung. Hier hingegen scheint die Vorhersage optisch auf jeden Fall realitätsgetreuer. Einzig allein die Ausschläge um Januar 2018 herum konnte nicht gut vorhergesagt werden. Diese sind aber auch extreme Ausreißer, welche den Maximal- und Minimalwert der ganzen zwei Jahre beinhalten. In diesem Bereich ist es daher sehr schwierig, eine gute Vorhersage zu bestimmen.

Tabelle 14: Ergebnisse des Vorhersage-Evaluators inklusive zusätzlicher Inputs. Daten: Fallzahlen von Keuchhusten 2013-2018

Fallzahl: Keuchhusten

Input	Reisezeit	Vorhersage-Modell	MAE	RMSE	MASE
-	-	LR-2	1.8835	2.3562	0.8476
Variante 1	60	LR-1	1.7686	2.2810	0.7959
Variante 2	60	LR-2	1.8802	2.3537	0.8461
Variante 3	60	LR-1	1.7686	2.2810	0.7959
Variante 4	60	LR-1	1.7726	2.2950	0.7977
Variante 1	**30**	**LR-1**	**1.7469**	**2.2507**	**0.7861**
Variante 2	30	LR-2	1.8654	2.3416	0.8394
Variante 3	30	LR-1	1.8101	2.2765	0.8145
Variante 4	30	LR-1	1.8137	2.2977	0.8161

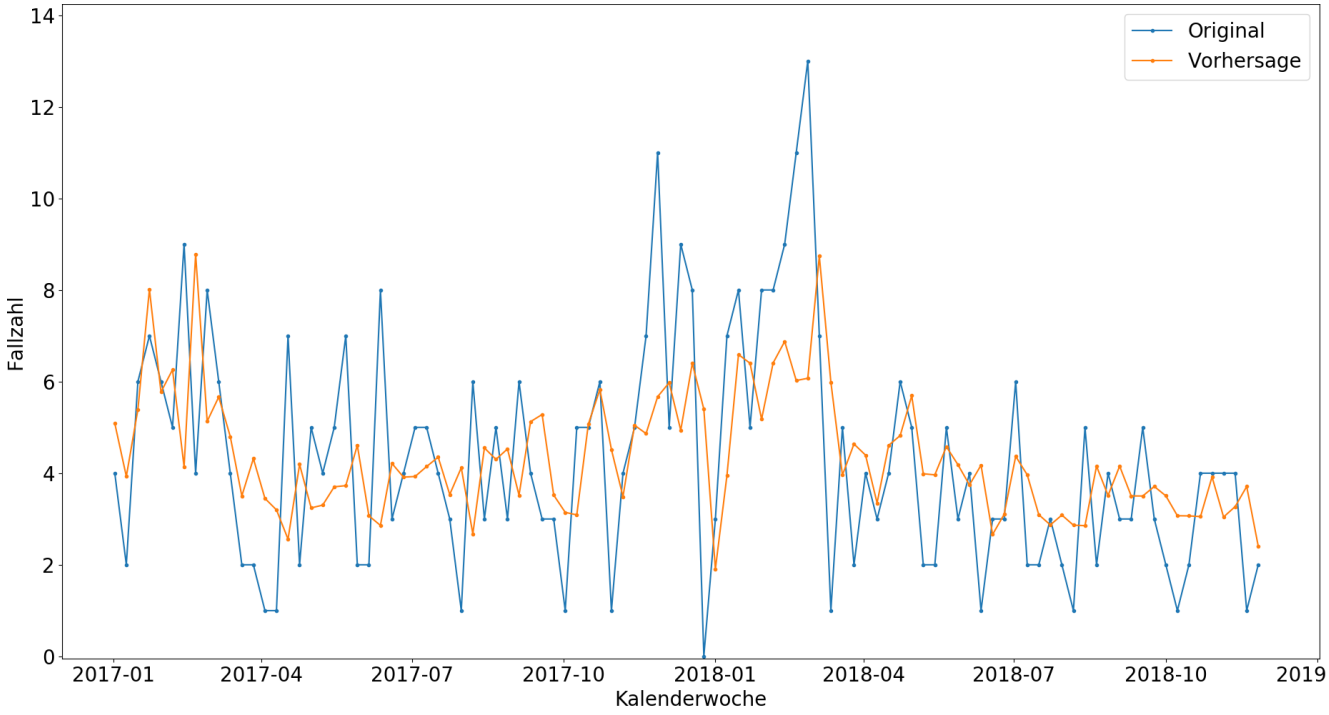

Abbildung 18: Vorhersage des besten Modells im Vorhersage-Evaluator inklusive zusätzlicher Inputs. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Keuchhusten 2013-2018, Modell: Lineare Regression mit einer Fenstergröße von 1, Zusätzliche Information: Variante 1 mit einer Reisezeit von 30 Minuten

Tabelle 15: Ergebnisse des Vorhersage-Evaluators inklusive zusätzlicher Inputs. Daten: Fallzahlen von Keuchhusten 2013-2018

Fallzahl: Campylobacter-Enteritis									
Input	Reisezeit	Transf.	Dekomp.	Trend	Saison	Rest	MAE	RMSE	MASE
-	-	Original	Original	-	-	LR-2	1.8835	2.3562	0.8476
-	-	Original	STL-200	S-Naive	Naive	LR-9	1.8181	2.2849	0.8181
V1	60	ABC-1.5	STL-20	Average	LR-21	RF-31	1.8396	2.2325	0.8278
V2	60	ABC-1.5	STL-200	S-Naive	Naive	LR-9	1.8176	2.2816	0.8179
V3	60	ABC-1.5	STL-200	S-Naive	Naive	LR-1	1.7855	2.2373	0.8035
V4	**60**	**ABC-1.5**	**STL-20**	**Average**	**Naive**	**RF-2**	**1.6894**	**2.1023**	**0.7602**
V1	30	ABC-1.5	STL-20	Average	Drift	RF-6	1.7304	2.2057	0.7787
V2	30	ABC-1.5	STL-200	S-Naive	Naive	LR-9	1.8031	2.2696	0.8114
V3	30	ABC-1.5	STL-200	S-Naive	Naive	LR-1	1.7726	2.2265	0.7977
V4	30	ABC-1.5	Original	-	-	LR-1	1.7591	2.2526	0.7916

Vorhersage mit Transformations- und Dekompositions-Verfahren

Abschließend wurde der Vorhersage-Evaluator auf die Zeitreihe von Keuchhusten angewandt. Auch hier wurden die Regressions-Modelle mit den acht Variationen an zusätzlichen Informationen erweitert. Da die ABC-Transformation mit $\lambda_1 = 1,5$ und $\lambda_1 = 2$ in Abschnitt 4.1.2 die besten Ergebnisse produzierten, werden lediglich diese Transformationen im Folgenden betrachtet. Als Dekomposition wurde, wie bei Campylobacter, alleinig die STL-Transformation verwendet.

Die Ergebnisse dazu sind in Tabelle 15 abgebildet. In dieser sind ebenfalls die Ergebnisse der besten Vorhersagen von Keuchhusten aus dem ersten und zweiten Teil des Experiments. Das Hinzufügen der zusätzlichen Informationen konnte ohne Ausnahme den Fehler verbessern, wobei es sich nicht immer um eine deutliche Verbesserung handelte. Im Vergleich zu den Ergebnissen des Evaluators aus dem zweiten Teil des Experiments wurde der RMSE bei der besten Vorhersage hier um knapp 8% verringert. Der MAE zeigte ebenfalls eine Verbesserung von ungefähr 7% auf. Diese Verbesserungen sind sehr ähnlich zu denen bei Campylobacter. Wird diese Vorhersage mit der normalen Vorhersage aus Abschnitt 4.1.1 verglichen, dann ergibt sich sogar eine Verminderung des RMSE und MAE von ca. 11%. Die Verringerung des Fehlers ist bei Campylobacter minimal größer ausgefallen. Dies bestätigt wiederum die Vermutung, dass Campylobacter, bedingt durch seine scheinbar einfachen Muster, besser vorherzusagen ist.

In Abbildung 19 ist die beste Vorhersage, welche in der Tabelle fett markiert ist, dargestellt. Im Vergleich zu der Vorhersage auf der vorherigen Seite wurde die extreme Abnahme der Fallzahlen im April 2017 frühzeitig erkannt. Ebenfalls wurde der Nullwert im Januar 2018 besser angenähert. In der Vorhersage in Abbildung 18 war die Verminderung lediglich eine Reaktion auf den Nullwert. Hier wurde dieser bereits eine Woche vorher erkannt. Die positiven Ausschläge, um diesen Nullwert herum, konnten jedoch nur teilweise erkannt werden.

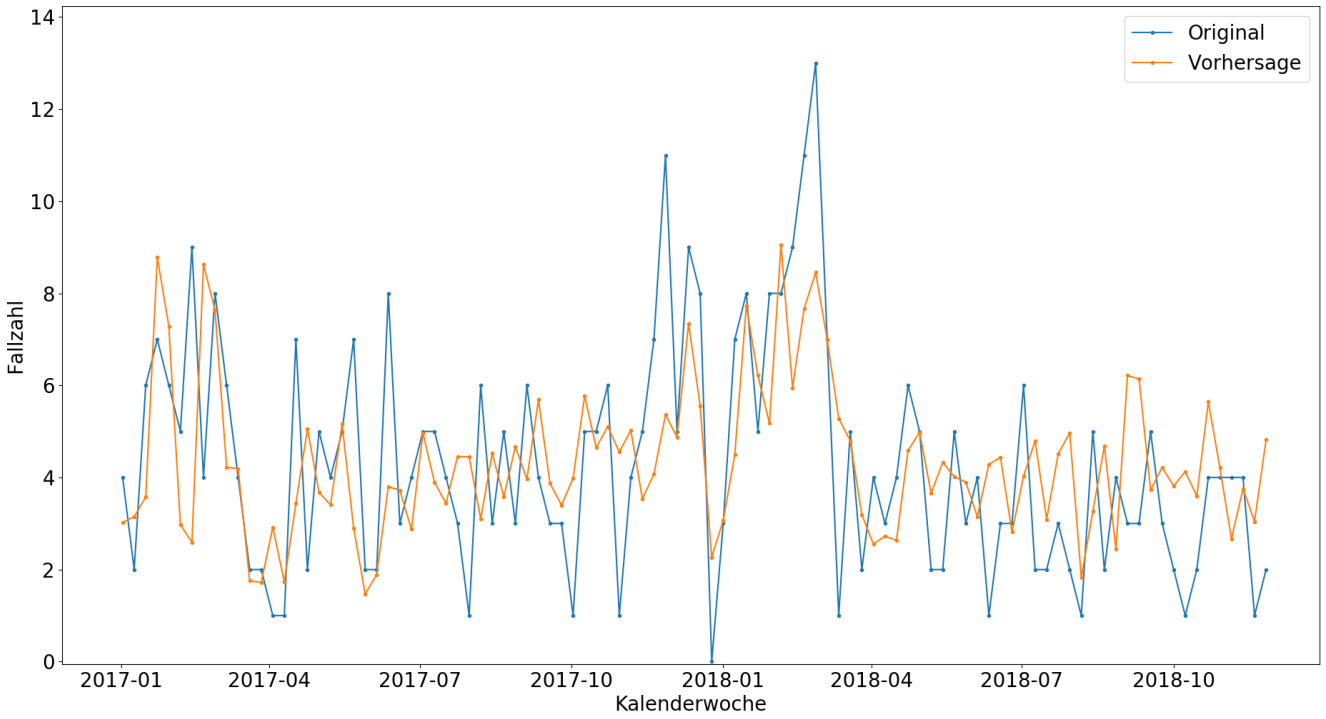

Abbildung 19: Vorhersage des besten Modells im Vorhersage-Evaluator inklusive zusätzlicher Inputs. Evaluation der letzten 100 Zeitpunkte. Daten: Fallzahlen von Keuchhusten 2013-2018, Transformation: ABC mit $\lambda_1 = 1,5$, Dekomposition: STL-20 (Trend-Modell: Average, Saison-Modell: Naive, Rest-Modell: Random Forest Regression Fenstergröße=2)

4.2 Fazit

Das Experiment war in drei Schritte aufgeteilt. Im Ersten wurden die Zeitreihen von Campylobacter-Enteritis und Keuchhusten mit Vorhersage-Modellen, welche zurzeit die renommiertesten Techniken sind, mit einer Time Series Cross-Validation über 100 Zeitperioden vorhergesagt. Diese Vorhersagen wurden als Vergleich für die Ergebnisse aus dem zweiten und dritten Schritt des Experiments genutzt. Im zweiten Schritt wurde der Vorhersage-Evaluator, inklusive der Transformations- und Dekompositions-Verfahren, zur Vorhersage verwendet. Abschließend wurde im letzten Schritt der Evaluator um die zusätzlichen Informationen von den Nachbar-Kreisen erweitert.

Die Anwendung des Vorhersage-Evaluator zeigte bei beiden Krankheiten eine klare Verbesserung des Fehlers. Unter Betrachtung des RMSE betrug diese in beiden Fällen ca. 6%. Der MAE hingegen verbesserte sich bei Campylobacter sogar bis zu 15% und das, obwohl nach dem besten RMSE gesucht wurde. Mit der Erweiterung der Regressions-Modelle um die zusätzlichen Inputs wurde der Fehler der Vorhersage erneut beträchtlich verringert. In der Summe wurde der MAE und RMSE bei Campylobacter im Vergleich zu der normalen Vorhersage, ohne jegliche Erweiterung der Modelle bzw. Evaluations-Strategie, um knapp 13% verbessert. Bei Keuchhusten hingegen wurde der Fehler nur um ca. 11% verringert. Dies könnte mit der bereits erwähnten Vermutung zusammenhängen, dass optisch keine eindeutigen Muster zu erkennen sind. Die Zeitreihe scheint größtenteils einer Normalverteilung zu folgen. Bei Campylobacter ist definitiv zusätzlich ein hoher Anteil an Rauschen vorhanden, gleichwohl ebenfalls eine Saisonalität erkennbar ist. Unter Betrachtung der letzten vier Jahre der Zeitreihe von Campylobacter auf Seite 40 ist ebenfalls ein leichter positiver Trend sichtbar. Dies ist ein einfach vorhersagebares Muster. Es sollte je-

doch nicht außer Acht gelassen werden, dass die Verfahren im ersten Schritt des Experiments bereits versuchen, diese Muster möglichst gut zu bestimmen. Diese Annäherungen der Muster wurden in den zwei darauffolgenden Schritten optimiert.
Wird die Verwendung der zusätzlichen Daten der Nachbarn mit dem Evaluator verglichen, zeigt sich, dass die Vorhersagen mit Hilfe der Daten der Nachbarn, ohne großen Aufwand oder hohe Rechenarbeit, in besseren Ergebnissen resultieren. Auch wenn die Laufzeit in dieser Arbeit zweitrangig war, sollte hier erwähnt werden, dass diese, im Falle der Wahl mehrerer Transformationen sowie Dekompositionen, sehr hoch ist. Die Hinzunahme der zusätzlichen Informationen hingegen verursacht in dem Vorhersage-Modell quasi keine extra Laufzeit. Daher sollten die Informationen der Nachbar-Zeitreihe immer mit in die Vorhersage einfließen, egal ob ein schnelles oder detailliertes Ergebnis gefordert ist. Allerdings waren die Variationen der zusätzlichen Informationen in dieser Arbeit auf acht Stück festgelegt. Würden hier ebenfalls besonders viele verschiedene getestet werden, könnte selbst dies in einer hohen Laufzeit resultieren. Der Evaluator sollte für detailreiche Vorhersagen ebenfalls immer zur Hand genommen werden. Falls jedoch schnell ein Ergebnis erforderlich ist, sollte dieser nicht verwendet werden, da die Auswertung, je nach Konfiguration, schnell bis zu einem Tag oder sogar länger dauern kann. In Kapitel 5 wird abschließend kurz auf die Laufzeit und mögliche zukünftige Optimierungen dieser eingegangen.
Eine weitere wichtige Erkenntnis war, dass eine größere Menge an Daten zum Training nicht gleichzeitig in besseren Ergebnissen resultiert. Gezeigt wurde dies an Campylobacter. Einige Vorhersage-Modelle konnten die zusätzlichen Datenpunkte verwenden, um eine bessere Vorhersage zu generieren. Andere produzieren mit diesen zusätzlichen Daten hingegen schlechtere Ergebnisse als mit der Verwendung eines kleineren Trainings-Datensatzes.

5 Resümee

In dieser Arbeit wurden Strategien ausgearbeitet, welche die Vorhersage von epidemiologischen Zeitreihen in Frankfurt am Main optimieren. Diesbezüglich wurden zunächst renommierte Verfahren zur Vorhersage recherchiert, mit denen die Zeitreihen der Krankheitsfallzahlen von Campylobacter-Enteritis und Keuchhusten vorhergesagt wurden. Für diese Vorhersage wurde als Evaluations-Strategie die Time Series Cross-Validation angewandt. Die Fehler dieser Vorhersagen dienten für den Rest der Arbeit als Vergleichswert für neu entwickelte Ideen bzw. Strategien. Zwei Optimierungsansätze wurden in dieser Arbeit ausführlich analysiert. Diese zeigten beide eine klare Verringerung des Fehlers der Vorhersage.
Bei dem ersten Ansatz handelte es sich um den Vorhersage-Evaluator. In diesem wurden einige vorgestellte Verfahren zur Transformation und Dekomposition kombiniert. Das Ziel von dem Evaluator war, eine Transformation und Dekomposition zu finden, deren Einsatz die Vorhersage bestmöglich optimiert. Dieser Ansatz reduzierte den Fehler in der Vorhersage von beiden Krankheiten um ungefähr 7% und war daher ein klarer Erfolg. Einen Nachteil des Evaluators stellt die Laufzeit dar, welche in dieser Arbeit jedoch zweitrangig war. Im nächsten Abschnitt, dem Ausblick, wird auf mögliche Laufzeitoptimierungen eingegangen.
Im zweiten Ansatz wurden zusätzliche Informationen generiert, welche mit in die Vorhersage der vorgestellten Regressions-Modelle eingeflossen sind. Diese Informationen wurden aus den Fallzahlen der betrachteten Krankheit, von den Land- und Stadtkreisen aus der regionalen Umgebung von FFM, erzeugt. Hierzu wurden vier Varianten von zusätzlichen Informationen erarbeitet und evaluiert. Diese wurden auf Basis von zwei unterschiedlichen Nachbar-Zeitreihen generiert. Für die Erstellung der Nachbar-Zeitreihe musste festgelegt werden, welche Kreise mit in diese aufgenommen werden. Dafür wurde die Reisezeit eingeführt, welche in der Auswertung auf 30 und 60 Minuten definiert wurde. Es zeigte sich kein klarer Trend der Verbesserung in den unterschiedlichen Varianten der Information, was dazu führt, dass in Zukunft immer alle getestet werden müssen. In der Auswertung stellte die Variante mit der besten Vorhersage eine deutliche Verbesserung zu der normalen Vorhersage aus dem ersten Teil des Experiments dar. Der Fehler konnte bei beiden Krankheiten um ca. 5% reduziert werden. Der Vorteil in diesem Ansatz ist, dass die Laufzeit nicht merklich ansteigt. Ein Nachteil wiederum ist, dass aktuell lediglich vier ausgewählte Varianten von zwei unterschiedlichen Nachbar-Zeitreihen betrachtet werden. Mögliche Erweiterungen hierzu werden jedoch im Ausblick angesprochen.
Zusätzlich wurden die zwei Ansätze kombiniert betrachtet. Die Erwartung war, dass die Kombination dieser in einer noch besseren Reduzierung des Fehlers resultiert. In der Auswertung wurden als Erstes die besten Transformationen und Dekompositionen ermittelt. Anschließend wurden mit diesen acht Testläufe durchgeführt, wobei die Regressions-Modelle in jedem Durchlauf andere, zusätzliche Informationen zur Verfügung hatten. Die Ergebnisse waren in jedem Testlauf besser als die der besten Vorhersage des Evaluators aus dem ersten Ansatz. Der Fehler konnte im besten Durchlauf bei beiden Krankheiten, im Vergleich zu der einfachen Vorhersage, um ungefähr 13% reduziert werden. Diese Ergebnisse hatten jegliche Erwartungen übertroffen.

Ausblick

Die Arbeit bietet mehrere Stellen, welche erweitert bzw. optimiert werden können. Eine dieser Stellen wurde bereits im Unterkapitel 3.5 beschrieben, zwei weitere wurden ebenfalls in vorherigen Kapiteln

erwähnt. Die angesprochene Laufzeitoptimierung ist teilweise in den folgenden Erweiterungen vertreten, da einige dieser bereits die Laufzeit verringern. Bevor jedoch auf die Erweiterungen eingegangen wird, muss erläutert werden, warum die Laufzeit von enormer Bedeutung ist. Eine Vorhersage der Fallzahlen wird niemals perfekt sein, da in der Zeitreihe immer ein Zufallsfaktor verschlüsselt sein wird. Es muss, je nach Anwendungsszenario und der zur Verfügung stehenden Zeit, festgelegt werden, wie genau die Vorhersage letztendlich sein sollte. Aus dieser Grundlage wird anschließend das Verfahren entwickelt. In dem Szenario dieser Arbeit werden Fallzahlen der nächsten Woche vorhergesagt. Daher sollte die Laufzeit bestenfalls unter einem Tag bleiben, aber auf keinen Fall über zwei Tage hinaus ansteigen. In den folgenden Erweiterungen besteht das Ziel darin, die Vorhersage weiter zu verbessern und bestenfalls die Laufzeit zu verringern bzw. zu halten. Dies ist jedoch nicht immer möglich.

Als Erstes werden die Regressions-Modelle betrachtet. Es wurde im Abschnitt 3.2.1 gezeigt, dass der Fehler bezüglich der Fenstergröße nicht monoton steigt oder fällt. Eine Erweiterung wäre, die Fenstergröße automatisch zu wählen, sodass der Fehler minimiert wird. Hierzu müsste eine Minimierungsstrategie ausgearbeitet werden, welche nicht in einem lokalen Minimum des Fehlers festhängt. Mit dieser Erweiterung würde die Menge an Vorhersage-Modellen um eine Vielzahl kleiner werden. Dadurch würde der Aufwand einer Zusammenführung von Dekompositions-Komponenten beträchtlich sinken. Zusätzlich sollte bei einer guten Minimierungsstrategie der letztendliche Fehler geringer sein, als von den Modellen mit fester Fenstergröße.

Eine weitere Idee für die Random Forest Regression wäre, die Anzahl der Bäume automatisch zu wählen. Für diese Erweiterung muss im Vorhinein abgeschätzt werden, wie viel Zeit der Evaluator mit den gewählten Einstellungen benötigen wird. Abhängig von dieser Dauer und davon, wie lange die Auswertung maximal dauern darf, müsste dahingehend die Anzahl der Bäume definiert werden. Hierbei handelt es sich daher eher um ein Feintuning des Modells.

Die folgenden Anpassungen betreffen den Vorhersage-Evaluator. In der zukünftigen Anwendung wird dieser je Krankheit einmal pro Woche ausgeführt, um die Vorhersage für die nächste Woche zu bestimmen. Die zur Verfügung stehenden Daten ändern sich in einer Woche jedoch kaum. Es wird der Datenpunkt der neuen Woche hinzugefügt und etwaige Fehler oder Änderungen in der Vergangenheit werden angepasst. Daher ist es unwahrscheinlich, dass Verfahren, die vorher schlecht performt haben, in dem neuen Durchlauf gute Ergebnisse erzielen. Aufgrund dessen könnte vorerst eine abgemagerte Version des Evaluators ausgeführt werden. In dieser werden alle Transformationen, Dekompositionen und zusätzlichen Informationen, welche in der Woche zuvor keine guten Ergebnisse erzielt haben, nicht verwendet. Mit dieser Variante könnte relativ schnell ein gutes Ergebnis erzielt werden. Anschließend wird die ausführliche Version ausgeführt, um eine präzisere Aussage zu haben. Ebenfalls bestimmen die Ergebnisse der ausführlicheren Auswertung die Verfahren für die nächste Woche.

In der Auswertung stellte sich heraus, dass ein größerer Trainings-Datensatz nicht zwingend in einer besseren Vorhersage resultiert. Eine mögliche Erweiterung des Evaluators könnte sein, dass er selbständig überprüft, wie viele Daten aus den vorhandenen benötigt sind. Bezüglich dieser Wahl wäre das Ziel, den Fehler in der Vorhersage zu minimieren. Zusätzlich sollte hier darauf geachtet werden, dass der Datensatz nicht zu groß wird. In diesem Fall würde sehr wahrscheinlich die Laufzeit zu sehr darunter leiden. Da einige Modelle scheinbar keinen Vorteil von einem größeren Trainings-Datensatz hatten, wäre es auch möglich, den unterschiedlichen Modellen verschiedene Größen von Datensätzen zu übergeben.

In der letzten Erweiterung werden die zusätzlichen Informationen betrachtet. Insgesamt wurden in dieser Arbeit acht unterschiedliche Varianten ausgewählt und anschließend verwendet. Möglicherweise wäre es möglich, im Vorhinein die beste Reisezeit und Variante für die gegebene Krankheit zu ermitteln. Hierzu müsste eine ausführliche Studie durchgeführt werden, in der getestet wird, wie sich der Fehler verhält. Zusätzlich könnten weitere Varianten der Informations-Generierung aus den Nachbar-Zeitreihen ausgearbeitet werden.

Literaturverzeichnis

Breiman, Leo (2001). *Random Forest*. In: *Machine Learning* 45, S. 5–32.

Chen, Zhuo und Yuhong Yang (Apr. 2004). *Assessing Forecast Accuracy Measures*. Draft.

Cleveland, Robert B. u. a. (1990). *STL: A Seasonal-Trend Decomposition Procedure Based on Loess*. In: *Journal of Official Statistics* 6, S. 3–73.

Cleveland, William S. (Dez. 1979). *Robust Locally Weighted Regression and Smoothing Scatterplots*. In: *Journal of the American Statistical Association* 74, S. 829–836.

Cleveland, William S. und Clive Loader (Aug. 1996). *Smoothing by Local Regression: Principles and Methods*. In: *Härdle W., Schimek M.G. (eds) Statistical Theory and Computational Aspects of Smoothing. Contributions to Statistics. Physica-Verlag HD*, S. 10–49.

Franses, Philip Hans (März 2016). *A note on the Mean Absolute Scaled Error*. In: *International Journal of Forecasting* 32, S. 20–22.

Holt, Charles C. (2004). *Forecasting seasonals and trends by exponentially weighted moving averages*. In: *International Journal of Forcasting* 20.

Hyndman, Rob J. und George Athanasopoulos (2018). *Forecasting: Principles and Practice*. 2nd print edition. OTexts. 382 S.

Hyndman, Rob J. und Yeasmin Khandakar (Juli 2008). *Automatic Time Series Forecasting: The forecast Package for R*. In: *Journal of Statistical Software* 27.

Hyndman, Rob J. und Anne B. Koehler (2006). *Another look at measures of forecast accuracy*. In: *International Journal of Forecasting* 22, S. 679–688.

Kiehl, Wolfgang (2015). *RKI-Fachwörterbuch Infektionsschutz und Infektionsepidemiologie*. Berlin.

Lütkepohl, Helmut und Fang Xu (Juni 2009). *The Role of log Transformation in Forecasting Economic Variables*. In: *Empirical Economics* 42, 619–638.

Mitchell, Thomas (1997). *Machine Learning*. McGraw-Hill Education Ltd. 432 S.

Raychaudhuri, Samik (Dez. 2008). *Introduction to Monte Carlo simulation*. In: S. 91–100.

Sarkar, Ram Rup und Chandrajit Chatterjee (2017). *Application of Different Time Series Models on Epidemiological Data - Comparison and Predictions for Malaria Prevalence*. In: *SM Journal of Biometrics and Biostatistics* 2(4), S. 1022–1031.

Schneider, Astrid, Gerhard Hommel und Maria Blettner (2010). *Linear Regression Analysis*. In: *Deutsches Ärzteblatt International* 107(44), S. 776–782.

Steinruecken, Christian u. a. (2018). *The Automatic Statistician*. Draft.

A Abkürzungsverzeichnis

ABC-Transformation	Angepasste-Box-Cox-Transformation
AIC	Akaike's Information Criterion
AR	Autoregressives
ARIMA	Autoregressive Integrated Moving Averages
nARIMA	Non-Seasonal ARIMA
sARIMA	Seasonal ARIMA
ARIMA-R	ARIMA in R-Paket Variante
ES	Exponential Smoothing
ES-HW	Exponential Smoothing in Holt-Winters Variante
ES-R	Exponential Smoothing in R-Paket Variante
FFM	Frankfurt am Main
KPSS-Test	Kwiatkowski-Phillips-Schmidt-Shin-Test
LK	Landkreis
Log-Transformation	Logarithmus-Transformation
LR	Lineare Regression
LSE	Least Square Estimation
LOWESS	Locally Weighted Estimated Scatterplot Smoothing
MA	Moving Averages
MAE	Mean Absolute Error
MASE	Mean Absolute Scaled Error
MC-Simulation	Monte-Carlo-Simulation
RF	Random Forest Regression
RKI	Robert Koch-Institut
RMSE	Root Mean Squared Error
S-Naive	Seasonal Naïve
SES	Simple Exponential Smoothing
SK	Stadtkreis