

BACHELOR THESIS  
Timo Andreas Türk

# Anwendung von Maschinellem Lernen und statistischer Verfahren zur Prognose von Energiedaten in Python

---

FAKULTÄT TECHNIK UND INFORMATIK  
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science  
Department of Information and Electrical Engineering

Timo Andreas Türk

# Anwendung von Maschinellem Lernen und statistischer Verfahren zur Prognose von Energiedaten in Python

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang *Bachelor of Science Elektro- und Informationstechnik*  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kolja Eger  
Zweitgutachter: Prof. Dr. Frerk Haase

Eingereicht am: 20. März 2024

**Timo Andreas Türk**

## **Thema der Arbeit**

Anwendung von Maschinellem Lernen und statistischer Verfahren zur Prognose von Energiedaten in Python

## **Stichworte**

Energiedaten, Python, statistische Verfahren, maschinellem Lernen, Prognose, Prophet, SARIMA

## **Kurzzusammenfassung**

Im Kontext ständig wechselnder Umwelteinflüsse streben viele Unternehmen eine kontinuierliche Optimierung ihrer Produktion und ihres Energieverbrauchs an. Diese Studie untersucht die indirekten Auswirkungen auf die Kostenreduzierung beim Strombezug von gewerblichen Kunden, die ihren Verbrauch alle 15 Minuten angeben müssen. Dabei wird auf maschinelles Lernen zurückgegriffen, um den präzisen Strombedarf vorherzusagen. Zwei Modelle, Prophet und SARIMA, werden eingehend analysiert und anhand historischer Daten miteinander verglichen. Das Hauptziel besteht darin zu evaluieren, ob Prophet und SARIMA geeignet sind, um genaue Prognosen bezüglich des Energieverbrauchs zu erstellen. Die verwendeten Daten stammen von der Informationsplattform SMARD der Bundesnetzagentur für den deutschen Strommarkt. Diese Untersuchung bietet wertvolle Einblicke in die Wirksamkeit verschiedener Prognosemodelle für den Energieverbrauch von Unternehmen und kann somit einen Beitrag zur Effizienzsteigerung und Kosteneinsparung in der Industrie leisten. Dabei weisen die besten Modelle eine Genauigkeit in der Neujahrsnacht von 2022 eine für zwei Stunden, eine Abweichung von SARIMA = 3,04% und Prophet = 0,57% auf.

---

**Timo Andreas Türk**

**Title of Thesis**

Application of machine learning and statistical methods for forecasting energy data in Python

**Keywords**

Energy data, Python, statistical methods, machine learning, forecasting, Prophet, SARIMA

**Abstract**

In the context of constantly changing environmental influences, many companies strive for continuous optimization of their production and energy consumption. This study examines the indirect effects on cost reduction in electricity procurement for commercial customers who must report their consumption every 15 minutes. Machine learning is used to predict precise electricity demand. Two models, Prophet and SARIMA, are thoroughly analyzed and compared using historical data. The main goal is to evaluate whether Prophet and SARIMA are suitable for creating accurate forecasts of energy consumption. The data used comes from the information platform SMARD of the Federal Network Agency for the German electricity market. This study provides valuable insights into the effectiveness of various forecast models for business energy consumption and can thus contribute to efficiency improvement and cost savings in the industry. The best models show an accuracy in predicting energy consumption for the New Year's Eve of 2022 within two hours, with a deviation of SARIMA = 3.04 % and Prophet = 0.57%.

# Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
Abkürzungsverzeichnis	ix
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>3</b>
2.1 Allgemeine Definitionen . . . . .	3
2.2 Grundlagen: Prophet . . . . .	4
2.3 Grundlagen: SMARD . . . . .	6
2.4 Grundlagen: SARIMA . . . . .	7
<b>3 Materialien</b>	<b>8</b>
3.1 Implementierung . . . . .	8
3.2 Aufarbeitung der SMARD Daten . . . . .	10
<b>4 Simulation</b>	<b>12</b>
4.1 Prophet . . . . .	12
4.1.1 Prophet: unangepasstes Modell . . . . .	13
4.1.2 Prophet: angepasstes Modell . . . . .	19
4.1.3 Prophet: weitere Modelle . . . . .	25
4.2 SARIMA . . . . .	26
4.2.1 SARIMA Modelle . . . . .	27
<b>5 Ergebnisse</b>	<b>30</b>
5.1 Kalibrierungsergebnisse . . . . .	30
<b>6 Beurteilung der Ergebnisse</b>	<b>32</b>
6.1 Vergleichbare Arbeiten und weitere Aussichten dieser Arbeit . . . . .	34

<b>7 Fazit</b>	<b>35</b>
<b>Literaturverzeichnis</b>	<b>37</b>
<b>A Anhang</b>	<b>40</b>
A.1 Verwendete Hilfsmittel . . . . .	40
<b>Selbstständigkeitserklärung</b>	<b>41</b>

# Abbildungsverzeichnis

2.1	Ablauf Grafik Prophet . . . . .	5
2.2	SMARD realisierte Erzeugung 2022 . . . . .	6
3.1	Fourier-Analyse der SMARD-Daten 2020/01-2022/12 . . . . .	10
3.2	SMARD realisierter Verbrauch . . . . .	11
4.1	Prophet Default Komponenten . . . . .	13
4.2	Prophet Default Werte, Prognose eine Woche . . . . .	14
4.3	Prophet Default Werte, Prognose 2 h . . . . .	15
4.4	Prophet MAPE unangepasst für zwei Stunden . . . . .	16
4.5	Prophet MAPE unangepasst . . . . .	17
4.6	Prophet Fehler Darstellung . . . . .	18
4.7	Prophet angepasstes Modell, Prognose eine Woche . . . . .	20
4.8	Prophet angepasstes Modell, Komponenten . . . . .	21
4.9	Prophet angepasstes Modell, MAPE zwei Stunden . . . . .	22
4.10	Prophet MAPE angepasst . . . . .	23
4.11	Prophet Fehler Darstellung vollständig angepasst . . . . .	24
4.12	Autokorrelationsfunktion . . . . .	26
4.14	SARIMA ohne dem Parameter m . . . . .	27
4.15	SARIMA mit dem Parameter m=2 . . . . .	28
4.16	SARIMA mit dem Parameter m=4 . . . . .	29
5.1	Prophet Default Mode Heatmap Jahr . . . . .	30
5.2	SMARD- Daten verbraucht gemessener Daten aus dem Jahr 2016: Heat- map Jahr . . . . .	31
6.1	Prophet Abweichung über eine Woche, alle Modelle . . . . .	33

# Tabellenverzeichnis

3.1	Genutzte Bibliotheken . . . . .	9
4.1	Vergleich der SARIMA Modelle . . . . .	28
4.2	Vergleich der SARIMA Modelle, bei geringeren historischen Daten . . . . .	29
6.1	Vergleich aller Modelle, bei einer Prognose von zwei Stunden . . . . .	32
6.2	Vergleich der Prophet Modelle und SMARD, bei einer Prognose von einer Woche . . . . .	33
A.1	Verwendete Hilfsmittel und Werkzeuge . . . . .	40



# Abkürzungsverzeichnis

**IDE** Integrierte Entwicklungsumgebung

**SARIMA** Seasonal, Autoregressive, Integrated, Moving Average

**SMARD** Strommarktdaten für Deutschland

**MAPE** Mean absolute percentage error

# 1 Einleitung

Angesichts der sich stetig verändernden Umwelteinflüsse, sind viele Unternehmen danach bestrebt, ihre Produktion und deren Energieverbrauch kontinuierlich zu optimieren. Hierfür gibt es verschiedene Ankerpunkte, an denen man ansetzen kann. Die vorliegende Arbeit beschäftigt sich indirekt mit der Kostenreduzierung beim Kauf von Strom. Gewerbekunden müssen beim Kauf des Stroms den Verbrauch pro 15 min angeben.

Sollten sie den zuvor angegebenen Verbrauch für den genannten Zeitraum überschreiten, kommen besondere Kosten auf das Unternehmen zu. Um den genauen Strombedarf von Unternehmen prognostizieren zu können, existieren verschiedenste Verfahren. Ein mögliches Verfahren ist die Entwicklung eines Programms, welches den Anwender mittels maschinellen Lernens unterstützt, Prognosen auf Basis von Trainingsdaten und angepassten Parametern zu erstellen.

Maschinelles Lernen ist ein Teilgebiet der Künstlichen Intelligenz und befasst sich damit, Modelle mittels statistischer Verfahren und Algorithmen zu trainieren [11][13]. Ein Bereich des maschinellen Lernens beschäftigt sich mit der Vorhersage von Ereignissen auf Basis historischer Daten. Dieser Bereich ist Gegenstand dieser Arbeit. Die Prognose des Energieverbrauchs eines Unternehmens ist Ziel des Forschungsprojekts Ecocharge. Dieses beschäftigt sich mit der Senkung der Energiekosten von Einzelhandelsvertretungen durch das Ausnutzen von Lastverschiebung beim Ladevorgang von Elektrofahrzeugen. Es ist von Interesse, auf Basis von Deep-Learning-Verfahren möglichst genaue Prognosen über den Stromverbrauch zu bestimmten Zeiten zu erhalten. Daher beschäftigt sich diese Arbeit mit der Analyse von Modellen zur Prognose von Zeitreihen mittels Verfahren des maschinellen Lernens.

Ebenfalls werden in der vorliegenden Arbeit zwei Modelle zur Prognose von Zeitreihen analysiert. Beim ersten Modell handelt es sich um Prophet, welches von einem Data Science Team von Facebook im Jahr 2017 veröffentlicht wurde. Prophet ist eine Open-Source-Software, die mit ihrer besonderen Robustheit in Bezug auf fehlende Daten sowie mit saisonalen Events wirbt. Das zweite Modell, welches in dieser Arbeit analysiert wurde, ist SARIMA.

Bei SARIMA handelt es sich um ein erweitertes Modell des ARIMA Modells. SARIMA ist ein besonders benutzerfreundliches Modell, da es nur sieben Parameter zum Prognostizieren benutzt. Des Weiteren die am häufigsten genutzten Modelle zur Prognose von Zeitreihen [7]. Sie wurden von George E.P. Box und Gwilym M. Jenkins in den 1970er-Jahren entwickelt[4].

Die zuvor genannten Modelle werden anhand verschiedener Prognose-Zeiträume und Parametrierung sowie mit den historischen Daten miteinander verglichen. Die historischen Daten wurden von SMARD zur Verfügung gestellt. Bei SMARD handelt es sich um eine Informationsplattform der Bundesnetzagentur für den deutschen Strommarkt [6].

Das Ziel dieser Arbeit war zu beurteilen, inwiefern sich Prophet und SARIMA dafür eignen, Prognosen in Bezug auf den Energieverbrauch zu nutzen.

## 2 Grundlagen

In diesem Kapitel werden die genutzten Grundlagen und Methoden für diese Arbeit dargestellt. Dabei wird auf die Prognoseverfahren Prophet und SARIMA, sowie auf die Informationsplattform SMARD detaillierter eingegangen.

### 2.1 Allgemeine Definitionen

- **Zeitreihenvorhersage:**  
Zeitreihenvorhersagen beziehen sich auf die Vorhersage zukünftiger Werte, basierend auf historischen Daten. Diese Technik wird häufig in Bereichen wie Finanzen, Meteorologie, Wirtschaft und Ingenieurwissenschaften eingesetzt, um Trends zu identifizieren und zukünftige Entwicklungen vorherzusagen[11].
- **Maschinelles Lernen:**  
Maschinelles Lernen ist ein Bereich der künstlichen Intelligenz, der Algorithmen entwickelt, die es Computern ermöglichen, aus Daten zu lernen und Vorhersagen zu treffen, ohne explizit für den jeweiligen Anwendungsfall trainiert zu werden[11].
- **Statistische Verfahren:**  
Bei statistischen Verfahren handelt es sich um Verfahren, die verwendet werden, um Daten zu analysieren, Muster zu identifizieren und anhand dieser Schlussfolgerungen zu ziehen. Diese Verfahren beruhen auf mathematischen Modellen[13].
- **Neuronale Netze:**  
Neuronale Netze sind eine Klasse von Algorithmen, die von der Struktur und Funktionsweise des menschlichen Gehirns inspiriert sind. Sie bestehen aus künstlichen Neuronen, welche in Schichten angeordnet und miteinander verbunden sind. Diese Netze können komplexe Muster in Daten erkennen und lernen, indem sie Gewichte anpassen, um Fehler zu minimieren[11].

## 2.2 Grundlagen: Prophet

Bei Prophet handelt es sich um ein Open-Source-Verfahren zur Vorhersage von Zeitdaten. Wie bereits erwähnt, wurde es vom Data Science Team von Facebook im Jahre 2017 entwickelt. Es weist eine Besonderheit auf, in Bezug auf die Robustheit von Daten sowie auf Trends mit starken saisonalen Effekten. Prophet existiert als Implementierung für die Programmiersprachen Python und R. Bei Prophet handelt es sich um ein additives Modell, welches am besten mit mehreren Saisons an historischen Daten arbeitet.

Dabei arbeitet Prophet mit der folgenden Formel:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (2.1)$$

Bei der Formel handelt es sich um eine Zusammensetzung aus dem Zeitreihenmodell von Harvey & Peters aus dem Jahr 1990 und den drei Hauptkomponenten des Prophet-Modells[20] [15]. Dabei handelt es sich bei  $g(t)$  um die Trendfunktion der Zeitreihe, welche die nicht periodischen Änderungen der Werte moduliert. Bei  $s(t)$  handelt es sich um die periodische Änderung der Saisonalitäten und  $h(t)$  steht für die Auswirkung von Feiertagen.

Der Fehlerterm  $\epsilon_t$  steht für alle idiosynkratischen Veränderungen, die nicht mithilfe des Modells erfasst werden können. Die zuvor erwähnte Stärke von Prophet ist die Einbindung von besonderen Events. Besondere Events können einen sportlichen Ursprung haben, aber auch einen gesetzlichen Ursprung haben, wie z.B. gesetzliche Feiertage.

Dafür können separate Dataframes erstellt werden, die die jeweiligen besonderen Events berücksichtigen. Ein weiterer Vorteil ist beispielsweise die Benutzerfreundlichkeit. Wegen seiner Funktion, automatisch saisonale Muster zu erkennen, ist Prophet auch für Anfänger geeignet. Diese Muster können mittels Fouriere-Reihen und weiteren Funktionen näher spezifiziert werden. Durch diese Erweiterung der Funktionen und Dataframes, kann das Modell präzisere Prognosen treffen. Außerdem hat Prophet eigene Darstellungen für die Prognosen, um die Interpretation der Daten zu vereinfachen. Der Ablauf der Benutzung von Prophet ist in der Grafik 2.1 vereinfacht dargestellt.

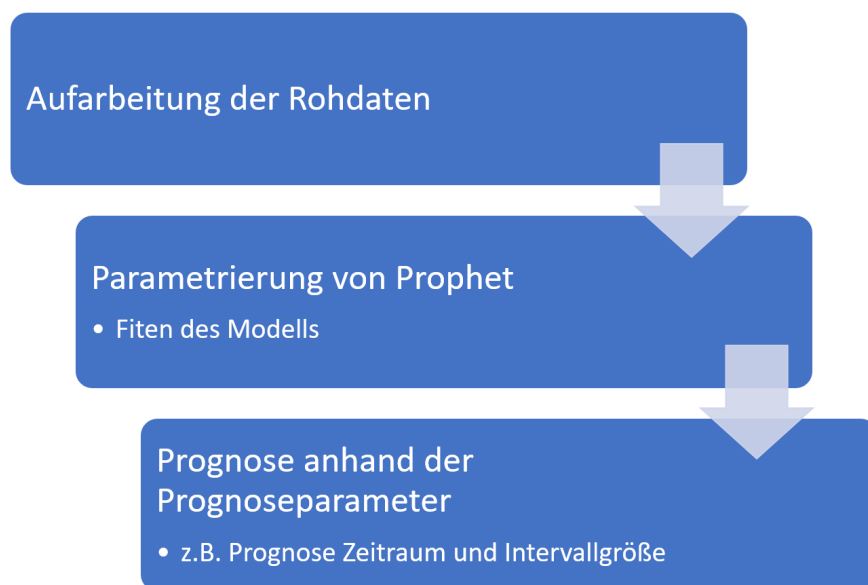


Abbildung 2.1: Ablauf Grafik Prophet

## 2.3 Grundlagen: SMARD

Die gespeicherten Daten der Informationsplattform lassen sich in vier Bereiche unterteilen: Stromerzeugung, Stromverbrauch, Markt und Systemstabilität. Die SMARD Daten haben in dieser Arbeit einen besonderen Wert, da diese die Jahre 2015-2023 umfassen und als Trainings- sowie Validierungsdaten dienen. Mithilfe der SMARD Daten wird eine Aussage über die Genauigkeit von Prophet und SARIMA getroffen. SMARD bietet auch Day-Ahead Prognosen an, welche basierend auf ökologischen und meteorologischen Daten den Stromverbrauch des nächsten Tages um 10 Uhr (UTC/GMT +1) veröffentlichen. Diese Prognosen weisen auch einen 15-minütigen Zyklus auf. Bei den betrachteten Daten handelt es sich dabei um die Daten der gesamten Netzlast. Die gemessenen Daten können in einer Taktfrequenz von 15-minütigen Blöcken abgefragt werden. Hierdurch wird für den Anwendungsfall der Industriekunden ein vereinfachter Vergleich ermöglicht. Bei der Abbildung 2.2 handelt es sich um eine Darstellung, welche SMARD zur Verfügung stellt [6]. Zusätzlich gibt es auch die Möglichkeit, die Daten als CSV-Dateien herunterzuladen. Bei den unterschiedlichen Farben handelt es sich um die Arten der Stromerzeugung. Die dünne orange Linie visualisiert die gesamte Netzlast.

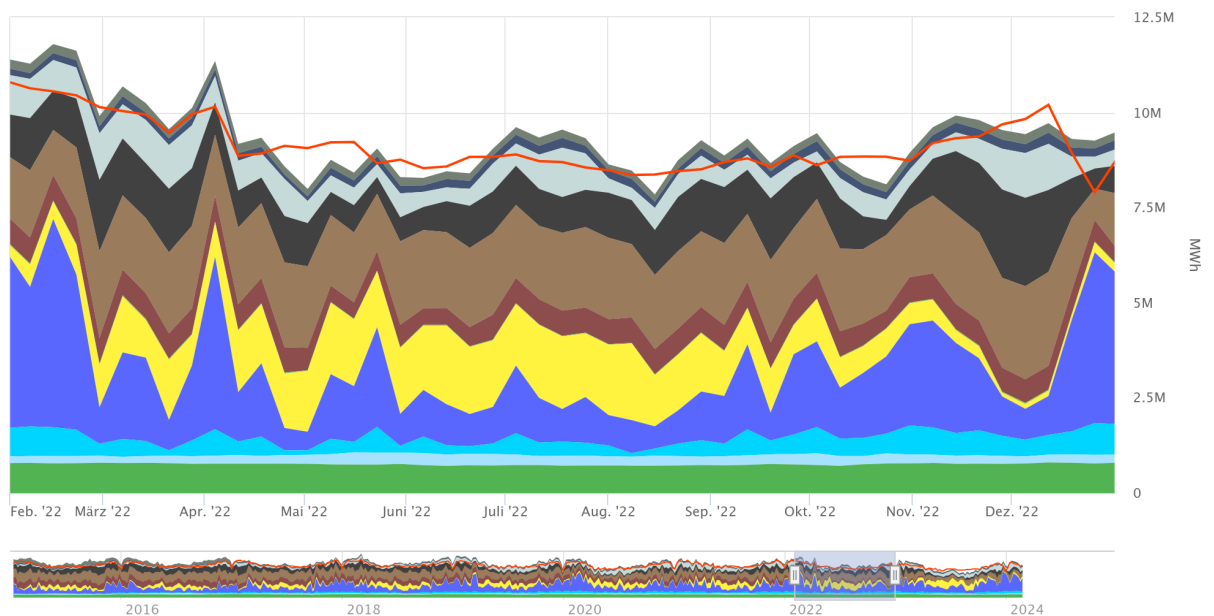


Abbildung 2.2: SMARD realisierte Erzeugung 2022

## 2.4 Grundlagen: SARIMA

SARIMA ist ein erweitertes Modell des ARIMA Modells. ARIMA steht für Autoregressive Integrated Moving Average. Dabei handelt es sich um die drei Hauptelemente dieses Modells. Diese drei Hauptelemente stehen für:

- Autogressive (AR) Komponente: Diese bezieht sich auf die Abhängigkeit der aktuellen Beobachtung von vorherigen Beobachtungen mit einem linearen Zusammenhang
- Integrierte (I) Komponente: Diese Komponente bezieht sich auf die Integration, um die Rohdaten stationär zu machen
- Moving Average (MA) Komponente: Dies bezieht sich auf den Einsatz eines gleitenden Durchschnitts von vorherigen Störungen, um das Muster der Residuen zu modellieren.

Das ARIMA Modell ist durch die Formel 2.2 zu beschreiben.

$$y_t = \alpha + \theta_{1ydt-1} + \theta_{pydt-p} + \theta_{1ct-1} + \theta_{qct-q} + \epsilon_t \quad (2.2)$$

Das ARIMA Modell wird durch einen weiteren Parameter, welcher die Saisonalität berücksichtigt, erweitert. Daraus ergibt sich die Formel 2.3 [15] [11]

$$SARIMA = (p, d, q)x(P, D, Q)m \quad (2.3)$$

Dabei handelt es sich bei den Parametern  $p, d, q$  um dieselben Parameter wie auch zuvor bei dem ARIMA Modell. Die weiteren Parameter  $P, D, Q$  beziehen sich auf die Saisonalität des SARIMA-Modells. Der Parameter  $m$  steht für die Anzahl, wie oft eine Beobachtung pro Monat auftaucht.



## 3 Materialien

In diesem Kapitel werden die genutzten Materialien dieser Arbeit vorgestellt. Als Entwicklungsumgebung wurde die IDE Visual Studio Code ausgewählt, da diese in dem Forschungsprojekt Ecocharge vertreten ist. In Visual Studio Code wurde die Erweiterung Jupyter Notebooks genutzt, die es ermöglicht Programmabschnitte, in Notizen und Code Bereiche zu unterteilen. Durch diese Unterteilung ist es möglich, Beschreibungen für explizite Abschnitte im Code detaillierter zu beschreiben. Ein Vorteil dieser Erweiterung ist, dass die Einarbeitung von Personen in die Arbeit erleichtert wird.

### 3.1 Implementierung

Als Programmiersprache wird Python genutzt, dabei handelt es sich um eine Programmiersprache, die im Bereich des Data Science weit vertreten ist[3]. Zudem wird diese Programmiersprache auch in anderen Bereichen des Forschungsprojekts genutzt.

Prophet besitzt, wie zuvor in dem Kapitel 2 erwähnt, eine Implementierung in Python. Damit Prophet in Visual Studio Code genutzt werden konnte, wurde außerdem die Distribution Anaconda installiert. Diese Distribution ermöglicht es, virtuelle Umgebungen zu erschaffen. Diese Umgebungen haben den Vorteil, dass man in verschiedenen Umgebungen verschiedene Versionen von Bibliotheken haben kann. Hierdurch wird ein leichter Wechsel zwischen verschiedenen Versionen und Projekten, an denen gleichzeitig gearbeitet wird, ermöglicht.

Die relevantesten genutzten Bibliotheken wurden in der Tabelle 3.1 hinterlegt[22].

Tabelle 3.1: Genutzte Bibliotheken

Name	Version
Prophet	1.0.1
Python	3.7.0
pandas	1.3.5
numpy	1.21.5

Im Kapitel 2 wurden die SMARD Daten und deren Bestandteile erwähnt. Es wurden die realisierten Stromverbräuche und die von Prophet prognostizierten Verläufe heruntergeladen und als CSV-Dateien gespeichert. Die realisierten Verbräuche werden zur Validierung der Prognosen genutzt.

## 3.2 Aufarbeitung der SMARD Daten

In diesem Unterkapitel wurden die SMARD Daten des realisierten Stromverbrauchs der letzten Jahre analysiert, um Zusammenhänge in den Messdaten zu erkennen. Zunächst wurde eine Fourier-Analyse der SMARD Daten durchgeführt, welche in der Grafik 3.1 dargestellt worden ist. Bei den genutzten Daten handelt es sich um den Zeitraum Januar 2020 bis Dezember 2022.

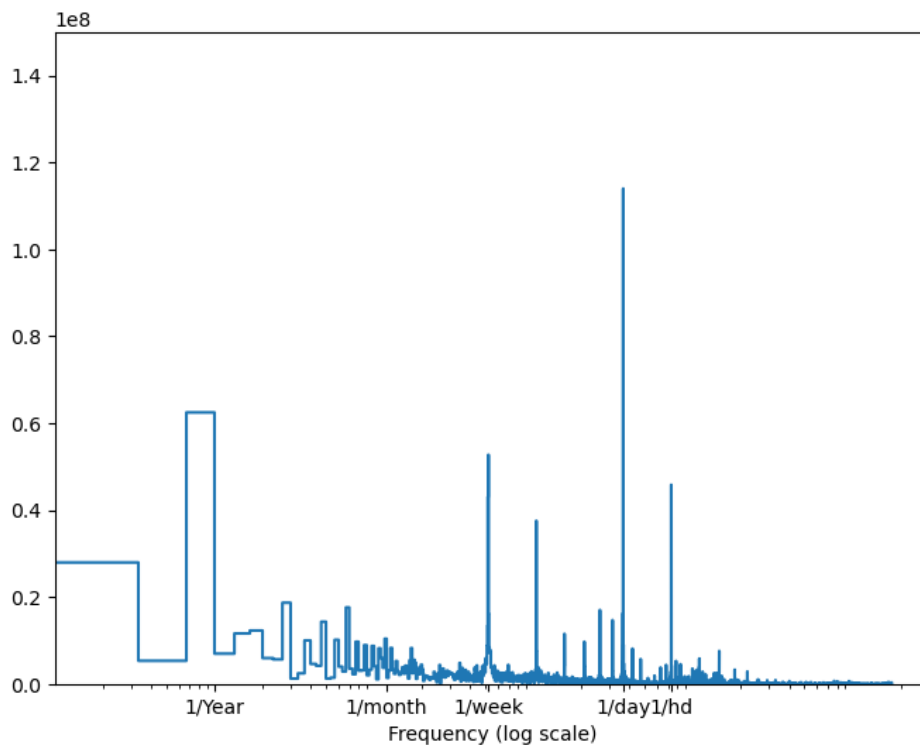


Abbildung 3.1: Fourier-Analyse der SMARD-Daten 2020/01-2022/12

Der Grafik 3.1 ist der Einfluss der einzelnen Tage, Wochen, Monate und des Jahres auf die Saisonalität zu entnehmen.

Diese Zusammenhänge werden in den nächsten Kapiteln für die Kalibrierung der Modelle genutzt. Außerdem ist zu erkennen, dass weitere Saisonalitäten existieren, welche nicht eindeutig einem täglichen, wöchentlichen, monatlichen oder jährlichen Verlauf zuzuordnen sind.

Zur Darstellung des jährlichen Verbrauches wurde eine Heat Map erstellt (vgl. Abbildung 3.2). Bei der Heat Map handelt es sich um eine Veranschaulichung des gemittelten täglichen Stromverbrauchs. Der Zeitraum erstreckt sich von Januar 2015 bis zum Dezember 2022.

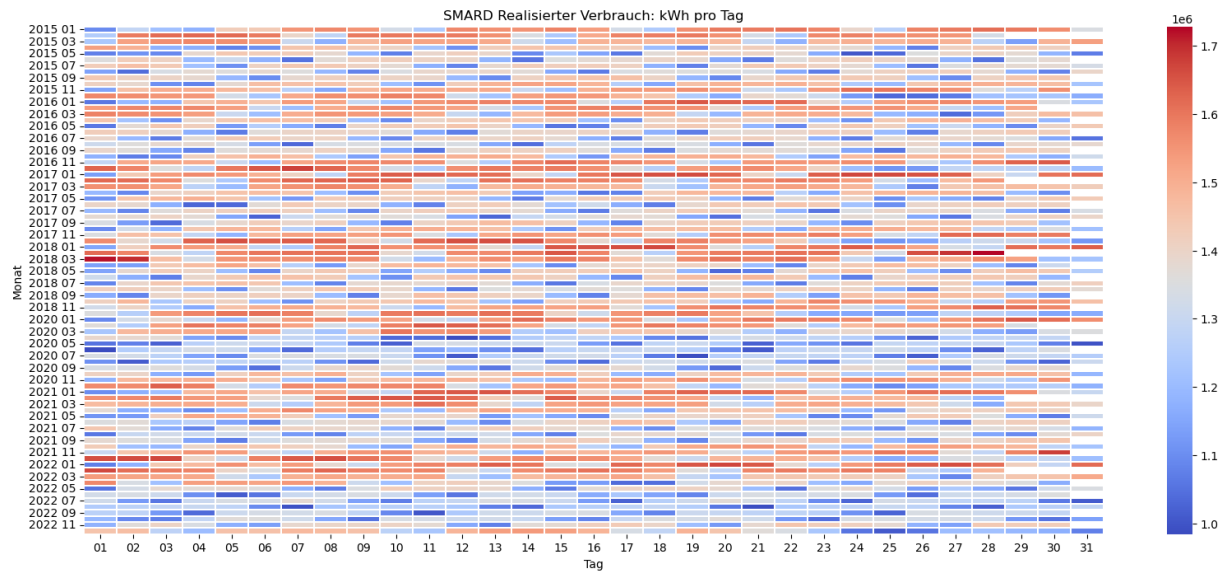


Abbildung 3.2: SMARD realisierter Verbrauch

Anhand der Abbildung 3.2 wurden ein Überblick über den jährlichen Stromverbrauch geschaffen.

Des Weiteren ist hier zu erkennen, dass ein jährlicher Zyklus sowie wöchentlicher Zyklus existiert. Dies entspricht dem Ergebnis der Fourier-Analyse der Daten.

Die SMARD Daten geben auch über die Energieerzeugung von Pumpspeichern, Auskunft. Diese werden in dieser Arbeit nicht genutzt und daher aus den Daten entfernt.

# 4 Simulation

## 4.1 Prophet

Für die Erstellung eines Modells mithilfe von Prophet werden bestimmte Anforderungen an die Daten gestellt (vgl. Abb. 2.1). Hierfür wurde die zuvor im Kapitel 2 erwähnte Excel-Datei mit den Daten so formatiert, dass Prophet jene auch nutzen kann. Deshalb arbeitet Prophet mit zwei Eingangsdaten. Zunächst mit den zeitlichen Werten. Hierfür existiert die Spalte ds, in der die zeitlichen Daten gespeichert sind. In der Spalte y werden die Messdaten gespeichert.

Prophet ist besonders gut für additive Modelle, da es auch die Standard-Einstellung besitzt [9]. Für die Prognose von Energiedaten funktionieren additive Modelle jedoch weniger gut [9], da es sich bei der Prognose von Energiedaten um multiplikative Modelle handelt. Daher wurde die Berechnung gezielt auf multiplikative Verfahren gestellt.

### 4.1.1 Prophet: unangepasstes Modell

In diesem Unterkapitel wurde ein Prophet Modell erstellt, welches keine weitere Anpassung der Parameter hat. Dabei handelt es sich um das Ausgangsmodell. Anhand dessen werden Optimierungen vorgenommen. In der Abbildung 4.1 sind die einzelnen Verläufe des Modells dargestellt. Dabei handelt es sich um eine Analyse der zur Verfügung gestellten Daten.

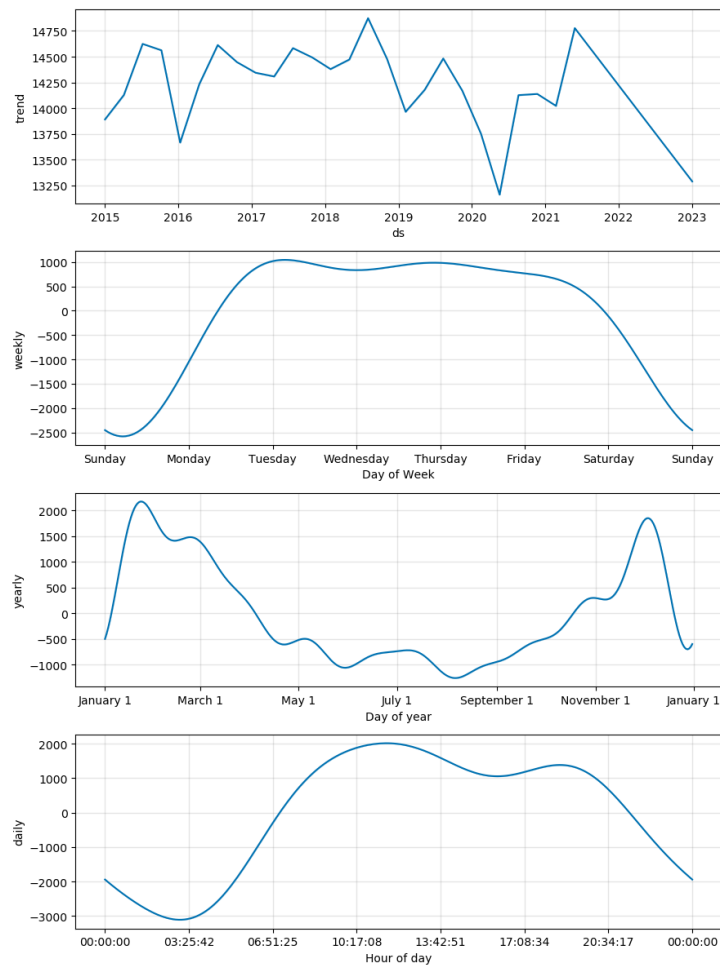


Abbildung 4.1: Prophet Default Komponenten

Es sind in der Abbildung 4.1 gleichmäßige Verläufe zu erkennen. Dieser resultiert aus der unangepassten Betrachtung von Prophet.

In der Abbildung 4.2 wurde eine Prognose mit den Trainingsjahren 2015-2022 angefertigt, um die erste Woche des Jahres 2022 zu prognostizieren. Bei der blauen Kurve handelt es sich um die Prognose von Prophet und bei der grünen um die realen Messdaten, die an jenen Tag gemessen worden sind. Außerdem wurde hier von Prophet standardmäßig ein additives Verfahren angenommen und kein multiplikatives Verfahren [9].

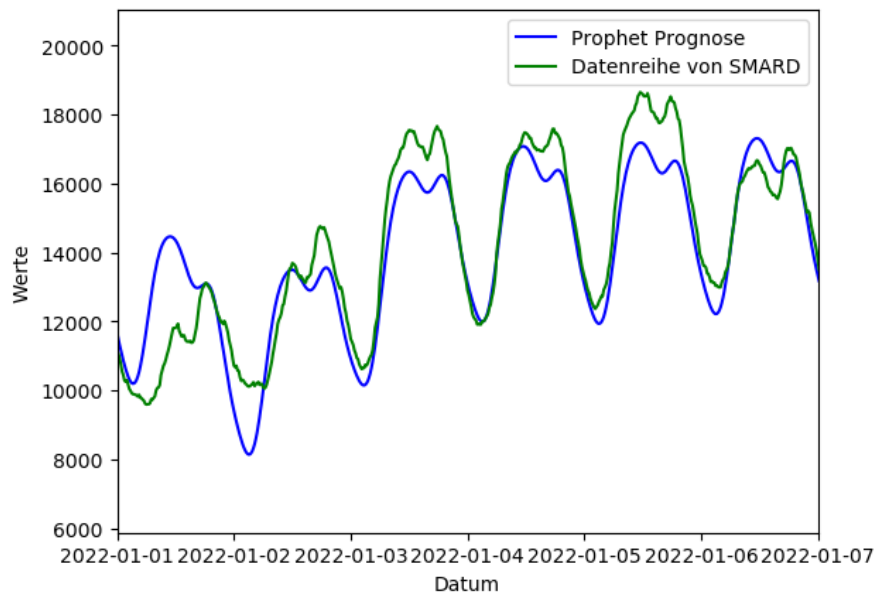


Abbildung 4.2: Prophet Default Werte, Prognose eine Woche

Anhand der Abbildung 4.2 ist zu erkennen, dass Prophet bei der Prognose einer Woche, meist einen geringeren Verbrauch prognostiziert, als tatsächlich verbraucht wurde. Hervorzuheben ist, dass Schwierigkeiten zu sehen sind, wenn das Modell den Jahreswechsel prognostizieren soll.

Für eine Prognose für die nächsten zwei Stunden wurde die Abbildung 4.3 erstellt.

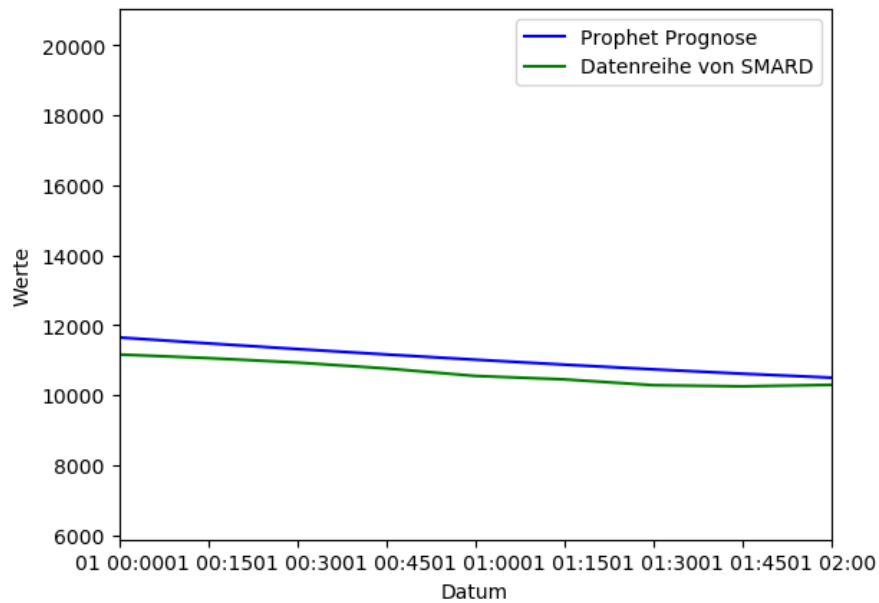


Abbildung 4.3: Prophet Default Werte, Prognose 2 h

Anhand der Abbildung 4.3 ist zu entnehmen, dass für einen Zeitraum von 2 Stunden eine gleichmäßige Abweichung existiert. Die Prognose des nicht angepassten Prophet-Modells, prognostiziert einen höheren, als den tatsächlichen Verbrauch.



Prophet besitzt eine Funktion namens “cross\_validation“. Diese Kreuzvalidierungsfunktion ermöglicht es, Prognosefehler mithilfe von historischen Daten zu messen[9]. Dies geschieht durch die Auswahl von Abschnittspunkten in der Historie. Für jeden von ihnen wird das Modell nur mit Daten bis zu diesem Abschnittspunkt angepasst. Anschließend können wir die prognostizierten Werte mit den tatsächlichen Werten vergleichen.

Dafür wurde der Funktion ein Trainingszeitraum von zwei Jahren und eine Prognose von zwei Stunden gegeben. Anhand dieser Daten wurden von Prophet verschiedene Statistiken zur Vorhersageleistung bestimmt. Unter anderem kann der MAPE bestimmt werden [8]. Dieser wurde für eine Prognose von zwei Stunden in der Abbildung 4.4 dargestellt.

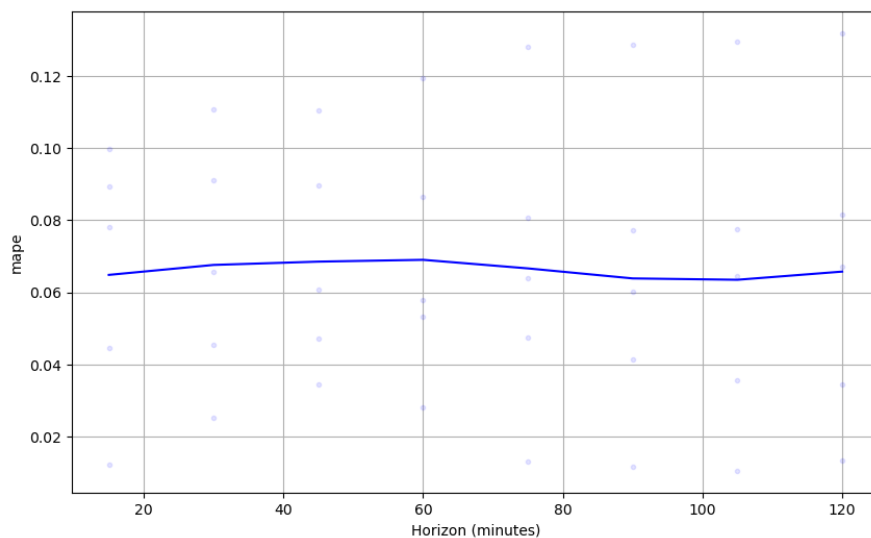


Abbildung 4.4: Prophet MAPE unangepasst für zwei Stunden

Zu erkennen ist, dass dieses Modell einen durchschnittlichen MAPE von 6-7% aufweist.

In der Abbildung 4.5 wurde eine Kreuzvalidierung für einen Zeitraum von 180 Tagen bestimmt. Bei der Abbildung 4.5 wird sichtbar, dass es sich um zwei Plots in einem

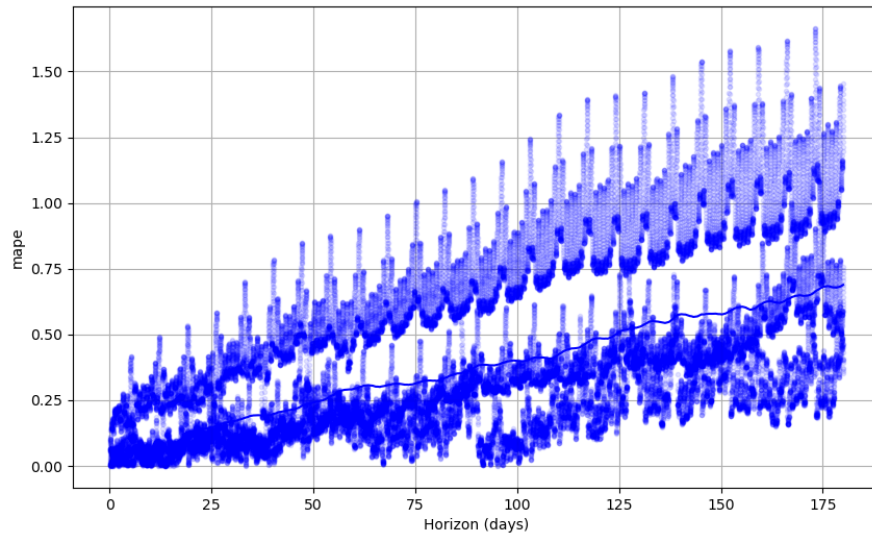


Abbildung 4.5: Prophet MAPE unangepasst

handelt. Aus der Abbildung wurde ersichtlich, dass bei der Nutzung von der Funktion “cross\_validation“ darauf geachtet werden muss, welcher Zeitraum betrachtet wird.

In der Abbildung 4.6 wurde der Fehler dargestellt, den Prophet im Vergleich zum tatsächlichen Verbrauch über den Zeitraum von einer Woche prognostiziert hatte. Die Abbildung

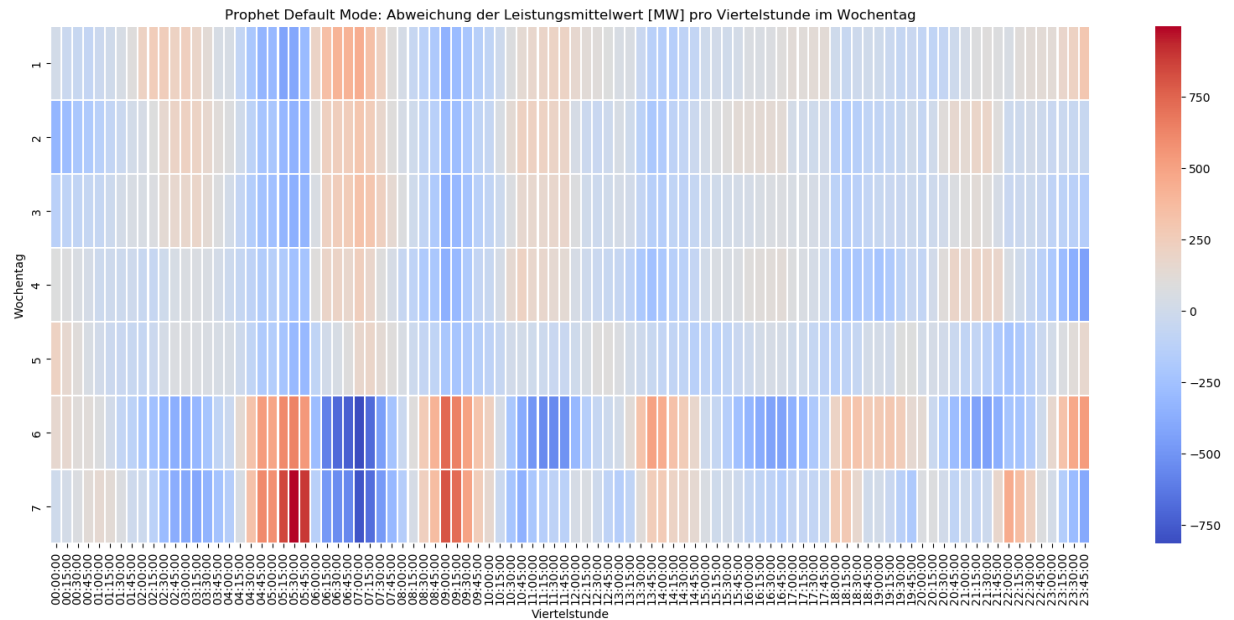


Abbildung 4.6: Prophet Fehler Darstellung

4.6 veranschaulicht, dass Prophet die größten Ungenauigkeiten bei der Prognose von den Morgenstunden der Samstags und Sonntags aufweist. Bei den anderen Wochentagen ist zu erkennen, dass Prophet meist eine geringe Abweichung besitzt, als bei den Wochenendtagen. Dies ist auch in der Abbildung 4.2 zu sehen, da es eine größere Abweichung zwischen der Prognose und des verbrauchten Stroms gibt. Bei den meisten Prognosenwerten von Prophet handelt es sich um Werte, die unterhalb des tatsächlichen Verbrauchs lagen.

### 4.1.2 Prophet: angepasstes Modell

In diesem Kapitel wurde das Modell mit den besten Parametern dargestellt, um einen geringen MAPE-Wert zu erzielen. Für das Bestimmen des nachfolgenden Modells wurden die folgenden Parameter angepasst:

- interval width
- daily seasonality
- weekly seasonality
- yearly seasonality
- holidays
- holidays priorscale
- seasonality mode

Eine Anpassung dieser Parameter wirkt sich auf den MAPE aus. Der Wert der Parameter ergibt sich aus dem Rang der Fourier-Ordnung. Die Parameter von Prophet erkennen eigenständig die Saisonalitäten und gewichten diese in der Berechnung anhand deren Ordnung mit ein.

Für die Erstellung des Modells wurde ein neuer Dataframe erstellt, welcher alle gesetzlichen Feiertage sowie die Schulferien aller Bundesländer enthält erstellt, um diese in die Berechnung mit einzubringen. Die Daten für die gesetzlichen Feiertage und die Schulferien wurden die Daten von den Webseiten FerienWiki[10] und Schulferien.org[2] genommen und mit den Daten bekannter Feiertage verglichen[16]. Die erstellten Dataframes wurden abermals in die jeweiligen Bundesländer unterteilt. Durch solch eine Segmentierung ist es möglich, gezielt Bundesländer aus einer Prognose zu entfernen und hinzuzufügen.

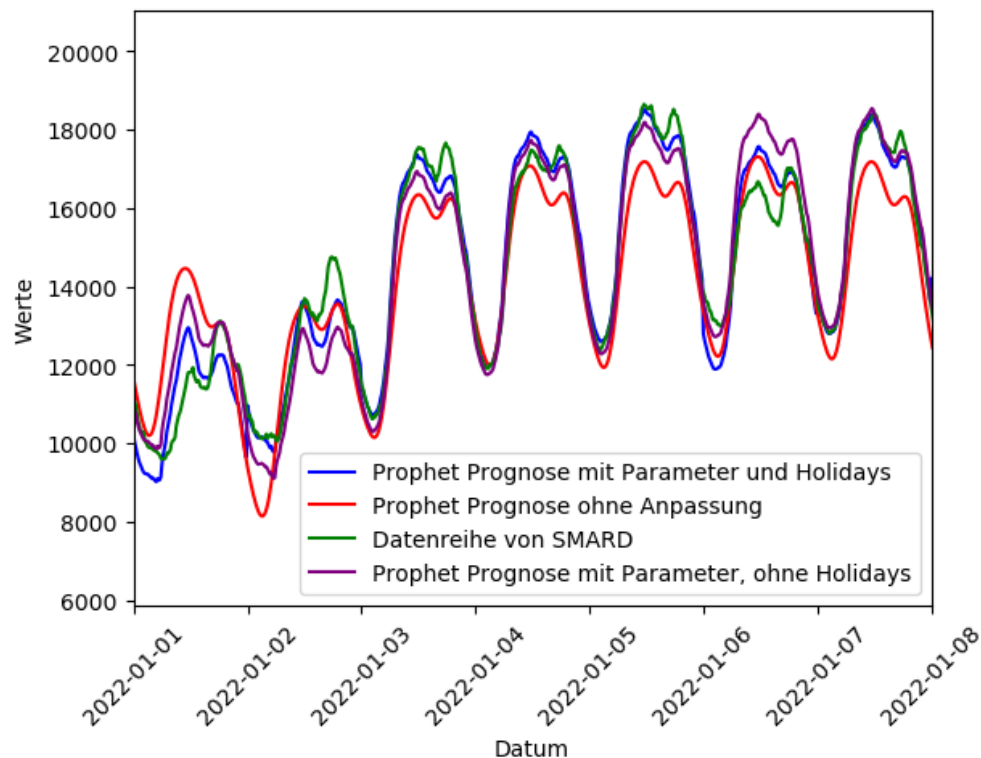


Abbildung 4.7: Prophet angepasstes Modell, Prognose eine Woche

Bei der Abbildung 4.7 handelt es sich um die Prognose einer Woche. In der Abbildung wurden zwei Prognosen angezeigt, wobei es sich bei beiden Prognosen um die gleichen Parameter handelt. Die Besonderheit liegt darin, dass es sich bei der blauen Prognose zusätzlich die erstellten Dataframes der Ferien- und Feiertage handelt.

In der Abbildung 4.8 handelt es sich um die Darstellung aller Faktoren, welche eine Auswirkung auf die Modellierung haben.

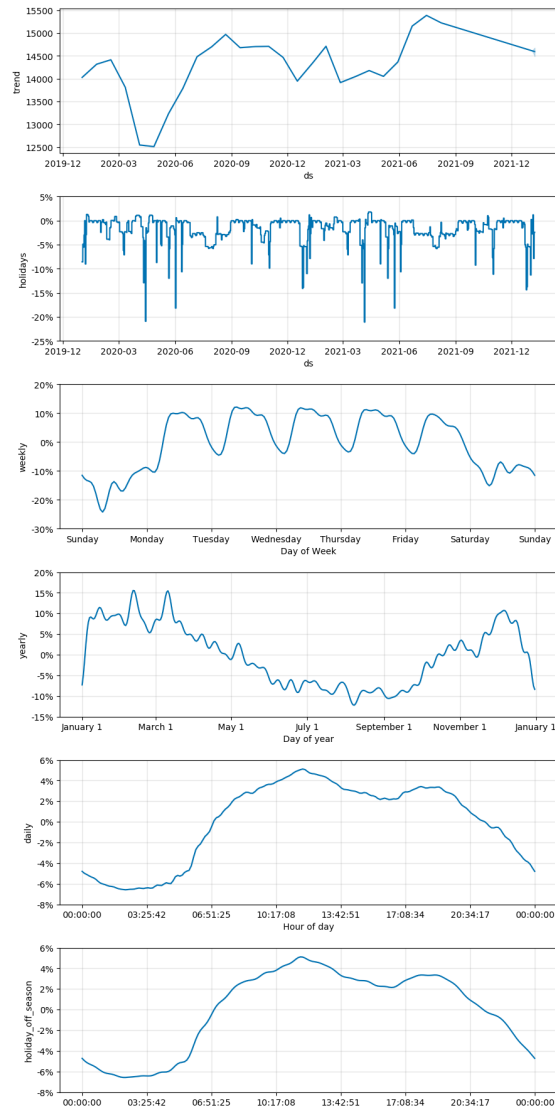


Abbildung 4.8: Prophet angepasstes Modell, Komponenten

Bei der Abbildung 4.8 ist zu erkennen, dass diese ungleichmäßiger verläuft, als bei der Abbildung 4.1. Der ungleichmäßige Verlauf resultiert aus der Einbeziehung der Parameter. Durch einen ungleichmäßigeren Verlauf ist zu erkennen, dass das Modell weitere Parameter in der Berechnung berücksichtigt hat.

Für das angepasste Modell wurde auch eine Kreuzvalidierung vorgenommen. Betrachtet wird auch hier der MAPE, für eine Prognose von zwei Stunden

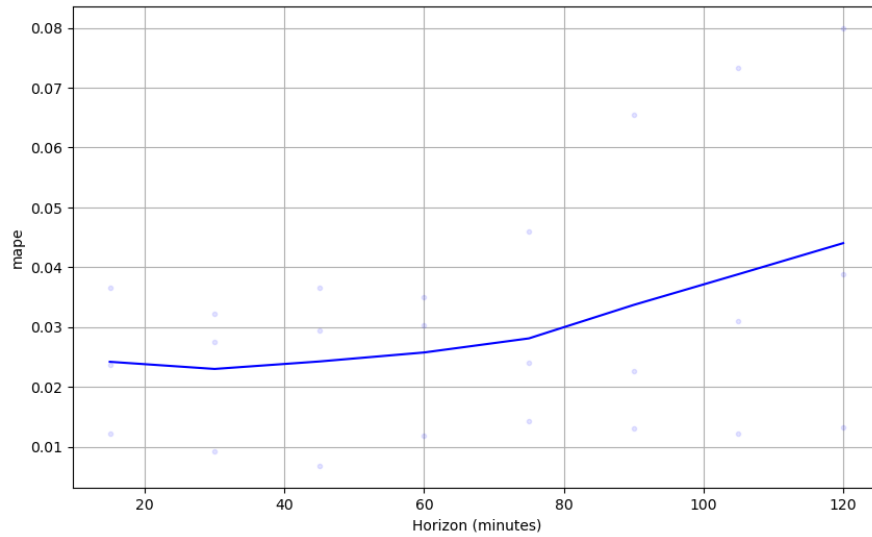


Abbildung 4.9: Prophet angepasstes Modell, MAPE zwei Stunden

Bei der Abbildung 4.9 ist zu erkennen, dass der anfängliche MAPE-Wert bei ca 2,5 % startet. Dies ist ca. 4 % besser als mit dem unangepassten Modell (vgl. Abb. 4.4). Des Weiteren ist zu erkennen, dass der MAPE-Wert bei steigender Zeit sich linear erhöht.

Außerdem wurde auch für dieses Modell die Kreuzvalidierung für einen Zeitraum von einem Jahr prognostiziert.

Anhand der Abbildung 4.5 und 4.10 ist zuerkennen, dass diese einen ähnelnden Verlauf haben. Dennoch verweilen die meisten Daten bei der Abbildung 4.10 im selben Bereich verweilen und die gesamte Abweichung weist konstante Züge auf.

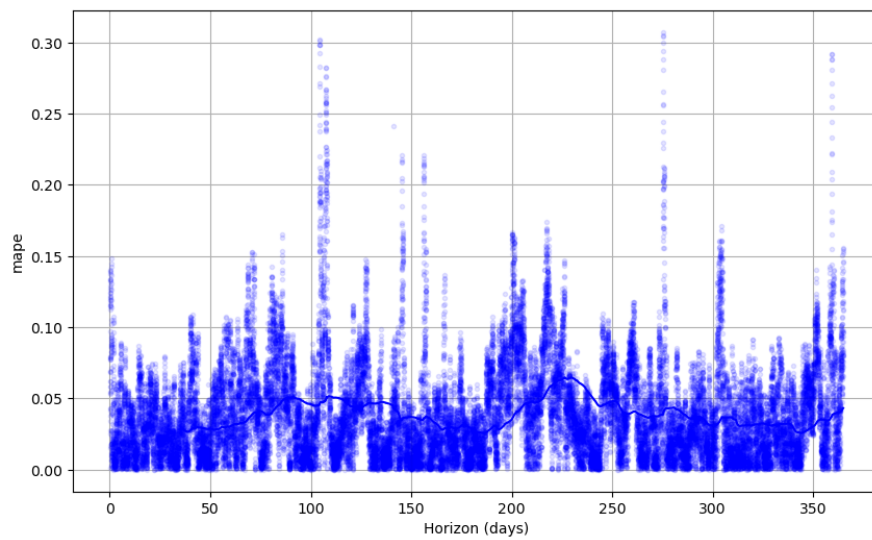


Abbildung 4.10: Prophet MAPE angepasst



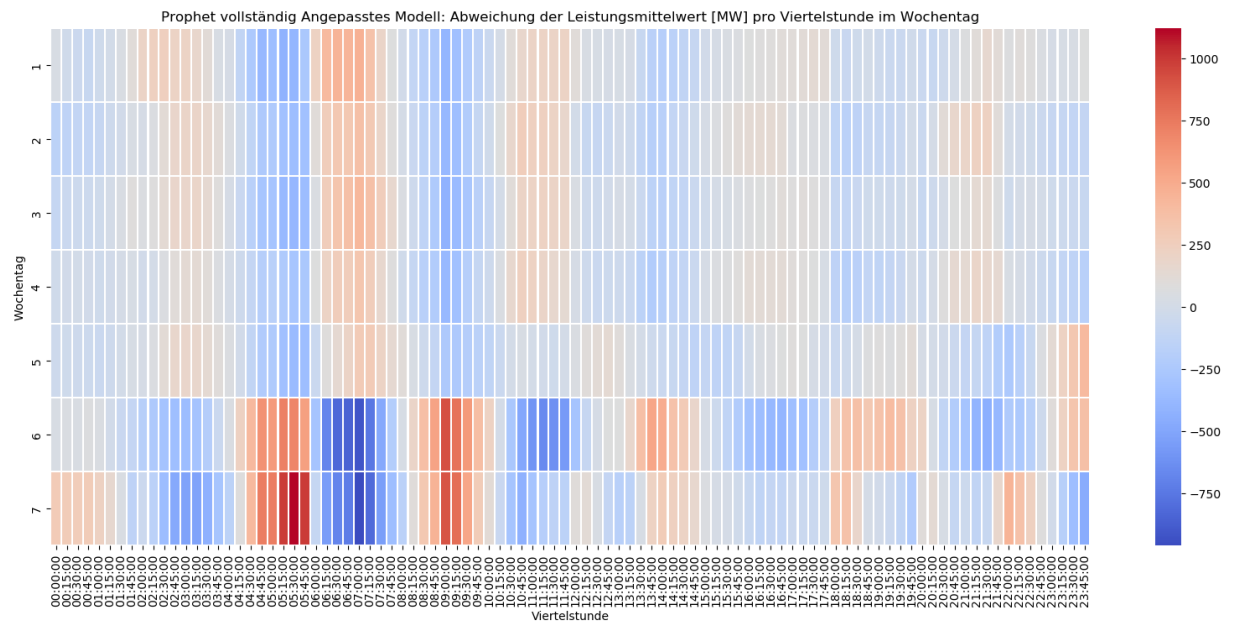


Abbildung 4.11: Prophet Fehler Darstellung vollständig angepasst

Bei der Abbildung 4.11 handelt es sich um die Darstellung des Fehlers in Bezug zu den gemessenen Daten. In einem Vergleich zwischen der Abbildung 4.11 und der Abbildung 4.6 ist zu sehen, dass das Prophet Modell “Prophet vollständig angepasste Parameter“ einen geringeren Fehler aufweist.

### 4.1.3 Prophet: weitere Modelle

Es gibt die Möglichkeit bei Prophet mittels MCMC Verfahren Prognosen durch Zuführen. Beim MCMC-Verfahren handelt es sich um das Markow-Chain-Monte-Carlo Verfahren [11] [19], welches die erwünschte Verteilung als ihre stationäre Verteilung aufweist.

Dieses Verfahren gehört zur bayesschen Statistik, dabei ist auch zu erwähnen, dass jenes Verfahren genauer wird, sobald man die Schritte erhöht. Zu beachten ist, dass eine Schrittweite von über 1000 eine bessere Genauigkeit erweist.

Ein Kritikpunkt an diesem Verfahren ist, dass es, aufgrund der Vielschrittigkeit, eine um vielfach höhere Berechnungszeit hat und bestimmte Funktionen von Prophet deaktiviert. Es wurden kleinere Modelle erstellt und berechnet, um einen Vergleich mit den anderen Prophet-Modellen herzustellen. Aber um vergleichbare Modelle zu erstellen, reichte die Rechenleistung der genutzten Systeme nicht aus. Die Trainingsdaten waren im Bezug auf ihre Saisonalität zu komplex, um mit kleineren Schrittweiten präzise Ergebnisse zu erhalten.

## 4.2 SARIMA

In Kapitel 2.4. wurden die Anforderungen genannt, um SARIMA korrekt zu bedienen. Bei der Bestimmung der Parameter existieren verschiedene Möglichkeiten. Ein weit verbreitetes Verfahren ist, dass man eine Autokorrelationsfunktion und eine partielle Autokorrelationsfunktion [7].

Die Autokorrelationsfunktion misst die Korrelation zwischen Datenpunkten in Zeitreihen bei verschiedenen Verzögerungslängen. Die partielle Autokorrelationsfunktion ähnelt der Autokorrelationsfunktion, jedoch berücksichtigt sie die Werte zwischen zwei Beobachtungspunkten.

Die zuvor genannten Funktionen wurden auf die Daten von SMARD angewandt. Mit Hilfe dieser können die Parameter bestimmt werden, welche SARIMA für eine genaue Prognose benötigt. Für die Arbeit mit diesen Funktionen wurden die Funktionen aus der Python Bibliothek "statsmodels" genutzt [12].

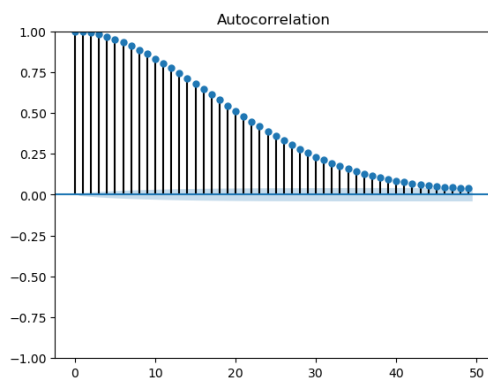


Abbildung 4.12: Autokorrelationsfunktion

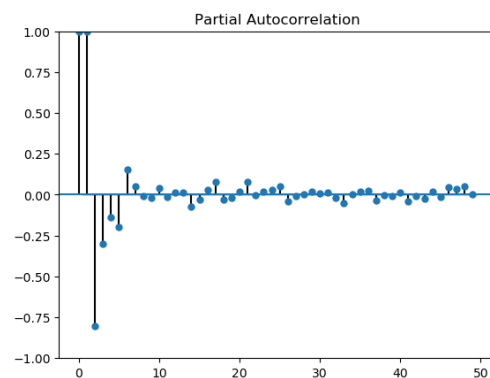


Abbildung 4.13: Partielle Autokorrelationsfunktion

Auf der x-Achse der beiden Abbildungen sind die sogenannten "Lags" vertreten. Jene spiegeln wider, wie viele Zeitschritte zwischen zwei Datenpunkten liegen. Anhand dieser können Rückschlüsse auf die Parameter fürs SARIMA Modell gezogen werden. Eine weitere Möglichkeit ist es, die Parameter mittels der Funktion "auto\_arima" bestimmen.

Dabei durchläuft die Funktion eine Kombination aus Werten für die Parameter, um die beste Kombination zu finden, sich den Testdaten zu nähern [1]. Hierzu werden lediglich die Saisonalität auf TRUE gesetzt und der Parameter  $m$  bestimmt, welcher die Periodizität der Zeitreihe in Tagen bezeichnet. Besitzt die Zeitreihe beispielsweise wöchentliche Daten, so würde man  $m = 7$  setzen und bei jährlichen Daten auf  $m = 365$ . Für die Nutzung des Parameters  $m$  ist zu beachten, dass dieser einen Integerwert erfordert. Dieser Parameter ist ein optionaler Parameter der Funktion "auto\_arma" und muss daher nicht zwingend angegeben werden, verkürzt aber die Berechnungszeit und präzisiert die Berechnungszeit darüber hinaus.

### 4.2.1 SARIMA Modelle

Für die Erstellung der Modelle wurde die zuvor erwähnte Funktion "auto\_arma" genutzt und dessen Funktion der Saisonalität  $a$  aktiviert. Allen nachfolgenden Modellen wurden die gleichen historischen Daten, von SMARD, zur Verfügung gestellt. Die Modelle haben dadurch mit den bekannten Daten von 2015 bis 2022 gearbeitet. Dabei haben die SARIMA Modelle mit 80% der Daten als Trainingsdaten gearbeitet und die letzten 20% der Daten wurden zur Validierung genutzt.

Zunächst wurde der optionale Parameter  $m$  nicht genutzt, um einen Vergleich innerhalb der einzelnen SARIMA Modelle zu ermöglichen.

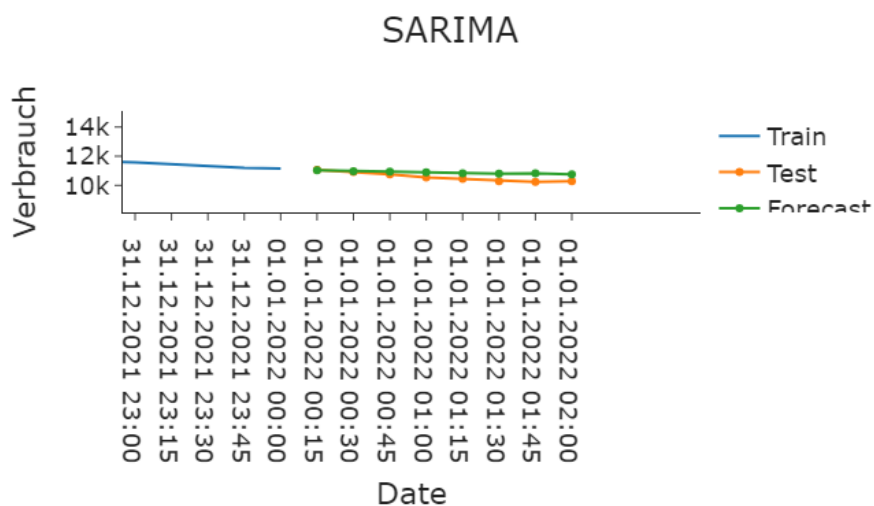


Abbildung 4.14: SARIMA ohne dem Parameter  $m$

Für den weiteren Vergleich wurden der zuvor erwähnte Parameter  $m$  mit in die Prognosen einbezogen.

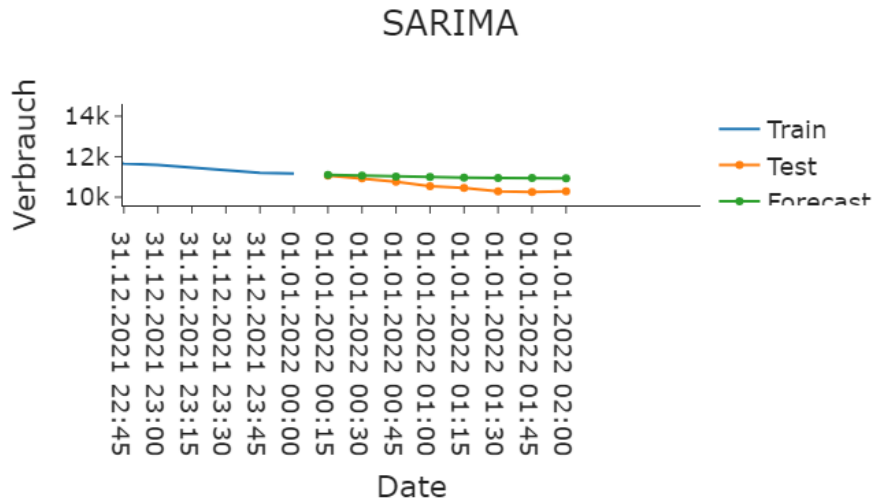
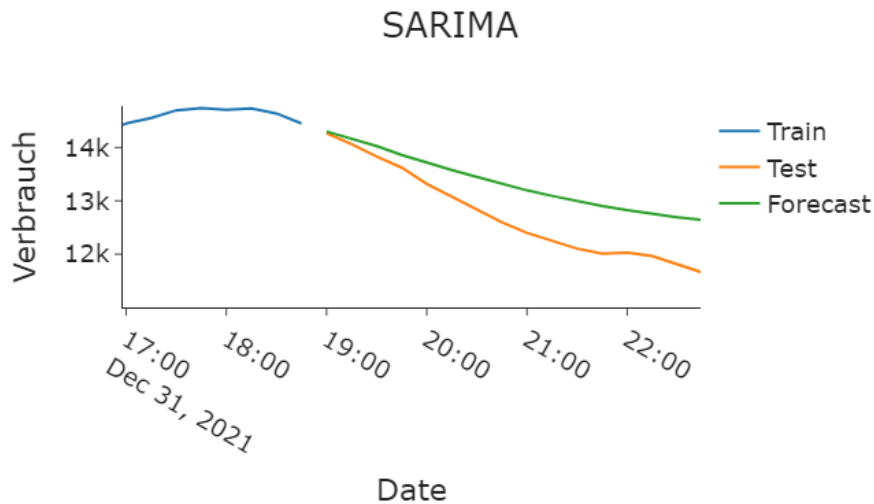


Abbildung 4.15: SARIMA mit dem Parameter  $m=2$

Eine Besonderheit, die bei der Simulation aufgefallen ist, ist folgende. Die Daten weisen eine große Komplexität von Saisonalitäten auf. Dies hat zur Folge, dass die Funktion “auto\_arma“ Werte im Bereich der Tausender als Parameter für  $p, d, q, P, D, Q$  Einsetzen möchte, um eine möglichst gute Annäherung zu den Trainingsdaten zu erhalten. Daraus resultierte, dass die für diese Arbeit zur Verfügung stehenden Hardwarekomponenten das Ausmaß der Daten nicht bearbeiten konnten. Anhand der Modelle 4.14 und 4.15 wurde die nachfolgende Tabelle 4.1 erstellt, welche im Kapitel 6 herangezogen wird, um eine Beurteilung über die Modelle zu treffen. Für eine Prognose mit den gleichen historischen Daten, welche auch Prophet nutzt, wurde der MAPE erstellt und in der Tabelle zusammengefasst.

Tabelle 4.1: Vergleich der SARIMA Modelle

Wert $m$	MAPE 2H
0	3,039
2	4,072

Abbildung 4.16: SARIMA mit dem Parameter  $m=4$ 

Bei der Abbildung 4.16 wurden zwei Stunden prognostiziert, mit einem geringeren Datensatz von nur zwei Jahren. Dies wurde getan, da die Funktion “auto\_arima“ nicht genügend Hardwarespeicher zur Verfügung hat um die Berechnung durchzuführen. Die Simulationen mit den geringeren Datensätzen wurde auch mit den vorherigen SARIMA-Modellen durchgeführt und daraus der MAPE für zwei Stunden erstellt. Der MAPE wurde in der Tabelle 4.1 dargestellt.

Tabelle 4.2: Vergleich der SARIMA Modelle, bei geringeren historischen Daten

Wert m	MAPE 2H
0	3,55
2	3,75
4	3,06

# 5 Ergebnisse

## 5.1 Kalibrierungsergebnisse

Zur Kalibrierung wurde die Grafik 3.1 herangezogen, um anhand dieser eine Bewertung der Gewichtung der einzelnen Saisonalitäten zu erstellen. Anhand der Grafik 3.1 ist zu erkennen, dass die SMARD-Daten eine hohe Saisonalität in Bezug auf eine tägliche Wiederholung aufweisen.

Prophet bietet die Funktion, die Saisonalitätsspezifische Fourier-Ordnung anzugeben. Anhand der Grafik 3.1 und der Prophet eignen Funktionen wurden dann Kalibrierungen vorgenommen, um die besten Parameter für die Saisonalitäten Tag, Monat und Jahr zu finden. Außerdem wurde der Saisonalitäts-Modus von Prophet auf multiplikativ gestellt, da dieser Modus für die Prognose von Energiedaten besser geeignet ist [9].

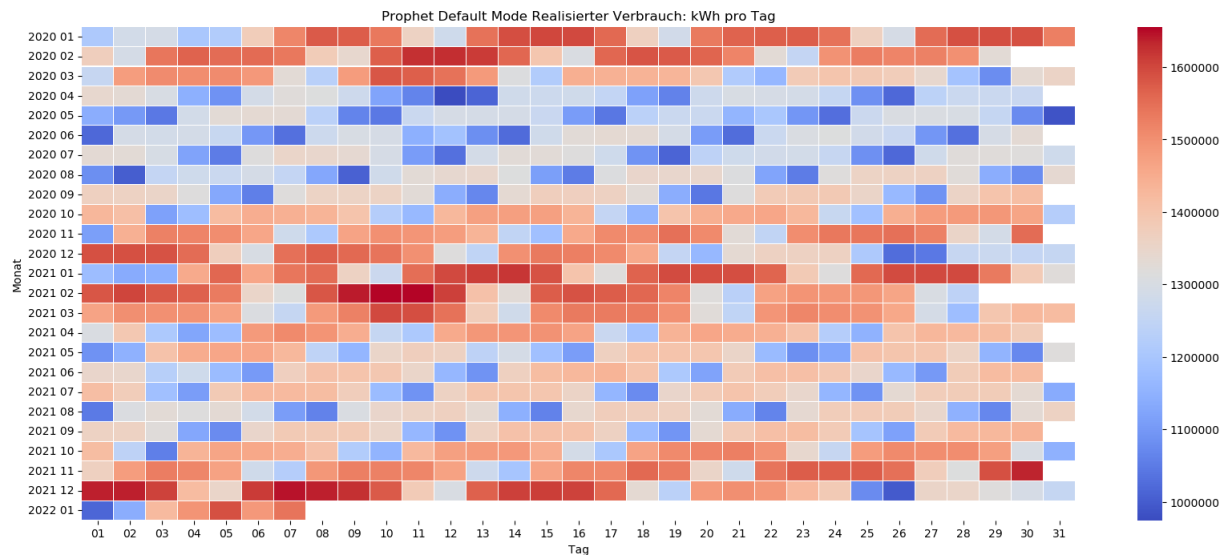


Abbildung 5.1: Prophet Default Mode Heatmap Jahr

Im Kapitel 4.1.2 wurde die Funktion “holidays“ erläutert, diese wurde bei dem nachfolgenden Modell ergänzt, hierfür wurden die Schulferien und gesetzlichen Feiertage der Jahre 2015-2022 in separate Dataframes hinterlegt. Diese Dataframes dienen Prophet zur weiteren Prognose. Dass diese Feiertage und Schulferien einen Einfluss auf den Energieverbrauch haben, lässt sich aus der Heatmap vom Jahr 2016 (siehe Abb.5.2) lesen. Als Beispiel kann man sehen, dass der Verbrauch über die Osterfeiertage im März geringer ist als an andern Tagen. Diese Besonderheit wurde im Modell mitberücksichtigt.

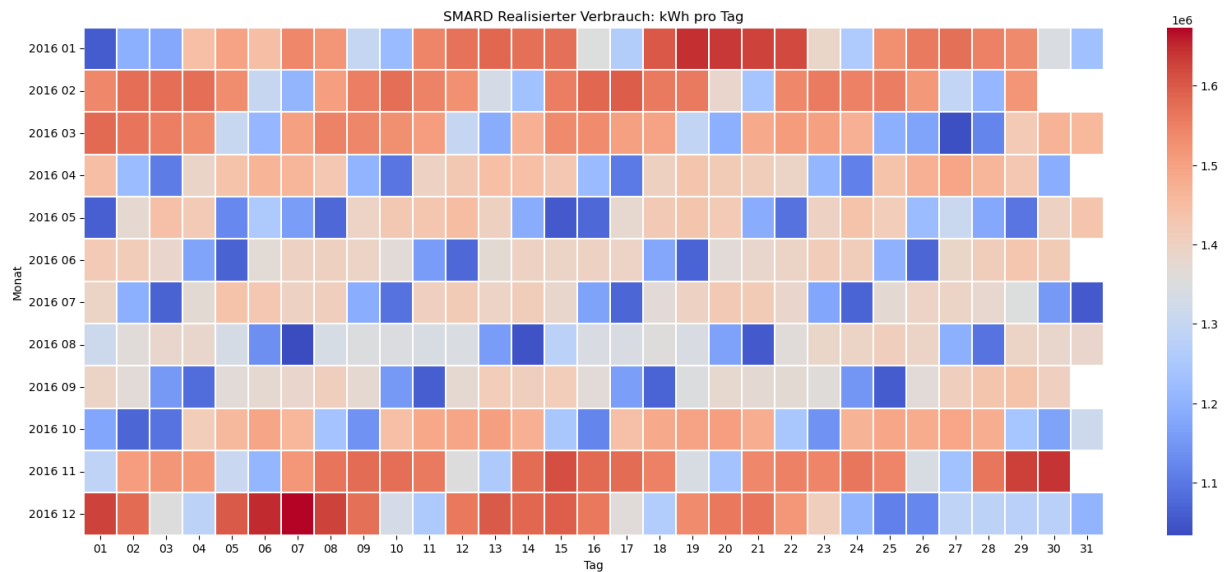


Abbildung 5.2: SMARD- Daten verbraucht gemessener Daten aus dem Jahr 2016: Heatmap Jahr

Des Weiteren wurde die Annahme getroffen, dass sich der Energieverbrauch von Sonn- und Feiertagen ähnlich verhält. Das wurde mit einer erhöhten Priorität in die Berechnung der Prognosen berücksichtigt. Die Annahme konnte bestätigt werden, da in der Abbildung 5.2 zu erkennen ist, wann Ostern oder vergleichbare Feiertage liegen. Diese weisen in der Heatmap eine vergleichbare blaue Färbung auf. Des Weiteren wurden noch weitere Saisonalitäten erstellt. Zunächst wurde ein Regressor angelegt, welcher bestimmt, ob es sich bei dem jeweiligen Tag um einen Feiertag handelt, dies wurde anschließend von Prophet bei der Modellierung berücksichtigt.



## 6 Beurteilung der Ergebnisse

In diesem Kapitel sollen die Ergebnisse aus den Kapiteln 4 und 5 zusammengefasst und beurteilt werden. Hierfür werden die Prognosen der Modelle Prophet, SARIMA und SMARD, miteinander verglichen. Dabei handelt es sich bei den Prognosen von SMARD um die Referenz.

Der Fokus der Beurteilung liegt darauf, wie die Modelle einen Zeitraum von zwei Stunden prognostizieren. Zusätzlich wurden mit Prophet eine Prognose von sieben Tagen erstellt, welche ebenfalls beurteilt wird. In Tabelle 6.1 werden die Prognosen aller Modelle für einen Zeitraum von zwei Stunden verglichen.

Tabelle 6.1: Vergleich aller Modelle, bei einer Prognose von zwei Stunden

<b>Modell</b>	<b>MAPE 2H</b>
<b>SARIMA ohne m</b>	3,04
<b>SARIMA m = 2</b>	4,07
<b>Prophet unangepasst</b>	5,03
<b>Prophet angepasste Parameter</b>	0,53
<b>Prophet vollständig angepasst</b>	8,08
<b>SMARD Prognose</b>	3,91

Anhand der Tabelle 6.1 ist zuerkennen, dass das Modell “Prophet angepasste Parameter“den besten MAPE- Wert liefert. Besonders zu erwähnen ist, dass es sich hierbei um die Prognose des 01.01.2022 handelt, und somit um das Neujahr.

Für den Vergleich der Prophet Modelle für die Prognose einer Woche wurde die Abbildung 6.1 erstellt. Anhand dieser ist zu erkennen, dass Prophet mit bei der Prognose des Neujahres zu höheren Abweichungen neigt. In den folgenden Tagen stabilisiert sich der MAPE-Wert.

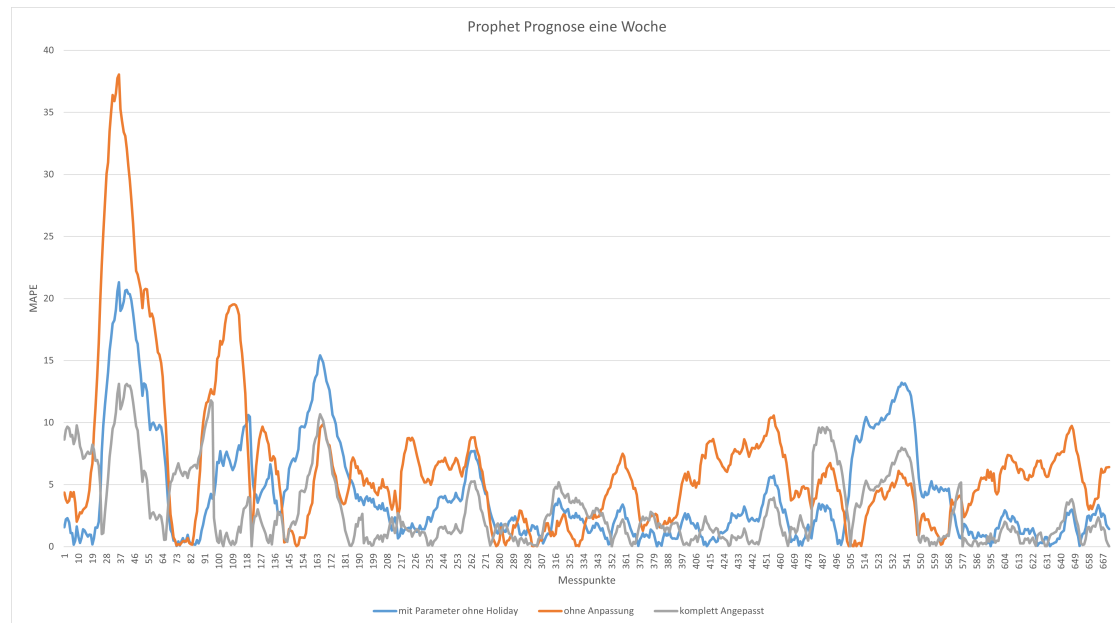


Abbildung 6.1: Prophet Abweichung über eine Woche, alle Modelle

Der MAPE-Wert für die Prognose einer Woche ist in der Tabelle 6.2 dargestellt. Die Prophet Prognose “Prophet vollständig angepasste Parameter“ konnte ein minimal besseren MAPE-Wert erzielen, als es die SMARD Prognose tut.

Tabelle 6.2: Vergleich der Prophet Modelle und SMARD, bei einer Prognose von einer Woche

Modell	MAPE eine Woche
Prophet unangepasst	6,39
Prophet angepasste Parameter	4,17
Prophet vollständig angepasst	2,97
SMARD Prognose	3,01

Nun wurden die Tabellen 6.1 und 6.2 herangezogen und verglichen. So fällt auf, dass das Prophet-Modell “Prophet vollständig angepasste Parameter“ , welches mit allen Parametern und der Holiday-Funktion arbeitet, besser ist für eine Prognose von größeren

Zeiträumen ist. Dies ist der Fall, da es sich bei dem Prognosezeitraum für die zwei Stunden Prognose, um die ersten zwei Stunden des Jahres 2022 handelte. In diesem Zeitraum haben die meisten Modelle nicht gut abgeschnitten und in Bezug auf diese Schwierigkeit wurden Anpassungen vorgenommen.

Außerdem ist das Prophet-Modell mit den angepassten Werten für eine Prognose einer Woche um 0,04 % besser als die Prognose von SMARD. Dabei muss darauf geachtet werden, dass SMARD jeden Tag um 10 Uhr eine neue Prognose für die kommenden 24 Stunden erstellt und keine Prognose für eine gesamte Woche. Daraus folgt, dass die SMARD Prognose die Kombination aus sieben 24-Stunden-Prognosen ist.

### 6.1 Vergleichbare Arbeiten und weitere Aussichten dieser Arbeit

Vergleichbare Arbeiten nutzen Prophet zur Bestimmung von verschiedenen Daten und vergleichen diese mit dem SARIMA Modell. Die genutzten Daten weisen ein großes Spektrum von Passagier Prognosen aus dem Flugverkehr bis zu Prognosen für den Energieverbrauch auf.

Die vorliegende Arbeit wurde mit den vier folgenden Arbeiten verglichen. Bei dem Vergleich wurde auf vergleichbare Resultate geachtet. Bei den verglichenen Arbeiten handelt es sich um die folgenden Quellen [21] [14] [18] [5] [17]. Die untersuchten Arbeiten weisen einen vergleichbaren Maßstab an Abweichung des MAPE auf. Die wissenschaftliche Arbeit [21] beschäftigt sich mit den Prognosen von Stromspitzen Zahlen um die Energiekosten zu prognostizieren. Eine weitere Arbeit [14] beschäftigt sich mit der Prognose von Flugpassagieren. In den verglichenen Arbeiten wurden das Prophet-Modell mit anderen Modellen verglichen wie z.B. dem SARIMA-Modell. Bei den meisten Prognosen in denen Prophet genutzt wurde in den Arbeiten handelte es sich um Prognosen über einen größeren Zeitraum als zwei Stunden, wie in dieser Arbeit. Diese Arbeit kann mit zuvor untersuchten meteorologischen und sozioökonomischen Werten vermutlich erweitert werden, um die Genauigkeit weiter zu erhöhen. Dieser Ansatz wird auch vom SMARD Modell genutzt, um präzise Prognosen erstellen zu können[6]. Bei den meteorologischen Werten könnte es sich um besondere Natureinflüsse handeln. Bei den sozioökonomischen Werten könnte es sich um kulturelle Aktivitäten handeln wie z.B. Ernten.

## 7 Fazit

Bei der Erstellung der Prognosen ist ihre Komplexität hervorgetreten. Bei der Analyse der Datensätze konnten viele Besonderheiten aus dem Alltag Deutschlands ermittelt werden. Zum einen ist ersichtlich, dass in Deutschland an Samstagen und Sonntagen weniger gearbeitet wird. Der niedrigere Energieverbrauch korreliert hier sichtbar (vgl. Abb. 5.2). Bei dieser Arbeit wurden die Modelle mit den SMARD Daten verglichen, welche sich auf Gesamt-Deutschland beziehen. Würde man einen kleineren Teil Deutschlands betrachten, so könnte man weitere Faktoren in die Prognose mit einbeziehen. Mittels der Prognose-Modelle konnten gute Ergebnisse im Vergleich mit den Prognosen der Bundesnetzagentur erstellt werden [6]. Die Modelle wiesen MAPE-Werte für eine zweistündige Prognose von unter 10 Prozent auf. Die Prognosen von SMARD belaufen sich auf ca. 3,9 %. Somit lässt sich sagen, dass die untersuchten Modelle einen guten Ansatz bieten, um Prognosen für Energieverbräuche zu erstellen. Bei den erstellten Prognosen handelt es wie zuvor erwähnt um Prognosen für gesamt Deutschland. Außerdem wurde der Zeitraum 01.01.2022 von 00:00 Uhr bis 02:00 Uhr untersucht. Dieser Zeitbereich wurde als Ausgangspunkt genommen, um anhand dessen die Anpassung vorzunehmen. Dies wurde gewählt, da bei der Analyse der Daten aufgefallen ist, dass dieser Zeitraum schwerer zu prognostizieren ist als andere Nächte. Bei der Auswertung der Prognosen viel auf, dass es Prognosen mit einer Abweichung von unter einem Prozent gab. Die Aussagen der Prozente wurden mit gerundeten Werten angenommen und können daher Abweichungen enthalten.

Somit lässt sich zur Untersuchung der Modelle abschließend aufstellen, dass die Modelle zur Prognose von Energieverbräuchen geeignet sind. Die Präzision der Prognosen kann durch Eingrenzen des Prognosebereiches verbessert werden. Da hierdurch weitere Parameter in die Prognose mit einbezogen werden können.

Ein möglicher Ausblick könnte sein, dass bei weiterer Untersuchung zusätzlicher Faktoren, die den Energieverbrauch der Personen beeinflussen. Ein möglicher Ansatz könnte die Identifizierung von Verhaltensmustern der Menschen sein. Mit Hilfe dessen könnten weitere Anpassungen an den Modellen vorgenommen werden. Ein Ansatz wäre es, die typischen Urlaubsmuster der Deutschen oder besondere kulturelle Events mit in die Prognosen einzubeziehen. Hierfür wirbt Prophet in manchen Modellen, da in Amerika der Sport American Football sehr weit vertreten ist [9]. Die Besonderheit an diesem ist, dass sich dieser an bestimmten Tagen im Jahr immer wiederholt oder daher mit in die Berechnung mit einbezogen werden kann.

Bei der Erstellung der Ergebnisse wurde insbesondere auf die Neujahrsnacht eingegangen. Diese wies die größten Schwankungen auf und wurde daher genauer untersucht. Die SMARD Daten beziehen sich auf Gesamt-Deutschland. Dies hat zufolge, dass Seasonalitäten die gezielte Bereiche Deutschlands beeinträchtigen nicht mit einbezogen werden konnten. Die Einbeziehung würde nur für Gesamt-Deutschland gehen und dies würde das Ergebnis beeinträchtigen. Außerdem ist bei den Simulationen die Berechnungszeit der Modelle zum Vorschein getreten. Die Berechnungszeiten der Simulationen müssen bei Simulationen mit berücksichtigt werden. Bei der Nutzung von der Funktion "auto\_arima" muss darauf geachtet werden, dass wenn die genutzten Historischen Daten eine hohe Seasonalität aufweisen, genügend Rechenleistung zur Verfügung stehen muss.

# Literaturverzeichnis

- [1] *pmdarima autoraima Funktion*. Web. – URL [https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto\\_arima.html](https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html). – Zugriffsdatum = 14.02.2024
- [2] *Schulferien der Bundesländer*. Web. Jan. 2024. – URL <https://www.schulferien.org/>. – Zugriffsdatum = 02.01.2024
- [3] AWS: *Was ist Python ?* Web. – URL [https://aws.amazon.com/de/what-is/python/#:~:text=Python%20ist%20eine%20Programmiersprache%2C%20die,Learning%20\(ML\)%20verwendet%20wird..](https://aws.amazon.com/de/what-is/python/#:~:text=Python%20ist%20eine%20Programmiersprache%2C%20die,Learning%20(ML)%20verwendet%20wird..) – Zugriffsdatum = 14.03.2024
- [4] BAJAJ, Aayush: *ARIMA SARIMA: Real-World Time Series Forecasting*. Web. Aug. 2023. – URL <https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide>. – Zugriffsdatum = 14.02.2024
- [5] BOUHADDOUR, Samya: *Tourism in Singapore, prediction model using SARIMA and PROPHET*. Web. – URL <https://pubs.aip.org/aip/acp/article/2508/1/020042/2878797/Tourism-in-Singapore-prediction-model-using-SARIMA>. – Zugriffsdatum = 26.02.2024
- [6] BUNDESNETZAGENTUR: *SMARD | Netz*. Web. – URL <https://www.smard.de/home/energiemarkt-erklaert/netz>. – Zugriffsdatum = 16.12.2023
- [7] BUTGEREIDT, Sören: *Diplomarbeit - Analyse der Prognosegüte alternativer Prognosemethoden bei kleinen Stichproben*. Web. – URL [https://publik.tuwien.ac.at/files/publik\\_281087.pdf](https://publik.tuwien.ac.at/files/publik_281087.pdf). – Zugriffsdatum = 14.02.2024
- [8] DEVELOPERS scikit-learn: *MAPE*. Web. – URL [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean\\_absolute\\_percentage\\_error.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_percentage_error.html). – Zugriffsdatum = 29.02.2024

- [9] FACEBOOK: *Prophet Quick Start*. Web. Sep. 2017. – URL [https://facebook.github.io/prophet/docs/quick\\_start.html#python-api](https://facebook.github.io/prophet/docs/quick_start.html#python-api)
- [10] GMBH, Codemi: *Feiertage Deutschland*. Web. 2015. – URL <https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide>. – Zugriffsdatum = 02.01.2024
- [11] IAN GOODFELLOW, Aaron C.: *Deep Learning - Das umfassende Handbuch*. mitp, 2018. – ISBN 978-3-95845-700-3
- [12] JOSEF PERKTOLD, Jonathan T.: *statistical models, hypothesis tests, and data exploration*. Web. – URL <https://www.statsmodels.org/stable/index.html>. – Zugriffsdatum = 14.02.2024
- [13] KAMPS, Prof. Dr. U.: *Definition: Was ist statistische Testverfahren?* Web. – URL <https://wirtschaftslexikon.gabler.de/definition/statistische-testverfahren-41922#:~:text=Statistische%20Testverfahren%20sind%20diejenigen%20Methoden,mithilfe%20eines%20Stichprobenbefundes%20getroffen%20wird.&text=%C3%9Cber%20200%20Experten%20aus%20Wissenschaft%20und%20Praxis..> – Zugriffsdatum = 16.03.2024
- [14] KORRA SATHYA BABU, Santos Kumar Das Abhirup A.: *Time Series based Air Pollution Forecasting using SARIMA and Prophet Model*. Web. Aug. 2019. – URL <https://dl.acm.org/doi/abs/10.1145/3355402.3355417>. – Zugriffsdatum = 26.02.2024
- [15] KORSTANJE, Joos: *Advanced Forecasting with Python*. Apress, 2021. – URL <https://doi.org/10.1007/978-1-4842-7150-6>. – ISBN 978-1-4842-7150-6
- [16] KULTUSMINISTERIUM: *Archiv der Ferientermine*. Web. Dez. 2023. – URL <https://www.kmk.org/service/ferien/archiv-der-ferientermine.html>. – Zugriffsdatum = 02.01.2024
- [17] OSKAR TRIEBE, Polina Pilyuginad Nikolay Laptevb Christoph Bergmeirc Ram R.: *NeuralProphet: Explainable Forecasting at Scale*. Web. Nov. 2021. – URL <https://arxiv.org/pdf/2111.15397.pdf>. – Zugriffsdatum = 26.02.2024
- [18] OSKAR TRIEBE, Polina Pilyuginad Nikolay Laptevb Christoph Bergmeirc Ram R.: *Predicting city-scale daily electricity consumption using data-driven models*. Web. Mai. 2021. – URL <https://www.sciencedirect.com/science/article/pii/S2666792421000184#sec0004>. – Zugriffsdatum = 26.02.2024

- [19] RAINER SCHLITTEGEN, Cristina S.: *Angewandte Zeitreihenanalyse mit R*. De Gruyter Oldenbourg, 2020. – ISBN 978-3-11-069409-3
- [20] SEAN J. TAYLOR, Benjamin L.: *Forecasting at scale*. Web. Sep. 2017. – URL <https://peerj.com/preprints/3190/>. – Zugriffsdatum = 16.12.2023
- [21] STEFANO FRIZZO STEFENON, Laio Oriol S.: *Aggregating Prophet and Seasonal Trend Decomposition for Time Series Forecasting of Italian Electricity Spot Prices*. Web. – URL <https://www.mdpi.com/1996-1073/16/3/1371>. – Zugriffsdatum = 26.02.2024
- [22] STEFMOLIN: *Pandas Workshop*. Web. Dez. 2023. – URL <https://github.com/stefmolin/pandas-workshop>. – Zugriffsdatum = 16.12.2023



# A Anhang

## A.1 Verwendete Hilfsmittel

In der Tabelle A.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle A.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
L <sup>A</sup> T <sub>E</sub> X	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments
CD	Der Programmcode befindet sich auf der CD und kann beim Erstprüfer eingesehen werden

## **Erklärung zur selbständigen Bearbeitung**

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

---

Ort

Datum

Unterschrift im Original