

Machine-Learning-basiertes Forecasting des Elektromobilitätshochlaufs in Deutschland

Masterarbeit

zur Erlangung des Grades Master of Science Wirtschaftsinformatik
des Fachbereichs Wirtschaft der
Technischen Hochschule Brandenburg

vorgelegt von:

Taj Mohammad Ghulam Zada
geb. am 01. Juli 1991 in Kunduz, Afghanistan
5. Semester

Betreuerin: Prof. Dr. Olga Levina
Zweitgutachter: Dr.-Ing. Alexander Wanitschke

Ghulam Zada, Taj Mohammad Master Wirtschaftsinformatik 5. Semester	Binsenhof 39 14478 Potsdam ghulamza@th-brandenburg.de Matr.-Nr. 20219673 Abgabe: 06.12.2022
--	---

Potsdam, den 29. November 2022

Abstract

Forecasting is an important decision-making tool for businesses, organizations and governments because it saves a lot of resources. The forecast of the market share of new BEV car registrations in Germany can help efficiently evaluate strategic and future-oriented decisions, e.g. the climate targets set by the federal government. Machine learning methods were used to implement forecasting.

Two machine learning models were used to forecast of the market share of new BEV car registrations in Germany: SARIMA(X) and DeepAR model. Due to variety in input datasets, four univariate and multivariate time series analysis models were trained. Afterwards all four trained models were compared with each other using the RMSE metric. The last six months of the dataset were used to evaluate the model quality.

The result of model comparison shows that SARIMAX model needs the exogenous variables to create a forecast of the multivariate data sets. However, the DeepAR model does not require any exogenous variables to generate forecast. Furthermore, if the input dataset contains at least 300 observations, then the implementation of DeepAR model is much more advantageous than SARIMA(X) model. DeepAR model not only supports the univariate and multivariate datasets, but also creates a global model that generates a forecast per variable. This is a great advantage over the SARIMA(X) model, which is not capable of this.

Keywords: Forecasting, Time Series Analysis, SARIMA(X), DeepAR, Hyperparameter Optimization, Battery Electric Vehicles (BEV), Electric Mobility

Zusammenfassung

Forecasting ist eine wichtige Methode zur Entscheidungsfindung für Unternehmen, Organisationen und Regierungen, da so der Ressourcenverbrauch reduziert werden kann. Der Forecast des prozentualen Marktanteils der neuen BEV-PKW-Zulassungen in Deutschland kann helfen, um strategische und zukunftsorientierte Entscheidungen, z.B. die vom Bund gesetzten Klimaziele, besser steuern zu können. Zur Durchführung des Forecastings wurde der Einsatz unterschiedlicher Machine-Learning-Methoden ausgenutzt.

Es wurden zwei Machine-Learning-Modelle für das Forecasting des prozentualen Marktanteils der BEV-PKW-Neuzulassungen in Deutschland verwendet: Die SARIMA(X)- und DeepAR-Modelle. Aufgrund der Unterschiede in den Eingabedatensätzen wurden vier Modelle der Univariate- und Multivariate-Zeitreihenanalyse trainiert. Mit Hilfe der RMSE-Fehlermetrik wurden die vier trainierten Modelle miteinander verglichen. Die letzten sechs Monate der Eingabedatensätze wurden zur Evaluierung der Modellqualität genutzt.

Die Ergebnisse dieser Masterarbeit zeigen, dass das SARIMAX-Modell zur Erstellung einer Prognose der multivariaten Datensätze die exogenen Variablen benötigt. Allerdings braucht das DeepAR-Modell keine exogenen Variablen, um eine Prognose generieren zu können. Wenn der Eingabedatensatz mindestens 300 Zeitreihen beträgt, eignet sich das DeepAR-Modell besser als das SARIMA(X)-Modell. Das DeepAR-Modell unterstützt nicht nur die univariaten- und multivariaten Datensätze, sondern generiert auch ein globales Modell, wodurch eine Prognose je Variable erstellt wird. Das ist ein großer Vorteil gegenüber dem SARIMA(X)-Modell, welches dazu nicht in der Lage ist.

Schlagwörter: Forecasting, Zeitreihenanalyse, SARIMA(X), DeepAR, Hyperparameteroptimierung, Batterieelektrisches Fahrzeug (BEV), Elektromobilität

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
1 Einleitung	1
1.1 Motivation der Masterarbeit	1
1.2 Stand der Forschung	2
1.3 Forschungsfragen.....	4
1.4 Zielsetzung	4
1.5 Eingabedatensatz	5
1.6 Aufbau der Masterarbeit	7
2 Theoretische Grundlagen	8
2.1 Machine Learning	8
2.2 Forecasting	10
2.3 Modellvergleich.....	11
2.3.1 Autoregressive (AR) Model.....	11
2.3.2 Moving Average (MA) Model	12
2.3.3 Autoregressive Moving Average (ARMA) Model	13
2.3.4 Autoregressive Integrated Moving Average (ARIMA) Model ...	14
2.3.5 Seasonal Autoregressive Integrated Moving Average (SARIMA) Model	16
2.3.6 ARCH- und GARCH Model.....	16
2.3.7 Vector Autoregression (VAR) Model.....	16
2.3.8 ETS-Model	17
2.3.9 Prophet Model	18
2.3.10 Recurrent Neural Network (RNN) Model	19
2.3.11 DeepAR Model	20
2.4 Fehlermetriken.....	20
2.4.1 Mean Absolute Error (MAE).....	21
2.4.2 Mean Absolute Percent Error (MAPE).....	21

2.4.3	Mean Squared Error (MSE)	22
2.4.4	Root Mean Squared Error (RMSE).....	22
3	Methodologie	25
3.1	Technische Ausstattung und Codestruktur.....	25
3.2	Explorative Datenanalyse	26
3.2.1	Explorative Analyse der univariaten Daten	26
3.2.2	Explorative Analyse der multivariaten Daten	31
3.3	Univariate Zeitreihenanalyse	33
3.3.1	Train-Test-Split	33
3.3.2	SARIMA	35
3.3.3	DeepAR	37
3.3.4	Hyperparameteroptimierung (HPO).....	41
3.4	Multivariate Zeitreihenanalyse	45
3.4.1	Train-Test-Split	45
3.4.2	SARIMAX.....	46
3.4.3	DeepAR	47
3.4.4	Hyperparameteroptimierung (HPO).....	50
4	Ergebnisse	53
4.1	Hyperparameteroptimierung	53
4.2	Univariate Zeitreihenanalyse	55
4.2.1	SARIMA	55
4.2.2	DeepAR	61
4.3	Multivariate Zeitreihenanalyse	65
4.3.1	SARIMAX.....	65
4.3.2	DeepAR	70
4.4	Vergleich der HPO-Methoden und der Modelle.....	73
5	Diskussion und Zusammenfassung	76
5.1	Reflexion der Ergebnisse	76
5.2	Einschränkungen	78
5.3	Zusammenfassung und Ausblick.....	79
	Literaturverzeichnis	82
	Anhang	89
	Ehrenwörtliche Erklärung.....	96

Abbildungsverzeichnis

Abbildung 1: Prozentuale Marktanteil der BEV-PKW-Neuzulassungen in Deutschland	5
Abbildung 2: Prozentuale Marktanteil der BEV-PKW-Neuzulassungen der 33 Ländern.....	6
Abbildung 3: ETS-Dekomposition des univariaten Datensatzes	27
Abbildung 4: Trend des univariaten Datensatzes.....	28
Abbildung 5: Saisonalität des univariaten Datensatzes	29
Abbildung 6: Saisonalität des univariaten Datensatzes pro Quartal	29
Abbildung 7: Monatliche Saisonalität des univariaten Datensatzes	30
Abbildung 8: Residual des univariaten Datensatzes	31
Abbildung 9: Bereinigte Eingabedatensatz zur Multivariate-Zeitreihenanalyse	32
Abbildung 10: Train-Test-Split des univariaten Datensatzes	34
Abbildung 11: Zusammenfassung des DeepAR Modells.....	39
Abbildung 12: Train-Test-Split des multivariaten Datensatzes.....	45
Abbildung 13: Modellzusammenfassung des SARIMA-Modells	56
Abbildung 14: Optimierungsverlaufs des SARIMA-Modells	57
Abbildung 15: Hyperparameterwichtigkeit des SARIMA-Modells.....	58
Abbildung 16: Modell-Diagnostik des SARIMA-Modells.....	59
Abbildung 17: SARIMA-Modellevaluierung	60
Abbildung 18: Prognose des SARIMA-Modells.....	61
Abbildung 19: Optimierungsverlaufs des Univariate-DeepAR-Modells	62
Abbildung 20: Hyperparameterwichtigkeit des Multivariate-DeepAR-Modells ..	62
Abbildung 21: Modellevaluierung des Univariate-DeepAR-Modells.....	63
Abbildung 22: Prognose des Univariate-DeepAR-Modells.....	64
Abbildung 23: Modellzusammenfassung des SARIMAX-Modells	65
Abbildung 24: Optimierungsverlaufs des SARIMAX-Modells.....	66

Abbildung 25: Hyperparameterwichtigkeit des SARIMAX-Modells	67
Abbildung 26: Modell-Diagnostik des SARIMAX-Modells	68
Abbildung 27: SARIMAX-Modellevaluierung	69
Abbildung 28: Optimierungsverlaufs des Multivariate-DeepAR-Modells.....	70
Abbildung 29: Hyperparameterwichtigkeit des Multivariate-DeepAR-Modells ..	71
Abbildung 30: Modellevaluierung des Multivariate-DeepAR-Modells.....	72
Abbildung 31: Prognose des Multivariate-DeepAR-Modells	72
Abbildung 32: RMSE-Werte der trainierten Modelle	74

Tabellenverzeichnis

Tabelle 1: Parameter des SARIMA-Modelltrainings.....	35
Tabelle 2: Parameter des SARIMA-Modells zur HPO.....	37
Tabelle 3: Parameter des Univariate-DeepAR-Modelltrainings.....	39
Tabelle 4: Parameter des Univariate-DeepAR-Modells zur HPO.....	41
Tabelle 5: Eingeschränkte Parameterwerte des Univariate-DeepAR-Modells	42
Tabelle 6: Auto-ARIMA-Parameter zur HPO (Univariate-Zeitreihenanalyse)..	43
Tabelle 7: Parameter des SARIMAX-Modelltrainings	46
Tabelle 8: Parameter des SARIMAX-Modells zur HPO	47
Tabelle 9: Parameter des Multivariate-DeepAR-Modelltrainings	48
Tabelle 10: Parameter des Multivariate-DeepAR-Modells zur HPO	49
Tabelle 11: Eingeschränkte Parameterwerte des Multivariate-DeepAR-Modells	50
Tabelle 12: Auto-ARIMA-Parameter zur HPO (Multivariate-Zeitreihenanalyse)	51
Tabelle 13: Vergleich der HPO-Methoden	54
Tabelle 14: RMSE-Werte der HPO	73
Tabelle 15: Anteil der BEV-PKW-Neuzulassungen in Deutschland.....	89
Tabelle 16: Vergleich der Forecasting-Modelle.....	90
Tabelle 17: Die ersten 20 Grid-Search-Ausgabe des SARIMA-Modells	91
Tabelle 18: Die ersten 20 Grid-Search-Ausgabe des Univariate-DeepAR-Modells	92
Tabelle 19: Die ersten 20 Grid-Search-Ausgabe des SARIMAX-Modells.....	93
Tabelle 20: Die ersten 20 Grid-Search-Ausgabe des Multivariate-DeepAR- Modells.....	94
Tabelle 21: Prognose des SARIMA-Modells	95
Tabelle 22: Prognose des Univariate-DeepAR-Modells.....	95
Tabelle 23: Prognose des Multivariate-DeepAR-Modells.....	95

Abkürzungsverzeichnis

AR	Autoregressive
MA	Moving Average
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
ACF	Autocorrelation Function
PACF	Partial Autocorrelation Function
SARIMA	Seasonal Autoregressive Integrated Moving Average
SARIMAX	Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors
RNN	Recurrent Neural Network
DeepAR	Deep Autoregressive Neural Network
ARCH	Autoregressive Conditional Heteroscedasticity
GARCH	Generalized Autoregressive Conditional Heteroscedasticity
VAR	Vector Autoregression
VARMA	Vector Autoregression Moving Average
GluonTS	Gluon Time Series
MAE	Mean Absolute Error
MAPE	Mean Absolute Percent Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
BEV-PKW	Batterieelektrische Personenkraftwagen
HPO	Hyperparameteroptimierung
BEVs	Batterieelektrische Fahrzeuge

TPE Tree-Structured Parzen Estimator

ADFuller-Test Augmented Dickey Fuller Test

1 Einleitung

In diesem Kapitel wird einführend die Motivation und die Zielsetzung der vorliegenden Masterarbeit beschrieben. Daraufhin werden der Stand der Forschung und die in dieser Arbeit behandelten Forschungsfragen beleuchtet. Es folgt eine Einführung zum Eingabedatensatz, um eine gemeinsame Basis für den Einsatz von Zeitreihenanalysen des prozentualen Marktanteils der neuen batterieelektrischen Personenkraftwagen-Zulassungen in Deutschland zu schaffen. Abschließend wird der Aufbau dieser Masterarbeit vorgestellt.

1.1 Motivation der Masterarbeit

Forecasting ist eine wichtige Methode zur Entscheidungsfindung für Unternehmen, Organisationen und Regierungen, da sie es ermöglicht, den Ressourcenverbrauch zu reduzieren. Beispielsweise kann der Produktionsablauf entsprechend einer Marktnachfrage effizienter geplant, Kundennachfrage optimal ausgeglichen bzw. die Kundenzufriedenheit signifikant erhöht werden. Der Forecast des prozentualen Marktanteils batterieelektrischer Elektromobilität, also die batterieelektrischen Personenkraftwagen (BEV-PKW)-Neuzulassungen, in Deutschland kann helfen, um strategische und zukunftsorientierte Entscheidungen, z.B. die vom Bund gesetzten Klimaziele, besser steuern zu können. Laut dem Klimaziel der Bundesregierung sollen Deutschlands CO₂ Emissionen bis zum Jahr 2030 um 65 Prozent sinken, während bis zum Jahr 2045 die Treibhausgasneutralität erreicht werden soll (*Generationenvertrag für das Klima*, 2022). Batterieelektrische Fahrzeuge (BEVs) sind ein wichtiges Mittel, um das Klimaziel zu erreichen, da BEVs keine bzw. sehr wenig CO₂ verursachen.

Der Forecast des prozentualen Marktanteils der BEV-PKW-Neuzulassungen in Deutschland in dieser Abschlussarbeit wurde im Rahmen der Aktivitäten des Teams Daten und Analysen der NOW GmbH durchgeführt. „Die NOW GmbH arbeitet für die Zukunft emissionsfreier Technologien in einem integrierten Energiesystem. Als bundeseigene GmbH nimmt NOW GmbH im Bereich nachhaltige Mobilität und Energieversorgung Aufträge von Bundesministerien an.“

Zumeist umfassen diese Aufträge die Umsetzung und Koordination von Förderprogrammen.“ (NOW GmbH, 2022) .

Um die Prognose des prozentualen Marktanteils der neuen BEV-PKW-Zulassungen in Deutschland erzielen zu können, werden Machine-Learning-Verfahren, vor allem Zeitreihenanalyse (engl. time series analysis), zum Einsatz gebracht. Es werden verschiedene Zeitreihenanalyse-Modelle untersucht, die eine Prognose des prozentualen Marktanteils der neuen BEV-PKW-Zulassung in Deutschland treffen können. Die Datengrundlage hierfür ist bereits geschaffen. Es liegen bereinigte und validierte Daten zu monatlichen PKW-Neuzulassungen mit verschiedenen Attributen vor, darunter vor allem Zulassungsbezirk und Antriebsart der neuzugelassenen PKW. Dadurch könnten beispielweise politische Entscheidungsträger die Prognose zum Markthochlauf der BEV-PKW-Neuzulassungen nutzen, um Marktförderung und andere Policy-Mechanismen optimal definieren zu können.

1.2 Stand der Forschung

„Batterieelektrische Fahrzeuge (BEVs) werden mit einem elektrischen Traktionsmotor anstelle eines Verbrennungsmotors betrieben [...]. Da das Fahrzeug mit Strom betrieben wird, stößt es keine Abgase aus und benötigt keine typischen flüssigen oder gasförmigen Kraftstoffkomponenten wie Kraftstoffpumpe, Kraftstoffleitung oder Kraftstofftank.“ (Wanitschke, 2021, p. 8). Durch die Substitution der konventionellen Antriebsarten durch Elektrofahrzeuge wird versucht den Ausstoß des klimaschädlichen Treibhausgases langfristig zu senken. Es wurden bereits Forschungen durchgeführt, um die BEV-PKW-Neuzulassungen in Deutschland prognostizieren zu können, die im Folgenden beschrieben werden.

Hennes et al. haben den Einfluss von CO₂-Preisen auf den Gebäude-, PKW-Verkehrs- und Energiesektor durch die Kopplung von ökonometrischen Modellen mit einem technoökonomischen Energiesystemmodell analysiert (Hennes *et al.*, 2021). Durch die ökonometrischen Modelle wurden die Prognosen der BEV-PKW-Neuzulassungen und anderer Antriebsarten erstellt. Dabei wurde für jede Antriebsart eine individuelle Schätzgleichung aufgestellt. Um die BEV-PKW-Neuzulassungen zu prognostizieren, wurden die Faktoren wie PKW-Kaufpreis und Kraftstoffpreise sowie Zubau an öffentlichen Ladestationen berücksichtigt. Infolgedessen wurden jährliche Prognosen bis 2050 erstellt, wobei festgestellt wurde, dass die Anzahl der BEV-PKW-Neuzulassungen von Strompreis und die

Anzahl der öffentlichen Ladestationen abhängig ist. Obwohl die ökonometrischen Modelle zur Prognose der BEV-PKW-Neuzulassungen durchgeführt wurden, wurde kein reines Machine-Learning-Verfahren bzw. keine Zeitreihenanalyse zur Erstellung der Prognosen der BEV-PKW-Neuzulassungen in Deutschland implementiert.

Darüber hinaus haben Gnann et al das sogenannte ALADIN-Modell verwendet, um die BEV-PKW-Neuzulassungen in Deutschland bis 2030 zu prognostizieren (Gnann *et al.*, 2022, pp. 5–18). Das ALADIN-Modell wurde durch das Fraunhofer-Institut für System- und Innovationsforschung entwickelt und enthält ein detailliertes Simulationsmodell für die Marktdurchdringung von Fahrzeugen mit alternativen Antrieben. Das ALADIN-Modell beruht auf agentenbasierter Simulation. Hierbei handelt es sich um eine Simulationsmethode, die auf der Modellierung von einzelnen Agenten, wie etwa verschiedenen Arten von PKWs, basiert. Es wurden drei verschiedene Szenarien zum jährlichen Markthochlauf der BEV-PKW-Neuzulassungen bis zum Jahr 2030 simuliert und analysiert, ob das im Bundes-Klimaschutzgesetz (KSG) festgelegte Reduktionsziel für den Verkehrssektor zu erreichen ist. Allerdings sind die agentenbasierten Simulationen für kurze Datensätze nicht geeignet. Eine Einführung zu den Eingabedatensätze dieser Masterarbeit sind auf Seite 5 beschrieben.

Eine weitere Forschungsarbeit zur Prognose der Anzahl der BEV-PKW-Neuzulassungen auf der regionalen Ebene in Deutschland wurde durch Speth et al. durchgeführt, wobei neben dem ALADIN-Modell auch die multiple lineare Regression verwendet wurde (Speth *et al.*, 2020, pp. 1–7). Hierbei wurden die aktuellen regionalen Fahrzeugbestände und regionalen Entwicklungsdaten, wie z.B. Bevölkerung, Einkommen und räumliche Struktur, als Variablen im Modell verwendet, um jährliche Prognosen pro Bundesland bis zum Jahre 2050 zu treffen. Es ergab sich, dass eine höhere Kaufkraft und höhere Bevölkerungszahl die Hauptgründe für den höheren Bestand an Elektrofahrzeugen der verschiedenen Gebiete in den Jahren 2030 und 2050 sein werden. Allerdings wurden in den obengenannten Prognose-Methoden keine Machine-Learning-Verfahren verwendet oder monatliche Prognosen erstellt. Aus diesem Grund werden in dieser Abschlussarbeit verschiedene Machine-Learning-Modelle, die den prozentualen Marktanteil der neuen BEV-PKW-Zulassungen vorhersagen können, verglichen und davon zwei passende Modelle zur Erstellung der monatlichen Prognose ausgewählt.

1.3 Forschungsfragen

Für diese Masterarbeit fiel die Wahl auf die quantitative empirische Forschung. Folgende Forschungsfragen wurden zu Beginn dieser Masterarbeit definiert:

1. Welche Möglichkeiten bietet Machine-Learning für das Forecasting des Elektromobilitätshochlaufs?
2. Zwischen welchen Zielkonflikten ist bei der Auswahl und dem Design eines Forecasting-Machine-Learning-Modells abzuwägen?
3. Wie kann das vielversprechendste Machine-Learning-Modell so trainiert und hyperparametrisiert werden, dass maximale Forecasting-Güte erzielt werden kann?

1.4 Zielsetzung

Übergeordnetes Ziel der Masterarbeit ist das Problem selbst sowie mögliche Lösungsansätze konzeptionell zu verstehen, um bestmögliche Entscheidungen zur Machine-Learning-basierten Umsetzung eines Forecastings treffen zu können. Es gibt selbstverständlich Alternativen zum Forecasting mit einfachem ARIMA-Modell. Dies wird in Kapitel zwei auf Seite 11 ausführlich erläutert.

Im Rahmen dieser Arbeit werden verschiedene Machine-Learning-Modelle, die den Forecast des prozentualen Marktanteils der neuen BEV-PKW-Zulassungen in Deutschland ermitteln können, miteinander verglichen. Es wurden zwei unterschiedliche Datensätze zum Forecast der neuen BEV-PKW-Zulassungen zur Verfügung gestellt: Der univariate und der multivariate Datensatz. Anhand der Datensatz-Merkmale werden die Univariate- und Multivariate-Zeitreihenanalysen-Modelle analysiert und davon zwei passende Modelle ausgewählt.

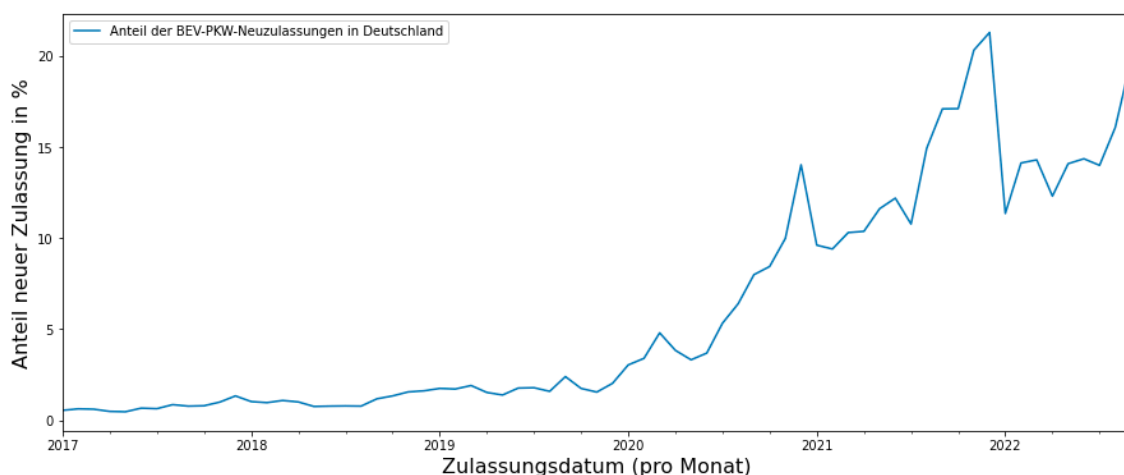
Darüber hinaus werden die gängigsten Fehlermetriken analysiert und eine Fehlermetrik ausgewählt, um die Qualität der ausgewählten Univariate- und Multivariate-Zeitreihenanalyse-Modelle auszuwerten. Zum Trainieren der Modelle und Finden der besten Modellparameterwerte werden die Methoden der Hyperparameteroptimierung (HPO), wie Auto-ARIMA, Grid-Search und Optuna genutzt.

1.5 Eingabedatensatz

Diese Abschlussarbeit beschäftigt sich mit dem Forecasting des Elektromobilitätshochlaufs in Deutschland. Zu diesem Zweck wird der Eingabedatensatz der prozentualen Marktanteile der neuen BEV-PKW-Zulassungen verwendet. Der prozentuale Marktanteil der BEV-PKW-Neuzulassungen wird aus der Division von BEV-PKW-Neuzulassungen durch PKW-Neuzulassung mit allen anderen Antriebsarten, wie z.B. Diesel, Benzin oder Gas sowie der Multiplikation mit 100 gebildet. Es wurden zwei Datensätze jeweils für Univariate- und Multivariate-Zeitreihenanalyse gebildet.

In einer Univariate-Zeitreihenanalyse werden Modelle verwendet, die nur eine Zeitreihe (engl. time series) als Zielvariable zur Erstellung einer Prognose enthalten. In einer Multivariate-Zeitreihenanalyse werden Forecasting-Modelle verwendet, um die Beziehungen zwischen mehreren Zeitreihen zu beschreiben, zu analysieren und eine Prognose zu treffen (Box *et al.*, 2015, p. 505; Tsay, 2015, p. 1). Das Hauptziel zur Verwendung der Multivariate-Zeitreihenanalyse ist die Untersuchung der dynamischen Beziehungen zwischen den Variablen und die Verbesserung der Prognosegenauigkeit.

Der Eingabedatensatz für die Univariate-Zeitreihenanalyse besteht aus einer Variable bzw. einem Feature. Diese ist in dieser Arbeit der prozentuale Marktanteil der BEV-PKW-Neuzulassungen in Deutschland. Dieser Datensatz umfasst eine Laufzeit von 69 Monaten bzw. 69 Zeitpunkten, wie in der folgenden Abbildung dargestellt ist.



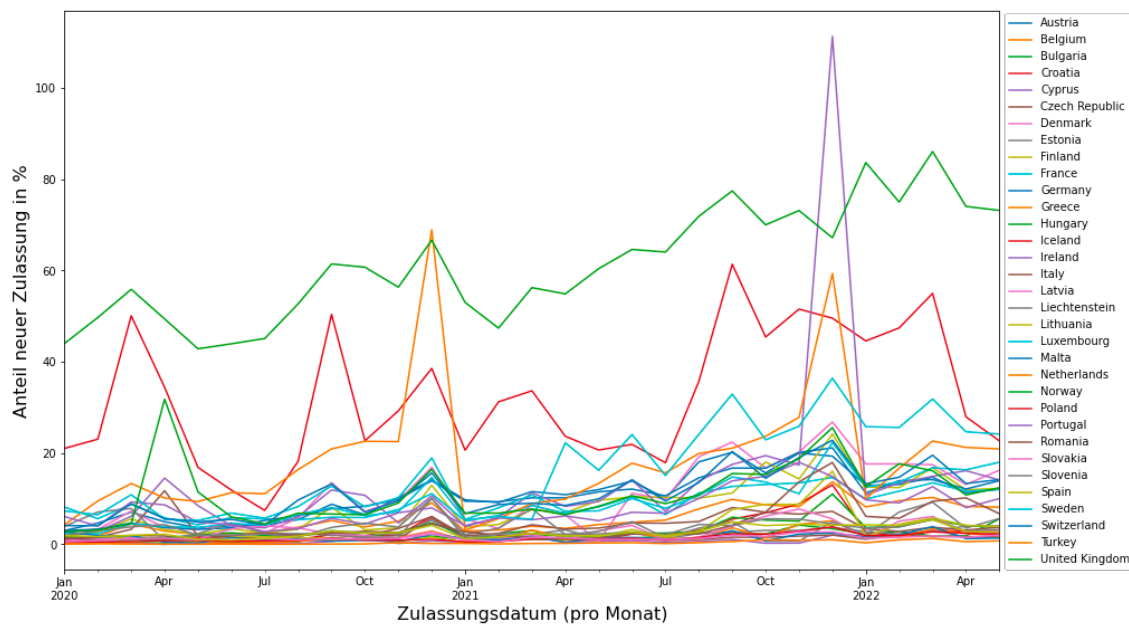
Quelle: Eigene Darstellung, basierend auf Kraftfahrt-Bundesamt, 2022

Abbildung 1: Prozentuale Marktanteil der BEV-PKW-Neuzulassungen in Deutschland

Die Quelle des Eingabedatensatzes zur Univariate-Zeitreihenanalyse ist das Kraftfahrt-Bundesamt (*Kraftfahrt-Bundesamt*, 2022). Der Datensatz umfasst den

Zeitraum vom Januar 2017 bis einschließlich September 2022. Die Tabelle 15 im Anhang zeigt den vollständigen Eingabedatensatz der Univariate-Zeitreihenanalyse.

Der Datensatz für die Multivariate-Zeitreihenanalyse besteht aus 33 Variablen bzw. Features. Diese umfassen die prozentualen Marktanteile der BEV-PKW Neuzulassungen der 27 EU-Länder¹ und weiteren sechs Ländern (Island, Liechtenstein, Norwegen, Schweiz, Türkei und Vereinigtes Königreich). Die Abbildung 2 stellt diese 33 Variablen dar.



Quelle: Eigene Darstellung, basierend auf European Alternative Fuels Observatory, 2022
Abbildung 2: Prozentuale Marktanteil der BEV-PKW-Neuzulassungen der 33 Ländern

Der Datensatz zur Multivariate-Zeitreihenanalyse wurde von European Alternative Fuels Observatory (EAFO) genommen (*European Alternative Fuels Observatory, 2022*). Im Gegenteil zum Eingabedatensatz der Univariate-Zeitreihenanalyse fängt der Eingabedatensatz zur Multivariate-Zeitreihenanalyse ab Januar 2020 an und endet im Mai 2022. Infolgedessen enthält der Multivariate-Datensatz 29 Monaten, während der Univariate-Datensatz aus 69 Monaten besteht. Die Hauptvariable oder Zielvariable (engl. target variable) bei der Multivariate-Zeitreihenanalyse ist die BEV-PKW-Neuzulassungen in Deutschland, während die anderen 32 Variablen bzw. Ländern als exogene

¹ EU-Länder (Stand vom November 2022): Belgien, Bulgarien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, Niederlande, Österreich, Polen, Portugal, Rumänien, Schweden, Slowakei, Slowenien, Spanien, Tschechische Republik, Ungarn und Zypern

Variablen, auch externe Faktoren genannt, in der Multivariate-Zeitreihenanalyse verwendet werden.

1.6 Aufbau der Masterarbeit

Die Erarbeitung dieser Arbeit gliedert sich in fünf Phasen, die nachfolgend kurz beschrieben werden.

Basierend auf der vorangegangenen Literaturrecherche und -analyse werden im zweiten Kapitel die Grundlagen zur Zeitreihenanalyse eingeführt. Dabei werden die Methoden und Theorien des Machine-Learning und Forecasting beschrieben. Danach werden unterschiedliche Forecasting-Modelle miteinander verglichen und zwei passende Modelle zur technischen Implementierung ausgewählt. Abschließend werden die wichtigsten Fehlermetriken zur Zeitreihenanalyse vorgestellt und eine Fehlermetrik unter Berücksichtigung der Datenlage bzw. der Machbarkeit ausgewählt.

Im dritten Kapitel wird die Methodologie zur Umsetzung der Univariate- und Multivariate-Zeitreihenanalysen detailliert erklärt. Neben der technischen Ausstattung, Codestruktur und der explorativen Datenanalyse des Eingabedatensatzes werden auch die SARIMA(X)- und DeepAR-Modelle der Univariate- und Multivariate-Zeitreihenanalyse sowie deren HPO erläutert. Darüber hinaus werden die verwendeten Parameter zum Training der jeweiligen Modelle detailliert interpretiert.

Im vierten Kapitel erfolgen die Ergebnisse dieser Masterarbeit, nämlich die Prognosen der Univariate- und Multivariate-Zeitreihenanalysen sowie der Vergleich der HPO-Methoden. Dabei werden insbesondere die Grid-Search- und Optuna-Methoden zur Optimierung der zu trainierenden Modelle beschrieben. Zuletzt erfolgt der Vergleich der zwei Modelle der Univariate- und Multivariate-Zeitreihenanalyse, nämlich SARIMA(X)- und DeepAR-Modell.

Im fünften Kapitel werden die Ergebnisse dieser Arbeit zusammengefasst und reflektiert. Abschließend werden die Grenzen dieser Arbeit sowie zukünftige Forschungsmöglichkeiten in diesem Bereich beleuchtet.

2 Theoretische Grundlagen

In diesem Kapitel werden die Grundlagen zur Zeitreihenanalyse erklärt. Zuerst werden die Methoden und Theorien des Machine-Learnings- und Forecastings beschrieben. Hierbei liegt der Fokus auf den Definitionen von Machine-Learning und Forecasting. Danach werden unterschiedliche Forecasting-Modelle miteinander verglichen und zwei passende Modelle zur technischen Implementierung ausgewählt. Abschließend werden die wichtigsten Fehlermetriken zur Zeitreihenanalyse vorgestellt und eine Fehlermetrik mit Begründung ausgewählt. Ziel der Auswahl eines Fehlermaßes ist die Evaluierung aller Modelle beispielsweise Univariate- und Multivariate-Zeitreihenmodelle.

2.1 Machine Learning

Das sogenannte Machine Learning ist ein Zweig der Informatik, der allgemein darauf abzielt, Computern das „Lernen“ zu ermöglichen (Bi *et al.*, 2019, p. 2222). Auf Grund der Vielfalt großer Datenmengen, die mit hoher Geschwindigkeit sowohl bei Unternehmen als auch bei Einzelpersonen anfallen, spricht man heute von Big Data, deren Auswertung eine Herausforderung darstellt. Je mehr Daten zur Verfügung stehen, desto größer ist der Wissensgewinn, aber umso schwerer ist auch das gezielte Ableiten der Informationen. Machine-Learning, ein Bereich der künstlichen Intelligenz (KI), konzentriert sich auf die Verwendung von Daten und Algorithmen zur Nachahmung des menschlichen Lernens. Ihr Ziel ist es, dass sich Maschinen im Laufe der Zeit verbessern, bei Vorhersagen oder Klassifizierungen immer genauer werden oder datengesteuerte Erkenntnisse aufdecken können (*3 Types of Machine Learning You Should Know*, 2022). Die Anwendungen von Machine-Learning sind, beispielsweise Sprach- und Bilderkennung, Spam- und Betrugserkennung oder Data Mining.

Durch Machine-Learning-Methoden werden Computer so programmiert, dass sie ein Leistungskriterium anhand von Beispieldaten oder Erfahrungen aus der Vergangenheit optimieren (Alpaydın, 2014, pp. 2–4). In Computersystemen existiert Erfahrung in Form von Daten. Infolgedessen besteht die Hauptaufgabe

des Machine-Learning darin, Lernalgorithmen zu entwickeln, die Modelle aus Daten erstellen (Zhou, 2021, pp. 2–3). Ein Machine-Learning-Modell kann vorhersagend (engl. predictive) sein, um Prognosen für die Zukunft zu treffen, oder beschreibend (engl. descriptive), um Erkenntnisse aus Daten zu gewinnen, oder beides. Machine-Learning kann sich auf vier Arten klassifizieren lassen, nämlich Supervised Machine Learning, Unsupervised Machine Learning, Semi-Supervised Machine Learning und Reinforcement Learning.

Beim Supervised-Machine-Learning wird vorausgesetzt, dass der Datensatz gelabelt ist (engl. labeled data). Das bedeutet, dass der Ergebniswert einer bestimmten Beobachtung bekannt ist. Daten mit bestimmten Ergebniswerten werden als „Labeled Data“ bezeichnet. Das Supervised-Machine-Learning setzt sich aus der Klassifikation und der Regression zusammen. Die Klassifikation bezieht sich auf die Vorhersage kategorialer Ergebnisse (engl. categorical values), während sich die Regression auf die Vorhersage kontinuierlicher Ergebnisse (engl. continuous values) bezieht. Viele Machine-Learning-Algorithmen wurden so entwickelt, dass sie sowohl zur Durchführung von Klassifikation als auch zum Lösen von Regressionsproblemen geeignet sind. Supervised-Machine-Learning-Methoden umfassen in der Regel sowohl lineare und logistische Regression, Decision Tree oder Support Vector Machine (Alpaydın, 2014, pp. 5–10, 21–23; Bi *et al.*, 2019, p. 2223) .

Beim Unsupervised-Machine-Learning versucht der Algorithmus natürliche Beziehungen und Gruppierungen innerhalb der Daten ohne Bezugnahme auf irgendein Ergebnis zu identifizieren. Gängige Implementierungen für Unsupervised-Machine-Learning umfassen Clustering-Algorithmen, wie K-Means-, und Hierarchical-Clustering oder Gauß'sche Mischungsmodelle. Clustering-Algorithmen können Beobachtungen auf der Grundlage ähnlicher Datenmerkmale gruppieren (Alpaydın, 2014, pp. 11–12).

Semi-Supervised-Machine-Learning wird für Datensätze mit begrenzten Labeled Data und einer Menge von Un-Labeled Data eingesetzt, um die Modellperformance zu verbessern (Pedregosa *et al.*, 2011). Semi-Supervised-Machine-Learning ist eine Mischung aus Supervised-Machine-Learning und Unsupervised-Machine-Learning.

In einem Reinforcement Learning wird das Algorithmus mit der Umgebung durch eine Reihe von Aktionen im Lauf der Zeit interagieren. Diese Aktionen führen zu einer Art von Belohnung oder Bestrafung, womit sich das Algorithmus evaluieren und in die Zukunft verbessern kann. Allerdings muss das nicht unbedingt eine Belohnung oder Bestrafung sein, sondern kann dies jedes Feedback sein, das

für die Bestimmung zukünftiger Aktionen für das Algorithmus nützlich ist (Neapolitan and Jiang, 2018, p. 332). Mit anderem Wort lernt das Algorithmus in einer Reinforcement-Learning-Methode durch das sogenannte „Trails and Errors“ und versucht, verschiedene Optionen und Möglichkeiten zu erkunden, um das optimale Ergebnis herauszufinden.

2.2 Forecasting

Forecasting ist eine Methode, um fundierte Vorhersagen zu treffen, indem historische Daten als Eingabe für die Bestimmung des Verlaufs zukünftiger Trends verwendet werden. Unternehmen und Organisationen nutzen Prognosen für viele verschiedene Zwecke, beispielsweise um zukünftige Ausgaben zu prognostizieren und um zu bestimmen, wie sie ihr Budget zuweisen. Beispielsweise kann der Forecast der BEV-PKW-Neuzulassungen in Deutschland dazu beitragen, politische Entscheidungsträger bessere strategische Entscheidungen in Bezug auf Nachhaltigkeit zu treffen. Generell gibt es zwei Forecasting-Methoden: Qualitatives-, und quantitatives Forecasting (Tuovila, 2022).

Qualitative Forecasting-Methoden sind in hohem Maße auf Expertenmeinungen angewiesen und eignen sich am besten für kurzfristige Vorhersagen. Qualitative Forecasting ist geeignet, wenn keine historischen Daten zur Verfügung stehen. Beispiele für qualitatives Forecasting sind Interviews, Besuche vor Ort oder Marktforschung. Quantitative Forecasting-Methoden werden verwendet, wenn historische Daten schon existieren. Die meisten quantitativen Forecasting-Methoden verwenden Zeitreihendaten, die im Laufe der Zeit in regelmäßigen Abständen gesammelt werden. Die meistverwendete quantitative Forecasting-Methode ist Forecasting durch die Zeitreihenanalyse (Hyndman and Athanasopoulos, 2021, sec. 1.4).

Es gibt eine Reihe von Unterschieden zwischen Forecasting-, Regression- und Klassifikation-Verfahren. Auf die wichtigsten Unterschiede wird im Folgenden eingegangen. Der Hauptunterschied der drei Verfahren besteht darin, dass der Eingabedatensatz im Forecasting in zeitlichen Abständen (engl. time frequency) erfolgt. In der Regression und Klassifikation erfordert der Datensatz jedoch keine zeitliche Reihenfolge, weshalb die Daten vor dem Train- und Test-Split-Verfahren erst gemischt werden sollten (engl. shuffling dataset). Dies ist in der Zeitreihenanalyse nicht möglich, da sonst die Chronologie des

Eingabedatensatzes verloren geht (Hyndman and Athanasopoulos, 2021, sec. 5.4).

In einem Klassifikationsverfahren sollen die Zielvariablen kategorische Label haben. Ein Labelling ist beim Forecasting nicht notwendig, allerdings können die Labels als sogenannte exogenous Variable verwendet werden. Der letzte Unterschied ist die Modellevaluierungsmethode. In einer Klassifikation- und Regression-Methode werden versucht, die Genauigkeit des Modells (engl. model accuracy) zu erhöhen, im Gegensatz dazu spielt in der Zeitreihenanalyse der Modellfehler und dessen Minimieren eine große Rolle (Zhou, 2021, pp. 32–38).

2.3 Modellvergleich

In folgendem Abschnitt werden unterschiedliche Forecasting Modelle vorgestellt, die auf Machine-Learning basieren und die für die Vorhersage der Entwicklung der Elektromobilität in Deutschland genutzt werden können. Es wird die Theorie hinter den Modellen beschrieben und deren Nutzen für das Lösen der Problemstellung dieser Abschlussarbeit. Es werden zwei geeignete Modelle abhängig von den Merkmalen im Eingabedatensatz und den Modelleigenschaften ausgewählt. Diese Modelle werden dann technisch implementiert und im Kapitel 3 beschrieben. Dabei werden beispielsweise Modelltraining, Modellparameter, HPO, Modellevaluierung, Fehlermetriken und Prognose erläutert.

2.3.1 Autoregressive (AR) Model

Eine der Methoden der Zeitreihenanalyse ist das sogenannte Autoregressive Model (AR-Modell). Hierbei spielen die Variablenwerte der Vergangenheit, welche durch Beobachtungen gewonnen werden, eine wichtige Rolle. Das AR-Modell prognostiziert die Zielvariable durch die lineare Kombination von Variablen der Vergangenheit (Hyndman and Athanasopoulos, 2018, sec. 8.3). Beispielsweise wird die Vorhersage des Marktanteils von BEV-PKW-Neuzulassungen für den folgenden Monat durch die Werte von BEV-PKW-Neuzulassungen der vergangenen Monate getroffen. Das AR-Modell wird definiert als $AR(p)$, wobei die Variable p den Wert zu einem vergangenen Zeitpunkt beschreibt. Zum Beispiel verwendet $AR(1)$ den Wert vom letzten Monat und $AR(2)$ beschreibt den Wert der letzten zwei vergangenen Monate usw., um die Prognose für die Ziel-Variablen zu bestimmen. Die mathematische

Beschreibung für die AR(p)-Reihenfolge (engl. AR order) wird wie folgt dargestellt (Hyndman and Athanasopoulos, 2018, sec. 8.3):

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

In dieser Gleichung ist c eine Konstante und ε_t der Fehler im Modell (engl. noise) zum aktuellen Zeitpunkt t_0 . Die nächste Komponente y_{t-1} ist der letzte Wert der y-Variable, also gestern, letzte Woche usw. und y_{t-2} ist der Wert der Variable von vor zwei Zeitreihen, also vorgestern, vor zwei Wochen usw. Der letzte p-Wert der y-Variable ist y_{t-p} , wobei p im Modell als AR-Order gegeben wird. ϕ_1 , ϕ_2 , und ϕ_p sind die Koeffizienten, die mit der Ziel-Variable y multipliziert werden. Dementsprechend ist die mathematische Beschreibung für das AR(1)-Modell:

$$y_t = c + \phi_1 y_{t-1} + \varepsilon_t$$

Diese Regression zeigt, dass zur Prognose der BEV-PKW-Neuzulassungen der nächsten Monate ein konstanter Wert, ein Koeffizient multipliziert mit dem Wert der batterieelektrischen PKWs vom letzten Monat und Noise benötigt wird. Würde für ein bestimmtes Modell zum Beispiel AR(9) verwendet, wird die Regression bzw. das Modell dementsprechend kompliziert. So wird es auch zunehmend schwieriger die Gleichung zu schreiben und zu kalkulieren, wenn das Modell einen Datensatz mit Tausenden prognostizieren soll. Für ein AR(1) Modell beispielsweise muss der p-Wert zwischen 0 und 1 (mind. 0 und max. 1) und für ein AR(2) zwischen 0 und 2 (mind. 0 und max. 2) liegen (Hyndman and Athanasopoulos, 2021, sec. 9.3). Um eine richtige AR-Order für das Modell zu finden bzw. einzugeben, werden normalerweise PACF und ACF-Plots verwendet. Entsprechend der Lag-Orders von PACF und ACF wird es AR(1), AR(2), AR(3) usw. geschrieben.

Ein Merkmal des AR-Modells besteht darin, dass dies nur für stationäre Datensätze geeignet ist, weil es für nicht-stationäre Datensätze wegen der Saisonalität und/oder des Trends schlechte Ergebnisse bietet. Da der zugrundeliegende Datensatz nicht-stationäre Merkmale enthält, wird das AR-Modell in dieser Abschlussarbeit nicht ausgewählt.

2.3.2 Moving Average (MA) Model

Ein AR-Modell kann auf Basis der vergangenen Daten eine Prognose darstellen, Änderung in einem Modell werden jedoch nicht ausreichend miteinbezogen. Diese Änderungen können durch ein sogenannte Moving Average Model (MA-Modell) erfasst werden. Im Gegenteil zum AR-Modell, wobei die vergangenen

Variablenwerte der Regression verwendet werden, werden im sogenannte MA-Modell (MA-Modell) die vergangenen Fehler von Variablen in der Regression genutzt. Mathematisch wird das MA-Modell folgendermaßen dargestellt (Hyndman and Athanasopoulos, 2021, sec. 9.4):

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$$

In dieser linearen Regression sind die ε_t , ε_{t-1} , ε_{t-2} und ε_{t-q} Noise bzw. die Fehler der Regression zu den vergangenen Zeitpunkten und c ein konstanter Wert. Die nächsten Komponenten sind θ_1 , θ_2 , und θ_p , Koeffizienten genannte, die mit dem Noise der jeweiligen Zeitpunkte multipliziert werden. Diese Regression wird MA(q) Modell genannt, wobei MA das Moving Average Modell und q der Fehlerwert zu den vergangenen Zeitpunkten darstellt. „Natürlich beobachten wir die Werte von ε_t nicht, es handelt sich also nicht wirklich um eine Regression im üblichen Sinne“ (Hyndman and Athanasopoulos, 2021, sec. 9.4).

Um eine richtige MA-Order für das Modell zu finden, werden normalerweise, wie auch im AR-Modell, PACF und ACF-Plots verwendet. Entsprechend der Lag-Orders von PACF und ACF wird es MA(1), MA(2) usw. geschrieben. Allerdings gibt es ein Unterschied zwischen *Moving Average Modell (MA-Modell)* und *Moving Average Smoothing*. Während *Moving Average Modell* zum Forecasting verwendet wird, wird *Moving Average Smoothing* zur Einschätzung des Trend-Zyklus der vergangenen Werte im Datensatz genutzt.

Ein Merkmal des MA-Modells ist die Aufnahmen der Änderung oder Spitzen (engl. peaks) im Datensatz, der beispielsweise durch Markt- oder Preisänderung entsteht. Außerdem ist das MA-Modell nur für den stationären Datensatz geeignet. Da der Datensatz sowohl eine Saisonalität als auch einen Trend enthält, wird das MA-Modell ebenfalls nicht für die vorliegende Arbeit verwendet.

2.3.3 Autoregressive Moving Average (ARMA) Model

In vielen Fällen kann weder ein AR- noch ein MA-Modell ein gutes Forecasting-Ergebnis liefern, da die verschiedenen Datensätze sehr komplexere Merkmale enthalten. Eine Kombination von AR- und MA-Modell kann jedoch zu einer Verbesserung bei der Zeitreihenanalyse führen. Das sogenannte Autoregressive Moving Average Model (ARMA-Modell) enthält sowohl eine autoregressive (AR)- als auch eine Moving Average (MA) Komponente. Dementsprechend beinhaltet ein ARMA-Modell AR(p) und MA(q). Das ARMA-Modell kann wie folgt dargestellt werden (Zaiontz, 2022):

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$$

Verkürzt kann diese Regression folgendermaßen geschrieben werden:

$$y_t = c + \sum_{j=1}^p \phi_j y_{t-j} + \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j}$$

Wie in der Regression zu sehen ist, enthält ein ARMA-Modell zwei Bestandteile – eine AR- und eine MA-Komponente. Grundsätzlich wird diese als ARMA(p,q) geschrieben. Deswegen wird die Kombination von AR(1) und MA(1) auch als ARMA(1,1) bezeichnet. Der Datensatz in einem ARMA-Modell soll stationär sein (Brockwell and Davis, 2016, p. 74) und ist daher für einen nicht-stationären Datensatz nicht geeignet. In einem ARMA-Modell existiert kein integrierender Teil, der für das Abbilden von nicht-stationären Merkmalen notwendig ist. Dieses Problem kann durch das nachfolgende Modell gelöst werden.

2.3.4 Autoregressive Integrated Moving Average (ARIMA) Model

Datensätze haben verschiedene Merkmale. Dies gilt vor allem für nicht-stationäre Datensätze. Um eine Zeitreihenanalyse auf Grundlage solcher Datensätze implementieren zu können, müssen die Datensätze erst in eine stationäre Form umgewandelt werden (Brockwell and Davis, 2016, p. 157). Dies kann durch die „I“ Komponente im sogenannten Autoregressive Integrated Moving Average Model (ARIMA-Modell), die für Integration steht, erreicht werden. Integration des Datensatzes bedeutet Differenzierung (engl. differencing). Durch die Differenzierung eines nicht-stationären Datensatzes wird diese in eine stationäre Form konvertiert und ein neuer Datensatz erzeugt. Dieser kann zum Modelltraining sowie weitere Verfahren in der Zeitreihenanalyse herangezogen werden.

Der Autoregressive Integrated Moving Average (ARIMA) Prozess hat drei Komponenten, nämlich p , d und q . Die p Komponente steht für das AR-Modell und q für das MA-Modell. Durch d -Order-Differenzierung des ARMA(p,q)-Modells entsteht das ARIMA(p,d,q) Modell (Zaiontz, 2022). Ein ARIMA(1,0,1) ist gleich wie ein ARMA(1,1), weil die d -Komponente null ist, das heißt, dass kein Differenzierungs-Wert für Stationarität des Datensatzes eingegeben wurde. Dementsprechend ist ein ARIMA(1,0,0) das Gleiche wie ein AR(1)-Modell und ARIMA(0,0,1) das Gleiche wie ein MA(1)-Modell.

Differenzierung kann den Durchschnittswert (engl. mean value) von Zeitreihe stabilisieren und den Trend oder die Saisonalität eliminieren bzw. reduzieren (Hyndman and Athanasopoulos, 2018, sec. 8.1). Differenzierung der originellen Zeitreihe für die Erst-Reihe-Differenzierung (engl. first order differencing) entsteht durch Differenzierung des letzten Werts im Datensatz. First-Order-Differencing kann wie im Folgend beschrieben werden:

$$y'_t = y_t - y_{t-1}$$

In dieser Gleichung ist y'_t die differenzierte Zeitreihe und y_t ist die ursprüngliche Zeitreihe, während y_{t-1} zur letzten Zeitreihe im Datensatz verweist. Die differenzierte Zeitreihe oder y'_t wird Werte von T-1 haben (Hyndman and Athanasopoulos, 2018, sec. 8.1). In einfachen Worten wird y'_t dadurch entstehen, wenn die aktuelle Zeitreihe z.B. Zeitreihe des aktuellen Monats durch die letzte Zeitreihe z.B. Zeitreihe des letzten Monats abgerechnet wird. Manchmal ist der differenzierte Eingabedatensatz sogar nach der Differenzierung nicht in einer stationären Form. In diesen Fällen ist die First-Order-Differenzierung nicht ausreichend und es ist notwendig, der Datensatz ein zweites Mal zu differenzieren. Dieser Vorgang wird Second-Order-Differenzierung genannt und wird folgendermaßen geschrieben (Hyndman and Athanasopoulos, 2018, sec. 8.1):

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned}$$

In diesem Fall wird y''_t einen Wert von T-2 haben. Mit einfachen Worten werden die „Änderung der Änderungen“ von der Originaldaten (engl. changes of changes in time series) im Forecasting-Modell trainiert. Es ist abhängig von einzelnen Use-Cases und Datensätze, aber meistens ist eine Second-Order-Differenzierung schon ausreichend (Hyndman and Athanasopoulos, 2018, sec. 8.1).

Ein Merkmal des ARIMA-Modells ist der Umgang mit dem Trend (Brockwell and Davis, 2016, p. 158). Allerdings sollte der Datensatz zunächst in eine stationäre Form umgewandelt werden. Jedoch kann das ARIMA-Modell keine guten Ergebnisse für Datensätze mit Saisonalität ausgeben. Hierfür ist ein ARIMA-Modell mit einer Saisonalitäts-Komponente besser geeignet – nämlich das SARIMA-Modell.

2.3.5 Seasonal Autoregressive Integrated Moving Average (SARIMA) Model

Das sogenannte Seasonal Autoregressive Integrated Moving Average Model (SARIMA-Modell) ist eine Erweiterung des ARIMA-Modells und ermöglicht Forecasting auf Grundlage von nicht-stationären Datensätzen. Generell ist ein SARIMA-Modell ein Seasonal ARIMA(p,d,q)x(P,D,Q)_s (Brockwell and Davis, 2002, p. 203). SARIMA wird für Univariate-Zeitreihenanalysen verwendet, während SARIMAX bei Multivariate-Zeitreihenanalysen zum Einsatz kommt. SARIMA(X) wird im Kapitel 3 detailliert erklärt, denn es ist eines der zwei ausgewählten Modelle, die alle Merkmale des Eingabedatensatzes miteinbeziehen können.

2.3.6 ARCH- und GARCH Model

Unterliegt der Datensatz einer zeitlich veränderlichen Variabilität (engl. time varying variability) wird dies Volatilität in Zeitreihen genannt (engl. volatility in time series). Solche Zeitreihen finden ihre Anwendung häufig im Finanzwesen, zum Beispiel bei Zeitreihen für Aktienrenditen (Mills, 2019, p. 361). Diese Zeitreihen zeigen normalerweise eine Nicht-Linearität und weisen Abschnitte mit niedriger Volatilität gefolgt von Abschnitten mit höherer Volatilität auf. Um solche finanziellen Datensätze prognostizieren zu können, wird das sogenannte Autoregressive Conditional Heteroscedastic Model (ARCH-Modell) benutzt, weil hierbei diese Änderung miteinbezogen werden können. Das sogenannte Generalized Conditional Heteroscedastic Model (GARCH-Modell) ist eine Erweiterung des ARCH-Modells (Box *et al.*, 2015, p. 362). Die ARCH- und GARCH-Modelle wurden für diese Abschlussarbeit nicht ausgewählt, da sie für die Datensatzmerkmale dieser Abschlussarbeit nicht geeignet sind. Beispielsweise enthalten die Eingabedatensätze für die Univariate- und Multivariate-Zeitreihenanalyse keine Volatilität, um das ARCH- oder GARCH-Modell anzuwenden.

2.3.7 Vector Autoregression (VAR) Model

In einem sogenannten Vector Autoregression Model (VAR-Modell) wird die Kausalität zwischen den Variablen der Zeitreihenanalyse betrachtet. Univariate Modelle, vor allem das ARIMA basierte Modelle, haben die Einschränkung, dass sie eine unidirektionale Beziehung besitzen. Das bedeutet, dass die Prognosevariable von den Prädiktorvariablen beeinflusst wird, nicht aber

umgekehrt. (Hyndman and Athanasopoulos, 2021, sec. 12.3). Es gibt viele Fälle, bei denen sich die Variablen gegenseitig beeinflussen, weil Korrelationen zwischen den Variablen existieren. Die Maßnahmen während der Corona-Pandemie sind ein Beispiel hierfür: Im Jahr 2022 hat die deutsche Regierung finanzielle Unterstützung für Familien (*Finanzielle Unterstützung*, 2022) und Unternehmen sowie für Selbstständige gefördert (*Umfassende Corona-Hilfen für Unternehmen und Soloselbstständige*, 2022). In solchen Fällen können bidirektionale Beziehungen beobachtet werden – die finanzielle Unterstützung durch die Regierung wird die Kaufkraft der Bevölkerung sowie den Vertrieb der Unternehmen erhöhen und umgekehrt.

Ein VAR-Modell ist für solche Datensätze gut geeignet, da bei diesem alle Variablen gleichbehandelt werden. Mathematisch wird dies durch eine Gleichung pro Variable ausgedrückt (Hyndman and Athanasopoulos, 2021, sec. 12.3). Eine Erweiterung des VAR-Modells ist das sogenannte Vector Autoregression Moving Average Model (VARMA-Modell), das um eine Moving-Average-Komponente erweitert wurde. Sowohl im VAR-Modell als auch im VARMA-Modell sollte der Datensatz stationär sein (Chakraborty, 2020). Da VAR- und VARMA-Modelle vor allem für Multivariate-Zeitreihenanalysen eingesetzt werden (Tsay, 2015, pp. 27, 37, 105), wurden sie nicht für diese Abschlussarbeit ausgewählt.

2.3.8 ETS-Modell

Die sogenannten ETS-Models (ETS-Modelle) sind Univariate-Zeitreihenanalyse-Modelle, die aus den drei Komponenten bestehen: Fehler (engl. error), Trend und Saisonalität (engl. seasonality). Der Unterschied zwischen ARIMA-basierten Modellen und ETS-Modellen bzw. Exponential Smoothing besteht darin, dass aktuelle Beobachtungen wichtiger als ältere Beobachtungen sind. „Vorhersagen, die mit Exponential Smoothing erstellt wurden, sind gewichtete Durchschnitte vergangener Beobachtungen, wobei die Gewichte exponentiell abnehmen, wenn die Beobachtungen älter werden. Mit anderen Worten, je aktueller die Beobachtung, desto höher das damit verbundene Gewicht“ (Hyndman and Athanasopoulos, 2021, sec. 8).

Es gibt drei Methoden von Exponential Smoothing: Simple Exponential Smoothing wird für die Datensätze ohne Trend und Saisonalität verwendet. Double Exponential Smoothing wird für Datensätze mit einem Trend eingesetzt und als Holt's Linear Trend Method bezeichnet. Triple Exponential Smoothing ist für die Datensätze mit einer Saisonalitäts-Komponente geeignet und wird manchmal als Holt-Winters Exponential Smoothing bezeichnet. Diese

Bezeichnung des Tripple Exponential Smoothing wurde nach Charles Holt und Peter Winters benannt, die bei der Entwicklung der Methode mitwirkten (Brownlee, 2018; Hyndman and Athanasopoulos, 2021, secs 8.1, 8.2, 8.3).

Für die Holt-Winters Exponential Smoothing existieren zwei Methoden: Die additive und die multiplikative Methode. Die additive Methode wird bevorzugt, wenn die Saisonalität über die Zeitreihe hinweg relativ konstant ist, während die multiplikative Methode verwendet wird, wenn sich die Saisonalität proportional zum Niveau der Zeitreihe ändert. Das heißt: wenn die Zeitreihe einen linearen Trend hat, ist sie additiv, wenn sie exponentiell ansteigt, ist ein multiplikatives Modell besser geeignet (Pawar, 2020b; Hyndman and Athanasopoulos, 2021, sec. 8.3). Für den Datensatz dieser Abschlussarbeit würde sich die multiplikative Methode eignen, da der Eingabedatensatz einen exponentiellen Trend hat. Trotzdem wurde das ETS-Modell in dieser Masterarbeit nicht ausgewählt, da ETS-Modelle nur Univariate-Zeitreihenanalysen ermöglichen.

2.3.9 Prophet Model

Das sogenannte Prophet Model (Prophet) wurde durch das Data Science Team von Facebook als Open Source Library im Markt eingeführt. Das Prophet Zeitreihenanalyse-Modell wird mit drei Hauptkomponenten nämlich Trend, Saisonalität und Feiertage verwendet. Mathematisch wird das Modell wie folgt dargestellt (Taylor and Letham, 2017, pp. 5–7):

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

In dieser Gleichung ist $g(t)$ die Trend-Funktion, die nicht periodische Änderungen im Datensatz modelliert, und $s(t)$ stellt periodische Änderungen beispielsweise eine wöchentliche oder eine jährliche Saisonalität dar. Die nächste Komponente $h(t)$ zeigt die Auswirkungen von Feiertagen, die in potenziell unregelmäßigen Zeitplänen über einen oder mehrere Tage auftreten. Der letzte Teil der Gleichung ε_t zeigt den Modellfehler.

Prophet ist in Python- und R-Programmiersprache verfügbar und bietet verglichen mit den bisherigen Modellen eine Reihe von Vorteilen. Prophet kann hervorragend für Datensätze mit einer sehr starken Saisonalität oder mit stündlichen, täglichen-, oder wöchentlichen Beobachtungen mit einem Vorlauf von einigen Monaten oder einem Jahr eingesetzt werden. Darüber hinaus ist Prophet in der Lage Feiertage für den zukünftigen Forecast zu berücksichtigen. Prophet erleichtert den Umgang bei fehlenden Beobachtungen (engl. missing

values), bei großen Ausreißern (engl. outliers) sowie bei historischen Trendänderungen, beispielsweise auf Grund von Produkteinführungen oder Trends, bei denen es sich um nichtlineare Wachstumskurven handelt. Hauptsächlich wurde Prophet entwickelt, um einfach, schnell und automatisch zu sein. Dies Modell ist vor allem für die interne Verwendung in einem Unternehmen beispielsweise bei der Vertriebs- und Kapazitätsvorhersage zweckmäßig (Brownlee, 2020b). Obwohl Prophet für den in dieser Abschlussarbeit verwendeten Datensatz geeignet ist, wurde es zugunsten besser geeigneter Modelle nicht gewählt.

2.3.10 Recurrent Neural Network (RNN) Model

Im Gegenteil zu den ARIMA-basierten Modellen ist Recurrent Neural Network (RNN) ein nicht-lineares Forecasting-Modell (Runkler, 2020, p. 89). Neuronale Netzwerke können in unterschiedlichen Bereichen zum Einsatz kommen. Sie eignen sich vor allem für große Datensätze. Ein neuronales Netzwerk besteht aus einer Reihe miteinander verbundener Knoten, die als Neuronen bezeichnet werden und in der Regel in Schichten (engl. layer) angeordnet sind. Ein einfaches neuronales Netzwerk besteht aus mindestens einem Eingabe-Layer, einem Hidden-Layer und einem Ausgabe-Layer (Lewis, 2016, p. 17).

Neuronale Netze können in dynamisch und statisch eingeteilt werden. Statische neuronale Netzwerke sind beispielsweise Feed Forward Netzwerk, indem die Ausgabe (engl. output) direkt durch einen Input-Layer berechnet wird. Dynamische neuronale Netzwerke sind beispielsweise RNN. In einem dynamischen neuronalen Netzwerk hängt die Ausgabe von der aktuellen Eingabe in das Netzwerk und den vorherigen Eingaben, Ausgaben und/oder verborgenen Zuständen (engl. hidden state) des Netzwerks ab (Lewis, 2016, p. 68). Da RNNs sogenannte Hidden States enthalten, sind sie in der Lage, effizient mehrere Informationen über die Vergangenheit zu speichern. Die nicht-lineare Dynamik ermöglicht es den RNNs, ihren Hidden State auf komplizierte Weise zu aktualisieren (Lewis, 2016, p. 67).

RNNs sind geeignet für sequenzielle Dateien wie Audio-Dateien oder historische Datensätze. RNNs werden heutzutage in mehreren Bereichen erfolgreich eingesetzt, insbesondere beim Forecasting oder Natural Language Processing (NLP) z.B. Spracherkennung. Obwohl RNN für diese Abschlussarbeit sehr gut geeignet ist, wurde DeepAR ausgewählt, da dieses im Hintergrund RNN verwendet.

2.3.11 DeepAR Model

DeepAR ist ein Zeitreihenanalyse-Modell aus der Gluon-Time-Series (GluonTS) Library, das auf Apache MXNet basiert. GluonTS wurde durch Amazon Data Scientist entwickelt und Ende 2019 als Open-Source Library zur Verfügung gestellt (Gasthaus and Januschowski, 2019). Im Hintergrund nutzt DeepAR RNN und ist ein relativ neues Forecasting-Modell. Darüber hinaus unterstützt DeepAR die Vorhersage aller Merkmale des in dieser Abschlussarbeit verwendenden Eingabedatensatzes. Deswegen wurde neben SARIMA(X) auch DeepAR als zweites Forecasting-Modell ausgewählt. Weitere Details zu SARIMA(X) und DeepAR folgen im Kapitel 3.

Alle der obengenannten Forecasting-Modelle mit ihren Eigenschaften stehen zu Vergleichszwecken in der Tabelle 16 im Anhang zur Verfügung.

2.4 Fehlermetriken

Zur Evaluierung der Güte eines Forecast-Modells müssen die Zeitreihenanalyse-Modelle durch ein Fehlermaß ausgewertet werden. Fehlermetriken ermöglichen es die Güte eines Modells zu quantifizieren, zu berechnen oder verschiedene Modelle miteinander zu vergleichen. In den anderen Klassifikations- und Regression-Verfahren in Supervised-Machine-Learning z.B. Random-Forest- oder Logistic-Regression-Klassifikation soll die Genauigkeit des Modells (engl. accuracy) optimiert und erhöht werden. Daher gilt: Je größer die Accuracy, desto höher wird die Güte des Modells betrachtet. Die Accuracy eines Modells ist in der Regel zwischen 0 und 1 (0–100%), wobei 0% das schlechteste und 100% das beste Modell beweist. Allerdings wird in einer Zeitreihenanalyse der Modellfehler (engl. model error) und dessen Minimieren im Hauptfokus stehen. Deswegen ist ein Modell mit kleinem Fehler besser als ein Modell mit größerem Fehler.

Fehlermetriken können in verschiedene Kategorien eingeteilt werden. Hyndman und Athanasopoulos haben die Forecast-Fehler in vier Kategorien eingeteilt (Hyndman and Athanasopoulos, 2021), während Gilliland, Tashman und Sglavo die Fehlermetriken in drei Kategorien eingeteilt haben (Gilliland, Tashman and Sglavo, 2015, p. 151), nämlich grundlegende Metriken in den ursprünglichen Einheiten der Daten, grundlegende Metriken in Prozentform und Metriken für relative Fehler. Im Folgenden werden einige der wichtigsten und am häufigsten verwendeten Fehlermetriken für Zeitreihenanalyse beschrieben, allerdings ohne Kategorisierung. Es werden mathematische Formeln zur Berechnung und

Beschreibung sowie deren Eigenschaften erfasst und abschließend bewertet, ob sich die jeweilige Fehlermetrik für diese Abschlussarbeit eignet.

2.4.1 Mean Absolute Error (MAE)

Das sogenannte Mean Absolute Error (MAE), auch als mittlerer absoluter Fehler bekannt, ist ein Fehlermetrik, der den Durchschnitt zwischen den prognostizierten Werten (engl. predicted values) und den tatsächlichen Werten (engl. observed values) darstellt. Mathematisch wird es folgendermaßen ausgedrückt (Shcherbakov *et al.*, 2013, pp. 171, 172):

$$MAE = \frac{1}{n} \sum_{t=1}^{t=n} |y' - y|$$

In dieser Gleichung ist y' das predicted value und y das observed value, während n die Gesamtzahl der Werte in der Testmenge (engl. testset) darstellt. MAE zeigt die Größe des Fehlers, welchen man im Durchschnitt vom Forecast-Modell erwarten kann. Obwohl MAE eine der meistverwendeten Fehlermetriken in der Zeitreihenanalyse ist, besitzt diese auch einige Nachteile. Es ist mit MAE schwierig große Fehler von kleinen Fehlern zu differenzieren oder Probleme mit geringem Datenvolumen zu überdecken. (*Fehlermaße: Wie Sie die Güte Ihrer Forecasts auswerten*, 2022). Obwohl MAE einfach zu messen und zu verstehen ist, ist es nicht für Datensätze mit Extremwerten geeignet. Diese Metrik wurde zur Evaluierung der Forecast-Modelle in dieser Abschlussarbeit nicht ausgewählt, weil der Eingabedatensatz auch Extremwerte enthält.

2.4.2 Mean Absolute Percent Error (MAPE)

Eine andere Fehlermetrik zur Evaluierung des Forecasting-Modells ist das sogenannte Mean Absolute Percent Error (MAPE), das auch als mittlerer absoluter prozentualer Fehler genannt wird. MAPE ist der prozentuale Mittelwert der absoluten Differenz zwischen predicted Values und observed Values, geteilt durch observed Values. MAPE ist mathematisch folgendermaßen dargestellt (Shcherbakov *et al.*, 2013, p. 172):

$$MAPE = \frac{1}{n} \sum_{t=1}^{t=n} \frac{|y' - y|}{y} * 100$$

Auch in dieser Gleichung ist y' das predicted value und y das observed value, während n die Gesamtzahl der Werte im Testset darstellt. Da y im Nenner steht, eignen sich nur Daten ohne Null- und Extremwerte. Das Modell wird als optimal betrachtet, wenn der MAPE-Wert niedriger ist. MAPE ist nicht für die Datensätze mit Extremwerten geeignet, selbst wenn es leicht verständlich ist. Deswegen wurde MAPE für die Evaluierung der Modelle in dieser Abschlussarbeit nicht ausgewählt. Dies Problem mit den Extremwerten kann durch der nächste Fehlermetrik Mean Squared Error (MSE) gelöst werden.

2.4.3 Mean Squared Error (MSE)

Der sogenannte Mean Squared Error (MSE), der auch mittlerer quadratischer Fehler genannt wird, ist ebenfalls eine weitverbreitete Fehlermetrik, mit der die Güte von Zeitreihenanalyse-Modellen ausgewertet werden können. MSE ist der erwartete quadratische Abstand zwischen predicted value und observed value und wird wie folgt beschrieben (*Fehlermaße: Wie Sie die Güte Ihrer Forecasts auswerten*, 2022):

$$MSE = \frac{1}{n} \sum_{t=1}^{t=n} (y' - y)^2$$

In dieser Gleichung ist y' das predicted value und y das observed value, während n die Gesamtzahl der Werte im Testset darstellt. Ein Vorteil der MSE-Metrik besteht darin, dass große Fehler oder Ausreißer mehr als kleine Fehler berücksichtigt oder bestraft werden. Wie bei anderen Zeitreihenanalyse-Metriken ist ein geringerer MSE-Wert besser als ein größerer MSE-Wert. In einfachen Worten, je näher der MSE-Wert an Null liegt, desto optimaler ist das Zeitreihenanalyse-Modell. MSE ist schwer zu interpretieren, da die Werte während der Berechnung quadriert werden müssen (Gilliland, Tashman and Sglavo, 2015, p. 254). Aus diesem Grund wurde MSE zur Evaluierung der Zeitreihenanalyse in dieser Abschlussarbeit nicht ausgewählt.

2.4.4 Root Mean Squared Error (RMSE)

Der sogenannte Root Mean Squared Error (RMSE), auch als Wurzel des mittleren quadratischen Fehlers bekannt, ist eine Erweiterung von MSE und einer der bekanntesten Fehlermetriken zur Modellevaluierung der Zeitreihenanalyse. RMSE ist die Quadratwurzel des MSE. Die mathematische Formel für RMSE wird

wie folgt geschrieben (*Fehlermaße: Wie Sie die Güte Ihrer Forecasts auswerten*, 2022):

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{t=n} (y' - y)^2}$$

In der obigen RMSE-Gleichung ist y' das predicted value und y das observed value, während n die Gesamtzahl der Werte im Testset darstellt. Wie bei MSE werden auch bei RMSE die großen Fehler oder Ausreißer stärker berücksichtigt. Allerdings ist RMSE leichter zu verstehen als MSE, weil der RMSE-Wert in der gleichen Einheit wie der prognostizierte Wert ist. Eine Erweiterung von RMSE ist NRMSE, die durch Normalisierung von RMSE entsteht. Eine Normalisierung von RMSE kann entweder durch die Verwendung des Mittelwerts oder der Differenz von Minimal- und Maximalwert berechnet werden. NRMSE wird verwendet, um mehrere Datensätze oder Forecast-Modelle miteinander zu vergleichen, wobei auch hier kleine Datenpunkte (engl. data points) übersehen werden können (Shcherbakov *et al.*, 2013, pp. 172–174; Hyndman and Athanasopoulos, 2018, sec. 3.4, 2021, sec. 5.8).

Es ist schwierig, eine einzige Fehlermetrik auszuwählen, weil alle Fehlermetriken ihre Vor- und Nachteile haben. Die Auswahl einer Fehlermetrik ist abhängig von jeweiligem Use-Case (Hyndman and Koehler, 2005, p. 16) und soll entsprechend der Datensatzmerkmale und des Forecast-Modells selektiert werden. In dieser Abschlussarbeit wurde wegen der folgenden Gründe der RMSE zur Evaluierung der Forecast-Modelle gewählt:

Der erste Grund zur Auswahl der RMSE-Metrik ist die Interpretation. RMSE verglichen zu MSE ist leichter zu interpretieren, da der RMSE-Wert die gleiche Skala hat, wie die prognostizierten Werte (*Fehlermaße: Wie Sie die Güte Ihrer Forecasts auswerten*, 2022). In dieser Abschlussarbeit werden die zukünftigen sechs Monate prognostiziert. Wenn der RMSE-Wert 2,3 ist, bedeutet dies, dass die Werte in den prognostizierten sechs Monate durchschnittlich 2,3 Einheiten in der gleichen Skala abweichen, wie im Eingabedatensatz. Der Begriff „durchschnittlich“ ist beim Verstehen des RMSE-Werts wichtig. Beispielsweise könnte es sein, dass der Forecast im ersten oder zweiten Monat genau stimmt, aber in den anderen Monaten nicht. Allerdings berechnet RMSE den durchschnittlichen Wert aller prognostizierten sechs Monate.

RMSE ist ebenfalls nützlich, wenn der Datensatz Extremwerte, (Hyndman and Koehler, 2005, p. 8) z.B. eine starke Saisonalität besitzt. Jedoch ist RMSE

gegenüber Ausreißern empfindlich. Der Eingabedatensatz, der in dieser Abschlussarbeit verwendet wurde, besitzt jedoch keine Ausreißer bzw. wurden die Ausreißer schon vor dem Modelltraining behandelt. Allerdings hat der Eingabedatensatz Extremwerte vor allem im letzten Quartal des Jahres. Aus diesem Grund ist RMSE eine ideale Fehlermetrik zur Evaluierung der Univariate- und Multivariate-Zeitreihenanalyse.

Darüber hinaus wird die Verwendung der RMSE-Fehlermetrik empfohlen, wenn der Modellfehler eine normale Verteilung hat (Chai and Draxler, 2014, p. 1249). Im Kapitel 4 wird dargestellt, dass der Modellfehler im SARIMA(X)-Modell eine normal Verteilung besitzt. Da am Ende der Abschlussarbeit alle Modelle miteinander verglichen werden sollten, ist die Auswahl einer einheitlichen Fehlermetrik wesentlich. Sowohl SARIMA(X) und DeepAR unterstützen zur Evaluierung RMSE-Fehlermetrik reibungslos.

Als letzte Begründung zur Auswahl von RMSE ist die Wichtigkeit des Forecasts oder der prognostizierten Punkte. Beispielsweise empfiehlt AWS-Forecast die Verwendung von RMSE-Metrik, wenn einige große falsche Vorhersagen eines Modells sehr kostspielig Folgen hätte (Das, Robinson and Mankala, 2020). Der Eingabedatensatz weist eine starke Saisonalität, insbesondere im Oktober, November und Dezember jeweiligen Jahres auf. Dies ist ein wesentliches Kriterium für die richtige Auswahl des Modells. Durch die Verwendung der RMSE-Metrik kann festgestellt werden, ob das ausgewählte Modell die Saisonalität in den oben genannten Monaten miteinbezogen hat.

3 Methodologie

In diesem Kapitel wird die Methodologie zum Forecasting der BEV-PKW-Neuzulassungen in Deutschland erklärt. Es wird zunächst die technische Ausstattung und Codestruktur sowie die explorative Datenanalyse des Eingabedatensatzes beschrieben. Abschließend werden die Univariate- und Multivariate-Zeitreihenanalyse sowie deren Hyperparameteroptimierung erläutert.

3.1 Technische Ausstattung und Codestruktur

Die Zeitreihenanalyse wurde auf einem MacBook Pro mit dem Intel Core i7 Prozessor (2019) und einem MacBook Pro mit dem M1-Chip (2021) berechnet. Als Programmiersprache wurde Python genutzt. Zur Implementierung des SARIMA-Modells wurde die Version 0.13.2 des Statsmodels-Library (Seabold and Perktold, 2010) und für das DeepAR-Modell die Version 0.10.7 des GlounTS-Library (Gasthaus and Januschowski, 2019) verwendet.

Die Struktur der Codes zur Zeitreihenanalyse beginnt mit dem Importieren der Libraries und dem Öffnen des Datensatzes sowie der explorativen Datenanalyse, z.B. Stationaritätstest, PACF- und ACF-Plots, ETS-Dekomposition, Feature Engineering und Data Scaling. Feature-Engineering und Standard-Scaling hatten in dem Fall keinen Einfluss auf die weitere Verbesserung der Forecasting-Modelle, deswegen wurden sie in den Python-Codes auskommentiert. Darauf folgt der Code zum Train-Test-Split, Modelltraining, zur Modellevaluierung und Prognose zur Verfügung. Abschließend erfolgt die Speicherung des endgültigen Forecasting-Modells im pickle- und JSON-Format, damit es später einfacher wiederverwendet oder es in der Cloud bereitgestellt (engl. deploy) werden kann. Der Code zur Univariate- und Multivariate-Zeitreihenanalyse steht in GitHub unter dem Link <https://github.com/ghulamzada/masterthesis-time-series-analysis> zur Verfügung.

3.2 Explorative Datenanalyse

Explorative Datenanalyse sowie Datenvorbereitung (engl. data preprocessing) sind der erste Schritt in einem Machine-Learning-Verfahren, weil die Machine-Learning-Modelle mit den unbereinigten Eingabedatensätzen nicht umgehen können. Aus diesem Grund ist es wesentlich, dass der Eingabedatensatz zuerst bereinigt und dementsprechend weitere Merkmale des Datensatzes analysiert werden.

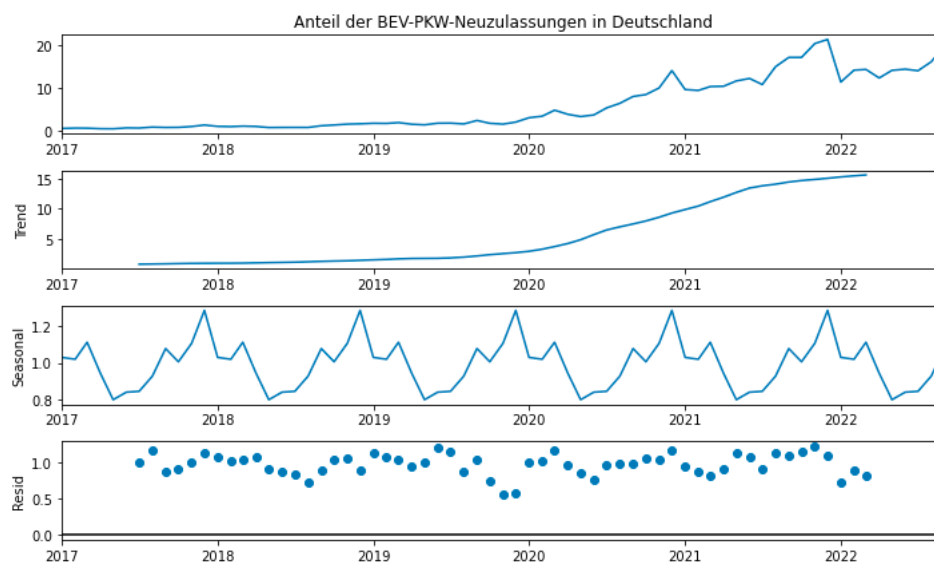
Der univariate Eingabedatensatz, der dieser Masterarbeit zum Forecasting zur Verfügung gestellt wurde, benötigte keine Bereinigung, z.B. gab es keine fehlenden Werte (engl. missing values), kategoriale Werte (engl. categorical values) oder Ausreißer (engl. outliers). Allerdings musste der multivariate Eingabedatensatz bereinigt werden, damit sich die Genauigkeit der Prognose der BEV-PKW-Neuzulassungen in Deutschland verbessert.

Generell können die Datensätze entweder stationär (engl. stationary) oder nicht-stationär (engl. non-stationary) sein. Ein Datensatz muss einen konstanten Mittelwert und eine konstante Varianz enthalten, damit er stationärer Datensatz genannt werden kann (Montgomery, Jennings and Kulahci, 2015, pp. 35–36; Pawar, 2020a). Mit anderen Worten ist eine Zeitreihe stationär, wenn der Mittelwert und die Varianz zwischen zwei beliebigen äquidistanten Punkten festgelegt sind. Datensätze mit einem Trend, einer Saisonalität oder mit beiden Merkmalen sind nicht-stationär. Um Datensätze nach Stationarität (engl. stationarity) oder Nicht-Stationarität (engl. non-stationarity) zu untersuchen, wurde der sogenannte Augmented-Dicky-Fuller-Test (ADF_{Fuller}-Test) durchgeführt. Der ADF_{Fuller}-Test ist ein statistischer Test für Stationarität. Wenn der p-Wert im ADF_{Fuller}-Test-Ergebnis kleiner als 0,05 (5%) ist, ist die Zeitreihe als stationär einzustufen, andernfalls als nicht-stationär (Pawar, 2020a).

3.2.1 Explorative Analyse der univariaten Daten

Der Eingabedatensatz zur Univariate-Zeitreihenanalyse ist der prozentuale Marktanteil der BEV-PKW-Neuzulassung in Deutschland, wie in der Abbildung 1 dargestellt. Nach der Durchführung des ADF_{Fuller}-Tests wurde festgestellt, dass der univariate Datensatz nicht-stationär ist. Um die Merkmale und Eigenschaften des univariaten Datensatzes noch genauer zu untersuchen, wurde die sogenannte ETS-Dekomposition, auch als STL-Dekomposition bekannt, durchgeführt. Die ETS-Dekomposition wird den Datensatz nach Trend, Saisonalität und Restwert (engl. residual) analysieren und visualisieren.

Es gibt zwei ETS-Dekompositions-Methoden: Die additive Dekomposition und die multiplikative Dekomposition. Die additive Dekomposition ist geeignet, wenn die Stärke der saisonalen Schwankungen nicht mit der Höhe der Zeitreihe variiert. Eine multiplikative Dekomposition ist angemessener, wenn die Variation des saisonalen Musters proportional zum Niveau der Zeitreihe ist (Hyndman and Athanasopoulos, 2021, sec. 3.2). Eine multiplikative Dekomposition wurde aufgrund der saisonalen Faktoren im univariaten Datensatz implementiert. Nach der Implementierung der ETS-Dekomposition wurde festgestellt, dass der Eingabedatensatz alle drei Merkmale enthält, nämlich Saisonalität, Trend und Residual.



Quelle: Eigene Darstellung, basierend auf Kraftfahrt-Bundesamt, 2022

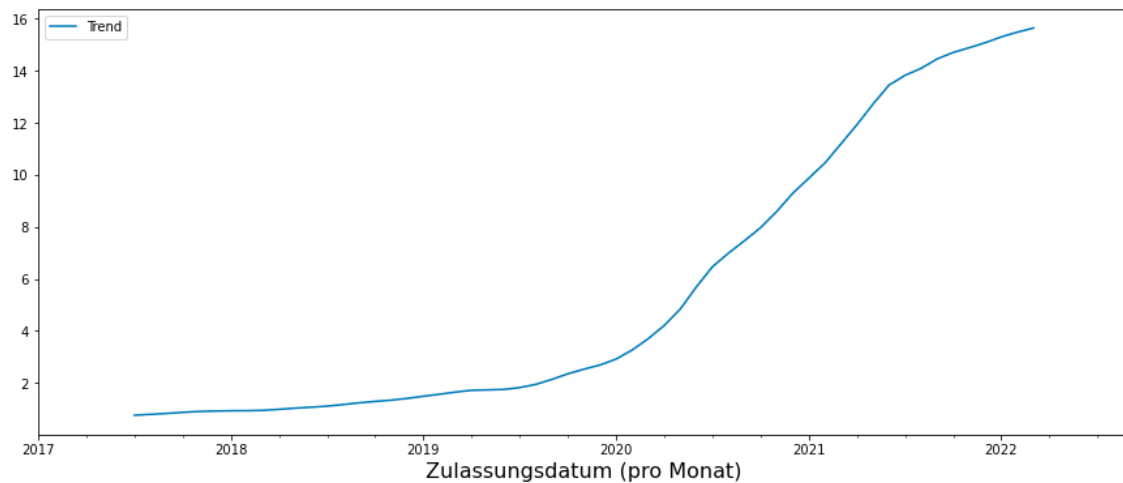
Abbildung 3: ETS-Dekomposition des univariaten Datensatzes

Wie in der Abbildung 3 dargestellt, zeigt die ETS-Dekomposition vier Visualisierungen. Die erste Darstellung zeigt den originalen Datensatz (engl. observed values), während die zweite und dritte Visualisierung jeweils den Trend und die Saisonalität (engl. seasonality) aufweisen. Grundsätzlich wird durch die ETS-Dekomposition der Trend und die Saisonalität aus dem Datensatz dargestellt. Alles, was nicht als Trend oder Saisonalität dargestellt wurde, wird als Residual in der vierten Visualisierung gezeigt. Diese drei Merkmale des univariaten Datensatzes werden im Folgenden erläutert.

3.2.1.1 Trend

Ein Trend in der Zeitreihenanalyse liegt vor, wenn sich der Datensatz im Laufe der Zeit exponentiell oder in Form einer Kurve entwickelt bzw. die Daten langfristig steigen (engl. upward) oder fallen (engl. downward) (Hyndman and

Athanasopoulos, 2021, sec. 2.3). Die ETS-Dekomposition analysiert den Eingabedatensatz und gibt den Trend wieder. Wie es in der Abbildung 4 zu sehen ist, handelt es sich um einen nicht-linearen Trend.



Quelle: Eigene Darstellung, basierend auf Kraftfahrt-Bundesamt, 2022

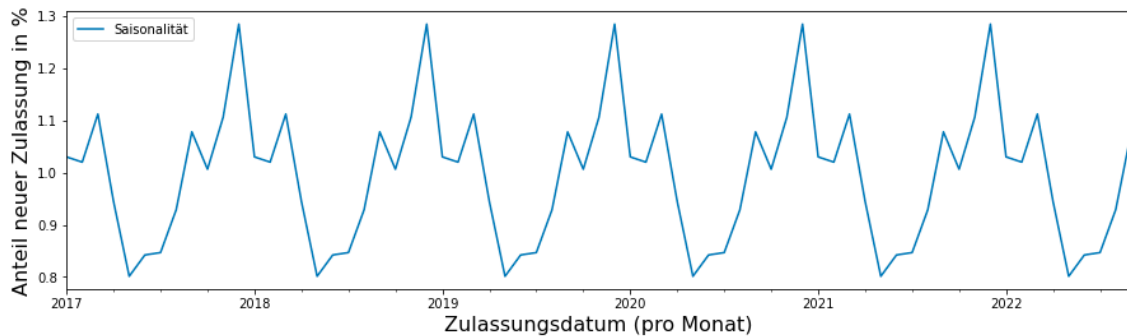
Abbildung 4: Trend des univariaten Datensatzes

Die Entwicklung des prozentualen Marktanteils der BEV-PKW-Neuzulassungen zwischen Januar 2017 bis Ende 2019 ist sehr langsam bzw. es gibt kaum erkennbaren Trend. Allerdings existiert ab Januar 2020 bis September 2022 ein starker Trend nach oben. Einer der Gründe des steigenden Trends im Datensatz ab 2020 ist die deutliche Förderung in Deutschland ab 2020; beim Kauf eines neuen BEV-PKWs wurden ab diesem Zeitpunkt sowohl Steuervorteile wie Steuerfreiheit als auch Kaufanreize in Form von Geldprämien gewährt (*Overview – Electric vehicles: tax benefits & purchase incentives in the European Union, 2022*). Darüber hinaus gewinnt der Flottengrenzwert zunehmend an Bedeutung. Größtenteils durch politische Maßnahmen hervorgerufen, versuchen die Hersteller den Marktanteil der Elektrofahrzeuge deutlich zu erhöhen, um den Flottengrenzwerte einzuhalten (Gnann *et al.*, 2022, pp. 6–7). Allerdings ist umstritten, ob diese von der EU festgelegten Flottengrenzwerte effektiv genug sind (*Daimler und BMW nutzen Schlupflöcher, um ohnehin schwache EU-Flottengrenzwerte für Neuwagen einzuhalten, 2021*).

3.2.1.2 Saisonalität

Die Saisonalität im Datensatz ist die regelmäßige Schwankung zwischen bestimmten Zeitpunkten. Eine Saisonalität tritt auf, wenn eine Zeitreihe von saisonalen Faktoren wie der Jahres- oder Monatszeit sowie dem Wochentag beeinflusst wird. Die Saisonalität hängt immer von einem festen und bekannten

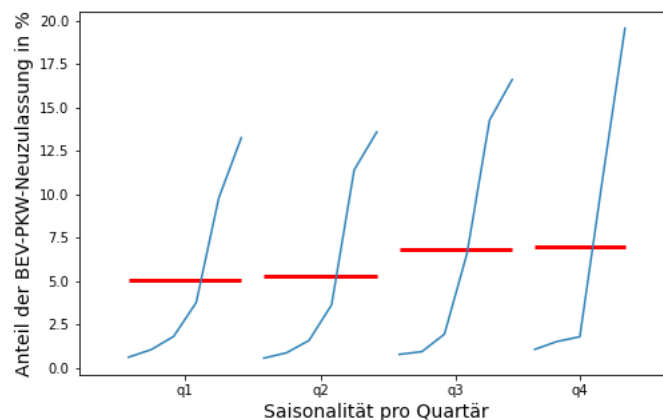
Zeitraum ab (Hyndman and Athanasopoulos, 2021, sec. 2.3). In der Abbildung 5 kann die Saisonalität des univariaten Datensatzes festgestellt werden.



Quelle: Eigene Darstellung, basierend auf Krafftahrt-Bundesamt, 2022

Abbildung 5: Saisonalität des univariaten Datensatzes

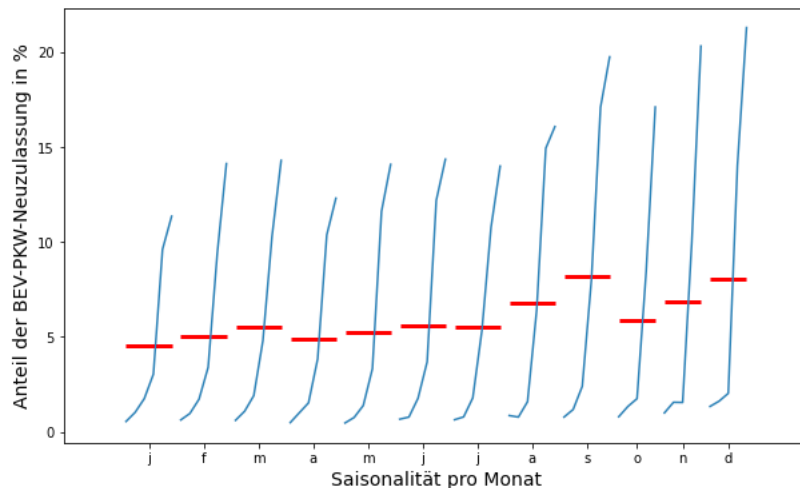
In der Abbildung 5 ist zu sehen, dass eine starke saisonale Ausprägung vorhanden ist. Zum Beispiel steigt am Ende des jeweiligen Jahres der Anteil der neuen BEV-PKW-Zulassungen an. Zur besseren Darstellung der Saisonalität in den Quartalen bzw. Monaten wurden die Abbildung 6 und Abbildung 7 mit Hilfe der Statsmodels-Library (Seabold and Perktold, 2010) erstellt.



Quelle: Eigene Darstellung, basierend auf Krafftahrt-Bundesamt, 2022

Abbildung 6: Saisonalität des univariaten Datensatzes pro Quartal

In der Abbildung 6 stellen die blauen Linien den sortierten prozentualen Marktanteil der BEV-PKW-Neuzulassungen pro Quartal dar, während die roten Linien den Mittelwert der BEV-PKW-Neuzulassungen pro Quartal zeigen. Dadurch wird nochmal verdeutlicht, dass im letzten Quartal (Q4) der Anteil BEV-Neuzulassungen ansteigt. Die folgende Visualisierung hilft dabei, den univariaten Eingabedatensatz genauer zu untersuchen und die monatliche Saisonalität darzustellen.



Quelle: Eigene Darstellung, basierend auf Kraffahrt-Bundesamt, 2022
Abbildung 7: Monatliche Saisonalität des univariaten Datensatzes

Auch in der Abbildung 7 stellen die schwarzen Linien den prozentualen Marktanteil der BEV-PKW-Neuzulassungen pro Monat dar, während die roten Linien den Mittelwert der BEV-PKW-Neuzulassungen pro Monat zeigen. Die Abbildung 7 zeigt, dass die Anzahl neuer Zulassungen in den letzten Monaten des Jahres, also Oktober, November und Dezember, höher als in den anderen Monaten sind. Jedoch ist der Anteil der prozentuale Marktanteil der neuen BEV-PKW-Zulassungen im Juli geringer als in den Restmonaten. Die Abbildung 7 ist eine Bestätigung für die Abbildung 5 und Abbildung 6, die jeweils die Saisonalität allgemein und die Saisonalität am Ende jeweiligen Jahres visualisierten. Zwar gibt es derzeit keine wissenschaftlichen Quellen, aber es ist zu beobachten, dass eine Saisonalität der BEV-PKW-Neuzulassungen am Ende des Jahres bzw. in den letzten drei Monaten des Jahres existiert. Für die beiden starken Jahresendzulassungswellen in den Jahren 2020 und 2021 kann es folgende Gründe geben:

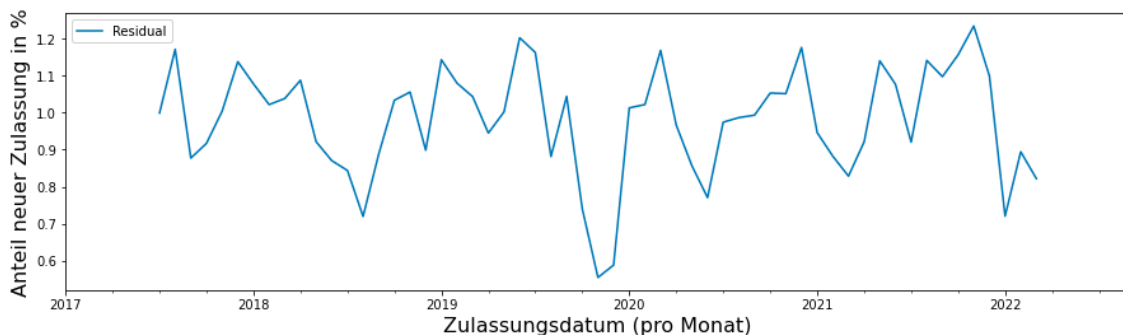
Ab dem Jahr 2020 gilt das System der Flottengrenzwerte für PKW ('Das System der CO₂-Flottengrenzwerte für Pkw und leichte Nutzfahrzeuge', 2020). Das hat mutmaßlich dazu geführt, dass viele Hersteller zum Ende des Jahres 2020 noch sehr viele BEVs in den Markt gebracht haben. Greenpeace hatte unter anderem Volksagen vorgeworfen, durch Eigenzulassungen im Jahr 2020 ca. 140 Millionen Euro EU-Strafe vermieden zu haben (Hage and Traufetter, 2021). Außerdem wurde im Juli 2020 die Innovationsprämie für BEVs verdoppelt. Mit dieser Verdopplung des staatlichen Anteils am Umweltbonus können für Elektrofahrzeuge bis zu 9.000 Euro als Fördersumme beantragt werden. Das hat vermutlich dazu geführt, dass viele mit den Zulassungen auf die zweite

Jahreshälfte gewartet haben, um noch mehr Förderung für Ihr BEV zu erhalten (Leichsenring, 2021).

Darüber hinaus war die ursprüngliche Innovationsprämie für BEVs auf das Ende 2021 begrenzt und sollte eigentlich verlängert werden. Diese Verlängerung wurde jedoch durch die Bundestagswahlen und die Koalitionsverhandlungen hinausgezögert, so dass bis Ende 2021 unklar war, ob es 2022 noch Förderung für BEVs geben wird. Vermutlich sind deshalb Ende 2021 noch sehr zahlreiche Zulassungen auf den Markt gekommen, um die Förderung zu nutzen (*Bundesregierung passt Förderrichtlinie für Elektrofahrzeuge an*, 2021).

3.2.1.3 Residual

Die ETS-Dekomposition gibt den Trend und die Saisonalität wieder. Allerdings wird alles in der letzten Visualisierung als Residual gezeigt, was nicht im Trend und in der Saisonalität dargestellt wird. Das Residual zeigt keine Muster von Trend oder Saisonalität. Das durch ETS-Dekomposition analysierte Residual lässt sich folgendermaßen darstellen:



Quelle: Eigene Darstellung, basierend auf Kraftfahrt-Bundesamt, 2022

Abbildung 8: Residual des univariaten Datensatzes

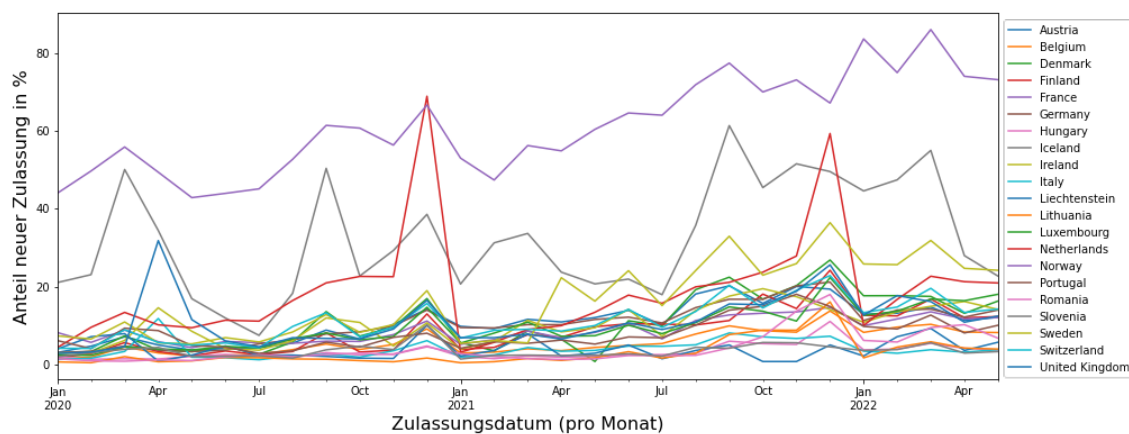
Das Residual des univariaten Eingabedatensatzes zeigt keine Muster von einem Trend oder einer Saisonalität, weil das Residual alles ist, was nach dem Trend und der Saisonalität übrigbleibt. Aber das ist bei der Zeitreihenanalyse vom Residual zu erwarten und ist ein positives Zeichen.

3.2.2 Explorative Analyse der multivariaten Daten

Der Eingabedatensatz zur Multivariate-Zeitreihenanalyse besteht aus 33 Variablen, wie in Abbildung 2 gezeigt wurde. Diese Variablen sind der prozentuale Marktanteil der BEV-PKW-Neuzulassungen in 33 Ländern, also in allen 27 EU-Ländern sowie Island, Liechtenstein, Norwegen, der Schweiz, der

Türkei und dem Vereinigten Königreich. Die Zielvariable hierbei ist weiterhin der prozentuale Marktanteil der BEV-PKW-Neuzulassungen in Deutschland, während die anderen 32 Variablen exogene Variablen sind. Die exogenen Variablen sollen dazu beitragen, den Forecast der BEV-PKW-Neuzulassungen in Deutschland genauer zu prognostizieren.

Die Datensatz-Merkmale des univariaten Datensatzes, z.B. Trend und Saisonalität sind auch in den anderen Ländern bzw. im multivariaten Datensatz zu beobachten. Da die Zielvariable der Multivariate-Zeitreihenanalyse dieselbe wie bei der Univariate-Zeitreihenanalyse ist, bleiben die Datensatzmerkmale, beispielsweise Nicht-Stationarität, Trend und Saisonalität, auch identisch. Allerdings sollten die Ausreißer im multivariaten Datensatz behandelt werden. „Ein Ausreißer ist ein Datenwert, der ungewöhnlich groß oder klein ist, der deutlich über oder unter dem liegt, was für einen bestimmten Zeitraum erwartet wurde“ (Gilliland, Tashman and Sglavo, 2015, pp. 95–98). Die Ausreißer führen nicht nur zu einem ungenauen Forecasting-Ergebnis, sondern können auch für die RMSE-Fehlermetrik problematisch sein. RMSE ist gegenüber einem Ausreißer sehr empfindlich, wie im zweiten Kapitel erläutert, daher sollte dieser im Datensatz vor dem Modelltraining behandelt werden. Der bereinigte multivariate Eingabedatensatz sieht folgendermaßen aus:



Quelle: Eigene Darstellung, basierend auf European Alternative Fuels Observatory, 2022

Abbildung 9: Bereinigter Eingabedatensatz zur Multivariate-Zeitreihenanalyse

Abbildung 9 zeigt den multivariaten Datensatz nach der Bereinigung. Er beinhaltet im Gegensatz zum unbereinigten Datensatz nur 21 Länder. Ein Ausreißer existierte im Dezember 2021 in der Irland-Variable, wie in der Abbildung 2 schon dargestellt wurde. Gilliland et al. haben eine passende Methode vorgeschlagen, wobei die Ausreißer im Datensatz vor dem Modelltraining mit den ähnlichen Werten zu korrigieren (Gilliland, Tashman and Sglavo, 2015, pp. 95–98). Der Ausreißer im multivariaten Datensatz, also in der

Irland-Variable, wurde wegen der Saisonalität im letzten Quartal des Jahres mit dem Durchschnittswert der Werten im November 2021 und Januar 2022 ersetzt. Darüber hinaus wurden die Variablen gelöscht, die zum Modelltraining und zur Prognose mutmaßlich keine Verbesserung beizutragen haben. Diese Variablen waren die 12 Ländern, deren Durchschnittswerte geringer als der kleinste Wert der BEV-PKW-Neuzulassungen in Deutschland waren. Der kleinste Werte in der Zeitreihe der BEV-PKW-Neuzulassungen in Deutschland beträgt 2,98, daher wurden alle 12 Länder mit einem Durchschnittswert kleiner als 2,98 aus der Berechnung herausgenommen. Infolgedessen steht der multivariate Eingabedatensatz mit 21 Variablen und 29 Zeitpunkten zur Verfügung, die zum Modelltraining und im weiteren Verfahren verwendet werden.

Der multivariate Datensatz wird zur Multivariate-Zeitreihenanalyse genutzt, um einen sechsmonatigen Forecast zu erstellen. Der univariate Eingabedatensatz, der zur Univariate-Zeitreihenanalyse vorbereitet wurde, wird im Folgenden beschrieben.

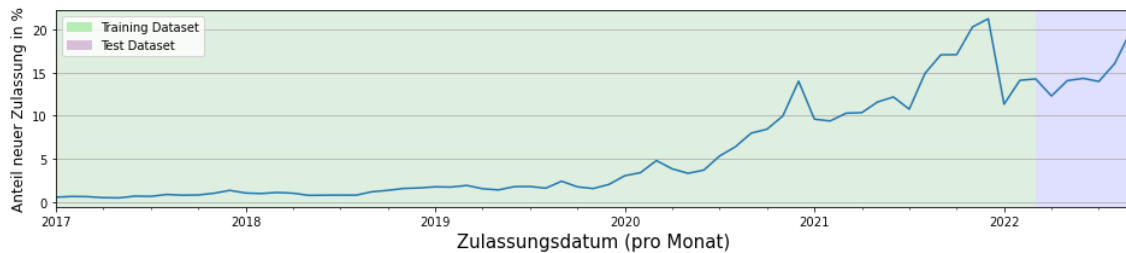
3.3 Univariate Zeitreihenanalyse

Es wurden entsprechend der Merkmale des Eingabedatensatzes sowie der Modelleigenschaften zwei Modelle zur Zeitreihenanalyse im zweiten Kapitel ausgewählt. Vor dem Modelltraining sollte der Eingabedatensatz zum Zweck des sogenannten Train-Test-Splits aufgeteilt werden. Der Train-Test-Split und die zwei Modelle, also SARIMA(X) und DeepAR-Modell, werden im Folgenden detailliert erläutert.

3.3.1 Train-Test-Split

Der Train-Test-Split ist generell für die Machine-Learning-Modelle wesentlich, um ihre Güte auszuwerten. Durch den Train-Test-Split wird der Datensatz in einen Trainingsdatensatz (engl. training-dataset) und einen Testdatensatz (engl. test-dataset) aufgeteilt. Während der Training-Dataset zum Modelltraining und Feststellen der Modellparameter genutzt wird, wird der Testdatensatz zur Auswertung der Güte des Modells bzw. zur Evaluierung der Modellgenauigkeit (engl. model accuracy) verwendet. Der Umfang des Testdatensatzes beträgt typischerweise etwa 20% des Eingabedatensatzes, jedoch hängt dieser Wert davon ab, wie groß der Datensatz ist und wie weit das Modell prognostizieren soll. Der Testdatensatz sollte idealerweise mindestens so groß sein wie der

maximal benötigte Prognosehorizont (Hyndman and Athanasopoulos, 2021, sec. 5.8). Der Train-Test-Split des univariaten Datensatzes, der prozentuale Marktanteil der BEV-PKW-Neuzulassungen lässt sich wie folgt darstellen:



Quelle: Eigene Darstellung, basierend auf Kraftfahrt-Bundesamt, 2022

Abbildung 10: Train-Test-Split des univariaten Datensatzes

Abbildung 10 zeigt die Training- und Testdatensätze, die mit grün und blau im Hintergrund markierten sind. Die ersten 63 Monate wurden als Trainingsdatensatz (vom Januar 2017 bis März 2022) und die letzten 6 Monate als Testdatensatz (vom April 2022 bis September 2022) ausgewählt. Der Testdatensatz entspricht ca. 8% des gesamten Datensatzes. Es gibt folgende Gründe zur Auswahl eines kleinen Testdatensatzes:

Der erste Grund ist die Größe des gesamten Eingabedatensatzes. Die Univariat-Zeitreihenanalyse umfasst insgesamt 69 Monate, was einer vergleichsweise geringen Datenmenge entspricht. Eine solche Situation kommt beim Forecasting vermehrt vor, wenn ein neues Produkt im Markt eingeführt wird und deswegen wenige bzw. keine historischen Daten für eine Prognose existieren. Da das Modell selbst zum Training historische Daten benötigt, sollten ein Großteil der Datensätze zum Training eingegeben werden, sonst erstellt das Modell wegen mangelnden Daten falsche Prognosen.

Ein zweiter Grund zur Auswahl des Testdatensatzes von sechs Monaten ist die praktische Prognoseanforderung im Umfang von sechs Monaten. Das endgültige Ziel der Univariate-Forecasting-Modelle ist das Forecasting der zukünftigen sechs Monate, da eine längere Prognose nicht seriös erscheint. Hyndman and Athanasopoulos (Hyndman and Athanasopoulos, 2021, sec. 5.8) empfehlen, dass der Testdatensatz mindestens so groß wie der Prognosehorizont sein soll. Da der zukünftige Prognosehorizont sechs Monate beträgt, wurde der Testdatensatz ebenfalls für sechs Monate gewählt.

Im Folgenden werden die nächsten Schritte nach dem Train-Test-Split-Verfahren erläutert: Das Modelltraining, die Modellevaluierung und die Hyperparameteroptimierung mit SARIMA und DeepAR zur Univariat-Zeitreihenanalyse.

3.3.2 SARIMA

Das SARIMA-Modell ist eine Erweiterung des ARIMA-Modells, jedoch mit saisonalen Komponenten. Das ARIMA-Modell wurde schon im zweiten Kapitel beschrieben. Generell kann ein SARIMA-Modell wie folgt geschrieben werden (Seabold and Perktold, 2010):

$$SARIMA = (p, d, q) \times (P, D, Q)_m$$

Es gibt zwei Bestandteile im SARIMA-Modell: Einen nicht-saisonalen (engl. non-seasonal) und einen saisonalen Teil des Modells. Die Komponenten mit den Kleinbuchstaben p, d und q bezeichnen den nicht-saisonalen Teil des Modells, während die Komponenten mit den Großbuchstaben P, D und Q den saisonalen Teil des SARIMA-Modells darstellen und m die saisonale Periode (engl. seasonal period) aufweist, z.B. Anzahl der Beobachtungen pro Monat, (Hyndman and Athanasopoulos, 2021, sec. 9.9). Die Komponenten im nicht-saisonalen Teil des SARIMA-Modells p, d und q entsprechen jeweils p für das AR-Modell, q für MA-Modell und d für Differenzierung des nicht-stationären Datensatzes. Die AR- und MA-Modelle sowie Differenzierung wurden schon im zweiten Kapitel erläutert. Um das SARIMA-Modell zu trainieren, wurden folgende Parameter verwendet.

Parameter	Werte
endog	df_train
order	3, 2, 1
seasonal_order	1, 0, 0, 12
freq	'MS'
n_jobs	-1

Tabelle 1: Parameter des SARIMA-Modelltrainings

Es wurden insgesamt fünf Parameter zum Training des SARIMA-Modells verwendet, wie es in der Tabelle 1 dargestellt ist. Der sogenannte *endog* Parameter ist die Zielvariable, die im univariaten Datensatz die BEV-PKW-Neuzulassungen in Deutschland ist. In diesem Fall wurde zum Zweck des Modelltrainings der Trainingsdatensatz (*df_train*) eingegeben. Der zweite Parameter ist *order*, der die Komponenten mit den Kleinbuchstaben p, d und q umfasst, also das AR-Modell, die Differenzierung und das MA-Modell zum nicht-saisonalen Teil des SARIMA-Modells. Die Werte 3, 2 und 1 an sich bedeuten, dass die p -, d - und q -Komponente jeweils 3, 2 und 1. Oder das AR und die Differenzierung sind jeweils 3 und 2, allerdings ist das MA 1, um den nicht-saisonalen Teil des Modells zu berechnen.

Das SARIMA-Modell kann die Saisonalität im Datensatz miteinbeziehen und dies wird durch den *seasonal_order* Parameter kalkuliert. Der Parameter *seasonal_order* ist der saisonale Teil, der die Komponenten mit den Großbuchstaben *P*, *D* und *Q* und die *m*-Komponente beinhaltet. Die *P*-, *D*-, und *Q*-Komponenten sind jeweils 1, 0 und 0 und die *m*-Komponente ist 12. Mit anderen Worten ist das AR-Modell 1, allerdings die Differenzierung und das MA-Modell jeweils 0. Zur Berechnung und Miteinbeziehung der Saisonalität der BEV-PKW-Neuzulassungen im SARIMA-Modell wird im Hintergrund die *m*-Komponente mit den *P*-, *D*- und *Q*-Komponenten multipliziert. Beispielsweise wird die *m*-Komponente mit der *P*-Komponente ($12 \times 1 = 12$) und der *D*-Komponente ($12 \times 0 = 0$) sowie der *Q*-Komponente ($12 \times 0 = 0$) multipliziert. Das bedeutet, dass das SARIMA-Modell jeweils die letzten 12 Werten im Datensatz für das AR-Modell, aber keinen Wert für die Differenzierung und das MA-Modell im Datensatz verwendet, um den saisonalen Teil des SARIMA-Modells zu berechnen. Mit anderen Worten werden zur Berechnung der Saisonalität der BEV-PKW-Neuzulassungen jeweils die Werte der letzten 12 Monate für *P*-Komponente, aber keinen Wert für *Q*- und *D*-Komponente verwendet.

Die letzten beiden Parameter sind *freq* und *n_jobs*. Der Parameter *freq* ist die Häufigkeit (engl. data frequency) der Datensätze. Da der Eingabedatensatz der BEV-PKW-Neuzulassungen monatlich ist, wird der Wert 'MS' eingegeben, das „pro Monat“ bedeutet. Beispielsweise wird für den täglichen, wöchentlichen, oder jährlichen Datensätze jeweils 'D', 'W' oder 'A' eingegeben. Das Parameter *n_jobs* ist für die Anzahl der CPU-Cores zuständig, die beim Modelltraining verwendet werden sollte. Der Wert -1 bedeutet, dass alle zur Verfügung gestellten CPU-Cores genutzt werden müssen, wodurch sich die Modelltrainingszeit reduziert.

Es wurde für das Modelltraining der Trainingsdatensatz und zur Modellevaluierung der Testdatensatz genutzt. Die ersten 63 Monate wurden zum Modelltraining und die letzten 6 Monate zur Modellevaluierung verwendet. Die RMSE-Fehlermetrik wurde zur Auswertung der Güte des Modells ausgewählt, die schon im zweiten Kapitel erläutert wurde.

Generell können die Werte für den nicht-saisonalen Teil des SARIMA-Modells *p*, *d* und *q* entsprechend den ACF- und PACF-Plots und manuelle Differenzierung festgestellt werden. Das ist jedoch beim Forecasting mit dem SARIMA-Modell nicht effektiv genug. Einerseits ist es schwierig, einen richtigen Wert für die *p*, *d* und *q*-Komponenten durch die ACF- und PACF-Plots einzuschätzen. Andererseits enthält SARIMA nicht nur nicht-saisonalen Teil (*p*, *d* und *q*-Komponente), sondern auch saisonalen Teil (*P*, *D* und *Q*-

Komponenten). Aus diesem Grund ist der Einsatz der HPO essenziell, um die richtigen Werte für den sowohl saisonalen als auch nicht-saisonalen Teil des SARIMA-Modells festzustellen.

Zur Hyperparameteroptimierung der SARIMA-Modellparametern wurden drei Methoden verwendet: Die erste Methode ist das sogenannte Grid-Search, das durch eine For-Schleife (engl. for loop) in Python implementiert wurde. Die zweite Methode ist das sogenannte Auto-ARIMA, das durch die PMDARIMA-Library (Smith, 2017) ausgeführt wurde. Die dritte Methode ist das sogenannte Optuna, das an sich eine externe Library ist und mehrere Möglichkeiten bietet die HPO-Ergebnisse einzusehen und dementsprechend eine Entscheidung bezüglich der HPO zu treffen (Akiba *et al.*, 2019). Folgende Parameter wurden zur Optimierung der Univariate-Zeitreihenanalyse zum SARIMA-Modell ausgewählt, weil diese die wesentlichen Komponenten des saisonalen und nicht-saisonalen Teil des SARIMA-Modells enthalten.

Parameter	Min. Wert	Max. Wert
p	0	4
d	0	4
q	0	4
P	0	4
D	0	4
Q	0	4

Tabelle 2: Parameter des SARIMA-Modells zur HPO

Wie in der Tabelle 2 dargestellt ist, sind die p , d und q -Parameter zum nicht-saisonalen Teil und die P , D und Q -Komponente für den saisonalen Teil zuständig. Pro Parameter wurde für den Minimalwert 0 und für den Maximalwert inklusive 4 eingegeben. Das bedeutet, dass während der HPO für den jeweiligen Parameter ein optimaler Wert zwischen 0 und 4 gefunden werden soll, wodurch sich das Modell bestmöglich trainieren kann. Die HPO zum SARIMA-Modell der Univariate-Zeitreihenanalyse wird auf Seite 41 detailliert erklärt. Zunächst wird im Folgenden das DeepAR-Modell der Univariate-Zeitreihenanalyse erläutert.

3.3.3 DeepAR

Salinas, Flunkert and Gasthaus (Salinas, Flunkert and Gasthaus, 2017) haben das sogenannte Deep Autoregressive Neural Network (DeepAR) Modell entwickelt, das ein Deep-Learning-Zeitreihenanalyse-Modell aus der GlounTS-Library ist (Gasthaus and Januschowski, 2019). DeepAR erzeugt ein globales

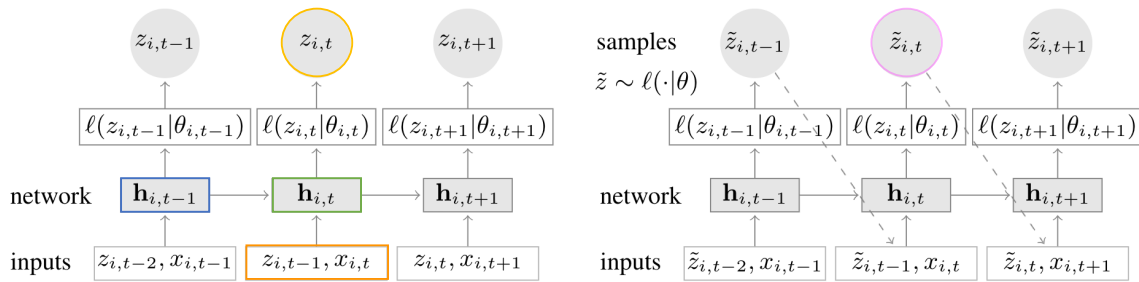
Modell aus allen im Datensatz befindlichen historischen Daten, statt ein Modell pro Zeitreihe zu erstellen (Salinas, Flunkert and Gasthaus, 2017, p. 1). Beispielsweise erstellt DeepAR für den multivariaten Datensatz mit 21 Variablen ein Modell, wobei eine Prognose pro Zeitreihe dargestellt wird. Mit dem SARIMA oder anderen klassischen Modellen müsste zur Prognose pro Variable 21 separate Modelle erstellt werden. Darüber hinaus kann DeepAR Daten aus verschiedenen Distributionen, z.B. Binomial, Bernoulli, Student-T oder Gaussian, verarbeiten. Mit anderen Worten verwendet DeepAR die Distributionen der Quantile, die in der Literatur als Quantilsregressionen bezeichnet werden (Tsay, 2005, pp. 288–301). Allerdings kommt dieser Unterschied nur bei einem multivariaten Datensatz vor. DeepAR und SARIMA erstellt beispielsweise beim univariaten Datensatz jeweils nur ein Modell, weil der univariate Datensatz nur eine Variable enthält. DeepAR passt eine LSTM-basierte RNN-Architektur an und ist ein probabilistisches Forecasting-Modell. Das probabilistische Forecasting schätzt die Wahrscheinlichkeitsverteilung der Zukunft in einer Zeitreihe angesichts ihrer Vergangenheit.

DeepAR hat folgende Vorteile verglichen mit den anderen klassischen Forecasting-Modellen. Als Erstes benötigt DeepAR minimales manuelles Feature Engineering, weil DeepAR die Saisonalität und andere zugrundeliegenden Abhängigkeiten und Kovariaten (engl. covariates) in der verwendeten Zeitreihe während des Modelltrainings erlernt. Als Zweites kann das DeepAR-Modell wegen seiner globalen Eigenschaft die Zusammenhänge zwischen den Variablen extrahieren, um die Prognosen mit hoher Genauigkeit zu erstellen (Salinas, Flunkert and Gasthaus, 2017, p. 2).

Im DeepAR-Modell wird die bedingte Verteilung der Zeitreihe i zum Zeitpunkt t durch $Z_{i,t}$ bezeichnet.

$$P(Z_{i,t_0:T} \mid Z_{i,1:t_0-1}, X_{i,1:T})$$

In dieser mathematischen Formel stellt T das Ende des Prognosehorizonts dar und die Komponenten $Z_{i,1:t_0-1}$ sind die vergangenen Werte. Die Komponente t_0 ist der Beginn des Vorhersagebereichs, während die Komponenten $X_{i,1:T}$ die Kovariaten sind. Zusammenfassend kann die obige Formel folgendermaßen beschrieben werden: Die früheren kontinuierlichen Werte $Z_{i,1:t_0-1}$ und die zugehörigen Kovariaten $X_{i,1:T}$ existieren bereits. Dementsprechend ist die bedingte Wahrscheinlichkeitsdichteverteilung (engl. probability density distribution) gewünscht, die die Wahrscheinlichkeit zur Beobachtung der zukünftigen Werte $Z_{i,t_0:T}$ maximiert.



Quelle: Salinas, Flunkert and Gasthaus, 2017, p. 3)

Abbildung 11: Zusammenfassung des DeepAR Modells

Abbildung 11 zeigt die Zusammenfassung des DeepAR-Modells. Der linke Teil der Abbildung 11 zeigt die Trainingsphase, auch als Encoder bezeichnet (Hou *et al.*, 2021, pp. 269–272). In der Trainingsphase empfängt das Modell die Kovariaten $X_{i,t}$ und den Zielwert im vorherigen Zeitschritt (engl. target values) $Z_{i,t-1}$ (markiert mit der orangen Farbe) sowie die vorherige Netzwerkausgabe $h_{i,t-1}$ (markiert mit der blauen Farbe). Zudem ist $t < t_0$, und gibt das sogenannte Hidden State $h_{i,t}$ (markiert mit der grünen Farbe) aus. Das Hidden State wird verwendet, um die sogenannte Loss Function und die Modellparameter θ anzupassen. Der Verlauf der Zeitreihe (engl. time series history) $Z_{i,t}$ (markiert mit der gelben Farbe) wird zur einstufigen Prognose $\hat{Z}_{i,t+1}$ verwendet. Der rechte Teil der Abbildung 11 zeigt die Forecasting-Phase, auch als Decoder bekannt (Hou *et al.*, 2021, pp. 269–272). DeepAR empfängt Informationen früherer Werte durch das Hidden State und gibt die Parameter einer Distribution aus. Hier wird eine Stichprobe $\hat{Z}_{i,t}$ (markiert mit der Magenta-Farbe) gezogen und zum nächsten LSTM-Layer im Netzwerk weitergeleitet, bis das Ende des Prognosehorizonts erreicht ist (Salinas, Flunkert and Gasthaus, 2017, pp. 3–4). Zum DeepAR-Modelltraining wurden folgende Parameter verwendet:

Parameter	Werte
freq	"M"
context_length	46
prediction_length	6
num_layers	7
num_cells	50
cell_type	"Istm"
dropout_rate	0.07783495658767649
epoch	42
batch_size	12
num_batches_per_epoch	20
learning_rate	0.05331129708197498

Tabelle 3: Parameter des Univariate-DeepAR-Modelltrainings

Der erste Parameter ist der sogenannte *freq*, der das sogenannte Data Frequency der Zeitreihe definiert. Im trainierten DeepAR-Modell wurde *freq* als "M" bezeichnet, dass für den monatlichen Datensatz steht. Im Gegensatz zum SARIMA-Modell unterstützt DeepAR einen oder mehreren stündlichen sowie einen oder mehreren minütigen Datensatz. Beispielsweise wird für einen Datensatz mit zweistündigen Daten als "2h" angegeben. Der zweite Parameter ist der sogenannte *context_length*, der für die Anzahl der zu verwendenden Datenreihe im RNN-Netzwerk zum Modelltraining zuständig ist. Der dritte Parameter *prediction_length* ist für den Prognosehorizont des Modells verantwortlich. Hierdurch wird im Modell befohlen, wie viele Zeitpunkte vom trainierten DeepAR-Modell prognostiziert werden. Nachdem das DeepAR-Modell trainiert wurde, kann das Modell nur so viele Zeitpunkte vorhersagen, die im Parameter *prediction_length* angegeben wurden. Wird einen längeren Prognosehorizont als im Parameter *prediction_length* benötigt, müsste ein neues DeepAR-Modell trainiert werden (*DeepAR Hyperparameters*, 2022). Die Parameter *num_layers* und *num_cell* zeigen jeweils die Anzahl der sogenannten Hidden-Layers und Cells in RNN-Architektur auf.

Der Parameter *cell_type* definiert die Art der RNN-Zelle, wobei der Werte "*lstm*" angegeben wird. Das sogenannte Long Term Short Memory (LSTM) Cell ermöglicht dem RNN-Netzwerk die vergangenen Werten zu erinnern und dementsprechend eine hochwertige Prognose zu treffen. Der nächste Parameter *dropout_rate* ist ein Regularisierungsparameter und wird zur Vermeidung des sogenannten Overfitting verwendet. Der Parameter *dropout_rate* trägt dazu bei, eine kleine Portion der Koeffizient-Gewichte während des Modeltrainings zufällig im Netzwerk herauszunehmen und dadurch das Overfitting zu vermeiden. Der Parameter *epoch* stellt die Anzahl der Epochen dar, die das Netzwerk trainiert, während der Parameter *batch_size* die Größe eines Batches oder die Anzahl der Beobachtung in einem Batch zeigt. Der Parameter *num_batches_per_epoch* definiert die maximale Anzahl des sogenannten Batches pro Epochen. Der letzte Parameter *learning_rate* definiert den Wert, womit sich das DeepAR seine internen Parameter anpasst. Ist der Parameter *learning_rate* sehr gering, wird das Modell längere Zeit zum Modelltraining benötigen und umgekehrt.

Beim erneuten DeepAR-Modelltraining ist es möglich, einen leicht abweichenden Wert zu bekommen, obwohl das DeepAR-Modell mit den gleichen Parametern trainiert wird. Der Grund dafür besteht darin, dass die Optimierung des DeepAR-Modells mit Zufall durchgeführt wird. Um dieses Problem zu vermeiden, wurde das sogenannte Seed bei dem DeepAR-Modelltraining, der Modellevaluierung und der HPO verwendet, damit die Modellausgabewerte stabil bleiben.

Zur Modellevaluierung im DeepAR-Modell wurde der Testdatensatz genutzt, während der Trainingsdatensatz zum Modelltraining verwendet. Die ersten 63 Monate wurden zum Modelltraining und die letzten 6 Monate zur Modellevaluierung verwendet. Um das DeepAR-Modell zu evaluieren, wurde die RMSE-Fehlermetrik verwendet, die Güte des Modells auszuwerten. Um eine richtige Kombination der obengenannten Parameterwerten in Deep Learning Algorithmen wie DeepAR zu finden, ist eine sehr zeitaufwendige Prozess. Aus diesem Grund ist der Einsatz der HPO wesentlich.

Zur Hyperparameteroptimierung (HPO) der DeepAR-Modellparameter wurden zwei Methoden verwendet: Grid-Search und Optuna. Folgende Parameter wurden zur Optimierung der Univariate-Zeitreihenanalyse zum DeepAR-Modell ausgewählt:

Parameter	Min. Wert	Max. Wert
num_layers	2	30
num_cells	5	100
dropout_rate	0.0	0.2
epochs	5	50
batch_size	5	30
num_batches_per_epoch	5	30
learning_rate	0.001	0.1
Context_length	2	63

Tabelle 4: Parameter des Univariate-DeepAR-Modells zur HPO

Wie in der Tabelle 4 dargestellt ist, wurde pro Parameter Minimalwert und Maximalwert eingegeben. Das bedeutet, dass während der HPO für den jeweiligen Parameter einen optimalen Wert zwischen dem gegebenen minimalen und maximalen Wert gefunden werden soll, wodurch sich das Modell bestmöglich trainieren lässt. Die HPO zum DeepAR-Modell der Univariate-Zeitreihenanalyse wird im Folgenden erläutert.

3.3.4 Hyperparameteroptimierung (HPO)

In diesem Abschnitt wird die Hyperparameteroptimierung (HPO), auch als Hyperparameter Optimization bzw. Hyperparameter Tuning bekannt, für das SARIMA- und DeepAR-Modell erklärt. „HPO ist der Prozess, die richtige Kombination von Hyperparameterwerten zu finden, um in angemessener Zeit die maximale Leistung für die Daten zu erzielen“ (David, 2020). Die einfachste Form einer HPO ist Grid-Search.

In einem Grid-Search werden grundsätzlich die Kombinationen aller möglichen Parameter untersucht. Infolgedessen werden diese Modelle entsprechend der niedrigsten RMSE-Werte in einer aufsteigenden Reihenfolge aufgelistet. Da Grid-Search die Kombination aller eingegebenen Parameter untersucht, braucht Grid-Search-Prozess viel Zeit und CPU-Ressourcen. Für das SARIMA-Modell benötigte Grid-Search ohne Parallelisierung auf der oben genannten Hardware beispielsweise 69 Stunden und für das DeepAR-Modell der Univariate-Zeitreihenanalyse sogar 10,5 Tage. Um die Trainingszeit des DeepAR-Modells beim Grid-Search zu verringern, mussten die Anzahl der Parameter eingeschränkt werden. Die Tabelle 5 stellt die eingeschränkten Parameterwerte des DeepAR-Modells zum Grid-Search dar:

Parameter	Werte
num_layers	2,5,8,10,15,20,22,25,27,30
num_cells	2,3,4,5,10,15,20,30,40
epochs	5,10,15,20,25
batch_size	5,10,20,30
num_batches_per_epoch	5,10,15,20,25,30

Tabelle 5: Eingeschränkte Parameterwerte des Univariate-DeepAR-Modells

Die eingeschränkten DeepAR-Parameter haben nur einige vordefinierte Werte. Diese wurden entsprechend der Einleitung und Vorschläge des AWS-Hyperparameter-Tunings (*Tune a DeepAR Model*, 2022; *DeepAR Hyperparameters*, 2022) und des sogenannten Trial and Error-Prozesses eingegeben. Die ersten 20 Grid-Search-Ausgabe der SARIMA- und DeepAR-Modelle zur Univariate-Zeitreihenanalyse sind im Anhang in der Tabelle 17 und Tabelle 18 verfügbar, wobei eine Zeile das Modellergebnis für eine bestimmte Hyperparameter-Kombination darstellt. Sequenzielle HPO-Methoden, wie beispielsweise Optuna (Akiba et al., 2019) und PMDARIMA (Smith, 2017), bieten eine effizientere Art und Weise, die Hyperparameter zu optimieren. Im folgenden Abschnitt werden die HPO-Methoden Auto-ARIMA und Optuna erläutert.

3.3.4.1 Auto-ARIMA

Auto-ARIMA ist eine HPO-methode, die durch eine externe Library PMDARIMA (Smith, 2017) zur Verfügung steht. Auto-ARIMA funktioniert ähnlich wie Grid-Search, indem es alle Modelle mit den nicht-saisonalen (p,d,q) und saisonalen (P,D,Q) Komponenten untersucht und entsprechend der sogenannten AIC-, AICC- oder BIC-Kriterien das Modell mit dem niedrigsten gegebenen Fehlerwert ausgibt. Diese als Informationskriterien bezeichneten Werte dienen unter

anderem zur Auswertung der Modellqualität. Als Beispiel soll an dieser Stelle der AIC-Wert näher erläutert werden. Dabei gilt: Je kleiner der AIC-Wert, desto optimaler wird das trainierte Modell betrachtet. Allerdings zieht der AIC-Wert die Modelle mit den einfachsten Parameterkombination vor als die Modelle mit komplexeren Parameterkombinationen (Brownlee, 2020a).

Der Unterschied zum Grid-Search-Verfahren besteht darin, dass Auto-ARIMA viel schneller das Ergebnis, also den optimalen Modellparameter, ermittelt und dadurch viele Ressourcen und Zeit einspart. Darüber hinaus wurde Auto-ARIMA nur für die ARIMA-basierten Modellen entwickelt. Aus diesem Grund wurde Auto-ARIMA nur für das SARIMA-Modell implementiert, jedoch nicht für das DeepAR-Modell. Folgende Parameter wurden eingegeben, um das SARIMA-Modell durch Auto-ARIMA zu optimieren:

Parameter	Werte
Y	df
seasonal	True
m	12
max_order	200
max_p	10
max_q	10
max_d	10
max_P	10
max_D	10
max_Q	10
start_p	0
start_q	0
start_P	0
start_Q	0
information_criterion	"AIC"
n_jobs	-1
alpha	0.05
maxiter	100
out_of_sample_size	int(len(df)*0.20)

Tabelle 6: Auto-ARIMA-Parameter zur HPO (Univariate-Zeitreihenanalyse)

Der Parameter *y* ist die univariate Ziel-Variable, auch als endogene Variable bekannt. Der Parameter *seasonal* bestimmt, ob Auto-ARIMA die Saisonalität miteinbeziehen soll. Der Parameter *m* ist für die Saisonalität im Datensatz zuständig, z.B. ist die Saisonalität im Eingabedatensatz pro Monat, daher wurde der Wert 12 angegeben. Für einen Datensatz pro Quartal würde der Wert 4

gewählt werden. Der nächste Parameter *max_order* zeigt die maximale Anzahl der zu untersuchende Parameterkombinationen.

Die Parameter *max_p*, *max_q*, *max_d*, *max_P*, *max_D* und *max_Q* stellen die Maximalwerte dar, also die p-,d-,q-,P-,D- und Q-Komponenten des SARIMA-Modells. Die Parameter *start_p*, *start_q*, *start_P* und *start_Q* bilden die Startwerte der p-, q-, P- und Q-Komponenten. Das bedeutet, dass Auto-ARIMA zwischen den Startwerten und Maximalwerte aller Parameter die passende Modelkombination angibt.

Der Parameter *n_jobs* zeigt die Anzahl der zu verwendenden CPU-Cores, wobei -1 alle zur Verfügung stehenden CPU-Cores einbezieht. Der andere Parameter *alpha* ist der Grenzwert zur Interpretation vom p-Value, wobei der Wert 0,05 gewählt wurde. Mit anderen Worten zeigt *alpha* das Niveau des Tests für Signifikanztests. Der p-Value gibt die Wahrscheinlichkeit an, dass der erzeugte Wert genauso hoch wäre, wenn er nur durch Zufall erzeugt worden wäre (Null-Hypothese). Wenn der p-Value kleiner als 0,05 ist, dann kann die Null-Hypothese als widerlegt betrachtet and von statistischer Signifikanz gesprochen werden (McNeese, 2017). Die letzten Parameter *maxiter* und *out_of_sample_size* sind jeweils für die maximale Iteration pro Modell und die Anzahl der Datensätze als Validierungsbeispiel zuständig. Wobei nur die letzten 20% des Eingabedatensatzes zur Validierung verwendet werden.

3.3.4.2 Optuna

Optuna ist eine Library zur automatischen Hyperparameteroptimierung, die speziell für maschinelles Lernen entwickelt wurde (Akiba *et al.*, 2019). In Optuna wird ein sogenannte Objective Function erstellt, indem die Art und Weise die zu optimierenden Parameter und des Algorithmus zum Modelltraining, z.B. DeepAR- oder SARIMA(X)-Modell, eingegeben wird. Danach werden ein Optimierungsalgorithmus und die Anzahl der sogenannten Trials hinzugefügt. Der verwendete Optimierungsalgorithmus in Optuna ist der sogenannte Tree-Structured Parzen Estimator (TPE), der eine Alternative zum Gaußschen Prozess ist (Bergstra *et al.*, 2011, p. 4) und die Bayes-Regel anwendet. Darüber hinaus ist TPE in der Lage, bessere Hyperparameter zu finden als die zufällige Suche mit der gleichen Anzahl von Versuchen. Folglich reduziert TPE die Laufzeit der HPO und stellt bessere Ergebnisse im Testdatensatz dar (Koehrsen, 2018). Im Gegensatz zu Auto-ARIMA wurde Optuna entwickelt, um sowohl für SARIMA als auch für DeepAR-Modell zu optimieren. In Optuna wurde der Trainingsdatensatz zur Optimierung und der Testdatensatz zur Evaluierung

eingefügt. Die HPO-Parameter, die im SARIMA- und DeepAR-Modell zur HPO in der Tabelle 2 und Tabelle 4 erklärt wurden, wurden ebenfalls zur Modelloptimierung mit Optuna verwendet.

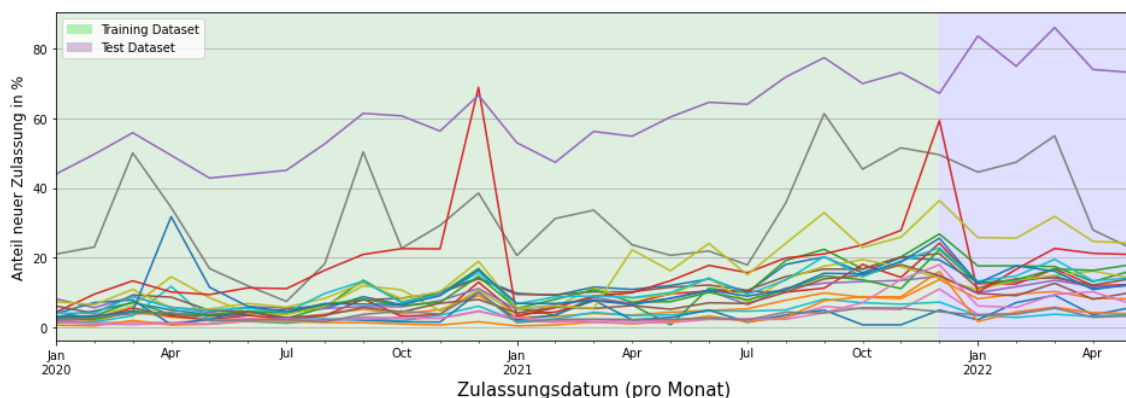
Optuna bietet die Möglichkeit die HPO-Ergebnisse anzuschauen und dementsprechend weitere Entscheidungen bezüglich der Modelloptimierung zu treffen. Beispielsweise visualisiert Optuna die sogenannten Objective Values während der Optimierung, wobei der niedrigste Wert als bester Parameter angegeben wird. Darüber hinaus stellt Optuna die Hyperparameter-Wichtigkeit (engl. hyperparameter importance) dar, die während der HPO erkannt werden.

3.4 Multivariate Zeitreihenanalyse

In diesem Abschnitt wird die Multivariate-Zeitreihenanalyse mit den SARIMAX- und DeepAR-Modellen erläutert. Der Eingabedatensatz zur Multivariate-Zeitreihenanalyse besteht aus 21 Variablen bzw. BEV-PKW-Neuzulassungen der 21 Ländern, wie im Abschnitt Explorative Analyse der multivariaten Daten auf Seite 31 beschrieben wurde. Zunächst wird der Train-Test-Split beschrieben. Danach werden das Modelltraining und die Modellevaluierung erklärt. Abschließend wird die HPO zur Multivariate-Zeitreihenanalyse erläutert.

3.4.1 Train-Test-Split

Um das trainierte Modell auszuwerten und das Modell zu evaluieren, wird der multivariaten Datensatz vor dem Modelltraining aufgeteilt. Infolgedessen werden ein Trainingsdatensatz (engl. training dataset) und ein Testdatensatz (engl. test dataset) erstellt. Der Train-Test-Split des multivariaten Datensatzes der BEV-PKW-Neuzulassungen sieht wie folgt aus:



Quelle: Eigene Darstellung, basierend auf European Alternative Fuels Observatory, 2022
Abbildung 12: Train-Test-Split des multivariaten Datensatzes

Abbildung 12 zeigt die Trainings- und Testdatensätze, die jeweils mit der grünen und blauen Farbe hervorgehoben sind. Die Größe des multivariaten Datensatzes beträgt 29 Monate und umfasst den Zeitraum vom Januar 2020 bis Mai 2022. Die ersten 23 Monate werden als Trainingsdatensatz mit dem Zeitraum Januar 2020 bis November 2021 festgelegt und die letzten 6 Monate dienen als Testdatensatz. Der Testdatensatz umfasst die Zeit von Dezember 2021 bis Mai 2022. Er entspricht ca. 20% des Gesamtdatensatzes. Da der benötigte Prognosehorizont sechs Monate beträgt, ist die Größe des Testdatensatzes ausreichend.

3.4.2 SARIMAX

Das sogenannte Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors Model (SARIMAX-Modell) ist eine Erweiterung des SARIMA-Modells. Das SARIMAX-Modell kann die exogene Variable zum Modelltraining miteinbeziehen, die zu einer Verbesserung der Modellgenauigkeit führt. Der Hauptunterschied des SARIMAX-Modells zum SARIMA-Modell besteht darin, dass die exogenen Variablen auch während der Prognose-Erstellung zum trainierten Modell eingespeist werden (Prabhakaran, 2021). Auch In der Multivariate-Zeitreihenanalyse mit dem SARIMAX-Modell ist die Zielvariable die BEV-PKW-Neuzulassungen in Deutschland und die exogenen Variablen sind die BEV-PKW-Neuzulassungen der 20 Länder, wie auf Seite 31 beschrieben wurde. Folgende Parameter wurden zum SARIMAX-Modelltraining verwendet:

Parameter	Werte
endog	df_train_endog
exog	df_train_exog
order	4, 0, 1
seasonal_order	0, 0, 3, 12
freq	'MS'
n_jobs	-1

Tabelle 7: Parameter des SARIMAX-Modelltrainings

Es wurden insgesamt sechs Parameter zum Training des SARIMAX-Modells verwendet, wie in der Tabelle 7 dargestellt sind. Der sogenannte *endog* Parameter ist die Zielvariable, die im univariaten Datensatz die BEV-PKW-Neuzulassungen in Deutschland ist. Der zweite Parameter ist *exog* und repräsentiert die exogenen Variablen bzw. die BEV-PKW-Neuzulassungen der 20 Länder. Die Parameter *order* und *seasonal_order* zeigen jeweils den nicht-saisonalen Teil und saisonalen Teil des SARIMAX-Modells. Ein weiterer

Parameter ist *freq*, der die Häufigkeit der Datensätze zeigt. Da der Eingabedatensatz der BEV-PKW-Neuzulassungen monatlich ist, wird der Wert 'MS' eingegeben. Der letzte Parameter *n_jobs* ist für die Anzahl der CPU-Cores zuständig, die beim Modelltraining verwendet werden sollen. Der Trainingsdatensatz wurde zum Modelltraining und das Testdatensatz zur Modellevaluierung verwendet. Mit anderen Worten wurden die ersten 23 Monate für das Modelltraining und die letzten 6 Monate zur Modellevaluierung genutzt. Auch beim SARIMAX-Modell wurde die Fehlermetrik RMSE ausgewählt, um die Güte des Modells auszuwerten. Auf Grund der Tatsache, dass es manuell sehr schwierig ist, die Parameterwerte zum optimalen Modelltraining zu finden, wurde die HPO verwendet.

Zur HPO der SARIMAX-Modellparameter wurden erneut Grid-Search und Auto-ARIMA sowie Optuna verwendet. Folgende Parameter wurden zur Optimierung des SARIMAX-Modells der Multivariate-Zeitreihenanalyse ausgewählt, weil diese den saisonalen und nicht-saisonalen Teil des SARIMAX-Modells enthalten.

Parameter	Min. Wert	Max. Wert
p	0	4
d	0	4
q	0	4
P	0	4
D	0	4
Q	0	4

Tabelle 8: Parameter des SARIMAX-Modells zur HPO

Wie in der Tabelle 8 dargestellt ist, wurde pro Parameter für den Minimalwert 0 und für den Maximalwert inklusive 4 angegeben. Das bedeutet, dass das SARIMAX-Modell bestmöglich durch einen optimalen Wert zwischen 0 und 4 trainiert werden kann. Die HPO zum SARIMAX-Modell der Multivariate-Zeitreihenanalyse wird unter dem Punkt 3.4.4 detailliert erklärt. Zunächst wird im Folgenden das DeepAR-Modell der Multivariate-Zeitreihenanalyse erläutert.

3.4.3 DeepAR

Das DeepAR unterstützt auch wegen seines globalen Modells die multivariaten Datensätze und erzeugt pro Zeitreihe ein separates Modell. Folglich werden für den Eingabedatensatz mit 21 Variablen je 21 Prognosen erstellt. Die Eigenschaften des minimalen manuellen Feature Engineering und die Miteinbeziehung der Saisonalität und andere Beziehungen zwischen den

Zeitreihen werden an dieser Stelle ausgenutzt, um eine Prognose mit hoher Genauigkeit zu erstellen. Zum Training des DeepAR-Modells wurden folgende Parameter verwendet:

Parameter	Werte
freq	"M"
context_length	19
prediction_length	6
use_feat_static_cat	True
cardinality	[1]
num_layers	23
num_cells	2
cell_type	"lstm"
dropout_rate	0.1122435424261366
epoch	50
batch_size	10
num_batches_per_epoch	14
learning_rate	0.09685043383629997

Tabelle 9: Parameter des Multivariate-DeepAR-Modelltrainings

Der erste Parameter ist *freq*, der die Data Frequency der Zeitreihe definiert. Im trainierten DeepAR-Modell wird *freq* als "M" bezeichnet, dass für den monatlichen Datensatz steht. Der zweite Parameter ist die *context_length*, der für die Anzahl der zu verwendenden Datensätze im RNN-Netzwerk zum Modelltraining zuständig ist, wobei entsprechend des HPO-Ergebnisses der Wert 19 gegeben wurde. Ein weiterer Parameter ist die *prediction_length*, der für den Prognosehorizont des Modells verantwortlich ist. Da das Ziel dieser Abschlussarbeit der Forecast der BEV-PKW-Neuzulassungen im Zeithorizont von sechs Monaten umfasst, wurde der Wert 6 gewählt.

Alle multivariaten Variablen müssen während des Train-Test-Split-Prozesses einzeln kodiert bzw. ein kategorischer Wert pro multivariate Variable erstellt werden. Diese Kodierung bzw. kategorischer Werte sollen dann als separates Feature im Datensatz eingegeben werden, damit das DeepAR-Modell versteht, dass ein globales Modell zur Multivariate-Zeitreihenanalyse erzeugt werden muss. Aus diesem Grund wurde der Parameter *use_feat_static_cat* als *True* bezeichnet, worauf geschlossen werden kann, dass der Datensatz kategorische Daten enthält. Darüber hinaus wurde der Parameterwert *cardinality* als 1 eingegeben. Das bedeutet, dass es nur eine kategorische Variable im Eingabedatensatz gibt.

Die Parameter *num_layers* und *num_cell* zeigen jeweils die Anzahl der Hidden-Layers und Cells, wobei der Parameter *cell_type* die Art der RNN-Zelle aufweist. Um das Overfitting zu vermeiden, wurde der Parameter *dropout_rate*, der ein Regulierungsparameter ist, verwendet. Der nächste Parameter *epoch* stellt die Anzahl der Epochen und der Parameter *batch_size* die Größe eines Batches dar. Der Parameter *num_batches_per_epoch* zeigt die maximale Anzahl des sogenannten Batches pro Epochen. Der Parameter *learning_rate* definiert einen Wert, womit sich das DeepAR seine internen Parameter anpassen kann. Auch beim DeepAR-Modell wurde der Trainingsdatensatz zum Modelltraining und der Testdatensatz zur Modellevaluierung genutzt. Der Trainingsdatensatz enthält erneut die ersten 23 Monate und der Testdatensatz die letzten 6 Monate. Zur Evaluierung der Qualität des DeepAR-Modells wurde die RMSE-Fehlermetrik verwendet. Um die richtigen Kombinationen der obengenannten Parameterwerte im DeepAR-Modell herauszufinden, sind die folgenden zwei HPO-Methoden zum Einsatz gekommen: Grid-Search und Optuna. Folgende Parameter wurden zur Optimierung der Multivariate-Zeitreihenanalyse des DeepAR-Modells ausgewählt:

Parameter	Min. Wert	Max. Wert
num_layers	2	30
num_cells	2	100
dropout_rate	0.0	0.2
epochs	5	50
batch_size	5	30
num_batches_per_epoch	5	30
learning_rate	0.001	0.1
Context_length	2	23

Tabelle 10: Parameter des Multivariate-DeepAR-Modells zur HPO

Wie in der Tabelle 10 dargestellt ist, wurde pro Parameter je ein Minimalwert und ein Maximalwert eingegeben. Das bedeutet, dass während der HPO für den jeweiligen Parameter ein Optimalwert zwischen den gegebenen minimalen und maximalen Werten gefunden werden soll, wodurch das Modell bestmöglich trainiert wird. Im Folgenden wird die HPO zum DeepAR-Modell der Multivariate-Zeitreihenanalyse erläutert.

3.4.4 Hyperparameteroptimierung (HPO)

In diesem Abschnitt wird die HPO für die zuvor erwähnten SARIMAX- und DeepAR-Modelle erklärt. In einem Grid-Search werden die Kombinationen aller möglichen Parameter standardmäßig untersucht. Infolgedessen werden diese Modelle entsprechend der niedrigsten RMSE-Werte in einer aufsteigenden Reihenfolge aufgelistet.

Grid-Search des SARIMAX-Modells hat mit den obengenannten Parametern 10 Stunden gedauert. Die Anzahl der DeepAR-Parameter beim Grid-Search musste eingeschränkt werden, damit die Ausführungszeit des Grid-Search verkürzt wird. Grid-Search für das DeepAR-Modells der Multivariate-Zeitreihenanalyse hat ca. 4 Tage (99,03 Stunden) gedauert, obwohl die eingeschränkten Parameterwerten eingegeben wurden. Die folgende Tabelle stellt die eingeschränkten Parameterwerte des DeepAR-Modells in Grid-Search dar:

Parameter	Werte
epochs	5,10,15,20,25
num_batches_per_epoch	5,10,15,20,25,30
batch_size	5,10,20,30
num_layers	2,5,8,10,15,20,22,25,27,30
num_cells	2,3,4,5,10,15,20,30,40

Tabelle 11: Eingeschränkte Parameterwerte des Multivariate-DeepAR-Modells

Die eingeschränkten DeepAR-Parameter haben nur einige vordefinierten Werte. Diese wurden entsprechend der Einleitung und Vorschläge des AWS-Hyperparameter-Tunings (*Tune a DeepAR Model*, 2022; *DeepAR Hyperparameters*, 2022) und des Trial and Error-Prozesses eingegeben. Die ersten 20 Grid-Search-Ausgabe der SARIMAX- und DeepAR-Modelle zur Multivariate-Zeitreihenanalyse sind im Anhang (Tabelle 19 und Tabelle 20) zu finden, wobei eine Zeile eine Modellkombination zeigt. Die anderen verwendeten HPO-Methoden Auto-ARIMA und Optuna werden im Folgenden erläutert.

3.4.4.1 Auto-ARIMA

Auto-ARIMA untersucht alle Modelle mit den nicht-saisonalen (p,d,q) und saisonalen (P,D,Q) Komponenten. Entsprechend der sogenannten AIC-, AICC- oder BIC-Informationskriterien gibt das Modell den niedrigsten Wert der ausgewählten Informationskriterien aus. Auto-ARIMA ermittelt das Ergebnis deutlich schneller, z.B. findet die p-, d-, q-, P-, D-, und Q-Werte. Sonst müssen

einen Stationaritätstest und manuelle Differenzierung sowie ACF- und PACF-Plots implementiert werden, dass zeitaufwändig und schwierig ist. Auto-ARIMA wird allerdings nur mit den ARIMA-basierten Modellen zum Einsatz gebracht. Aus diesem Grund wurde Auto-ARIMA nur für das SARIMA(X)-Modell implementiert. Folgende Parameter wurden zur Implementierung des Auto-ARIMA zur SARIMAX-HPO verwendet:

Parameter	Werte
y	df["Germany"]
exogenous	df[["alle Länder außer DTLD "]]
seasonal	True
m	12
max_order	200
max_p	10
max_q	10
max_d	10
max_P	10
max_D	10
max_Q	10
start_p	0
start_q	0
start_P	0
start_Q	0
information_criterion	"AIC"
n_jobs	-1
alpha	0.05
maxiter	100
out_of_sample_size	int(len(df)*0.20)

Tabelle 12: Auto-ARIMA-Parameter zur HPO (Multivariate-Zeitreihenanalyse)

Der Parameter *y* ist als die endogene Variable bekannt, wobei der Prozentaule Marktanteil der BEV-PKW-Neuzulassungen in Deutschland, `df["Germany"]`, als Wert eingegeben wurde. Der Parameter *exogenous* ist für die exogene Variable zuständig, deswegen wurden die anderen BEV-PKW-Neuzulassungen der 20 Länder verwendet. Der nächste Parameter *seasonal* ist dafür zuständig, ob das SARIMA-Modell die Saisonalität miteinbezogen wird, während der Parameter *m* die Anzahl der Saisonalität im Datensatz darstellt. Da die Saisonalität im Eingabedatensatz monatlich ist, wird der Wert 12 mit angegeben. Der Parameter *max_order* zeigt die maximale Anzahl der Modellkombinationen, die Auto-ARIMA untersuchen sollte. Die Parameter *max_p*, *max_q*, *max_d*, *max_P*, *max_D* und

max_Q definieren die maximalen Werte zur saisonalen- und nicht-saisonalen Komponenten. Die Parameter *start_p*, *start_q*, *start_P* und *start_Q* zeigen der Startwerte der p-, q-, P- und Q-Komponenten. Folglich gibt Auto-ARIMA die passende Modellkombination zwischen den Startwerten und maximalen Werten aller Parameter entsprechend den gegebenen Informationskriterien z.B. AIC aus. Der Parameter *n_jobs* zeigt die Anzahl der zu verwendenden CPU-Cores, wobei -1 die Nutzung aller zur Verfügung stehenden CPU-Cores bedeutet. Der andere Parameter *alpha*, auch als p-Value bekannt, ist für die Wichtigkeit der Parameterwerte verantwortlich. Der Wert 0.05 bedeutet, dass nur Parameter gleich oder kleiner als 5% p-Value in der HPO durch Auto-ARIMA im Fokus stehen. Der Parameter *maxiter* ist definiert die maximale Iterationsanzahl pro Modell. Der Parameter *out_of_sample_size* ist für die Anzahl der Datensätze zur Validierung während des Trainings zuständig, indem nur die letzten 20% des Eingabedatensatzes angegeben wurde.

3.4.4.2 Optuna

Im Gegensatz zu Auto-ARIMA ist Optuna sowohl für das SARIMAX als auch für das DeepAR-Modell verwendbar. In Optuna wird, wie bei der Univariate-Zeitreihenanalyse, der Trainingsdatensatz zur Modelloptimierung und der Testdatensatz zur Modellevaluierung eingegeben. Die HPO-Parameter, die im SARIMAX- und DeepAR-Modell zur HPO erwähnt wurden (Tabelle 8 und Tabelle 10), werden ebenfalls zur Modelloptimierung mit Optuna verwendet. Der verwendete Optimierungsalgorithmus ist das TPE, das sowohl in Optuna als auch im DeepAR-Modell der Multivariate-Zeitreihenanalyse Anwendung fand. Optuna bietet die Möglichkeit, die HPO-Ergebnisse zu visualisieren, damit die Entscheidungen zur Hyperparameteroptimierung erleichtert wird. Beispielsweise kann die Parameterwerte entsprechend der Visualisierung der Parameterwichtigkeit angepasst werden. Darüber hinaus visualisiert Optuna die sogenannten Objective Values während der Optimierung, wobei der niedrigste Wert als beste Parameter ausgegeben wird. Die Endergebnisse der HPO, vor allem die von Optuna, werden im Kapitel 4 detailliert erläutert.

4 Ergebnisse

In diesem Kapitel werden die Ergebnisse dieser Abschlussarbeit zum Forecasting der BEV-PKW-Neuzulassungen in Deutschland erläutert. Zunächst werden die HPO-Ergebnisse beschrieben. Danach werden die Ergebnisse der Univariate- und Multivariate-Zeitreihenanalyse erklärt. Abschließend werden die zwei Modelle der Univariate- und Multivariate-Zeitreihenanalyse, d.h. SARIMA(X)- und DeepAR, miteinander verglichen.

4.1 Hyperparameteroptimierung

Um die Modelle der Univariate- und Multivariate-Zeitreihenanalyse zu optimieren, mussten zunächst die richtigen Parameterkombinationen zum Modelltraining gefunden werden. Zudem wurde insgesamt drei HPO-Methode implementiert: Auto-ARIMA, Grid-Search und Optuna.

Auto-ARIMA ist schnell und gibt entsprechend der sogenannten Informationskriterien AIC, BIC, AICC oder HQIC eine Modellkombination aus. Allerdings ist es auch möglich, dass die ausgegebene Kombination nicht die optimalen Parameterwerte repräsentiert, denn im Gegensatz zu Grid-Search durchsucht es nicht alle möglichen Parameterkombinationen. Beispielsweise kann Auto-ARIMA einen niedrigen AIC-Wert zeigen, jedoch das trainierte Modell während der Modellevaluierung oder bei der Prognose einen schlechten RMSE-Wert ausgibt. Das Akaike Information Criteria (AIC) ist eine weit verbreitete Metrik eines statistischen Modells. Es quantifiziert im Grunde die Anpassungsgüte und die Einfachheit des Modells in einer einzigen Statistik (Keshvani, 2013). Die HPO durch Auto-ARIMA wurde in dieser Arbeit implementiert, da aber die HPO-Methoden Grid-Search und Optuna bessere Parameterwerte geliefert haben, wurden im Folgenden nur diese beiden Methoden behandelt.

Im Vergleich dazu hat Grid-Search alle möglichen Parameterkombinationen gesucht. Womit zwischen den gegebenen Parameterwerten die optimale Kombination gefunden werden, das einen Vorteil darstellt. Der Nachteil dabei ist allerdings, das Grid-Search viel Zeit benötigt, die Suche nach den

Parameterkombination zu beenden und dementsprechend das optimale Modell darzustellen. Das ist vor allem problematisch für die Modelle, die auf neuronalen Netzwerken basieren, weil die neuronalen Netzwerk-basierten Modelle lange Trainingszeit benötigen. Aus diesem Grund hat Grid-Search beim DeepAR-Modelltraining mehr Zeit als das SARIMA(X)-Modell in Anspruch genommen. Um die Suchzeit zu verkürzen, musste die Anzahl der gewählten Parameterwerte eingeschränkt werden.

Im Gegensatz zu Auto-ARIMA und Grid-Search sucht Optuna die optimalen Parameterwerte in einigen ausgewählten Versuchen, die als Trials bezeichnet werden. In Optuna wurde die Anzahl der Trials auf 100 festgelegt, damit die Ergebnisse aller Modelle einheitlich verglichen werden können. Das bedeutet, dass Optuna innerhalb der gegebenen 100 Trials die optimalen Parameterwerte bzw. Parameterkombination findet. Das ist ein enormer Vorteil insbesondere für die Deep Learning-Modelle wie z.B. DeepAR, denn dadurch kann viel Zeit und Ressourcen gespart werden. Die Tabelle 13 fasst den HPO-Unterschied zwischen Auto-ARIMA, Grid-Search und Optuna zusammen.

HPO	Univariate-Zeitreihenanalyse		Multivariate-Zeitreihenanalyse	
	SARIMA	DeepAR	SARIMAX	DeepAR
Auto-ARIMA	< 2 Min.	–	< 2 Min.	–
Grid Search	69 Std.	253,5 Std.	10 Std.	99 Std.
Optuna	3 Min.	6,7 Std.	5 Min.	3,8 Std.

Tabelle 13: Vergleich der HPO-Methoden

Die Tabelle 13 zeigt die Laufzeit aller verwendeten HPO-Methoden zur Univariate- und Multivariate-Zeitreihenanalyse. Wie in der Tabelle 13 dargestellt, findet Auto-ARIMA die Modellparameter schneller als die beiden anderen Methoden. Allerdings ist Auto-ARIMA für das DeepAR-Modell nicht geeignet, weil es für die Optimierung der ARIMA-basierten Modellen entwickelt wurde. Aus diesem Grund gibt es in der Tabelle 13 zum DeepAR-Modell keine Werte. Die Tabelle 13 zeigt, dass die HPO-Laufzeit der Univariate-Zeitreihenanalyse-Modelle länger als die Multivariate-Zeitreihenanalyse-Modelle dauert. Der Hauptgrund dafür ist die Größe des univariaten Eingabedatensatzes. Zum Beispiel beträgt die Größe des univariaten Eingabedatensatzes 69 Zeitpunkte, während der multivariate Eingabedatensatz nur 23 Zeitpunkte umfasst.

Die Laufzeit des Grid-Search-Prozesses ist generell abhängig von dem zu trainierenden Modell, Anzahl der Parameterkombination und der Größe des Eingabedatensatzes. Grid-Search für die SARIMA- und SARIMAX-Modelle

haben jeweils 69 und 10 Stunden gedauert. Die Grid-Search-Laufzeit des DeepAR-Modells der Univariate-Zeitreihenanalyse beträgt 253,5 Stunden, während es für das DeepAR-Modell der Multivariate-Zeitreihenanalyse 99 Stunden gedauert hat. Grid-Search benötigt mehr Zeit als die anderen Methoden. Es brauchte mehrere Stunden für die DeepAR-Modelle, die Parameter zu optimieren, obwohl nur eine eingeschränkte Anzahl der Parameterwerten zur Optimierung angegeben wurde. Wäre die vollständige Anzahl der Parameterwerte gewählt worden, würde die Grid-Search des DeepAR-Modells mit hoher Wahrscheinlichkeit mehrere Tage in Anspruch nehmen.

Optuna ist verglichen zu den anderen zwei Methoden sehr gut geeignet, da trotz der Nutzung der vollständigen Anzahl der Parameterwerte die Dauer der HPO weniger als 7 Stunden beträgt. Darüber hinaus stellt Optuna alle gesuchten HPO-Ergebnisse in Form eines Dataframes dar und zeigt die Fehlerverläufe, Parameterwichtigkeit und weitere Visualisierungen mit einer Codezeile. Deswegen wird im Folgenden zum Vergleichszweck Optuna als die Haupt-HPO-Methode verwendet.

4.2 Univariate Zeitreihenanalyse

In diesem Abschnitt werden die Ergebnisse des Modelltrainings, der Modellevaluierung und des Forecasts mit den SARIMA- und DeepAR-Modellen der Univariate-Zeitreihenanalyse erläutert.

4.2.1 SARIMA

Das SARIMA-Modell wurde mit den Parametern trainiert, die in der Tabelle 1 auf Seite 35 beschrieben wurden. Das SARIMA-Modell bietet die Möglichkeit, nach dem Modelltraining die sogenannte Model Summary (Modellzusammenfassung) einzusehen. Die Abbildung 13 zeigt die verwendeten Parameter und die Modelzusammenfassung des trainierten SARIMA-Modells.

SARIMAX Results						
Dep. Variable:	Anteil der BEV-PKW-Neuzulassungen in Deutschland				No. Observations:	63
Model:	SARIMAX(3, 2, 1)x(1, 0, [], 12)				Log Likelihood	-114.625
Date:	Thu, 03 Nov 2022				AIC	241.251
Time:	23:13:01				BIC	253.916
Sample:	01-01-2017 - 03-01-2022				HQIC	246.215
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	-0.1585	0.127	-1.251	0.211	-0.407	0.090
ar.L2	-0.1887	0.152	-1.239	0.215	-0.487	0.110
ar.L3	0.1482	0.228	0.650	0.515	-0.298	0.595
ma.L1	-0.9996	2.572	-0.389	0.697	-6.040	4.041
ar.S.L12	0.5729	0.170	3.368	0.001	0.239	0.906
sigma2	2.1961	5.713	0.384	0.701	-9.002	13.394
Ljung-Box (L1) (Q):			0.00	Jarque-Bera (JB):	274.58	
Prob(Q):			1.00	Prob(JB):	0.00	
Heteroskedasticity (H):			232.25	Skew:	-2.05	
Prob(H) (two-sided):			0.00	Kurtosis:	12.55	

Abbildung 13: Modellzusammenfassung des SARIMA-Modells

In der Abbildung 13 sind mehrere Informationen zu sehen. Einige sind sehr einfach, zu verstehen, z.B. die Zielvariable, Zeit und das Datum des Modelltrainings, und andere sind komplexer. Allerdings werden nur die wichtigsten Informationen der SARIMA-Modellzusammenfassung erklärt, die zur Modellauswahl essenziell sind. Die verwendete Modellkombination im SARIMA-Modell wird, mit blau markiert, in der Modellzusammenfassung dargestellt. Die Modellkombination (3,2,1)x(1,0,0,12) sind dieselben Parameterwerte, die in der Tabelle 1 beschrieben wurden, nämlich die p-,d-,q-,P-,D- und Q sowie die m-Komponente des SARIMA-Modells.

Ein weiterer in magentafarben markierter wichtiger Punkt, ist der AIC-Wert, der die Modellqualität beschreibt. Allerdings wurde der RMSE-Wert zur Modellevaluierung ausgewählt, weil das AIC-Informationskriterium zur Auswertung der DeepAR-Modellgüte nicht implementiert werden kann. Aus diesem Grund kann es sein, dass das verwendete Modell einen höheren AIC-Wert hat, aber niedriger RMSE-Wert ausgibt. Das ist allerdings nicht problematisch, weil durch die RMSE-Fehlermetrik sowohl SARIMA(X) als auch DeepAR-Modell ausgewertet werden kann.

Die sogenannte Lag Variable des AR- und MA-Modells im SARIMA-Modell sowie deren Koeffizienten sind mit orange markiert. Diese Werte werden entsprechend durch den p-, q-, P- und Q-Komponenten dargestellt. Mit anderen Worten gibt es drei AR-Lags und ein MA-Lag, weil das AR- und MA-Modell des trainierten SARIMA-Modells jeweils einen Wert von 3 und 1 enthält. Die p-Komponenten, markiert mit grün, zeigen die statistische Signifikanz der Lag-Variablen. Ist der Wert der p-Komponente kleiner als 0,05%, wird die gegebene Lag-Variable als wichtig betrachtet, ansonsten wird sie als unwichtig eingestuft (Smigel, 2022). Im trainierten SARIMA-Modell sind die ersten 4 Lag-Variablen statistisch unwichtig,

weil ihre p-Werte größer als 0,05% sind. Allerdings ist die fünfte Lag-Variable, die für den saisonalen Teil bzw. die P-Komponente des trainierten SARIMA-Modells steht, statistisch signifikant.

Das HPO-Ergebnis von Auto-ARIMA enthält eine Parameterkombination mit den p-Werten kleiner als 0,05%. Jedoch waren der RMSE-Wert und die Prognose nicht besser als bei Optuna und Grid-Search. Die HPO durch Optuna zeigte eine Parameterkombination mit dem RMSE-Wert von 1,26 Prozentpunkten, während die HPO durch Grid-Search einen RMSE-Wert von 0,88 Prozentpunkten ausgegeben hat. Im Gegenteil zu Grid-Search und Auto-ARIMA ist es möglich den Optimierungsverlauf der Fehlerwerten innerhalb der gegebenen Trial-Anzahl zu visualisieren.

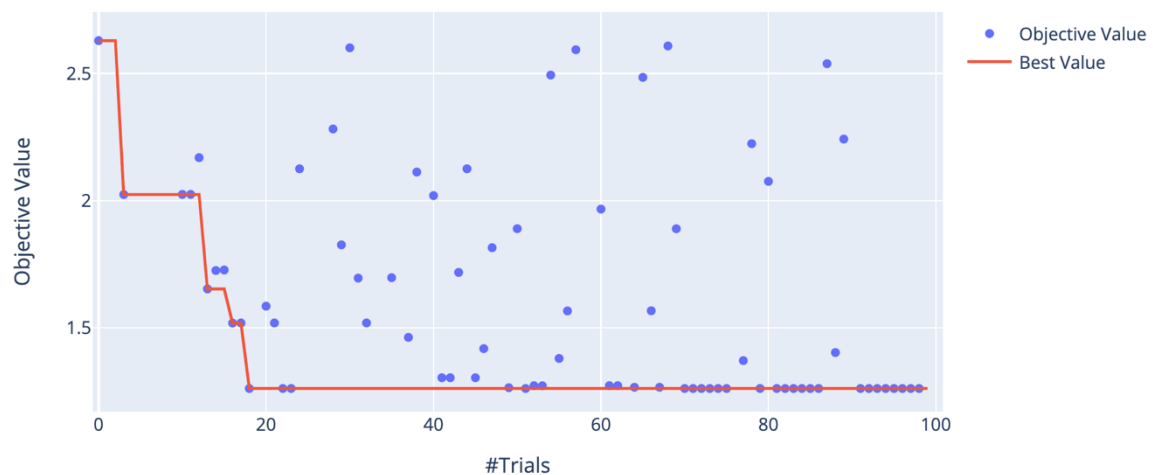


Abbildung 14: Optimierungsverlauf des SARIMA-Modells

Abbildung 14 zeigt die RMSE-Fehlermetrik, die durch die Optuna-HPO aufgezeichnet wurde. Das sogenannte Objective Value in der Abbildung 14 weist die RMSE-Fehlermetrik auf, die bereits in der Objective-Funktion von Optuna definiert wurde. Das sogenannte Best Value zeigt den niedrigsten Fehlerwert im Verlauf der Optuna-HPO, die mit der roten Linie dargestellt wurde. Obwohl 100 Trials bzw. Versuche zur HPO angegeben wurden, konnte Optuna schon innerhalb der ersten 20 Versuche die optimale Parameterkombination finden. Die blauen Punkte sind die Objective-Values, die eine nicht-optimale Parameterkombination zeigen. Darüber hinaus stellt Optuna die Wichtigkeit der Hyperparameter (engl. hyperparameter importance) der optimalen Parameterkombination dar.

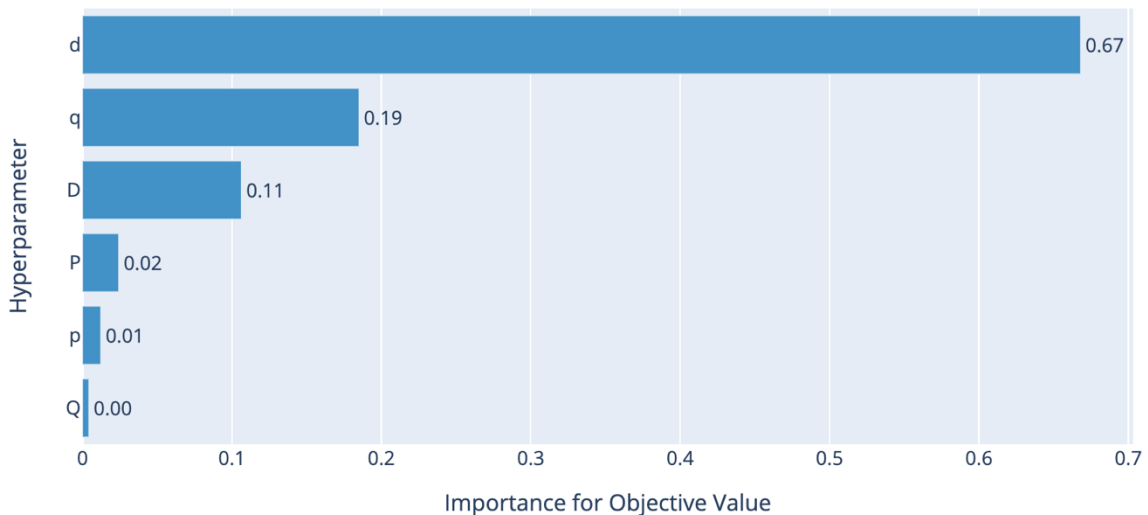


Abbildung 15: Hyperparameterwichtigkeit des SARIMA-Modells

Abbildung 15 stellt alle in der Objective-Function eingegebenen Parameter mit ihrer prozentualen Wichtigkeit dar. Beispielsweise zeigt die Abbildung 15, dass die wichtigsten Parameter die d-q- und D-Komponenten des SARIMA-Modells jeweils mit 67%, 19% und 11% sind. Die anderen Parameter, nämlich p-, P- und Q-Komponente, spielen beim Modelltraining eine geringe Rolle. Das bedeutet, dass die drei wichtigsten genannten Parameter großen Einfluss beim Training und bei der Optimierung des SARIMA-Modells haben.

In der Praxis wird wegen der kürzeren Laufzeit immer eine HPO-Methode wie Optuna ausgewählt. Allerdings wurde stattdessen die Parameterkombination des Grid-Search-Ergebnisses trotz längerer Laufzeit genutzt, weil dadurch sich der RMSE-Wert noch reduziert hat. Das SARIMA-Modell stellt auch die Modell-Diagnostik dar, nach dem das Modell trainiert wurde. Die Modell-Diagnostik des trainierten SARIMA-Modells sieht folgendermaßen aus:

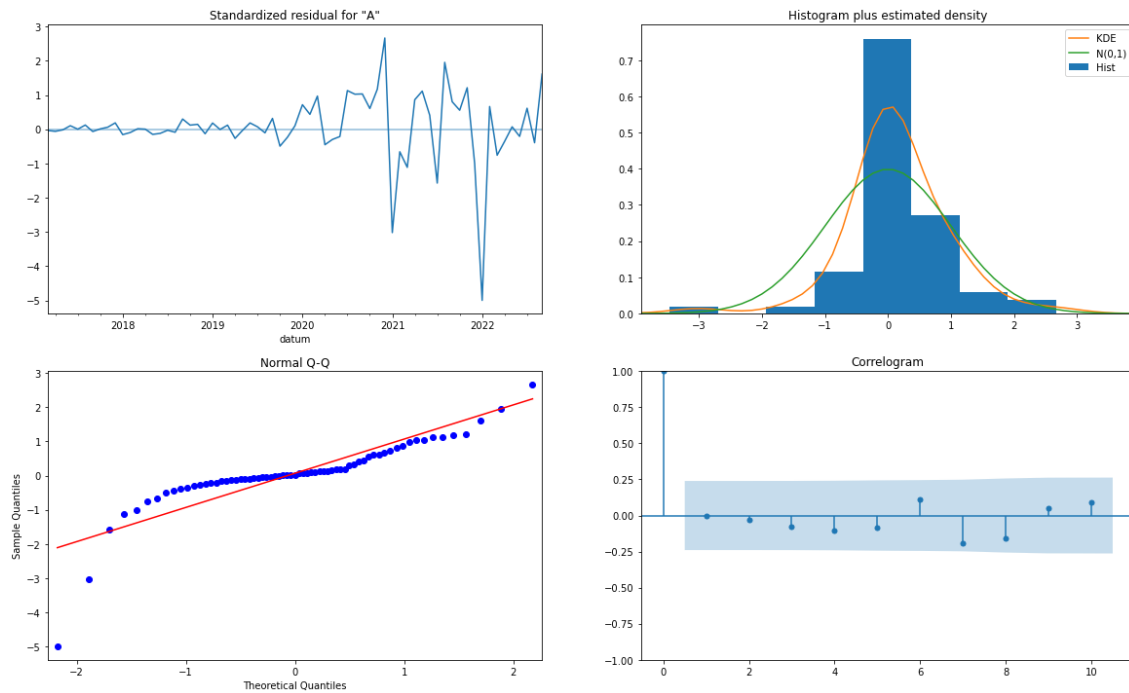


Abbildung 16: Modell-Diagnostik des SARIMA-Modells

Abbildung 16 stellt vier Visualisierungen dar. Die Visualisierung oben links veranschaulicht das Residual des SARIMA-Modells, das einen Mittelwert von 0 enthält. Oben rechts wird das sogenannte Density Plot des SARIMA-Modell-Residuals gezeigt. Das Density Plot des Residuals zeigt eine normal Verteilung. Hätte das Residual des trainierten Modells keine normal Verteilung, würde das Modell als nicht-optimal betrachtet werden. Die Visualisierung unten links stellt das Q-Q-Plot dar. Je näher die blauen Punkte an der roten Linie liegen, desto besser ist das Modell. Eine signifikante Abweichung würden bedeuten, dass die Verteilung verzerrt ist. Die letzte Visualisierung rechts unten zeigt das Korrelogramm bzw. ACF-Plot des Residuals. Eine Autokorrelation im Korrelogramm würde bedeuten, dass das Residual ein Muster aufweist (Prabhakaran, 2021), das im Modell nicht miteinbezogen wurde. Dementsprechend musste das trainierte Modell mit besseren Parameterwerten erneut optimiert und trainiert werden. Da das Residual keine Korrelation und kein Muster im Residual des trainierten SARIMA-Modells zeigt, wird das trainierte SARIMA-Modell als optimal betrachtet. Das trainierte SARIMA-Modell wird dann verwendet, um die Modellevaluierung und Prognose zu erstellen.

4.2.1.1 Modellevaluierung

Um die Qualität des trainierten SARIMA-Modells auszuwerten, wurde der Testdatensatz genutzt. Die Modellevaluierung des SARIMA-Modells wird wie folgt dargestellt:

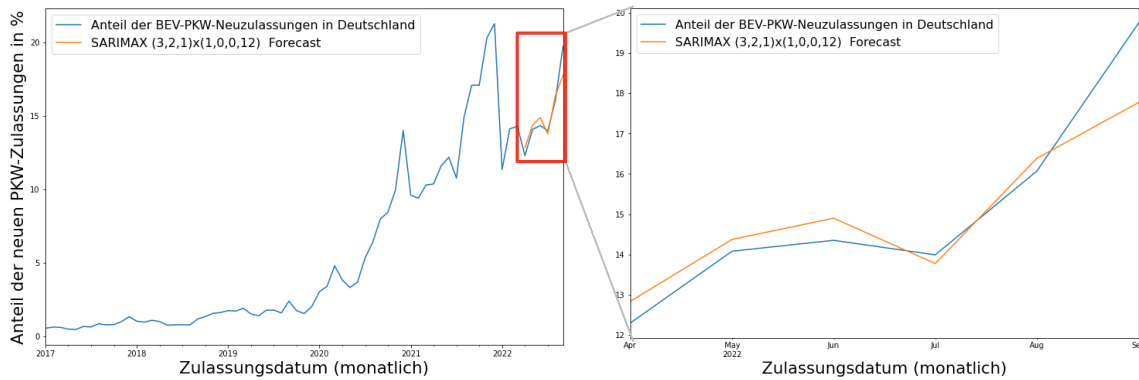


Abbildung 17: SARIMA-Modellevaluierung

Abbildung 17 zeigt die Modellevaluierung des SARIMA-Modells gegen den Testdatensatz. Die Visualisierung auf der linken Seite stellt den Trainings- und Testdatensatz sowie die Prognose auf sechs Monate dar. Die Visualisierung auf der rechten Seite weist die Prognose gegen den Testdatensatz auf. Entsprechend der orangenen Linie auf der rechten Visualisierung kann festgestellt werden, dass die prognostizierten Werte ungefähr denen im Testdatensatz folgen. Auch die Saisonalität der letzten Monate in 2022 wurde durch das Modell scheinbar miteinbezogen, z.B. liegen die prognostizierten Werte höher als die Werte im Testdatensatz im September 2022. Je näher die Prognose dem Testdatensatz ist, desto geringer ist der RMSE-Wert. Der RMSE-Wert des trainierten SARIMA-Modells beträgt 0,88 Prozentpunkte. Das bedeutet, dass alle prognostizierten Punkte des Modells einen durchschnittlichen Fehler bzw. eine Abweichung von 0,88 Prozentpunkten vom prozentualen Marktanteil der BEV-PKW-Neuzulassungen in Deutschland haben.

4.2.1.2 Prognose

Das Ziel des Modelltrainings und der Modellevaluierung war eine Prognose im Zeithorizont von sechs Monaten zu erstellen. Nach der Modellevaluierung und entsprechend des niedrigen RMSE-Werts lässt sich vermuten, dass sich das Modell und die Parameterwerte der HPO-Methode zum SARIMA-Modell sehr gut für diesen Zweck eignen. Aus diesem Grund wird das SARIMA-Modell erneut mit dem gesamten Eingabedatensatz trainiert, also dem Trainings- und Testdatensatz. Die gleichen Parameter, die während der Modellevaluierung verwendet wurden, werden auch dazu genutzt, um den Forecast zu erstellen. Die prognostizierten Werte des SARIMA-Modells lassen sich folgendermaßen darstellen:

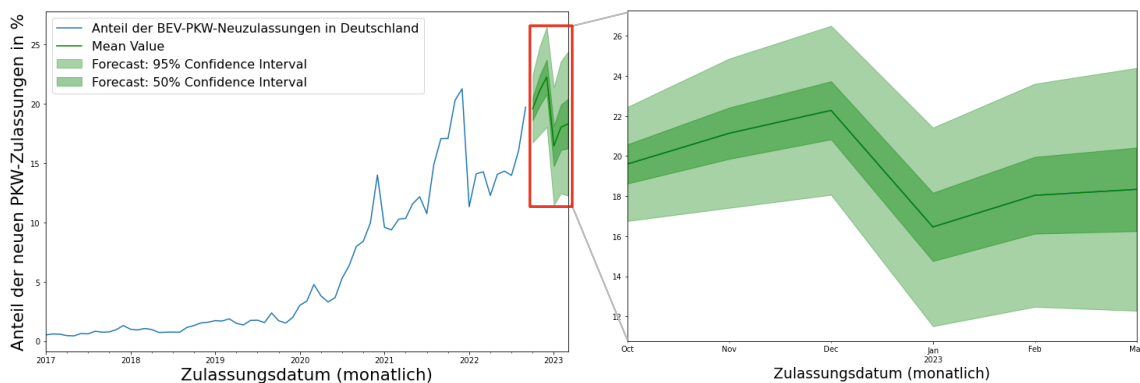


Abbildung 18: Prognose des SARIMA-Modells

Abbildung 18 zeigt die Prognosen zur BEV-PKW-Neuzulassung in Deutschland durch das SARIMA-Modell. Diese Prognosen umfassen die nächsten 6 Monate vom Oktober 2022 bis einschließlich März 2023. Die Visualisierung auf der linken Seite stellt den gesamten Eingabedatensatz mit der Prognose auf sechs Monate und den Durchschnittswerten sowie die Konfidenzintervalle von 95% und 50% dar. Da die Prognose auf der linken Visualisierung sehr klein ist, zeigt die rechte Visualisierung nur den Ausschnitt mit der Prognose.

Das Konfidenzintervall von 95% bedeutet, dass statistisch in 95% der Fälle der zukünftige Wert im Bereich dieser 95% liegen wird. Der niedrigste Wert in einem Konfidenzintervall wird als Lower Bound und der höchste Wert als Upper Bound bezeichnet. Beispielsweise liegt das Lower-Bound des Konfidenzintervalls von 95% im Oktober 2022 bei 16,75% und das Upper-Bound bei 22,44%. Das bedeutet, dass das Intervall von 16,75% bis 22,44% zu 95% den wahren Wert des Oktober-Marktanteils enthält. Das Gleiche gilt für das 50% Konfidenzintervall: Das Lower-Bound und Upper-Bound des 50% Konfidenzintervalls zur Prognose im Oktober 2022 liegen jeweils bei 18,62% und 20,58%. Das bedeutet, dass das Intervall von 18,62% bis 29,58% zu 50% den wahren Wert des Oktober-Marktanteils enthält. Die Tabelle 21 im Anhang zeigt die vollständigen prognostizierten Werte mit den Durchschnittswerten sowie die 95%- und 50%- Konfidenzintervalle.

4.2.2 DeepAR

Die Parameter in der Tabelle 3 auf Seite 39 wurden verwendet, um das DeepAR-Modell der Univariate-Zeitreihenanalyse zu trainieren. Im Gegensatz zum SARIMA(X)-Modell kann das DeepAR-Modell kein Model Summary oder keine Modell-Diagnostik anzeigen. Die optimale Modellkombination entsprechend des Grid-Search hat einen RMSE-Wert von 0,66 Prozentpunkten. Allerdings wurde die optimale Modellkombination des Grid-Search nicht ausgewählt, weil die

Parameterkombination der Optuna-HPO mit 1,29 RMSE-Wert, eine bessere Prognose dargestellt hat. Da die gefundene Hyperparameteroptimierung mit Grid-Search die eingeschränkte Anzahl von Parameterwerten enthielt, zeigt das endgültige trainierte DeepAR-Modell einen schlechten RMSE-Wert an. Der Optimierungsverlauf der Fehlerwerte durch Optuna wird wie folgt dargestellt:

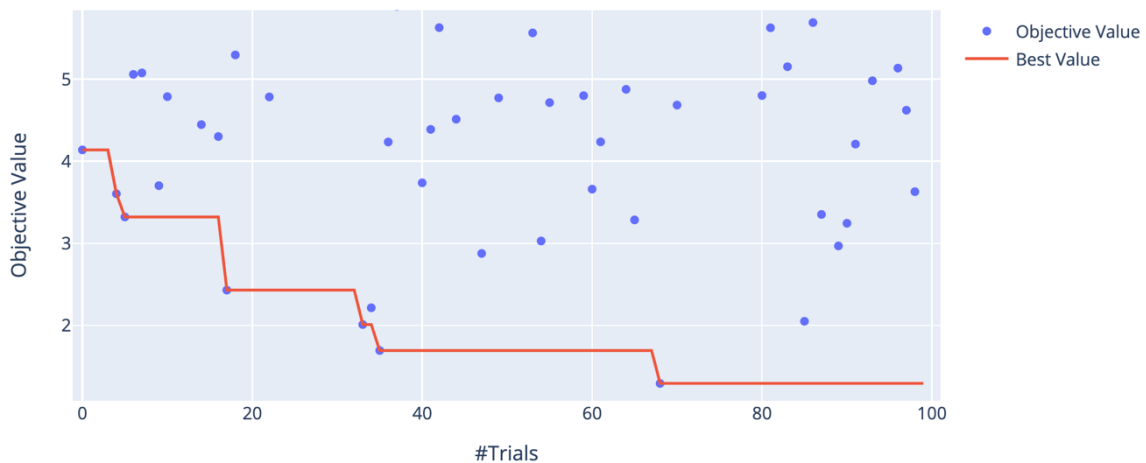


Abbildung 19: Optimierungsverlauf des Univariate-DeepAR-Modells

Abbildung 19 zeigt die RMSE-Fehlermetrik bzw. das Objective-Value des Optimierungsprozesses der Optuna-HPO. Der Best-Value, der durch die rote Linie dargestellt ist, zeigt den niedrigsten RMSE-Wert während der Optuna-HPO. Die Fehlerwerte sanken stetig und konnten mit der Parameterkombination des 68. Trials das beste DeepAR-Modell mit dem niedrigsten RMSE-Wert trainieren. Die blauen Punkte sind die Objective-Values, die die nicht-optimale Parameterkombinationen darstellen. Die Wichtigkeit der Hyperparameter der optimalen Parameterkombination lässt sich folgendermaßen darstellen:

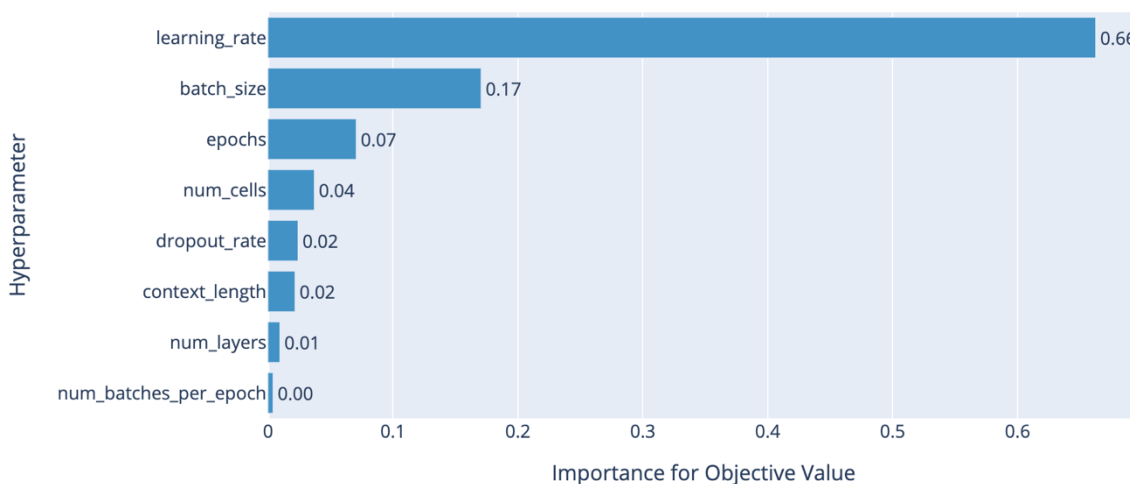


Abbildung 20: Hyperparameterwichtigkeit des Multivariate-DeepAR-Modells

Wie in der Abbildung 20 dargestellt, sind die Parameter `learning_rate`, `batch_size` und `epochs` jeweils mit 66%, 17% und 7% die wichtigsten drei Parameter. Die

anderen Parameter zum DeepAR-Modelltraining der Univariate-Zeitreihenanalyse haben eine Wichtigkeit von nicht mehr als 4%. Das an sich bedeutet, dass die Parameter `learning_rate` und `batch_size` sowie `epochs` insgesamt 90% Einfluss auf das DeepAR-Modelltraining haben.

4.2.2.1 Modellevaluierung

Zur Auswertung der Qualität des trainierten DeepAR-Modells wurde der Testdatensatz und zum Modelltraining der Trainingsdatensatz verwendet. Die Modellevaluierung des DeepAR-Modells lässt sich folgendermaßen darstellen:

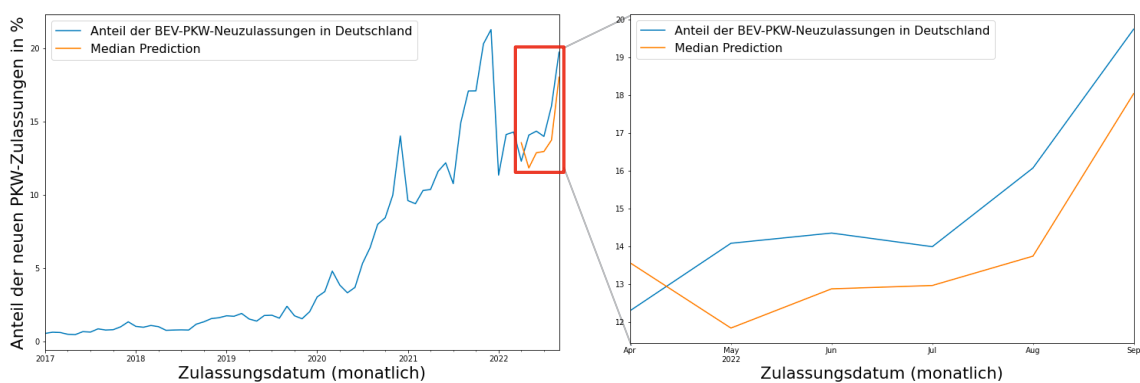


Abbildung 21: Modellevaluierung des Univariate-DeepAR-Modells

Abbildung 21 zeigt die Modellevaluierung des DeepAR-Modells gegen den Testdatensatz. Die Visualisierung auf der linken Seite stellt den Trainings- und Testdatensatz sowie die Prognose im Prognosehorizont von sechs Monaten dar. Die Visualisierung auf der rechten Seite zeigt eine Vergrößerung des Abschnittes mit Vergleich der Prognose gegen den Testdatensatz. Die prognostizierten Werte, die orangefarbene Linie, liegen nahe an den Punkten des Testdatensatzes. Je näher die prognostizierten Werten zum Testdatensatz liegen, desto geringer ist der RMSE-Wert und optimaler wird das Modell betrachtet.

Das endgültige DeepAR-Modell hat einen RMSE-Wert von 1,29 Prozentpunkten. Das bedeutet, dass ein durchschnittlicher Fehler von 1,29 Prozentpunkten bei den prognostizierten Werten zum prozentualen Marktanteil der BEV-PKW-Neuzulassungen zu erwarten ist. Die Modellevaluierung des DeepAR-Modells basiert auf Median-Werten. Im Folgenden wird die Prognose des prozentualen Marktanteils der BEV-PKW-Neuzulassungen in Deutschland erläutert.

4.2.2.2 Prognose

Nach der Modellevaluierung und entsprechend des RMSE-Werts von 1,29 Prozentpunkten wird eine Prognose auf sechs Monate erstellt. Zudem wurde der gesamte Eingabedatensatz mit den gleichen Parametern verwendet, um das DeepAR-Modell erneut zu trainieren. Die Prognose des DeepAR-Modells wird wie folgt visualisiert:

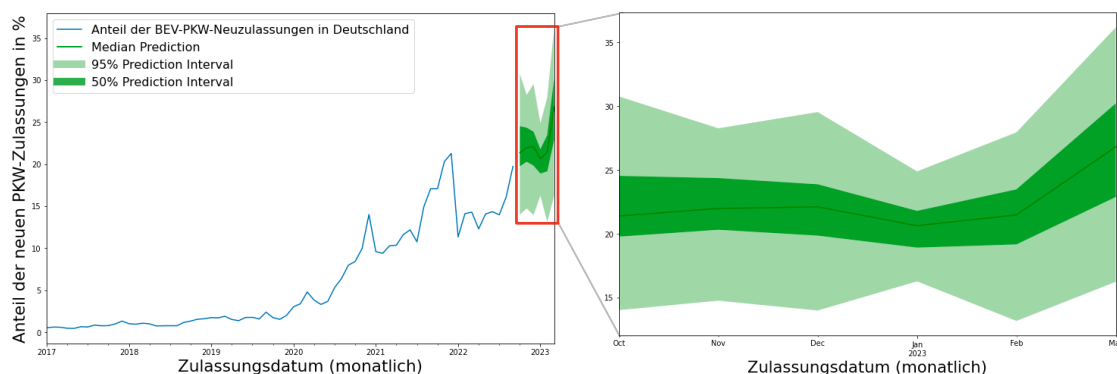


Abbildung 22: Prognose des Univariate-DeepAR-Modells

Abbildung 22 stellt die Prognose der nächsten 6 Monaten des prozentualen Marktanteils der BEV-PKW-Neuzulassung in Deutschland vom Oktober 2022 bis einschließlich März 2023 dar. Die Visualisierung auf der linken Seite zeigt den gesamten Eingabedatensatz und die Prognose der nächsten 6 Monate mit den Medianwerten sowie den Konfidenzintervalle von 95% und 50%. Die Visualisierung auf der rechten Seite zeigt die Detailansicht der Prognose der nächsten 6 Monate mit den Medianwerten und den Konfidenzintervallen, für eine bessere Orientierung. Das Konfidenzintervall von 95% bedeutet, dass statistisch in den 95% der Fälle die zukünftigen Werte zwischen dem gegebenen 95% Konfidenzintervall liegen wird.

Das Lower-Bound im Oktober 2022 des 95%-Konfidenzintervalls liegt bei 13,99% und das Upper-Bound bei 30,75%. Das bedeutet, dass das Intervall von 13,99% bis 30,75% zu 95% den wahren Wert des Oktober-Marktanteils enthält. Das gleiche gilt für das 50%-Konfidenzintervall. Hier liegen das Lower-Bound und Upper-Bound des 50%-Konfidenzintervalls zur Prognose im Oktober 2022 jeweils bei 19,76% und 24,52%. Das bedeutet, dass das Intervall von 19,76% bis 24,52% zu 50% den wahren Wert des Oktober-Marktanteils enthält. Die vollständigen Prognosewerte durch das DeepAR-Modell sind im Anhang in der Tabelle 22 zu finden.

4.3 Multivariate Zeitreihenanalyse

In diesem Abschnitt werden die Ergebnisse des Modelltrainings, der Modellevaluierung und des Forecasts mit den SARIMAX- und DeepAR-Modellen der Multivariate-Zeitreihenanalyse erläutert.

4.3.1 SARIMAX

Mit den Parametern, die in der Tabelle 7 auf Seite 46 beschrieben wurden, wurde das SARIMAX-Modell trainiert. Die Modelzusammenfassung des trainierten SARIMAX-Modells sieht folgendermaßen aus:

SARIMAX Results						
Dep. Variable:	Germany			No. Observations:	23	
Model:	SARIMAX(4, 0, 1)x(0, 0, [1, 2, 3], 12)			Log Likelihood	12.218	
Date:	Tue, 08 Nov 2022			AIC	33.564	
Time:	17:50:26			BIC	66.493	
Sample:	01-01-2020 - 11-01-2021			HQIC	41.846	
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
Austria	0.7034	2.126	0.331	0.741	-3.463	4.870
Belgium	0.0354	6.270	0.006	0.995	-12.254	12.324
Denmark	-0.1937	0.232	-0.834	0.404	-0.649	0.261
Finland	-0.0147	0.717	-0.021	0.984	-1.419	1.390
France	-0.1546	2.052	-0.075	0.940	-4.176	3.867
Hungary	0.0584	7.053	0.008	0.993	-13.766	13.882
Iceland	-0.0542	0.405	-0.134	0.893	-0.847	0.739
Ireland	0.0478	0.809	0.059	0.953	-1.537	1.633
Italy	0.2320	1.809	0.128	0.898	-3.313	3.777
Liechtenstein	-0.2242	0.766	-0.293	0.770	-1.726	1.277
Lithuania	-0.2181	1.894	-0.115	0.908	-3.930	3.494
Luxembourg	0.4124	2.128	0.194	0.846	-3.758	4.583
Netherlands	0.0075	0.463	0.016	0.987	-0.899	0.914
Norway	0.0226	0.150	0.151	0.880	-0.271	0.316
Portugal	0.2725	3.323	0.082	0.935	-6.240	6.785
Romania	0.3611	3.382	0.107	0.915	-6.267	6.990
Slovenia	0.2254	2.691	0.084	0.933	-5.050	5.500
Sweden	-0.0448	0.873	-0.051	0.959	-1.756	1.667
Switzerland	0.0943	2.305	0.041	0.967	-4.423	4.612
United Kingdom	-0.2037	1.313	-0.155	0.877	-2.778	2.370
ar.L1	-0.0594	1.108	-0.054	0.957	-2.230	2.111
ar.L2	-0.2619	6.675	-0.039	0.969	-13.345	12.821
ar.L3	0.0792	1.120	0.071	0.944	-2.116	2.275
ar.L4	-0.3235	3.075	-0.105	0.916	-6.350	5.703
ma.L1	-0.0624	4.558	-0.014	0.989	-8.996	8.871
ma.S.L12	0.0015	3.610	0.000	1.000	-7.073	7.076
ma.S.L24	-0.0014	0.317	-0.004	0.996	-0.622	0.619
ma.S.L36	-0.0002	0.005	-0.029	0.976	-0.011	0.011
sigma2	0.0211	0.032	0.659	0.510	-0.042	0.084

Abbildung 23: Modellzusammenfassung des SARIMAX-Modells

In der Abbildung 23 sind die Informationen der SARIMAX-Modellzusammenfassung markiert, die die Ergebnisse dieser ausgewählten Parameter wiedergeben. Die trainierte Parameterkombination des SARIMAX-Modells ist rosafarben markiert. Die Kombination $(4,0,1) \times (0,0,3,12)$ entspricht denselben Parameterwerten, die in der Tabelle 7 beschrieben sind. Mit anderen Worten stehen diese Werte für die p-,d-,q-,P-,D-, und Q sowie die m-Komponente

des SARIMAX-Modells. Der AIC-Wert ist mit lila markiert und ist für die Informationskriterien zur Auswertung der Modellgüte zuständig, allerdings ist nur beim SARIMA(X)-Modell implementierbar. Die Lag-Variable des AR- und MA-Modells und deren Koeffizienten sind mit orange markiert. Die exogenen Variablen und deren Koeffizienten sind mit blau markiert. Diese Werte werden entsprechend den p-, q-, P- und Q-Komponenten dargestellt. Die p-Values sind mit grün markiert und zeigen die statistische Signifikanz der Lag-Variablen. Die gegebenen Lag-Variablen werden als wichtig betrachtet, wenn der Wert der p-Komponente kleiner als 0,05% ist. Im trainierten SARIMAX-Modell sind alle Variablen statistisch nicht-signifikant, weil ihre p-Werte größer als 0,05% sind. Allerdings ist es in der Praxis bei einem Modell mit vielen exogenen Variablen normal, insbesondere wenn ein niedriger Fehlerwert gewünscht ist. Das Auto-ARIMA-HPO-Ergebnis beinhaltet eine Modellkombination mit den p-Values kleiner als 0,05%, aber die RMSE- und Prognose-Werte sind nicht besser als bei Optuna und Grid-Search.

Die Optuna-HPO zeigt eine Parameterkombination mit dem RMSE-Wert von 2,27 Prozentpunkten. Dagegen weist Grid-Search einen RMSE-Wert von 2,15 Prozentpunkten auf. Jedoch haben die gleichen Grid-Search-Parameterwerte während der Modellevaluierung einen RMSE-Wert von 2,25 Prozentpunkten dargestellt, dass dennoch besser als die Optuna-HPO-Ausgabe ist. Durch Grid-Search wurde 15.625 Parameterkombinationen gesucht. Mit Hilfe des niedrigsten RMSE-Werts wurde die Parameterkombination des Grid-Search ausgewählt. In Optuna wurde danach der Optimierungsverlauf der Fehlerwerte innerhalb der gegebenen Trial-Anzahl visualisiert.

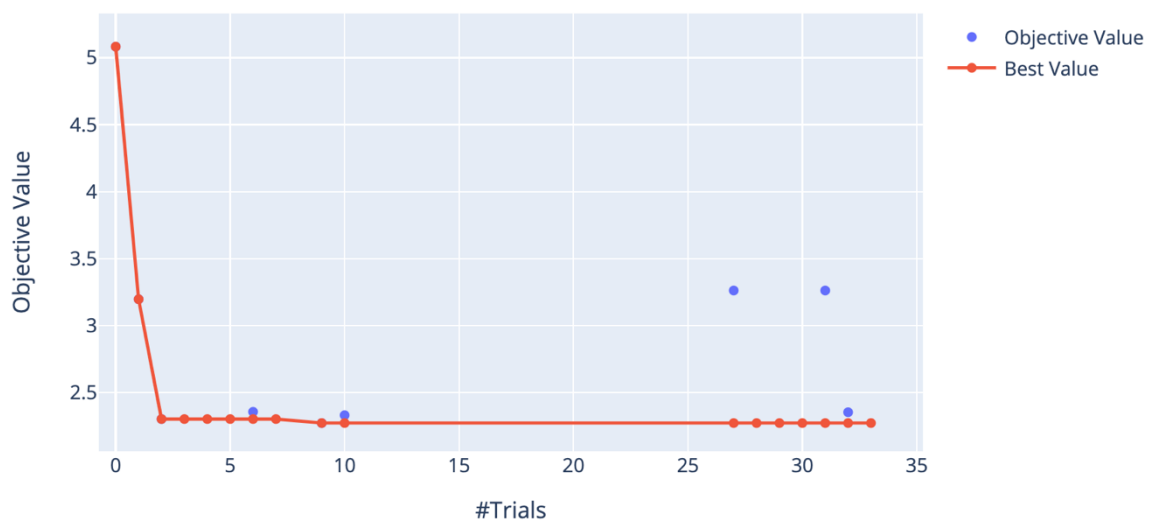


Abbildung 24: Optimierungsverlaufs des SARIMAX-Modells

Abbildung 24 zeigt den Optimierungsverlauf der Optuna-HPO für das SARIMX-Modell. Die Objective Values, markiert mit blauen Punkten, sind die RMSE-Fehlermetriken. Der Best Value stellt den niedrigsten RMSE-Wert mit der roten Linie dar. Obwohl 100 Trials zur HPO angegeben wurden, liefen nur 33 Trials erfolgreich, weil die restlichen durchsuchten Trials bzw. Parameterkombinationen wegen der geringen Anzahl des multivariaten Datensatzes und des Umfangs der Parameterwerte nicht fehlerfrei durchlaufen konnten. Darüber hinaus sind einige Objective-Values innerhalb der 33 Trials in der Abbildung 24 gezeigt, weil die restlichen Werte zu hoch und dadurch der Best-Values undeutlich ist. Aus diesem Grund musste die Visualisierung vergrößert werden, damit das Ergebnis genauer dargestellt werden konnte. Optuna konnte dennoch innerhalb der ersten 10 Versuche die optimale Parameterkombination finden. Die Wichtigkeit der Hyperparameter der optimalen Parameterkombination wird wie folgt dargestellt:

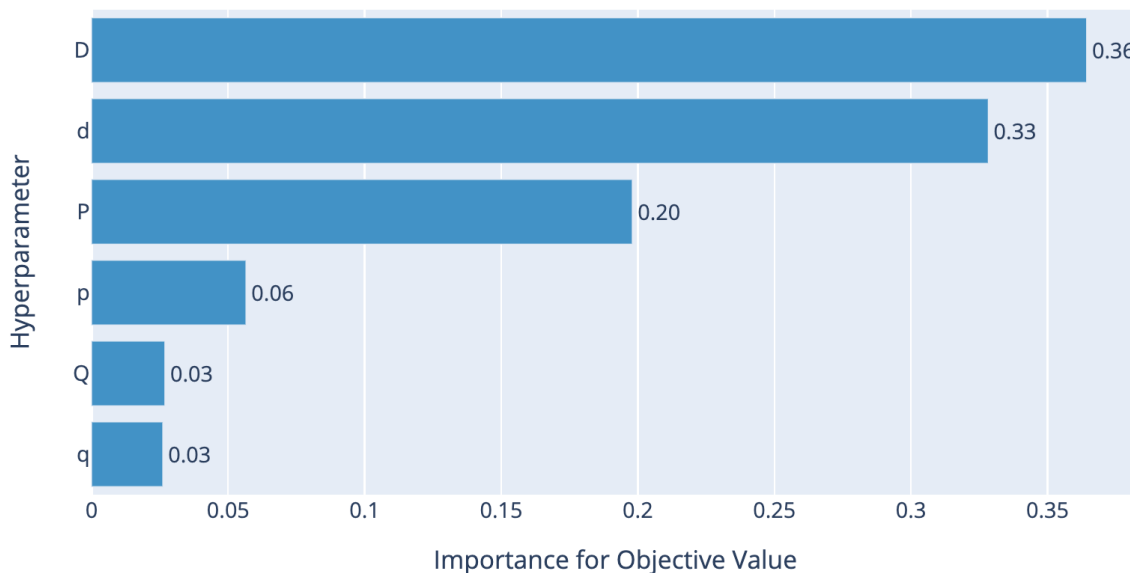


Abbildung 25: Hyperparameterwichtigkeit des SARIMAX-Modells

Die drei wichtigsten Parameter entsprechend der Optuna-HPO sind die D-, d- und p-Komponente jeweils mit 36%, 33% und 20%, wie in der Abbildung 25 dargestellt. Das bedeutet, dass die Parameter D, d und p insgesamt 89% Einfluss auf das SARIMAX-Modelltraining haben. Die anderen Parameter nehmen mit 6% deutlich weniger Einfluss auf das Modelltraining. Die Modell-Diagnostik des SARIMAX-Modells stellt sich folgendermaßen dar:

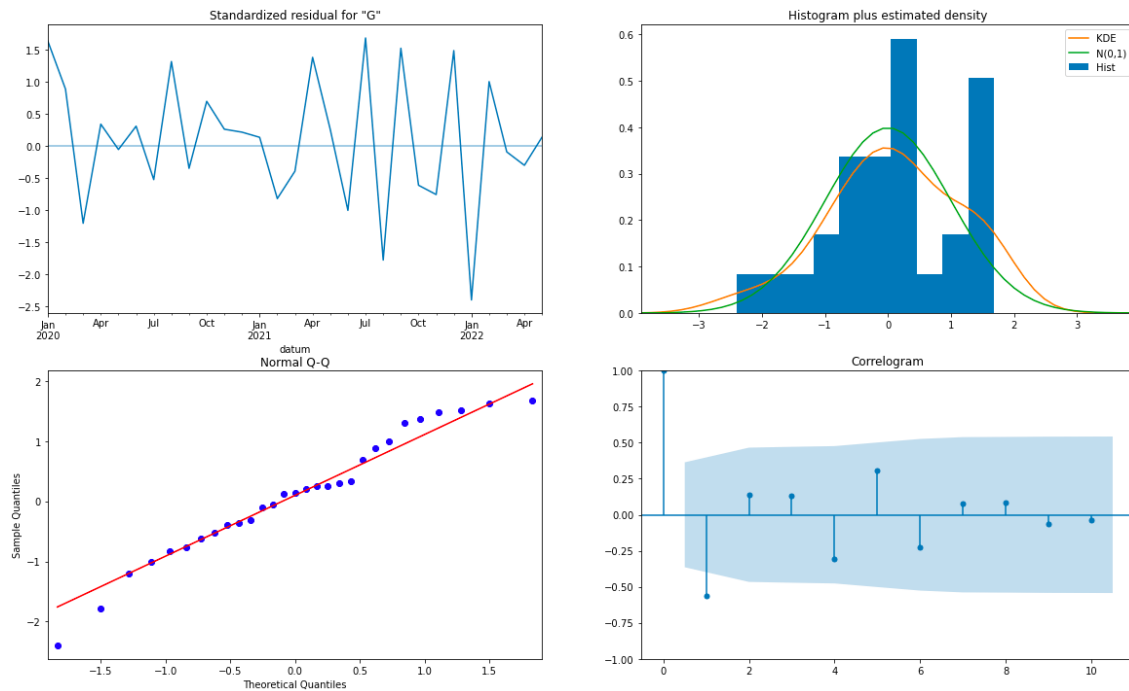


Abbildung 26: Modell-Diagnostik des SARIMAX-Modells

Die Visualisierung oben links veranschaulicht das Residual und enthält einen Mittelwert von 0, während die Visualisierung oben rechts das Density Plot des Residuals zeigt. Das Density Plot des Residuals stellt eine normal Verteilung dar, daher wird das SARIMAX-Modell als optimal betrachtet. Die Visualisierung unten links präsentiert das Modell-Residual in einem Q-Q-Plot und zeigt auf, dass alle blauen Punkten an der roten Linie liegen. Je näher die blauen Punkte an dieser roten Linie verteilt sind, desto besser ist das Modell trainiert. Die letzte Visualisierung rechts unten zeigt das Korrelogramm bzw. ACF-Plot des Residuals. Eine Autokorrelation im Korrelogramm würde bedeuten, dass das Residual ein Muster aufweist (Prabhakaran, 2021), das im trainierten Modell nicht miteinbezogen wurde. Aus diesem Grund muss das trainierte Modell mit besseren Parameterwerten erneut optimiert und erneut trainiert werden. Das Korrelogramm und Residual weisen keine Korrelation und kein Muster auf. Im Folgenden werden das trainierte SARIMAX-Modell evaluiert.

4.3.1.1 Modellevaluierung

Die Abbildung 27 zeigt die Modellevaluierung des SARIMAX-Modells, wobei der Testdatensatz zur Evaluierung und der Trainingsdatensatz zum Modelltraining verwendet wurden.

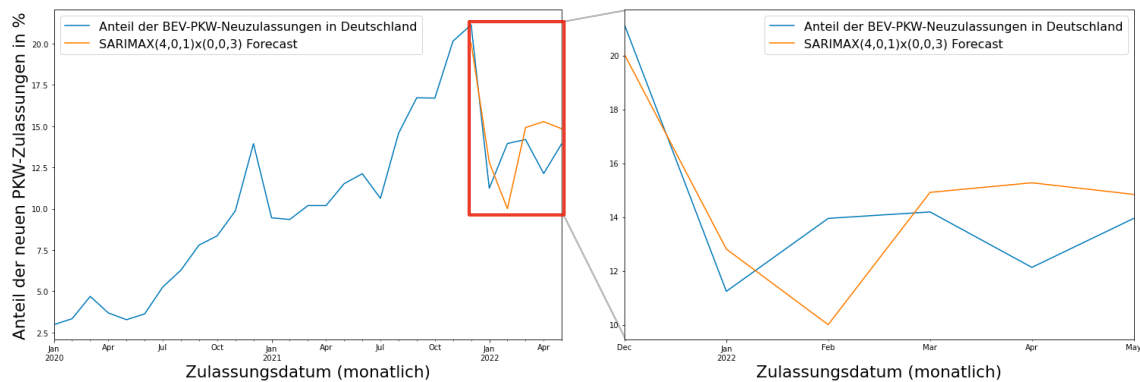


Abbildung 27: SARIMAX-Modellevaluierung

Die Visualisierung auf der linken Seite der Abbildung 27 stellt sowohl die Trainings- und Testdatensätze als auch die Prognose im Prognosehorizont von sechs Monaten dar, während die Visualisierung auf der rechten Seite der Abbildung 27 die Prognose gegen den Testdatensatz aufweist. Entsprechend der orangenen Linie in der rechten Visualisierung kann festgestellt werden, dass die prognostizierten Werte nahe an den Punkten des Testdatensatzes liegen.

Der RMSE-Wert des trainierten SARIMAX-Modells beträgt folglich 2,25 Prozentpunkte. Das bedeutet, dass die prognostizierten Werte durch das Modell durchschnittlich eine Abweichung von 2,25 prozentualem Marktanteil der BEV-PKW-Neuzulassungen in Deutschland haben. Im Vergleich zum SARIMA-Modell, wobei der RMSE-Fehler 0,88 Prozentpunkte beträgt, ist das SARIMAX-Modell nicht optimal. Trotz der kleinen Eingabedatensätze, insgesamt 23 Zeitreihen zum Modelltraining, ist das ein akzeptables Ergebnis.

4.3.1.2 Prognose

Um eine Prognose mit dem SARIMAX-Modell zu erstellen, müssen die exogenen Variablen angegeben werden. Die Bereitstellung der exogenen Variablen stellt hier eine Herausforderung dar, da die prozentualen Marktanteile der BEV-PKW-Neuzulassungen für Deutschland und den anderen Ländern nicht im Voraus bekannt sind. Eine theoretische Alternative ist die Erstellung einer separaten Prognose für exogenen Variablen, aber dafür müssten zunächst 20 separate Univariate-Zeitreihenanalyse-Modelle erstellt werden, was unpraktisch und zeitaufwändig ist.

Prabhakaran hat die Anwendung der Saisonalität der letzten Zeitreihen entsprechen des Prognosehorizonts vorgeschlagen (Prabhakaran, 2021). Jedoch hat die Saisonalität der letzten 6 Monaten von exogenen Variablen für dieses Modell ein schlechtes Ergebnis gezeigt, weil die Datensätze der BEV-

PKW-Neuzulassungen in Deutschland nicht nur Saisonalität, sondern auch einen nicht-linearen Trend enthalten.

Darüber hinaus haben Elamin and Fukushige eine andere Methode zur Erstellung der exogenen Variablen verwendet, nämlich die Multiplikation der spezifischen zur Verfügung stehenden Variablen (Elamin and Fukushige, 2018, pp. 1–6). Um die gleiche Methode nachzuahmen, wurde die Saisonalität des prozentualen Marktanteil der BEV-PKW-Neuzulassungen in Deutschland vom Juni 2021 bis November 2021 extrahiert, weil der gleiche Zeitraum im Jahr 2022 prognostiziert werden soll. Die extrahierte Saisonalität vom Jahr 2021 wurde demnach mit den sechs Zeitreihen der exogenen Variablen des Eingabedatensatzes, den bereinigten Daten von 20 Ländern, multipliziert. Allerdings hat dieser Versuch ebenfalls schlechtere Prognosen mit sich gebracht.

Als letzte Möglichkeit kann die Anwendung von ähnlichen Daten für die exogenen Variablen zum Einsatz kommen, jedoch stehen derzeit solche Datensätze nicht zur Verfügung. Infolgedessen konnte aus diesem Grund keine Prognose erstellt werden. Im folgenden Abschnitt wird das Ergebnis des DeepAR-Modells der Multivariate-Zeitreihenanalyse detailliert erläutert.

4.3.2 DeepAR

Das DeepAR-Modell der Multivariate-Zeitreihenanalyse wurde mit den Parametern in der Tabelle 9 auf Seite 48 trainiert. Diese Parameterkombination wurde durch die Optuna-HPO ausgewählt, weil die Grid-Search-Ausgabe trotz niedrigen RMSE-Wertes (0,80 Prozentpunkte) schlechte Prognosen erstellt hat. Der Optimierungsverlauf der Fehlerwerte der Optuna-HPO lässt sich wie folgt visualisieren:

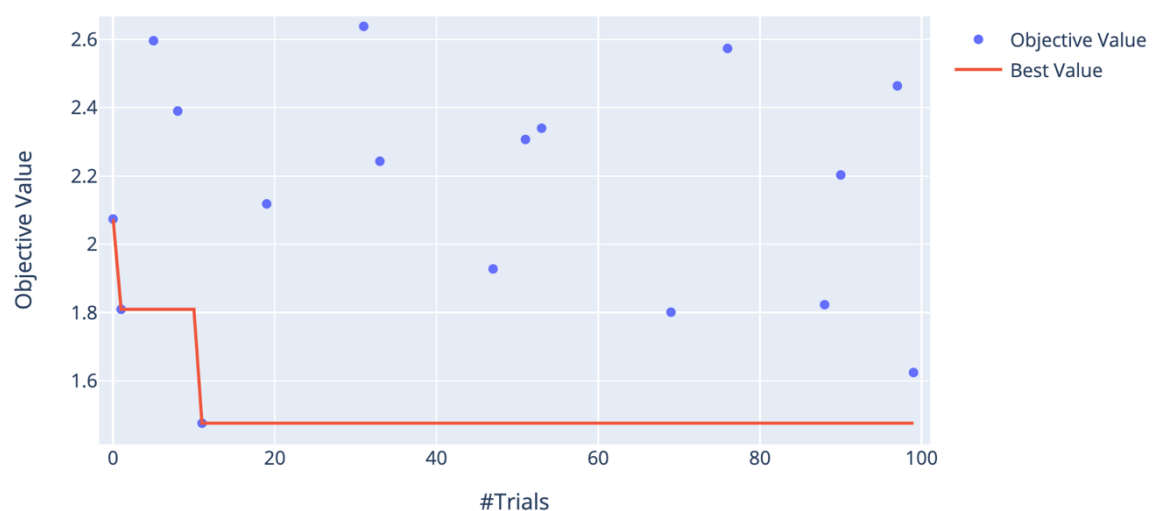


Abbildung 28: Optimierungsverlaufs des Multivariate-DeepAR-Modells

In der Abbildung 28 wird der Best-Value durch die rote Linie dargestellt, der den niedrigsten RMSE-Wert während der Optuna-HPO abbildet. Optuna konnte innerhalb der ersten 20 Trials die optimale DeepAR-Parameterkombination mit dem niedrigsten RMSE-Wert finden, obwohl 100 Trials angegeben wurden. Die blauen Punkte bilden die Objective-Values ab, die die nicht-optimalen Parameterkombinationen aufweisen. Die Wichtigkeit der Hyperparameter der optimalen Parameterkombination wird wie folgt visualisiert:

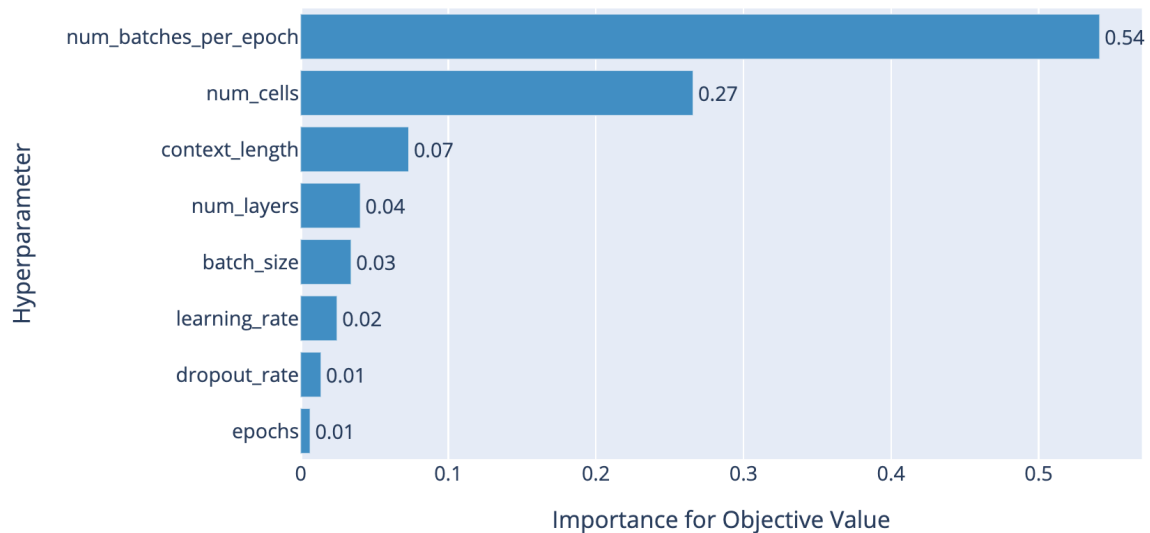


Abbildung 29: Hyperparameterwichtigkeit des Multivariate-DeepAR-Modells

Die Parameter *num_batches_per_epoch*, *num_cells* und *context_length* sind jeweils mit 54%, 27% und 7% die wichtigsten drei Parameter beim DeepAR-Modelltraining, wie in der Abbildung 29 dargestellt. Das bedeutet, dass die Parameter *num_batches_per_epoch*, *num_cells* und *context_length* insgesamt 88% Einfluss auf das DeepAR-Modelltraining haben. Die restlichen Parameter zum DeepAR-Modelltraining der Univariate-Zeitreihenanalyse haben eine Wichtigkeit von maximal 4%.

4.3.2.1 Modellevaluierung

Um die Güte des DeepAR-Modells zu evaluieren, wurde der Testdatensatz genutzt. Folgende Abbildung zeigt die Modellevaluierung des DeepAR-Modells der Multivariate-Zeitreihenanalyse:

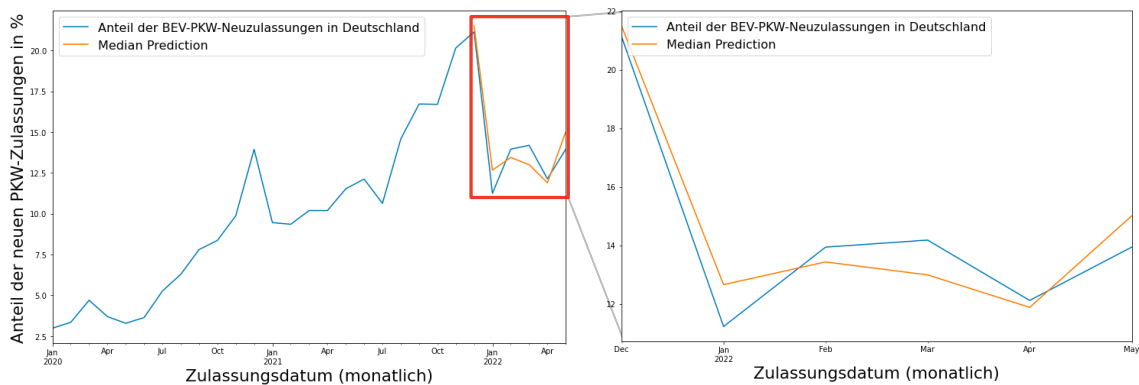


Abbildung 30: Modellevaluierung des Multivariate-DeepAR-Modells

Abbildung 30 zeigt die Modellevaluierung des DeepAR-Modells gegen den Testdatensatz. Die Visualisierung auf der linken Seite stellt die Trainings- und Testdatensätze mit der Prognose im Prognosehorizont von sechs Monaten dar. Die Visualisierung auf der rechten Seite zeigt die Prognose im Vergleich zum Testdatensatz. Die prognostizierten Werte, die orangefarbene Linie, liegen nahe an den Punkten des Testdatensatzes, die blaue Linie. Das endgültige DeepAR-Modell hat einen RMSE-Wert von 1,48 Prozentpunkten. Das bedeutet, dass ein durchschnittlicher Fehler von 1,48 Prozentpunkten bei den prognostizierten Werten zum prozentualen Marktanteil der BEV-PKW-Neuzulassungen erwartet werden kann. Die Prognose des prozentualen Marktanteils der BEV-PKW-Neuzulassungen in Deutschland wird im Folgenden erläutert.

4.3.2.2 Prognose

Nach der Modellevaluierung und entsprechend des RMSE-Werts von 1,48 Prozentpunkten wurde eine Prognose auf sechs Monate erstellt. Dafür wurde der gesamte Eingabedatensatz mit den gleichen Parametern verwendet, um das DeepAR-Modell erneut zu trainieren. Die Prognose des DeepAR-Modells der Multivariate-Zeitreihenanalyse sieht folgendermaßen aus:

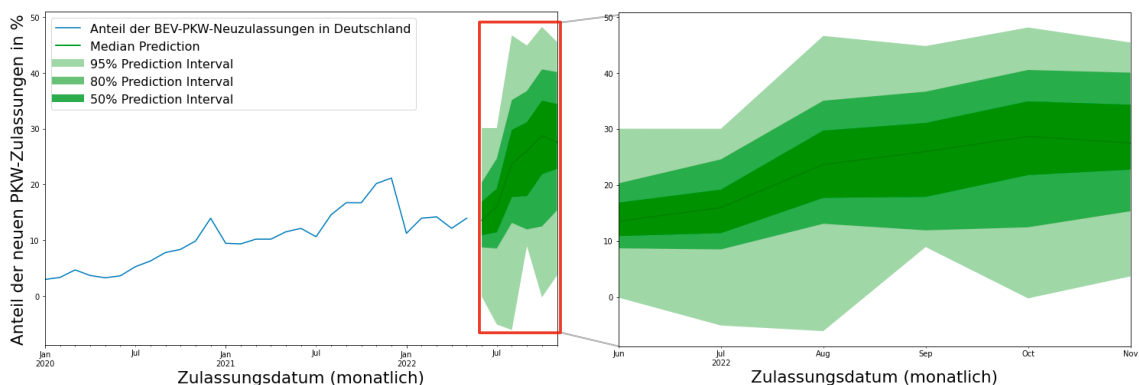


Abbildung 31: Prognose des Multivariate-DeepAR-Modells

Abbildung 31 zeigt die Prognose der nächsten 6 Monaten des prozentualen Marktanteils der BEV-PKW-Neuzulassung in Deutschland vom Juni 2022 bis einschließlich November 2022. Die Visualisierung auf der linken Seite zeigt den gesamten Eingabedatensatz und die Prognose der nächsten 6 Monaten mit den Medianwerten sowie den Konfidenzintervalle von 95%, 80% und 50%. Auf der rechten Seite kann diese Visualisierung im Detail betrachtet werden. Das Konfidenzintervall von 95% bedeutet, dass statistisch in 95% der Fälle die zukünftigen Werte im Bereich des gegebenen Intervalls liegen wird. Das Konfidenzintervall von 95% sieht sehr bereit aus. Zum Beispiel sehen die ersten drei Monate, also Juni, Juli und August 2022, unrealistisch aus, weil die Werte des Lower-Bound unter 0% und Upper-Bound in einigen Monaten über 40% liegen.

Dagegen sieht das Konfidenzintervall von 80% realistischer aus, weil die prognostizierten Werte des Lower-Bound und Upper-Bound eng beieinanderliegen. Das Lower-Bound im Juni 2022 des 80%-Konfidenzintervalls liegt bei 8,73% und das Upper-Bound bei 20,33%. Das bedeutet, dass das Intervall von 8,73% bis 20,33% zu 80% den wahren Wert des Marktanteils im Juni 2022 enthält. Das gleiche gilt für das 50%-Konfidenzintervall, dort liegt das Lower-Bound und Upper-Bound zur Prognose im Juni 2022 jeweils 10,92% und 16,87%. Das bedeutet, dass das Intervall von 10,92% bis 16,87% zu 50% den wahren Wert des Marktanteils im Juni 2022 enthält. Die vollständigen Prognosewerte durch das DeepAR-Modell der Multivariate-Zeitreihenanalyse sind im Anhang in der Tabelle 23 zu finden.

4.4 Vergleich der HPO-Methoden und der Modelle

Um die implementierten Modelle der Univariate- und Multivariate-Zeitreihenanalyse dieser Abschlussarbeit miteinander vergleichen zu können, werden in der folgenden Tabelle die RMSE-Werte der zwei implementierten HPO-Methoden, nämlich Grid-Search und Optuna präsentiert.

HPO Methode	SARIMA	SARIMAX	DeepAR Univariate- Zeitreihenanalyse	DeepAR Multivariate- Zeitreihenanalyse
Grid-Search	0,88	2,25	0,66	0,8
Optuna	1,29	2,27	1,29	1,48

Tabelle 14: RMSE-Werte der HPO

In der Tabelle 14 sind die RMSE-Werte der Grid-Search- und Optuna-HPO aufgezeigt. Die in orange dargestellten RMSE-Werte zeigen die HPO-Ausgabewerte, die zum Modelltraining nicht ausgewählt wurden. Die RMSE-Werte waren u. a. entweder größer als die Optuna-RMSE-Werte oder haben trotz geringerer Werte schlechte Prognosen erstellt. Zum Beispiel waren die RMSE-Werte des Grid-Search beim DeepAR-Modell 0,66 Prozentpunkte, jedoch zeigte das trainierte endgültige DeepAR-Modell einen RMSE-Wert größer als der Optuna-RMSE-Wert an, weil die Grid-Search eine eingeschränkte Anzahl der Parameterwerte enthielt.

Generell kann festgestellt werden, dass die RMSE-Werte der Grid-Search-HPO-Ausgabe für das Training der SARIMA- und SARIMAX-Modelle geeignet sind, während die RMSE-Werte der Optuna-HPO-Ausgabe eher für das DeepAR-Modelltraining nutzbar ist. Der Hauptgrund ist die Laufzeit des Grid-Search. Grid-Search zur Optimierung des DeepAR-Modells dauert sehr lang, z.B. hat Grid-Search des DeepAR-Modells der Univariate-Zeitreihenanalyse 10,5 Tage gedauert. Aus diesem Grund wurde die Grid-Search-HPO-Ausgabewerte beim DeepAR-Modelltraining nicht verwendet, da die eingeschränkte Anzahl der Parameterwerte zu den schlechten Prognosen geführt haben. Beim Modelltraining der SARIMA- und SARIMAX-Modelle ist die Grid-Search-Laufzeit akzeptabel, obwohl die HPO des SARIMA-Modells ca. 3 Tage gedauert hat. In der Tabelle 14 wurde Auto-ARIMA nicht miteinbezogen, weil einerseits die Auto-ARIMA-Ausgabe zu keinem Modelltraining verwendet wurde, andererseits unterstützt Auto-ARIMA das DeepAR-Modell nicht. Die endgültigen trainierten Forecasting-Modelle enthielten für die Modellevaluierung folgende RMSE-Werte:

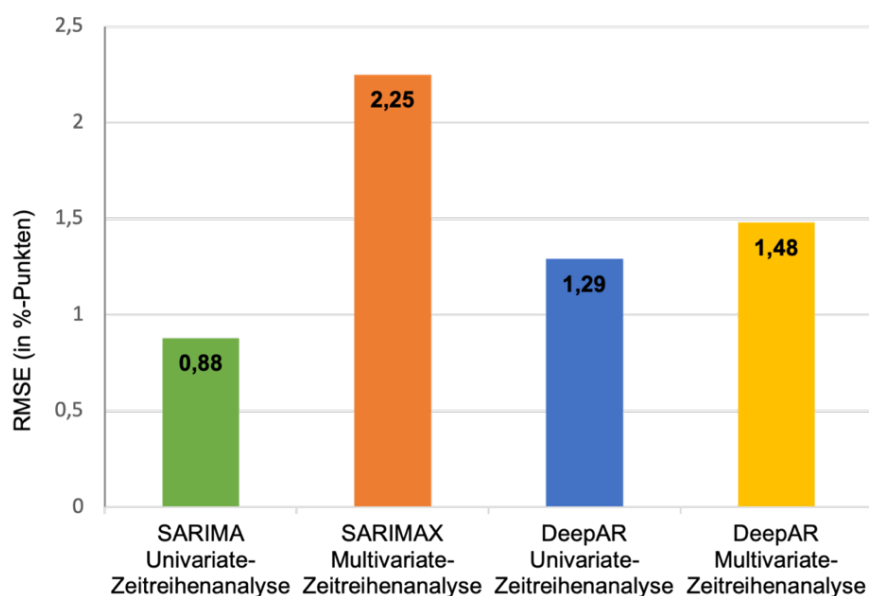


Abbildung 32: RMSE-Werte der trainierten Modelle

Abbildung 32 zeigt die RMSE-Werte der vier trainierten Zeitreihenanalyse-Modelle. Bei der Univariate-Zeitreihenanalyse hat das SARIMA-Modell den niedrigsten RMSE-Fehler (0,88 Prozentpunkte), während bei Multivariate-Zeitreihenanalyse das DeepAR-Modell der klare Gewinner (1,48 Prozentpunkte) ist. Zwischen dem SARIMA- und dem DeepAR-Modell der Univariate-Zeitreihenanalyse beträgt die Differenz nur 0,41 Prozentpunkte RMSE, obwohl DeepAR für die längeren Datensätze besser geeignet ist. Jedoch beträgt der Unterschied zwischen dem SARIMAX- und dem DeepAR-Modell der Multivariate-Zeitreihenanalyse 0,77 Prozentpunkte RMSE.

5 Diskussion und Zusammenfassung

In diesem Kapitel werden die Ergebnisse dieser Arbeit über das Forecasting des prozentualen Marktanteils der BEV-PKW-Neuzulassungen in Deutschland zusammengefasst. Zunächst werden die eigenen Ergebnisse reflektiert und anschließend die Einschränkungen für diese Masterthesis beschrieben. Eine Zusammenfassung sowie ein Ausblick runden diese Arbeit ab.

5.1 Reflexion der Ergebnisse

Nach der Implementierung der Univariate- und Multivariate-Zeitreihenanalyse zum Forecasting des prozentualen Marktanteils der neuen BEV-PKW-Zulassungen in Deutschland werden die Ergebnisse des vierten Kapitels im Folgenden reflektiert.

Es wurden drei HPO-Methoden implementiert, Auto-ARIMA, Grid-Search und Optuna. Im Folgenden werden aber nur Optuna und Grid-Search zu Vergleichszwecken beschrieben, da Auto-ARIMA nur ARIMA-basierte Modelle optimieren kann. Die Optuna-HPO war effektiver als Grid-Search, weil Optuna in weniger Trials und kürzerer Zeit die passende Parameterkombination finden konnte und dadurch sowohl Ressourcen als auch viel Zeit sparte. Obwohl Grid-Search die optimale Parameterkombination finden konnte, hat diese aber auch Nachteile. Zum Beispiel dauert der Optimierungsprozess des Grid-Search sehr lang, vor allem beim DeepAR-Modell, wohingegen ähnliche Ergebnisse durch Optuna sehr viel schneller erzielt werden können. Die Optuna-HPO funktioniert mit allen Machine-Learning-Modellen, allerdings ist Optuna bei der Optimierung des DeepAR-Modells auf Grund der HPO-Laufzeit besser geeignet. Die HPO hat einen großen Einfluss auf das Modelltraining und die Modellauswahl, weil dadurch die passenden Modellparameterwerte ausgewählt werden.

Beim Modelltraining und bei der Prognose nur einer Variable, in dieser Arbeit der prozentuale Marktanteil der neuen BEV-PKW-Zulassungen in Deutschland, bietet das SARIMA-Modell der Univariate-Zeitreihenanalyse bessere Ergebnisse. Einerseits ist die Modelltrainingszeit kürzer, andererseits ist der

Prognosefehler, nämlich der RMSE-Wert, geringer als beim DeepAR-Modell der Univariate-Zeitreihenanalyse. Der Unterschied der RMSE-Werte zwischen den SARIMA- und DeepAR-Modellen der Univariate-Zeitreihenanalyse beträgt 0,41 Prozentpunkte, wie in der Tabelle 14 auf Seite 73 dargestellt. Das bedeutet, dass der Fehler des SARIMA-Modells 0,41 Prozentpunkte kleiner ist als der RMSE-Wert des DeepAR-Modells. Jedoch wurde beim SARIMA-Modelltraining die Parameterkombination des Grid-Search und beim DeepAR-Modelltraining die Parameterkombination der Optuna-HPO verwendet. Falls auf Grund der HPO-Laufzeit die Optuna-Parameterkombination zum Modelltraining der Univariate-Zeitreihenanalyse verwendet würde, wäre der RMSE-Wert des SARIMA-Modells auch 1,29 Prozentpunkte. In dem Fall hätte das DeepAR- und SARIMA-Modell der Univariate-Zeitreihenanalyse die gleichen RMSE-Werte und würden sehr ähnliche Ergebnisse zeigen.

Die Situation beim Modelltraining und bei der Prognose mit mehreren Variablen ist allerdings umgekehrt, wobei die Zielvariable der prozentuale Marktanteil der neuen BEV-PKW-Zulassungen in Deutschland ist. Das DeepAR-Modell der Multivariate-Zeitreihenanalyse ist viel effizienter und bietet bessere Ergebnisse mit niedrigerem RMSE-Wert als das SARIMAX-Modell an. Der Unterschied zwischen den SARIMAX- und DeepAR-Modellen der Multivariate-Zeitreihenanalyse beträgt 0,77 Prozentpunkte RMSE, wie in der Tabelle 14 auf Seite 73 dargestellt. Das bedeutet, dass der Fehler des SARIMAX-Modells 0,77 Prozentpunkte größer als beim DeepAR-Modell ist, was fast die doppelte Differenz zwischen dem SARIMA- und dem DeepAR-Modell der Univariate-Zeitreihenanalyse ist. Das DeepAR-Modell erstellt ein globales Modell, deswegen erkennt es die Zusammenhänge zwischen den Variablen effizienter. Diese Eigenschaft des DeepAR-Modells kann sowohl die Prognosegenauigkeit verbessern als auch eine separate Prognose pro Variable erstellen. Das ist mit dem SARIMAX-Modell nicht möglich, da das SARIMAX-Modell kein globales Modell erstellen kann.

Darüber hinaus müssen die exogenen Variablen bei der Prognose durch das SARIMAX-Modell angegeben werden, sonst kann das SARIMAX-Modell keine Prognose erstellen. Alternativ können ähnliche Daten als exogene Variable verwendet werden oder die exogenen Variablen werden manuell erstellt, z.B. durch die Multiplikation mit den anderen Variablen. Es ist jedoch möglich, dass exogene Daten, die auf diese Weise erstellt wurden, zu schlechten Prognosen führen. Da keine exogenen Variable zum SARIMAX-Modell existierten und aus den erstellten exogenen Variablen auch keine bessere Prognose resultierte, wurde durch das SARIMAX-Modell auch keine Prognose erzeugt.

Dieses Problem mit den exogenen Variablen existiert mit dem DeepAR-Modell nicht, weil das DeepAR-Modell die Zusammenhänge bereits intern erkennt und die Prognose ohne exogene Variablen erstellen kann. Das ist ein großer Vorteil gegenüber dem SARIMAX-Modell der Multivariate-Zeitreihenanalyse, wenn man Prädiktion (und nicht Präskription) anstrebt. Allerdings existiert keine Methode bzw. Funktion im DeepAR-Modell, um die prognostizierten Werte vom trainierten Modell leicht zu extrahieren, das muss aktuell noch manuell gemacht werden. Die Extrahierung der prognostizierten Werte ist beim Einsatz des SARIMA(X)-Modells deutlich leichter.

Für die multivariaten Datensätze, also den prozentualen Marktanteil der neuen BEV-PKW-Zulassungen in Deutschland mit anderen Ländern, liefert das DeepAR-Modell der GluonTS-Library sehr gute Ergebnisse. Trotz der langen Trainingszeit hat das DeepAR-Modell einige Vorteile, insbesondere da keine exogenen Variablen benötigt werden. Darüber hinaus können die univariaten Datensätze mit zusätzlichen Variablen, also die multivariaten Datensätze, beim DeepAR-Modelltraining verwendet werden. Bei Bedarf können auch die Prognosen der zusätzlichen Variablen des globalen DeepAR-Modells extrahiert werden. Die zusätzlichen Variablen können beispielsweise der prozentuale Marktanteil der BEV-PKW-Neuzulassungen anderer Länder oder der prozentuale Marktanteil der BEV-PKW-Neuzulassungen der verschiedenen Zulassungsbezirke in Deutschland sein. Falls diese zusätzlichen multivariaten Datensätze die Daten von verschiedenen Zulassungsbezirken Deutschlands wären, könnte eine Prognose sowohl für die BEV-PKW-Neuzulassung in Deutschland als auch eine für jeden Zulassungsbezirk erstellt werden.

5.2 Einschränkungen

In diesem Abschnitt werden die Einschränkungen dieser Masterarbeit und mögliche Vorschläge für die zukünftigen Implementationen erläutert.

Eine Einschränkung ist die Modellgenauigkeit bezüglich zukünftiger politischer Entscheidungen zur Förderung batterieelektrischer Fahrzeuge. Die BEV-Neuzulassungen werden im Jahr 2023 nur teilweise gefördert, z.B. wird nur batterieelektrische Fahrzeuge maximal 4500 Euro statt wie bisher mit 6000 Euro vom Staat bezuschusst (Jess, 2022). Dies kann den Kaufanreiz der BEV-PKWs in Deutschland negativ beeinflussen, vor allem mit der bereits sehr hohen Inflation im Jahr 2022. Dies kann Auswirkungen auf die Prognose der BEV-PKW-

Neuzulassungen mit sich bringen, die so in den bisherigen Machine-Learning-Modellen nicht enthalten sind.

Darüber hinaus existieren Einschränkungen in den Eingabedatensätzen der Univariate- und Multivariate-Zeitreihenanalyse. Beispielsweise umfasst der univariate Eingabedatensatz zum Zeitpunkt dieser Abschlussarbeit 69 Zeitpunkte, und der multivariate Eingabedatensatz nur 29 Zeitpunkte, weil das Thema batterieelektrische Fahrzeuge noch neu ist. Von denen mussten jedoch bereits 6 Zeitpunkte zur Modellevaluierung genutzt werden. Obwohl die Prognose durch das DeepAR-Modell sehr gut aussieht, wird das DeepAR-Modell standardmäßig für mittelgroßen Datensätze oder für Datensätze mit mindestens 300 Beobachtungen empfohlen (Salinas, Flunkert and Gasthaus, 2017, p. 9; *DeepAR Forecasting Algorithm*, 2022).

Die Erstellung einer Prognose durch das SARIMAX-Modell ist nur mit Hilfe der exogenen Variablen möglich. Da die zukünftigen exogenen Variablen derzeit nicht existieren bzw. aus den manuell erstellten exogenen Variablen schlechte Prognosen resultieren, sollte stattdessen ein anderes Modell zum Einsatz gebracht werden. Die VAR- und VARMA-Modelle, zwei andere ARIMA-basierte Modelle, könnten das Problem mit den exogenen Variablen umgehen. Die VAR- und VARMA-Modelle sind für einen multivariaten Eingabedatensatz, den EAFO-Datensatz, geeignet, weil die Variablen des multivariaten Datensatzes höhere Korrelationen enthalten. Jedoch funktionieren die VAR- und VARMA-Modelle nur mit den multivariaten Datensätze und nicht mit dem univariaten Datensatz dieser Masterarbeit. Deswegen können die VAR- und VARMA-Modelle mit dem Forecasting-Modell der Univariate-Zeitreihenanalyse nicht verglichen werden. Hieraus resultiert der Vorschlag, die VAR- und VARMA-Modelle mit einem geeigneten multivariaten Eingabesatz in einer zukünftigen Abschlussarbeit zu betrachten. Die VAR- oder VARMA-Modelle könnten sogar mit dem DeepVAR-Modell, einem weiteren GluonTS-Modell, verglichen werden und dann die Prognosen der VAR-, VARMA- und DeepVAR-Modelle zur BEV-PKW-Neuzulassungen effektiver ausgewertet werden.

5.3 Zusammenfassung und Ausblick

Forecasting ist eine wichtige Methode zur Entscheidungsfindung für Unternehmen, Organisationen und Regierungen, da sie es ermöglicht, den Ressourcenverbrauch zu reduzieren. Beispielsweise kann der Produktionsablauf entsprechend der Marktnachfrage effizienter geplant, Kundennachfragen

optimaler ausgeglichen bzw. die Kundenzufriedenheit signifikant erhöht werden. Der Forecast des prozentualen Marktanteils der neuen BEV-PKW-Zulassungen in Deutschland kann helfen, um strategische und zukunftsorientierte Entscheidungen, z.B. die vom Bund gesetzten Klimaziele, besser steuern zu können. Zur Durchführung des Forecastings wurde der Einsatz unterschiedlicher Machine-Learning-Methoden ausgenutzt.

Es wurden zwei Machine-Learning-Modelle für das Forecasting des prozentualen Marktanteils der BEV-PKW-Neuzulassungen in Deutschland verwendet: Die SARIMA(X)- und DeepAR-Modelle. Die Eingabedatensätze enthielten sowohl univariate als auch multivariate Eigenschaften. Infolgedessen wurden vier Modelle der Univariate- und Multivariate-Zeitreihenanalyse trainiert, um eine Prognose des prozentualen Marktanteils der neuen BEV-PKW-Zulassungen zu erstellen und die trainierten Modelle miteinander zu vergleichen. Der Prognosehorizont umfasst sechs Monate. Aus diesem Grund wurden die letzten sechs Monate zur Modellevaluierung genutzt. Bei der Univariate-Zeitreihenanalyse war folglich das SARIMA-Modell mit einem knappen RMSE-Vorsprung von 0,41 Prozentpunkten besser als das DeepAR-Modell. Allerdings war bei der Multivariate-Zeitreihenanalyse das DeepAR-Modell mit einem RMSE-Vorsprung von 0,77 Prozentpunkten besser als das SARIMAX-Modell.

Das SARIMAX-Modell benötigt zur Erstellung einer Prognose der multivariaten Datensätze die exogenen Variablen. Darüber hinaus besteht der Vorteil des SARIMA(X)-Modells darin, dass die Modelltrainingszeit kürzer als beim DeepAR-Modell ist. Die Extrahierung der prognostizierten Werte ist beim Einsatz des SARIMA(X)-Modells deutlich einfacher als beim DeepAR-Modell. Jedoch benötigt das DeepAR-Modell keine exogenen Variablen, um eine Prognose erstellen zu können. Wenn der Eingabedatensatz mindestens 300 Beobachtungen beträgt, ist das DeepAR-Modell besser als das SARIMA(X)-Modell. Das DeepAR-Modell unterstützt nicht nur die univariaten- und multivariaten Datensätze, sondern erstellt auch ein globales Modell, wodurch eine Prognose je Zeitreihe erstellt wird. Das ist ein großer Vorteil gegenüber dem SARIMA(X)-Modell, welches dazu nicht in der Lage ist.

In einem nächsten Schritt könnten weitere Zeitreihenanalyse-Modelle zum Forecasting des prozentualen Marktanteils der neuen BEV-PKW-Zulassungen untersucht werden, wie z.B. Prophet und RNN-Modelle. Das Prophet-Modell unterstützt Datensätze mit Saisonalität und das RNN-Modell ist zum Forecasting der Eingabedatensätze dieser Masterarbeit gut geeignet. Aus diesem Grund wäre es sinnvoll, mit den genannten Modellen die Zeitreihenanalyse zu

implementieren und deren Ergebnisse mit dem DeepAR-Modell und SARIMA(X)-Modell zu vergleichen.

Darüber hinaus ist es erstrebenswert, die in dieser Masterarbeit entwickelten Modelle über Schnittstellen bestimmten Nutzer*innen zur Verfügung zu stellen. Hierfür würde es sich eignen, die Modelle in einer Cloud-Umgebung zu deployen und regelmäßig mit neuen Datensätzen zu trainieren und neue Prognosen zu erstellen.

Literaturverzeichnis

3 *Types of Machine Learning You Should Know* (2022) Coursera. Available at: <https://www.coursera.org/articles/types-of-machine-learning> (Accessed: 22 September 2022).

Akiba, T. *et al.* (2019) 'Optuna: A Next-generation Hyperparameter Optimization Framework', in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Available at: <https://doi.org/10.1145/3292500.3330701>.

Alpaydın, E. (2014) *Introduction to Machine Learning*. 3rd edn. The MIT Press.

Bergstra, J. *et al.* (2011) 'Algorithms for hyper-parameter optimization', in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*.

Bi, Q. *et al.* (2019) 'What is Machine Learning? A Primer for the Epidemiologist', *American Journal of Epidemiology*, 188(12), pp. 2222–2239. Available at: <https://doi.org/10.1093/AJE/KWZ189>.

Box, G.E.P. *et al.* (2015) *Time Series Analysis: Forecasting and Control*. 5th edn. Edited by David J. Balding *et al.* Wiley.

Brockwell, P.J. and Davis, R.A. (2002) *Introduction to Time Series and Forecasting*. 2nd edn, Springer-Verlag. 2nd edn. Springer.

Brockwell, P.J. and Davis, R.A. (2016) *Introduction to Time Series and Forecasting*. 3rd edn. Cham: Springer International Publishing (Springer Texts in Statistics). Available at: <https://doi.org/10.1007/978-3-319-29854-2>.

Brownlee, J. (2018) *A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python, Machine Learning Mastery*. Available at: <https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/> (Accessed: 9 September 2022).

Brownlee, J. (2020a) *Probabilistic Model Selection with AIC, BIC, and MDL, Machine Learning Mastery*. Available at: <https://machinelearningmastery.com/probabilistic-model-selection-measures/> (Accessed: 11 November 2022).

Brownlee, J. (2020b) *Time Series Forecasting With Prophet in Python, Machine Learning Mastery*. Available at: <https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/> (Accessed: 1 September 2022).

Bundesregierung passt Förderrichtlinie für Elektrofahrzeuge an (2021) *Die Bundesregierung*. Available at: <https://www.bundesregierung.de/bregde/suche/faq-umweltbonus-1993830> (Accessed: 5 October 2022).

Chai, T. and Draxler, R.R. (2014) 'Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature', *Geoscientific Model Development*, 7(3), pp. 1247–1250. Available at: <https://doi.org/10.5194/GMD-7-1247-2014>.

Chakraborty, P. (2020) *Vector AutoRegression — The Multivariate Time Series Forecasting Method We Must Know*, *Medium*. Available at: <https://cprosenjit.medium.com/multivariate-time-series-forecasting-using-vector-autoregression-3e5c9b85e42a> (Accessed: 8 September 2022).

Daimler und BMW nutzen Schlupflöcher, um ohnehin schwache EU-Flottengrenzwerte für Neuwagen einzuhalten (2021) *Transport & Environment (T&E)*. Available at: <https://www.transportenvironment.org/discover/daimler-und-bmw-nutzen-schlupflocher-um-ohnehin-schwache-eu-flottengrenzwerte-fur-neuwagen-einzuhalten/> (Accessed: 26 September 2022).

Das, N., Robinson, D. and Mankala, V. (2020) *Measuring forecast model accuracy to optimize your business objectives with Amazon Forecast, AWS Machine Learning Blog*. Available at: <https://aws.amazon.com/blogs/machine-learning/measuring-forecast-model-accuracy-to-optimize-your-business-objectives-with-amazon-forecast/> (Accessed: 15 September 2022).

'Das System der CO₂-Flottengrenzwerte für Pkw und leichte Nutzfahrzeuge' (2020) *Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz (BMUV)* [Preprint]. Available at: www.bmu.de (Accessed: 5 October 2022).

David, D. (2020) *Hyperparameter Optimization Techniques to Improve Your Machine Learning Model's Performance*, freeCodeCamp. Available at: <https://www.freecodecamp.org/news/hyperparameter-optimization-techniques-machine-learning/> (Accessed: 16 October 2022).

DeepAR Forecasting Algorithm (2022) AWS. Available at: <https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html> (Accessed: 13 November 2022).

DeepAR Hyperparameters (2022) AWS. Available at: https://docs.aws.amazon.com/sagemaker/latest/dg/deepar_hyperparameters.html (Accessed: 14 October 2022).

Elamin, N. and Fukushige, M. (2018) 'Modeling and forecasting hourly electricity demand by SARIMAX with interactions', *Energy*, 165, pp. 257–268. Available at: <https://doi.org/10.1016/j.energy.2018.09.157>.

European Alternative Fuels Observatory (2022) *European Alternative Fuels Observatory*. Available at: <https://alternative-fuels-observatory.ec.europa.eu/> (Accessed: 19 September 2022).

Fehlermaße: Wie Sie die Güte Ihrer Forecasts auswerten (2022) Jedox GmbH. Available at: <https://www.jedox.com/de/blog/fehlermasze-guete-von-forecasts-ermitteln/> (Accessed: 12 June 2022).

Finanzielle Unterstützung (2022) BMFSFJ. Available at: <https://www.bmfsfj.de/bmfsfj/themen/corona-pandemie/finanzielle-unterstuetzung> (Accessed: 8 September 2022).

Gasthaus, J. and Januschowski, T. (2019) 'Gluon Time Series, an Open-Source Time Series Modeling Toolkit', *Amazon Machine Learning* [Preprint]. Available at: <https://aws.amazon.com/blogs/opensource/gluon-time-series-open-source-time-series-modeling-toolkit/> (Accessed: 13 June 2022).

Generationenvertrag für das Klima (2022) Bundesregierung. Available at: <https://www.bundesregierung.de/breg-de/themen/klimaschutz/klimaschutzgesetz-2021-1913672> (Accessed: 18 November 2022).

Gilliland, M., Tashman, L. and Sglavo, U. (2015) *Business Forecasting*. Edited by M. Gilliland, L. Tashman, and U. Sglavo. Hoboken, NJ, USA: John Wiley & Sons, Inc. Available at: <https://doi.org/10.1002/9781119244592>.

Gnann, T. *et al.* (2022) *Markthochlaufszzenarien für Elektrofahrzeuge - Rückblick und Ausblick bis 2030*. S 05/2022. Karlsruhe. Available at: https://www.isi.fraunhofer.de/content/dam/isi/dokumente/sustainability-innovation/2022/WP05-2022_Markthochlaufszzenarien_E-Fahrzeuge_GNT-final.pdf (Accessed: 26 September 2022).

Hage, S. and Traufetter, G. (2021) *Greenpeace-Kritik: Elektro-Boom bei VW wird durch Eigenzulassungen getrieben*, *DER SPIEGEL*. Available at: <https://www.spiegel.de/wirtschaft/unternehmen/volkswagen-vws-elektro-boom-wird-durch-eigenzulassungen-getrieben-greenpeace-kritik-a-00000000-0002-0001-0000-000175304179> (Accessed: 5 October 2022).

Hennes, O. *et al.* (2021) 'Auswirkungen von CO₂-Preisen auf den Gebäude-, Verkehrs- und Energiesektor', *Zeitschrift für Energiewirtschaft*, 45(2), pp. 91–107. Available at: <https://doi.org/10.1007/S12398-021-00305-0>.

Hou, C. *et al.* (2021) 'A Deep-Learning Prediction Model for Imbalanced Time Series Data Forecasting', 4(4). Available at: <https://doi.org/10.26599/BDMA.2021.9020011>.

Hyndman, R.J. and Athanasopoulos, G. (2018) *Forecasting: Principles and Practice*. 2nd edn, Monash University, Australia. 2nd edn. OTEXTS. Available at: <https://otexts.com/fpp2/> (Accessed: 13 June 2022).

Hyndman, R.J. and Athanasopoulos, G. (2021) *Forecasting: Principles and Practice*. 3rd edn, Monash University, Australia. 3rd edn. OTEXTS. Available at: <https://otexts.com/fpp3/> (Accessed: 12 June 2022).

Hyndman, R.J. and Koehler, A.B. (2005) 'Another look at measures of forecast accuracy'. Available at: <http://www.forecasters.org/data/m3comp/m3comp.htm> (Accessed: 12 June 2022).

Jess, C. (2022) *Elektro-Kaufprämie: Das wird im neuen Jahr 2023 anders*, *Auto Bild*. Available at: <https://www.autobild.de/artikel/e-autos-kaufpraemie-umweltbonus-kuerzung-foerderung-bundesregierung-8535657.html> (Accessed: 5 October 2022).

Keshvani, A. (2013) *Using AIC to Test ARIMA Models*, *CoolStatsBlog*. Available at: <https://coolstatsblog.com/2013/08/14/using-aic-to-test-arima-models-2/> (Accessed: 23 November 2022).

Koehrsen, W. (2018) *A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning, Towards Data Science*. Available at: <https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f> (Accessed: 11 November 2022).

Kraftfahrt-Bundesamt (2022) *Kraftfahrt-Bundesamt*. Available at: https://www.kba.de/EN/Home/home_node.html (Accessed: 19 September 2022).

Leichsenring, S. (2021) *Innovationsprämie: Elektroauto-Förderung wird bis 2025 verlängert, insideevs*. Available at: <https://insideevs.de/news/519080/innovationspramie-elektroautos-verlangerung-2025/> (Accessed: 5 October 2022).

Lewis, N.D.C. (2016) *Deep Time Series Forecasting With Python : An Intuitive Introduction to Deep Learning for Applied Time Series Modeling*.

McNeese, B. (2017) *Interpretation of Alpha and p-Value, SPC Excel*. Available at: <https://www.spcforexcel.com/knowledge/basic-statistics/interpretation-alpha-and-p-value#alpha> (Accessed: 11 November 2022).

Mills, T.C. (2019) *Applied Time Series Analysis: A Practical Guide to Modeling and Forecasting*. 1st edn, Elsevier. 1st edn.

Montgomery, D.C., Jennings, C.L. and Kulahci, M. (2015) *Introduction to Time Series Analysis and Forecasting*. 2nd edn. Edited by D.J. Balding et al. New Jersey: Wiley.

Neapolitan, R.E. and Jiang, X. (2018) *Artificial Intelligence With An Introduction to Machine Learning*. 2nd edn. Edited by R. v. Yampolskiy. CRC, Chapman & Hall.

NOW GmbH (2022). Available at: <https://www.now-gmbh.de/ueber-uns/wer-wir-sind/> (Accessed: 18 November 2022).

Overview – *Electric vehicles: tax benefits & purchase incentives in the European Union* (2022) ACEA - *European Automobile Manufacturers' Association*. Available at: <https://www.acea.auto/fact/overview-electric-vehicles-tax-benefits-purchase-incentives-in-the-european-union-2022/> (Accessed: 26 September 2022).

Pawar, S. (2020a) *Time series Forecasting in Python & R - Part 1*. Available at: <https://pawarbi.github.io/blog/forecasting/r/python/rpy2/altair/2020/04/21/timeseries-part1.html> (Accessed: 13 June 2022).

Pawar, S. (2020b) *Time series Forecasting in Python & R - Part 2*. Available at: https://pawarbi.github.io/blog/forecasting/r/python/rpy2/altair/fbprophet/ensemble_forecast/uncertainty/simulation/2020/04/21/timeseries-part2.html (Accessed: 13 June 2022).

Pedregosa, F. *et al.* (2011) 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, 12(85), pp. 2825–2830. Available at: <http://jmlr.org/papers/v12/pedregosa11a.html> (Accessed: 22 September 2022).

Prabhakaran, S. (2021) *ARIMA Model - Complete Guide to Time Series Forecasting in Python, Machine Learning Plus*. Available at: <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/> (Accessed: 30 October 2022).

Runkler, T.A. (2020) *Data Analytics Models and Algorithms for Intelligent Data Analysis*. 3rd edn. München: Springer Vieweg. Available at: <https://doi.org/https://doi.org/10.1007/978-3-658-29779-4>.

Salinas, D., Flunkert, V. and Gasthaus, J. (2017) 'DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks'. Available at: <https://doi.org/https://doi.org/10.48550/arXiv.1704.04110>.

Seabold, S. and Perktold, J. (2010) 'Statsmodels: Econometric and Statistical Modeling with Python', *Proceedings of the 9th Python in Science Conference [Preprint]*. Available at: <http://statsmodels.sourceforge.net/> (Accessed: 5 October 2022).

Shcherbakov, M.V. *et al.* (2013) 'A Survey of Forecast Error Measures', *World Applied Sciences Journal*, 24(24), pp. 171–176. Available at: <https://doi.org/10.5829/IDOSI.WASJ.2013.24.ITMIES.80032>.

Smigel, L. (2022) *How to Interpret ARIMA Results, Analyzing Alpha*. Available at: <https://analyzingalpha.com/interpret-arima-results> (Accessed: 4 November 2022).

Smith, T.G. (2017) *pmdarima: ARIMA estimators for Python*. Available at: <http://www.alkaline-ml.com/pmdarima> (Accessed: 11 October 2022).

Speth, D. *et al.* (2020) *Future regional distribution of electric vehicles in Germany*. Karlsruhe. Available at: <https://doi.org/https://doi.org/10.5281/zenodo.4026722>.

Taylor, S.J. and Letham, B. (2017) 'Forecasting at Scale', *PeerJ Preprints* 5:e3190v2 [Preprint]. Available at: <https://doi.org/10.7287/peerj.preprints.3190v2>.

Tsay, R.S. (2005) *Analysis of financial time series*. 2nd edn. Wiley. Available at: <https://www.wiley.com/en-us/Analysis+of+Financial+Time+Series%2C+2nd+Edition-p-9780471746188> (Accessed: 18 August 2022).

Tsay, R.S. (2015) *Multivariate Time Series Analysis with R and Financial Applications*, *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*. Chicago: Wiley.

Tune a DeepAR Model (2022) AWS. Available at: <https://docs.aws.amazon.com/sagemaker/latest/dg/deepar-tuning.html> (Accessed: 16 October 2022).

Tuovila, A. (2022) *Forecasting: What It Is, How It's Used in Business and Investing*, *Investopedia*. Available at: <https://www.investopedia.com/terms/f/forecasting.asp> (Accessed: 23 September 2022).

Umfassende Corona-Hilfen für Unternehmen und Soloselbstständige (2022) *Bundesfinanzministerium*. Available at: <https://www.bundesfinanzministerium.de/Content/DE/Standardartikel/Themen/Schlaglichter/Corona/ueberbrueckungshilfe.html> (Accessed: 8 September 2022).

Wanitschke, A. (2021) *Are battery electric vehicles the future?* Technische Universität Berlin.

Zaiontz, C. (2022) 'Time Series Analysis', *Real Statistics Using Excel* [Preprint]. Available at: <https://www.real-statistics.com/time-series-analysis/> (Accessed: 13 June 2022).

Zhou, Z.-H. (2021) *Machine Learning*. Singapore: Springer Singapore. Available at: <https://doi.org/10.1007/978-981-15-1967-3>.

Anhang

Datum	Anteil der BEV-PKW Neuzulassungen in DTLD	Datum	Anteil der BEV-PKW Neuzulassungen in DTLD
2017-01-01	0,55	2019-12-01	2,03
2017-02-01	0,63	2020-01-01	3,04
2017-03-01	0,61	2020-02-01	3,4
2017-04-01	0,49	2020-03-01	4,8
2017-05-01	0,47	2020-04-01	3,84
2017-06-01	0,67	2020-05-01	3,32
2017-07-01	0,64	2020-06-01	3,69
2017-08-01	0,86	2020-07-01	5,33
2017-09-01	0,78	2020-08-01	6,4
2017-10-01	0,8	2020-09-01	7,99
2017-11-01	1,0	2020-10-01	8,44
2017-12-01	1,34	2020-11-01	9,98
2018-01-01	1,03	2020-12-01	14,02
2018-02-01	0,97	2021-01-01	9,61
2018-03-01	1,09	2021-02-01	9,4
2018-04-01	1,01	2021-03-01	10,3
2018-05-01	0,76	2021-04-01	10,37
2018-06-01	0,78	2021-05-01	11,61
2018-07-01	0,79	2021-06-01	12,19
2018-08-01	0,78	2021-07-01	10,77
2018-09-01	1,18	2021-08-01	14,93
2018-10-01	1,34	2021-09-01	17,09
2018-11-01	1,56	2021-10-01	17,1
2018-12-01	1,62	2021-11-01	20,31
2019-01-01	1,75	2021-12-01	21,28
2019-02-01	1,72	2022-01-01	11,35
2019-03-01	1,91	2022-02-01	14,12
2019-04-01	1,53	2022-03-01	14,29
2019-05-01	1,39	2022-04-01	12,3
2019-06-01	1,77	2022-05-01	14,08
2019-07-01	1,79	2022-06-01	14,35
2019-08-01	1,59	2022-07-01	13,99
2019-09-01	2,4	2022-08-01	16,07
2019-10-01	1,75	2022-09-01	19,74
2019-11-01	1,55		

Tabelle 15: Anteil der BEV-PKW-Neuzulassungen in Deutschland

Modelle	Nicht-stationäre Datensätze	Stationäre Datensätze	Multivariate-Zeitreihenanalyse
AR Autoregressive	Nein	Ja	Nein
MA Moving Average	Nein	Ja	Nein
ETS-Modell Error, Trend, Seasonality	Ja*	Nein	Nein
ARMA Autoregressive Moving Average	Nein	Ja	Nein
ARCH- und GARCH (Generalized) Autoregressive Conditional Heteroskedasticity	Nein	Ja	Nein
ARIMA Autoregressive Integrated Moving Average	Ja	Ja	Nein
SARIMA Seasonal Autoregressive Integrated Moving Average	Ja	Ja	Nein
SARIMAX Seasonal Autoregressive Integrated Moving Average Exogenous Regressors	Ja	Ja	Ja
VAR Vector Autoregressive	Nein	Ja	Ja
RNN Recurrent Neural Network	Ja	Ja	Ja
Prophet Facebook's Prophet	Ja	Nein	Ja
DeepAR Deep Autoregressive Neural Network	Ja	Nein	Ja
* Nein = Mit dem Simple Smoothing Ja = Mit dem Double Exponential Smoothing			

Tabelle 16: Vergleich der Forecasting-Modelle

No.	pdq	seasonal_PDQs	RMSE
1	(3, 2, 1)	(1, 0, 0, 12)	0,885
2	(3, 2, 2)	(1, 0, 0, 12)	0,897
3	(2, 2, 2)	(1, 0, 0, 12)	0,944
4	(3, 1, 2)	(1, 0, 0, 12)	0,991
5	(4, 0, 1)	(0, 1, 0, 12)	0,995
6	(2, 1, 1)	(1, 0, 0, 12)	0,995
7	(4, 0, 0)	(0, 1, 0, 12)	1,000
8	(0, 0, 1)	(2, 0, 0, 12)	1,011
9	(3, 0, 1)	(0, 1, 0, 12)	1,013
10	(2, 1, 0)	(1, 0, 0, 12)	1,067
11	(3, 1, 1)	(1, 0, 0, 12)	1,082
12	(3, 1, 0)	(1, 0, 0, 12)	1,086
13	(3, 2, 1)	(2, 0, 0, 12)	1,089
14	(4, 2, 2)	(1, 0, 0, 12)	1,095
15	(3, 2, 0)	(0, 2, 1, 12)	1,113
16	(3, 0, 0)	(0, 1, 0, 12)	1,116
17	(3, 2, 0)	(1, 2, 1, 12)	1,121
18	(3, 2, 0)	(0, 2, 2, 12)	1,124
19	(2, 2, 1)	(1, 0, 0, 12)	1,125
20	(3, 2, 0)	(1, 2, 2, 12)	1,126

Tabelle 17: Die ersten 20 Grid-Search-Ausgabe des SARIMA-Modells

No.	epochs	num_batches per_epoch	batch_size	num_layers	num_cells	RMSE
1	20	30	20	27	40	0,663
2	10	25	20	10	30	0,676
3	20	30	30	15	3	0,948
4	5	25	10	20	5	1,057
5	5	20	5	20	5	1,172
6	25	10	5	8	2	1,185
7	20	30	5	20	40	1,246
8	20	25	20	2	2	1,272
9	15	15	20	10	4	1,318
10	15	25	5	5	10	1,355
11	20	30	5	2	3	1,366
12	10	15	10	27	4	1,370
13	25	15	30	10	15	1,370
14	25	15	20	20	2	1,379
15	25	5	10	20	2	1,446
16	5	10	5	27	2	1,463
17	10	5	5	15	5	1,500
18	25	30	10	5	30	1,504
19	5	10	30	15	5	1,509
20	25	15	5	22	4	1,522

Tabelle 18: Die ersten 20 Grid-Search-Ausgabe des Univariate-DeepAR-Modells

No.	P,d,q	seasonal_P,D,Q,s	RMSE
1	(4, 0, 1)	(0, 0, 3, 12)	2,157
2	(4, 0, 1)	(2, 0, 0, 12)	2,158
3	(4, 0, 1)	(0, 0, 2, 12)	2,163
4	(4, 0, 1)	(0, 0, 4, 12)	2,166
5	(4, 0, 1)	(3, 0, 0, 12)	2,167
6	(4, 0, 1)	(1, 0, 1, 12)	2,168
7	(4, 0, 3)	(2, 0, 4, 12)	2,170
8	(4, 0, 1)	(4, 0, 0, 12)	2,172
9	(4, 0, 3)	(2, 0, 0, 12)	2,175
10	(4, 0, 3)	(0, 0, 0, 12)	2,179
11	(4, 0, 3)	(2, 0, 1, 12)	2,179
12	(4, 0, 3)	(4, 0, 1, 12)	2,181
13	(4, 0, 1)	(4, 0, 1, 12)	2,182
14	(4, 0, 3)	(1, 0, 1, 12)	2,184
15	(4, 0, 3)	(1, 0, 4, 12)	2,185
16	(4, 0, 1)	(0, 0, 0, 12)	2,185
17	(4, 0, 3)	(2, 0, 3, 12)	2,186
18	(4, 0, 0)	(0, 0, 0, 12)	2,188
19	(4, 0, 1)	(0, 0, 1, 12)	2,188
20	(4, 0, 0)	(1, 0, 3, 12)	2,189

Tabelle 19: Die ersten 20 Grid-Search-Ausgabe des SARIMAX-Modells

No.	epochs	num_batches per_epoch	batch size	num_ layers	num_ cells	RMSE
1	25	20	30	10	20	0,802
2	20	30	30	2	30	0,879
3	25	10	30	20	30	0,896
4	20	25	30	5	20	0,898
5	25	15	30	8	40	0,953
6	25	25	20	22	20	0,968
7	25	25	10	10	30	0,986
8	5	10	20	25	4	1,032
9	15	30	30	15	40	1,039
10	25	30	20	5	40	1,054
11	25	10	10	8	30	1,055
12	20	15	5	8	40	1,071
13	25	15	30	8	15	1,093
14	20	20	30	10	20	1,103
15	20	15	20	27	40	1,129
16	25	30	20	27	4	1,129
17	5	25	20	15	3	1,130
18	20	25	20	25	2	1,135
19	25	15	5	27	5	1,137
20	15	25	10	25	30	1,139

Tabelle 20: Die ersten 20 Grid-Search-Ausgabe des Multivariate-DeepAR-Modells

Datum	Predicted Mean	Lower Bound 95%_CI	Upper Bound 95%_CI	Lower Bound 50%_CI	Upper Bound 50%_CI
2022-10-01	19,6	16,75	22,45	18,62	20,58
2022-11-01	21,14	17,41	24,86	19,85	22,42
2022-12-01	22,28	18,06	26,5	20,83	23,73
2023-01-01	16,46	11,5	21,42	14,75	18,16
2023-02-01	18,04	12,47	23,61	16,12	19,95
2023-03-01	18,33	12,27	24,39	16,25	20,42

Tabelle 21: Prognose des SARIMA-Modells

Datum	Predicted Median	Lower Bound 95%_CI	Upper Bound 95%_CI	Lower Bound 50%_CI	Upper Bound 50%_CI
2022-10-01	21,37	13,99	30,75	19,76	24,52
2022-11-01	21,97	14,73	28,25	20,30	24,35
2022-12-01	22,09	13,96	29,53	19,84	23,86
2023-01-01	20,62	16,25	24,87	18,90	21,77
2023-02-01	21,46	13,14	27,93	19,15	23,46
2023-03-01	26,80	16,21	36,17	22,87	30,20

Tabelle 22: Prognose des Univariate-DeepAR-Modells

Datum	Predicted Median	Lower Bound 95%_CI	Upper Bound 95%_CI	Lower Bound 80%_CI	Upper Bound 80%_CI	Lower Bound 50%_CI	Upper Bound 50%_CI
2022-06-01	13,57	-0,09	30,09	8,73	20,33	10,92	16,87
2022-07-01	16,00	-5,07	30,08	8,53	24,62	11,44	19,20
2022-08-01	23,70	-6,07	46,68	13,11	35,10	17,75	29,75
2022-09-01	26,01	8,95	44,86	11,94	36,72	17,93	31,10
2022-10-01	28,72	-0,22	48,18	12,48	40,59	21,82	34,97
2022-11-01	27,55	3,70	45,50	15,35	40,11	22,79	34,37

Tabelle 23: Prognose des Multivariate-DeepAR-Modells

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde.

Potsdam, den 29. November 2022

Unterschrift

A handwritten signature in black ink, consisting of several overlapping loops and horizontal strokes, with a small 'm' visible in the upper right part of the signature.

Taj Mohammad Ghulam Zada