


**Schriftenreihe
Verkehr
Heft 37**

Lena Elisa Schneegans

**Verfahrensentwicklung für Schaltzeitprognosen
an verkehrsabhängigen Lichtsignalanlagen
mit Hilfe maschinellen Lernens**

**Institut für Verkehrswesen
Fachgebiet Verkehrsplanung
und Verkehrssysteme
Universität Kassel**

kassel
university 
press

Schriftenreihe Verkehr der Universität Kassel

Herausgeber:

Institut für Verkehrswesen

Lena Elisa Schneegans

**Verfahrensentwicklung für Schaltzeitprognosen
an verkehrsabhängigen Lichtsignalanlagen
mit Hilfe maschinellen Lernens**

Die vorliegende Arbeit wurde vom Fachbereich Bauingenieur- und Umweltingenieurwesen an der Universität Kassel als Dissertation zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften (Dr.-Ing.) angenommen.

Erster Gutachter: Prof. Dr.-Ing. Robert Hoyer, Universität Kassel

Zweiter Gutachter: Prof. Dr.-Ing. Silja Hoffmann

Tag der mündlichen Prüfung

12. Oktober 2023

Zu dieser Publikation gehört ein Anhang mit weiteren Informationen zu verwandten Projekten, Beschreibungen der betrachteten Knotenpunkte, Beschreibungen zu einzelnen Details der Versuche und weitere Ergebnisse der Versuche, der online veröffentlicht wurde bei KOBRA (Kasseler OnlineBibliothek, Repository und Archiv), dem digitalen Archiv für die wissenschaftlichen Dokumente der Universität Kassel:

<https://doi.org/10.17170/kobra-202401199387>



Diese Veröffentlichung – ausgenommen Zitate und anderweitig gekennzeichnete Teile – ist unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International (CC BY-SA 4.0: <https://creativecommons.org/licenses/by-sa/4.0/deed.de>) lizenziert.

Zugl.: Kassel, Univ., Diss. 2023

ISBN: 978-3-7376-1162-6

DOI: <https://doi.org/10.17170/kobra-202401199387>

Schriftenreihe Verkehr

Heft 37 – Januar 2024

Herausgeber:

Institut für Verkehrswesen

Universität Kassel

34109 Kassel

© 2024, kassel university press, Kassel

<https://kup.uni-kassel.de>

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über

<http://dnb.dnb.de> abrufbar

Vorwort des Herausgebers

Die regelmäßige Unterbrechung von Verkehrsströmen an lichtsignalgeregelten Knotenpunkten führt zu Brems-, Halte- und Beschleunigungsvorgängen und somit zu einer Erhöhung von Emissionen, Energieverbrauch und Verschleiß an Fahrzeugen und Fahrbahn gegenüber einer ungehinderten Fahrt. Durch eine Vorausschau auf die bevorstehenden Schaltzeitpunkte kann die Fahrweise im Zulauf auf eine Lichtsignalanlage so angepasst werden, dass Fahrzeuge je nach Situation energiesparend auf Rot zufahren oder bei bevorstehender Freigabe sogar einen Halt vermeiden. Voraussetzung für diese sogenannte Ampelphasenassistenz ist die Prognose der Schaltzeitpunkte, welche bei verkehrsabhängigen Lichtsignalanlagen eine immer noch große Herausforderung darstellt.

Die vorliegende Arbeit greift diese Problematik auf und untersucht verschiedene Verfahren des maschinellen Lernens (ML), welche durch die Auswertung von Signalgeberzuständen, Detektordaten und ÖPNV-Anforderungsnachrichten eine Prognose der Schaltzeiten unabhängig vom jeweiligen Signalsteuerungsverfahren ermöglichen.

Anhand von aufgezeichneten Prozessdaten zweier Lichtsignalanlagen in Kassel wurde im Rahmen einer Voranalyse die Prognosequalität von 12 verschiedenen ML-Methoden ermittelt und einem Ranking unterzogen. Hier stellte sich der entscheidungsbaumbasierte XGBoost-Algorithmus als der leistungsfähigste heraus.

Die Datenmodellierung bzw. die Festlegung von Merkmalen und deren Behandlung im maschinellen Lernprozess ist der Schlüssel zu einem hochqualitativen Prognoseverfahren. Für den ausgewählten XG-Boost-Algorithmus wurden zahlreiche Experimente zum iterativen Finden erfolgreicher Merkmalskombinationen durchgeführt. Nicht immer führten vielversprechend erscheinende Erweiterungen des Merkmalsraumes zu einer Verbesserung. Dies traf auch auf die Erweiterung um Detektormerkmale benachbarter Knotenpunkte zu.

Die Verfahren erwiesen sich aufgrund der zunächst signalgruppenorientierten Merkmalsfestlegung als sehr aufwändig, da die Umschaltzeitpunkte für jede Signalgruppe separat zu prognostizieren waren. Daraufhin wurde zur Vorhersage von Signalisierungsphasen übergegangen, welche üblicherweise nicht in den Prozessdaten ausgewiesen sind und daher durch ein geeignetes Verfahren aus dem Zeitverlauf der Signalgeberzustände abgeleitet werden müssen. Hiermit wurde schließlich eine deutliche Qualitätsverbesserung der Schaltzeitprognose erreicht.

Frau Schneegans liefert mit dieser Arbeit einen bedeutenden Beitrag zur Konzeption, Realisierung und Bewertung von Verfahren zur Schaltzeitprognose mit Methoden des maschinellen Lernens. Die konzipierten Lösungen liefern wichtige Erkenntnisse zur Gewinnung belastbarer Vorhersagen unter den Herausforderungen der verkehrstechnischen Praxis.

Zusammenfassung

Im urbanen Raum, insbesondere vor signalisierten Knotenpunkten, entstehen die meisten Emissionen des Straßenverkehrs. Eine Möglichkeit, um diese Emissionen zu senken, sind sogenannte GLOSA-Systeme (Green Light Optimal Speed Advisory) und das Smart Routing, um am Straßenverkehr beteiligten Personen Handlungsempfehlungen geben zu können. Für die Berechnung der Informationen, die diese Systeme ausgeben, wie beispielsweise Geschwindigkeitsempfehlungen für die bestmögliche Nutzung von Grünzeiten und Zeiten bis zum Grünbeginn, werden Vorhersagen der Schaltzeiten von Lichtsignalanlagen (LSA) benötigt. Die Entwicklung ausreichend guter Vorhersageverfahren ist Gegenstand der Forschung.

Zentrum dieser Arbeit ist die Verbesserung der bestehenden Vorhersageverfahren zur Schaltzeitprognose (SZP) von LSA. Verbessert werden sollte die Übertragbarkeit, die Zuverlässigkeit und der Prognosehorizont von SZP. Durch die Übertragbarkeit kann der Schritt von der individuellen Prognoselösung für jede einzelne LSA zu einem weitgehend allgemeingültigen Verfahren, das flächendeckend eingesetzt werden kann, erfolgen. Die Zuverlässigkeit soll das Vertrauen der Nutzer in die Anwendung bestätigen, indem möglichst genaue Prognose getroffen werden.

Hierauf sowie auf den in der Dissertation von Weisheit [Wei17] benannten Bedarfe wurde ein möglichst allgemeingültiges Konzept zur SZP entworfen und experimentell teil-evaluiert, welches das Systemwissen zum LSA-Entwurf und -Betrieb sowie verschiedene Arten von LSA-Steuerungen integrieren können soll. Der Schwerpunkt der Evaluation des Konzepts lag auf der Methodik vor dem Hintergrund des Anwendungsgebietes. Für die Evaluation wurden OCIT-Daten aller Signalgruppen zweier für die Stadt Kassel typischen Knotenpunkte verwendet. Die verwendeten Evaluationsmetriken Korrekturklassifikationsrate (KKR) und Wurzel der mittleren quadratischen Abweichung (RMSE) wurden vor dem Hintergrund der Metriken bestehender SZP ausgewählt.

Besonders gute Ergebnisse konnten erzielt werden, indem vor der eigentlichen SZP die kommenden Phasen und ihre Schaltzeitpunkte vorhergesagt wurden. Diese vorherzusagen sowie ein Phasenmerkmal, das den Gesamtsignalisierungszustand der LSA beschreibt, dienten als zusätzliche Eingangsmerkmale für die SZP. Eine solche phasenorientierte SZP wird als ein Modell mit besonders hohem Potenzial für die erfolgreiche Integration in GLOSA-Systeme angesehen. Weiterhin wurden Detektionsmerkmale und ÖPNV-Meldungen in die Eingangsdatenmodelle integriert. Erkenntnis ist, dass die Detektionsmerkmale noch zielführender in eine SZP integriert werden könnten, wenn sie ebenfalls in ein den Gesamtzustand beschreibendes Merkmal überführt würden. Die ÖPNV-Meldungen verbessern die Prognosequalität. Auch die Integration von Merkmalen benachbarter Knotenpunkte konnte teilweise Verbesserungen der Prognosequalität bewirken. Als geeignetste Methode für SZP als auch Phasenvorhersagen wurde EXtreme Gradient Boosting (XGBoost) als Klassifikator unter den Methoden des maschinellen Lernens identifiziert.

Abstract

Most emissions caused by road traffic occur in urban areas, especially in front of signalized intersections. One approach to reducing these emissions is to use so-called GLOSA systems (Green Light Optimal Speed Advisory) and Smart Routing to provide recommendations to those involved in road traffic. Predictions of the switching times of traffic signal systems are needed to calculate the information that GLOSA and Smart Routing systems provide. These include, for example, speed recommendations for green times and times until the green light begins. The development of sufficiently good prediction methods is the subject of this research.

The focus of this work is to improve existing prediction methods for switching time prediction of traffic signals. The transferability, reliability, and prediction horizon of these estimations are to be improved. The transferability enables the step from an individual forecasting solution for each traffic signal to a generally applicable methodology. Reliability helps the user to confirm their confidence in the application by making predictions that are as accurate as possible.

Based on these considerations and on the needs named in the dissertation by Weisheit [Wei17], a concept for SZP was designed and experimentally evaluated. This concept shall be as universally applicable as possible and able to integrate the system knowledge for traffic signal design and operation as well as different types of traffic signal controllers. The concept evaluation focuses on the methodology in light of the application domain, evaluating all signal data of two intersections representative of the city of Kassel. Evaluation metrics are accuracy (ratio of correct classifications) and Root Mean Square Error (RMSE). The selection regarded the existing estimation metrics.

Satisfying results were obtained by estimating the upcoming phases and their switching times before the actual signal time prediction. Predicting a phase feature describing the overall signaling state of the traffic light served as additional input features for the switching time estimation. We consider such a phase-oriented switching time prediction to be a concept with high potential for successful integration into GLOSA systems. Furthermore, we integrated detection features and public transport messages into the input data models.

The finding is that the detection features could be integrated into an estimation more purposefully being merged into features describing the overall detection state. The public transport messages improve the forecast quality. Integrating features of neighboring intersections could also partially improve the forecast quality. We identified EXtreme Gradient Boosting (XGBoost) as the most suitable method for traffic signal switching time prediction among machine learning methods.

Danksagung

An dieser Stelle danke ich allen, die mich in diesen sechs Jahren unterstützt und getragen haben, insbesondere:

Danke, Professor Hoyer, für die Möglichkeit der Promotion und die ausschlaggebende Idee zur Themenfindung!

Danke, Professor Hoffmann für die Übernahme des Koreferats!

Danke, Leonie, Kevin, Lara und Max für Eure wunderbaren studentischen Arbeiten!

Danke, Jan, Sven und Viktor für die vielen Diskussionen zum Thema!

Danke, Petra, Maren, Astrid, Rike, Tanja, Kristina, Julia, Kira und Kirsten für die geballte Frauenpower!

Danke, liebe Freunde, dass ihr immer für mich da seid!

Danke, liebe Familie für die Unterstützung in allen Lebenslagen!

Inhaltsverzeichnis

1	Einleitung	17
1.1	Problemstellung	19
1.2	Zielsetzung	21
1.3	Aufbau	22
2	Grundlagen zu Lichtsignalanlagen und der Vorhersage ihrer Schaltzeiten	23
2.1	Grundlegendes zu Lichtsignalanlagen	23
2.2	Perspektiven auf Verkehrsnetzwerke	25
2.3	Begriffe und Systemgrößen in Zusammenhang mit Lichtsignalanlagen	26
2.3.1	Allgemeine Begriffe	26
2.3.2	Ausgangsgrößen	29
2.3.3	Eingangsgrößen	31
2.4	Kommunikative Straßenverkehrsinfrastruktur	34
2.4.1	Datenquellen und Informationsübertragung	36
2.4.2	Open Communication Interface for Road Traffic Control Systems (OCIT)	38
2.5	Entscheidungslogiken	40
2.5.1	Grundlegende Prinzipien	40
2.5.2	Programmtypen	43

2.6	Forschungsprojekte mit Bezug zu Schaltzeitprognosen	49
2.7	Literaturübersicht der Verfahren zur Schaltzeitprognose	51
2.7.1	Statistische Ansätze	51
2.7.2	Ansätze des maschinellen Lernens	53
2.7.3	Vergleich der Verfahren	54
2.8	Anforderungen an Verfahren zur Schaltzeitprognose	56
2.8.1	Zuverlässigkeit und Korrektheit	56
2.8.2	Rechtzeitigkeit und Prognosehorizont	57
2.8.3	Verfügbarkeit	57
2.9	Weitere Anmerkungen	58
3	Einführung in das maschinelle Lernen	59
3.1	Arten von maschinellem Lernen	60
3.1.1	Überwachtes Lernen	60
3.1.2	Unüberwachtes Lernen	61
3.1.3	Bestärkendes Lernen	62
3.2	Grundsätzliche Verfahrensweisen	63
3.2.1	Vorgehen zur Erstellung eines ML-Modells	63
3.2.2	Lernstrategien	65
3.2.3	Regularisierung	66
3.2.4	Kreuzvalidierung	68
3.2.5	Automatisierte Regularisierung	69
3.3	Automatisierte Merkmalsselektion	70
3.3.1	Statistische Einflussanalyse	71
3.3.2	Informationsbasierte Einflussanalyse	72
3.3.3	Weitere Einflussanalysen	73

3.4	Evaluationsmetriken	73
3.4.1	Metriken zur Klassifikation	74
3.4.2	Metriken zur Regression	76
3.4.3	Spezielle Metriken für SZP	77
3.5	Modelle des maschinellen Lernens	78
3.5.1	k-nächste Nachbarn	78
3.5.2	Support Vektor Machines	78
3.5.3	Entscheidungsbäume	80
3.5.4	Ensembles	82
3.5.5	Neuronale Netze	83
3.5.6	Markov-Ketten	86
3.5.7	Bayes'sche Netze	86
3.5.8	Diskussion der Methoden	88
4	Methodisches Konzept	93
4.1	Datengrundlage	93
4.1.1	Datengrundlage der Stadt Kassel	93
4.1.2	Qualitätsanalyse der Datengrundlage	97
4.2	Konzept eines Prognoseverfahrens	104
4.2.1	Konzept des Verfahrens	104
4.2.2	Konzept zum Konfigurieren einer Prognose	105
4.2.3	Zugrundeliegende Überlegungen	106
4.2.4	Vorgehen zur Umsetzung des Konzepts	110
4.2.5	Zwischenfazit	112
4.3	Implementierung	113
4.3.1	Bereinigen der Daten	113

4.3.2	Merkmalsberechnung	115
4.3.3	Experimentelle Durchführung	116
4.4	Bewertungsgrößen von Schaltzeitprognosen	120
5	Experimente zur Methodenwahl der signalgruppenorientierten Vorhersage	123
5.1	Grundlegende Auswahl der ML-Methoden	123
5.2	Versuchsaufbau	126
5.2.1	Vorgehen zum Vergleich der Methoden	126
5.2.2	Sonstige Einstellungen der Experimente	126
5.3	Ergebnisse für die Datenmodelle SG_s und SG	127
5.4	Ergebnisse des Vergleichs mit der Methode von TTS	135
5.4.1	Versuchsaufbau	135
5.4.2	Versuchsdurchführung	135
5.4.3	Ergebnisse	136
5.4.4	Schlussfolgerungen	141
5.5	Zwischenfazit	141
6	Experimente zu Datenmodellen zur signalgruppenorientierten Vorhersage	143
6.1	Merkmale aus Sicht des Anwendungsgebiets	143
6.1.1	Bekannte Merkmale	144
6.1.2	Merkmalsentwicklung	144
6.1.3	Überlegungen zu Merkern	149
6.2	Versuchsaufbau	150
6.2.1	Datenmodelle	150
6.2.2	Sonstige Einstellungen der Experimente	151

6.3	Vergleich der Repräsentationen	152
6.4	Vergangene Signale der eigenen SG vs. allen Signalen als Eingangsgrößen .	158
6.5	Erkenntnisse zu den Experimenten der Merkmals-selektion	162
6.6	Experimente zu Merkmalen benachbarter Knoten-punkte	163
6.6.1	Versuchsaufbau	163
6.6.2	Ergebnisse der Vorhersage von SG_3	165
6.6.3	Ergebnisse der Vorhersage von SG_4	170
6.7	Zwischenfazit	174
7	Experimente zur phasensorientierten Vorhersage	175
7.1	Ermittlung der Phasenmerkmale	175
7.1.1	Merkmal des Signalisierungszustands	177
7.1.2	Phasenmerkmal nach Weidemann	178
7.1.3	Phasenmerkmal nach Weidemann, ergänzt	180
7.1.4	Phasenspeicher	181
7.1.5	Phasenmerkmal bei Phasenwechseln ohne Phasenübergang	182
7.1.6	Statistische Bereinigung	182
7.2	Merkmalsevaluation	184
7.2.1	Evaluationsdurchführung	184
7.2.2	Evaluation der Phasenmerkmale nach Weidemann	185
7.2.3	Evaluation der Optimierung	186
7.2.4	Zwischenfazit	188
7.3	Versuchsaufbau	188
7.4	Methoden- und Datenmodellvergleich mit grundlegenden Datenmodellen: Phasenprognose	191
7.4.1	Phasenprognose der nächsten Phase	191

7.4.2	Phasenprognose der übernächsten Phase	194
7.5	Methoden- und Datenmodellvergleich mit grundlegenden Datenmodellen: Phasenschaltzeitprognose	196
7.6	SZP mit Phaseninformation	197
7.7	Schlussfolgerungen	200
	Literaturverzeichnis	201
	Symbolverzeichnis	217
	Abkürzungsverzeichnis	222
	Abkürzungsverzeichnis	223
	Abbildungsverzeichnis	237
	Tabellenverzeichnis	242

Kapitel 1

Einleitung

Ein Ziel der Bundesregierung ist es, klimaschädliche CO₂-Emissionen zu senken. Der Verkehrssektor ist hierbei der zweitgrößte Verursacher von Treibhausgasemissionen in Deutschland und ist für 25% aller Emissionen aus Deutschland verantwortlich [Umw21]. In urbanen Ballungszentren ist der Straßenverkehr der Hauptverursacher von gesundheits-schädlichen Lärm-, Feinstaub- und Stickoxid-Emissionen. Grenzwerte für diese Emissionen werden in europäischen Großstätten regelmäßig überschritten.

Das Nadelöhr für die Verkehrsströme in Straßennetzen sind Kreuzungen, weil Verkehrsbe-teiligte als Bestandteil sich kreuzender Verkehrsströme ihre gemeinsamen Konfliktflächen nur zeitlich nacheinander passieren können. Lichtsignalanlagen (LSA) teilen diesen Ver-kehrsströmen systematisch Freigaben zu und regulieren so den Verkehrsfluss möglichst konflikt- und damit unfallfrei. Durch die hierfür notwendigen Sperrungen der signalisier-ten Verkehrsströme fallen Energieverbrauch und Emissionen höher aus als bei einer un-gehinderten Durchfahrt, da es zu zyklischem Abbremsen und Wiederanfahren der Kraft-fahrzeuge kommt. Hinzu kommt, dass die Zahl der motorisierten Beteiligten schneller wächst als die Kapazität der Verkehrswege. Staus werden also länger. Wo in Deutschland 1960 73 Millionen Menschen und nur 4,5 Millionen Fahrzeuge bewegt wurden, bewegen sich heute 82,5 Millionen Menschen in 46,5 Millionen Fahrzeugen [Sta18]. Die Länge des Straßentransportnetzes nach dem Wiederaufbau nach dem zweiten Weltkrieg wuchs we-niger [Sta18]. So entsteht ein wesentlicher Teil der bundesweiten Emissionen in urbanen Gebieten des Straßenverkehrs durch Staus und Halte vor (signalisierten) Kreuzungen.

Eine Verminderung von Staus und Halten ist entsprechend wichtig, um schädliche Um-welteinflüsse zu reduzieren und um volkswirtschaftliche Schäden zu vermeiden. Zudem stellt eine funktionsfähige Straßenverkehrsinfrastruktur, welche möglichst wenige Halte verursacht, einen wichtigen Standortfaktor im wirtschaftlichen Wettbewerb dar. „Neben werbenden Maßnahmen zur Veränderung der Verkehrsmittelwahl ist das energie- und emissionsreduzierte Fahren mit Verbrennungsmotoren auf absehbare Zeit das effektivs-te Mittel zur Senkung des Ausstoßes von Treibhausgasen im Straßenverkehr“ führt et-wa [HS21] aus.

Ein solches Verkehrsverhalten soll hinsichtlich der Straßenverkehrsinfrastruktur unter anderem über die Digitalisierung der Straßeninfrastruktur unterstützt werden, in die insbesondere Assistenzsysteme für Beteiligte integriert werden können. Abbildung 1.1 zeigt hierfür ein Beispiel. Im urbanen Raum basieren viele dieser Assistenzsysteme auf Schaltzeitprognosen SZP¹, welche Brems- und Beschleunigungsvorgänge reduzieren sollen. Dies setzt voraus, dass beispielsweise mit Hilfe dieser Assistenzsysteme das Wissen vermittelt wird, wann bei Annäherung an eine LSA das Ende der Freigabezeit zu erwarten und ein Ausrollen zwecks Emissionseinsparung sinnvoll ist. So werden für die Zukunft innerörtlicher Verkehrsmanagementsysteme vor allem kooperative (digitale) Systeme, welche Informationen mit Fahrzeugen und anderen Infrastrukturelementen austauschen, als tragend angesehen. Dies betrifft insbesondere kooperative LSA [Nol19]. Konzepte zur Umsetzung werden derzeit in Expertengremien diskutiert, wobei SZP zentraler Bestandteil sind.



Abbildung 1.1: Rest-Rot- und Grünbandanzeige auf Basis einer SZP, Projekt AKTIV [Ott11]

Zu den Assistenzsystemen für LSA gehören sogenannte GLOSA-Systeme (Green Light Optimal Speed Advisory) beziehungsweise Ampelphasenassistenten. Sie schätzen Schaltzeitpunkte zwischen Freigabe und Sperrung in der unmittelbaren LSA-Zufahrt, wodurch unnötige Beschleunigungs- und Bremsmanöver verhindert werden sollen, also eine sinnvolle Wahl einer gleichmäßigen Geschwindigkeit begünstigt wird. Zudem sollen Anwendungen unter der Bezeichnung „smart routing“ entlang einer Route über mehrere Knotenpunkte hinweg und somit unter Berücksichtigung der SZP mehrerer LSA den energie- und emissionsärmsten Weg aufzeigen. Dies kombiniert die Optimierung von Routen- und Geschwindigkeitswahl und stellt höhere Anforderungen an die SZP als GLOSA-Systeme, insbesondere hinsichtlich des Prognosehorizonts.²

Die Verwendung für (mikroskopische) Verkehrsvorhersagen von Verkehrszentralen ist eine weitere Verwendungsmöglichkeit. Diese Zentralen können SZP für die Verkehrssteuerung nutzen. Mithilfe der SZP kann beispielsweise die Bildung von Pulks prognostiziert werden, da Pulks über die Grünphasen von LSA gebildet werden. Hierauf können LSA-Steuerungen präziser, zum Beispiel mit „grünen Wellen“, reagieren [Se14, Ke16].

¹Achtung: Nicht zu verwechseln mit der sonst üblichen Abkürzung SZP für SignalZeitenPläne, welche in dieser Arbeit nicht verwendet wird.

²In dieser Arbeit werden die Begriffe Prognose und Vorhersage synonym verwendet.

Demzufolge können zuverlässige Methoden zur Schätzung der Signalzeiten einer der notwendigen Schlüssel sein, Emissionen zu reduzieren. Forschungsprojekte belegen die signifikanten Vorteile der Schätzung von Schaltzeiten. So wurden bereits in den 1980er Jahren im Projekt „Wolfsburger Welle“ die Anzahl der Halte um 10 % bis 30 % und die Haltezeiten um 20 % bis 25 % reduziert sowie der Kraftstoffverbrauch bis 5 % gesenkt [Zim83]. In Bezug auf LSA und die Optimierung von Annäherungsgeschwindigkeiten stellt Menig fest, dass Stopps fast vollständig vermieden werden können. Einschlägige Artikel zeigen auch die Wirksamkeit von Geschwindigkeitsempfehlungen bei unvermeidbaren Stopps [Re11, AV10]. Die Simulationsexperimente von Richter [Ric05] ergaben, dass die Beeinflussung des Fahrverhaltens durch Geschwindigkeitsempfehlungen zu einem bis zu 20 % höheren Sättigungsverkehrsaufkommen bei kurzen Grünzeiten und zu einem bis zu 8 % höheren Sättigungsverkehrsaufkommen bei langen Grünzeiten führte. Es wurden Kraftstoffeinsparungen von durchschnittlich 15 % und maximal 35 % nachgewiesen. Dennoch mangelt es an Verfahren zur Vorhersage von Schaltzeiten für diese Anwendungen.

1.1 Problemstellung

Eine Literaturrecherche zeigt, dass es mehr Veröffentlichungen zu Assistenzsystemen, welche SZP verwenden, gibt als Verfahren zur SZP selbst. Es gibt zum Zeitpunkt der Verfassung dieser Arbeit keine allgemeingültigen, auf den gesamten D-A-CH-Raum übertragbaren Verfahren, die Informationen benachbarter Knotenpunkte mit einbeziehen. Auch gibt es zu diesem Zeitpunkt keine Untersuchungen, welche mögliche Datenmodelle und Methoden der künstlichen Intelligenz jeweils vergleichend einander gegenüberstellen. Zudem wurden keine Konzepte veröffentlicht, wie eine allgemeingültige Übertragbarkeit realisiert werden könnte. Es werden jedoch zwingend Vorhersagen der Freigabezeiten von LSA benötigt, um kooperative Ansätze zwecks Umweltschutz umsetzen zu können [Se14].

LSA werden in Deutschland auf zwei verschiedene Arten gesteuert. Schätzungen zu folge werden etwa ein Drittel der LSA durch Festzeitsignalprogramme gesteuert. Diese schalten nach einem festen Zeitplan zwischen den einzelnen Freigaben, weshalb die Schätzung der Signalzeiten trivial ist für diese. Eine Anpassung an das aktuelle Verkehrsgeschehen findet nicht statt. Die zweite Methode zur Steuerung von LSA in Deutschland sind verkehrsadaptive Steuerungen, entweder durch verkehrsabhängige Entscheidungslogiken oder modellbasierte Verfahren. Diese berücksichtigen Daten aus der Umgebung der Kreuzung, um die nächsten Freigaben zu wählen. Neben den Daten von Verkehrsdetektoren kann ein Steuergerät Informationen aus einer Vielzahl von Quellen, wie beispielsweise Priorisierungsmeldungen des öffentlichen Verkehrs, verwenden. Für diese komplexeren und oft speziell an den Knotenpunkt angepassten Programme sind die Signalzeiten oft schwieriger abzuschätzen [Nol19].

Da das Steuerungsverhalten solcher verkehrsabhängiger Signalisierungen, für welche der Verkehr mittels Detektoren erfasst wird, selbst bei einer Analyse der umfassenden Planungs- und Entwurfsunterlagen kaum ausreichend aufgeklärt werden kann, werden Vorhersagen üblicherweise aus einer automatischen Analyse von LSA-Prozessdaten gewonnen. Hier wird das Steuerungsverhalten der LSA anhand der protokollierten Signali-

sierungszustände und gegebenenfalls auch unter Berücksichtigung der zugrunde liegenden Detektordaten des Individualverkehrs (IV) und der Meldungsdaten von ÖPNV-Fahrzeugen geschätzt. Am Institut wurden hierzu in einem eigenständigen Dissertationsprojekt [Wei19a] Methoden entwickelt, um Ersatzsteuerungen für zweiphasige LSA zu erzeugen.

Die entwickelten Methoden und Analysen erfüllen die Voraussetzungen, um sie für die von Weisheit in seiner Dissertation über SZP [Wei17] aufgezeigten Forschungsbedarfe insbesondere für die Aufbereitung der Datenbasis nutzen zu können. Forschungsbedarfe sind die Verbesserung der Datenmodelle, eine automatisierte Zuweisung zwischen Detektoren und Signalgruppen; ferner können die Detektoren auf logische beziehungsweise kombinatorische Zusammenhänge hin untersucht werden; die ÖPNV-Meldepunkte benachbarter Knotenpunkte können in die Datenmodelle einbezogen werden; und es sollten weitere Methoden des maschinellen Lernens für SZP getestet, sowie bezüglich ihrer Performanz bestehenden Verfahren mit Realdaten gegenübergestellt werden. Methoden maschinellen Lernens werden in der Literatur und insbesondere in [Wei17] als besonders geeignetes Methodenfeld für die Erstellung von präzisen Schaltzeitprognosen SZP für verkehrsabhängige Signalisierungen angesehen.

1.2 Zielsetzung

Das Hauptaugenmerk dieser Arbeit liegt auf der Entwicklung eines Verfahrens zur SZP. Das Gesamtziel dieser Dissertation ist entsprechend die Entwicklung eines möglichst allgemeingültigen und weitgehend automatisierbaren, modularen Verfahrens zur Vorhersage der Schaltzeiten von LSA. Mit einem solchen Verfahren ist beabsichtigt, ein auf verschiedenste LSA-Programme anwendbares und zugleich zuverlässigeres Prognoseverfahren zur Verfügung zu stellen. Dies kommt Forderungen aus der Literatur wie [Wei17, Pro16, Wei19a] nach.

Im Einzelnen werden die folgenden Forschungsfragen bearbeitet:

1. Wie kann ein allgemeingültiges, übertragbares Konzept für SZP aussehen, welches das Systemwissen zum LSA-Entwurf und -Betrieb sowie verschiedene Arten von LSA-Steuerungen integrieren kann? (Kapitel 4)
2. Wie sieht die Zusammenstellung einer entsprechenden Evaluation aus? (Kapitel 3.4)
3. Was sind geeignete Datenmodelle für eine SZP mittels maschinellen Lernens? (Kapitel 6 und 7)
4. Wird eine Verbesserung durch die Integration von Daten benachbarter Knotenpunkte erreicht? (Kapitel 6.6)
5. Was sind geeignete Methoden des maschinellen Lernens für eine Prognose der Schaltzeiten? (Kapitel 5)
6. Wie ist eine Vorhersage der Phasen hinsichtlich Methoden und Datenmodellen zu gestalten, und führen diese zu einer Verbesserung der SZP? (Kapitel 7)
7. Können Verbesserungen im Vergleich zu bisher bestehenden SZP erzielt werden? (Kapitel 5.4)

Die Forschungsarbeit teilte sich in zwei Phasen auf: Zunächst erfolgte die Entwicklung des Prognosekonzepts. Hierauf folgte seine exemplarische Umsetzung auf Basis von Prozessdaten realer LSA. Während der Konzeptphase wurde der Schwerpunkt auf die Ausarbeitung eines modularen Konzepts und Verfahrens für regelbasierte LSA-Steuerungen nach den Richtlinien für Lichtsignalanlagen (RiLSA) [FGS10] gelegt. Die exemplarische Evaluation findet an ausgewählten LSA auf der Basis von für Kassel bereits vorliegenden OCIT-Daten statt. Die exemplarische Umsetzung beschränkt sich aufgrund der Verfügbarkeit entsprechender Prozessdaten auf LSA-Steuerungen mit regelbasierten Entscheidungslogiken freier Signalprogrammgebung. Die Stadt Kassel ist aufgrund ihrer fortgeschrittenen

Digitalisierung ihrer Verkehrsinfrastruktur und ihres digitalen Testfelds für die Evaluation des Verfahrens besonders geeignet.

Eine Simulation der Knotenpunkte, eine vollständige Automatisierung des zu entwickelnden Konzepts sowie eine (Echtzeit-)Implementierung mit entsprechenden Anforderungen sowie der Test in einer (Assistenz-)Anwendung sind außerhalb des Rahmens dieses Dissertationsvorhabens. Entsprechend werden weder eine Einbettung in die bestehende Straßenverkehrsinfrastruktur noch praktischen Anwendungsversuche im Straßenverkehr unternommen. Zentrum ist das Verfahren zur SZP selbst, nicht die umgebende Prozesskette.

1.3 Aufbau

Diesem einleitenden Kapitel folgt in Kapitel 2 die Erläuterung der Grundlagen zu LSA und der Vorhersage ihrer Schaltzeitpunkte. Dort werden neben den zentralen Begriffen des betrachteten technischen Systems die grundlegenden Entscheidungsprozesse innerhalb der Steuerung besprochen. Zudem werden die bekannten Verfahren zur SZP sowie die an sie gestellten Anforderungen diskutiert.

In Kapitel 3 werden die Grundlagen für die verwendeten Methoden des maschinellen Lernens vorgestellt. Aufbauend auf dieser Wissensgrundlage wurde ein Konzept für ein möglichst übertragbares und automatisierbares Vorhersageverfahren entwickelt, welches in Kapitel 4 zur Methodik dokumentiert ist. Darüber hinaus sind dort seine Implementierung im Rahmen der exemplarischen Umsetzung des Konzepts beschrieben, welche für die Durchführung der Experimente notwendig war, sowie die zugehörige Datengrundlage und das Evaluationskonzept.

In Kapitel 5 werden die Methoden des maschinellen Lernens verglichen, welche ausgewählte Datenmodelle aus Kapitel 6 verwenden.

Kapitel 6 fasst alle Experimente mit dem Fokus auf Datenmodelle zusammen. Hierzu zählen auch solche, welche Daten benachbarter Knotenpunkte enthalten. Hierfür wird die beste Methode aus Kapitel 5 verwendet.

Experimente zu phasenorientierten SZP sind in Kapitel 7 zu finden. Hier liegt der Schwerpunkt auf Vorhersagen von LSA-Phasen sowie auf solchen Prognosen beruhenden Schätzungen der Schaltzeitpunkte einzelner Signalgruppen.

Abschließend findet sich in Kapitel ?? die Zusammenfassung dieser Arbeit sowie ein Überblick über den verbleibenden Forschungsbedarf.

Kapitel 2

Grundlagen zu Lichtsignalanlagen und der Vorhersage ihrer Schaltzeiten

Die Vorhersage von Ausgangssignalen einer LSA, auch als Schaltzeitprognose (SZP) bezeichnet, setzt ein grundlegendes Verständnis des technischen Systems LSA und seines Kontextes voraus. Deshalb werden die zugehörigen Grundlagen (Kennzahlen, Begriffe, Logiken) sowie Kontexte an dieser Stelle erläutert. Zunächst wird auf die Rolle von LSA im Straßenverkehr eingegangen. Darauf folgt eine Erläuterung der zentralen Begriffe und der Entscheidungsprozesse der Steuerungsmechanismen von LSA. Abschließend werden hier Forschungsprojekte dargestellt, welche SZP betrachten, sowie Verfahren zur SZP diskutiert.

2.1 Grundlegendes zu Lichtsignalanlagen

Eine LSA ist eine technische Einrichtung zur Verbesserung der Qualität des Verkehrsablaufs, höchste Priorität hat hierbei die Sicherheit. LSA koordinieren den Verkehr, damit teilnehmende Personen und Fahrzeuge an Knotenpunkten und Einmündungen diese gefahrlos queren können. Durch diese koordinierten Eingriffe werden die sich räumlich überlagernden Verkehrsströme zeitlich entkoppelt. Somit werden Kollisionen vermieden. Zudem kann so die Kapazität des Knotenpunktes zumeist erhöht werden, siehe Abschnitt 2.3. LSA können die Zielkonflikte zwischen den verschiedenen Verkehrsbeteiligten an Knotenpunkten ausbalancieren. Zu den Verkehrsbeteiligten zählen motorisierter Individualverkehr (mIV), öffentlicher Personennahverkehr (ÖPNV) und zu Fuß gehende Personen (FG). Die Gruppe der FG umfasst für diese Arbeit auch Verkehrsbeteiligte, die Fahrrad fahren, falls diese nicht über einen eigenen Fahrstreifen bzw. -weg verfügen. Diese sind im Straßenverkehr besonders gefährdet.

Neben den Aufgaben, über die Koordination miteinander in Konflikt stehende Verkehrsströme zu trennen und somit Sicherheit zu gewährleisten, erfüllen LSA noch weitere Aufgaben: Sie betreffen die Bereiche von Wirtschaftlichkeit und Umweltverträglichkeit des Verkehrsablaufs. Zur Wirtschaftlichkeit zählen nicht nur der von der LSA für den Betreiber verursachte Aufwand für Betrieb (inkl. Handhabbarkeit), Installation und Wartung, sondern auch Interessen von Gruppen Verkehrsbeteiligter an der Reduktion von Halten, Reise- und Wartezeiten sowie des Kraftstoffverbrauchs. Insbesondere bezüglich des ÖPNV spielen zudem Aspekte des Fahrkomforts und der Einhaltung von Fahrplänen eine Rolle. In Deutschland wird der ÖPNV an Knotenpunkten aufgrund politischer Überlegungen bevorzugt.

Die technische Umsetzung umfasst hierfür eine Reihe an Komponenten. Zum einen werden die Signale einer LSA über mit zum Teil farbigen Abdeckscheiben bedeckte Leuchtmittel kommuniziert. Je nach Alter der Anlagen kann es sich bei den Leuchtmitteln beispielsweise um Glühlampen oder farbige LEDs handeln. Zum anderen werden die Signale auch über Lautsprecher und Summtaster kommuniziert. All diese Elemente sind an Masten befestigt. Wenn diese Signale an das aktuelle Verkehrsaufkommen angepasst werden sollen, finden sich zudem Detektoren und Taster an den Masten und unter der Fahrbahndecke, um dieses Aufkommen zu erfassen. Am Rand eines signalisierten Knotenpunktes befindet sich zudem ein Schaltschrank mit im Wesentlichen Netzteil, Steuerbaugruppe, Sicherheitsbaugruppe, Signaleingang und -ausgang sowie Lüftung. Auf diesen Komponenten ist die Software zur Steuerung und Regelung der Lichtsignale, im Folgenden als LSA-Programme bezeichnet, und zur Prüfung der Ausgabesignale bezüglich der Verkehrssicherheit installiert. Teilweise ist die Software für Steuerung und Regelung auch auf Zentralrechnern gebündelt.

In dieser Arbeit werden LSA als Regelungs- und Steuerungssysteme entsprechend dem Verständnis aus der Regelungstechnik betrachtet. Das steht teilweise im Gegensatz zum Fachwortschatz der Verkehrstechnik, welche ausschließlich über Steuerungen spricht. Regelung ist in der Verkehrstechnik im Kontext von Verkehrsregeln, Richtlichtern und Ähnlichem zu verstehen. Davon abweichend wird hier zwischen Regelung und Steuerung unterschieden: Sobald ein System aktiv Daten erfasst, um darauf basierend Entscheidungsprozesse abzuarbeiten (zum Beispiel über Detektoren), wird in dieser Arbeit von einer Regelung, in allen anderen Fällen weiterhin von einer Steuerung gesprochen.

Für die Gestaltung von LSA-Programmen gibt es länderspezifische Gestaltungshinweise. Eine umfangreichere Übersicht zu Ländern, ihren Institutionen und LSA-Richtlinien findet sich in [Wei19a]. Für Deutschland sind diese beispielsweise in den RiLSA [FGS10] sowie den zugehörigen Merkblättern und Hinweisen unter anderem für die Bauelemente, Abmessungen und Beschilderungen zu finden, siehe [FGS14a, FGS15, FGS93, FGS99, FGS14b, FGS17, FGS91]. In diesen Richtlinien sind Bestimmungen und Empfehlungen zu Betrieb und Errichtung von LSA zu finden. Die RiLSA haben in Deutschland rechtlich erhebliches Gewicht, Abweichungen von ihr müssen stichhaltig begründet werden.

Die RiLSA stellt aus Sicht der Schweizer Normen [Se10] die wichtigste ausländische Richtlinie für LSA dar, da die Schweiz selbst nur über Berechnungs-, allerdings nicht über Gestaltungsvorgaben verfügt. Vergleichbare Hinweise für die Erstellung von Signalprogrammen (SP) finden sich beispielsweise in Österreich und Japan [Be06b, Be06a].

International stellen die RiLSA somit einen hohen Stand der Entwicklung zum Bau von LSA und der Erstellung von LSA-Programmen dar [BF07]. Deshalb lehnen sich alle in dieser Arbeit dargestellten Erläuterungen zum Stand der Technik von LSA an die RiLSA und die gängige Arbeitspraxis in Deutschland an. Für Großbritannien spielen die RiLSA beispielsweise keine Rolle. Dort werden LSA nach ihrer Erstellung praktisch getestet und über eine Software optimiert. Hierfür wurde ein eigenes Verfahren entwickelt: Split Cycle Offset Optimisation Technique (SCOOT), siehe auch Abschnitt 2.5.2. Es findet nicht nur in Großbritannien, sondern auch teilweise in den USA, Lateinamerika und Asien Anwendung. Für die praktische Umsetzung von LSA-Programmen wird spezielle Software für die Planung und Programmierung der Logiken und ihre erste simulative Prüfung eingesetzt.

2.2 Perspektiven auf Verkehrsnetzwerke

In der Verkehrstechnik gibt es zwei verschiedene Blickwinkel, aus denen ein Straßenverkehrsnetz betrachtet werden kann: **makroskopisch** und **mikroskopisch**, siehe [Ott11, SP16, Wei17, Wei19a].

Die **makroskopische** Perspektive betrachtet Netzausschnitte als Ganzes, von einem Netzwerk aus wenigen Knotenpunkten bis hin zur Betrachtung einer gesamten Stadt. Verkehrsbeteiligte werden so zu Gruppen beziehungsweise Strömen zusammengefasst und über entsprechende Größen beschrieben; das Individuum ist hierbei nicht mehr identifizierbar. Diese Sichtweise ermöglicht insbesondere die Betrachtung der Verkehrsentwicklung und -belastung über längere Zeiträume oder eine Momentaufnahme eines größeren Netzausschnittes, welche keine sekundengenaue Abbildung benötigt.

Die **mikroskopische** Sicht betrachtet üblicherweise einzelne Knotenpunkte und kleinere Streckenabschnitte bzw. Straßennetze, so dass die Verkehrsbeteiligten einzeln erfasst und beschrieben werden können. Somit werden individuelle Eigenschaften und Verhaltensweisen von Verkehrsbeteiligten mit abgebildet, wie beispielsweise eine Beschreibung des Fahrverhaltens. Hiermit kann das Verkehrsgeschehen subsekundengenau abgebildet werden, um zum Beispiel Routenwahlen (Abbiegeraten) oder Ankunftszeiten zu ermitteln. Für die Parametrierung von Signalprogrammen werden so Freigabebedarfe simulativ ermittelt.

Teilweise werden auch Steuerungsebenen nach diesen Perspektiven benannt, um den räumlichen Einflussbereich des verwendeten Verfahrens deutlich zu machen, zum Beispiel in [FGS10], hierzu mehr in Abschnitt 2.4. Die zugehörigen Größen werden, falls für diese Arbeit relevant, in Abschnitt 2.3 näher erläutert.

2.3 Begriffe und Systemgrößen in Zusammenhang mit Lichtsignalanlagen

In diesem Abschnitt werden die für diese Arbeit relevanten Begriffe und Systemgrößen für die Beschreibung von LSA erläutert. Zunächst werden allgemeine Begriffe zur Beschreibung der Software und Größen im Zusammenhang mit Knotenpunkten definiert (Abschnitt 2.3.1). Danach werden die Ausgangssignale einer LSA (Abschnitt 2.3.2) und zuletzt ihre Eingangssignale beschrieben (Abschnitt 2.3.3).

2.3.1 Allgemeine Begriffe

Zunächst werden die Begriffe im Bereich Software, die in dieser Arbeit eigene Begriffsdefinitionen erhalten, definiert: Unter einem **(LSA-)Programm** wird eine Firmware verstanden, die alle Elemente zur Umsetzung der Steuerung beziehungsweise Regelung einer LSA (Freigaben und Sperrungen von signalisierten Knotenpunkten) inkl. Übertragungsprotokollen zur Netzanbindung etc. enthält. Abbildung 2.1 zeigt die hier verwendeten Bezeichnung der Programmbestandteile und ihres Zusammenhangs.

Ein **(LSA-)Algorithmus** ist ein Teil eines Programms und wird ausschließlich von dem Teil der Software gebildet, der zur Festlegung von Schaltzeitpunkten (Freigeben/Sperren, siehe unten) benötigt wird. Er enthält die (Entscheidungs-)Logik und zugehörige Signalverarbeitung (Vorlogik) wie beispielsweise die Ableitung von Kenngrößen und Merkern aus Detektorsignalen, siehe Abschnitt 2.3.3. In Merkern wird ein Systemzustand gespeichert, wie beispielsweise ob Zeitlückenkriterien von einzelnen Detektorgruppen erfüllt oder maximale Wartezeiten überschritten sind.

Die **(Entscheidungs-)Logik** ist das zentrale Element eines Programms und Algorithmus, siehe Abbildung 2.1. Es verwendet logische beziehungsweise kombinatorische Verknüpfungen ausgewählter Kenngrößen und Merker zur Findung von Steuerungs-/Regelungsentscheidungen über die Schaltzeitpunkte der Verkehrsströme.

„Unter einem Signalprogramm versteht man die hinsichtlich Dauer und Zuordnung festgelegten Signalzeiten einer Lichtsignalanlage“, nach RiLSA [FGS10]. Ein LSA-Programm kann diese Arbeit betreffend mehrere Algorithmen und Logiken enthalten, welche zu unterschiedlichen Zeiten aktiv sind, um Signalzeiten festzulegen. Algorithmen und Logiken, welche in gleichen Zeiträumen aktiv sind, werden in dieser Arbeit als **Signalprogramm** (SP) zusammengefasst.

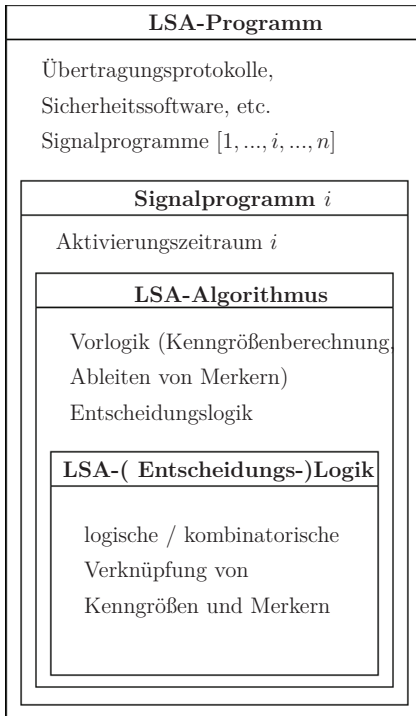
Zumeist existieren für ein LSA-Programm vier Signalprogramme, jeweils eins für den Berufsverkehr morgens und abends sowie für Schwachlastzeiten und nachts.

Die gemittelte Geschwindigkeit zwischen Knotenpunkten wird als Entwurfsgröße bezüglich der **Progressionsgeschwindigkeit** v_p in $[\text{km/h}]$ herangezogen. Die Progressionsgeschwindigkeit liegt üblicherweise bei ca. 80-90% der zulässigen Höchstgeschwindigkeit.

Das Verhalten von Gruppen der Verkehrsbeteiligten bestimmt den **Verkehrszustand und -ablauf**. Hierzu gibt es verschiedene beschreibende Größen, welche in dieser Arbeit allerdings nur eine untergeordnete Rolle spielen, siehe auch [FGS15].

An Verkehrsknotenpunkten treffen (**Zufluss-**)**Verkehrsströme** von Verkehrsbeteiligten i, j aufeinander. Ein Verkehrsstrom ist die Zusammenfassung von Verkehrsbeteiligten innerhalb einer Verkehrsbeziehung.

Die wichtigste beschreibende Größe für die Auslegung von LSA ist die **Verkehrsstärke/-belastung** q_i in $[\frac{\text{Fz}}{\text{h}}]$, welche die Anzahl an Fahrzeugen (Fz), die einen Querschnitt passieren, in zeitlichen Bezug setzt.



Weitere Größen sind die (**Knotenpunkt-**) **Kapazität** und die zugehörigen **Auslastungs-** beziehungsweise **Sättigungsgrade** α_i in [%] sowie **Zeitbedarfs-** **werte** τ_B in $[\frac{\text{s}}{\text{Fz}}]$. Sättigungsgrade bei denen kein Rückstau vorliegt, liegen für Fahrstreifen des mIV erfahrungsgemäß bei ca. 80% [Str19]. Der Zeitbedarfswert gibt die durchschnittliche Bruttozeitliche zwischen zwei Fahrzeugen an, welche die Haltlinie passieren. Die **Verkehrsdichte** k_i in $[\frac{\text{Fz}}{\text{km}}]$ setzt die Anzahl an Fahrzeugen (Fz) in räumlichen Bezug.

Eine verkehrsabhängige LSA kann als Regelkreis, siehe Abbildung 2.2, betrachtet werden. Abbildung 2.3 veranschaulicht dies beispielhaft für LSA. Zunächst soll ein Überblick über das System LSA erfolgen. Die verwendeten Begriffe werden in den anschließenden Abschnitten beginnend mit den Ausgangsgrößen näher erläutert. Die hier genannten Symbole erscheinen in dieser Arbeit nur an dieser Stelle und finden sich daher nicht im Symbolverzeichnis. Die bezüglich des Regelungssystems LSA sichtbarsten Größen sind die Ausgangsgrößen y .

Abbildung 2.1: LSA-Programmbestandteile

Ausgangsgrößen sind Signale in Form von farbigen Lichtzeichen, die den am Straßenverkehr beteiligten Personen Sperrungen und Freigaben von Verkehrsrichtungen anzeigen. Nähere Erläuterungen zum Begriffsfeld in Zusammenhang mit Lichtzeichen finden sich in Abschnitt 2.3.2. Mit Hilfe dieser Zeichen wird der Verkehr gesteuert. Welche dynamische Größe als Representation des Straßenverkehrs für die Regelung gewählt wird, hängt von der verwendeten Art der LSA-(Entscheidungs-)Logik ab. Der Straßenverkehr kann neben Lichtzeichen durch weitere Einflussfaktoren, die in diesem Kontext Störgrößen z darstellen, beeinflusst werden. Baustellen, Unfälle und Witterung am und im Umkreis vom Knotenpunkt sind beispielsweise solche.

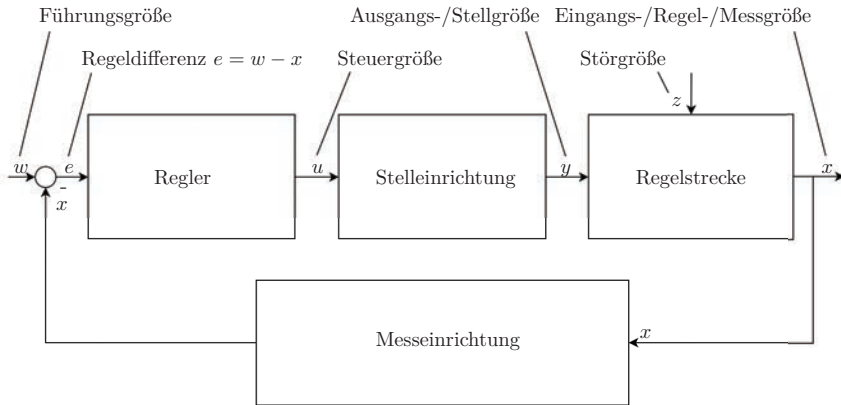


Abbildung 2.2: Schema eines Regelkreises

Entsprechende Messgrößen x werden mittels Detektoren erfasst. Hierfür können beispielsweise in den Asphalt eingelassene Induktivschleifen, Taster für FG oder Kameras verwendet werden. Messgrößen dienen als Eingangsgrößen x für die Regelung der Lichtzeichen, siehe Abschnitt 2.3.3. Die LSA-Logik ist im nichtflüchtigen Programmspeicher auf der Steuerplatine implementiert. Führungsgröße und Regeldifferenz sind zumeist in dieser Logik integriert und vom Typ der LSA-Logik abhängig, siehe Abschnitt 2.5. Die Führungsgröße stellt in der Regel den Sollwert der Ausgangsgröße dar, umgerechnet in entsprechende Messwerte. Die Entscheidungen der Logik werden als Steuergrößen u zunächst innerhalb des LSA-Programms an die Signalsicherungskomponente übergeben, welche getrennt von der Steuerplatine verbaut wird. Auf der Platine der Signalsicherung ist eine Firmware installiert, die Fehlsignale wie beispielsweise gleichzeitiges Rot und Grün für unverträgliche Verkehrsströme verhindern sollen. Diese Software ist spezifisch an die jeweilige Hardware angepasst. Die korrekten Signalbefehle werden daraufhin an die Signalgeber weitergegeben.

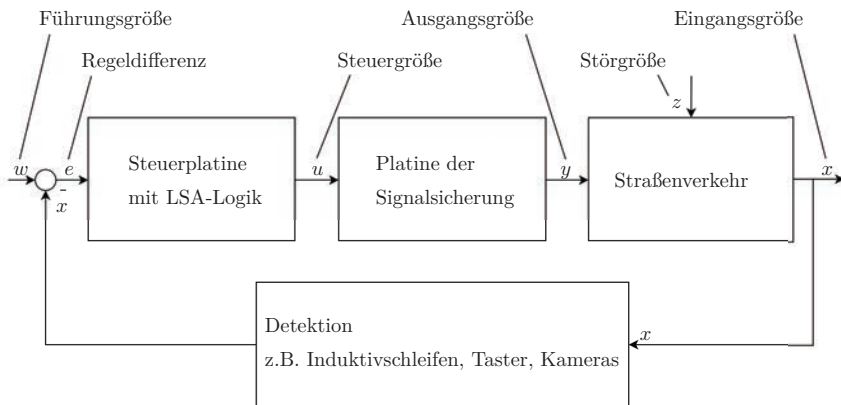


Abbildung 2.3: Beispiel eines Regelkreises für eine LSA

2.3.2 Ausgangsgrößen

Alle Größen dieses Abschnitts werden von LSA-Steuerungen beziehungsweise -Regelungen zeitdiskret verarbeitet. Die digitale Zeitschrittweite für Entscheidungen der Logik und Ansteuerung der SG liegt bei einer Sekunde. Detektionseinrichtungen weisen kleinere Schrittweiten von 0,1 oder 0,01 s auf. Größen sind hier alle Eigenschaftsbeschreibungen einer LSA-Steuerung und des Straßenverkehrs, welche quantitativ dargestellt werden können. Größen, welche in dieser Arbeit während der Laufzeit eines LSA-Programms Eingangsgrößen für die Prognosen bilden können, sind mit * gekennzeichnet und solche, welche vorhergesagt werden könnten, mit °. Nicht gekennzeichnete Größen werden mittels anderer Größen erfasst oder sind zur Laufzeit eines LSA-Programms unveränderlich und können so direkt in die Konzeption einer Prognose einfließen.

Ausgangsgrößen von LSA sind Sperrungen und Freigaben des Verkehrsflusses für verschiedene Verkehrsströme. Diese Sperrungen und Freigaben werden über die angezeigten **Farben** optisch sowie akustisch über eigene Signalgeber und/oder haptisch z.B. an FG-Tastern kommuniziert. Die übliche Abfolge in Deutschland für den ausschließlich optisch signalisierten mIV ist Rot, Rot-Gelb, Grün, Gelb, Rot, für FG Rot, Grün. Für den ÖPNV gibt es gesonderte Symbole nach Straßenbahn-Bau- und Betriebsordnung (BO-Strab) [Ges87]. Diese optischen Signale werden mit Hilfe von **Signalgebern** angezeigt. Ihre räumliche Position ist in **Lageplänen** verzeichnet, siehe Anhang B. Ein Lageplan erfasst maßstäblich die „für den Entwurf [der LSA] wesentlichen örtlichen Gegebenheiten“ [FGS10]. Ein oder mehrere Signalgeber, die zur Abwicklung eines Verkehrsstromes eingesetzt werden, bilden eine **Signalgruppe (SG)**. Die Signalgeber einer $SG_s | s \in n_s$ zeigen immer identische Informationen. Unter einer **Signalgruppe** werden nach RiLSA [FGS10] „ein oder mehrere Signalgeber, die bestimmte Verkehrsströme gemeinsam steuern und zu jeder Zeit das gleiche Signalbild zeigen“ verstanden. Die Menge aller Signalgruppen sei n_s und die Nummer einer Signalgruppe s . Das **Signalbild** eines Signalgebers ist die Farbe, die an ihm zu einem bestimmten Zeitpunkt aufleuchtet. Den zeitlichen Verlauf einer SG bildet der Signalzustandsverlauf ab.

Der zeitliche Zustandsverlauf von SG ist von **Freigabe-** und **Sperrzeiten** geprägt, siehe Abbildung 2.4, über welche die einzelnen Verkehrsströme koordiniert werden. Ihr Beginn beziehungsweise Ende zu einem diskreten Zeitpunkt k werden als **Grünanfang (GA)***° $t_{GA,s}(k)$ respektive **Grünende (GE)***° $t_{GE,s}(k)$ bezeichnet. Der diskrete Zeitpunkt kann verschieden dargestellt werden, beispielsweise in Sekunden oder Minuten seit Mitternacht, in Unix Time oder Umlaufsekunden, welche später definiert werden. Eine Freigabe einer Signalgruppe s ist in Signalzeitenplänen als leeres Rechteck dargestellt, ihre Dauer wird als **Freigabe-** beziehungsweise **Grünzeit***° $t_{F,s}(k)$ in [s] bezeichnet, siehe Abbildung 2.5. Die signaltechnisch gesicherte **Mindestfreigabezeit** beträgt fünf Sekunden [FGS10]; im Signalprogramm können höhere minimale Freigabezeiten vorgesehen sein.

Der Wechsel zwischen Freigabe zu Sperrung beim mIV wird mit Gelb angekündigt, verzeichnet als diagonal durchgestrichenes Rechteck. Diese **Gelbzeit** t_G in [s] beträgt, abhängig von der zulässigen Höchstgeschwindigkeit, drei bis fünf Sekunden [FGS10]. Für den ÖPNV gelten gesonderte Gelbzeiten.

Auch der Wechsel von Sperrung zu Freigabe wird über die **Rot-Gelb-Zeit** t_{RG} in [s] über eine Sekunde Rot-Gelb angekündigt, verzeichnet als diagonal und horizontal durchgestrichenes Rechteck. Eine Sperrung wird mit einer hervorgehobenen Linie im Signalplan vermerkt. Die zugehörige Dauer wird als **Rotzeit**[°] t_R in [s] bezeichnet. In Bezug zu SZP wird die **Sperrzeit** $t_{Rs,s}(k)$ einer Signalgruppe s betrachtet. Diese umfasst die Rot-, Gelb- und Rot-Gelb-Zeit.

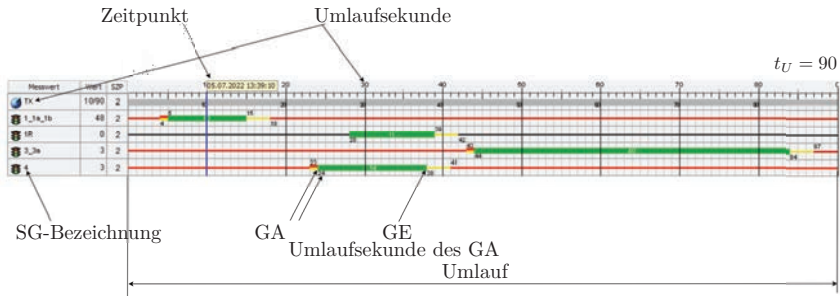


Abbildung 2.4: Beispiel für den Zustandsverlauf von Signalgruppen aus einem Anwenderprogramm, LSA 722, SP 2 (Darstellung nicht RiLSA-konform)

In Abbildung 2.5 ist beispielhaft das Signalbild von fünf Signalgruppen und Detektoren dargestellt. Es ist zu erkennen, dass bevor eine Signalgruppe freigegeben wird, nachdem eine andere gesperrt wurde, beide Signalgruppen für eine bestimmte Zeit Rot signalisieren. Der zeitliche Abstand zwischen den Freigaben von zeitlich aufeinanderfolgenden in Konflikt stehenden $SG_{i,j}$ beziehungsweise deren Verkehrsströmen wird als **Zwischenzeit** $t_{Z,i,j}$ in [s] bezeichnet. Nach RiLSA: „Die Zwischenzeit ist die Zeitdauer zwischen dem Ende der Freigabezeit eines Verkehrsstroms und dem Beginn der Freigabezeit eines anschließenden freigegebenen nicht verträglichen Verkehrsstroms“ [FGS10]. Zur Berechnung der Zwischenzeit werden die Überfahrzeit $t_{\bar{u}}$, der Räumzeit t_r und der Einfahrzeit t_e herangezogen:

$$t_Z = t_{\bar{u}} + t_r - t_e | t_{\bar{u}} + t_r \geq t_G + 1. \tag{2.1}$$

Die Überfahrzeit beträgt 2 s bis 7 s, abhängig von der Knotenpunktgeometrie und dem betrachteten Fahrzeug. Die Räumzeit ist die Zeit, die benötigt wird, um den gemeinsamen Konfliktpunkt vollständig mit dem Fahrzeugheck zu verlassen. Die Einfahrzeit ist die Zeit, die benötigt wird, um den gemeinsamen Konfliktpunkt mit der Fahrzeugfront zu erreichen. Der Berechnung zugrunde liegende Längen, Geschwindigkeiten und Positionierungen der Konfliktpunkte können der RiLSA entnommen werden, [FGS10]. Zwischenzeiten sind zentral für die Sicherheit der Verkehrs an LSA. Sie dient dazu, bei einem

GE das Ende des letzten Fahrzeugs des endenden Verkehrsstroms über den gemeinsamen Konfliktpunkt hinaus abfließen zu lassen, bevor nach einem GA das erste Fahrzeug des nachfolgend beginnenden Stroms diesen Punkt mit der Front einnehmen kann. Das heißt, es wird sichergestellt, dass Fahrzeuge verschiedener Richtungen nicht gleichzeitig einen möglichen Kollisionspunkt erreichen können; so werden Kollisionen vermieden. Zwischenzeiten werden gebündelt in einer **Zwischenzeitenmatrix** dokumentiert.

Verkehrsströme, die in Konflikt miteinander stehen, bei denen es bei gleichzeitiger Freigabe aufgrund gemeinsamer Konfliktflächen zu einem Unfall kommen könnte, werden auch als **unverträglich** bezeichnet. **Verträgliche** Verkehrsströme sind hingegen solche, welche beim Passieren einer Kreuzung keine gemeinsamen Konfliktflächen aufweisen. Und **bedingt verträglich** bedeutet, dass die betroffenen Verkehrsströme miteinander in Konflikt stehen, jedoch trotzdem eine gleichzeitige Freigabe erhalten können. Dies ist zum Beispiel häufig bei gemeinsam mit dem geradeaus fahrenden Verkehr signalisierten Rechtsabbiegern und parallel zur Geradeausrichtung freigegebenen FG der Fall.

Die Signalgruppen können weiter zu **Phasen** (P_i, P_j)*° zusammengefasst werden, innerhalb welcher definierte Signalgruppen freigegeben werden, siehe Anhang B. „Eine Phase ist derjenige Teil eines Signalprogramms, während dessen ein bestimmter Grundzustand der Signalisierung unverändert bleibt. Die Freigabezeiten für die freigegebenen Verkehrsströme können zu verschiedenen Zeitpunkten beginnen und Enden.“ [FGS10] Ein Wechsel zwischen Phasen wird als **Phasenübergang** ($PU_{i,j}$ von i zu j)*° bezeichnet, innerhalb welcher in der Regel alle Wechsel zwischen Freigaben und Sperrungen unter Berücksichtigung der Zwischenzeiten erfolgen, vgl. Abbildung 2.5. Phasen und ihre Phasenübergänge werden in Phasenfolgeplänen festgehalten, siehe Anhang B.

In diesem Zusammenhang findet der Begriff der **Umlaufzeit***° t_U beziehungsweise T_U in [s] Verwendung, siehe Abbildung 2.4 und 2.5. „Die Umlaufzeit eines Signalprogramms ergibt sich allgemein aus der Summe der erforderlichen Freigabezeiten für die maßgebenden Signalgruppen der einzelnen Phasen und den dazwischen liegenden erforderlichen Zwischenzeiten“ [FGS10]. Falls jede Signalgruppe beispielsweise nur einer Phase zugeordnet ist, ist die Umlaufzeit in der Regel die Zeit, die benötigt wird, um jede Phase unter Berücksichtigung der Kapazität des Knotenpunktes (ausreichend lange Freigabezeiten für alle Verkehrsströme) einmal zu durchlaufen. Die Umlaufzeit wird üblicherweise im urbanen Umfeld bei der Anlagenplanung in Abhängigkeit zu den umgebenden LSA festgelegt. Die diskrete Zeit wird sekundlich innerhalb der repetierten Umlaufzeit von einer LSA-Logik gezählt. Diese Zählungen werden als Umlaufsekunden (t_X) bezeichnet.

2.3.3 Eingangsgrößen

Entscheidungslogiken ermöglichen die Eigenschaft der Verkehrsabhängigkeit. Das heißt, um ein Schaltverhalten zu zeigen, welches dem aktuellen Verkehrsgeschehen angepasst ist,

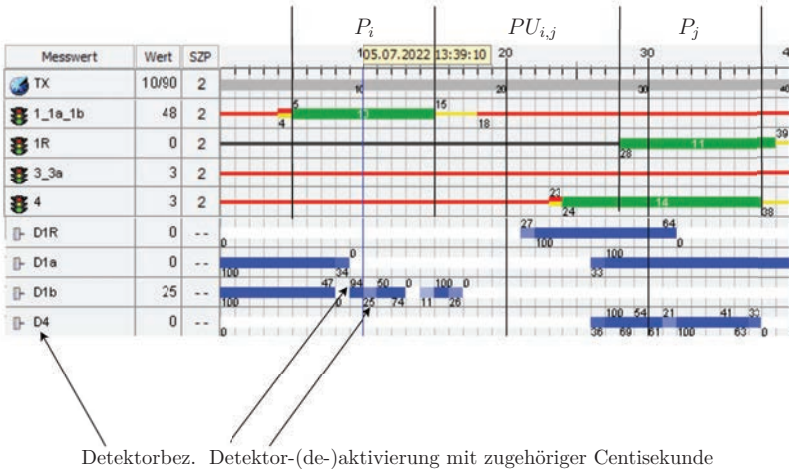


Abbildung 2.5: Beispiel eines Signalzeitenplans mit Detektoren auf Basis von Prozessdaten aus einem Anwenderprogramm, LSA 722, SP 2, (Darstellung nicht RiLSA-konform)

ist eine Erfassung des aktuellen Verkehrszustands notwendig. Dies bedeutet den Schritt von der Verkehrssteuerung zur -regelung. Die zugehörige Datenerfassung geschieht im Regelfall stationär über meist in die Straße eingelassene oder am Signalmast angebrachte Detektoren mit einer Abtastrate von meist 0,1s beziehungsweise 0,01s, was aus den verwendeten Daten der Stadt Kassel abgeleitet werden kann. Detektoren bestehen aus einer Sensorik für die Erfassung von Kenngrößen und einer Auswerteeinheit, welche das Sensorsignal aufbereitet und die gewünschten Daten weitergibt.

Es besteht eine Vielfalt an verfügbaren Detektoren. Da in dieser Arbeit nur ihr Ausgangssignal von Interesse ist, werden an dieser Stelle die Möglichkeiten zu ihrer Gruppierung nur kurz benannt: Messprinzip und Betriebsart [Kob07] beziehungsweise Verwendungszweck, Ausgabegrößen und Anordnungsmöglichkeiten [Wei19a]. Induktivschleifendetektoren bieten die höchste Genauigkeit [Ott11], weshalb sie häufig Verwendung finden. Weitere Informationen zu Detektoren finden sich in den entsprechenden Merkblättern zu den RiLSA [FGS91].

In Abbildung 2.5 sind in der unteren Hälfte vier Detektorsignale beispielhaft aufgezeigt. Protokolliert werden die Überfahreignisse üblicherweise als binäre Information (0, 1 / unbesetzt, besetzt). Alle anderen hier beschriebenen Größen werden aus den Binärsignalen von Detektoren abgeleitet. Die folgenden Begriffsdefinitionen sind an das Merkblatt über Detektoren im Straßenverkehr angelehnt [FGS91].

Anwesenheit beziehungsweise Belegung* BEL_d : Belegungen liegen als Binärsignal vor und geben an, ob sich eine Person beziehungsweise deren Fahrzeug im Wahrnehmungsbereich des Detektors befindet. Sie beziehen sich zumeist auf den mIV zum Beispiel für Anforderungen an der Haltlinie oder Rückstauerkennungen insbesondere an kurzen Abbiegestreifen.

Anforderung* ANF : Eine Anforderung über die Detektoren einer LSA ist eine Anmeldung an der LSA mit der Bitte der Beteiligten um eine Freigabe. Diese Anforderung kann durch mIV (Induktivschleifen), FG (Taster) oder ÖPNV (spezielle Meldeverfahren) zum Beispiel in Form einer Belegung (Passieren eines Sensorfeldes) erfolgen.

An- und Abmeldung*: Sie entsprechen Belegungen und/oder Anforderungen, betreffen den ÖPNV und können in drei Formen vorliegen: Binärsignal, R09.16 Telegramm mit AMLi-Datensatz oder Cooperative Intelligent Transport System (C-ITS)-Nachricht, siehe Abschnitt 2.4.1. Üblicherweise werden Informationen zu 2 bis 5 Annäherungszeitpunkten: Voranmeldungen, Hauptanmeldungen, Abfahrbereitschaften und Abmeldungen (VA, HA, ABF, AB) sowie Türschluss (TUR) versendet.

Belegungszeit beziehungsweise Anwesenheitszeit* $t_{B,d}$ in [s]: Die Belegungszeit gibt die Dauer einer Belegung ($BEL=TRUE$) eines Detektors d an und wird in der Regel für die Ermittlung von Rückstaus und Wartezeiten verwendet.

Wartezeit $t_{W,d}$ in [s]: Dies ist die Dauer von Halten von Verkehrsbeteiligten ggf. inkl. geringen Verzögerungen und Beschleunigungen im Rahmen von Halten, wie beispielsweise kurze Bewegungen bei stockendem Verkehr oder beim Aufrücken in Rückstaus an Knotenpunkten. Wartezeiten werden nicht nur für Fahrzeuge sondern auch für Fußgehende erfasst, da diese besonders geringe Toleranzen gegenüber Wartezeiten in ihrem Verkehrsverhalten zeigen.

Zeitlücke* $t_{L,d}$ in [s]: Eine Zeitlücke ist die Zeit zwischen den Anwesenheiten von Verkehrsbeteiligten im Wahrnehmungsbereich des Detektors d , eine Art Abwesenheitszeit. Es wird zwischen Brutto- und Nettozeitlücken unterschieden. Eine Bruttozeitlücke ist der zeitliche Abstand vom Beginn der Anwesenheitsdetektion eines Fahrzeugs zum Beginn der Anwesenheitsdetektion des ihm nachfolgenden Fahrzeugs. Zur Ermittlung einer Nettozeitlücke wird das Ende der Anwesenheitsdetektion dem Beginn der Anwesenheit des nachfolgenden Fahrzeugs gegenübergestellt. Über Zeitlücken werden **Zeitlückenkriterien** (ZL in $\{TRUE, FALSE\}$) abgefragt, zum Beispiel $ZL > 3$ s [FGS10]. In dieser Arbeit werden im Weiteren nur Nettozeitlücken betrachtet.

Verkehrsstärke q_i in [Fz/h]: Die Verkehrsstärke gibt an, wie viele Verkehrsbeteiligte in einer Stunde einen bestimmten räumlichen Querschnitt überschreiten. Bezüglich FG wird dies in $Personen/h$ angegeben.

Verkehrsdichte k_i in $[Fz/km]$: Die Verkehrsdichte gibt an, wie viele aktive Verkehrsbeteiligte sich mit den von ihnen geführten Fahrzeugen zu einem Zeitpunkt in einem ausgewählten Gebiet, zum Beispiel aller Fahrstreifen einer Verkehrsrichtung, befinden.

Fahrzeugtyp: Damit wird bezeichnet, um welche Fahrzeugart es sich handelt: beispielsweise LKW, PKW, Fahrrad.

Auslastungsgrad α in [%]: Der Auslastungsgrad gibt an, zu wie viel Prozent ein Fahrstreifen beziehungsweise eine Fahrtrichtung ausgelastet ist. Als maximale Auslastung werden üblicherweise ca. 1800 bis 2000 Fahrzeuge pro Fahrstreifen und Stunde angenommen. Der Auslastungsgrad wird in urbanen Gebieten zumeist für unsignalisierte Abschnitte zwischen signalisierten Knotenpunkten zwecks Schätzung der Auslastung des Straßenverkehrsnetzwerkes verwendet und dient als Grundlage zur Dimensionierung sowie Ermittlung der Leistungsfähigkeit von Knotenpunkten und LSA.

Anzahl der Halte n_H : Die Anzahl der Halte gibt an, wie viele und wie häufig Fahrzeuge in der entsprechenden Verkehrsrichtung vor der LSA gehalten haben. Diese Anzahl unterscheidet sich von einer möglichen resultierenden Warteschlangenlänge, da sie alle Halte innerhalb eines definierten Zeitraums (zum Beispiel Erfassungszeitraum) erfasst.

Die **Warteschlangenlänge** entspricht der Länge des beobachteten Rückstaus zu einem Zeitpunkt.

2.4 Kommunikative Straßenverkehrsinfrastruktur

Straßenverkehrsnetze sind räumlich verteilte Systeme, in denen die am Verkehr beteiligten Personen als Individuen Entscheidungen treffen. Es wird versucht, die Anwesenheit und das Verhalten der Beteiligten im System zu beeinflussen, beispielsweise über Beschilderungen und LSA. Es wird versucht den Anwesenheitszeitpunkt im Verkehrsnetzwerk beispielsweise über Verkehrsprognosen zu beeinflussen, um das Verkehrsaufkommen in Stößzeiten zu reduzieren. Neuer ist die Einflussnahme mittels kommunikativer Infrastrukturelemente, welche Informationen von Beteiligten erhalten können oder über digitale Medien an diese übermitteln können. Schaltzeitprognosen liefern hierfür grundlegende Daten, siehe [Sch18].

Die Entwicklung der Digitalisierung der Straßenverkehrsinfrastruktur bezüglich LSA ging ausgehend von der lokalen Lichtsignalanlage, welche ausschließlich Signalgeber schaltet, über die Funktionsüberwachung von LSA, in der ihr Schaltverhalten sowie Betriebszustände und Fehler protokolliert werden, hin zur Vernetzung von LSA, die über eine Steuerzentrale mit gegebenenfalls untergeordneten Steuerzentralen für Netzabschnitte verfügen. Die neueste Entwicklung sind die C-ITS-fähigen LSA(-Netzwerke). C-ITS sehen die Kommunikation aller Elemente im Netzwerk (Systembestandteil-X zu Systembestandteil-

X (X2X)) vor, insbesondere zwischen Infrastrukturelementen wie LSA (Infrastruktur zu Infrastruktur-Kommunikation (I2I)) und Fahrzeugen (Vehicle to Vehicle Kommunikation (V2V)). Ein Forschungsprojekt, das sich mit C-ITS-fähigen LSA beschäftigte, war das Projekt VERONIKA am Fachgebiet Verkehrstechnik und Transportlogistik der Universität Kassel [HMSN20].

Eine zentrale Frage für die Strukturierung von Datennetzwerken, wie denen innerhalb der Verkehrsinfrastruktur, ist die Entscheidungshierarchie beziehungsweise -struktur. Es gibt drei Möglichkeiten, C-ITS zu strukturieren: zentral, hierarchisch und verteilt. In zentralen Strukturen werden Entscheidungen von einem Element, wie zum Beispiel einer Steuerzentrale, zentral für alle Elemente des (LSA-)Systems getroffen. Ein Beispiel hierfür ist das Sydney Coordinated Adaptive Traffic System (SCATS), welches vorwiegend in Australien, Neuseeland und China eingesetzt wird, siehe auch [Pro16].

In verteilten Systemen werden Entscheidungen von jedem Systemelement selbst getroffen. Bei Bedarf wird eine Kommunikationsmöglichkeit zu anderen Elementen des Systems aufgebaut, um zusätzliche Informationen zu erhalten. Falls keine Kommunikation mit anderen Elementen vorliegt, handelt es sich um eine rein lokale Steuerung beziehungsweise Regelung. In hierarchischen Systemen werden Entscheidungen auf verschiedenen Ebenen getroffen. Neben den Entscheidungen, die den einzelnen LSA überlassen werden, gibt es übergeordnete Entscheidungsträger, die Entscheidungen für Gruppen oder das gesamte Netzwerk treffen. Das heißt, Instanzen übergeordneter Ebenen geben den ihnen untergeordneten Ebenen einen veränderlichen Entscheidungsrahmen.

Ein hybrides System vereinigt Eigenschaften von hierarchischen und verteilten Systemen. Ein Beispiel hierfür ist Split Cycle Offset Optimisation Technique (SCOOT), welches in Großbritannien entwickelt wurde und zusätzlich Anwendung in den USA, Lateinamerika und Asien findet, siehe Abschnitt 2.5.2 und [Pro16].

Intelligent Transport Systems ITS verteilen ihre Entscheidungen temporal und spatial über das Netzwerk auf verschiedene Elemente. Sie treffen ihre Entscheidungen in unterschiedlichen Takten. Üblich sind, falls vorhanden, zwei bis drei Ebenen der Entscheidungsfindung. Die RiLSA sehen bis zu zwei Ebenen vor (siehe Tabelle 2.1 auf Seite 41). Entscheidungsprozesse im Straßenverkehr, die von Nutzern und Teilen der Infrastruktur durchgeführt werden, werden in drei Arten unterteilt: lang- (strategisch), mittel- (taktisch) und kurzfristig (operativ).

Langfristig geht es um die Frage, wann, wie und womit der anstehende Reiseweg bewältigt werden soll. Für diese Entscheidungen relevante Kenngrößen sind makroskopische Kenngrößen, wie die mittlere Geschwindigkeit, die Verkehrsstärke und Verkehrsdichte. Auf dieser Ebene werden Entscheidungen in einem Zeitraum von beziehungsweise mit Auswirkung auf die nächsten 5 bis 15 Minuten getroffen.

Mittelfristig werden Fragen der Wegplanung (wo und wie) beantwortet.

Kurzfristig werden Entscheidungen im aktuellen Moment getroffen, wie zur Wegwahl an Kreuzungen für den Fahrstreifenwechsel; diese basieren eher auf mikroskopische Kenngrößen wie zum Beispiel auf den Geschwindigkeiten und dem (Fahr-)verhalten der umgeben-

den am Straßenverkehr Beteiligten. Entscheidungen von Seiten der Infrastrukturelemente fallen auf dieser Entscheidungsebene sekundlich. Die Benennung dieser Ebenen erfolgt entweder mit den Perspektiven – mikro- und makroskopisch – oder nach den Arten der Entscheidungsfindung – strategisch, taktisch und operativ.

Die Entscheidung zwischen zentralen oder dezentralen Entscheidungsstrukturen über ITS hinaus wird disziplinübergreifend in der Literatur diskutiert, zum Beispiel für den Ausbau des elektrischen Versorgungsnetzes. Es gibt einige Argumente für (hierarchisch) verteilte Systeme: Sie sind aufgrund ihrer lokalen Entscheidungsinstanzen robuster gegen Störungen und zuverlässiger bei Ausfall von Entscheidungseinheiten, da nicht das Gesamtsystem, sondern nur einzelne Komponenten ausfallen, während die verbleibenden Instanzen weiterarbeiten. Sie sind in der Regel durch die Verteilung der Entscheidungen auf mehrere Computer für Echtzeitanwendungen geeigneter, da sie eine höhere Entscheidungsfrequenz und -kapazität aufweisen. Lösungen, die von verteilten Systemen realisiert werden, können im Gegensatz zu zentralen Systemen jedoch nicht mehr als mathematisch optimal im Hinblick auf eine optimale Steuerung/Regelung angesehen werden. Allerdings erzielen sie näherungsweise optimale Ergebnisse, siehe zum Beispiel [DC10] oder [FC12].

Ein weiterer Entscheidungsfaktor bezüglich der Netzwerkinfrastruktur ist der Steuerungsbezug: Soll eine LSA an einem einzelnen Knotenpunkt, an mehreren LSA eines Straßenzugs oder an LSA eines gesamten Netzwerks aufeinander abgestimmt werden? Entsprechend gibt es knotenpunkt-, linien- und netzbezogene Systemauslegungen. Die Autoren von [Se16] präsentieren beispielsweise ein Konzept für ITS namens Autonomic Transport Management System (ATMS). Sie entwerfen eine Netzwerkstruktur mit verschiedenen Hierarchieebenen [Ae16]. Das Konzept beschreibt eine Netzwerkstruktur, die Daten über Services in einem Backend sammelt und bereitstellt. Dieses Netzwerk ist mit autonomen Systemen verbunden, zum Beispiel mittels On-Board-Units (OBU) und Anwendungen für Smartphones. Die kooperierenden Geräte treffen ihre eigenen Entscheidungen auf der Grundlage der Netzwerkdaten. Die Verbindung zwischen dem Backend und den lokalen Sensoren und Detektoren ist in [ŽK16] beschrieben.

2.4.1 Datenquellen und Informationsübertragung

Für den reibungslosen Ablauf innerhalb von ITS werden verschiedene Datenquellen benötigt, die die in Abschnitt 2.3 besprochenen Größen zur Verfügung stellen können. Zunächst wird auf spezielle Kommunikationsmethoden für den ÖPNV eingegangen und hiernach auf den (nicht) motorisierten Individualverkehr (n)MIV.

Die Kommunikation der ÖPNV-Fahrzeuge mit LSA dient dazu, die Bevorrechtigung des ÖPNV beispielsweise in Deutschland umzusetzen und erfolgt mittels R09.16-Telegrammen oder C-ITS-Nachrichten. Der ÖPNV kann aufgrund seiner Bevorrechtigung Freigaben an Knotenpunkten anfordern, welche dem entsprechenden Fahrzeug möglichst zeitnah zum Bedarfszeitpunkt zugesprochen werden. R09.16-Telegramme stellen ein Protokoll für sol-

che Meldungen dar, das seit den 1980er-Jahren erfolgreich im Einsatz ist. Mittels R09.16-Telegrammen werden genormte Meldungen an ortsfesten Meldepunkten von den ÖPNV-Fahrzeugen an zugehörige LSA versendet. Die Meldung wird mit Hilfe von Odometrie ausgelöst und per Funk an die umgebenden LSA übertragen, von welchen das Signal zur auf dem Fahrweg nächsten LSA geleitet wird. Referenz für die Odometrie ist ein Infrarot-signal, das beim Passieren des Meldepunktes ausgetauscht wird. In den LSA-Logiken sind die für sie relevanten Meldepunkte über ihre Nummer hinterlegt. Ein Meldung enthält Informationen zu

- Datum und Uhrzeit der Versendung,
- Meldepunktnummer,
- Priorität,
- Zuglänge,
- Richtung Hand (manuelle Anforderung des Fahrers),
- Linien-, Routen- und Kursnummer sowie
- Fahrplanabweichung.

Anmeldungen werden, falls keine Abmeldung erfolgt, innerhalb einer definierten Zeitspanne in Form einer Zwangsabmeldung verworfen. Von LSA können mehrere gleichzeitige ÖPNV-Meldungen untereinander priorisiert werden.

Die AMLi-Datensatzerweiterung stellt des Weiteren

- Teilknotennummer,
- Modifikation (Annahmefähigkeit der LSA von ÖPNV-Meldungen),
- Zeitschalter (t_x der Meldung),
- Signalplan,
- laufende Phase,
- gewünschte Phase,
- (angenommene) Fahrzeit von An- bis Abmeldung und
- GA und GE ÖPNV-SG

zur Verfügung.

C-ITS-Nachrichten stellen einen neuen Entwicklungsschritt dieser Meldetechnik dar, über die mehr Informationen als über die R09.16-Telegramme übertragen werden können [OCI22b]. Der Standard für C-ITS wird in den Kommunikationsnormen für Intelligent

Transport Systems (ITS) festgelegt: ETSI EN 302 665 [Eur10] und ISO 21217 [Int14]. Für die Informationsübertragung mittels C-ITS ist normalerweise an den LSA eine Road Side Unit (RSU) angebracht; auf der Gegenseite befindet sich je Fahrzeug eine On-Board-Unit (OBU). Im Gegensatz zu R09.16-Telegrammen werden hier verschiedene Nachrichten über standardisierte Protokolle ausgetauscht: Cooperative Awareness Message (CAM), Straßentopologie (MAP), Signal Phase and Timing (SPaT) und Decentralised Environmental Notification Message (DENM). In Kassel wird der Einsatz der C-ITS-Technik für den ÖPNV, Einsatzfahrzeuge der Feuerwehr und Rettungsdienste nicht nur getestet [HMSN20], sondern flächendeckend eingerichtet. In Zukunft ist vorgesehen, dass diese Kommunikation allen Verkehrsbeteiligten zur Verfügung steht.

Die C-ITS-Technik ermöglicht es, Floating Car Data (FCD) direkt zwischen Fahrzeugen sowie mit der Infrastruktur auszutauschen. FCD sind alle Daten, welche die Sensorik eines Fahrzeugs aufnehmen und zum Beispiel über eine OBU zur Verfügung stellen kann (FCD). Forschungsansätze hierzu bestehen bereits seit den 1980er Jahren [Ott11]. Sie spielen eine wichtige Rolle für die Entwicklung intelligenter Infrastruktur und des autonomen Fahrens, da Informationen der unmittelbaren Umgebung von LSA hierdurch verfügbar sind. Aktuell spielen FCD keine Rolle für Entscheidungsprozesse von LSA, da LSA nur (historische) Daten stationärer Sensoren verarbeiten. FCD spielen in dieser Arbeit keine Rolle.

2.4.2 Open Communication Interface for Road Traffic Control Systems (OCIT)

Für die Übertragung und Aufzeichnung von Ein- und Ausgangsdaten von LSA existiert das Open Communication Interface for Road Traffic Control Systems (OCIT). Es setzt Standards für die Datenübertragung fest, um eine Kommunikationsfähigkeit zwischen Infrastrukturbauteilen herstellerübergreifend gewährleisten zu können. Die Entstehung des OCIT geht auf eine Initiative aus dem Jahr 1999 zurück, in der sich Signalbaufirmen, Hersteller und Betreiber für Verkehrssteuerungs- und -managementsysteme, Baulastträger und der Verband der Ingenieurbüros für Verkehrstechnik zusammengeschlossen haben [Wei17]. Es gibt mehrere Schnittstellenstandards [OCI22c], zum Beispiel OCIT-LED für die Kommunikation zwischen Steuergerät und Signalgeber, OCIT-O für die Kommunikation zwischen Steuergerät und Steuerzentrale und OCIT-C für die Kommunikation zwischen Steuerzentralen. Eine Sammlung der Standards findet sich in [OCI22a]. In diesem Rahmen legt dieser Standard auch Daten von LSA fest, welche zu protokollieren sind, um zum Beispiel den Anforderungen an eine Qualitätssicherung der RiLSA [FGS10] nachzukommen.

Die übertragenen Daten werden zumeist in einer (Verkehrs-)Zentrale gebündelt. Die Zeitspanne von der Datenerfassung an der LSA bis zum Eintreffen der Daten in der Verkehrszentrale wird **Latenzzeit** genannt. Sie kann mehrere Sekunden betragen, siehe Abschnitt ???. Zudem können bei der Datenerfassung an verschiedenen Punkten der Datenübertragung zwischen Sensor, Detektor, Eingabe ins Steuergerät, Datenübermittlung an die Zentrale und Speichermedium Daten verloren gehen.

OCIT-Daten werden in einer standardisierten Struktur gespeichert. Hierzu gehören sechs wesentliche Unterteilungen:

- „IST-Vektoren“, auch „IST“ bezeichnet, enthält die Betriebszustände der LSA,
- „SIG“, auch „sg“ bezeichnet, umfasst die Zustände der Signalgeber: Signalbild, siehe Abschnitt 2.3.2,
- „det“, Detektordaten, siehe Abschnitt 2.3.3,
- „oev“, ÖPNV Meldepunkte beziehungsweise Meldungen, siehe Abschnitt 2.4.1,
- „tx“, Umlaufsekunde t_x , siehe Abschnitt 2.3.2,
- „sen“, aggregierte Sensordaten an den Abflüssen des KP

„IST“: Die Datengruppe des IST-Vektors enthält die aktuelle

- Signalprogrammnummer, ganzzahlig zwischen 0 und 255 (8-bit unsigned integer),
- Sammelstörungsklasse in 5 Zuständen von keine Störung, über Teilstörungen bis zur Abschaltung der Anlage,
- Betriebsart in 6 Zuständen: Lokaler Betrieb, Zentralenbetrieb, Sonderbetrieb, ...,
- KnotenEinAus in 6 Zuständen: Ein, Aus mit Blinken ...,
- Sondereingriffe (max. 254)
- VAEinAus (Verkehrabhängigkeit) in 3 Zuständen: Freigabe, Ein, Aus,
- IVAEinAus: Zustand der VA, wenn eingeschaltet,
- OepnvEinAus in 3 Zuständen: Steuerung mit / ohne / freigegebene ÖPNV-Bevorzugung,
- Zeitpunkt der letzten Änderung in Millisekunden im Unixzeitformat,
- Teilknotennummer- und Zustand, der die Einstellung bezüglich des Gesamtknotens beschreibt.

„SIG“: Bei einem Wechsel des Signalbilds einer Signalgruppe wird der neue Zustand gespeichert. Die wichtigsten Kodierungen sind:

- 00 Dunkel,
- 03 Rot,
- 12 Gelb,
- 48 Grün,
- 15 Rot/Gelb,

- 24 gelb blinken (1 Hz),
- 60 Besonderheit für Kassel: FG-Grün und Blindensignal; eigentlich gleichzeitiges Gelb und Grün.

„**det**“: Die Speicherung von Detektorzuständen an den Zuflüssen des Knotenpunkts erfolgt als Binärinformation und ist wie das Signalbild ebenfalls ereignisorientiert, wobei der Ereigniszeitpunkt auf eine Millisekunde genau festgehalten wird.

„**oev**“: Unter den ÖPNV-Daten werden die R09.16-Telegramme und vergleichbare Daten gespeichert, siehe Abschnitt 2.4.1.

„**tx**“: Die Umlaufsekunde t_x wird sekundlich gezählt und beginnt mit 0. Zudem kann die Umlaufzeit t_u variieren, da durch die regelmäßige Synchronisation der Uhrzeiten aller LSA im Netz Umlaufsekunden entfallen oder hinzugefügt werden können.

„**sen**“: Die aggregierten Sensordaten der KP-Abflüsse enthalten über jeweils einen Umlauf gebündelte Daten, zum Beispiel Belegungsraten, -häufigkeiten, Geschwindigkeiten sowie Fahrzeugtypen und werden in dieser Arbeit nicht verwendet, da sie im Allgemeinen keine Eingangsgrößen von LSA sind, sondern der Verkehrszählung dienen.

Alle oben beschriebenen Datenstrukturen werden bei einer Änderung ihres Zustands, also wenn ein Ereignis eintritt, als neue Zeile in eine Protokolldatei geschrieben. Hierbei ist eine Spalte für die Kodierung der Ereignisart vorgesehen, Detektoren werden beispielsweise mit 41.32 und 41.135 für SG kodiert.

2.5 Entscheidungslogiken

In den vorausgegangenen Abschnitten wurden zu LSA gehörige Kenngrößen sowie ihre Speicherungsprotokolle beschrieben. Wie diese Kenngrößen bestimmt werden, soll in diesem Abschnitt beschrieben werden.

2.5.1 Grundlegende Prinzipien

Im Folgenden werden die grundlegenden Prinzipien der Steuerungs- und Regelungsverfahren für LSA vorgestellt. Neben der Architektur von Verkehrsinfrastruktur mit ihren Ebenen können LSA über ihre Programme beschrieben werden. Dieser gesamte Abschnitt orientiert sich an den RiLSA [FGS10]. Die nachfolgenden Absätze sprechen die **makroskopischen (Steuerungs-)Verfahren** an, vgl. Tabelle 2.1.

Neben knotenpunkt-, linien- und netzbezogenen Steuerungsprinzipien, entsprechend den Perspektiven aus Abschnitt 2.2, gibt es Prinzipien zur Systemgestaltung:

	Steuerungsverfahren		Daten- erfassung	Aktivierung		verkehrsabhängig veränderbare Elemente der Signalprogramme				
	Oberbegriff	Hauptmerkmal der Veränderbarkeit der Signalprogramme		zeitplan- abhängig	verkehrs- abhängig	Umlauf- zeit	Phasen- folge	Phasen- anzahl	Freigabe- zeiten	Versatz- zeiten
Makroskopische Steuerungsebene	Signalprogramm- auswahl	zeitplanabhängige Auswahl der Signal- programme	offline	X		in Kombination mit einem Steuerungsverfahren der mikroskopischen Steuerungsebene veränderbare Elemente des Signalprogramms				
		verkehrsabhängige Auswahl der Signal- programme	online		X					
	Bildung Rahmen- signalprogramm	verkehrsabhängige Bildung der Rahmen- signalprogramme	online		X					
Mikroskopische Steuerungsebene	Festzeitsignal- programm		offline	Aktivierung gemäß Steuerungsverfahren der makroskopischen Steuerungsebene						
		Freigabezeit- anpassung	online					X		
	Signalprogramm- anpassung	Phasentausch	online			X				
		Phasen Anforderung	online				X	X		
		Versatzzeit Anpassung	online						X	
	Signalprogramm- bildung	freie Veränderbarkeit	online			X	X	X	X	X

Tabelle 2.1: Übersicht über die Steuerungsverfahren [FGS10]

- **Verkehrsführung, angebotsorientiert** und
- **Verkehrs Anpassung, nachfrageorientiert.**

Bei der Verkehrsführung wird der Verkehr meist über (koordinierte) Streckenzüge geführt, ohne dass die Freigaben aufwändiger an das tatsächliche Verkehrsaufkommen angepasst werden. Ihre Auslegung beruht auf offline erfassten Daten, wie beispielsweise Verkehrszählungen. Die Grünzeiten können als Angebote an den Verkehr betrachtet werden. Dieses Prinzip tendiert dazu, in Form einer Festzeitsteuerung realisiert zu werden.

Die Verkehrs Anpassung hingegen orientiert sich am aktuellen Verkehrsaufkommen und versucht die Grünzeit entsprechend dieser Nachfrage zu regeln. Hierzu ist eine Online-Erfassung der Verkehrsdaten notwendig, weshalb Detektoren für diese LSA installiert sein müssen. Dies bedeutet einen technischen Mehraufwand im Vergleich zur Verkehrsführung.

Weitere Steuerungsprinzipien sind Orientierungen:

- **signalgruppenorientiert** und
- **phasenorientiert.**

Entscheidungslogiken ohne Phasendefinition, die anhand von einzelnen Signalgruppen Entscheidungen treffen, werden als signalgruppenorientiert bezeichnet. Dies bedeutet, dass über jeden Verkehrsstrom, im Idealfall orientiert am aktuellen Verkehrsaufkommen, sekundlich über den GA und das GE jeder Signalgruppe entschieden wird.

Verbreiteter, da intuitiver, sind phasenorientierte Logiken, welche über den Wechsel zwischen Phasen entscheiden. Die Abfolge von Phasen kann fest oder variabel sein. Viele LSA-Programme arbeiten mit phasenorientierten Logiken, welche trotzdem über GA und GE einzelner Signalgruppen entscheiden, siehe Abschnitt 2.5.2.

An einem Knotenpunkt läuft zu einem Zeitpunkt meist nur ein LSA-Programm. Dieses Signalprogramm wird üblicherweise

- **zeitplanabhängig** oder
- **verkehrsabhängig**

gewechselt. Dies wird aus Sicht der RiLSA als **makroskopische Steuerungsebene** betrachtet und ist mit jedem Verfahren der mikroskopischen Ebene kombinierbar, siehe Tabelle 2.1.

Ein zeitplanabhängiger Austausch des Programms findet dann statt, wenn das Verkehrsaufkommen vorhersehbar ist, zum Beispiel wenn der Berufsverkehr keinen größeren Schwankungen unterliegt. Der Tauschzeitpunkt zwischen den Programmen ist dann vom Wochentag und von der Uhrzeit abhängig. Die Regelmäßigkeit des Verkehrsaufkommens stellt damit eine Kenngröße dar, die durch Verkehrszählungen ermittelt wurde und auf zukünftige Zeitpunkte projiziert wird.

Ein verkehrsabhängiger Tausch des Programms geschieht in Abhängigkeit von online erfassten Verkehrsdaten und wird meist von Schwellwertüberschreitungen ausgelöst. Gewünscht ist hierbei eine Frequenz der Programmwechsel, die mehrere Umläufe in einem Programm vor einem Wechsel zulässt. Die verkehrsabhängige Signalprogrammauswahl stellt ein makroskopisches Steuerungsverfahren dar.

Neben der Signalprogrammauswahl ist es möglich, durch die makroskopischen Größen (siehe Abschnitt 2.2) lediglich ein **Rahmenprogramm** mit **Erlaubnisbereichen** für Freigaben vorzugeben, welches unter Zuhilfenahme von mikroskopischen Größen möglichst optimal zu einem Signalplan ergänzt wird.

2.5.2 Programmtypen

Bei den **mikroskopischen Steuerungsverfahren** wird nach den RiLSA zwischen Festzeitsignalprogramm, Signalprogrammanpassung und Signalprogrammbildung unterschieden. Tabelle 2.1 zeigt die Klassifikation von Steuerungsverfahren nach den RiLSA [FGS10]. Sie teilt die Verfahren entsprechend Blickwinkel (Abschnitt 2.2), (nicht) vorhandener Erfassung aktueller Verkehrsdaten und veränderbarer Elemente (Variablen) ein und benennt sie nach Hauptmerkmalen. Zudem können Steuerungs- und Regelungsverfahren von LSA nach Implementierungsart Festzeitsteuerung, regel- und modellbasierten Verfahren unterschieden werden. Dies wird in dieser Arbeit aufgrund des Fokus auf die Methoden zur Vorhersage von LSA-Programmen bevorzugt. Der Zusammenhang zwischen diesen Entscheidungen ist folgender: Signalprogrammanpassung und Signalprogrammbildung können regel- und modellbasiert sein. Bei der Signalprogrammanpassung werden einzelne Kenngrößen zur Laufzeit verändert, mitunter auch in Kombination. Bei der Signalprogrammbildung können mehrere Kenngrößen gleichzeitig verändert werden.

Eine weitere Perspektive auf diese strategischen, makroskopischen Aspekte von Steuerungsverfahren wird in [Pro16] geboten, siehe Abbildungen 2.6 und 2.7 aus [Pro16, Sta14]. Die Kategorien A bis D sind mit fallender Realisierbarkeit einer Schaltzeitprognose und Zuverlässigkeit von 100 % bis nicht vorhersagbar geordnet:

- Kategorie A: festzeitgesteuerte LSA, keine Verkehrsabhängigkeit,
- Kategorie B: festzeitgesteuerte LSA mit gelegentlichen Modifikationen, schwach verkehrsabhängige LSA,
- Kategorie C: stark verkehrsabhängige sowie modellbasierte LSA oder Festzeit-LSA mit ÖPNV-Priorisierung,
- Kategorie D: LSA ohne Verbindung zu einer Verkehrszentrale; keine Aussage zu Vorhersagen möglich.

Es ist zu sehen, dass die meisten der befragten Städte über LSA verfügen, für die eine Vorhersage möglich ist. Die Frage nach einer ausreichenden Zuverlässigkeit solcher Prognosen ist damit nicht beantwortet, da der Anteil an LSA der Kategorie C (besonders schwer vorhersagbar) hoch ist und damit zu erwarten ist, dass mit der Digitalisierung der Verkehrsinfrastruktur ihr Anteil weiter steigen wird.

Die in Abbildung 2.8 zusammengefasste Studie bestätigt diese Forschungsergebnisse für Deutschland. Die Kategorisierung ist hier anders gewählt. Die hinter den Kategorien stehenden Verfahren der mikroskopischen Steuerungsebene werden im Folgenden näher erläutert.

Allgemein ist zu bemerken: Je mehr die verkehrsabhängigen Aspekte in einer LSA-Regelung bzw. Steuerung berücksichtigt werden, desto komplexer wird ihre Logik. Dies erfordert eine Vernetzung der LSA und geringe Latenzzeiten für den hierfür notwendigen Informationsaustausch. Je komplexer die Schaltlogiken werden, desto schwieriger ist es, deren Zeitverhalten vorherzusagen. Dies steht im Gegensatz zur Entwicklung ihrer Relevanz für C-ITS.

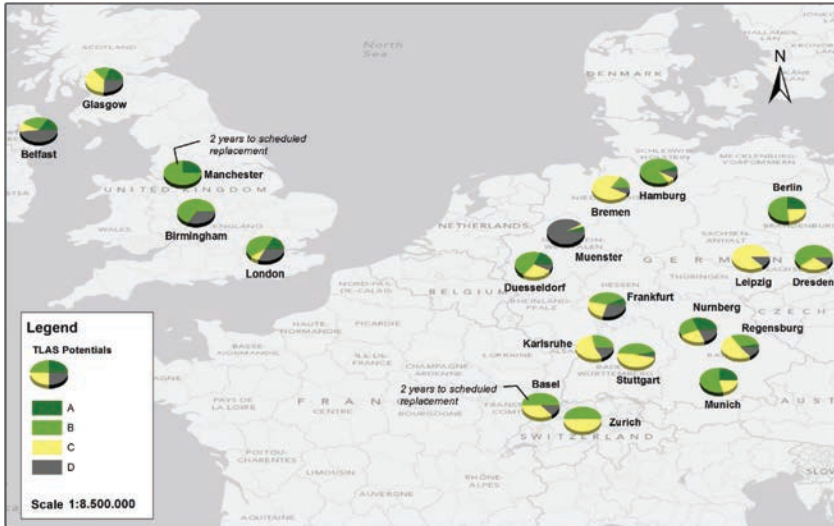


Abbildung 2.6: Anwendung von Steuerungsverfahren verschiedener Städte in GB und im D-A-CH-Raum [Pro16]

Festzeitprogramme

Festzeit-Programme bzw. FZ-Programme sind die einfachsten, robustesten und ausfallsichersten Steuerungsprogramme. Sie folgen einem festen Signalzeitenplan und arbeiten diesen zyklisch ab. Umlaufzeit, Phasenfolge, Freigabedauern sind terminiert. Grundlage für die Auslegung des Plans sind historische (makroskopische) Verkehrskenngrößen. Es handelt sich um eine reine Steuerung des Verkehrs ohne die Verwendung von Detektoren.

FZ-Programme weisen den geringsten Aufwand an Zeit und Kosten für Entwicklung und Wartung auf. Sie sind zu 100 % vorhersagbar. Jedoch kann mittels dieser keine Anpassung an den aktuellen Verkehr erfolgen, das heißt auch der ÖPNV kann bei einer reinen FZ-Steuerung nur priorisiert werden, indem ihm im Umlauf feste Freigaben eingeräumt werden.

FZ-Programme werden eingesetzt, um die Grundlage für die Entwicklung verkehrshängiger Regelungen bzw. Steuerungen zu bilden. Außerdem werden sie für Linien- und Netzkoordinierungen verwendet. Zudem gewährleisten sie eine Rückfallebene für VA-Regelungen bzw. Steuerungen. In Abbildung 2.8 sind FZ-Programme anteilig unter festzeitgesteuerten LSA und verkehrshängiger Signalprogrammauswahl zu finden. Für diese Arbeit spielen sie eine untergeordnete Rolle, da sie vollständig rekonstruiert werden können.

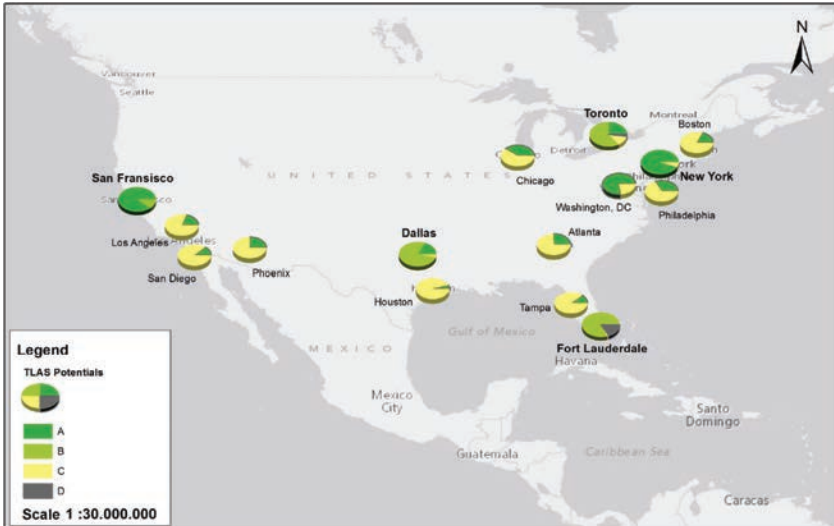


Abbildung 2.7: Anwendung von Steuerungsverfahren verschiedener Städte in den USA [Pro16]

Alle im Weiteren beschriebenen Verfahren sind verkehrabhängig (VA). Damit unterliegen diese Verfahren einer Veränderlichkeit, wodurch eine SZP mit einer Zuverlässigkeit von 100 % nicht mehr möglich ist, da sie von Millisekunden-genauen Detektoraktivierungen abhängen, welche aufgrund des Faktors des menschlichen Verhaltens nicht so präzise vorhergesagt werden können.

Regelbasierte Programme

Ca. 70 % der LSA in Deutschland sind regelbasierte (RB) VA-Logiken. In Abbildung 2.8 sind sie den verkehrabhängigen Steuerungsverfahren zugeordnet. Regelbasiert bedeutet, dass das LSA-Programm über Wenn-Dann-Logikimplementierungen Entscheidungen trifft.

Der Ablauf eines solchen Programms ist in Abbildung 2.9 links zu sehen. Die Kenngrößen werden erfasst und mit einer Auswertelogik ausgewertet. Hierzu werden logische, zeitliche oder sonstige individuell an den Knotenpunkt angepasste Bedingungen sowie die aktuellen

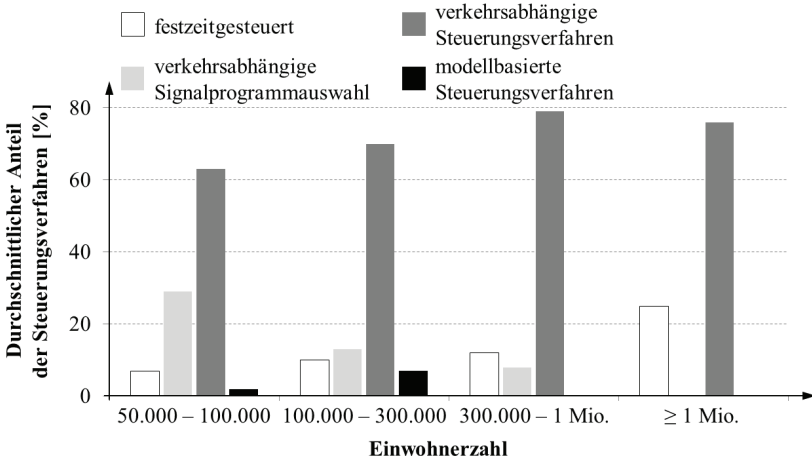


Abbildung 2.8: Anwendung von Steuerungsverfahren verschiedener Städte in Deutschland [Fe14]

und vergangenen Systemzustände der LSA herangezogen. Ihre Kombination entscheidet deterministisch über die ausgegebenen Schaltbefehle. Nicht deterministisch sind im Gegensatz hierzu die Belegungen der Detektoren durch den Verkehr, welche das Verhalten der am Straßenverkehr Beteiligten spiegelt.

Diese Logiken können als **Signalprogrammmanpassung** und Signalprogrammbildung implementiert werden. Bei der Realisierung als Signalprogrammmanpassung erfolgen nur einzelne der folgenden Punkte:

- **Freigabezeitanpassung.**
- **Phasenanforderung.**
- **Phasentausch** oder
- **Versatzzeitanpassung.**

Bei einer **Freigabezeitanpassung** existiert ein Erlaubnisbereich, in dem die Freigabe in jedem Fall erfolgen sollte. Ein früherer GA oder GE der SG kann beispielsweise über eine **Zeitlückensteuerung**, eine Belegungsgrad- oder eine Rückstauerfassung realisiert werden. Bei der **Phasenanforderung** von sogenannten Bedarfsphasen wird innerhalb einer festen Phasenfolge eine Phase eingeschoben; dies geschieht hauptsächlich, um die

Bevorrechtigung des ÖPNV zu realisieren oder für FG, falls diese nicht in den stets geschalteten Phasen vorgesehen sind. Bei einem **Phasentausch** ist eine Veränderung der Phasenfolge möglich, indem bestimmte Phasen Übersprungen oder einer anderen Reihenfolge abgearbeitet werden. Bei einer **Versatzzeitanpassung** wird der zeitliche Versatz zwischen Freigaben mit benachbarten Knotenpunkten koordiniert, mit denen gemeinsame Verkehrsströme bestehen.

Dem gegenüber steht die **Signalprogrammgebung**. Dabei können alle oben genannten Anpassungen stattfinden. Zusätzlich können die möglichen Phasen entsprechend der möglichen Phasenübergänge und Freigabezeiten frei hintereinander geschaltet und angepasst werden. Auch die Umlaufzeit kann variieren. Solche Programme schalten zumeist nur bei Anforderung, was bedeutet, dass sie ohne Anforderungen in einen Grundzustand fallen. Mögliche **Grundzustände** sind:

- **Alles-Rot/Sofort-Grün**,
- **Hauptrichtung Dauergrün**,
- **Freie Signalprogrammgebung**.

Bei **Alles-Rot/Sofort-Grün** sind alle Signalgruppen gesperrt, bis eine Anforderung erfolgt, für welche daraufhin möglichst zeitnah eine Freigabe erfolgt. Erfolgt die Umsetzung über die **Hauptrichtung Dauergrün**, wird die Hauptrichtung, bis eine Anforderung der Nebenrichtungen erfolgt, freigegeben. Dies birgt das Risiko, dass in Hauptrichtung vor dem Knotenpunkt beschleunigt wird, „um noch kurz über die Kreuzung zu kommen“. Hierdurch steigt das Unfallrisiko, da sich dieser Effekt über mehrere Knotenpunkte hinweg kumulieren kann. Bei der **freien Signalprogrammgebung** verweilt das Programm in der letzten angeforderten Phase, bis eine dem entgegenstehende Anforderung erfolgt.

Bei RB-Logiken handelt es sich um Ablauflogiken, bei denen manuell im Vorhinein alle Reaktionsmöglichkeiten festgelegt werden. Dies führt dazu, dass es zu zum Teil plötzlichen und starken Eingriffen in den Verkehrsablauf kommt; zudem kommt diese Möglichkeit der Programmgestaltung bei hohen Knotenpunktauslastungen an ihre Grenzen [Ott11].

Modellbasierte Signalprogramme

Bei modellbasierten (MB) Signalprogrammen handelt es sich um eine Ausprägungsform der modellprädiktiven Regelung (MPR). MPR ist eines der Regelungsverfahren, die in der Industrie breite Anwendung finden [May14]. Mayne gibt in [May14] einen Überblick über die grundlegenden Ausprägungen von MPR zusammen mit einer knappen Einführung. In [QB03] wird ein Überblick über die Geschichte des Verfahrens im Kontext industrieller Anwendungen gegeben.

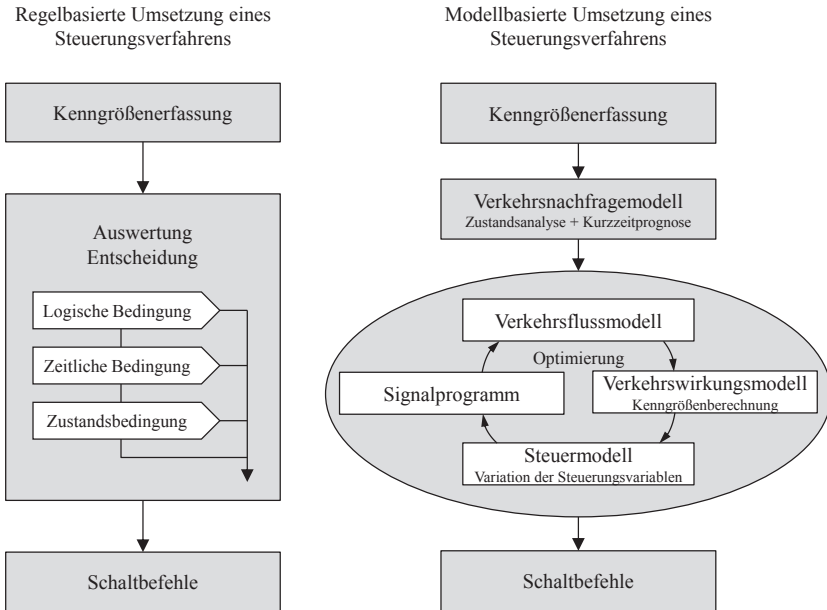


Abbildung 2.9: Umsetzung von regel- und modellbasierten Steuerungsverfahren [FGS10]

MPR steuert ein System mit einem mathematischen Modell dieses Systems. Mittels dieser Abbildung des Systemverhaltens können auf Basis des aktuellen Zustands die zukünftigen Zustände vorhergesagt werden. Diese Vorhersage wird in ein Optimierungsproblem mit Randbedingungen umformuliert. Als Optimierungsgrößen können beispielsweise Wartezeiten, Rückstaulängen oder Halte verwendet werden. Die Lösung der Optimierung ist die neue Steuervariable, bei LSA das zugehörige Signalbild. Die Vorhersage erfolgt zu jedem Zeitpunkt über einen Vorhersagehorizont H_p in i -Schritten. Das Optimierungsproblem wird in der Regel über eine Kostenfunktion formuliert, die im klassischen Fall eine Abweichung von Systemvariablen zu definierten Referenzen bestraft. Bei LSA sind beispielsweise möglichst geringe Rückstaulängen oder auch Wartezeiten sinnvolle Referenzen. Die Abweichung wird minimiert mit dem Ziel, Systemvariablen an ihre Referenzen anzunähern [Mac02].

Abbildung 2.9 rechts zeigt entsprechend den RiLSA das Zusammenwirken der Modelle und deren Optimierung. Die Kenngrößen werden erfasst und die Verkehrsnachfrage prognostiziert. Auf Basis dieser Prognose werden dann Steuerungsentscheidungen mit Hilfe weiterer Modelle optimiert. Dies stellt eine mehrstufige MPR dar.

Die Vorteile dieser Methode sind insbesondere der intuitive Ansatz, die hohe Flexibilität bei der Modellbildung und der Gestaltung der Kostenfunktion sowie die Möglichkeit, variable Einschränkungen zu berücksichtigen. Verkehrsnetzwerke sind aus Sicht der Regelungstechnik ein ideales Anwendungsfeld, da sie mathematisch betrachtet stabil, da sie natürlich begrenzt sind, [Te08], insbesondere bei der Parametrierung mit makroskopischen Größen. Zudem können sie verteilt und hierarchisch umgesetzt werden. Einen Überblick über verteilte MPR geben [Ce13b], [Ce13a] und [NM14]. Die Autoren von [Sca09] geben einen Einblick in Architekturen für verteilte MPR. Die Regelung vernetzter Systeme wie Energieversorgungs- und Verkehrsnetzwerke werden aktuell in der Regelungstheorie erforscht. Es ist zu erwarten, dass diese Forschung in den nächsten Jahrzehnten neue LSA-Regelungen ermöglichen wird.

Die verwendeten Kenngrößen für modellbasierte LSA-Programme sind aufgrund der Designfreiheit weniger stark selektiert als bei den regelbasierten (RB)-Programmen und können sowohl über eine Signalprogrammanpassung als auch eine Signalprogrammabbildung umgesetzt werden. Beispielsweise können Bewertungsgrößen für die Qualität direkt in die Regelung einbezogen werden. Eine beispielhafte Übersicht über modellbasierte Verfahren findet sich in Tabelle A.1 in Anhang A, basierend auf den ausführlicheren Darstellungen der Verfahren in [Wei17] und [Ott11]. Eine umfassendere Darstellung ist in [Be11b] zu finden.

2.6 Forschungsprojekte mit Bezug zu Schaltzeitprognosen

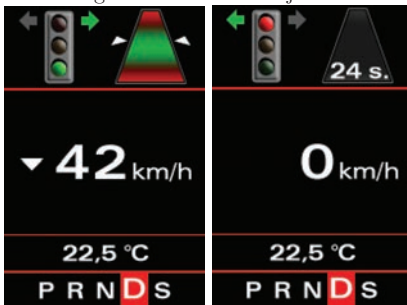
In diesem Abschnitt wird anhand der recherchierten Fachliteratur die Anwendung von SZP vorgestellt. Im Zentrum der Entwicklung von Infrastrukturprojekten, die dem Thema Smart City (Digitalisierung der Infrastruktur) gewidmet sind, steht die Kommunikationsfähigkeit der verkehrstechnischen Infrastruktur. Egal, ob die Kommunikation von Infrastruktur zu Infrastruktur (I2I) oder von Fahrzeugen mit der Infrastruktur (Vehicle to Infrastructure Communication, umgekehrt I2V (V2I)) oder untereinander (V2V) geschieht: Sobald Verkehrsnetzwerke im urbanen Raum mit einbezogen werden, spielt die Vorhersage von Verkehrszuständen eine zentrale Rolle. Die Schaltzeitprognose ist für diese Kommunikation mit den Beteiligten ein essenzieller Bestandteil, da LSA maßgeblich den Verkehrsfluss bestimmen [Ott11]. In Tabelle A.2 in Anhang A ist eine Auswahl von Forschungsprojekten aus Deutschland, die SZP mit betrachten, aufgeführt. Elaborierte Darstellungen der Projekte finden sich bereits in [Wei17, Pro16] und [Lü15] sowie [Se14].

Die Projekte zeigen eine Entwicklung von Kommunikationstechnologien für die Verkehrsinfrastruktur hin zur Entwicklung von Applikationen zur Fahrerassistenz, wie zum Beispiel in den Abbildungen 2.10 und 1.1. Während dieser Entwicklung wurden Bedarfe aufgezeigt.

Im Zuge der Entwicklung dieser Kommunikationstechnologien werden die Auswirkungen der zusätzlich verfügbaren Informationen auf Fahrer und LSA und damit auf den Verkehrsfluss und die Verkehrssicherheit untersucht. Positive Wirkungen konnten bereits in [Ott11] nachgewiesen werden. Der Autor von [Ott11] führt an,

- dass Betreffend der „Wolfsburger Welle“ mittels Grünbandinformation die Anzahl der Halte um 10 % bis 30 % reduziert und die Haltezeiten um 20 % bis 25 % und der Kraftstoffverbrauch um 0 % bis 5 % gesenkt wurden [Zim83].
- dass die Simulationsversuche in [Ric05] zur Aussage „Die Beeinflussung des Fahrverhaltens über Geschwindigkeitsempfehlungen führte zu bis zu 20 % höheren Sättigungsverkehrsstärken bei kurzen Grünzeiten und zu bis zu 8 % höheren Sättigungsverkehrsstärken bei langen Grünzeiten. Es konnte eine Kraftstoffersparnis von im Mittel 15 % und maximal 35 % nachgewiesen werden“ führte [Ott11] aus.

Weitere Ergebnisse sind unter anderem ein Leitfaden für die Einrichtung kooperativer Systeme auf öffentlicher Seite [Ke16] und verschiedenste Applikationen zur Fahrerbeeinflussung. Die Fülle an Projekten in diesem Bereich zeigt die Nachfrage an SZP auf.



Die wichtigsten direkten Anwendungen von SZP, welche aus Projekten resultieren, sind Fahrerinformationssysteme. Ziel von Fahrerinformationssystemen ist es, zum einen dem Fahrer zusätzliche Komfortfunktionen für das Bewältigen des urbanen Verkehrsgeschehens zu bieten, zum anderen wird durch diese Assistenzfunktionen das Fahrverhalten der direkt und indirekt informierten Verkehrsbeteiligten beeinflusst.

Abbildung 2.10: GLOSA, Projekt TRAVOLUTION [Me08]

Dies reduziert Staus und Emissionen [Ott11], siehe oben. Zu den Assistenzfunktionen gehören beispielsweise Geschwindigkeitsempfehlungen (engl. *Green Light Optimal Speed Advisory (GLOSA)*) bei Annäherung an LSA, Rest-Grün- und -Rot-Anzeigen sowie die Anzeige der virtuellen Position im Grünband und Streckenvorschläge über Smart Routing. Beim Smart Routing werden für einen Streckenvorschlag zusätzliche Informationen wie Antriebstyp, Steigungen und Verkehrsdichte sowie insbesondere SZP mit einbezogen. Beispiele für Schnittstellen zu Fahrern sind in den Abbildungen 2.10 aus dem Projekt TRAFic eVOLUTION (TRAVOLUTION) und 1.1 aus dem Projekt Adaptive und Kooperative Technologien für den Intelligenten Verkehr (AKTIV) zu sehen.

2.7 Literaturübersicht der Verfahren zur Schaltzeitprognose

Die nachfolgend besprochene Zusammenstellung von recherchierten Verfahren zur Schaltzeitprognose gibt einen Überblick über die bekannten Verfahren. Für eine Reihe von Verfahren finden sich tiefergehende Erläuterungen in [Wei17] und [Pro16].

Verkehrsunabhängige LSA-Steuerungen verhalten sich grundsätzlich deterministisch, und ihr Schaltverhalten müsste zu 100 % vorhersagbar sein. Es wäre zunächst naheliegend, deren Steuerungslogik für die Prognose anhand verkehrstechnischer Dokumentationen möglichst exakt nachzubilden. Wie auch aus der mikroskopischen Verkehrssimulation bekannt, führt dieser Weg angesichts oft lückenhafter, veralteter und fehlerhafter Unterlagen kaum zum Erfolg. Die Nachprogrammierung verkehrsunabhängiger Signalsteuerungen ist zudem sehr zeitaufwändig, fehleranfällig und nicht praktikabel [HHS04]. Ein Ausweg aus dieser Situation besteht darin, die Schaltzeiten auf der Grundlage des beobachteten Steuerungsverhaltens vorherzusagen.

2.7.1 Statistische Ansätze

Die frühesten Ansätze zur SZP beruhen auf statistischen Ansätzen. Der Übergang von rein statistischen Ansätzen zu Ansätzen mittels maschinellen Lernens ist fließend.

In [Me08] wird ein Konzept des „informierten Fahrers“ vorgestellt, der seine Fahrweise auf einen bevorstehenden, aber durch eine Beobachtung noch nicht erkennbaren Signalwechsel abstimmen kann. Das Prognoseverfahren basiert auf Markov-Ketten, bei denen aus Tupeln historischer Detektorbelegungen und Phasenübergänge eine Matrix von Übergangswahrscheinlichkeiten gebildet wird. Auf deren Grundlage wird dann die nachfolgende Phase und ihr Beginn vorhergesagt. Das Verfahren wurde an drei verkehrsunabhängigen LSA im Testfeld Ingolstadt erprobt. Über einen Horizont von 10 s bis 60 s konnten laut [Be09] „die meisten, wenn auch nicht alle Situationen“ sekundengenau vorhergesagt werden. Einen verbesserten Ansatz mit Markov-Ketten beschreibt [OW13]. Hier wurden zusätzlich die Abfolge vorheriger Phasen und Phasenübergänge sowie ausgewählte Parameter des zugrunde liegenden modellbasierten Steuerungsverfahrens herangezogen. Ein weiterer statistischer Ansatz wird in [MV12] vorgestellt. Dieser basiert ebenso auf der Auswertung historischer Signaltupel, aus deren Häufigkeitsverteilung die Grünwahrscheinlichkeiten mit einem Prognosehorizont von wenigen Sekunden abgeleitet wurden.

In [MV12] wird ein Ansatz vorgestellt, welcher ebenfalls auf der Auswertung historischer Signaltupel basiert. Hierzu werden Häufigkeitsverteilungen abgeleitet und daraus Grünwahrscheinlichkeiten berechnet. Der Horizont der Vorhersage beträgt wenige Sekunden. Weitere probabilistische Ansätze haben Zhao [Ze17] und Du [De19] veröffentlicht. Zhao nutzte ausschließlich Smartphone-Daten (Accelerometer- und Magnetometer) sowie GPS-Daten. Er entwickelte einen eigenen Ansatz zur probabilistischen Auswertung des Schalt-

verhaltens an Knotenpunkten mit Annahmen zu rechtwinklig zueinander stehenden Knotenarmen. Es wurde eine Standardabweichung von weniger als 5s für die Schaltzeiten erreicht. Du nutzte seinen statistischen Ansatz nicht nur für die Vorhersage von Schaltzeiten, sondern auch für die Umlaufzeit und Programmwechsel. Eine Anwendung an 1000 LSA in Jinan, China, ergab Genauigkeiten im Bereich von 2s bis 9s.

Koukoumidis analysierte die Prognostizierbarkeit von Festzeitprogrammen in den USA und das modellbasierte Steuerungsverfahren SCATS in Singapur [Ke11b,Ke11a]. Für die Vorhersage registrierte fahrzeugseitige Smartphones während der Fahrt die aktuellen Signalbilder und lieferten so die Informationen für eine Schaltzeitprognose. Der Horizont reicht bis zu vier Phasen in die Zukunft, was den längsten Prognosehorizont der hier aufgeführten Verfahren darstellt. Die Prognosegenauigkeit lag für die Festzeitprogramme bei 0,5s bis 1s und für SCATS bei 2s bis 4s. Es ist allerdings davon auszugehen, dass sich der Horizont bei einer Anwendung auf die für Deutschland üblichen komplexeren Steuerungen verkürzen und sich die Genauigkeit im Vergleich zu den Ergebnissen mit SCATS verschlechtern würde.

Mit dem Ziel der Reichweitensteigerung von Elektrofahrzeugen wurde ebenfalls ein Verfahren zur Schaltzeitprognose entwickelt und in [Ke14,Ue14] vorgestellt. Mittels Autokorrelationsanalyse der Signaldaten wurden sich zyklisch wiederholende Muster identifiziert und daraus Umlaufzeiten geschätzt sowie ein Basisvektor berechnet. Dieser enthält für jede Sekunde des Umlaufs die Wahrscheinlichkeit für eine Freigabe. Getestet wurde das Verfahren an 180 LSA in Dresden.

In [PWF14] wird ein Ansatz zur Verarbeitung von FCD mittels Kalman-Filter in Kombination mit der Auswertung von Häufigkeitsverteilungen beschrieben. Als Anwendung wurden sowohl die Ampelphasenassistenz als auch das Smart Routing adressiert und an etwa 600 LSA in München evaluiert. Der Kalman-Filter erfordert die Modellierung des konkreten Systemverhaltens im Sinne der Ankunftszeiten und die Abbildung von LSA-Steuerungen.

Protschky präsentiert zudem in seiner Dissertation [Pro16] und den zugehörigen Veröffentlichungen [Pro13,PWF14] unter anderem SZP mit einem Ansatz über einen Kalman-Filter und Auswertungen von Häufigkeitsverteilungen. Das Ziel war nicht nur ein GLOSA-System, sondern auch eine Optimierung von Routen (Smart Routing) zu ermöglichen. Als Datenbasis wurden Floating Car Data verwendet. Diese Assistenz und das Smart Routing wurden in München (hauptsächlich modellbasiert) an ca. 600 der ca. 1100 zur Verfügung stehenden LSA angewendet. Die eigens entwickelten Qualitätskennzahlen erreichten bis zu 71 % Verfügbarkeit bei einer Genauigkeit von 95 %, siehe auch Abschnitt 4.4. Notwendig für die Verwendung von Kalman-Filtern ist eine Modellierung des Systemverhaltens, wie Ankunftszeiten der Beteiligten und die Abbildung der LSA-Steuerung, ähnlich modellbasierten Steuerungsverfahren. Nur wenn ein solches Modell adaptiv gestaltet wurde, ist es möglich auf Veränderungen des Systemverhaltens einzugehen. Dies schränkt eine Übertragbarkeit des Verfahrens ein.

Ebenfalls auf FCD basieren zwei weitere Ansätze. In [BF14] wird ausgeführt, dass die Markov-Ketten-basierte Schaltzeitprognose an zwei LSA in Braunschweig bei einem Prognosehorizont von 40 s eine Vorhersagesicherheit von etwa 80 % erreicht. Ein zweiter Ansatz, beschrieben in [APF15, AF16], nutzt die FCD für eine Schätzung von Fahrzeugtrajektorien und leitet daraus unter anderem mittels statistischer Dichtefunktionen Schaltzeitprognosen ab. Bedingung für das Verfahren sind Festzeitsignalisierungen oder nur solche verkehrsabhängigen Signalprogramme, für die die Phasenfolge und Umlaufzeit fest sind und sich somit festgelegte Muster ergeben.

Phasenorientierte Ansätze werden in [Ie18] und [Bod17] gezeigt. Die verbleibenden Phasendauer wird in [Ie18] mittels bedingter Wahrscheinlichkeit prognostiziert. Die Wahrscheinlichkeiten werden in Abhängigkeit zur Umlaufsekunde und Uhrzeit gesetzt. Der betrachtete Knotenpunkt wird regelbasiert kontrolliert. Es wurden zwei auf Anforderung geregelte Signalgruppen betrachtet. Der zugehörige Datensatz umfasst 2.000 Umläufe. Neben den SG-Daten werden auch Detektordaten verwendet. Ein weiterer Ansatz von [Ie18] nutzt Verteilungsfunktionen der relativen Häufigkeiten der Phasenlängen.

In [Bod17] wird ein Graphenmodell zur SZP verwendet. Dieser Transitionsgraph (TG) wird durch eine Graphentransformation aus dem Phasenfolgeplan erzeugt, wobei die Knoten des TG Phasenübergänge und Phasen als gerichtete Kanten betrachtet werden, was invers zum Phasenfolgeplan ist. Die Transitionswahrscheinlichkeiten werden ebenfalls abhängig von der Uhrzeit und vom Wochentag ermittelt. Zusätzlich werden Detektorinformationen verwendet, um die wahrscheinlichste, die früheste und die späteste Zeit eines Phasenwechsels zu schätzen. Zur Evaluation wurden jeweils 7.000 Vorhersagen herangezogen. Die mittlere Prognoseabweichung des Verfahrens liegt bei einem Horizont bis zu 5 s bei ca. 1 s und bei 10 s bis 30 s bei ca. 5 s. Weitere Informationen über die untersuchten LSA wurden nicht beschrieben.

2.7.2 Ansätze des maschinellen Lernens

In den letzten Jahren vor Veröffentlichung dieser Arbeit wurden klassische statistische Ansätze von Methoden maschinellen Lernens in der Forschung zu SZP abgelöst. Der unter den gegebenen Rahmenbedingungen in Deutschland bisher mit vielversprechendste Ansatz findet sich in [Wei17]. Mithilfe von Support Vector Machines (SVM) wurde das Verhalten von regelbasierten LSA-Steuerungen (freie Signalprogrammabbildung und -anpassung) vorhergesagt. Der Horizont betrug einen Umlauf und war mit einer Wahrscheinlichkeit von über 90 % sekundengenau. Getestet wurde das Verfahren in Kassel und Düsseldorf. Aufgrund seiner Übertragbarkeit und Präzision findet das Verfahren über ein Startup weltweit in Fahrzeugen und Städten verschiedener Länder Anwendung. Der Prognosehorizont mit brauchbarer Genauigkeit ist jedoch auch hier für Smart-Routing-Anwendungen noch nicht ausreichend. Auch die Übertragbarkeit ist dadurch eingeschränkt, dass das Verfahren immer noch individuell an jede LSA angepasst werden muss.

Ein weiterer Ansatz stammt von Genser [Ge20]. Er sagt unter den Schaltzeiten die verbleibende Zeit bis Grün voraus. Erstmals werden hier verschiedene ML-Methoden mit-

einander verglichen: LSTM (Long Short Term Memory Neural Networks), RFS (Random Survival Forests) und ARIMA (AutoRegressive Integrated Moving Average). Sie werden einer linearen Regression gegenübergestellt. Das entsprechende Framework wird an einer nur schwach adaptiven Lichtsignalanlage, vgl. Abbildung 1, im Zentrum von Zürich getestet. Den Tests werden Daten von nur einem Tag zugrunde gelegt. Dabei übertraf das LSTM die RFS, indem es für alle Signalgruppen einen Fehler von unter 0,5 s prognostizierte. Die Anwendung auf andere Kreuzungen mit hochadaptiven Schaltzeiten steht noch aus. Es ist zu erwarten, dass der Fehler hierbei größer ausfallen wird.

2.7.3 Vergleich der Verfahren

Zusammenfassend lassen sich einige Gemeinsamkeiten der vorgestellten Verfahren feststellen. So liegen die Vorhersagegenauigkeiten, hier die Präzision der Vorhersage eines einzelnen Schaltzeitpunktes, im Bereich von ein bis zwei Sekunden über einen Horizont von einer Phase bis zu einem Umlauf (ca. 90 s). Große Unsicherheiten bestehen bei Prognosen über den Horizont von über einem Umlauf. Praktische Tests fanden für alle Verfahren in kleinerem Umfang statt. In [Ke14, Ue14] wurden ganze Streckenzüge evaluiert.

Konsens besteht in der Anwendbarkeit der Verfahren: Es sind „nur rudimentäre“ und „keine allgemeinen“ Prognoseverfahren für verkehrsadaptive LSA bekannt [Ue14]. Dies bestätigt auch die hier vorliegende Literaturrecherche von Januar 2022 nicht nur für verkehrsadaptive, sondern auch für regelbasierte LSA, da die vorgestellten Verfahren nur auf jeweils eine Art der LSA-Steuerung (modell- oder regelbasiert, phasen- oder signalgruppenorientiert) ausgelegt sind.

Die vorgestellten Verfahren sind nur auf spezifische LSA-Arten anwendbar, was eine differenziertere Einteilung von Signalprogrammen als in MB, RB und FZ bedingt. Die Vorhersage von MB-LSA ist, verglichen mit RB-LSA, zumeist vergleichsweise einfach, da diese bereits selbst eine modellbasierte Vorhersage des eigenen Systemverhaltens treffen, um die nächsten Signalbilder auszuwählen.

In der Regel wird eine Vorhersage für einen längeren Zeitraum getroffen, als die damit optimierten Regelungseingriffe andauern. Die Vorhersage des Signalbilds bei einer solchen Steuerung bedingt, dass die Information aus dem System ausgelesen werden kann. Aufgrund von Firmengeheimnissen steht diese Information oft nicht zur Verfügung.

Unterschiede in den Verfahren lassen sich dadurch begründen, dass sie für unterschiedliche Regionen mit heterogener Datengrundlage und Zielsetzung entworfen wurden. Unterschiede aufgrund des Landes lassen sich darin begründen, dass geltend, rahmengebende Richtlinien beziehungsweise Empfehlungen und ähnliches für LSA länderspezifisch sind. Die Zielsetzungen der Prognoseverfahren divergieren bezüglich des Anwendungsfalls wie Geschwindigkeitsempfehlung bei Annäherung, intelligentes Routing und Schätzung der Verkehrssituation. Entsprechend unterschiedliche Gütekriterien werden an die Verfahren angelegt, wie beispielsweise Eintrittswahrscheinlichkeiten.

In den meisten Veröffentlichungen wird beschrieben, dass für die Vorhersage von Schaltzeiten verkehrsabhängiger LSA ausschließlich (historische) Signalbildinformationen ohne Detektorinformationen genutzt werden. In Anbetracht der Verkehrsabhängigkeit der meisten Anlagen ist es sinnvoll, alle Informationen, welche die Steuerungen von LSA in ihre Entscheidungen einbeziehen, für SZP zu verwenden und gegebenenfalls auch diese vorherzusagen. Aktuelle und ausführliche Verkehrsinformationen werden nur sehr eingeschränkt verwendet, siehe hierzu [WH17].

Mitunter wird das Signalbild über eine Bilderkennung hergeleitet. Dies benötigt zusätzliche Rechenzeit für die Umwandlung der Bilderinformation in die Signalbildinformation und unterliegt Unsicherheiten.

Ein weiterer Unterschied in der Literatur besteht bezüglich der Verwendung von ausschließlich historischen Signalbildern, wie in [MV12] gezeigt. Hierbei werden Informationen, die von verkehrsabhängigen LSA zur Entscheidungsfindung verwendet werden, nicht in die SZP miteinbezogen. Im Gegensatz hierzu steht die Verwendung von aktuellen Daten wie in [WH17]. Die Verfügbarkeit aktueller (Detektor-)Daten wird dabei aufgrund der Latenzzeiten zwischen LSA-Steuerung und SZP als kritisch bewertet. Die Verbindung von Methoden und Fachwissen zu LSA-Steuerungsprogrammen ist oft nicht vorhanden [Pro16].

Nicht gefunden wurden Quellen, die eine Voranalyse des Systems durchführen. Es werden kein Vor- und Fachwissen und keine Dokumentationen oder Systemanalysen verwendet. Solches wären beispielsweise die RiLSA [FGS10] oder auch die Rekonstruktion von LSA-Verhalten in [Wei19a].

Eine Schwäche in den Verfahren bezüglich Markov-Ketten ist die Markov-Eigenschaft, da hierdurch nur eine Phase bis maximal ein Umlauf vorhersagbar sind. Die Markov-Eigenschaft setzt die Abhängigkeit des aktuellen Systemzustands nur vom direkt vorhergehenden voraus, was bei LSA nicht gegeben ist. Die zumeist verwendeten Regressionen wie zum Beispiel SVM bedürfen sich wiederholender Muster in der Datengrundlage. Dies reduziert die Prognosequalität mit steigender Programmkomplexität. Für die Verwendung von Kalman-Filtern ist eine Modellierung des Systemverhaltens ähnlich zu modellbasierten Steuerungsverfahren notwendig. Die Modellierung von Kalman-Filtern ist zeitaufwändig und statisch.

Es existieren weitere unveröffentlichte Verfahren zur SZP, zum Beispiel von privaten Unternehmen. Die bekanntesten basieren hierbei auf Ergebnissen des Forschungsprojekts UR:BAN [Ke16]. Insgesamt ist die Anzahl an Verfahren gering, verglichen mit den Forschungsprojekten, die solche SZP verwenden, siehe oben. Nach [Kri17] wird das aktuell fortschrittlichste Verfahren in [Wei17] vorgestellt, was die im Rahme dieser Dissertation durchgeführte Literaturrecherche bestätigt.

2.8 Anforderungen an Verfahren zur Schaltzeitprognose

Anforderungen an SZP sind nach [Pro13] und [WH17] unter anderem: die Rechtzeitigkeit (Echtzeitfähigkeit) der Prognoseinformation, flächendeckende Verfügbarkeit, Zuverlässigkeit der Vorhersage sowie ein angemessener Prognosehorizont. Auch unvorhersehbare oder besondere Ereignisse wie FG-Anmeldungen oder Großveranstaltungen können die Qualität der Vorhersage mindern, da SZP darauf in der Regel nur schwer zu parametrisieren sind.

Die entscheidenden Faktoren für die Qualität von SZP können in zwei Bereiche unterteilt werden: Zum einen die Vorhersage selbst, deren Qualität beispielsweise durch eine mangelhafte Parametrierung an die betrachtete LSA-Steuerung gemindert ist. Zum anderen ist die umgebende Infrastruktur ein Faktor, wobei insbesondere Latenzzeiten und Datenverluste die Echtzeitfähigkeit von SZP einschränken können.

2.8.1 Zuverlässigkeit und Korrektheit

Die Akzeptanz von Fahrerassistenzsystemen hängt stark von deren Zuverlässigkeit ab. Die Erwartung am Straßenverkehr beteiligter Personen besteht zumeist darin, dass die sichtbare Signalisierung synchron und korrekt durch das Fahrerassistenzsystem wiedergegeben wird.

Die Zuverlässigkeit der Vorhersage beinhaltet die Eintrittswahrscheinlichkeit und die Genauigkeit beziehungsweise Korrektheit. Für die meisten Anwendungen ist eine Eintrittswahrscheinlichkeit von über 80 % bis 90 % gefordert, bei einer Genauigkeit von 1 s und gegebenenfalls einer Toleranz von 2 s bis 3 s [WH17].

Den Beteiligten muss vermittelt werden, dass es sich um eine Prognose mit einer gewissen Unschärfe handelt, was im Gegensatz zu einer eigentlich erwarteten Korrektheit steht. Einen Ausweg bietet das Unterdrücken der Anzeige einige Sekunden vor dem Signalwechsel. Unabhängig davon sollten aber Fahrempfehlungen wie beispielsweise „Jetzt ausrollen lassen, es wird in Kürze Rot!“ rechtzeitig abgegeben werden, damit sie zum gewünschten Ergebnis führen, nämlich ein Bremsmanöver zu vermeiden.

Die Korrektheit der Fahrhinweise hat eine inhaltliche und zeitliche Dimension. Nur weitgehend korrekte und zum richtigen Zeitpunkt ausgegebene Informationen führen zu Akzeptanz dieser Informationen. Erst wenn das Signalprogramm einen Phasenübergang eingeleitet hat, stehen die Umschaltzeitpunkte der Signalgruppen fest. Die dann noch verbleibende Fahrzeit zur Haltlinie beziehungsweise die Distanz dorthin kann allerdings zu kurz für energiesparende Fahrhinweise (wie etwa zum Ausrollen) sein. Daher ist es sinnvoll, die bei einer größeren Entfernung zur LSA noch bestehende Unsicherheit der Vorhersage den Nutzenden zu kommunizieren.

2.8.2 Rechtzeitigkeit und Prognosehorizont

Ohne Rechtzeitigkeit (Echtzeitfähigkeit) der Prognoseinformation über den gewünschten Zeithorizont besteht kaum Akzeptanz der Empfehlungen. Die Zuverlässigkeit einer Schaltzeitprognose beinhaltet die Eintrittswahrscheinlichkeit und die Genauigkeit. Für die meisten Anwendungen ist nach [Wei17] eine Eintrittswahrscheinlichkeit von über 80 % beziehungsweise 90 % gefordert, bei einer zeitlichen Genauigkeit von 1 s und gegebenenfalls einer Toleranz von 2 s bis 3 s. Bei zeitnahen Signalwechslern sollte die Genauigkeit besser ausfallen als bei zeitferneren. Ein angemessener Prognosehorizont für Kurzzeitvorhersagen kann im Bereich von etwa 30 s liegen und für Langzeitvorhersagen beispielsweise bei Smart Routing-Anwendungen mehrere Umläufe beziehungsweise mehrere Minuten betragen [LPZG15]. Die Eintrittswahrscheinlichkeit, die Genauigkeit und der Prognosehorizont können von technischen Störungen beeinträchtigt werden. Zu diesen zählen beispielsweise fehlerhafte oder fehlende Detektordaten sowie hohe oder gar unerwartet schwankende Latenzzeiten bei der Datenübertragung und -verarbeitung. Eine Prognose verliert im ungünstigsten Fall die Weite im Horizont, die an Latenzzeit vorliegt. Auch zeitlich unvorhersehbare oder besondere Ereignisse, diese reichen von FG- und ÖV-Anmeldungen bis hin zu Großveranstaltungen mit verkehrsbehindernder oder verkehrserzeugender Wirkung, können die Qualität von Schaltzeitprognosen mindern.

2.8.3 Verfügbarkeit

Insbesondere für Smart-Routing-Anwendungen spielt die flächendeckende Verfügbarkeit von Schaltzeitprognosen eine entscheidende Rolle. Eine freigabezeitsensible Routenoptimierung benötigt die Freigabezeitpunkte möglichst vieler, am besten aller LSA im berücksichtigten Straßennetz. Hieraus ergeben sich Anforderungen hinsichtlich der Übertragbarkeit und Generalisierbarkeit von Schaltzeitprognoseverfahren, um eine Verfügbarkeit innerhalb eines definierten Gebiets (räumlich) gewährleisten zu können. Die Hard- und Softwarekomponenten für die Verkehrssteuerung unterscheiden sich in den Einsatzgebieten wie beispielsweise Kommunen deutlich, so dass jeweils Individuallösungen gefunden werden müssten. Ein universelles Verfahren sollte sich daher an die verschiedenen Arten von Signalprogrammen automatisch anpassen können, auch betreffend der zentralen oder lokalen Positionierung der Hardware.

2.9 Weitere Anmerkungen

Aus den in den vorangegangenen Abschnitten dieses Kapitels beschriebenen Standards für LSA lassen sich die folgenden Erkenntnisse zusammenfassen:

- Anwendungen zur Information von Verkehrsbeteiligten mit integrierter SZP können den Verkehr signifikant positiv beeinflussen. Durch die Erfüllung der Anforderungen aus Abschnitt 2.8 ist zu erwarten, dass eine Akzeptanz der SZP-Information entsteht.
- Es besteht ein Bedarf an SZP, jedoch ein Mangel an gebrauchstauglichen Verfahren.
- Als Datenquelle werden nur OCIT-Daten für diese Arbeit verwendet, weil auf eine zeitaufwändige und fehleranfällige Rekonstruktion von SG- und Detektions-Daten zum Beispiel aus Bilddaten verzichtet werden soll.
- Die Prognose steht im Zentrum der Arbeit, ohne die umgebende Hard- und Software für die praktische Umsetzung und Einbettung. Somit wird die Informationsübertragung nicht betrachtet. Das Themengebiet der Latenzzeiten und Echtzeitfähigkeit wird damit für diese Arbeit nicht betrachtet und als eigenständiges Thema im Rahmen der Netzwerkarchitektur und -implementierung angesehen, siehe auch Abschnitt 2.8.
- Es werden keine Programmwechsel vorhergesagt sondern nur das Verhalten innerhalb eines Programms auf der mikroskopischen Ebene.
- Aufgrund der Heterogenität der Steuerungsverfahren, welche zum Teil nur schwer zu kategorisieren sind, ist ein Konzept für die angestrebte Vorhersage – wenn überhaupt – nur unvollständig. Ein erweiterbares Baukastensystem wird deshalb als sinnvoll angesehen.
- Protschky [Pro17] fordert unter anderem vergleichbare Bewertungsmaßstäbe. An solche soll sich diese Arbeit annähern.
- Eine Erarbeitung von verschiedenen Datenmodellen und deren Vergleich sollte stattfinden.
- Aktuelle Daten benachbarter Knotenpunkt (KP) könnten SZP verbessern.
- Ein umfassender Methodenvergleich mit Methoden verschiedener mathematischer Ansätze hat noch nicht stattgefunden.
- Eine Phasenvorhersage mit anderen Methoden als Markov-Ketten scheint ebenfalls zu fehlen.

Kapitel 3

Einführung in das maschinelle Lernen

In diesem Kapitel werden Grundlagen der verwendeten Methoden des maschinellen Lernens (ML) erläutert. Maschinelles Lernen kann als Teilgebiet der Künstlichen Intelligenz (KI) angesehen werden. Zwar gibt es verschiedene Ansätze, was die Definition von Intelligenz und die damit verbundene Einordnung verschiedener Systeme in die KI anbelangt, trotzdem besteht die Gemeinsamkeit aller Systeme, die sich unter KI zusammenfassen lassen, darin, dass sie versuchen, (menschliche) Intelligenz nachzuahmen [Wit19].

Methoden des maschinellen Lernens dienen der Mustererkennung in Datensätzen. Bishop [Bis09] zufolge sind die Begriffe maschinelles Lernen und Mustererkennung Bezeichnungen für das gleiche Methodenspektrum, jedoch aus Sicht zweier Disziplinen: der Informatik und dem Ingenieurwesen. Alle Methoden haben eins gemeinsam: Sie nutzen mathematische Modellierungen in Form von Funktionen ($y(x)$) kombiniert mit mathematischen Optimierungsverfahren, um logische Zusammenhänge nachzubilden. Als Kostenfunktion für die Optimierungsverfahren werden zumeist Größen aus der Statistik, Entscheidungs- oder Informationstheorie herangezogen.

Für die Nachbildung von Systemverhalten müssen Informationen über das betrachtete System wie beispielsweise Entscheidungslogiken von LSA gesammelt werden. Dies geschieht allgemein, indem Daten zu den auf das System einwirkenden Größen in Form von Stichproben gesammelt werden, ähnlich wie es in der Statistik geschieht, beispielsweise Detektorbelegungen und FG-Anforderungen an LSA. Aus n einzelnen Stichproben (engl. *sample*) werden die Eingangsvektoren ($x = [x_1, \dots, x_i, \dots, x_n]$) für das Modell ($y(x)$) gebildet. Die Stichproben setzen sich aus m Merkmalen $x_i = [x_{i,1}, \dots, x_{i,k}, \dots, x_{i,m}]^T$ (engl. *features*), welche sich zum Beispiel mit der Zeit (k) verändern, zusammen. Entsprechend ist ein Datenmodell eine Matrix $X = [\dots, x_i, \dots]$, bestehend aus Merkmalen und Stichproben.

Zumeist können Informationen über das vom System ausgehende Verhalten gesammelt werden, bezüglich eines Knotenpunktes beispielsweise die Verkehrsbelastung oder die Zeitpunkte von GA oder GE. Solche Ausgangsgrößen können als Zielvariablen ($y = [y_1, \dots, y_k, \dots, y_m]^T$) (engl. *target/label*) des ML ausgewählt werden.

Im Folgenden werden die Grundlagen zu den wichtigsten in dieser Arbeit verwendeten ML-Methoden näher erläutert. Falls nicht anderweitig gekennzeichnet, sind die folgenden Informationen dieses Kapitels an [Bis09] angelehnt.

3.1 Arten von maschinellem Lernen

Maschinelles Lernen wird zumeist in drei Arten unterteilt: überwachtes Lernen (engl. *supervised learning*), unüberwachtes Lernen (engl. *unsupervised learning*) und bestärkendes Lernen (engl. *reinforcement learning*). Im Folgenden wird vor allem auf das überwachte Lernen detaillierter eingegangen, da dieses Verfahren für die spätere Umsetzung der SZP verwendet wird.

3.1.1 Überwachtes Lernen

Für das **überwachte Lernen** sind *Zielvariablen notwendig*. Zusammen mit den Eingangsdaten x und den Zielvariablen/Ausgangsdaten y werden die Parameter der mathematischen Funktion $y(x)$ angepasst. Dies wird im Kontext des ML als Lernen bezeichnet. Mittels der aktuell eingestellten Parameter und Eingangsdaten können Ausgangsdaten geschätzt werden (engl. *prediction \hat{y}*). Da die tatsächlichen Ausgangsgröße y bekannt ist, können die geschätzten und tatsächlichen Ausgangsgrößen über entsprechende Fehlermetriken miteinander verglichen werden. So können die Parameter ausgehend vom Ausgang optimiert werden. Als überwacht wird dieses Lernen bezeichnet, da eine „überwachende“ Referenz in Form der tatsächlichen Ausgänge verfügbar ist.

Diese Art des Lernens kann auf zwei Problemtypen angewendet werden: Klassifikation und Regression, siehe Abbildungen 3.1 und 3.2.

Bei der **Klassifikation**, Abbildung 3.1, gilt es, anhand der Eingangsgrößen eine Entscheidung über die *Zuordnung* der Stichprobe zu *einer Klasse* zu geben. Der Raum der Eingangsgrößen wird bei einer Klassifikation versucht, durch Funktionen in Bereiche entsprechend der Klassen einzuteilen.

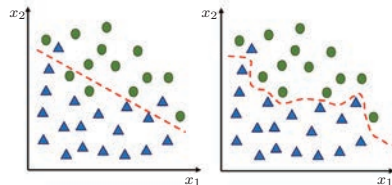


Abbildung 3.1: Veranschaulichung von Klassifikation; links: linear; rechts: nichtlinear

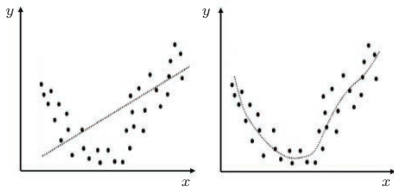


Abbildung 3.2: Veranschaulichung von Regression; links: linear; rechts: nichtlinear

Bei einer **Regression**, Abbildung 3.2, geht es darum, *kontinuierliche Prozesse abzubilden*, wie beispielsweise für Wettervorhersagen die Tageshöchsttemperatur. Eine einfache Sonderform der Regression ist die lineare Regression. Darin wird eine lineare Funktion $y(x) = ax + b$ definiert, deren Parameter a und b durch einen Algorithmus angepasst werden, bis sie optimal zwischen den Datenpunkten liegt.

Diese Methode ist stark durch die Vorannahme eingeschränkt, dass den Daten ein linearer Zusammenhang zugrunde liegt. Sobald die Daten nicht mehr linear voneinander abhängig sind, können lineare Regressoren die Verteilung der Datenpunkte nicht mehr annähern, siehe Abbildungen 3.1 und 3.2.

Bei einer linearen Regression wird die Annahme getroffen, dass der Zusammenhang zwischen Ein- und Ausgang linear ist. Gibt es eine ML-Methode, innerhalb derer mehrere Polynome getestet werden können, dann stellt der Grad p des Polynoms einen Parameter der Methode dar. Um solche Parameter von den Parametern der eigentlichen Funktion, wie oben a und b , von den Parametern, wie hier, p dieser Methode unterscheiden zu können, können die Methodenparameter als **Regularisierungsparameter** bezeichnet werden.

Während bei der Bewertung eines Regressors der Abstand eines vorhergesagten Wertes zum wahren Wert gemessen wird und die Zielgröße auf eine kontinuierliche Skala projiziert wird, zählt bei der Bewertung von Klassifikatoren, ob die Eingangsdaten der richtigen (diskreten) Klasse zugeordnet wurden. Die meisten Methoden des maschinellen Lernens können sowohl zur Lösung von Regressions- als auch Klassifikationsproblemen verwendet werden.

3.1.2 Unüberwachtes Lernen

Der Unterschied des unüberwachten Lernens (unsupervised learning) zum überwachten Lernen besteht darin, dass keine Zielgrößen bekannt sind. Das Ergebnis der Ausgabe ist offen und muss vom Algorithmus selbst erkannt werden, indem er Regelmäßigkeiten und Muster in den Daten erkennt. Dies nennt sich Clustering [Wit19]. Anders als bei der Klassifikation kann durch das unbekanntes Ergebnis der Vorhersage deren Güte nicht beurteilt werden. Die Definition der Ausgangsgrößen ist abhängig davon, welche Klassen als solche erkannt werden. Es kommt auf den Anwendungskontext an, ob die gewählten Klassen sinnvoll sind. Ein bekanntes Beispiel für unüberwachtes Lernen sind die Empfehlungen, die von Suchmaschinen zu Onlineshops gegeben werden, bei denen ein Algorithmus versucht, Muster im Suchverhalten zu erkennen und so passende Empfehlungen zu geben [Fro19]. Andersherum können im Rahmen der Ausreißererkennung auch Daten gesucht werden, die

aus einem Muster herausfallen, beispielsweise beim Auftreten von technischen Problemen oder Angriffen auf Netzinfrastrukturen, um diese zu erkennen und ihnen entgegenwirken zu können [Fro19]. Bezüglich LSA ist so eine Zuordnung der SG zu Gruppen wie ÖPNV, Rad, FG und mIV denkbar.

3.1.3 Bestärkendes Lernen

Bei der dritten Form des maschinellen Lernens, dem bestärkenden Lernen, geht es darum, in Form von Belohnungen und Bestrafungen dem Modell ein gewünschtes Verhalten anzutrainieren. Ein Agent als Instanz des Modells agiert mit seiner Umwelt und erhält je nachdem, ob oder wie sehr eine seiner Aktionen zielführend war, eine Rückmeldung. Dabei ist das Ziel bekannt, folglich kann beurteilt werden, wie hoch die Qualität des Ergebnisses ist. Somit können die Einzelaktionen, welche zu diesem Ergebnis geführt haben, bewertet und so verschiedene Strategien gelernt werden. Die Strategie zum Erreichen des optimalen Ergebnisses ist unbekannt [NZ21].

Bestärkendes Lernen wird beispielsweise in der Robotik verwendet. Es werden vom Agenten verschiedene Aktionen ausgeführt, woraufhin eine Rückmeldung erfolgt. Nach mehreren Iterationen kann das Programm somit eine Handlungssequenz entwickeln, welche Belohnungen maximiert und somit eine Strategie zur Lösung eines Problems darstellt [Alp19].

3.2 Grundsätzliche Verfahrensweisen

Die Vorgehensweise ist bei den meisten ML-Methoden gleich und betrifft insbesondere das Vorgehen, um von einer ML-Methode und einem Algorithmus zu einem parametrisierten Vorhersagemodell zu kommen. Auch Lernstrategien zur Parametrisierung der Modelle werden in diesem Abschnitt vorgestellt. Um dieses Lernen zu steuern, werden Regularisierungen verwendet, deren Einstellung automatisiert werden kann. Zudem wird die Kreuzvalidierung als Methode, um eine angemessene Genauigkeit der Modellanpassung zu gewährleisten, erläutert.

3.2.1 Vorgehen zur Erstellung eines ML-Modells

Das Vorgehen, um von einer ML-Methode ein parametrisiertes und regularisiertes Modell abzuleiten, kann in drei Schritte unterteilt werden, wie in Abbildung 3.3 zu sehen ist.



Abbildung 3.3: Vorgehen beim maschinellen Lernen

Der aufwändigste Schritt ist die **Datenaufbereitung**. Sie kann in ebenfalls drei Schritte eingeteilt werden, wie in Abbildung 3.4 zu sehen ist. Im ersten Schritt werden die Daten in die Software **eingeliesen und so strukturiert**, dass sie weiter verarbeitet werden können (engl. *parsing/parser*). Dies ist besonders aufwändig, wenn die benötigte Information beispielsweise in Fließtexten oder komplexen Protokollen vorliegt, für die zunächst eine Extraktionsmethode entwickelt werden muss.

Danach werden die Daten auf ihre Qualität hin geprüft und Fehler bereinigt. Zum einen können zur **Fehlerbereinigung** fehlerhafte Daten entfernt werden. Zum anderen können (daraus entstandene) Datenlücken beispielsweise mittels Interpolation, Approximation oder Schätzung ergänzt werden. Fehlerhafte Daten können beispielsweise Daten sein, die unter für die Modellierung unerwünschten Rahmenbedingungen entstanden sind, wie beispielsweise während gesonderter Betriebszustände von LSA.

Abschließend werden in der Datenaufbereitung Merkmale aus den **Daten extrahiert** (engl. *features extraction*) und zu einem Datenmodell **zusammengestellt** (engl. *feature selection*). In diesen Schritten kann auffallen, dass Daten für benötigte Merkmale fehlen oder ein weiterer Schritt für die Fehlerbereinigung benötigt wird.



Abbildung 3.4: Vorgehen bei der Datenaufbereitung

Für die weiteren Schritte werden die Daten des Datenmodells abschließend in einen **Trainings- und einen Testdatensatz** aufgeteilt. Üblicherweise werden die Stichproben hierfür randomisiert und im Verhältnis 2:1 geteilt. Nur für die Vorhersage von Zeitreihen sollte die chronologische Abfolge der Stichproben erhalten bleiben. Durch die Randomisierung wird der zeitliche Zusammenhang der Stichproben zerstört, welcher außer bei Zeitreihen nicht vorhanden sein sollte, um die Übertragbarkeit auf Daten anderer Zeitausschnitte zu erhalten. Die Unterteilung zwischen Trainings- und Testdatensatz dient der Evaluierung des Modells. Durch das zufällige Entnehmen von Daten aus dem Datensatz, auf die das Modell nicht parametrisiert wird, kann die Generalisierungsfähigkeit des Modells anhand von Daten aus dem selben Entstehungskontext verifiziert werden. Für einige Methoden, wie beispielsweise neuronale Netze, werden die Eingangsdaten zusätzlich **skaliert**, um die Merkmale auf ähnliche Skalenbereiche zu beschränken. Hierfür sind Standardisierungen oder Normierungen üblich.

Die Zusammenstellung geeigneter Merkmale zu einem **Datenmodell** stellt eine zentrale Aufgabe bei der Entwicklung von ML-Modellen dar und ist direkt mit der Leistungsfähigkeit eines Modells verbunden.

Mit den Trainingsdaten wird nun ein Modell der gewählten ML-Methode parametrisiert, was auch als **Training** oder Lernen bezeichnet wird. Hierfür sind die Regularisierungsparameter zuvor dem Algorithmus einzugeben. Die Ausgangsgrößen der Trainingsdaten dienen in diesem Schritt als Referenz, mit dessen Hilfe eine Fehlerfunktion berechnet werden kann. Die Parameter werden üblicherweise mittels Gradientenabstiegsverfahren oder Maximierung des Erwartungswerts, welche auf die Fehlerfunktion angewendet werden, optimiert. Wenn neue ML-Methoden evaluiert werden sollen, werden bereits öffentlich verfügbare Referenzdatenmodelle verwendet, für die die Datenaufbereitung entfällt.

Für die **Evaluation** des optimierten Modells werden die Eingangsdaten des Testdatensatzes in das Modell eingegeben und sein Ausgang mit den Ausgangsdaten des Testdatensatzes mit Hilfe von Evaluationsmetriken verglichen. Die Fehlerfunktion des Trainings und die Evaluationsmetrik sollten ähnlich, wenn nicht gar gleich sein, um ein zielorientiertes Training zu gewährleisten.

Sind die Ergebnisse der Evaluation zufriedenstellend, liegt ein fertig trainiertes **ML-Modell** vor, welches nun in Betrieb genommen werden kann. Ist das Ergebnis nicht zufriedenstellend, wird die gleiche oder eine andere Methode mit gegebenenfalls anderen Einstellungen der Regularisierungsparameter erneut trainiert. Ist auch dies nicht zielführend, sind zumindest Teile der Datenaufbereitung zu wiederholen. Es ist üblich, die Merkmalsgenerierung und -selektion mehrfach zu wiederholen, bis eine ausreichende Qualität der Ausgangsgrößen eines ML-Modells erreicht wird.

3.2.2 Lernstrategien

Für die Durchführung der Modellparametrierung (Training/Lernen) gibt es mehrere Lernstrategien. Zum einen kann das Training, wie oben dargestellt, vor der Anwendung des Modells erfolgen, also „offline“. Das Trainingsdatenmodell wird als Ganzes der ML-Methode zum Training übergeben und die Parameter entsprechend angepasst.

Bezüglich des Offline-Lernens wird das einmalige Durchgehen des Datensatzes als Epoche bezeichnet. Insbesondere für neuronale Netze ist es üblich, über mehrere Epochen zu trainieren.

Zum anderen können die Parameter des Modells auch während seiner Verwendung weiterhin an die aktuelle Situation angepasst werden. Dies wird als „Online-Lernen“ oder inkrementelles Lernen bezeichnet. Bei dieser Form des Trainings werden die Parameter, sobald neue Stichproben vorliegen, angepasst.

Wenn ein Merkmal für eine solche Stichprobe nicht ermittelt werden konnte und der Algorithmus die ihn bedienende Person oder einen anderen Algorithmus auffordern kann, ihm dieses zu benennen, wird von „aktivem Lernen“ gesprochen. Beispielsweise kann bei Objekterkennung die Erkennung des Objektes nochmals erfragt werden: „Ist dies ein Signalgeber, ja oder nein?“

Beide Strategien (online und offline) können in Abwandlungen angewandt werden. Beispielsweise indem die Eingangsdaten weder stichprobenweise noch in ganzen Epochen, sondern gruppiert und anhand ihrer gemeinsamen Bewertung der Parameter gelernt werden. Im Englischen wird dies als *batch-wise learning* bezeichnet.

3.2.3 Regularisierung

Gemeinsames Ziel der meisten Methoden ist es, Zusammenhänge in den Daten möglichst exakt nachzubilden, dabei jedoch eine ausreichende Generalisierung zu bieten, um auch für Merkmale außerhalb der Menge der Stichproben sinnvolle Ergebnisse ausgeben zu können. Dies kann auch als Robustheit des Modells bezeichnet werden und wird mittels Regularisierungsparametern versucht zu steuern. Um diese Generalisierung zu erreichen, muss eine ausreichende Datenmenge verwendet werden, damit mit dem Training der Rückschluss von der Stichprobe auf das allgemeine Verhalten des betrachteten Systems gelingen kann.

Wenn dies nicht gelingt, besteht eine Über- oder Unteranpassung (engl. *under/overfitting*). Abbildung 3.5 veranschaulicht dies für eine Regression und Abbildung 3.6 für eine Klassifikation. Die zugrunde liegende Funktion in Grün ist $\sin(2\pi x)$. Die Stichprobe ist mit einem normalverteilten Rauschen versehen: blaue Kreise. Die Regression wurde mit Polynomen unterschiedlichen Grades M durchgeführt.

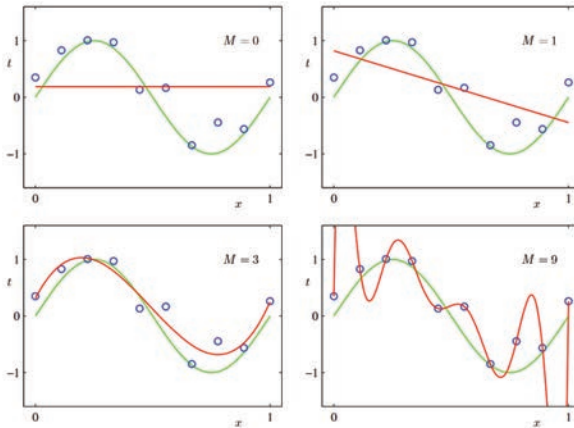


Abbildung 3.5: Über- und Unteranpassung [Bis09], M ist der Polynomgrad des Polynoms in Rot, Grün ist die Referenzfunktion $t = \sin(2\pi x)$.

Eine Unteranpassung bezeichnet die nicht ausreichende Anpassung eines Modells an Trainingsdaten, vgl. Abbildung 3.5 obere Zeile. In der Abbildung ist dies durch eine ungünstige Hyperparameterwahl (Polynomgrad) veranschaulicht. Dies kann unter anderem durch den Mangel an Daten oder eine nicht repräsentative Stichprobe hervorgerufen werden, welche einen Fehlschluss auf das Verhalten des zugrunde liegenden Systems impliziert. Die Ergebnisse für Training und Test sind ähnlich unzureichend, siehe Abbildung 3.7 $M = 0$ bis $M = 2$.

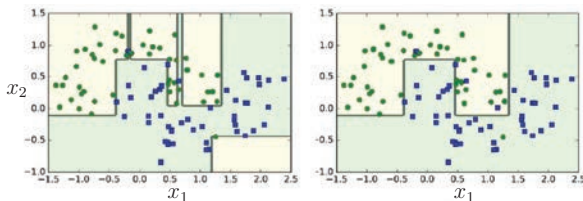


Abbildung 3.6: Regularisierung von Entscheidungsbäumen (links ohne, rechts mit Regularisierung)

Bei einer Überanpassung (Abbildung 3.5 rechts unten) besteht eine übermäßige Anpassung des Algorithmus an die Trainingsdaten. Mit der Anpassung werden zwar (fast) alle Trainingsdaten optimal abgebildet, jedoch das abstrakte Systemverhalten nicht abgebildet (Sinus-Funktion). Dies kann zwei Ursachen haben: Zum einen können nicht genügend Daten vorliegen, vgl. Abbildung 3.8, oder die ML-Methode kann so parametrisiert sein, dass die vorhandenen Daten (für Klassifikationen) auswendig gelernt werden können. Dies geschieht beispielsweise bei der Verwendung von sehr großen MLP als Basis für Deep Learning und gleichzeitiger Nutzung großer Trainingsdauern. Die Prognose von Testdaten verschlechtert sich, verglichen mit den Ergebnissen beim Training deutlich, da die Trainingsdaten zuvor nicht bekannt waren, vgl. Abbildung 3.8 für $M = 9$.

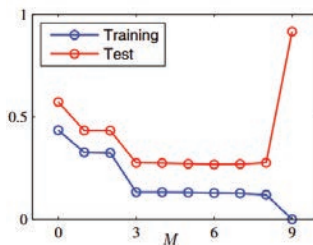


Abbildung 3.7: RMSE für das Beispiel aus Abbildung 3.5 [Bis09], Polynom in Rot mit Grad M , grün ist die Referenzfunktion $\sin(2\pi x)$

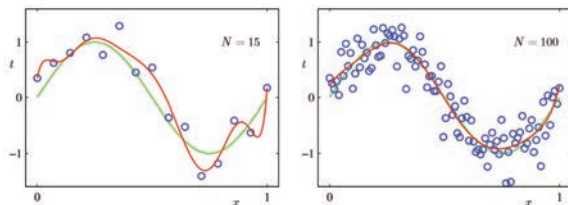


Abbildung 3.8: Trainingserfolg in Abhängigkeit von der Stichprobengröße N (blau [Bis09], mit einem Polynom 9. Grades in Rot, grün ist die Referenzfunktion $\sin(2\pi x)$)

Die Einschränkung der Freiheitsgrade eines Modells durch die Festlegung von Parametern wird Regularisierung genannt. Dadurch kann das Risiko der Überanpassung verringert und die allgemeine Performance eines Modells verbessert werden. Um ein Underfitting zu vermeiden, ist es hingegen hilfreich, Regularisierungen zu verringern oder das Modell mit

mehr Daten zu versorgen. Unter Regularisierungsparametern werden in dieser Arbeit alle Parameter eines ML-Algorithmus verstanden, welche nicht automatisiert beim Training bestimmt werden.

Die Begriffe Regularisierungs- und Hyperparameter werden in der Literatur unterschiedlich verwendet. Teilweise werden mit Regularisierungsparametern nur Parameter des mathematischen Modells gemeint oder die Parameter des Trainingsalgorithmus, jedoch nicht die Parameter des eigentlichen Modells, die vom Trainingsalgorithmus eingestellt werden, wie beispielsweise a und b einer linearen Funktion. In dieser Arbeit werden diese beiden Begriffe synonym verwendet und bezeichnen alle Parameter, die die Implementierung eines ML-Algorithmus für Modell, Training und Weiteres zur Verfügung stellt, um ein fertig trainiertes und evaluiertes Modell ausgeben zu können, außer den Parametern des eigentlichen Modells.

3.2.4 Kreuzvalidierung

Bei der Evaluierung eines Modells kann teilweise beobachtet werden, dass dieses nur eine hohe Performanz zeigt, weil die Testdaten besonders günstige Datenpunkte beinhalten beziehungsweise dass nicht genügend Trainingsdaten vorhanden sind, um das Modell ausreichend lernen zu lassen. Für eine zuverlässige Einschätzung der Performanz kann eine Kreuzvalidierung (*engl. Cross Validation*) (CV) durchgeführt werden. Hierzu wird der Trainingsdatensatz in k Gruppen (*engl. k -folds*) unterteilt, mit $k-1$ dieser Gruppen (blau) wird trainiert, wie das Beispiel mit $k = 5$ in Abbildung 3.9 zeigt. Mit der verbleibenden

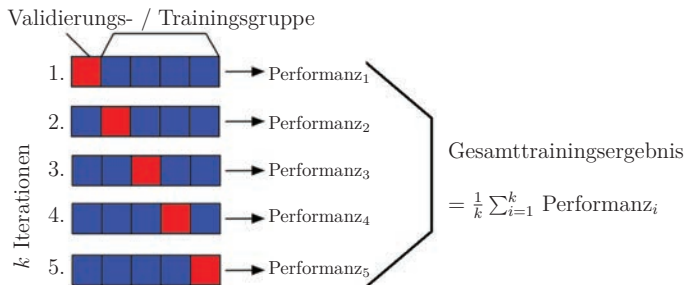


Abbildung 3.9: Kreuzvalidierung

Gruppe (rot) wird getestet. Dies ist eine Validierung innerhalb des Trainingsdatensatzes. Dies wird für alle Kombinationen der Gruppen wiederholt, bis jede Gruppe einmal für die Validierung verwendet wurde ist. Das Ergebnis der Kreuzvalidierung ist der Mittelwert der Einzelergebnisse (Performanzen) der k Durchläufe.

Dies wird ausschließlich mit den Trainingsdaten getan. Der Testdatensatz bleibt hiervon unberührt, ansonsten würde ein Informationsleck bestehen. Solch ein Leck liegt vor, wenn Informationen aus den Testdaten in die Trainingsdaten gelangen, womit die Testdaten dem Modell somit bereits bekannt sind, was die Evaluierung des Modells verfälscht. Die Generalisierungsfähigkeit kann so nicht mehr festgestellt werden.

3.2.5 Automatisierte Regularisierung

Zum Vorgehen wurde in Abschnitt 3.2.1 beschrieben, dass die Regularisierungsparameter der Modelle manuell gewählt werden. Es gibt drei Methoden, die dies automatisieren: Raster-Suche (engl. *grid search*), die probabilistische Bayes'sche Suche (engl. *Bayes-search*) und die randomisierte Suche (engl. *random search*). Abbildung 3.10 veranschaulicht Raster- und randomisierte Suche.

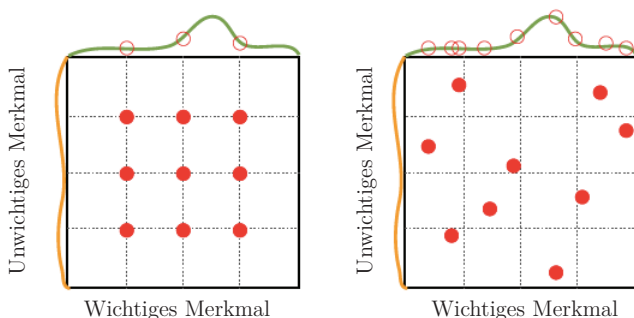


Abbildung 3.10: Raster- (links) und randomisierte Suche (rechts)

Alle drei Methoden trainieren ein Modell mit verschiedenen Regularisierungsparametern und geben die Regularisierungsparameter, die die geringsten Fehler aufweisen, zurück. Was sie unterscheidet, ist die ihnen zugrunde liegende Strategie (engl. *layout*) zur Auswahl der Regularisierungsparameter.

Bei der Raster-Suche, werden für jeden Regularisierungsparameter Werte fest gelegt, welche in allen möglichen Kombinationen getestet werden. Zumeist werden für die Parameter die Werte äquidistant im Parameterraum verteilt. Die Methode wird Raster-Suche benannt, da durch die Kombinationen ein Raster im Regularisierungsparameterraum vorgegeben wird. Da jeder Punkt des Rasteres getestet wird, ist diese Methode verglichen mit den anderen beiden Methoden rechenaufwändig.

Bei der Bayes'schen Suche wird statt einzelner Werte für jeden Regularisierungsparameter ein Suchbereich definiert. Innerhalb dieses Suchraumes wählt der Algorithmus, basierend auf bedingten Wahrscheinlichkeiten der bisherigen Ergebnisse seiner Suche, weitere Testpunkte. Die Initialisierung erfolgt zufällig mit mehreren Punkten. Dies spart meistens Rechenzeit gegenüber der Raster-Suche und folgt trotzdem einer logischen Wahl der Punkte im Suchraum.

Bei der randomisierten Suche (engl. *random-search*) werden im Suchraum zufällige Punkte getestet. Wie Abbildung 3.10 zeigt, kann die randomisierte Suche, wenn es sich um ungleich wichtige Parameter handelt, bessere Ergebnisse als die Raster-Suche zeigen.

3.3 Automatisierte Merkmalsselektion

Die Auswahl an Merkmalen eines Datenmodells für ein ML-Modell ist spezifisch anzupassen. Die Einflussfaktoren eines Systems haben ein unterschiedlich große Wichtigkeit bezüglich des Systemverhaltens. Entsprechend haben die Merkmale eines Datenmodells, als Repräsentanten dieser Einflüsse, für ML-Modelle ebenfalls einen unterschiedlich große Wichtigkeit (engl. *importance*) für die Vorhersage.

Die Merkmalsselektion verfolgt zwei Ziele: Zum einen kann über die Verwendung von weniger Merkmalen Rechenzeit gespart werden (engl. *dimension reduction*). Beispielsweise steigt die Rechenzeit bei eXtreme Gradient Boosting (s.u.) exponentiell mit der Anzahl der Merkmale.

In diesem Zusammenhang existiert zudem das „NoFree-Lunch“-Theorem, wonach es bei der Betrachtung aller möglichen Datensätze keine Methode geben kann, die in einem gegebenen Bewertungsrahmen für alle Datensätze die höchsten Bewertungen erzielt. Ist eine Methode des maschinellen Lernens für eine Teilmenge aller möglichen Datensätze überlegen, so ist sie in einem anderen unterlegen [Alp19]. Die optimale ML-Methode ist folglich nicht nur vom betrachteten System, sondern auch vom verwendeten Datenmodell abhängig.

Es gibt verschiedene Maße für die Wichtigkeit von Merkmalen unter anderem aus den Disziplinen der Statistik und Informationstheorie. Diesen ist gemeinsam, dass auf Basis eines Datenmodells Größen zu einzelnen Merkmalen und ihren Zusammenhängen ermittelt werden. Anhand dieser Größen werden die wichtigsten Merkmale behalten, während alle anderen aus dem Datenmodell verworfen werden. Der Zusammenhang zwischen Information und Merkmal wird in [LS94, KJ96] empirisch aufgezeigt. Wichtig ist, dass möglichst nur die Merkmale behalten werden, die wichtig für eine korrekte Vorhersage sind.

3.3.1 Statistische Einflussanalyse

Aus der Statistik werden die klassischen Einflussanalysen verwendet. Im folgenden werden die statistischen Berechnungen der Merkmalswichtigkeiten näher erläutert.

Bei einer Korrelationsanalyse wird die Kreuzkorrelation der Zeitreihe aus der Signaltechnik berechnet. Sie wird im ML häufig als Faltung bezeichnet. Die zugrunde liegende Formel für die Berechnung der Kreuzkorrelation mit dem Faltungsoperator $*$ ist:

$$c_{x_i,y}[t] = \sum_k x_i(k+t) * \bar{y}[k]. \quad (3.1)$$

Die Variable t ist hier eine Zeitdifferenz für die Zeit zwischen Zeitpunkt k und $k+t$. $\bar{y}[k]$ ist die elementweise Komplexkonjugierte von y . Eine speziell abgewandelte Form der korrelationsbasierten Merkmalsselektion für maschinelles Lernen stellt die Korrelationsbasierte Merkmalswahl (engl. *Correlation-based Feature Selection*) (CFS) dar [Ze10]. Bei nicht linearen Zusammenhängen ist die Aussagekraft dieser Größen eingeschränkt.

Für den χ^2 -Test wird zunächst für jedes Merkmal x_i sein Wert innerhalb der χ^2 -Verteilung

$$\chi_c^2 = \sum_j^n \frac{(O_{i,j} - E(x_{i,j}))^2}{E(x_{i,j})} \quad (3.2)$$

berechnet. $O_{i,j}$ ist die Häufigkeit der Klasse j innerhalb des Merkmals i aller Stichprobenklassen n . c bezeichnet den Freiheitsgrad, das heißt die Anzahl der Klassen $n - 1$. $E(x_{i,j})$ ist der Erwartungswert der Klasse des Merkmals. Die Werte der Merkmale werden anhand von Tabellen für die χ^2 -Verteilung miteinander verglichen, um zu prüfen, ob die Merkmale signifikant voneinander abhängig sind. Die Zielgrößen stellen die zentralen Vergleichsgrößen dar.

Des Weiteren können Merkmale über einen Permutationstest über die Permutationswichtigkeit selektiert werden. Für diesen Test wird ein trainiertes Modell verwendet. Für jedes Merkmal wird ein Test durchgeführt. Vor dem Test wird das betrachtete Merkmal permutiert. Durch diese Veränderung der Reihenfolge und damit das Zerstören des Kontexts zwischen diesem und den anderen Merkmalen sowie insbesondere der Zielgröße kann sein Einfluss ermittelt werden. Wenn das Ergebnis schlechter wird verglichen mit der eigentlichen Zuordnung innerhalb des Merkmals, ist es für die Vorhersage von Bedeutung.

3.3.2 Informationsbasierte Einflussanalyse

Aus der Informationstheorie können weitere Größen für die Merkmalswichtigkeit ermittelt werden. Um den mittleren Informationsgehalt zwischen zwei Zufallsgrößen zu bestimmen, wurde die Größe der Transinformation $I(x_i; y)$ (engl. *mutual information*), auch gegenseitige Information, Syntropie genannt, entwickelt. Sie basiert auf der (bedingten) Entropie $H(x_i)$, begründet auf der Hypothese, dass mit dem Informationsgehalt eines Merkmals seine Entropie steigt.

Die Entropie $H(x_i)$ kann als Erwartungswert des Informationsgehalts interpretiert werden [Sha48] und ist definiert als

$$H(x_i) = - \sum_{x_{i,k} \in x_i} p_{x_i}(x_{i,k}) \log(p_{x_i}(x_{i,k})). \quad (3.3)$$

Für diesen und die folgenden Logarithmen gilt Basis 2 oder e, was frei gewählt werden kann.

Die Wahrscheinlichkeit für das Auftreten von $x_{i,k}$ bezüglich ihrer Wahrscheinlichkeitsverteilung p_{x_i} ist $p_{x_i}(x_{i,k})$. Die bedingte Entropie $H(x_i|y = y_k)$ ist entsprechend definiert als

$$H(x_i|y = y_k) = - \sum_{x_{i,k} \in x_i} p(x_{i,k}|y = y_k) \log(p(x_{i,k}|y = y_k)). \quad (3.4)$$

Die Wahrscheinlichkeit für das Auftreten von $x_{i,k}$ unter Bedingung von $y = y_k$ ist $p(x_{i,k}|y = y_k)$. Entsprechend lässt sich die Entropie bezüglich der Zielgröße als gewichtetes Mittel mit

$$H(x_i|y) = \sum_{y_k \in y} p_y(y_k) H(x_i|y = y_k) \quad (3.5)$$

berechnen.

Die Transinformation $I(x_i, y)$ ist als weitere Analysegröße definiert als

$$I(x_i, y) = \sum_{x_{i,k} \in x_i} \sum_{y_k \in y} p_{x_i, y}(x_{i,k}, y_k) \cdot \log \left(\frac{p_{x_i, y}(x_{i,k}, y_k)}{p_{x_i}(x_{i,k}) p_y(y_k)} \right) \quad (3.6)$$

mit $p_{x_i, y}(x_{i,k}, y_k)$ als Wahrscheinlichkeit für das gemeinsame Auftreten von $x_{i,k}$ und y_k bezüglich ihrer gemeinsamen Wahrscheinlichkeitsverteilung $p_{x_i, y}$ [De 91].

3.3.3 Weitere Einflussanalysen

Neben den oben genannten Methoden, welche der Statistik oder Informationstheorie zugeordnet werden können, können Methoden aus anderen Gebieten der Mathematik verwendet werden: So kann die pragmatische Idee, alle Merkmalskombinationen innerhalb des Datenmodells auszutesten, verwendet werden: Rekursive Merkmalselimination (*engl. Recursive Feature Elimination*) (RFE). Basierend auf einer ML-Methode und einem Datenmodell wird das ML-Modell jeweils unter Weglassung eines Merkmals trainiert. Wenn das Modell schlechter evaluiert wird als das ML-Modell mit dem vollständigen Datensatz, wird das Merkmal im Datensatz behalten, wenn nicht, wird es verworfen.

Mit RELIEFF werden Merkmale und Zielgrößen miteinander verglichen, indem die (Manhattan-)Distanz innerhalb der Stichprobe ermittelt wird. Abhängig von der Distanz wird die Wichtigkeit bei kleinen Werten nach oben korrigiert und umgekehrt, siehe [KR92,Ke97]. Die Manhattan-Distanz beschreibt den Abstand von zwei Punkten über die Summe ihrer absoluten Differenzen.

Eine Methode, die nicht einzelne Merkmale aus dem Datensatz entfernt, sondern aus dem gesamten Datensatz Information extrahiert und in neuen Merkmalen hinterlegt, ist die Hauptkomponentenanalyse (*engl. Principal Component Analysis*) (PCA). Hierbei wird ein Ellipsoid um die Stichproben im Merkmalsraum des Datensatzes gespannt und seine Radien werden bestimmt. Anhand dieser Radien wird eine Achsentransformation vorgenommen, auf welche die Stichproben umgerechnet werden. Hierbei reduziert sich die Dimension des Datenmodells durch die Projektion in einen Unterraum. Die Radien werden unter anderem über die Varianzen der Merkmale ermittelt. Näheres hierzu ist in [PL01,Hot33,Hot92] zu finden.

3.4 Evaluationsmetriken

Für die Bewertung von ML-Algorithmen, welche mittels überwachten Lernens arbeiten, existiert eine Reihe von statistischen Größen. Zunächst werden die Metriken für Klassifikatoren und Regressoren in den Abschnitten 3.4.1 und 3.4.2 benannt. Es gibt eine spezielle Metrik für SZP. In Abschnitt 3.4.3 werden die zugehörigen Formelzeichen eingeführt. Eine Herleitung und Definition dieser Größen ist aus rechtlichen Gründen hier nicht möglich. Die Begriffe Evaluationsmetrik und Fehlerfunktion benennen identische Maße mit unterschiedlichem Verwendungszweck. Wenn nicht anders gekennzeichnet sind die hier dargestellten Grundlagen in [Bis09] und [Pe11] zu finden.

3.4.1 Metriken zur Klassifikation

Die Metriken für Klassifikationen basieren auf binären Klassifikationsmaßen (Null- und Alternativhypothese). Die erweiterte Konfusionsmatrix (Tabelle 3.1) zeigt beispielhaft für zwei Klassen a und b die wichtigsten dieser Maße mit ihren deutschen, englischen und mathematischen Bezeichnungen.

Links oben ist die Konfusionsmatrix zu sehen. Die zugehörigen Begriffe sind fett gedruckt. Alle weiteren Elemente der Tabelle beschreiben daraus berechenbare Größen. Die tatsächliche, reale Klassenzugehörigkeit (real) ist der prädizierten (pred.) Klasse A und B gegenübergestellt. Positiv (pos.) respektive negativ (neg.) bedeuten: zur Klasse respektive nicht zur Klasse gehörig.

Die zugehörigen Mengen der Konfusionsmatrix sind in Abbildung 3.11 zusätzlich veranschaulicht. Die Symbole r und f stehen für richtig/right und falsch/false. Die Indizes n und p für negativ und positiv. Die im Machine-Learning verbreitetsten Bewertungsgrößen sind die Korrektklassifikationsrate *engl. accuracy* (KKR), Relevanz, Sensitivität und f_1 .

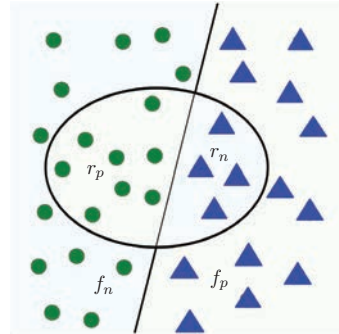


Abbildung 3.11: Mengen der Konfusionsmatrix, basierend auf der Grundgesamtheit; die Klassen sind: Kreise: positiv, Dreiecke: negativ

Das klassische zentrale Maß für Klassifikationen ist die Korrektklassifikationsrate (KKR) (en. accuracy)

$$K = P(\text{richtig}) = \frac{r_n + r_p}{r_n + f_n + r_p + f_p}. \quad (3.7)$$

Für diese Arbeit wird sie entsprechend verwendet.

Ein weiteres Maß für die Qualität einer Vorhersage ist die Verlustfunktion der Kreuzentropie $L_{\log}(y)$ (engl. *cross entropy loss*), ein Maß der Informationstheorie, vgl. 3.3.2. Sie wird für den Vergleich zwischen zwei Wahrscheinlichkeitsverteilungen beziehungsweise Stichprobenmengen verwendet. Die hier verwendete Formulierung ist

$$L_{\log}(y) = - \sum_{i=0}^N \sum_{j=0}^J \frac{1}{N} \log(p(y_{i,j} = 1) + (1 - y_{i,j}) \log(1 - p(y_{i,j} = 1))) | y_{i,k} \in \{0, 1\}. \quad (3.8)$$

K ist hier die Anzahl der Klassen und N die Anzahl der Stichproben [Pe11].

Der Gini-Index G_j wird für die Regularisierung von Entscheidungsbäumen verwendet, siehe Abschnitt 3.5.3. Er wird mithilfe des prozentualen Vorkommens $p_{l,j}$ einer Klasse $l \in L$ für den an Knoten j noch zutreffenden Anteil der Stichprobe ermittelt. Die mathematische Beschreibung des Gini-Indexes ist

$$G_j = 1 - \sum_{l=1}^L (p_{l,j}(1 - p_{l,j})). \quad (3.9)$$

Tabelle 3.1: Konfusionsmatrix mit zugehörigen statistischen Maßen (Zush. = Zusammenhang, en. = englisch)

Grundgesamtheit	real pos. (a)	real neg. (b)	mathem. Zush. real pos.	mathem. Zush. real neg.
pred. pos. (A)	richtig pos. (r_p)	falsch pos. (f_p); Fehler I. Art	Relevanz (en. precision) $P = P(a A) = \frac{r_p}{r_p+f_p}$	Falsche Annahme (I. Art, false discovery rate) $P(b A) = \frac{f_p}{r_p+f_p}$
pred. neg. (B)	falsch neg. (f_n); Fehler II. Art	richtig neg. (r_n)	Falsche Ablehnung (II. Art, en. false omission rate) $P(a B) = \frac{f_n}{r_n+f_n}$	Trennfähigkeit (en. negative predictive value) $P(b B) = \frac{r_n}{r_n+f_n}$
mathem. Zush. pred. pos.	Sensitivität (en. recall) $R = P(A a) = \frac{r_p}{r_p+f_n}$	Spezifität (en. specificity) $P(B b) = \frac{r_n}{r_n+f_p}$	KKR, Gl. 3.7	
mathem. Zush. pred. neg.	Falsch-Negativ-Rate (en. miss rate) $P(B a) = \frac{f_n}{r_p+f_n}$	Falsch-Positiv-Rate (en. fall-out) $P(A b) = \frac{f_p}{r_n+f_p}$		f_i -Maß (en. f_i -score, weighted precision recall) $f_i = \frac{(1+i^2)PR}{(1+i^2)r_p + i^2(P+R)} =$

3.4.2 Metriken zur Regression

Zentral für die Metriken für Regressoren ist die Frage: Wie weit ist die Vorhersage (im Mittel) von der Realität entfernt? Die Anzahl der Stichproben ist hier jeweils n . Klassische Maße sind der absolute Fehler als mittlere absolute Abweichung (engl. *Mean Average Error*) (MAE) zwischen Vorhersage und Realität

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i|. \quad (3.10)$$

Statt des Betrages kann auch das Quadrat der Abweichung verwendet werden, der mittlere quadratische Fehler (engl. *Mean Squared Error*) (MSE)

$$\text{MSE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2. \quad (3.11)$$

Große Fehler werden so stärker betont, da ihr Quadrat überproportional größer ist als der von kleinen Fehlern.

Da dies aufgrund der Quadratur allerdings nicht direkt die Intuition anspricht, wird stattdessen häufig die Wurzel der mittleren Quadratischen Abweichung (engl. *Root Mean Square Error*) (RMSE) verwendet:

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2}. \quad (3.12)$$

Die entspricht einer mittleren Abweichung mit einer Betonung größerer Fehler, da die Quadratur bei größeren Abweichungen überproportional größer ist und der Mittelwert vor der Wurzel ermittelt wird.

Des Weiteren gibt es als Beispiele für komplexere Metriken, welche an dieser Stelle aufgezeigt sein sollen, die erklärable Varianz (engl. *explained variance*) (R^2)

$$R^2(y, \hat{y}) = 1 - \frac{\text{Var}\{y - \hat{y}\}}{\text{Var}\{y\}}, \quad (3.13)$$

den maximalen Fehler (engl. *maximal error*):

$$\text{Max Error}(y, \hat{y}) = \max(|y_i - \hat{y}_i|) \quad (3.14)$$

und den mittleren quadratischen, logarithmischen Fehler (engl. *Mean Squared Logarithmic Error*) (MSLE):

$$\text{MSLE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (\log_e(1 + y_i) - \log_e(1 + \hat{y}_i))^2. \quad (3.15)$$

Er gewichtet die Fehler entsprechend einer logarithmischen Skala, was Ausreißer kompensieren kann.

3.4.3 Spezielle Metriken für SZP

Von Weisheit werden in [Wei19b] die speziell angepassten Qualitätsparameter *Prognoseabweichung* und *Prognosesprünge* mit einer zusammenfassenden Gesamtprognosegüte für die Bewertung von SZP verwendet. Für die Prognoseabweichung werden innerhalb einer Signalgruppe s die Abweichungen der prognostizierten Schaltzeitpunkte $t_{GE\ GA,s}^p(k)$ mit den tatsächlichen Schaltzeitpunkten $t_{GE\ GA,s}^r(k)$, für alle Zeitpunkte $k \in N$, zu denen eine Prognose berechnet wird, verglichen.

Ergebnis sind drei Qualitätsparameter definiert:

- Prognoseabweichung $Q_{Pr,s}$
- Prognosesprünge $Q_{sp,s}$
- Gesamte Prognosegüte $Q_{ges,s}$

Die Prognoseabweichung beschreibt eine KKR mit Toleranz. Die Toleranz ist abhängig von der realen Restzeit $\Delta t_s^r(k)$. Die (Prognose-)Toleranz beträgt ca. 10% der Restzeit. Der Parameter für Prognosesprünge steigt mit der Anzahl und Größe von Sprüngen in der Prognose. Die beiden Qualitätsparameter können zu einer Größe für die gesamte Prognosegüte $Q_{ges,s}$ zusammengeführt werden. Für diese Arbeit wurde diese Größe angepasst und ist definiert als:

$$Q_{ges,s} = 100 - \min(0.8 * Q_{sp,s}, 60) - \min(0.4 * Q_{Pr,s}, 40). \quad (3.16)$$

3.5 Modelle des maschinellen Lernens

Im Folgenden werden die für diese Arbeit wichtigsten Methoden des maschinellen Lernens vorgestellt. Diese sind Vektor- beziehungsweise Instanz-basierte Methoden wie k -nächste Nachbarn und Support Vector Machines. Als ein weiterer Methodenbereich werden Entscheidungsbäume und zugehörige Ensemblemethoden einschließlich Boosting beschrieben. Zudem werden verschiedene Arten neuronaler Netze sowie Markov-Ketten und Bayes'sche Netze dargestellt.

3.5.1 k -nächste Nachbarn

Bei der k -nächste Nachbarn-Methode (engl. *k-nearest neighbors*) (kNN) handelt es sich um eine nicht parametrische Methode des maschinellen Lernens, deren Grundprinzip von Fix und Hodges [FH51] entwickelt wurde. Eine Weiterentwicklung wurde von Altman [Alt92] vorgenommen. Die Methode eignet sich sowohl zur Klassifikation als auch zur Regression.

Als Nachbarn werden an dieser Stelle die nächstgelegenen Punkte im Merkmalsraum bezeichnet. Um auf Nachbarn zurückgreifen zu können, wird der Trainingsdatensatz mit Zielgrößen dem Modell dauerhaft als Referenz hinterlegt. Die Grundannahme der Methode ist, dass die Abstände zu den nächsten Nachbarn die Klassenzugehörigkeit implizieren. Um eine Stichprobe mit unbekannter Zielgröße zu klassifizieren, werden, basierend auf einer Abstandsmetrik wie beispielsweise der euklidischen Norm, die Abstände zwischen dem noch unklassifizierten Datenpunkt und den k nächstgelegenen Punkten der Referenz ermittelt. Abbildung 3.12 zeigt dies beispielhaft.

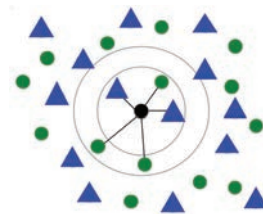


Abbildung 3.12: kNN Beispiel eines zweidimensionalen Merkmalsraumes

Neben der einfachen Klassenhäufigkeit können für die Zuordnung beispielsweise auch durch die Abstände gewichtete Häufigkeiten herangezogen werden. Für die Regression kann beispielsweise der Mittelwert der Zielgröße für die k nächsten Datenpunkte verwendet werden. k ist hierbei ein Hyperparameter.

3.5.2 Support Vektor Machines

Auf einer ähnlichen Idee wie kNN basieren Support Vector Machines SVM [Vap06, TK00, BGV92, Ben92], welche in den 1990er-Jahren entwickelt wurden. SVMs eignen sich ebenfalls sowohl für die Regression als auch für die Klassifikation. Im Folgenden wird ihre Anwendung zur Klassifikation beschrieben.

SVM-Algorithmen suchen eine trennende, lineare Hyperebene zwischen den Punkten verschiedener Klassen im Merkmalsraum. Der Abstand der Ebene zu den Trainingsdaten wird zwecks optimaler Trennung im Training maximiert. Die Abbildung 3.13 veranschaulicht das Prinzip der SVM zur Trennung zweier Klassen auf Basis von zwei Merkmalen. Die entsprechende Formel für die Hyperebenen ist ebenfalls angegeben.

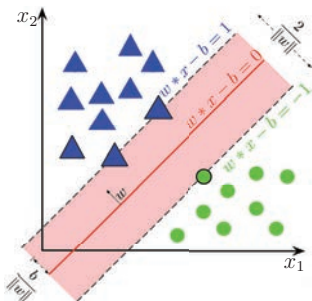


Abbildung 3.13: Beispiel einer SVM

Die Hyperebene, als rote Volllinie gekennzeichnet, liegt mittig zwischen den beiden Grenzen des Saumes (engl. *margin*), welcher maximiert wird [Pla99]. Je klarer die Klassen voneinander getrennt werden können, desto größer ist der Saum und damit die Generalisierungsfähigkeit des Modells. Es ist zu sehen, dass nur 3 Punkte diesen berühren. Diese Punkte werden auch als Stützvektoren bezeichnet, da sie für die Berechnung der Hyperebene und des Saums herangezogen werden. Sie verleihen der Methode ihren Namen.

Es gibt zwei Möglichkeiten mit dem Saum zu arbeiten. Zum einen kann er als harter Saum (engl. *hard margin*) verwendet werden, so dass alle Punkte außerhalb von ihm liegen müssen. Zum anderen können, um eine Generalisierungsfähigkeit zu erhalten, während des Trainings falsche beziehungsweise unsichere Klassifikationen zugelassen werden, indem die entsprechenden Punkte innerhalb des Saums oder auf der falschen Seite der Hyperebene liegen. Hierdurch ist die Methode robust gegenüber Ausreißern. In diesem Fall wird von einem weichen Saum (engl. *soft margin*) gesprochen. Der Umfang der Fehlertoleranz unterliegt Regularisierungen [CV95].

Sobald die Klassen innerhalb des Merkmalsraums nicht mehr linear trennbar sind, können Hyperebenen sie nicht mehr trennen. Durch eine Transformation in ein anderes Koordinatensystem mit gegebenenfalls höherer Ordnung kann dies umgangen werden. Mit Hilfe eines Kernels [ABR64] kann dies mit einem vergleichsweise geringen Rechenaufwand geschehen, denn mit Hilfe geeigneter Kernelfunktionen können Distanzen im höherdimensionalen Raum berechnet werden, ohne Koordinatentransformationen explizit ausführen zu müssen, was für die Anpassung einer Hyperebene in diesem Raum ausreichend ist. Praktisch wird die Transformation indirekt über die Metrik integriert.

Für die Trennung von mehr als zwei Klassen voneinander können verschiedene Verfahren angewendet werden. Ein indirektes Verfahren zu einer solchen Multiklassifikation ist der One-Versus-The-Rest-Ansatz [Vap98]. Hier wird für jede Klasse eine Hyperebene angepasst, welche diese von allen anderen Klassen, welche hierfür als eine Klasse betrachtet werden, abgrenzt. Beim One-Versus-One-Ansatz werden mit einem höheren Rechenaufwand für jede binäre Klassenkombination Hyperebenen ausgelegt.

3.5.3 Entscheidungsbäume

Entscheidungsbäume (*engl. Decision Trees*) (DT) können unter anderem als ML-Methode für Regressions- und Klassifikationsaufgaben (*engl. Classification And Regression Trees (CART)*) [Be84, Aun21] verwendet werden. Im Folgenden werden sie am Beispiel von Klassifikation erläutert.

Abbildung 3.14 zeigt ein Beispiel mit zwei Merkmalen. Links ist der Merkmalsraum samt den drei Klassen der Zielgröße dargestellt. Auch sind die Einteilungen durch den Entscheidungsbaum bereits eingetragen. Entscheidungsbäume unterteilen den Merkmalsraum immer in rechteckige Flächen beziehungsweise Körper. Der zugehörige Entscheidungsbaum ist auf der rechten Seite dargestellt. In den Knoten (*engl. nodes*) des Baumes, hier als Rechtecke dargestellt, sind die Entscheidungsregeln (Diskriminanzfunktionen) eingetragen, auf deren Basis der jeweils folgende Ast (*engl. branch*) – „True“ oder „False“ im Bild – ausgewählt wird. Der unterste und letzte Knoten einer Folge von Ästen wird als Blattknoten (*engl. leaf*) bezeichnet. Ihm ist eine Klasse zugewiesen und keine Entscheidungsregel.

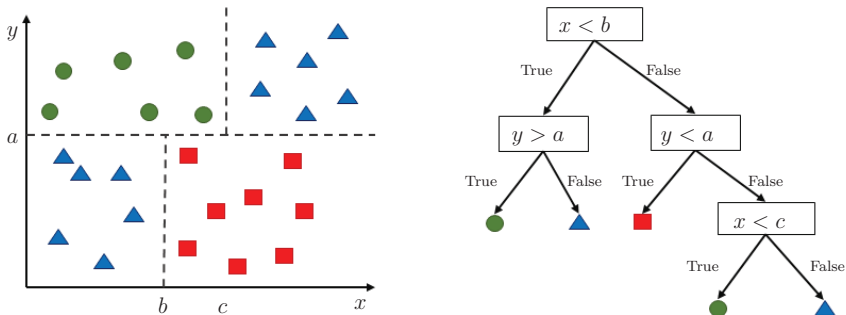


Abbildung 3.14: Beispiel eines Entscheidungsbaumes

Für jeden Weg zwischen dem Wurzelknoten (*engl. root*) – im Bild oben – und den Blattknoten gibt es so einen eigenen Satz an Entscheidungsregeln, der positiv oder negativ zu beantworten ist, um den Blattknoten zu erreichen. Entsprechend kann für jeden rechteckigen Körper im Merkmalsraum ein eigenes mathematisches Modell aufgestellt werden. In Abbildung 3.14 sind die beiden Flächen (Bild links) des Entscheidungsbaumes (Bild rechts) für die grünen Kreislabel bereits vereint. Durch die Entscheidungsregeln wird der gesamte Merkmalsraum abgedeckt. Bei Entscheidungsbaumen werden regelbasiert binäre Entscheidungen getroffen. Die Baumstruktur wird während des Trainings ausgebildet.

Das Training eines Entscheidungsbaumes erfolgt rekursiv in zwei Schritten: Baumwachstum (*engl. growing*) und Baumschnitt (*engl. pruning*). Das Baumwachstum erfolgt über

die weitere Aufteilung und das Hinzufügen von Entscheidungsknoten. Die Erforderlichkeit dieses Schrittes wird anhand der Fehlerrate des jeweiligen Blattknotens entschieden und erfolgt, bis keine Fehler mehr auftreten oder ein Regularisierungskriterium erreicht wird. Regularisierungen sind in diesem Kontext

- die Tiefe des Baumes: die Anzahl aufeinander folgender Äste,
- die Anzahl an Entscheidungsregeln pro Knoten,
- der Anteil der Trainingsdaten, welche zum Baumwachstum verwendet werden,
- die maximale Anzahl an ergänzten Blattknoten und
- die minimale Fehlerreduktion.

Häufig verwendete Fehlermaße sind beispielsweise die negative Kreuzentropie, siehe Gleichung 3.8, und der Gini-Index, siehe Gleichung 3.9. Alternativ können die dem Wurzelknoten am nächsten gelegenen Blattknoten weiter geteilt werden oder nur solche Blattknoten, an denen die größte Fehlerreduktion bewirkt werden kann. Dies kann bereits Teil des Baumschnitts sein. Ein Ast wird nur dann behalten beziehungsweise angelegt, wenn er wichtig genug ist. Hierfür wird die Modellkomplexität, als Anzahl der Knoten, dem Fehler gegenübergestellt. Entsprechende Regularisierungen sind unter anderem:

- die minimale Wichtigkeit des Blattknotens und
- der Regularisierungsparameter Gamma, der die minimal erforderliche Fehlerreduktion zum weiteren Teilen eines Knotens vorgibt.

Entscheidungsbäume können im Gegensatz zu den meisten Methoden des maschinellen Lernens als „White Box Modell“ betrachtet werden. Sobald ein Modell trainiert wurde, kann der gebildete Entscheidungsbaum in einem Diagramm dargestellt werden. Die hohe Interpretierbarkeit ist ein großer Vorteil von Entscheidungsbäumen [Alp19]. Die Baumarchitektur ist jedoch stark von Details im Datenmodell abhängig [HRJ09]. Trotzdem sind Entscheidungsbäume gegenüber irrelevanten Merkmalen robust, da sie relevante Merkmale für Entscheidungsregeln priorisieren.

Uneingeschränkte Bäume lernen den Trainingssatz mit Rauschen auswendig, indem sie den Merkmalsraum in eine Vielzahl an Bereichen einteilen. Abbildung 3.6 zeigt eine beispielhafte Regularisierung. Für jeden Ast müssen im Trainingsdatensatz mindestens vier Stichproben vorliegen. Durch diese übermäßige Stückelung geht die Generalisierungsfähigkeit des Modells verloren. Außerdem kann die Teilung des Merkmalsraum nur durch Parallelen zu seinen Achsen erfolgen; das heißt Diagonalen und ähnliches können nicht gezogen werden. Entscheidungsbäume werden im Gegensatz zu neuronalen Netzen als schwache Lerner betrachtet, da sie bei einer einfachen Verwendung meist vergleichsweise schlechte Ergebnisse erzielen.

3.5.4 Ensembles

Die Performanz von ML-Modellen kann verbessert werden, indem mehrere Modelle verwendet werden. Das Ergebnis einer so zusammengestellten Vorhersage kann dabei über verschiedenen Metriken bestimmt werden, wie beispielsweise Mittelwertbildung, die Ausgabe des wahrscheinlichsten Ergebnisses oder eines entsprechend gewichteten Mittels.

Diese Zusammenstellungen werden als Ensembles bezeichnet, im Englischen alternativ auch „*committees*“. Ensemble-Methoden haben den Vorteil, dass sie besonders parallelisiert werden können und somit die Rechenzeit verringern können.

Häufig wird diese Idee in einer Abwandlung verwendet, dem sogenannten „*Bagging*“, kurz für Bootstrap-Aggregation. Entsprechend einem Bootstrap-Test werden aus dem Datenmodell Teilmengen (der Merkmale) gebildet, wobei Überschneidungen der Mengen nicht ausgeschlossen sind.

Ein Bagging-Ensemble, welches aus einfachen Entscheidungsbäumen besteht, wird als Wald (engl. *forest*) bezeichnet. Eine besonders leistungsfähige Form hierfür sind Random Forests [Bre01, BC20], bei der die einzelnen Bäume mit einer zufälligen Untermenge an Merkmalen trainiert werden. Sie entscheiden mittels Mehrheitsentscheidung. Typische Regulationskriterien sind das Erreichen der maximalen Anzahl an Bäumen oder eines ausreichend kleinen Fehlers.

Eine Weiterentwicklung des Bagging ist das Boosting. Ein Ensemble wird hierbei sukzessive gebildet, indem neue Modelle ihren jeweiligen Vorgänger korrigieren und somit ein zunehmend besseres Modell erzeugen. Aus den Modellen wird am Ende des Trainings ein Ensemble gebildet. So können schwache Lerner wie Entscheidungsbäume stark werden, also ihre Leistungsfähigkeit einem „boost“ unterzogen werden. Zu den Boosting-Methoden gehören beispielsweise adaptive boosting (AdaBoost) [FS97, HRZZ09], Gradient Boosting und eXtreme Gradient Boosting (XGB) [CG16].

AdaBoost kann mit linearen Gleichungen, SVM und Entscheidungsbäumen arbeiten. Gradient Boosting basiert auf Entscheidungsbäumen (CART) und trainiert das Ensemble mittels Gradientenabstiegsverfahren. Der Unterschied zwischen Gradient Boosting und XGB ist die algorithmische Umsetzung. Das „Extreme“ in XGB steht für die optimierte Berechnung: Die Bäume können parallel berechnet werden; das Problem ist so formuliert, dass klassische Optimierer verwendet werden können; es können sowohl sehr große Datenmengen effizient verarbeitet sowie kleine Datenmengen (mit ungleichen Klassenhäufigkeiten) erfolgreich trainiert werden. Die benötigte Rechenzeit für das Training wächst exponentiell mit der Anzahl an Merkmalen.

3.5.5 Neuronale Netze

Künstliche neuronale Netze sollen die neuronalen Netzwerke menschlicher Gehirne nachbilden. Sie bestehen aus miteinander vernetzten Neuronen. Im Gegensatz zu Entscheidungsbäumen wird ihre Architektur über Regularisierungsparameter festgelegt. Obwohl neuronale Netze [MW43,Ros57] bereits zu Beginn des letzten Jahrhunderts erfunden wurden, hat es in Ermangelung eines effektiven Optimierungsalgorithmus [Dre90] bis zum Ende des Jahrhunderts gedauert, bis die Methode mit weiterentwickelten Optimierungsalgorithmen verbreitete Anwendung finden konnte. Ein vergleichsweise junger und verbreiteter Optimierungsalgorithmus ist Adam [KL15].

Abbildung 3.15 zeigt beispielhaft ein künstliches neuronales Netzwerk. Es besteht aus einer Eingabeschicht (engl. *input-layer*) (x bzw. i), einer Schicht verdeckter Neuronen (engl. *hidden-layer*) (h) und einer Ausgabeschicht (engl. *output-layer*) (o). Die Eingabeschicht besteht im Beispiel aus zwei Eingabeneuronen, in die der Input eingespeist wird. Die verdeckte Schicht besteht aus Neuronen, welche die sogenannten Ausgänge der Eingabeneuronen weiterverarbeiten und -leiten. Die Ausgabeschicht besteht in der Abbildung aus einem Ausgabeneuron, welches die Schätzung der Zielgröße ausgibt.

Bevor der Aufbau und die Funktionsweise der Neuronen selbst beschrieben werden, wird näher auf ihre Verknüpfung eingegangen. In neuronalen Netzen werden die Informationen der Merkmale mittels Neuronen verarbeitet und miteinander verknüpft. Hierfür werden sie durch ein Netzwerk aus Neuronen propagiert, in Abbildung 3.15 von links nach rechts. Hierbei müssen nicht alle Neuronen mit allen anderen verbunden sein. Sobald alle Neuronen von zwei aufeinanderfolgenden Schichten vollständig miteinander verknüpft sind und auch nur mit diesen, wird von einem *feed-forward*-Netzwerk gesprochen. Dies ist die verbreitetste Netzwerkkonstruktion.

In Abbildung 3.16 ist der Aufbau des ursprünglichen Perzeptrons (Neurons), welche für künstliche neuronale Netze verwendet wird, dargestellt. Es wird auch als Perzeptron bezeichnet. Diese Form, mathematische Formeln für maschinelles Lernen grafisch darzustellen, ist insbesondere für Neuronen üblich.

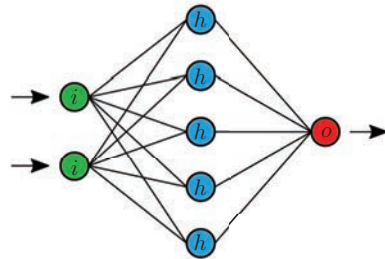


Abbildung 3.15: Veranschaulichung eines neuronalen Netzes

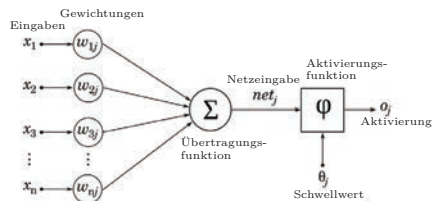


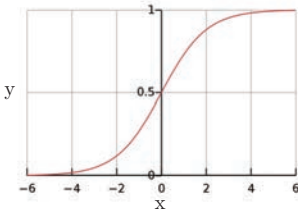
Abbildung 3.16: Veranschaulichung eines Neurons [Wut21]

Die von den Verknüpfungen kommende Information (Eingabe x) wird mittels einer gewichteten Summe

$$net_j = \sum_{i=1}^n (x_i w_{i,j}) \quad (3.17)$$

bewertet. x_i ist die Eingabe und $w_{i,j}$ sind die Gewichtungen. Ergebnis ist eine Netzeingabe net_j . Für die Bewertung der Netzeingabe wird eine Aktivierungsfunktion ϕ verwendet, um die Ausgangsgröße o_j zu erzielen. Ausgangsgrößen von neuronalen Netzen werden wie hier zum Teil als Aktivierung bezeichnet.

Die Wahl der Aktivierungsfunktion ist frei. Üblich ist es, für alle Neuronen die gleiche mathematische Funktion $f(net_j)$ aus der Menge der Sigmoidfunktionen zu verwenden. Beispiele hierfür sind die logistische Funktion,



$$f = \frac{1}{1 + e^{-net_j}}, \quad (3.18)$$

siehe auch Abbildung 3.17, der Tangenshyperbolicus

$$f = \tanh(net_j) = \frac{e^{net_j} - e^{-net_j}}{e^{net_j} + e^{-net_j}} \quad \text{und} \quad (3.19)$$

der Arcustangens

$$f = \arctan(net_j). \quad (3.20)$$

Abbildung 3.17: Plot der logistischen Funktion

All diese Funktionen haben eine ähnliche S-Form und horizontale Asymptoten für $x \Rightarrow \pm\infty$. Mittels eines zusätzlichen Parameters können sie angepasst werden, so dass die Ausformung der Bäuche des S unterschiedlich stark ist. Der Schwellwert (engl. *offset/bias*) bestimmt, ab welcher Größe, diese als Aktivierung des Neurons weitergegeben wird. Des Weiteren können, abweichend vom klassischen Perceptron, Rectified Linear Unit (ReLU)-Funktionen

$$f = \max(0, net_j) \quad (3.21)$$

verwendet werden, welche von Optimierungsalgorithmen schneller berechnet werden können.

Unabhängig von ihrer Netzarchitektur und Aktivierungsfunktion werden solche künstlichen neuronalen Netze aus einfachen Neuronen als Multi Layer Perzeptron (MLP) bezeichnet, sobald mehr als eine verborgene Schicht vorliegt MLP [Hin89, GB10].

Eine Weiterentwicklung stellen Recurrent Neural Networks RNN [WHR86] dar, siehe Abbildung 3.18. Die verdeckte Schicht (h) bildet bei RNN auf sich selbst ab, indem ihre Ausgaben von der letzten Eingabe (x) einer Stichprobe zur nächsten behalten und als Eingabe der Schicht verwendet werden. Die Gewichtungen der Aktivierungen sind hierbei mit getrennten Vektoren U , V und W bezeichnet. So haben vergangene Eingaben Einfluss auf die Ausgabe (o).

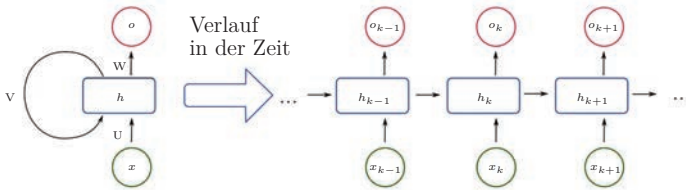


Abbildung 3.18: Veranschaulichung eines RNN

Neben den oben beschriebenen Neuronen können hierfür auch weiterentwickelte Neuronen verwendet werden. Die bekanntesten hierfür sind Long Short Term Memory Network (LSTM) [HS97, Elm90] und Gated Recurrent Unit Network (GRU) [HS17, DS17].

3.5.6 Markov-Ketten

Systeme, welche zwischen Zuständen wechseln, können mittels grafischer Darstellungen dieser Zustände und ihrer Zustandstransitionen beschrieben werden. Für LSA können so Phasen(-übergänge) beschrieben werden, wie in den Phasenfolgediagrammen in Anhang B zu sehen ist. Insofern die Übergangswahrscheinlichkeiten zwischen Phasen beziehungsweise deren Übergängen bekannt sind, handelt es sich um eine Markov-Kette [KURT-2020]. Voraussetzung für eine Modellierung von Zustandswechseln mit Hilfe einer Markov-Kette ist die Erfüllung der Markov-Eigenschaft. Sie besagt, dass die Transitionswahrscheinlichkeit zwischen dem aktuellen und nächsten Zustand nur vom aktuellen Zustand abhängig und damit konstant ist [BAS-2020]. Dies wird auch als Gedächtnislosigkeit bezeichnet.

Abbildung 3.19 links zeigt ein Beispiel für eine Markov-Kette; mit den Zuständen S_i als Knotenpunkten (engl. *nodes bzw. vertices*), welche über Kreise symbolisiert werden, und Transitionen (engl. *links, edges, arcs*) als diese verbindende Pfeile. Die Transitionswahrscheinlichkeiten $P_{i,j}|i, j \in M$, mit denen die Transitionen belegt sind, können in einer Matrix zusammengefasst werden, siehe Abbildung 3.19 rechts.

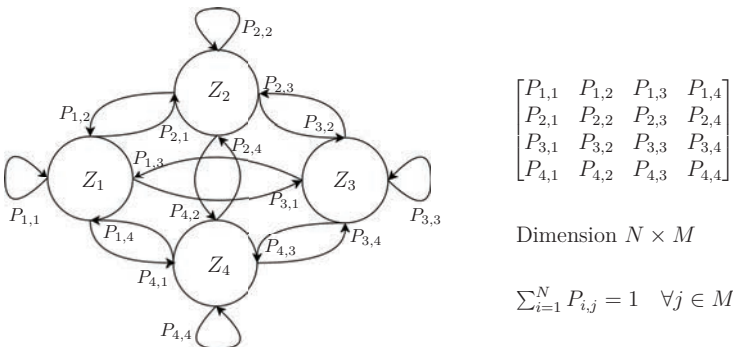


Abbildung 3.19: Veranschaulichung einer Markov-Kette

3.5.7 Bayes'sche Netze

Eine andere Form von Graphmodellen stellen Bayes'sche Netze (BN) (engl. *Bayesian networks / directed graphical models*) dar. Sie lassen sich ähnlich wie Markov-Ketten strukturieren. Die Markov-Eigenschaft ist für sie keine Bedingung, denn sie veranschaulichen über bedingte Wahrscheinlichkeiten die kausalen Zusammenhänge und Abfolgen des Systems. Die Transitionswahrscheinlichkeiten werden wie für Markov-Ketten in Tabellen (engl. *conditional probability tables*) (CPTs) zusammengefasst. Im Gegensatz zu Markov-Ketten werden alle Ausprägungsmöglichkeiten einer Zufallsvariablen in einem Kreis dargestellt, siehe Abbildung 3.20. Die mit ihr / dem Kreis in Zusammenhang stehenden Zufallsvariablen sind über Transitionen verbunden. Die bedingten Wahrscheinlichkeiten bilden den

aktuellen Zustand in Abhängigkeit zu den vorausgegangenen Zuständen der Zufallsvariablen ab. Entsprechend bestehen hierarchische Zusammenhänge. Sogenannte Kindknoten sind von Elternknoten abhängig. Bayes'sche Netze können in geschlossenen und offenen Graphen vorliegen, das heißt mit Ringschlüssen oder als Bäume.

Abbildung 3.20 zeigt ein Beispiel für ein Bayes'sches Netz, welches die kausalen Zusammenhänge zwischen den diskreten (binären) Zufallsvariablen „Wolkig“, „Regen“, „Rasensprenger“, und „Nasses Gras“ mit den zugehörigen CPT veranschaulicht. Der Knoten „Wolkig“, ist in diesem Beispiel der Elternknoten der beiden Zufallsvariablen „Rasensprenger“ und „Regen“. Der Knoten „Nasses Gras“ wiederum ist ein sogenannter Kinderknoten der Knoten „Rasensprenger“ und „Regen“. Die CPT von Knoten ohne Eltern, wie die des Knoten „Wolkig“, beinhalten lediglich unbedingte Wahrscheinlichkeiten [Hec21].

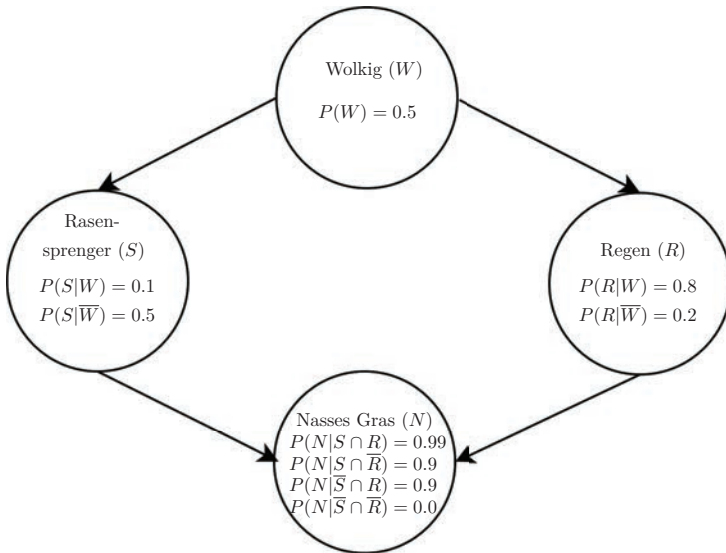


Abbildung 3.20: Veranschaulichung eines Bayes'schen Netzes

Für die Modellerstellung eines BN gibt es im Wesentlichen zwei unterschiedliche Ansätze. Sofern bedingte Abhängigkeiten zwischen Zufallsvariablen im Vorfeld bekannt sind, kann die grafische Struktur vorgegeben werden. In diesem Fall werden die CPT der Knoten beziehungsweise die Wahrscheinlichkeitsverteilung der Zufallsvariablen aus den Trainingsdaten ermittelt. Weiterhin besteht die Möglichkeit unter Anwendung spezieller Algorithmen sowohl die Struktur eines BN als auch die CPT aus Trainingsdaten zu erlernen [Hec21]. Bei beiden Ansätzen findet keine Differenzierung in Merkmale und Zielgrößen statt, d. h. mögliche Zielgrößen werden als Knoten des Netzes modelliert. Das BN gibt als Ausgabewert den ermittelten wahrscheinlichsten Zustand der Zielgröße basierend auf der zugehörigen Stichprobe aus.

3.5.8 Diskussion der Methoden

Die in Abschnitt 2.8 formulierten Anforderungen an SZP betreffen insbesondere die verwendeten ML-Methoden. Für die Echtzeitfähigkeit hat ein Modell nur begrenzte Ressourcen zur Verfügung. Die Berechnung einer Vorhersage muss rechtzeitig abgeschlossen sein. Auch muss das Training, da es regelmäßig wiederholt werden muss, für die Menge an Signalgruppen wortwörtlich über Nacht erfolgen können. Dies betrifft nicht nur die begrenzte Rechenzeit der Nachtstunden, sondern auch bereitgestellte Rechenleistung und allozierbaren Speicher.

Gleichzeitig sollten die Modelle im Betrieb über eine Robustheit gegenüber Ausreißern und Latenzen, also zum Vorhersagezeitpunkt fehlenden Daten, verfügen, wohingegen während ihrer Auslegung die Interpretierbarkeit und die Möglichkeiten zur Handhabung der Modelle von Interesse sind. Die in diesem Kapitel vorgestellten Methoden stellen einen Querschnitt über die Ansätze überwachten maschinellen Lernens dar und verfügen über entsprechend unterschiedliche Eigenschaften.

Ressourcenbedarf

Betreffend des Ressourcenbedarfs ist XGBoost die Methode, welche am tiefgreifendsten optimiert wurde. Sie umfasst Algorithmen, welche auf eine möglichst schnelle Berechnung bei gleichzeitiger Sparsamkeit betreffend des Speicherverbrauchs insbesondere für große Datensätze ausgelegt wurden. Die Datenverarbeitung für Training und Test läuft mit fast konkurrenzlosen Rechenzeiten durch eine möglichst umfassende Parallelisierung [CG16]. Allerdings steigt die Rechenzeit exponentiell mit der Anzahl der Merkmale.

Künstliche neuronale Netze können große Datensätze mit vielen Merkmalen schneller verarbeiten, sind allerdings auch auf diese Datenmengen angewiesen, um ausreichende Performanz zu zeigen. Sie können ebenfalls parallelisiert werden. Je komplexer die verwendeten Neuronen sind, desto mehr Speicher wird benötigt. LSTM verfügen über die komplexesten hier vorgestellten Neuronen.

Vergleichsweise speichereffizient sind SVM, da sie lediglich mit den Stützvektoren der Trainingsdaten arbeiten. Auch berechnen sie die Vorhersagen nach dem Training schneller als beispielsweise künstliche neuronale Netze. Ähnlich wie bei XGBoost steigt die benötigte Rechenzeit mit der Anzahl der Merkmale, in diesem Fall kubisch.

Langsamer als SVM sind kNN, da sie für jede Vorhersage einen Abgleich mit dem gesamten Trainingssatz durchführen. Die langsamste hier vorgestellte Methode sind Bayes'sche Netze, ihr Ressourcenbedarf kann so umfangreich sein, dass, insbesondere wenn der Algorithmus selbst einen Grafen ableiten soll, das Training abgebrochen werden muss. Ohne Abbruch werden die vergleichsweise längsten Trainingszeiten erreicht bei gleicher Verfügbarkeit der Ressourcen.

Robustheit

XGB gehört nicht nur bezüglich des Ressourcenbedarfs, sondern auch hinsichtlich der Robustheit mit zu den führenden Algorithmen. Sie bilden aus der vergleichsweise schwachen Methode der Entscheidungsbäume über Bagging und Boosting ein Ensemble, welches gegenüber irrelevanten Merkmalen robust ist, da diese für Zweige nicht ausgewählt werden und das Ensemble eine Selbstkorrektur vornimmt.

Diese Stärke der Generalisierungsfähigkeit zeigen ebenfalls SVM. Sie sind durch die Verwendung von ausschließlich den Stützvektoren gegenüber Ausreißern und Daten mit Bias robust. Eine Anpassungsfähigkeit an Nichtlinearitäten ist durch die Verwendung von Kernels gegeben. Durch die konvexe Formulierung als begrenztes Problem der quadratischen Programmierung wird das globale Optimum stets gefunden, auch wenn die Anpassung an die Daten unzureichend ist.

Eine ähnliche Robustheit gegenüber Ausreißern zeigen Bayes'sche Netze, da sie über die Häufigkeitsverteilungen innerhalb der Stichprobe agieren. Deswegen ist es besonders wichtig, dass die Stichprobe hohe repräsentative Eigenschaften besitzt. Wechselwirkungen im Sinne von Rückkopplungen zwischen den bedingten Wahrscheinlichkeiten können vom Modell nicht erfasst werden.

Die Methode der kNN ist hingegen sensitiv gegenüber Ausreißern und kann auch mit fehlenden Werten in Stichproben nicht umgehen. Ungleiche Häufigkeiten der Klassen in der Zielgröße können nicht ausgeglichen werden wie bei SVM, sondern führen zu einer Bevorzugung der häufigeren Klasse, da diese dadurch auch mit einer höheren Wahrscheinlichkeit häufigster Nachbar ist.

Besonders empfindlich betreffend irrelevanter Merkmale sind künstliche neuronale Netze, da diese erst über die Gewichte der Neuronen beim Training hin zu einem Gewicht von Null parametrisiert werden müssen. Deshalb verschlechtern irrelevante Merkmale im Datenmodell das Ergebnis des neuronalen Netzes. Neuronale Netze hingegen gleichen Rauschen und fehlende Merkmale innerhalb einer Stichprobe ebenso wie Nichtlinearitäten vergleichsweise gut aus.

Interpretierbarkeit

Mit neuronalen Netzen lassen sich Modelle bilden, die nur sehr schwer interpretierbar sind. Insofern die Netzelemente interpretiert werden sollen, müssen aufwändige Analyse- und Vergleichsverfahren herangezogen werden, um diese deuten zu können. Entsprechend können neuronale Netze unerklärliche Verhaltensweisen zeigen.

Besser interpretierbar sind SVM, wenn es möglich ist, ihre Hyperebenen grafisch zu visualisieren, beispielsweise durch eine Menge an Grafen, die die Untermengen des Merkmalsraumes darstellen. Selbiges gilt für kNN.

Bayes'sche Netze und Markov-Ketten sind einfach interpretierbar, da sie über grafische Strukturen hergeleitet werden. Diese sind vergleichsweise intuitiv verständlich. Zudem geben sie eine unmittelbare Interpretierbarkeit durch die verbundenen Wahrscheinlichkeiten. Unbekannte Zusammenhänge können zudem für kleine Datensätze mit Hilfe der zugehörigen Algorithmen mittels Grafenbildung aufgedeckt werden. Phasenfolgepläne werden in sehr ähnlichen Grafen veranschaulicht, weshalb diese Methoden für Phasenvorhersagen als besonders geeignet erscheinen.

Ebenfalls eine einfache Interpretierbarkeit bieten alle auf Entscheidungsbäumen basierenden Verfahren, da die Entscheidungsbäume einfach grafisch visualisiert werden können. Auch dies kann zum Teil unbekannte Strukturen aufdecken. Die Baumarchitektur ist jedoch von Details der Trainingsdaten abhängig und kann nicht zu endgültigen Rückschlüssen auf die Datenstruktur und das Systemverhalten führen. Regelbasierte Steuerungsprogramme von LSA können in Baumstrukturen visualisiert werden, weshalb Entscheidungsbäume für diese besonders geeignet erscheinen. Baumstrukturen weisen zudem ähnliche Strukturen auf wie die Grafen von Markov-Ketten und Bayes'schen Netzen.

Modellbildung

Am einfachsten zu modellieren sind kNN. Sie benötigen kein Training, ihnen muss lediglich der (Trainings-)Datensatz und die gewählte Abstandsmetrik sowie die Anzahl der zu berücksichtigenden Nachbarn übergeben werden. Entsprechend bestehen keine Voraussetzungen an die Daten. kNN sind besonders adaptiv, da, um ein anderes Verhalten abzubilden, nur ein neuer Datensatz hinterlegt werden muss. Eine besondere Herausforderung bietet die optimale Anpassung der Anzahl der Nachbarn, da diese mit der Position im Merkmalsraum wechselt.

Für die Modellierung von SVM ist das Kernel über die Hyperparameter einzustellen. Bei bereits bestehenden Implementierungen ist die Menge der verfügbaren Kernel begrenzt. Ihre Neuberechnung hat einen erhöhten Rechenaufwand zur Folge. Zudem ist die Anpassung bei großen Stichproben schwierig, da für diese häufig überlappende Klassen vorliegen. Da nur die Stützvektoren für die Parametrierung verwendet werden und die weiteren Trainingsdaten keine Rolle spielen, können keine probabilistischen Aussagen mit dieser Methode getroffen werden.

Die größte Zahl an Hyperparametern ist für XGBoost auszuwählen. Dies ist Vor- und Nachteil zugleich für diese Methode. XGBoost ist hierdurch sehr adaptiv betreffend der Datenmenge, der Handhabung von Datenlücken und ungleicher Klassenhäufigkeiten der Zielgrößen. Jedoch ist die Optimierung der Hyperparameter entsprechend komplex und benötigt umfangreiche Rechenzeit. XGBoost zeigt schneller Überanpassungen als das verwandte Bagging. Wie bei allen Entscheidungsbäumen können die Klassen nur über Parallelen zu den Achsen des Merkmalsraumes getrennt werden.

Für die Modellierung neuronaler Netze gibt es bisher keine allgemeingültigen Algorithmen, die zuverlässig die Netzarchitektur optimieren können, weshalb die Hyperparameteranpassung besonders zeitaufwändig ist und Vorwissen über das System und die Daten benötigt. Dafür können sie besonders komplexe Muster abbilden. Rekurrente neuronale Netze sind aufgrund ihres Gedächtnisses besonders geeignet für die Vorhersage von Zeitreihen.

Vergleichsweise einfach ist es, Markov-Ketten zu modellieren. Das abzubildende Merkmal wird ausgewählt samt seiner Ausprägungen. Die zugehörigen Transitionswahrscheinlichkeiten können die Algorithmen aus dem Trainingsdatensatz extrahieren. Der Rechenaufwand hierfür steigt mit der Komplexität des Datenmodells. Bei der Merkmalswahl ist die Markov-Eigenschaft zu berücksichtigen, was zu starken Einschränkungen in der Performanz des Modells führen kann. Diese Bedingung trifft auf Bayes'sche Netze nicht zu. Sie können beispielsweise den Entscheidungsprozess von LSA strukturiert nachbilden und sind nur von möglichen Schleifen, welche sie nicht abbilden können, beeinträchtigt. Bei der Nachbildung von komplexen Zusammenhängen stoßen die Algorithmen für die automatisierte Nachbildung der Strukturen jedoch an ihre Grenzen. Diese müssen dann händisch modelliert werden, gegebenenfalls zusammen mit der Berechnung der bedingten Wahrscheinlichkeiten. Bestehende Modelle können wie für Markov-Ketten nachträglich angepasst werden.

Alle in diesem Kapitel vorgestellten Methoden finden für die im Folgenden dargestellten Experimente Anwendung. Am vielversprechendsten erscheinen auf Entscheidungsbäumen beruhende Methoden wie XGBoost aufgrund ihrer strukturellen Ähnlichkeit zu LSA-Entscheidungslogiken.

Kapitel 4

Methodisches Konzept

Dieses Kapitel präsentiert die verwendete Datengrundlage, das im Rahmen der Aufgabenstellung entwickelte Konzept, seine Implementierung sowie die Evaluationsmethodik für diese Arbeit. Damit sollen die ersten beiden Forschungsfragen aus Abschnitt 1.2 betreffend eines geeigneten, allgemeingültigen und übertragbaren Konzepts beziehungsweise Verfahrens und der angemessenen Evaluationsmetriken zur Bewertung daraus resultierender SZP beantwortet werden. Zuvor wird in Abschnitt 4.1 die Datengrundlage beschrieben, da sie eine weitere Grundlage für die in den folgenden Kapiteln beschriebenen Experimente bildet.

4.1 Datengrundlage

Für das Training von ML-Algorithmen ist es wichtig, möglichst fehlerfreie, für das System repräsentative Daten zu verwenden. Hierzu werden in den folgenden Abschnitten die Ergebnisse der Qualitätsanalyse zu den eingelesenen OCIT-Daten vorgestellt.

4.1.1 Datengrundlage der Stadt Kassel

Als Datengrundlage für die exemplarische Umsetzung des Konzepts stehen OCIT-Daten der Stadt Kassel zur Verfügung [Str19]. Signallagepläne und Phasenfolgepläne zu den einzelnen Knotenpunkten finden sich in Anhang B. Die Häufigkeiten der Wartezeiten für GA und GE sind in Anhang C dargestellt.

Die Stadt Kassel nutzt zur Entscheidungsfindung von LSA ausschließlich RB-VA-Regelungen bzw. Steuerungen. Die LSA von Kassel arbeiten phasenorientiert; die Signalläne werden zeitplanabhängig geschaltet. In der Regel werden Wochentags vier und am Wochenende zwei Programme im Verlauf eines Tages geschaltet. Der genaue Zeitpunkt der Programmwechsel ist abhängig vom Knotenpunkt. Zusätzlich werden Rahmenpläne

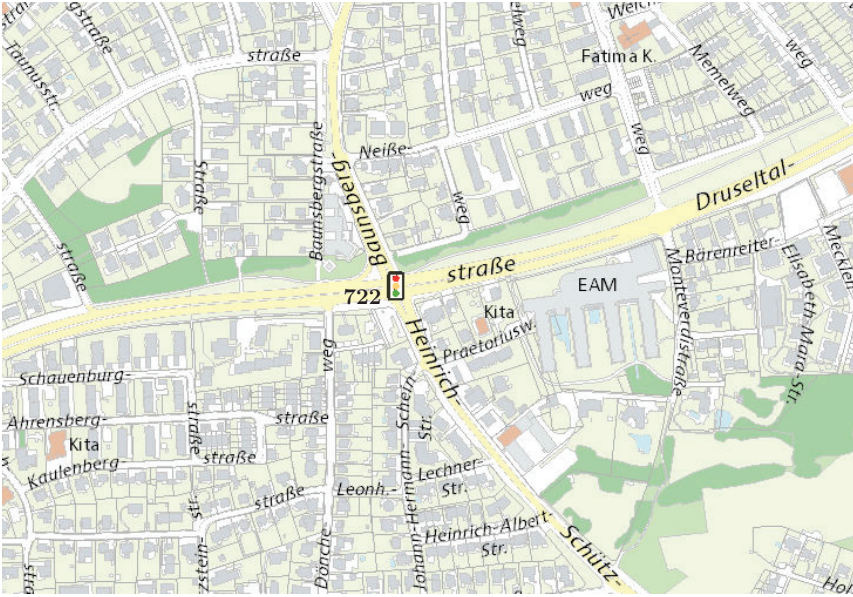


Abbildung 4.1: Lage der abgelegenen LSA im Stadtgebiet Kassel

ne verwendet, die beispielsweise Erlaubnisbereiche für Phasen festlegen. Über die Versorgungsdaten werden verschiedene Parameter wie beispielsweise die Rahmenpläne und Schwellwerte für Zeitlücken an die Schaltlogiken übermittelt. Falls eine Umlaufzeit vorgegeben wird, beträgt diese für die Innenstadt 100 s (nachts z. T. 80 s) und weiter außerhalb angepasst an die jeweiligen Verkehrsbedingungen 70 s bis 90 s. Koordinierungen sind Teil des Signalprogrammmentwurfs und über die Rahmensignalpläne realisiert.

Für die Stadt Kassel werden seit dem Projekt Veronika SZP getroffen. Die Vorhersagen des Unternehmens Traffic Technology Services Inc. Europe (TTS) wurden im Rahmen des Forschungsprojekts VERONIKA [Uni19] aufgesetzt und werden über einen Horizont von maximal 50 s getroffen. Die LSA-Programme der Stadt Kassel weisen in ihrer Komplexität einen für Deutschland anspruchsvollen Standard auf.

Für die hier betrachtete SZP wurden zunächst zwei Knotenpunkte ausgewählt. Am ersten Knotenpunkt Kreuzung „Druseltal-/Baunsbergstraße“ Nr. 722 sollten die Verkehrsbeteiligten möglichst unbeeinflusst von benachbarten Knotenpunkten an den Haltlinien der LSA eintreffen, damit die Detektoren entsprechende Charakteristiken aufzeichnen, siehe Abbildung 4.1.

Unbeeinflusst meint in diesem Kontext beispielsweise: nicht durch von benachbarten Knotenpunkten in Pulks eingeteilte Fahrzeuggruppen. Zudem sollte das LSA-Programm mög-

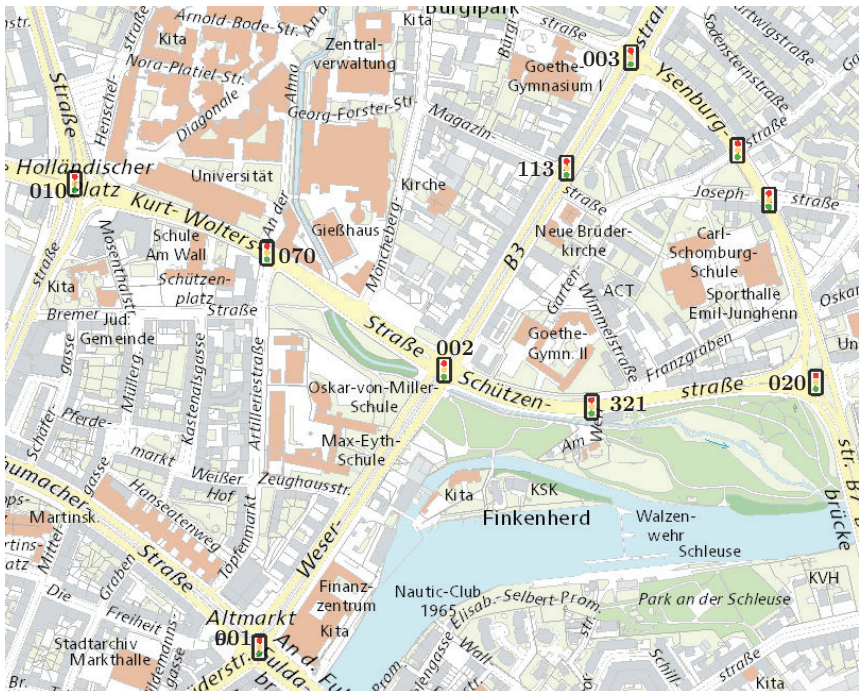


Abbildung 4.2: Lage der LSA in der Innenstadt von Kassel

lichst frei von spezifischen Anpassungen typischer Programmlogiken an den Knotenpunkt, also repräsentativ sein, um die Übertragbarkeit einer SZP möglichst wenig zu reduzieren. Es handelt sich um eine vierarmige Kreuzung mit einer Haupt- und zwei Nebenrichtungen und 22 SG. Die Hauptverkehrsrichtung ist mit einer koordinierten Dauergrün-Schaltung versehen. Es gibt zwei mögliche Phasenfolgen, welche bei Bedarf geschaltet werden. Zudem sind Phasentausche möglich. Drei der vier Zufahrten sind mit Meldepunktketten des ÖPNV ausgestattet.

Der zweite Knotenpunkt „Katzensprung“ Nr.002 soll ebenfalls möglichst repräsentativ sein, ist jedoch in ein engmaschiges Netz von LSA in der Innenstadt eingebettet, weshalb auch die Daten der benachbarten Knotenpunkte und dazwischen liegender FG-LSA aufgezeichnet wurden, um Informationen benachbarter Knotenpunkte in die SZP integrieren zu können, siehe Abbildung 4.2.

Es handelt sich um 27 SG, welche ebenfalls an einer vierarmigen Kreuzung geschaltet

werden. Alle Zufüsse sind mit Meldepunktketten ausgestattet. Die zwei Phasen für die sich kreuzenden Hauptverkehrsströme wechseln sich zyklisch ab. Alle weiteren Phasen werden bei Bedarf geschaltet. Freigabeverlängerungen und Phaseneinschübe sind vorgesehen.

Insgesamt liegen so OCIT-Daten und Dokumentationen zu neun LSA ab April 2019 bis Februar 2020 aus dem Stadtgebiet Kassel vor. Weitere Informationen sind den Abbildungen 4.2 und 4.1 sowie dem Anhang B zu entnehmen.

4.1.2 Qualitätsanalyse der Datengrundlage

Die folgende Qualitätsanalyse wurde im Rahmen eines Bachelorprojektes [Jan22] umgesetzt. Die Daten werden pro LSA und ggf. pro Monat bzw. Signalprogramm getrennt betrachtet. Vermeintliche Fehler, welche aufgrund von Lücken im Verlauf der Zeitreihe der Daten bedingt durch Programmwechsel entstehen, werden nicht als Fehler gewertet. Aus diesem Grund wird zwischen Fehlerrate und Unterbrechungsrate (FR und UR) unterschieden. Die Unterbrechungsrate umfasst im Gegensatz zur Fehlerrate nicht nur alle Fehler wie die Fehlerrate sondern auch Unterbrechungen durch Programmwechsel.

Für eine SZP ist es wichtig, dass die gleiche Information wie die der Logik vorliegt, da diese abgebildet werden soll. Deshalb werden die Daten, insofern anzunehmen ist, dass die protokollierte Information mit der dem Steuergerät vorliegenden identisch ist, als fehlerfrei bewertet.

Die aggregierten Sensordaten, ÖPNV-Meldungen und IST-Vektoren werden nicht untersucht. Die Sensordaten liegen an den Abflüssen der KP und werden von LSA-Logiken nicht verwendet. Sie werden in Kassel mit X im Bezeichner gekennzeichnet. Sie spielen für SZP nur für dem Abfluss nachfolgende Anlagen als indirekte Ankunftszeitinformation eine Rolle. Die dokumentierten IST-Vektoren können nicht überprüft werden, da sie die einzige zur Verfügung stehende Quelle für den LSA-Gesamtzustand darstellen. Plausibilitätsprüfungen sind im Zusammenhang mit den Zeitplänen für Programmwechsel erfolgt. Es wurden keine Abweichungen festgestellt. Selbiges trifft auch auf die ÖPNV-Meldungen zu. An dieser Stelle sei darauf hingewiesen, dass die untersuchten Daten bei Lücken in der Datenreihe von größer 5 s nochmals zugeschnitten werden und damit ein Großteil der hier beschriebenen Fehler aus dem Datensatz entfernt wird, siehe Abschnitt 4.3.2.

Qualität des Protokolls der Umlaufsekunden

Für die Ermittlung der Qualität der aufgezeichneten Umlaufsekunden wird geprüft, ob in jedem Umlauf die Umlaufsekunden t_x von 0 bis t_U durchlaufen werden und der Abstand zwischen den zugehörigen Zeitstempeln der Schrittlänge der Umlaufsekunden entspricht. In der Regel beträgt die Schrittlänge 1 s.

Systembedingte Abweichungen finden sich im ersten und letzten Umlauf eines Signalprogramms an einem Tag. Des Weiteren stellen Umläufe eine Abweichung dar, in denen in den normalen Betriebszustand von einem Fehlerzustand gewechselt wird. In beiden Fällen kann der aktuelle Umlauf abgebrochen oder mittendrin begonnen werden. Zudem kann der Ein- bzw. Ausschaltplan oder ein Plan zum Wechsel des Signalprogramms durchlaufen werden.

Zusätzlich wird regelmäßig die Uhrzeit an allen LSA der Stadt synchronisiert, weshalb geringfügige Zeitsprünge in den Zeitstempeln und Umlaufsekunden zu verzeichnen sind.

Eine solche Synchronisation geschieht dann üblicherweise zu Beginn oder Ende eines Umlaufs, wodurch die ersten oder letzten Umlaufsekunden entfallen oder wiederholt werden.

Die Unterbrechungsraten sind in Abbildung 4.3 zu sehen. Sie zeigt, dass die Unterbrechungsrate der LSA zwischen 0,5 % und 5 %, abhängig von der LSA, liegt. Abbildung 4.4 zeigt die prozentuale Aufteilung der UR nach Fehlern aufgrund von Programmwechseln. Es wird zwischen Fehlern im Zeitverlauf und in der Zählung der Umlaufsekunden und Programmwechseln, unterschieden. Abweichungen im Zeitverlauf konnten zum größten Teil auf die Uhrsynchronisation der LSA zurückgeführt werden, welche mehrmals täglich für alle LSA der Stadt durchgeführt wird. Insofern Abweichungen im Zeitverlauf und Programmwechsel von der Fehlerrate ausgeschlossen werden, wird die FR auf ca. ein Drittel der UR geschätzt.

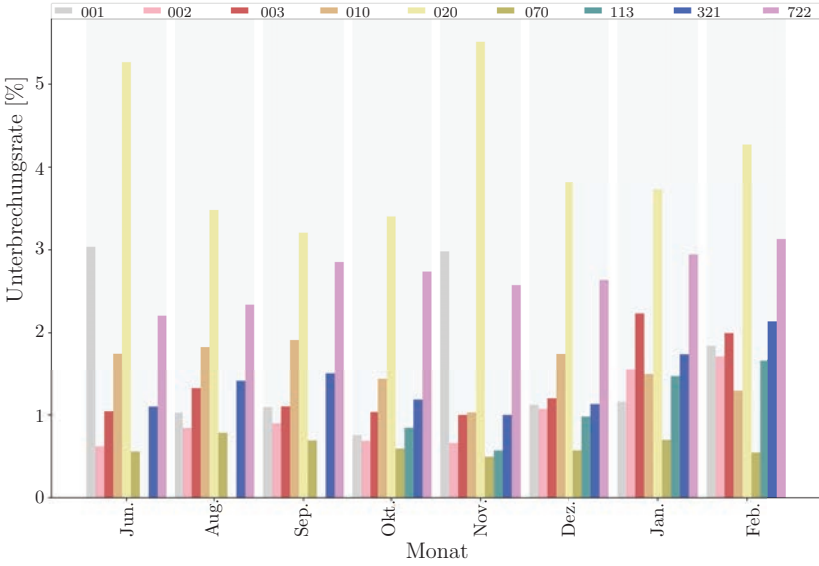


Abbildung 4.3: Unterbrechungsrate der Umlaufsekunden [Jan22]

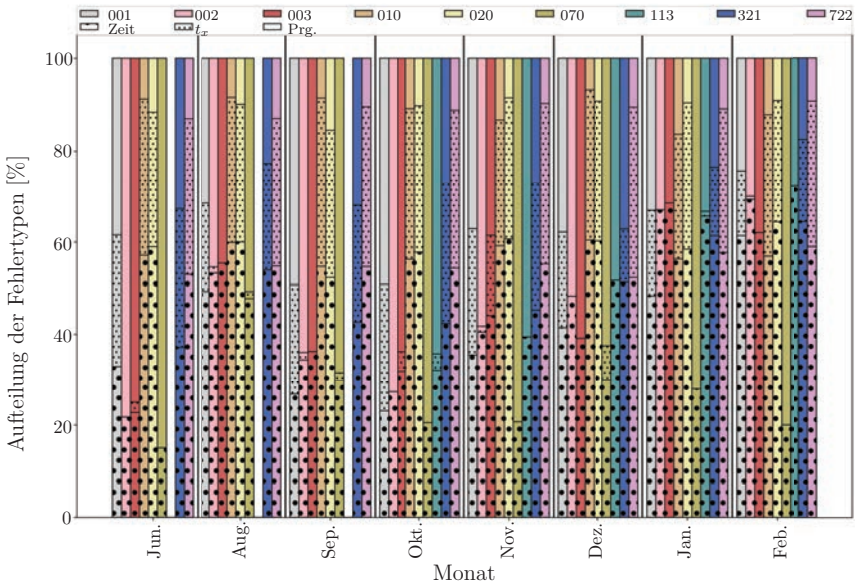


Abbildung 4.4: Prozentuale Differenzierung der Fehler der Umlaufzeit [Jan22]

Signalbilddatenqualität

Die Qualität der Signalbilddaten kann wesentlich durch Datenverluste bei der Datenübertragung zwischen Steuergerät und Zentrale reduziert werden. Die Einhaltung der Signalbildfolge ist grundsätzlich durch die Baugruppen-Signalsicherung an der LSA stets gewährleistet. Eine Abweichung der OCIT-Daten von den tatsächlich geschalteten Signalbildern verringert entsprechend die Qualität der Prognose.

Die Abfolge der Signalbilder ist je nach signalisierter Gruppe (mIV, Rad, ÖPNV, FG) unterschiedlich. Zu dem optischen Signal für FG (Rot, Grün) werden beispielsweise Blindensignale hinzugeschaltet. Dies ist in den Protokollen unter anderem durch mehrfache direkt hintereinander erfolgende Freigaben mit OCIT-Code 48 (Grün) und OCIT-Code 60 (Gelb-Grün) im Wechsel zu erkennen. Eine Überprüfung der Bildfolgen ergab, dass die Fehler- und Unterbrechungsraten LSA-spezifisch ausfallen und sich über die aufgezeichneten Monate hinweg gleichbleibend verhielten.

LSA 010 und 003 zeigen für die Mehrheit der Monate Unterbrechungsraten von über 8 % bis 10 %, siehe Abbildungen 4.5, 4.6. Für die andere LSA wurden in der Regel Fehlerraten von unter 1 % teilweise sogar von ca. 0,1 %, ermittelt. Diese Beobachtungen sind konsistent für alle Programme bis auf LSA 321 in Programm 4. Hier finden sich Unterbrechungsraten von 11 % bis 15 %.

Die durchschnittliche Unterbrechungsrate über alle betrachteten LSA, Programme und Monate beträgt 2,7 % und die Fehlerrate 2,5 %. Die Fehlerraten sind für Programm 4 erhöht im Vergleich zu den anderen Programmen, siehe Abbildungen 4.5, 4.6. Für LSA 001 und 070 wurden die geringsten Fehlerraten ermittelt.

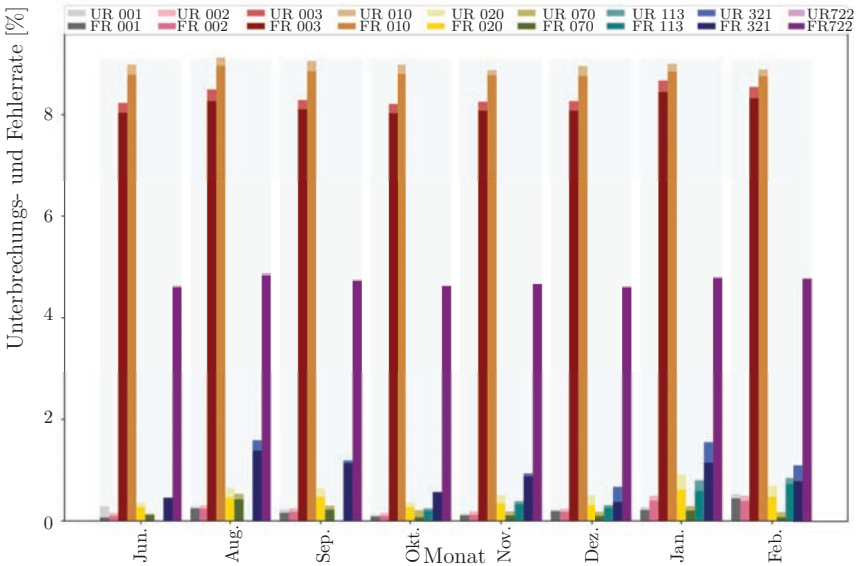


Abbildung 4.5: Fehler- und Unterbrechungsrate für SG aller betrachteter LSA, SP 2 [Jan22]

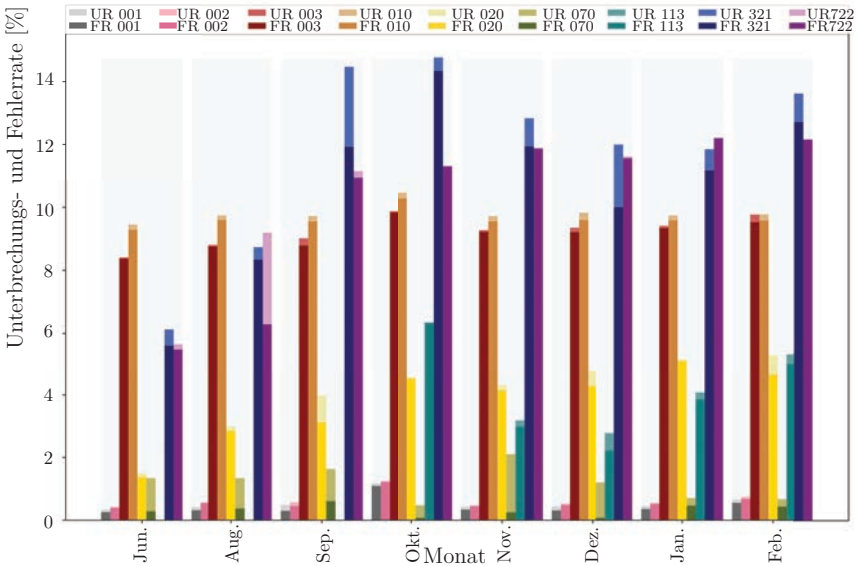


Abbildung 4.6: Fehler- und Unterbrechungsrate für SG aller betrachteter LSA, SP 4 [Jan22]

Detektordatenqualität

Die Detektordaten werden ebenfalls auf die korrekte Abfolge ihrer Aktivierungen (True und False) geprüft. Folgt ein True einem True oder ein False einem False folgt, handelt es sich um einen Fehler oder einen Programmwechsel.

Zusätzlich können weitere Fehler auftreten. So können Personen oder deren Fahrzeuge fälschlich (nicht) detektiert werden. Solange dies gleichermaßen auf die OCIT-Protokolle wie den Dateneingang der Logik zutrifft, hat ein solcher Fehler keine Auswirkungen auf eine Prognosequalität und wird deshalb nicht untersucht. Eine Überprüfung der Signalfolgen für die Daten der Stadt Kassel ergab, dass die Fehlerraten der Detektorfolgen, abhängig von Programm, LSA und Monat, zwischen 0,1 % und 50 % schwanken.

Für Programm 1 und 4 liegen die Fehlerraten zumeist unter 1 % und für Programm 2 und 3 bei unter 5 %, siehe Abbildungen 4.7 und 4.8. Die Unterbrechungsraten betragen das Doppelte bis Vierfache der Fehlerraten. Der Unterschied zwischen Detektorsignalen und SG-Signalen, was den Zusammenhang zwischen Fehler- und Unterbrechungsrate betrifft, kann auf die deutlich geringeren Zeiträume der Detektorerfassung im Vergleich zur Schaltdynamik von SG zurückgeführt werden.

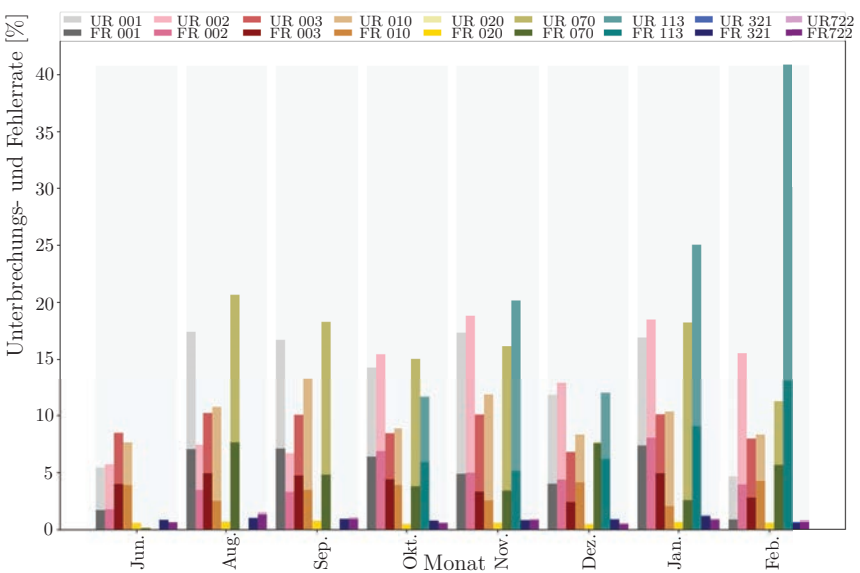


Abbildung 4.7: Fehler- und Unterbrechungsrate der Detektoren aller betrachteter LSA, SP 2 [Jan22]

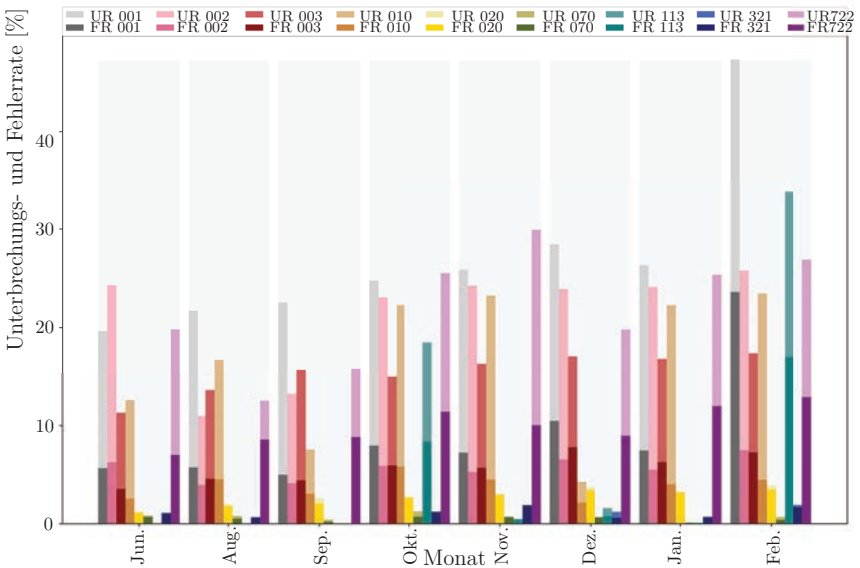


Abbildung 4.8: Fehler- und Unterbrechungsrate der Detektoren aller betrachteter LSA, SP 4 [Jan22]

4.2 Konzept eines Prognoseverfahrens

Zunächst wird im ersten Teil dieses Abschnitts das erarbeitete Konzept in Abschnitt 4.2.1 vorgestellt. Ansätze für die Ausgestaltung dieses sowie zugehörige Überlegungen werden in den anschließenden Unterabschnitten 4.2.2 und 4.2.3 dargestellt. Abschließend erfolgen die Beschreibung des Vorgehens sowie ein Zwischenfazit betreffend dieses Konzepts in den Abschnitten 4.2.4 und 4.2.5.

4.2.1 Konzept des Verfahrens

Das Konzept ist in seinem Ablauf in Abbildung 4.9 dargestellt. Für jedes Programm der zu präzisierenden Schaltzeiten der LSA ist eine eigens angepasste Prognose vorgesehen. Eingespist werden **aktuelle Daten**, welche in das Steuergerät der LSA gegeben oder von diesem ausgesendet werden, zum Beispiel in Form von OCIT-Daten. Diese Daten werden im ersten Schritt aufbereitet, das heißt, die einzelnen Merkmale des ausgewählten **Datenmodells werden sekundlich neu berechnet** und den folgenden Schritten übergeben. Danach erfolgen **Prognosen**. Zentral ist die SZP, dieser können andere Prognosen vorgeschaltet sein. Zudem können **mehrere SZP** gegeben werden, welche anschließend **zu einer SZP zusammengefasst** werden. Dadurch können bei den SZP unterschiedliche Schwerpunkte gesetzt werden, was die Abbildung der Logik betrifft und so in der Summe die Vorhersagequalität erhöht werden.

Die **vorgeschalteten Prognosen** können beispielsweise **Phasenvorhersagen oder Ankunftszeitprognosen** sein. Die Vorhersage der Phasen kann entsprechend Abschnitt 4.2.3 einen zentralen Teil der Logik erfassen und somit das wichtigste Element von RB-Entscheidungslogiken abbilden. Dies reduziert die zugrunde liegende Komplexität für die eigentliche SZP erheblich. Auf Basis der Phasenprognose kann die SZP teilweise sogar entfallen und durch eine einfache Berechnung der Schaltzeiten von GE und GA auf Basis des Schaltzeitpunkts der Phase ersetzt werden. Ankunftszeitprognosen können Detektoraktivierungen schätzen, welche das Schaltverhalten von LSA maßgeblich bestimmen.

Bei **komplexeren Entscheidungslogiken** kann es zudem angeraten sein, die oben beschriebenen SZP von **mehreren Methoden parallel** berechnen zu lassen. Die Zuverlässigkeit der Prognosen kann so für ausgewählte Aspekte gesteigert und durch ihre Vereinigung über ein **Ensemble** zu einer einzigen Prognose insgesamt gesteigert werden.

Zur Laufzeit einer solchen **SZP** ist eine regelmäßige Reparametrierung zum Beispiel einmal wöchentlich angeraten, um in der Parametrierung der Prognose Veränderungen im Verhalten der Beteiligten beziehungsweise im Verkehrsaufkommen abzubilden und aktuell zu halten. Diese sind beispielsweise durch jahreszeitliche Rahmenbedingungen, Schulferien oder länger andauernde Sonderereignisse wie Corona/CoVid-19 bedingt, siehe [Wei17].



Abbildung 4.9: Schema des Prognoseverfahrens

Dieses Schema ist vollständig automatisierbar und wurde ohne die optionalen Elemente bereits von Weisheit und Rudolph [Wei17,RH14] implementiert. Optional sind die vorgeschalteten Prognosen sowie die Verwendung von Ensembles. Individuell zu konfigurieren sind das Datenmodell sowie die verwendeten Methoden zu den einzelnen Prognosen.

4.2.2 Konzept zum Konfigurieren einer Prognose

Zu dem in Abschnitt 4.2.1 vorgestellten Ablauf einer Prognose stellt sich die Frage: Wie kann eine solche Prognose konfiguriert werden? Abbildung 4.10 zeigt das Vorgehen für eine solche Konfiguration. Basis sind **historische Daten**, welche der zu präzisierenden Logik zugeordnet wurden.

Der erste Schritt zum Aufsetzen einer SZP ist es, eine **Systemanalyse durchzuführen**. Ziel ist es, ein automatisiertes Aufsetzen einer Prognose mit individueller Anpassung an eine LSA-Entscheidungslogik zu ermöglichen. In der Analyse werden ermittelt:

1. Latenzzeiten und Datenverluste/-fehler für Anforderungen an die Robustheit
2. der Lösungsraum der Prognose, das heißt, der Wertebereich der Zielgrößen wie zum Beispiel der Gültigkeitsbereiche der Signalgruppen
3. logische Zusammenhänge zwischen Signalgruppen und Detektoren für die Zusammensetzung des Datenmodells
4. logische Zusammenhänge zwischen Signalgruppen (Phasendefinition) als Hinweis auf die Notwendigkeit einer Phasenprognose

Im nächsten Schritt werden **Prognosen ausgewählt**, siehe auch Abschnitt 4.2.3:

- Ist eine Phasenvorhersage beziehungsweise SZP verzichtbar?
- Wird ein Ensemble benötigt?
- Ist eine Ankunftszeitprognose verfügbar/möglich?

Auf Basis dieser Ergebnisse und der Erfahrung im Umgang mit diesem Verfahren, werden die zu verwendenden **Datenmodelle und Maschinelles Lernen (ML)-Methoden**

ausgewählt, welche für die betrachtete LSA möglichst optimale Ergebnisse für SZP ergeben sollten. Das Bilden von Erfahrungswerten im Rahmen dieser Dissertation ist nur in einem begrenzten Rahmen möglich, da lediglich eine vergleichsweise kleine Datengrundlage getestet werden kann. Dies betrifft auch die möglichen Lösungswege zur Beantwortung der oben genannten Fragen.



Abbildung 4.10: Schema der Voranalyse zum Konfigurieren einer Prognose

Ergebnis dieser Schritte ist eine automatisch für die jeweilig betrachtete LSA **konfigurierte Prognose**. Für die Integration von Erfahrungswissen, welches nicht in die Automatisierung integriert werden kann, wird empfohlen, eine grafische Bedienoberfläche, angelehnt an den Ablauf einer Softwareinstallation, zu integrieren, in der alle notwendigen Informationen für eine Entscheidung und automatisiert berechnete Vorschläge dargeboten werden.

4.2.3 Zugrundeliegende Überlegungen

In diesem Abschnitt sollen die bei der Erarbeitung des Konzepts zentralen Überlegungen dargestellt werden, mit denen versucht wurde, die Anforderungen aus Kapitel 2 in das Konzept zum Verfahren zu integrieren. Die Überlegungen orientieren sich an Tabelle 2.1 auf Seite 41. Modellbasierte Verfahren werden im Weiteren nicht differenziert betrachtet aufgrund der Anzahl verschiedener MB-Logiken und der mangelnden Verfügbarkeit detaillierter Beschreibungen.

Ein- und Ausgangsgrößen sowie Lösungsräume

Potenzielle Ein- und Ausgangsgrößen für eine SZP sind bereits in Abschnitt 2.3 mit * respektive ° gekennzeichnet. Es ist üblich, die Schaltzeitpunkte GA und GE für SZP zu verwenden. Sie können in verschiedenen Formaten ausgegeben werden, beispielsweise als Umlaufsekunde t_x in UTC-Zeit oder auch als Dauer in ganzen Sekunden seit ihrer letzten Änderung. Der Lösungsraum der Prognose wird von den tatsächlich auftretenden Schaltzeitpunkten gebildet. Sie stellen zusätzlich die verbreitetsten Eingangsgrößen von SZP dar, vgl. Abschnitt 2.3.3. Nach RiLSA sind bezüglich regelbasiert (RB) VA LSA-Programmen zudem die Versatzzeit, die Umlaufzeit und die Phasenfolge veränderliche (Ausgangs-)Größen eines Signalprogramms, vgl. Abs. 2.3.2 und 2.5.2.

Nachbildung von Entscheidungsprozessen

Allgemein erscheint es sinnvoll, die Entscheidungsprozesse von LSA-Programmen für SZP nachzubilden und so beispielsweise für jede Entscheidungsebene eine Prognose vorzusehen, siehe Abschnitt 2.4 und 2.5.1. Der Ausgang der Prognose einer Ebene kann als Eingang in die untergeordnete Ebene integriert werden. Hierdurch wird die natürliche Programmstruktur imitiert und damit die Prognose qualitativ verbessert.

Die RiLSA sehen verschiedene Entscheidungsebenen vor. Dabei handelt es sich auf der übergeordneten makroskopischen Ebene zum einen um die Signalprogrammwahl. Sie kann über eine Vorhersage der Signalprogrammwahl in ein SZP-Verfahren integriert und der eigentlichen SZP vorgeschaltet werden und diese auswählen. Zum anderen kann die Bildung von Rahmensignalprogrammen diese Ebene bilden. Dort kann eine Prognose über eine ebenfalls vorgeschaltete Vorhersage der Rahmenparameter abgedeckt werden. Falls keine Rahmensignalprogrammbildung vorliegt, werden die Rahmenparameter über die (statischen) Versorgungsdaten in ein LSA-Programm eingepflegt und ändern sich zur Laufzeit der LSA-Logik nicht. Insofern die Architektur von modellbasierten (MB-) Programmen bekannt ist, können die Ebenen analysiert und in eine SZP integriert werden.

Der Aufbau von RB VA LSA-Programmen auf der mikroskopischen Ebene lässt sich ebenfalls in verschiedene Schritte einteilen [Str19]. Experteninterviews zur Erstellung von LSA-Programmen zeigten auf, dass sich eine RB-Logik normalerweise in zwei Teile teilen lässt: eine Vor- und eine Hauptlogik.

Die Vorlogik wertet die Signaleingänge aus, um den Verkehrszustand zu erfassen. Kenngrößen wie Belegungsdauern, Zeitlücken etc. werden abgeleitet. Zudem werden Zustände und Merker (Kenngrößenkombinationen) mit Hilfe von logischen Verknüpfungen für Belegungen, Anforderungen, Prioritäten etc. gebildet. Ihre Auswahl ist Knotenpunkt-spezifisch.

In der Hauptlogik werden basierend auf den Aufbereiteten Daten aus der Vorlogik Entscheidungen über das zukünftige Schaltverhalten der LSA getroffen und über die Signalisierung an die Signalgeber weitergegeben. Die Weitergabe der Entscheidungen erfolgt teilweise getrennt von der Hauptlogik in einer sogenannten Nachlogik.

Es gibt Größen, die sich zur Laufzeit des LSA-Programms zwar im realen Verkehrsablauf verändern, jedoch in RB VA Programmen als konstant angenommen werden. Die Geschwindigkeit von Fahrzeugpuls ist beispielsweise abhängig von der Verkehrsdichte und -stärke und damit Tageszeit, wird jedoch als Konstante in Form der Progressionsgeschwindigkeit bei der Auslegung eines LSA-Programms angenommen. Ähnlich verhält es sich mit Zwischenzeiten, diese werden Ebenfalls beim Entwurf der Steuerung beziehungsweise Regelung festgelegt und sind zur Programmlaufzeit unveränderlich.

Nachbildung RB Vorlogik

Die Teilung von RB-VA-Programmen in Vor- und Hauptlogik kann in eine Prognosearchitektur übernommen werden. Es erscheint sinnvoll, die Vorlogik in die Merkmalsselektion

für Datenmodelle zu integrieren. Zum einen können verschiedene Merkmale zu hauptsächlich verwendeten Kenngrößen(-kombinationen) pro forma in das Datenmodell integriert werden, zum anderen können verdächtige Merkmale über eine Einflussanalyse automatisiert selektiert werden. Die Qualität von Prognosen mit Hilfe von maschinellem Lernen hängt zentral von der Auswahl der Merkmale für die jeweilige Methode ab, weshalb eine systemspezifische Merkmalsselektion, welche möglichst genau die Größen des prognostizierten Systems abbildet, wichtig ist.

Nachbildung RB Hauptlogik

Alle (aggregierten) Größen der Vorlogik werden in der Hauptlogik verarbeitet. Diese agiert üblicherweise phasenorientiert mit einer Phasenlogik als Kernelement, welche ähnlich einem Entscheidungsbaum veranschaulicht werden kann. Für die Entscheidung, ob Phasen geschaltet werden, werden zunächst die Wartezeitkriterien (t_R , Belegungsdauer, Rückstau und ähnliche) herangezogen. Falls von diesen keine Bedingungen für einen Phasenwechsel erfüllt sind, werden die Prioritäten abgearbeitet (zum Beispiel der ÖPNV). Zuletzt werden die Auswertungen anderer Kenngrößen (Zwischenzeit, Belegungsgrad etc.) herangezogen, um zu bestimmen, ob ein Phasenübergang erfolgen soll.

Potentiale einer Phasen(folge)prognose

Die Phasenorientierung der meisten Entscheidungslogiken bietet einen weiteren Anhaltspunkt für eine Prognosearchitektur, da sich hieraus erschließt, dass es sinnvoll ist, zunächst die Phasenfolge mit Phasendauer und Schaltzeitpunkt und darauf basierend erst die Schaltzeiten der SG zu prognostizieren. Insofern nur eine RB VA gemäß RiLSA als reiner Phasentausch oder reine Phasenforderung implementiert ist, sind damit bereits alle Schaltzeiten der SG vorhergesagt. Auch bei Versatzzeitanpassungen zwischen LSA kann bei einer Phasenorientierung davon ausgegangen werden, dass Phasen über Größen aller beteiligten LSA gezielt koordiniert werden, weshalb auch hier eine Vorhersage der Phasenfolge ausreichend erscheint.

Die veränderlichen Größen von Signalprogrammen stehen in direktem Bezug zu den Schaltzeiten. Die Festlegung einer der Größen, zum Beispiel GE einer SG, kann in Abhängigkeit von der Art des LSA-Programms alle anderen Schaltzeiten festlegen.

Bei einer Freigabezeitanpassung ist anzunehmen, dass bei einer Verlängerung einer Freigabe- oder Sperrzeit einer SG mit ihr vereinbare SG ebenfalls verlängert werden, was wiederum eine Verlängerung der Phasendauer darstellt. Auch in diesem Fall wäre eine Prognose der Phasenfolge ausreichend für eine SZP. Eine Vorhersage der Phasenfolge setzt eine Bestimmung der Phasenanzahl voraus.

Sollten SG bei einer Freigabezeitanpassung einzeln, unabhängig von anderen SG geschaltet werden, was unüblich ist, kann zunächst für das gesamte LSA-Programm die Phasenfolge vorhergesagt werden und darauf basierend der Schaltzeitpunkt der entsprechenden SG.

Eingrenzung der Lösungsräume von Prognosen

Die Überlegung zur Eingrenzung des Lösungsraums bieten in Kombination mit dem Vorschlag der Vorhersageabfolgen weitere Möglichkeiten. Über die Analyse der Rohdaten können Abhängigkeiten aufgedeckt werden. Zum Beispiel wird SG 1 in Phase 1 und 4 freigegeben, aber unterliegt nur in Phase 4 einer Freigabezeitverlängerung. Dann ist nur für Phase 4 eine auf einer Phasenfolgeprognose aufbauende Prognose der Schaltzeit von SG 1 erforderlich. Für Phase 1 genügt zur Ermittlung des Schaltzeitpunktes dann die Phasenvorhersage. Entsprechend kann eine Prognose, die Freigabezeitverlängerung betreffend, speziell für Phase 4 trainiert werden. Oder wenn SG 1 in beiden Phasen eine Freigabezeitverlängerung erfährt, ist zu erwarten, dass der Lösungsraum von der Phase abhängig ist. Entsprechend sollte eine Prognose pro Fall, zum Beispiel Phase, verwendet werden, die speziell für diesen trainiert wurde. Dies erfordert keine zusätzliche Rechenleistung der Prognose selbst, bietet jedoch die Möglichkeit der individuellen Anpassung der Prognose an das jeweilige LSA-Programm. Was sich dadurch jedoch umfangreicher gestaltet, ist der Trainingsaufwand der ML-Methoden, da ihre Anzahl steigt.

Variable Umlaufzeiten

Zusätzlich zu den oben beschriebenen Fällen kann nach RiLSA für phasenorientierte RB-VA-Programme eine freie Signalprogrammabildung umgesetzt werden. Dabei ist zusätzlich zu den bereits diskutierten Merkmalen und Variablen die Umlaufzeit veränderlich. Dies hat Auswirkungen auf den minimalen Prognosehorizont. Dieser sollte mindestens einem Umlauf beziehungsweise bei variabler Umlaufzeit der maximal möglichen Wartezeit entsprechen, da ohne weitere Informationen eine Freigabe einer SG zu jedem Zeitpunkt des Umlaufs stattfinden kann [FGS10]. In einem solchen Fall ergeben sich die Umlaufzeit ebenso wie die Schaltzeitpunkte aus der Phasenfolge. Dem entsprechend ist eine Vorhersage der Umlaufzeit zur Ermittlung der Schaltzeitpunkte nicht sinnvoll, sondern sollte über den Horizont in das Verfahren einfließen.

Umgang mit Signalgruppenorientierung

Bisher offen geblieben ist die Diskussion über signalgruppenorientierte VA LSA-Programme für RB und MB: Bei einer Signalgruppenorientierung ist davon auszugehen, dass sich ebenfalls Phasen ergeben, da es nur eine begrenzte Anzahl (bedingt) verträglicher Verkehrsströme an einem Knotenpunkt geben kann. Es ist anzunehmen, dass nicht die Freigaben einzelner SG verlängert beziehungsweise verkürzt werden, sondern alle mit der angepassten SG-Freigabe verträglichen, freigegebenen SG ebenfalls, was wiederum der Bildung von Phasen gleichkommt. Entsprechend erscheint auch für signalgruppenorientierte Schaltlogiken eine Vorhersageabfolge bestehend aus Phasenfolgevorhersage und Freigabezeitverlängerung als sinnvoll.

4.2.4 Vorgehen zur Umsetzung des Konzepts

Aus den vorangegangenen Abschnitten lässt sich eine Strategie für die Erstellung von SZP ableiten. Die Verfahrensarchitektur zur SZP setzt sich aus gekoppelten Einzelprognosen zusammen. Die Einzelprognosen können individuell an die jeweilige Verkehrsinfrastruktur und die integrierten LSA-Arten angepasst und im Netzwerk (räumlich) verteilt werden. Um eine solche Prognosearchitektur aufzubauen, sind die folgenden Schritte sinnvoll:

1. Ermittlung der Entscheidungshierarchie
2. Ermittlung der Entscheidungsprozesse innerhalb einer Hierarchieebene
3. Isolation und Ableitung der Einzelprognosen
4. Ableitung einer Prognosearchitektur
5. Auslegung von Einzelprognosen
 - (a) Ableitung der Ausgangsgrößen
 - (b) Ableitung der potenziellen Eingangsgrößen
 - (c) Ermittlung von Lösungsräumen
 - (d) Einflussuntersuchung
 - (e) Ermittlung von Abhängigkeiten
 - (f) Überarbeitung der Prognosearchitektur
6. Verknüpfung der Prognosen in der Architektur
7. Optimierung der Gesamtprognose

Für die Ermittlung der Entscheidungshierarchie (Schritt 1) sollte die gesamte an Entscheidungen beteiligte Infrastruktur betrachtet werden. Diese Betrachtung sollte für das gesamte Netzwerk über Elemente von Teilnetzen bis hin zu den einzelnen LSA an den Knotenpunkten stattfinden. Dies geht fließend in die Ermittlung der Entscheidungsprozesse (Schritt 2) innerhalb einer Hierarchieebene über. Auf Basis dieser zwei Analysen können erste Einzelprognosen (Schritt 3) gegeben und in einem Entwurf der Architektur (Schritt 4) gruppiert werden. All diese Schritte sind manuell auszuführen.

Nach dieser Betrachtung des Gesamtnetzes werden in Schritt 5. die rohen Ein- und Ausgangsgrößen, welche vom System protokolliert werden, sowie die gegebenenfalls vorhandenen Versorgungsdaten und Dokumentationen analysiert. Dies verfolgt das Ziel, die Ein- und Ausgangsgrößen und Parametrierung der Einzelvorhersagen zu bestimmen. Zudem kann die Darstellungsform der Rohdaten verändert werden, um deren zeitlichen Verlauf mit einem Algorithmus zu untersuchen.

Die Ausgangsgrößen der bisher bestimmten Einzelvorhersagen sind über die Schritte 1 bis 4 bereits bekannt und sind nun näher zu definieren (Teilschritt (a)), zum Beispiel bezüglich ihrer Darstellungsform. Ferner ist die Auswahl möglicher Einflussfaktoren (Wartezeiten, Prioritäten) auf diese Ausgangsgrößen einzugrenzen (Teilschritt (b)). Diese Auswahl ergibt sich zum einen aus Erfahrung, zum anderen aus Gesprächen mit dem Betreiber und Entwickler sowie der Dokumentation. Der Automatisierungsgrad der Merkmalsauswahl steigt mit der Erfahrung, da die Liste der grundsätzlich für die Einflussanalyse verwendeten Merkmale vollständiger und zuverlässiger wird.

Nur Merkmale, die in einer Untersuchung einen signifikanten Einfluss zeigen, sind für eine Prognose zu verwenden. Die Einflussuntersuchung (Teilschritt (d)) kann nach einer manuellen Vorselektion automatisiert stattfinden. Vorbereitend für diesen Schritt ist eine Untersuchung der Lösungsräume (Teilschritt (c)) durchzuführen, um beispielsweise zu ermitteln, welche der möglichen Phasen verwendet werden oder zu welchen Umlaufsekunden die betrachtete SG geschaltet wurde. Alle nicht verwendeten Möglichkeiten können vom Lösungsraum ausgeschlossen werden. Auch dies kann durch eine Zählung der Ereignishäufigkeit innerhalb der Rohdaten ermittelt werden. Abhängigkeiten (Teilschritt (e)) können unter anderem durch eine Differenzierung dieser Zählung nach dem vermuteten Faktor ermittelt werden, beispielsweise die Zählung der Umlaufsekunden der Schaltzeitpunkte in Abhängigkeit zur vorangegangenen Phase. Als Konsequenz daraus kann die untersuchte Einzelvorhersage gegebenenfalls in mehrere Prognosen geteilt werden. Entsprechend ist die Architektur (Teilschritt (f)) um diese Möglichkeit zu erweitern.

Um eine Prognose zu verbessern (Schritt 7), können (a) bis (f) und, falls notwendig, 1 bis 4 iterativ wiederholt werden. Um zu bewerten, ob dies sinnvoll ist, sollte zuvor das Zusammenspiel der Einzelvorhersagen in der Gesamtprognose(-architektur) überprüft werden (Schritt 6).

4.2.5 Zwischenfazit

Die Überlegungen in Abschnitt 4.2.3 zeigen, dass sich auf der mikroskopischen Ebene für RB VA LSA-Programme in allen Fällen entsprechend der RiLSA eine SZP über die Vorhersage der Phasenfolge ergibt, auf die gegebenenfalls Prognosen von einzelnen Schaltzeiten ausgewählter SG aufgebaut werden. MB VA LSA-Programme betreffend kann es abhängig vom Anwendungsfall erforderlich sein, dieses Prognosekonzept zu erweitern, insbesondere da zu erwarten ist, dass andere Kenngrößen und Logiken für die Erfassung und Beschreibung des Verkehrszustands verwendet werden, als es für RB VA üblich ist. Zentral für das Konzept ist eine Vorhersage der Phasenfolge unabhängig vom Steuerungsverfahren. Auf den makroskopischen Ebenen sind, insofern übergeordnete Entscheidungsebenen vorhanden sind, allen mikroskopischen Prognosen Vorhersagen dieser Ebenen vorzuschalten.

Ausschließlich die Betrachtung der mikroskopischen Ebene ist entsprechend der Aufgabenstellung und des Zeitrahmens Teil dieser Arbeit. Das Konzept stellt einen Baukasten dar, der sich automatisiert selbst selektieren können soll, um eine flexibel anwendbare, individuell angepasste SZP für möglichst jede LSA anzubieten. Dies erfüllt die Anforderungen aus der Aufgabenstellung. Das Konzept adressiert zudem die Anforderungen aus Abschnitt 2.8:

- Durch die Bündelung der SG-Vorhersagen in einer Phasenfolgevorhersage werden Rechenkapazitäten und -zeit gespart, selbst bei einer nachfolgenden teilweisen Vorhersage einzelner SG, da weniger beziehungsweise nur eine Vorhersage parallel erfolgen und für Phasen mehr Informationen zur Verfügung stehen als für einzelne SG.
- Zudem folgt aus der Vorhersageabfolge, dass SG auf Basis einer Phasenfolgevorhersage getroffen werden, was die Zuverlässigkeit und Genauigkeit erhöht, da auch hier mehr Informationen über das System genutzt werden.
- Selbiges gilt für die Annäherung der Prognosearchitektur an die tatsächliche Entscheidungsarchitektur.
- Alle möglichen Arten an LSA-Programmen sind in die Überlegungen eingeflossen, wodurch das Konzept, ohne an Flexibilität, Zuverlässigkeit und Genauigkeit einzubüßen, die Möglichkeit einer flächendeckenden Verfügbarkeit bietet.

Betreffend des weiteren Vorgehens in dieser Arbeit sind die Schritte 1 bis 4 und 5(a) durch die Überlegungen zum Konzept abgedeckt. Die Schritte 5(b) bis (f) und 6 werden in den nächsten Kapiteln exemplarisch untersucht.

4.3 Implementierung

Für die exemplarische Implementierung des Konzepts wurde eine Skript-Umgebung entwickelt. Sie besteht aus drei Teilen: Datenbereinigung, Merkmalsberechnung und experimentelle Durchführung. An dieser Stelle werden lediglich die Funktion und der grobe Ablauf samt Einstellungsmöglichkeiten aufgezeigt. Alle Skripte können sowohl lokal auf einem PC als auch auf einem Rechencluster ausgeführt werden. Die Implementierung erfolgt über Python 3/Anaconda [Ana20]. Um mehrere Experimente gleichzeitig durchführen zu können, wird das Linux-Cluster der Universität Kassel [Uni20] verwendet, welches jedoch keine GPU bietet, um beispielsweise kürzere Rechenzeiten für neuronale Netze zu erzielen. Auf lokalen Systemen (PCs) werden nur Voruntersuchungen ausgeführt, da in die zur Verfügung stehenden PCs nicht ausreichend RAM für die benötigten Datenmengen für vollständige Experimente eingebaut werden kann.

4.3.1 Bereinigen der Daten

Die Bereinigung der Daten dient dazu, die Rohdaten von der Stadt Kassel zunächst soweit aufzubereiten, dass sie in Python weiter verarbeitet werden können, und zum anderen zur Bereinigung der Daten von Fehlerzuständen. Zusätzlich wird die Datenqualität ermittelt. Für die Datenbereinigung und -aufbereitung werden die Bibliotheken Pandas [pan20] und NumPy [He20] verwendet. Abbildung 4.11 zeigt den Skriptablauf. Dieser kann der Berechnung des Datenmodells aus Abbildung 4.9 entnommen werden. In Abbildung 4.10 ist es der Bereinigung und Datenqualität der Systemanalyse zuzuordnen.

Ausgehend von den **OCIT-Protokollen**, siehe Abschnitt 2.4.2, und der Eingabe des gewünschten Zeitraumes samt Feier- und Brückentagen werden die in diesen enthaltenen Daten zunächst geladen. Sie liegen als ereignisorientierte, chronologische Liste vor und sind nach LSA und Monat unterteilt. Entsprechend arbeitet das Skript die eingegebenen LSA und Monate nacheinander ab. Unvollständige Datensätze, bei denen Informationen zu einzelnen Daten wie beispielsweise SG oder Det fehlen, werden nicht berücksichtigt.

Im ersten Schritt werden nur die **Daten** für die weitere Verarbeitung **selektiert**, für die **Zeiträume** vorgegeben wurden, zum Beispiel für Dezember der Zeitraum, während dem ein „normaler Betrieb“ nach IST-Daten besteht, Schulferien werden zunächst ausgelassen. In einem zweiten Durchlauf wird der Zeitraum der Weihnachtsferien separat behandelt. Daten zu Ferien werden in eigenen Dateien gespeichert und entsprechend gekennzeichnet. Zudem werden in diesem Schritt den Datumsangaben **Wochentage zugeordnet**: 1 für Montag bis 7 für Sonntag.

Feiertage werden als Sonntage und Tage vor Feiertagen unabhängig vom tatsächlichen Wochentag als Freitage gekennzeichnet. Brückentage erhalten eine gesonderte Markierung. Hiernach werden die **Ereignisse** entsprechend ihres Ursprungs (det, SG, sen, oev, tx, IST) **aufgeteilt**, siehe 2.4.2.

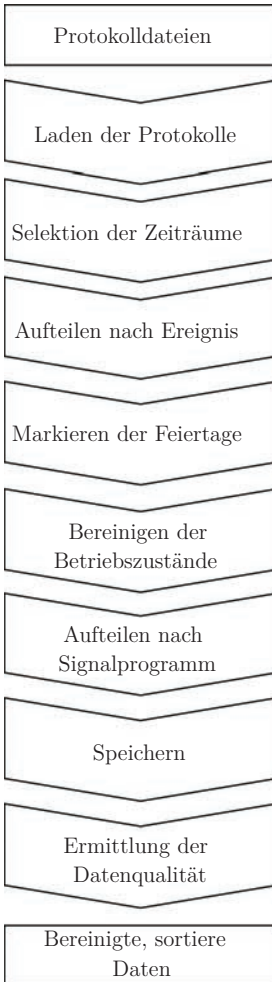


Abbildung 4.11: Ablauf der Datenbereinigung

Im nächsten Schritt werden nur Daten behalten, welche einem normalen Betrieb zuzuordnen sind. Solche, welche während eines Fehlerzustands der LSA entstanden sind, sowie Brückentage werden aus dem Datensatz entfernt. Als **fehlerfreier Betriebszustand** wird ausschließlich Zustand Nummer 5 „**Betrieb durch Zentrale**“ angesehen.

Der Zeitraum der entfernten Daten sowie der Entnahmegrund werden in einer eigenen Datei gespeichert, um zu einem späteren Zeitpunkt über die Möglichkeit zu verfügen, Zeitlücken in den Merkmalen nachvollziehen zu können.

Die nach Ereignisart sortierten und bereinigten Daten werden hiernach weiter unterteilt. Diesmal werden die Daten entsprechend der zugeordneten **Signalprogrammnummern unterschieden**, so dass in einer Liste nur Daten zu einem der vier über den Tag verteilt laufenden Signalprogramme enthalten sind.

Diese **chronologisch geordneten Listen** werden in einer entsprechenden **Datenstruktur abgelegt**. Ursprünglich wurde hierfür eine Datenbankarchitektur entworfen, welche aufgrund mangelnden Speicherplatzes in einer Datenbank zugunsten einer **Ordnerstruktur** verworfen wurde.

Zuletzt werden die bereinigten und sortierten Daten einer **Qualitätsanalyse** unterzogen, zu welcher die Ergebnisse ebenfalls abgespeichert werden, siehe Abschnitt 4.1. Die für diese Arbeit relevanten Informationen wie Betriebszustand und Programmwechsel der IST-Vektoren sind so in den Prozess zur Datenselektion und Differenzierung eingeflossen, dass sie selbst im Weiteren keine Rolle mehr spielen. Zusätzlich wird aus den IST-Vektoren eine Programmtafel erstellt, welche mit der dokumentierten verglichen werden kann.

4.3.2 Merkmalsberechnung

Die bereinigten Daten werden von der Merkmalsberechnung von ereignisorientierten Tabellen in eine Tabelle überführt, welche sekundlich den Systemzustand der LSA wieder spiegelt.

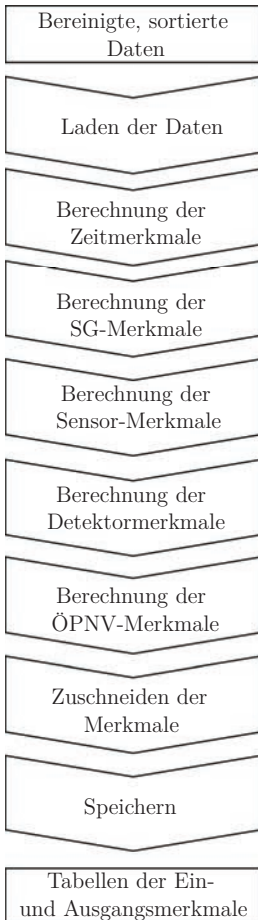


Abbildung 4.12: Ablauf der Merkmalsberechnung

Es werden alle in Abschnitt 6.1 aufgeführten Merkmale, abgesehen von der Darstellung in $t_{x,W}$, dem Belegungsgrad und Schätzungen der Progression berechnet. Die Umrechnung in $t_{x,W}$ findet nur beim Laden der Daten für die Experimente statt, da sie vom gewählten Ausgangsmerkmal abhängig ist.

Die Merkmale bilden Spalten. Die Zeilen beschreiben den sekundlich fortschreitenden Systemzustand. Der Systemzustand zu einer Sekunde ist eine Stichprobe.

Die Merkmalsberechnung ist der zentrale Schritt zur Berechnung der Eingangsdaten der ML-Algorithmen. Er kann in Abbildung 4.9 der Systemanalyse und teilweise der Zusammenstellung der Datenmodelle in Abbildung 4.10 zugeordnet werden. Das Skript zur Merkmalsberechnung ist für die Berechnung eines Signalprogramms eines Monats einer LSA vorgesehen. Sein Ablauf ist in Abbildung 4.12 dargestellt.

Zunächst werden die **bereinigten Daten geladen**. Hiernach erfolgt die **Berechnung der Merkmale, für die absoluten und relativen Zeitangaben**, wie zum Beispiel die Minuten seit Mitternacht. Auf Basis dieser Merkmale, insbesondere der Umlaufsekunde und des zugehörigen Zeitstempels, werden alle anderen Merkmale berechnet, um sekundlich den Zustand der LSA zu beschreiben.

Für die **Berechnung der SG-Merkmale** werden nur die Grünanfänge und -enden (GA/GE) verwendet und mit 1 respektive 0 codiert. Alle weiteren Ereignisse der SG, wie Gelb oder Gelb-Blinken, werden verworfen, da sie formal zu Sperren zählen. Die GA und GE werden über ihren Zeitstempel den Umlaufsekunden zugeordnet.

Für alle Umlaufsekunden, die mit diesem Schritt nicht mit einem Zustand für jede Signalgruppe versehenen wurden, werden die aktuellen Zustände extrapoliert. Hiernach werden weitere Darstellungsweisen der SG-Information für jeden Zeitstempel berechnet. Die Berechnung der Phasenmerkmale erfolgt separat, ebenso ihre Speicherung, da ihre Berechnung aufwändig ist.

Entsprechend wird für die **aggregierten Sensordaten, Detektorereignisse und ÖPNV-Meldungen** vorgegangen. Die Auswahl der zu berechnenden Merkmale können in ihren Klassenaufrufen angepasst und erweitert werden.

Für die **Detektoren** wird für die einfache Belegung der letzte bekannte Zustand sekundlich **als Merkmal** verwendet. Belegungen und ähnliche werden, sobald sie länger als die maximale Umlaufzeit der hier betrachteten Anlagen von 100 Sekunden andauern, auf 0 gesetzt, da jede wartende SG innerhalb eines Umlaufes eine Freigabe erhalten sollte.

Die **ÖPNV-Meldungen** werden nach Fahrtrichtung gebündelt. Die Informationen der letzten bekannten Meldung je Richtung werden, bis eine neue Meldung vorliegt, sekundlich wiederholt.

Vor dem Speichern der Daten in **jeweils einer Tabelle für die Ein- und Ausgangsdaten** werden die **Merkmale zugeschnitten**, das heißt, es werden die ersten und letzten 300 Zeilen sowie 300 Zeilen (ca. 3 Umläufe) vor und nach einem Zeitsprung von mehr als 5 Sekunden zwischen einer und der darauffolgenden Zeile entfernt. Dies dient dazu, Zeitpunkte zu entfernen, in denen fehlerhafte Systemzustände erfasst werden. Die 3 Umläufe wurden experimentell ermittelt. Ziel war es, möglichst wenige Daten zu entfernen, ohne die Prognosequalität zu mindern. Zudem werden in diesem Schritt nur Daten von ausgewählten Wochentagen behalten. Für das finale **Speichern der Daten** ist für den Dateinamen ein eigener String am Ende der Dateinamen vorgesehen, um die Auswahl der Merkmale zu beschreiben, zum Beispiel „full“ für: alle berechenbaren Merkmale sind enthalten.

4.3.3 Experimentelle Durchführung

Für die Durchführung der Experimente wurde ein Skript geschrieben, in das alle Einstellungen, abgesehen von den Suchräumen der Regularisierungsparameter, eingegeben werden können. Der Skriptablauf folgt dem grundsätzlichen Vorgehen des maschinellen Lernens.

Mit diesem Skript können die Schritte „vorgeschaltete Prognosen“, „SZP“ und „Ensemble“ in Abbildung 4.9 sowie die Zusammenstellung von Datenmodellen und ML-Methoden in Abbildung 4.10 durchgeführt werden. In den Skripten sind diese Schritte nicht miteinander verknüpft und müssen separat und manuell angestoßen werden. Abbildung 4.13 zeigt den Ablauf des Skriptes.

Die Ein- und Ausgangsdaten werden für eine LSA, ein Signalprogramm und einen Monat geladen. Zusätzlich können weitere Eingangsdaten zu anderen LSA und zu weiteren Merkmalen der gleichen LSA, welche nicht in den ursprünglichen **Merkmalstabellen** enthalten sind, wie zum Beispiel Phasenmerkmale, eingelesen und über einen gemeinsamen Zeitraum fusioniert werden. Des Weiteren wird nur die angegebene Anzahl an Zeilen für die gewählten Wochentage geladen.

Für die Ausgangsdaten (Zielgrößen) können Phasenmerkmale oder Merkmale einzelner Signalgruppen gewählt werden. Zu den Eingangsmerkmalen zählen immer die Umlaufsekunden, die Minuten seit Mitternacht sowie die Daten der Signalgruppe, welche vorhergesagt werden sollen. Für eine Phasenvorhersage sind die Daten von SG optional. Weitere Daten zu SG, Detektoren, aggregierten Sensordaten und ÖPNV-Informationen können über ihre Nummern ausgewählt werden. Die Wahlmöglichkeit der Formatierung ist für Ein- und Ausgangsdaten voneinander unabhängig möglich, jedoch für alle Eingangsdaten gleich.

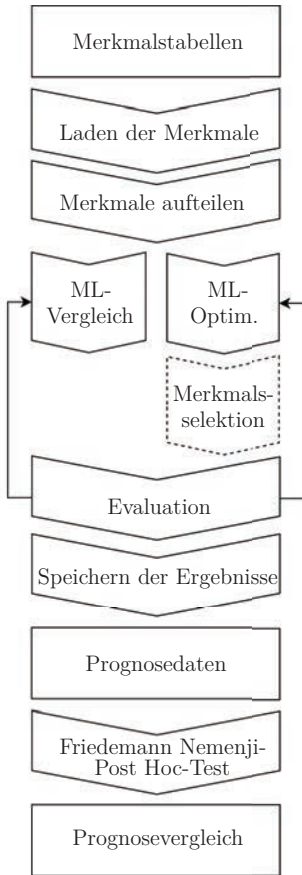


Abbildung 4.13: Ablauf der Experimente

Insofern die Formatierung in Umlaufsekunden relativ zur Ausgangsgröße ($t_{x,W}$) gewählt wurde, darf nur ein Zielgrößenvektor gewählt werden. Die zugehörigen Eingangsgrößen werden an dieser Stelle in dieses Format umgerechnet. Damit sind die gewünschten Daten geladen.

Die gewählten Daten müssen unterteilt werden, bevor sie den ML-Methoden übergeben werden können. Während der **Merkmalsaufteilung** werden zunächst die Zeitstempel entfernt, da diese aufgrund ihres Formats nicht weiter verarbeitet werden können. Hiernach werden die Daten in Trainings- und Testdaten im Verhältnis 2:1 unterteilt. Die unterteilten Daten werden aufgrund der Betrachtung mit neuronalen Netzen standardisiert. Zudem werden die Ausgangsdaten jeweils für jeden Zielgrößenvektor statistisch analysiert, das heißt Mittelwert, Median, Standardabweichung und Extrema.

Diese Daten können einem einfachen **Methodenvergleich** zugeführt werden, um einen Eindruck von der Eignung der Methoden zu gewinnen. Dies entspricht dem linken Ablauf in Abbildung 4.13. Die Regularisierungsparameter sind hierbei bei jeder darin betrachteten Methode festgesetzt. Für den Vergleich werden die **Methoden nacheinander trainiert, getestet und evaluiert**. Sobald die Liste der Methoden abgearbeitet ist, werden die Ergebnisse gebündelt gespeichert. Die Struktur der Ergebnisordner wird aus dem aktuellen Datum sowie den gewählten Einstellungen gebildet. Alle Einstellungen des Skriptes werden zusätzlich in einer eigenen Datei hinterlegt.

Alternativ zum Methodenvergleich können eine beziehungsweise mehrere Methoden gewählt werden, für die mittels Bayes-Search eine **Regularisierungsparameteroptimierung** durchgeführt wird. Dies entspricht dem rechten Ablauf in Abbildung 4.13. Hierbei wird die Methode mit verschiedenen Parametern trainiert. Die Zahl der durchgeführten Trainings ist abhängig von der Einstellung ebenso wie der Parameterraum, aus dem die getesteten Parameter ausgewählt werden. Der beste (optimierte) Regularisierungsparameter wird für jede Methode getestet und evaluiert.

Optional kann über die Eingabeparameter des Skriptes diesem Schritt eine Merkmalsselektion angeschlossen werden. Hierfür können Methoden ausgewählt werden, die lediglich einen Prozentanteil der ursprünglich der Optimierung übergebenen Merkmale beibehalten, um so nur Merkmale im Datenmodell zu erhalten, welche einen positiven Einfluss auf eine möglichst zutreffende Prognose haben.

Die *Selektionsmethoden werden iterativ getestet*, ähnlich dem Methodenvergleich. Nach einer Selektion wird die ML-Methode mit den optimierten Parametern erneut trainiert, getestet und evaluiert. Auch hiernach werden die Evaluationsergebnisse sowie die Einstellungen gebündelt gespeichert.

Für die **Evaluation** werden die Vorhersagen zu den Eingangsgrößen des Test-Datensatzes mit den Zielgrößen des Test-Datensatzes verglichen. Hierfür werden die KKR und der RMSE sowie die Gütemaße Q_{pr} , Q_{sp} und Q_{ges} verwendet. Des Weiteren werden grafische Darstellungen der Konfusionsmatrix und Ausschnitte der vorhergesagten und tatsächlichen Zielgröße gegenübergestellt. Insofern eine Merkmalsselektion stattfindet, werden die Merkmalsauswahl sowie gegebenenfalls mögliche grafische Darstellungen der Merkmalsanalyse ebenfalls gespeichert.

Um die Ergebnisse mehrerer Experimente miteinander vergleichen zu können, wurde der **Friedman-Nemenji-post-hoc-Test** implementiert, da dieser keine spezifische statistische Verteilung der Merkmale und Zielgrößen vorgibt. Für seine Durchführung werden zunächst alle Ergebnisse zu Experimenten sowie die zugehörigen Einstellungen in einer Tabelle gebündelt. Zusätzlich werden den Signalgruppen beteiligter Gruppen (mIV, FG, ÖPNV, sonstige) zugeordnet. Hierzu werden nicht nur das Endergebnis, sondern auch die Zwischenergebnisse gespeichert.

Für die meisten ML-Methoden sowie Evaluationsmetriken wurden die Implementierungen des Packages „scikit-learn“ [Pe11] verwendet. Für XGBoost wurde zudem das Package genutzt, welches die Entwickler der Methode veröffentlicht haben [CG16]. Für SARIMAX wurde das Package statsmodels [SP10] verwendet. Für das Training wurden jeweils die voreingestellten Metriken (KKR oder RMSE) verwendet und für XGBoost der RMSE. Die Klasse der MLP von scikit-learn wurde so angepasst, dass die Zahl der Schichten und darin enthaltenen Neuronen nicht über ein Tupel, sondern für maximal vier Schichten in separaten Vektoren übergeben wird, um eine Bayes'sche Regularisierungsparameteroptimierung durchführen zu können.

Algorithmen für die Verwendung von RNN, LSTM und GRU wurden mit Hilfe von Methoden aus [Pe19, Te17] zusammengestellt. Als Regularisierungsparameter stehen die Anzahl verdeckter Schichten und Neuronen, Lernrate, maximale Zahl an Epochen und die Größe der Trainingsgruppen zur Verfügung. Die Sequenzlänge entspricht der Umlauflänge. Für das Training wurden die Kreuzentropie als Fehlermetrik verwendet und Adam als Optimierer [KL15]. Die Anpassung an Nichtlinearitäten erfolgt automatisch.

Für die Merkmalsselektion wurden für PCA, RFE, χ^2 und mutual Information die Implementierungen von [Pe11] und für den Permutationstest aus Scikit-Optimize [sci20] verwendet. CFS, Fisherscore und relief wurden mittels skfeature [Le18] implementiert.

Grundsätzlich wurden die Methoden, welche Rechenzeit durch die Nutzung einer GPU sparen könnten, über die Programmierschnittstelle CUDA [VF20] implementiert. Das Cluster verfügte über keine GPU und die verfügbaren PCs verfügten nicht über ausreichend RAM, weshalb diese Möglichkeit nicht verwendet wurde. Betroffen sind hiervon nur auf Neuronen basierende Methoden sowie bestimmte Optionen für XGBoost, welche vom Package-eigenen Algorithmus automatisiert ausgewählt wurden. Dieser wählte diese für den hier betrachteten Anwendungsfall jedoch nicht. Alle anderen betrachteten Methoden waren nicht so implementiert, dass CUDA verwendet werden konnte.

4.4 Bewertungsgrößen von Schaltzeitprognosen

Genauigkeit und Zuverlässigkeit (Eintretenswahrscheinlichkeit) spielen eine wesentliche Rolle für die Akzeptanz von SZP, siehe auch Abschnitt 2.8. Sie werden maßgeblich von der Parametrierung der ML-Algorithmen bestimmt. Der Prognosehorizont ergibt sich indirekt aus diesen, da der Prognosehorizont über ein Fallen von Genauigkeit und Zuverlässigkeit unter eine gewählte Grenze festgelegt werden kann. Einen flächendeckenden Einsatz von SZP wird es nur geben, wenn sich das Konzept auf viele Anwendungsfälle/Knotenpunkte übertragen lässt. Dies bleibt für zukünftige Forschungsvorhaben offen und lässt sich für diese Arbeit lediglich schätzen.

Rückschlüsse auf die Echtzeitfähigkeit können ebenfalls nur in geringem Maße aus Rechenzeiten für Prognosen abgeleitet werden. Sie hängt hauptsächlich von der konkreten Umsetzung der Automatisierung ab, welche in dieser Arbeit nicht vorgenommen, sondern nur argumentativ ergründet wird, sowie von der individuellen Systemimplementierung.

Der Umfang der betrachteten Daten ist, wie in [Wei17] gezeigt, bereits für den Zeitraum einer Woche ausreichend, um zumeist alle in der Realität auftretenden Fälle zu enthalten. Es kann auch davon ausgegangen werden, dass die darin auftretenden Häufigkeiten der verschiedenen Klassen repräsentativ sind, da für eine normale Woche außerhalb der Urlaubszeit ohne Feier- und Brückentage die Abläufe wiederkehrend sind.

Sowohl Protschky [Pro16] als auch Weisheit [Wei17] verwenden als Qualitätsmetriken für die Genauigkeit ihrer Prognosen die KKR zum Teil mit Konfusionsmatrizen, jedoch ohne Begründung. Diese Wahl wird von der Problemstellung bestätigt: Für eine SZP ist die korrekte Vorhersage einer Freigabe (richtig pos.) ebenso wichtig wie die einer Sperrung (richtig neg.) eines Verkehrsstroms, da für beides (GA und GE) die Genauigkeit wichtig ist, um eine gegebenenfalls angemessene Reaktion zeigen zu können. Mögliche Reaktionen sind beispielsweise „auf Halt vorbereiten“ und „mit gleichmäßiger Geschwindigkeit weiterfahren“. Damit entfallen Sensitivität, Relevanz und f_1 -Maß als Qualitätsmetriken für SZP.

Um eine Vergleichbarkeit der Ergebnisse zu gewährleisten, wird für diese Arbeit ebenfalls die KKR mit Konfusionsmatrizen verwendet. Zusätzlich werden die Ergebnisse über den RMSE bewertet, da dieser für die nicht korrekt klassifizierten Zielgrößen ein Maß der Nähe zur gewünschten Klasse bietet.

Für das Projekt VERONIKA [Uni19] wurden von der Firma TTS zusätzlich speziell entwickelte Qualitätsmetriken, welche Zeitreihen analysieren, für die Genauigkeit verwendet. Diese können nur in begrenztem Maß für die Parametrierung von ML-Algorithmen verwendet werden, da die Implementierungen von ML-Algorithmen zum größten Teil nur eigene vorab implementierte Metriken für die Verlustfunktionen vorsehen. Zudem wird eine Parametrierung bis auf wenige Ausnahmen mit einem randomisierten Datensatz durchgeführt, was eine Bewertung von vorangegangenen Prognosen abhängigem Verlust innerhalb einer Zeitreihe verhindert. Solche Verlustfunktionen können beispielsweise für rekurrente neuronale Netze verwendet werden. Für die Qualitätsbewertung bereits parametrisierter Algorithmen ist diese Qualitätsmetrik jedoch aussagekräftig und wird daher zusätzlich hinzugezogen.

Protschky [Pro16] berechnet auf Basis eines „typischen Verlaufs einer Wahrscheinlichkeitskurve (Freigabe- oder Grünwahrscheinlichkeit)“ die Grünwahrscheinlichkeit. Dort fließt die aktuelle Verfügbarkeit ein, da nur korrekte Klassifikationen mit Hilfe eines Schwellwerts als verfügbar gelten. Die zugrunde gelegte Wahrscheinlichkeitskurve (sigmoidal fallend für GE und steigend für GA) ist für VA-LSA jedoch nur ein möglicher Fall, da bei einer Veränderung der Eingangssignale eine Entscheidung für eine Freigabe spontan verworfen werden kann, wodurch die Wahrscheinlichkeitskurven für GA und GE unstetig sind und spontan auf einen beliebigen unbekanntem Wert zwischen 0 und 1 springen können. In [Wei17] wird die Zuverlässigkeit bezugnehmend auf die relative Häufigkeit innerhalb der Stichprobe berechnet. In dieser Arbeit könnte die Eintretenswahrscheinlichkeit mit Hilfe von ML-Algorithmen ermittelt werden, da es sich um eine typische Anforderung für Prädiktionen mit ML-Algorithmen handelt. Zudem bieten diese Methoden aufgrund ihrer Standardisierung eine Verlässlichkeit, woran es allen oben genannten Wahrscheinlichkeitsberechnungen aus den genannten Gründen mangelt. Es werden andere Experimente zunächst priorisiert.

Kapitel 5

Experimente zur Methodenwahl der signalgruppenorientierten Vorhersage

Dieses Kapitel dokumentiert die Auswahl der ML-Methoden für das in Kapitel 4 vorgestellte Verfahren zur SZP. Die fünfte Forschungsfrage aus Abschnitt 1.2 bezüglich der Eignung verschiedener ML-Methoden für SZP soll hiermit beantwortet werden. Das Vorgehen und die Ergebnisse, die in diesem Kapitel vorgestellt werden, stehen in Wechselwirkung zu denen in Kapitel 6.

Zunächst werden die Auswahl der Methoden in Abschnitt 5.1 und das Vorgehen zur Durchführung der Experimente in Abschnitt 5.2 vorgestellt. Im folgenden Abschnitt finden sich die zugehörigen Ergebnisse. Anschließend wird in Abschnitt 5.4 ein Ansatz zur Beantwortung der siebten Forschungsfrage vorgestellt, ob sich Verbesserungen verglichen zu anderen Verfahren zur SZP ergeben.

5.1 Grundlegende Auswahl der ML-Methoden

Grundsätzlich handelt es sich bei einer SZP um eine Klassifikation, da die vorherzusagenden diskreten Schaltzeitpunkte einen ganzzahligen begrenzten Lösungsraum darstellen. Der Suchraum der Zielgrößen kann mehr als hundert mögliche Klassen haben, weshalb Regressionen zu qualitativ hochwertigeren Ergebnissen führen könnten. Deshalb wurden, wenn unterschiedliche Implementierungen für Klassifikation und Regression zur Verfügung standen, Klassifikation und Regression getestet. Dies betrifft 1. bis 7. der folgenden Aufzählung. Die Ergebnisse der Regressoren werden entsprechend auf ganze Zahlen gerundet. Da die Daten in chronologischer Reihenfolge vorliegen und Zeitpunkte vorherge-

sagt werden sollen, handelt es sich um ein Problem aus dem Bereich der Vorhersage von Zeitreihen.

Die ausgewählten ML-Methoden sind:

1. k Nearest Neighbors (kNN),
2. Support Vector Machines (SVM) (Radialbasisfunktion RBF γ und linear l kernel),
3. Decision Trees (DT),
4. Random Forests (RF),
5. AdaBoost (Ada),
6. EXtreme Gradient Boosting (XGB),
7. Multi Layer Perceptren (MLP),
8. Recurrent Neural Networks (RNN),
9. Long Short Term Memory neural networks (LSTM),
10. Gated Recurrent Unit neural networks (GRU),
11. Naive Bayes (NB) und
12. Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model (SARIMAX, SAR).

Überlegungen, die der Auswahl zugrunde liegen, sind folgende: Aus jedem Bereich der Denkansätze zu Methoden des maschinellen Lernens wird mindestens eine Methode ausgewählt: Entscheidungsbäume, Instanz-basiert, Neuronale Netze, Stochastik und Ensemble. Zudem werden die Popularität der Methoden in der Fachwelt, welche über Gespräche mit Experten und Internetrecherche in einschlägigen Foren ermittelt wurde, herangezogen. Des Weiteren wurde allgemein die Popularität von Methoden in den Medien und der wissenschaftlichen Welt allgemein sowie der Erfolg bei ML-/KI-Wettbewerben berücksichtigt. Entsprechend begründet sich die Auswahl wie folgt:

Unter künstlicher Intelligenz beziehungsweise maschinellem Lernen in den Medien und Wissenschaften werden zumeist neuronale Netze wie MLP sowie rekurrente Netze wie zum Beispiel Recurrent Neural Network (RNN), LSTM und GRU verstanden.

Bei Wettbewerben (zum Beispiel KAGGLE) respektive Veröffentlichungen zu Methodenvergleichen, wo auch die oben genannten populären Methoden betrachtet werden, sind zum einen Ensemblemethoden wie XGBoost respektive klassische statistische Methoden wie (S)ARIMA(X) besonders erfolgreich gewesen [Nie16, Ae10, Me18b, Me18a]. Für Methodenvergleiche werden in der Regel eine hohe Zahl an Anwendungsfällen betrachtet.

Auf Entscheidungsbäumen basierende Modelle werden zudem in Betracht gezogen, weil die Vermutung nahe liegt, dass sie aufgrund ihrer strukturellen Ähnlichkeit mit der Entscheidungslogik regelbasierter LSA diese besonders gut abbilden können. Deshalb werden zusätzlich zu XGBoost, einfache DT und RF in Betracht gezogen.

Neben XGB wird Ada als Ensemblemethode betrachtet. Sie kann außer mit Entscheidungsbäumen auch mit SVM arbeiten.

Für die Instanz-basierten Methoden wurden kNN und SVM als klassische Beispiele ausgewählt. Als zusätzliche statistische Methode wird zusätzlich zu (S)ARIMA(X) NB berücksichtigt, da VA LSA mit RB Logiken anhand von Bedingungen entscheiden.

5.2 Versuchsaufbau

Neben der zentralen Frage, welche maschinelle Lernmethode am besten für die Vorhersage von Signalzeiten geeignet ist, stellen sich für die hier beschriebenen Experimente weitere Fragen.

- Sollte das Problem aus praktischer Sicht als eines für Klassifikatoren oder Regressionen betrachtet werden?
- Können die Ergebnisse der Methoden über die Fortbewegungsarten der Personengruppen (mIV, FG, ÖPNV, Rad) unterschieden werden?
- Gibt es einen Unterschied zwischen Grünbeginn und -ende für die Prognosequalität?
- Führen die Ergebnisse der verschiedenen LSA und SP zu den gleichen Schlussfolgerungen?

5.2.1 Vorgehen zum Vergleich der Methoden

Zunächst wurden die oben genannten Methoden mit festen Hyperparametern betrachtet. Ihre Einstellung ist in Anhang F.7 beschrieben. Auf eine vorherige Hyperparameteroptimierung wurde aufgrund der dafür benötigten Rechenzeit, welche das verfügbare Maß überstieg, nicht vorgenommen.

Als Bewertungsparameter dienten RMSE und KKR. Die Experimente wurden mittels Friedman-Nemenji-post-hoc-Test verglichen und auf signifikante Unterschiede zwischen der Performanz der Methoden untersucht.

Die signifikant besten und schlechtesten Methoden wurden einer Optimierung der Hyperparameter unterzogen und erneut verglichen. Die Optimierung der Hyperparameter wurde mittels Baye'scher Suche automatisiert.

5.2.2 Sonstige Einstellungen der Experimente

Die Experimente basieren auf den in Abschnitt 6.7 beschriebenen Datenmodellen insbesondere SG_s^n und SG^n . Zielgrößen sind wie für Kapitel 6 $SG_{s,GA/GE} \forall s \in S$. Die Experimente wurden ebenfalls wie für Kapitel 6 für die drei SP, die tagsüber laufen, durchgeführt.

Es wurden, wenn nicht explizit anders beschrieben, ca. 100.000 Stichproben (Sekunden) der Wochentage Dienstag, Mittwoch und Donnerstag betrachtet. Die Aufteilung zwischen Test- und Trainingsdaten beträgt 1:2. Die Daten wurden mathematisch standardisiert (Normalverteilung), um den betrachteten neuronalen Netzen keinen Nachteil zu verursachen.

Für die Hyperparameteroptimierung wurden jeweils 150 Suchpunkte (`n_iter`) verwendet. Die für die Methoden gewählten Suchräume (search space) der Hyperparameter sind im Anhang F zu finden.

5.3 Ergebnisse für die Datenmodelle SG_s und SG

Die Ergebnisse für die Datenmodelle SG_s (eigene Signalgruppe) und SG (alle Signalgruppen) für KP Katzensprung (002) und KP Drusteltalstraße (722) führen zu den gleichen Erkenntnissen, weshalb an dieser Stelle die Ergebnisse der Drusteltalstraße repräsentativ diskutiert werden.

Die Ergebnisse für KP 722, SP 2 für SG_s sind für ausgewählte Zielgrößen / Signalgruppen in Tabelle 5.1 zu sehen. Die besten Ergebnisse sind grün und die schlechtesten rot hinterlegt. Weitere ähnlich gute respektive schlechte Ergebnisse sind für die einzelnen SG grün respektive rot gedruckt. Weitere Ergebnisse sowie die Ergebnisse für den Katzensprung sind im Anhang in Abschnitt G zu finden.

Die in der Tabelle aufgeführten Methoden sind absteigend nach der KKR von SG_3 sortiert. Es ist zu sehen, dass diese Sortierung im Groben auf die KKR der anderen SG und den RMSE übertragbar ist. Die höchsten KKR sowie niedrigsten RMSE werden von XGB, RF und DT erzielt, auch SVM und kNN zeigen zum Teil vergleichbare Ergebnisse. Zumeist ist die Klassifikation mit XGB führend bezüglich der Prognosequalität. Die letzten Tabellenplätze werden in der Regel von SAR, MLP, GRU, LSTM, RNN und NB in unterschiedlicher Reihenfolge angenommen. Der Qualitätsunterschied zwischen diesen beiden Gruppen der besten und schlechtesten ML-Methoden zeigt sich für die GE in Tabelle 5.1 besonders deutlich durch die teilweise einstellige KKR, siehe auch Tabellen G.1 bis G.6 im Anhang.

Die Prognosequalität unterscheidet sich zwischen den SG. SG_3 und SG_5 signalisieren geradeaus führende Verkehrsströme der Haupt- respektive Nebenrichtung und SG_7 einen vergleichsweise gering belasteten von der Hauptrichtung linksabiegenden Verkehrsweg, siehe Anhang B.1. Die vergleichsweise geringe Verkehrsbelastung von SG_7 kann eine mögliche Erklärung für ihre hohen RSME sein, siehe auch 6.4. SG_{13} und SG_{14} signalisieren einen Überweg für FG parallel zur Hauptrichtung (SG_3), sie werden mit der Hauptrichtung gemeinsam geschaltet, weshalb sie gut vorhergesagt werden können. Der Radverkehr wird beispielsweise mit SG_{24} , dessen Verkehrsstrom SG_5 entgegen gerichtet ist, signalisiert.

Tabelle 5.1: Ausgewählte Ergebnisse SG_s , GA obere Hälfte, GE untere, KP 722, Monat Sep., SP 2 (K = Klassifikation, R = Regression)

Methode	KKR [%]						RMSE [s]					
	3	5	7	13	14	24	3	5	7	13	14	24
K XGB	11	20	9	9	9	22	12	9	1385	15	16	8
K RF	7	15	6	6	6	18	10	9	1385	16	16	8
K DT	6	8	5	5	5	9	11	10	1384	16	16	8
K kNN	6	7	3	4	4	10	17	16	1386	23	23	14
K SVM _r	6	9	5	5	5	8	18	14	1385	22	22	13
K SVM _l	5	6	3	4	4	6	30	33	1385	34	34	33
R DT	5	8	3	5	5	9	9	9	1384	15	15	8
R RF	5	12	3	5	5	14	9	9	1384	15	15	8
K Ada	4	4	2	4	3	4	24	15	1385	20	23	13
R Ada	4	6	0	4	4	5	12	10	1252	22	22	10
R SVM _l	4	7	2	4	4	10	13	13	1385	19	19	12
R SVM _r	4	8	4	4	4	10	13	13	1385	17	17	12
R XGB	4	7	3	3	3	9	10	10	1385	16	16	9
K RNN	3	3	2	1	3	3	28	33	1385	33	30	23
R kNN	3	4	2	4	4	5	12	11	1385	19	19	11
K LSTM	2	1	1	1	1	3	35	38	1256	31	35	33
R NB	2	5	2	2	2	5	22	32	1388	58	59	27
K GRU	1	3	2	2	2	1	32	28	1387	28	30	29
K MLP	1	1	1	1	1	1	38	27	1254	36	28	38
R MLP	1	4	2	4	4	4	26	14	1253	15	15	13
K SAR		0	0		0			9e18	2.5e9		3e12	
K XGB	43	22	8	43	43	21	5	10	1253	10	10	10
R SVM _l	42	5	3	42	42	5	11	13	1253	15	15	13
K RF	27	11	6	25	25	11	5	13	1253	10	10	10
R RF	25	8	5	16	15	8	5	9	1252	9	9	8
R SVM _r	24	10	3	22	22	7	11	13	1253	14	14	13
K kNN	16	8	5	12	12	9	12	15	1254	18	18	14
R XGB	15	7	4	13	14	7	5	10	1253	11	11	10
R DT	14	8	4	11	11	7	5	9	1252	10	10	9
K SVM _r	13	7	5	14	14	9	12	17	1253	14	14	18
R kNN	13	7	5	13	13	6	8	10	1253	15	15	10
K DT	12	10	6	12	12	9	5	10	1252	10	10	9
K SVM _l	7	6	4	6	6	6	22	28	1254	25	25	27
K RNN	7	1	1	1	6	2	15	35	1254	51	24	22
R NB	6	4	4	3	3	5	14	15	1255	46	47	15
K Ada	4	4	4	5	5	4	15	17	1252	16	16	16
R MLP	2	5	1	1	3	2	18	13	1385	29	19	22
K GRU	2	3	2	2	2	3	29	28	1256	25	34	26
K LSTM	1	2	1	1	1	2	30	30	1387	33	32	29
K MLP	1	1	1	1	1	1	30	32	1386	30	30	27
K SAR		0	0	0	0	0	0.4e9	2e12	2e12	2e12	1e18	1e18

Für KP 002 wurde eine vergleichbare Auswahl an Zielgrößen getroffen: in Hauptrichtung SG_5 geradeaus und SG_7 links abbiegend, in Nebenrichtung SG_3 geradeaus und SG_4 links abbiegend, SG_{13} Überweg für FG parallel zur Hauptrichtung und SG_{20} ÖPNV, siehe G.1.

Detailliertere Erläuterungen der Prognosequalität für XGB, ausdifferenziert nach Zielgrößen, sind in Abschnitt 6.4 dargestellt. GA scheinen im Allgemeinen schwieriger vorherzusagen zu sein als GE, was auf die Begrenzung der maximalen Freigabezeitdauer während der Hauptverkehrszeit zurückzuführen ist, welche für Hauptrichtungen häufig auch während schwächerer Verkehrsbelastungen in Kassel Anwendung finden. Sie führen zu einer vergleichsweise konstanten Grünzeitdauer, aber nicht immer zu ähnlich regelmäßigen GA.

Für den Friedman-Nemenji-post-hoc-Test, welcher die zentralen Erkenntnisse zur Auswahl der ML-Methoden herausstellt, wurden die Ergebnisse aller SG und SP zusammenfassend für jeweils einen KP betrachtet. Die Tabellen 5.2 und 5.3 zeigen die Ergebnisse des Friedman-Nemenji-post-hoc-Test ausgewertet für RMSE, KKR und Q_{pr} . Die LSTM stehen repräsentativ für die Ergebnisse mit RNN und GRU. SVM_r erzielten zumeist bessere Ergebnisse als SVM_l . Ebenso verhält es sich mit der Klassifikation durch Ada im Vergleich zur Regression durch Ada.

In den Ergebnistabellen zu Friedman-Nemenji-post-hoc-Signifikanztests sind Werte der Übereinstimmung zwischen den Ergebnissen der einzelnen Methoden angegeben. Eine übereinstimmende Prognosequalität mit dem entsprechenden Evaluationsparameter liegt bei einem Wert von 0 vor und keine Übereinstimmung bei einem Wert von 1.

Tabelle 5.2: Signifikanzen des Friedman-Nemenji-post-hoc-Test SG_s für den RMSE mit KP 722, Monat Sep., SP 2 und 3, gekürzt, Konfidenzniveau 5% (K = Klassifikation, R = Regression)

Methode	Klassifikation										Regression					
	K Ada	K DT	K LSTM	K MLP	K kNN	K SVM_r	K RF	K XGB	R NB	R MLP	R kNN	R SVM_r	R RF	R XGB	SAR	
K Ada	0,00	0,01	0,36	0,13	0,90	0,90	0,01	1	0,17	0,90	0,47	0,65	1	0,05	1	
K DT	0,01	0,00	1	1	0,10	0,02	0,90	0,90	1	1	0,90	0,90	0,44	0,90	1	
K LSTM	0,36	1	0,00	0,90	0,07	0,26	1	1	0,90	0,86	1	1	1	1	0,06	
K MLP	0,13	1	0,90	0,00	0,02	0,09	1	1	0,90	0,58	1	1	1	1	0,20	
K kNN	0,90	0,10	0,07	0,02	0,00	0,90	0,11	0,01	0,02	0,90	0,90	0,90	1	0,29	1	
K SVM_r	0,90	0,02	0,26	0,09	0,90	0,00	0,02	0,00	0,12	0,90	0,58	0,75	1	0,08	1	
K RF	0,01	0,90	1	1	0,11	0,02	0,00	0,90	1	1	0,90	0,90	0,41	0,90	1	
K XGB	1	0,90	1	1	0,01	0,00	0,90	0,00	1	1	0,90	0,76	0,88	0,90	1	
R NB	0,17	1	0,90	0,90	0,02	0,12	1	1	0,00	0,65	1	1	1	1	0,15	
R MLP	0,90	1	0,86	0,58	0,90	0,90	1	1	0,65	0,00	0,08	0,18	1	0,00	1	
R kNN	0,47	0,90	1	1	0,90	0,58	0,90	0,90	1	0,08	0,00	0,90	0,01	0,90	1	
R SVM_r	0,65	0,90	1	1	0,90	0,75	0,90	0,76	1	0,18	0,90	0,00	0,00	0,90	1	
R RF	1	0,44	1	1	1	1	0,41	0,88	1	1	0,01	0,00	0,00	0,18	1	
R XGB	0,05	0,90	1	1	0,29	0,08	0,90	0,90	1	0,00	0,90	0,90	0,18	0,00	1	
SAR	1	1	0,06	0,20	1	1	1	1	0,15	1	1	1	1	1	0,00	

Für einen signifikanten Unterschied innerhalb dieser Skala sind Werte von über 95 % zu erreichen. Im Fall der in diesem Abschnitt sowie dem zugehörigen Anhang gezeigten Signifikanztests wurde, wenn ein signifikanter Unterschied ermittelt werden konnte, eine 1 ermittelt. Deshalb werden diese durch das Weglassen der Nachkommastellen hervorgehoben.

Die Friedman-Nemenji-post-hoc-Signifikanztests, siehe Tabellen 5.2 und 5.3, bestätigen die für Tabelle 5.1 diskutierten Beobachtungen. Die Klassifikation mit XGB ist für alle drei Evaluationsmaße als beste Methode signifikant besser, siehe vorangegangener Absatz, als Ada, LSTM, MLP, NB und SAR. Ebenso verhält es sich mit DT und RF gegenüber diesen Methoden, welche die gleichen signifikanten Unterschiede aufweisen außer für Ada. Für SVM und kNN, aber auch DT und RF befindet sich der Unterschied mit 90 % zu XGB häufig in der Nähe des Konfidenzbereichs und sie sind damit zumeist nicht vergleichbar gut für Prognosen geeignet wie XGB. Die genaue Ausprägung dieses Qualitätsunterschieds ist vom Evaluationsmaß und KP abhängig, siehe auch Tabellen G.7 bis G.16. Entsprechend ist davon auszugehen, dass ebenfalls eine Abhängigkeit zur Zielgröße besteht und damit einzelne Methoden in einzelnen Fällen mit XGB in der Prognosequalität konkurrieren können, diese jedoch nicht maßgeblich übertreffen. SAR erzielt signifikant schlechtere Ergebnisse als die meisten Methoden.

Damit zeigt XGB in diesen Experimenten die beste Prognosequalität und wird damit im Folgenden als beste Methode für SZP betrachtet. Es schließt sich eine Analyse des Prognoseverhaltens der verschiedenen Methoden an. Es konnte kein Unterschied zwischen den Gruppierungen der SG nach Fortbewegungsmittel (mIV, ÖPNV, FG, Rad) der LSA festgestellt werden, weshalb hierzu keine weiteren Ergebnisse dargestellt sind.

Tabelle 5.3: Signifikanzen des Friedman-Nemenji-post-hoc-Test SG_s für die KKR mit KP 722, Monat Sep., SP 2, 3, gekürzt, Konfidenzniveau 5% (K = Klassifikation, R = Regression)

Methode	K Ada	K DT	K LSTM	K MLP	K kNN	K SVM _r	K RF	K XGB	R NB	R MLP	R kNN	R SVM _r	R RF	R XGB	SAR
	C Ada	0,00	1	0,00	0,01	0,01	0,02	1	1	0,90	0,05	0,90	0,32	0,00	0,90
K DT	1	0,00	1	1	1	0,90	0,90	0,56	1	1	0,19	0,87	0,90	0,02	1
K LSTM	0,00	1	0,00	0,90	1	1	1	1	0,00	0,90	1	1	1	1	0,90
K MLP	0,01	1	0,90	0,00	1	1	1	1	0,01	0,90	1	1	1	1	0,90
K kNN	0,01	0,90	1	1	0,00	0,90	0,44	0,06	0,02	1	0,85	0,90	0,90	0,31	1
K SVM _r	0,02	0,90	1	1	0,90	0,00	0,32	0,03	0,03	1	0,90	0,90	0,90	0,43	1
K RF	1	0,90	1	1	0,44	0,32	0,00	0,90	1	1	1	0,02	0,72	1	1
K XGB	1	0,56	1	1	0,06	0,03	0,90	0,00	1	1	1	1	0,19	1	1
R NB	0,90	1	0,00	0,01	0,02	0,03	1	1	0,00	0,04	0,90	0,35	0,00	0,90	1
R MLP	0,05	1	0,90	0,90	1	1	1	1	0,04	0,00	1	1	1	1	0,74
R kNN	0,90	0,19	1	1	0,85	0,90	1	1	0,90	1	0,00	0,90	0,57	0,90	1
R SVM _r	0,32	0,87	1	1	0,90	0,90	0,02	1	0,35	1	0,90	0,00	0,90	0,90	1
R RF	0,00	0,90	1	1	0,90	0,90	0,72	0,19	0,00	1	0,57	0,90	0,00	0,11	1
R XGB	0,90	0,02	1	1	0,31	0,43	1	1	0,90	1	0,90	0,90	0,11	0,00	1
SAR	1	1	0,90	0,90	1	1	1	1	1	0,74	1	1	1	1	0,00

Die Abbildungen 5.1, 5.2 und 5.3 veranschaulichen die Prognosequalität für die betrachteten ML-Methoden mittels einer relativ skalierten Rangfolge der ML-Methoden, vergleiche mit den Abbildungen G.1 bis G.9. Die Abstände auf dem oberen Balken veranschaulichen den Qualitätsunterschied. Die Reihenfolge ist zwischen KKR und RMSE beziehungsweise Q_{Pr} invers, da die KKR mit der Qualität steigt, während RMSE und Q_{Pr} sinken. Die Klassifikation mit XGB befindet sich in der Rangordnung für alle Ergebnisse zumeist an der Spitze. Wenn dies nicht der Fall ist, befindet sich die Klassifikation mittels XGB unter den besten drei ML-Methoden. Teilweise vergleichbare Qualitäten werden mit RF und DT sowie vereinzelt mit SVM und kNN erzielt. Keine Methode hat in einer Analyse signifikant bessere Ergebnisse erzielt als XGB.

Die Abbildungen 5.4 bis 5.13 zeigen die Konfusionsmatrizen für SG 24 (GE) ausgewählter ML-Methoden. Die verschiedenen Prognosequalitäten sind durch typische Muster gekennzeichnet. In Konfusionsmatrizen werden Prognose und Zielgröße vergleichend einander gegenübergestellt. Die Farbskala gibt die Häufigkeit der Prognose-Zielgröße-Kombination an. Für jeden Wert der Zielgröße wird ermittelt, wie häufig welcher Wert (welche Klasse im Lösungsraum) als Prognose für diesen ausgegeben wurde. Da der Lösungsraum für diese Experimente mehr als 120 Klassen umfasst, wurden zwecks Lesbarkeit keine Zahlen sondern eine Farbskala zur Veranschaulichung der Häufigkeiten verwendet. Wenn Zielgröße und Prognose immer übereinstimmen, dann ergibt sich für ein Experiment eine gleichmäßig gefärbte Diagonale von oben links nach unten rechts, welche sich in der Farbintensität am rechten unteren Ende leicht abschwächt, da diese Werte im Lösungsraum vergleichsweise weniger häufig auftreten. Je weiter die Prognosen von der Zielgröße und damit Diagonale entfernt sind, desto größer ist die Abweichung zwischen Zielgröße und Prognose und damit schlechter die Prognosequalität.

Abbildung 5.4 zeigt eine SZP mit XGB. Die Diagonale ist deutlich zu sehen, jedoch werden GE, welche mehr als 20 s in der Zukunft liegen, mit einer größeren Unsicherheit vorhergesagt als zeitlich näher liegende GE, was die Parallelen zur Diagonale zeigen. Sie weisen eine erhöhte Unsicherheit in diesem Bereich auf, siehe Abbildung G.16. Es ist zu beobachten, dass für die Prognose der gesamte Lösungsraum ähnlich gleichmäßig genutzt wird wie von der Zielgröße. Zu sehen ist dies daran, dass es über die Breite der Matrix eine gleichmäßige Farbverteilung ohne Unterbrechungen gibt, wie es beispielsweise für Ada oder DT der Fall ist, siehe Abbildungen 5.6, 5.10 und 5.12.

Die Ergebnisse für Ada und DT (Abbildungen 5.6, 5.10 und 5.12) zeigen eine deutliche Priorisierung einzelner Klassen des Lösungsraumes auf, wodurch vertikale Streifen in der Matrix zu sehen sind. Dies lässt sich durch zu gering ausdifferenzierte Entscheidungsbäume, also zu wenige Zweige beziehungsweise trennende Ebenen für SVM erklären.

Der Lösungsraum wird hierdurch ungewollt gruppiert und jeweils ein Wert der Gruppe priorisiert, wodurch die anderen Werte der Gruppe nicht mehr vorhergesagt werden können. Entsprechend wird gefolgert, dass die betrachteten Modelle nicht umfangreich genug sind, um die Komplexität der LSA-Logiken abbilden zu können.

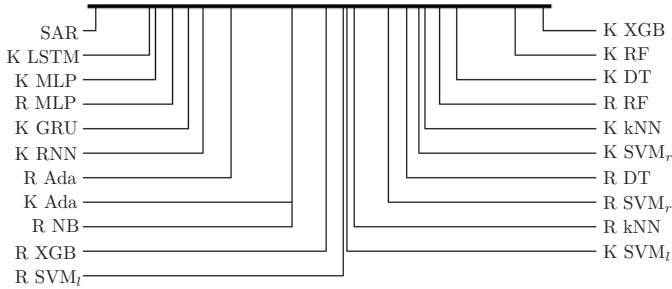


Abbildung 5.1: Relatives Ranking des Friedman-Nemenji-post-hoc-Test für den KKR, KP 722, SG_s , SP 2, 3

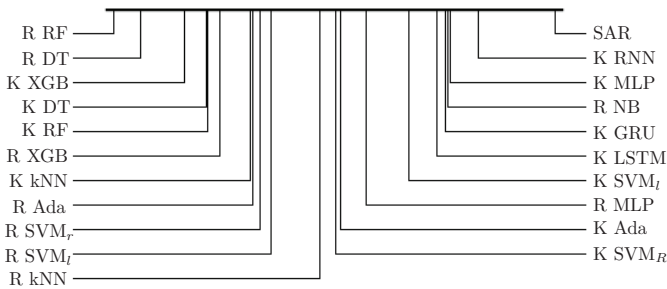


Abbildung 5.2: Relatives Ranking des Friedman-Nemenji-post-hoc-Test für den RMSE, KP 722, SG_s , SP 2, 3

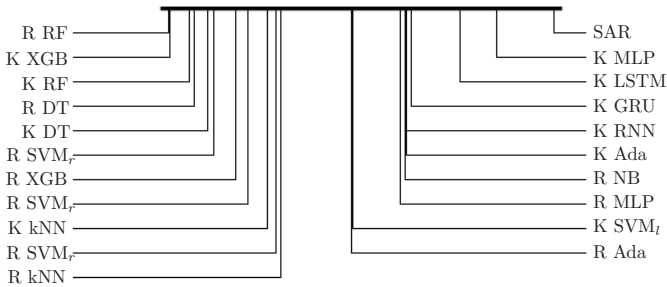


Abbildung 5.3: Relatives Ranking des Friedman-Nemenji-post-hoc-Test für Q_{PR} , KP 722, SG_s , SP 2, 3

Eine besonders gleichmäßige Gruppierung mit für die Gruppen zutreffender Vorhersage ist in Abbildung G.19 zu sehen. Ein ähnliches Verhalten zeigen auch die LSTM, wobei hierbei zudem die Gruppen nicht korrekt getroffen werden, was die Länge der vertikalen Streifen über die gesamte Matrixhöhe verdeutlichen, siehe Abbildung 5.11. Wenn sich die Prognosen der Zielgröße noch gleichmäßiger über den Lösungsraum verteilen, dann schwindet der farbliche Kontrast in der Matrixdarstellung fast vollständig. Deshalb können die Ergebnisse alleine auf Basis der Konfusionsmatrix als unzureichend bewertet werden, siehe Abbildung 5.13. Hyperparameteroptimierungen konnten unbefriedigende Ergebnisse nicht signifikant verbessern.

Die Verwendung von Klassifikation hat gegenüber der Verwendung von Regression in fast allen Fällen eine bessere Prognosequalität gezeigt. Die Experimente am Katzensprung bestätigen die hier vorgestellten Erkenntnisse für die Druseltalstraße, siehe Abbildungen G.10 bis G.27. Die beschriebenen Muster sind methodenspezifisch in den Experimenten zu beobachten.

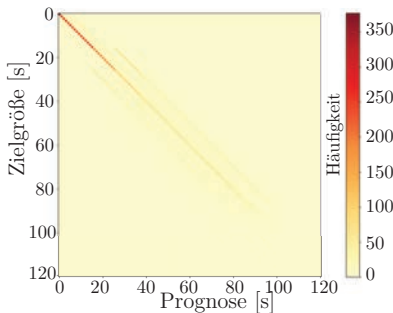


Abbildung 5.4: Konfusionsmatrix für SG 24
GE K XGB, KP 722, SG , SP 2

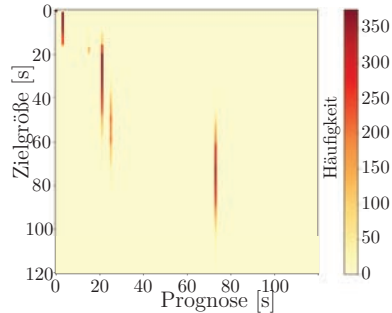


Abbildung 5.5: Konfusionsmatrix für SG 24
GE K Ada, KP 722, SG , SP 2

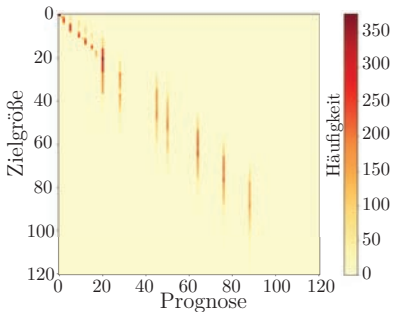


Abbildung 5.6: Konfusionsmatrix für SG 24
GE K DT, KP 722, SG , SP 2

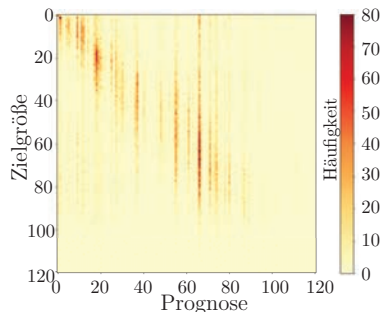


Abbildung 5.7: Konfusionsmatrix für SG 24
GE K LSTM, KP 722, SG , SP 2

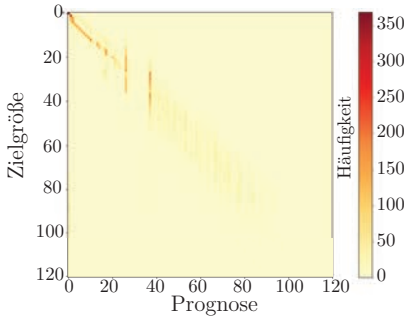


Abbildung 5.8: Konfusionsmatrix für SG 24 GE K SVM_r, KP 722, SG, SP 2

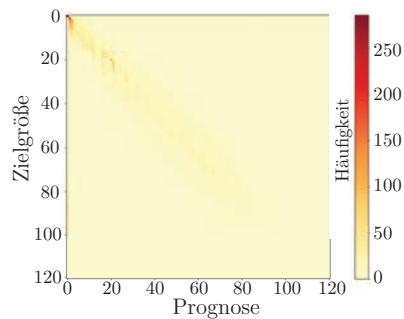


Abbildung 5.9: Konfusionsmatrix für SG 24 GE R kNN, KP 722, SG, SP 2

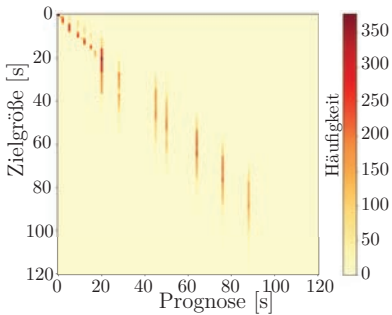


Abbildung 5.10: Konfusionsmatrix für SG 24 GE K DT, KP 722, SG, SP 2

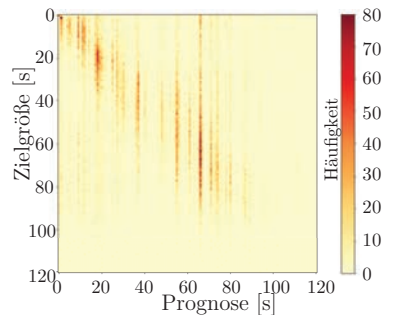


Abbildung 5.11: Konfusionsmatrix für SG 24 GE K LSTM, KP 722, SG, SP 2

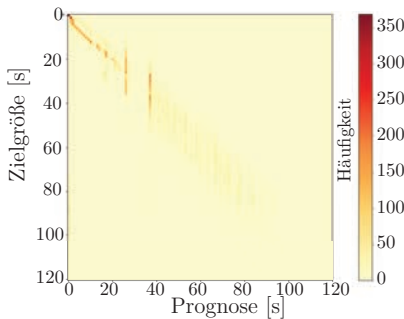


Abbildung 5.12: Konfusionsmatrix für SG 24 GE K SVM_r, KP 722, SG, SP 2

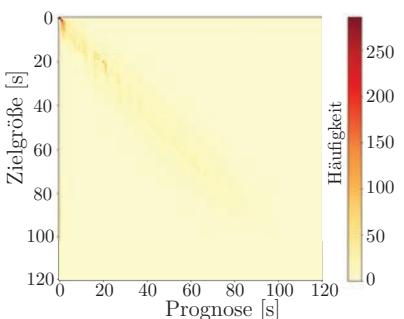


Abbildung 5.13: Konfusionsmatrix für SG 24 GE R kNN, KP 722, SG, SP 2

5.4 Ergebnisse des Vergleichs mit der Methode von TTS

Die hier verfolgten Ansätze wurden mit der SZP von Traffic Technology Services Inc. Europe (TTS) verglichen, um die Prognosequalität in den Kontext etablierter Verfahren zu setzen. Die konzeptuelle Vorbereitung der Experimente wird in Abschnitt 5.4.1 umrissen. Für den Vergleich wurden mehrere Datenmodelle und ML-Methoden verwendet, wie in Abschnitt 5.4.2 beschrieben ist. Die zugehörigen Ergebnisse und Schlussfolgerungen werden in den Abschnitten 5.4.3 und 5.4.4 dargestellt und beruhen auf [Sch21].

5.4.1 Versuchsaufbau

Als Datengrundlage der Versuche wurden Simulationsdaten verwendet. Die Simulationsdaten wurden mit VISSIM erzeugt. Die Daten der Simulation wurden sekundlich und in einem äquivalenten Umfang wie die bereits bestehende Datengrundlage aus dem Projekt VERONIKA [Uni19] für zwei Wochen und für alle vier Programme erfasst.

Da zum Zeitpunkt der Erarbeitung weder die Versorgungsdaten noch die Steuerungen der in allen anderen Teilen dieser Arbeit betrachteten Knotenpunkte für die Simulation zur Verfügung standen, wurde Knotenpunkt „005 Trompete“ betrachtet. Es handelt sich um einen dreiarmligen Knotenpunkt mit fünf SG. Zwei davon sind für die separat geführten Streifen des ÖPNV. FG-Verkehrsströme werden unterhalb der Straßenebene durch Unterführungen geführt und sind damit vollständig vom mIV am Knotenpunkt getrennt. Es besteht eine Freigabezeit Anpassung sowie die Möglichkeit zum Phaseneinschub.

5.4.2 Versuchsdurchführung

Für die Versuchsdurchführung wurden mehrere Datenmodelle entwickelt. Das Datenmodell von TTS verwendet millisekundengenau erfasste Detektordaten zur Merkmalsberechnung. Da diese Merkmalsberechnung proprietär für TTS ist, wurde auf eine entsprechend genaue Simulation und Rekonstruktion dieser Merkmale verzichtet. Das Datenmodell von TTS umfasst für die Vorhersage von GA beispielsweise Zeitstempel, Umlaufsekunden und verschiedene Zeitdifferenzen zwischen den GA beziehungsweise GE der SG, Detektoraktivierungen und ÖPNV-Meldungen, welche mit $t_{x,w}$ vergleichbar sind, siehe Abschnitt 6.1.1. Da aufgrund der Datenmodellierung von TTS keine eigenen Simulationsdaten der Vorhersage von TTS zugeführt werden können, wurde auf ein alternatives Vorgehen zurückgegriffen. Zusätzlich zu den Prozessdaten von LSA verwendet TTS ihre Versorgungsdaten für die Berechnung ihres Datenmodells. In diesen sind beispielsweise Erlaubnisbereiche von Freigaben, deren maximale und minimale Dauern hinterlegt.

Es wurde ein ML-Modell entwickelt, welches unter Verwendung der bereits vorliegenden Daten das Verhalten der TTS-Vorhersage möglichst gut nachbildet. Das Modell ist in seiner Struktur mit der von TTS identisch. Wie genau die Parametrierung der Struktur mit der ursprünglichen von TTS übereinstimmt, wurde experimentell ermittelt und die Ergebnisse beider Modelle verglichen, siehe Abschnitt 5.4.3. Dieses Modell wurde als Referenz für den Vergleich genutzt. Hierdurch wurde zudem der Prognosevergleich mit anderen Datenmodellen basierend auf Daten eigener Simulationen ermöglicht. So können auch Aussagen über den Zusammenhang zwischen Prozess- und Simulationsdaten eines Knotenpunktes getroffen werden. Verfügbare Merkmale für ein überarbeitetes Datenmodell auf Basis der Simulation sind:

- Zeitstempel,
- Programmnummer,
- Umlaufsekunde,
- SG-Zustand (1/0),
- SG-Zustandsdauer seit letztem Zustandswechsel [s],
- Belegungsdauer Detektoren
- Dauer seit Anforderung für Detektoren und
- ÖPNV Anforderung (0/1).

Als ML-Methoden wurden Entscheidungsbäume, RF und XGB für den Vergleich herangezogen. Zwecks Optimierung wurden manche Hyperparameter mittels Bayes'scher Suche angepasst. Auf eine vollständige Optimierung wurde wegen der begrenzten Ressourcen am Rechencluster abgesehen. Die Entscheidungsbäume wurden zur Regularisierung zusätzlich einem Baumschnitt (Cost Complexity Pruning (CCP)) unterzogen. Die Versuche wurden mit einem Stichprobenumfang von ca. 50.000 Sekunden für die Hauptverkehrszeit (Programm 2) durchgeführt. Die Test- und Trainingsdaten wurden im üblichen Verhältnis von 1:2 aufgeteilt.

5.4.3 Ergebnisse

Zunächst wurden die Prognosen der parametrierten Methoden mit der TTS-Prognose unter Verwendung des Datenmodells von TTS in einer ersten Versuchsreihe verglichen. Hiernach wurde das oben genannte Referenzmodell erstellt. Mit diesem wurden die Methoden unter der Verwendung der Rohdaten, wie sie von der Simulation angeboten werden, ohne weitere Verarbeitung miteinander verglichen. Zudem wurde im Rahmen der Arbeit ein überarbeitetes Datenmodell erstellt, zu dem abschließend Ergebnisse vorgestellt werden.

TTS-Datenmodell

Die Ergebnisse der ersten Versuchsreihe, welche in Abbildung 5.14 zu sehen sind, zeigen, dass die Methoden vergleichbar gut abschneiden. Sie sind in der Abbildung nach SG gruppiert. Die Prognosen für GA (links) erzielen betreffend dem KKR (Genauigkeit, oben) bessere Ergebnisse als die für GE (rechts). Diese zeigen jedoch einen geringeren und damit besseren RMSE (unten).

Tabelle 5.4: Gemittelte KKR [%], Modellvergleich mit TTS-Datenmodell [Sch21]

Modell	GA	GE
CCP-Baum	97	65
XGB	99	67
RFs	92	61
TTS	97	64

Insgesamt ist die KKR, wie in Tabelle 5.4 zu sehen, für die GA mit über 95 % sehr hoch, während sie für GE mit nur über 60 % keine zufriedenstellenden Ergebnisse zeigt. Dies kann darauf zurückgeführt werden, dass die Datenmodelle für GA nur Informationen zur Rotzeit und solche für GE nur Informationen zur Grünzeit enthalten.

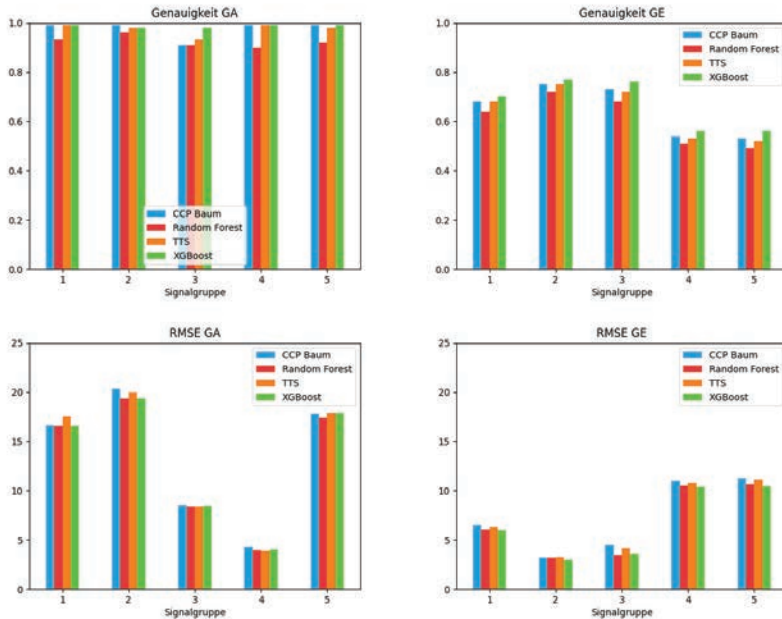


Abbildung 5.14: Modellvergleich KKR [%] (Genauigkeit) und RMSE [s] mit dem TTS-Datenmodell [Sch21]

Die Dauer der Rotzeit einer SG während eines Umlaufs ist deutlich länger als die der Grünzeit. Entsprechend unterschiedliche Mengen an Information können gesammelt werden. Zudem werden Freigabezeitenanpassungen zumeist durch Detektionen von Zeitlücken am Ende der Grünzeit umgesetzt. Dem zufolge liegt die entscheidende Information zu GE erst kurz vorher vor und kann kaum vorhergesagt werden. Der ungefähre Zeitraum für GE wird jedoch besser vorhergesagt als für GA, was der vergleichsweise geringere RMSE in der Abbildung veranschaulicht.

Die Tabelle zeigt zudem, dass Entscheidungsbäume bezüglich dieser Versuche die Ergebnisse von TTS nachbilden können und XGB die besten und RF die schlechteste Vorhersagen ergibt. Durch die Berechnung der Korrelationskoeffizienten des RMSE kann dies für GE mit Korrelationen von ca. -1 bestätigt werden. Für die GA zeigt nur XGB eine geringfügige negative Korrelation von -0,09, die verbleibenden Methoden zeigen eine positive Korrelation für die Vorhersage von GA.

Rohdatenmodell

Für die Vorhersage mittels eigener Simulationsdaten (im Rohzustand) wurde zunächst ein Referenzmodell (TTS Params) erstellt. Basis hierfür bildet die bereits bestehende Datengrundlage. Ein Vergleich zwischen Referenz und Original ist in Abbildung 5.15 dargestellt. Entsprechend ist betreffend der im Folgenden dargestellten Versuche davon auszugehen, dass das TTS-Modell für GE bessere Ergebnisse erzielt hätte als das Referenzmodell.

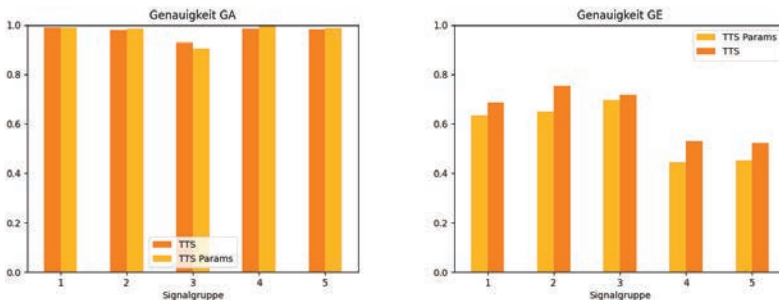


Abbildung 5.15: TTS Params-Modellreferenz vs. original TTS-Modell [Sch21]

Die Ergebnisse unter Verwendung des Rohdatenmodells, welche in Abbildung 5.16 und Tabelle 5.5 zu sehen sind, zeigen eine geringere Vorhersagequalität als die Vorhersagen der zuvor beschriebenen Experimente. XGB bietet nun die schlechteste Vorhersage und das Referenzmodell die beste.

Tabelle 5.5: Gemittelte KKR [%], Modellvergleich mit Rohdaten [Sch21]

Modell	GA	GE
XGB	35	49
RFs	55	57
TTS Params	62	64

Durch Hyperparameteroptimierungen konnte ausgeschlossen werden, dass XGB aufgrund mangelhafter Parametrierung das Schlusslicht dieser Versuchsreihe bildet, da sich daraus lediglich eine Verbesserung um 1 % ergab. Insgesamt ist die Vorhersagequalität mit diesem gegenüber dem zuvor beschriebenen Datenmodell insbesondere für GA gefallen.

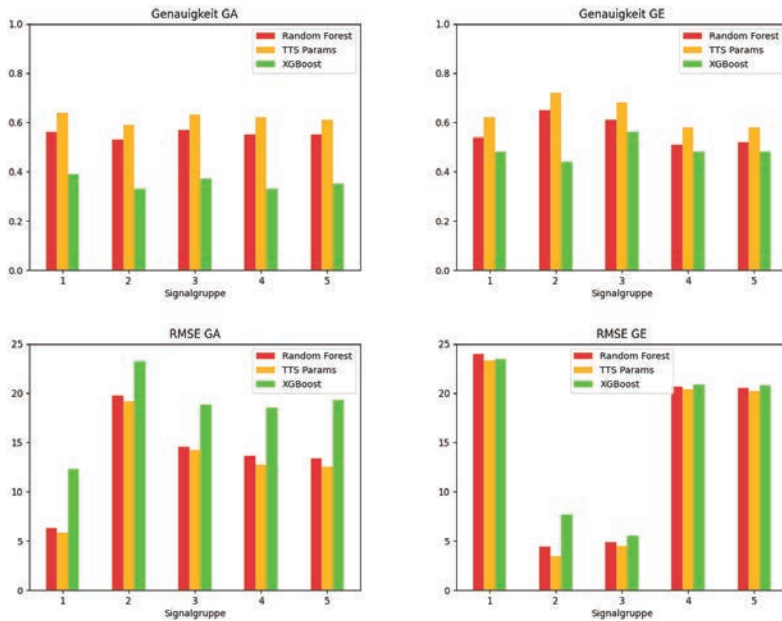


Abbildung 5.16: Modellvergleich KKR [%] (Genauigkeit) und RMSE [s] mit Rohdaten [Sch21]

Datenmodell nach [Sch21]

Die Versuchsreihe mit dem Datenmodell nach [Sch21] zeigt ebenfalls nicht die Vorhersagequalität, welche mit dem TTS-Datenmodell erzielt wurde, siehe Abbildung 5.17 und Tabelle 5.6. Auffällig ist jedoch, dass die KKR für die Prognosen der GE um ca. 20 % besser ausfallen, da die Datenmodelle für GA und GE jeweils beide Informationen der Grün- und Rotzeiten enthalten.

Tabelle 5.6: Gemittelte KKR [%], Modellvergleich mit überarb. Datenmodell [Sch21]

Modell	GA	GE
XGB	55	80
RFs	68	76
TTS Params	72	80

Dies könnte jedoch auch erklären, warum die Prognose des GA schlechter ausfällt. Die Bayes'sche Suche impliziert, dass XGB für diese Experimente eine ähnliche bis bessere Vorhersage stellen kann als TTS Params. Die Abbildung und die Tabelle zeigen die Ergebnisse vor der Suche.

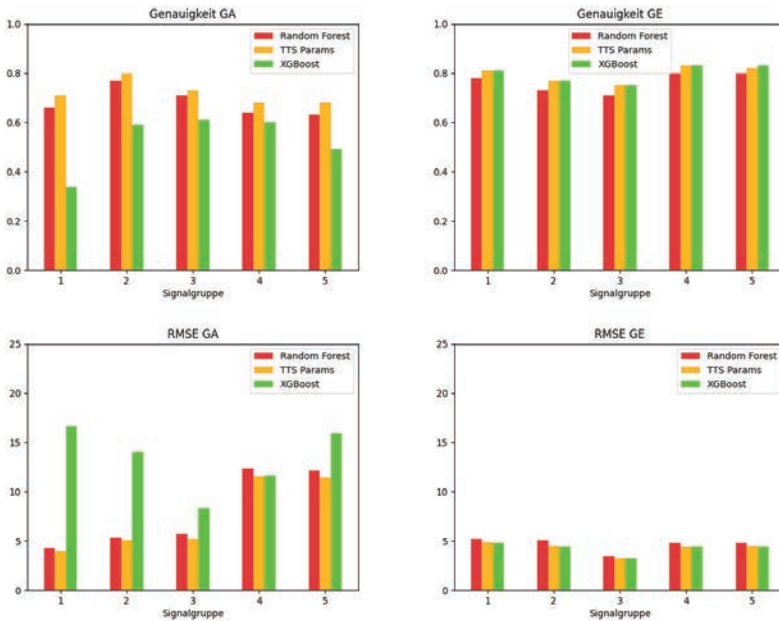


Abbildung 5.17: Modellvergleich KKR [%] (Genauigkeit) und RMSE [s] mit dem überarbeiteten Datenmodell [Sch21]

5.4.4 Schlussfolgerungen

XGB ist bei Betrachtung der Experimente, welche hier nicht alle gezeigt werden konnten, die bessere ML-Methode für aufbereitete Datenmodelle. Die Ergebnisse unter Verwendung des Rohdatenmodells verglichen mit den anderen Datenmodellen zeigen, dass eine Aufbereitung der Prozessdaten für eine SZP sinnvoll ist. Das TTS-Datenmodell scheint dahingehend besonders geeignet. Beachtenswert ist, dass die Ergebnisse für XGB ohne die Verwendung der Versorgungsdaten erfolgen konnte. Eine Überanpassung des TTS-Modells kann nicht ausgeschlossen werden.

Die Performanz war für diese Versuchsreihen an diesem Knotenpunkt mit (aufbereiteten) Prozessdaten (TTS-Datenmodell) besser als mit Simulationsdaten. Jedoch erlauben die Ergebnisse für diesen einen Knotenpunkt keinen Rückschluss auf die Gesamtheit, da die Stichprobe mit einem Knotenpunkt aus nur einer Stadt nicht repräsentativ ist. Die Datenmodelle, welche in allen anderen Abschnitten dieser Arbeit verwendet wurden, sind eher mit dem TTS-Datenmodell vergleichbar, als mit den Datenmodellen, welche auf der Simulation basieren; dies betrifft insbesondere die Verarbeitung der ÖPNV-Meldungen.

5.5 Zwischenfazit

In allen Experimenten zur Methodenwahl für SG und SG_s dominieren entscheidungsbaumbasierte Verfahren unter den fünf besten maschinellen Lernverfahren. Der Klassifikator von XGB zeigte sich als die beste Methode zur SZP mit den betrachteten Datenmodellen. Wenn dieser Klassifikator nicht der beste ist, handelt es sich um einfache DT oder RF. Darüber hinaus sind SVM auf der Grundlage von RBF und gelegentlich kNN unter den besten fünf zu finden, aber nie unter den besten drei. Die besten drei Methoden waren in der Regel deutlich besser als die schlechtere Hälfte der Prognosemethoden. SAR und alle auf neuronalen Netzen basierenden Methoden (RNN, LSTM, GRU und MLP) gehörten stets zur schlechteren Hälfte. SAR belegte in der Regel den letzten Platz. Die Ergebnisse wurden auf den Einfluss von schlechter Parametrisierung untersucht. Schlechte Parametrierung konnte als Ursache durch die Verwendung der Bayes'schen Suche [sci20] zur Hyperparameteroptimierung ausgeschlossen werden. Allgemein sind Post-Hoc Signifikanztests zufolge die Ergebnisse zur SZP Klassifikatoren geeigneter als Regressoren, was auch auf das diskrete, nicht stetige Schaltverhalten von Logiken zurückgeführt werden kann.

Aufgrund der eindeutigen Ergebnisse wird auf weitere Betrachtungen verzichtet und XGB als beste Methode zur SZP im Rahmen dieser Arbeit betrachtet. Dies wird von hier nicht dargestellten Vorexperimenten mit vielversprechenderen Datenmodellen aus Kapitel 6 gestützt. Zudem erscheint XGB mit den gängigen kommerziellen Methoden vergleichbar bezüglich der hier betrachteten KP.

Vor diesem Hintergrund kann das in Abschnitt 4.2 benannte Konzept ohne dedizierte Methodenwahl durchgeführt werden. Eine Methodenwahl erscheint nur im Fall einer durch XGB unzureichenden SZP notwendig.

Kapitel 6

Experimente zu Datenmodellen zur signalgruppenorientierten Vorhersage

Dieses Kapitel dokumentiert die Datenmodellierung für das Verfahren zur SZP, um die dritte und vierte Forschungsfrage aus Abschnitt 1.2 betreffend geeigneter Datenmodelle und der Integration von Information benachbarter LSA beantworten zu können. Das Vorgehen und die Ergebnisse dieses Kapitels stehen in Wechselwirkung mit denen aus Kapitel 5.

Zunächst werden in Abschnitt 6.1 mögliche Merkmale aus Sicht des Anwendungsgebiets diskutiert. Hiernach wird der Versuchsaufbau in Abschnitt 6.2 beschrieben.

Versuche werden zu verschiedenen Repräsentationen der Merkmale beispielhaft für SG in Abschnitt 6.3 gezeigt. Der Unterschied zwischen der Integration von nur einer SG im Vergleich zu allen verfügbaren SG einer LSA wird in 6.4 diskutiert und zuletzt werden Ergebnisse der Experimente zur automatisierten Merkmalsselektion mit SG, Det, ÖPNV in Abschnitt 6.5 vorgestellt. Vor dem dieses Kapitel abschließenden Zwischenfazit in Abschnitt 6.7, werden Ergebnisse zu Experimenten mit Daten benachbarter Knotenpunkte in Abschnitt 6.6 beschrieben.

6.1 Merkmale aus Sicht des Anwendungsgebiets

In diesem Abschnitt wird die Berechnung der Merkmale für die in den Experimenten verwendeten Datenmodelle aufgezeigt. Hierfür werden zunächst bekannte Merkmale von Datenmodellen vorgestellt (Abschnitt 6.1.1) und daraus weitere Möglichkeiten abgeleitet (Abschnitt 6.1.2).

6.1.1 Bekannte Merkmale

Grundlegende Merkmale, welche von Weisheit [Wei17] verwendet wurden, sind:

- GA und GE der SG,
- erfüllte Belegdauern > 5 s,
- letzte Detektion (Abflanke) des/der zugehörigen Detektors/en,
- letzte Detektion (Abflanke) +3s des/der zugehörigen Detektors/en falls keine Aufflanke erfolgt,
- Anmeldung vorhanden für FG und ÖV,
- aktuelle Verkehrsstärke (ermittelt über Detektordaten).

Zielgröße dieses Datenmodells sind das nächste Eintreten von GA beziehungsweise GE in Form eines Countdowns. Bis auf die Verkehrsstärke wurden alle Merkmale von Weisheit mit Zeitstempel versehen, welche in Bezug zum letzten Eintreten der Zielgröße gesetzt sind. Das heißt die Eingangsgrößen wurden in Form von Umlaufsekunden angegeben, welche ab dem letzten bekannten GA respektive GE der prädierten SG gezählt wurden, was gegenüber den von der LSA gezählten Umlaufsekunden eine Verschiebung darstellt. Diese relativen Umlaufsekunden nach der Definition von Weisheit werden im Folgenden mit $t_{x,w}$ symbolisiert in Unterscheidung zu den Umlaufsekunden der LSA t_x .

Diese Merkmale zählen zu den bisher umfangreichsten Datenmodellen auf Basis von LSA-Prozessdaten für SZP und zählen zu den wenigen, die Detektordaten einbeziehen, siehe auch Abschnitt 2.7. Detektordaten wurden unter anderem auch von [Bod17] verwendet. Die Datengrundlage der Dissertation zu SZP von Protschky wird nicht in Betracht gezogen, da es sich hierbei um FCD handelt, siehe [Pro16]. Das in Abschnitt 5.4 referenzierte proprietäre Datenmodell von TTS stellt eine Weiterentwicklung des oben beschriebenen Datenmodells dar, wird aus rechtlichen Gründen in dieser Arbeit jedoch nicht weiter beschrieben.

6.1.2 Merkmalsentwicklung

Basierend auf den Merkmalen von Weisheit wurden weitere entwickelt. Informationen, welche in Merkmale integriert werden können, betreffen:

- den aktuellen Zeitpunkt der Vorhersage,
- den Wochentag (1-7),
- den letzten GA von SG_s ,
- das letzte GE von SG_s ,
- Informationen zu Phasen und Phasenübergängen,

- die Belegung/Anforderung von DET_d ,
- das Bestehen von Rückstau an DET_d ,
- die Erfüllung von Mindestbelegungsdauern beziehungsweise -wartezeiten bezüglich DET_d ,
- die Erfüllung von Zeitlückenkriterien für DET_d ,
- die aktuelle Verkehrsstärke von DET_d ,
- ÖPNV-Meldungen zur Richtung_r mit den zugehörigen Informationen (siehe Abschnitt 2.4.1) jeweils als Merkmal,
- den Belegungsgrad von DET_d ,
- Schätzungen der Pulkgeschwindigkeit auf den zuführenden Verkehrswegen,
- Ankunftszeit- sowie Rückstauprognosen an SG_s beziehungsweise DET_d ,
- FCD und ähnliche.

Im Rahmen dieser Arbeit wird die Integration von Informationen zu den letzten vier Punkten dieser Liste ausgeschlossen. Auch wird nur die Verkehrsstärke ermittelt und nicht die Dichte.

Die Informationen können in verschiedenen Darstellungsweisen erfasst beziehungsweise berechnet werden. Mögliche Repräsentationen sind unter anderem

1. die native Darstellungsform der Information (ⁿ),
2. die Umlaufsekunde t_x zum Zeitpunkt einer Informationsveränderung,
3. die (relative) Umlaufsekunde, gezählt ab der letzten Änderung der Zielgröße (t_{x_w}),
4. die Zeit in Minuten, gerundet ab 0:00 Uhr der letzten Veränderung der Information (_{min}) und
5. die Dauer seit der letzten Veränderung der Information („Zeit seit“, t_s)

denkbar. Die native Darstellungsform der Information beschreibt die in der Datengrundlage verwendete Repräsentation sowie Darstellungsweisen, welche durch einfache Berechnungen gewonnen werden können, insbesondere in Form der in Abschnitt 2.3 beschriebenen Größen. Auf weitere mathematische Operationen wie beispielsweise die Quadratur einzelner Informationen wird verzichtet. Aus den Kombinationsmöglichkeiten von Information und Repräsentation ergibt sich eine große Anzahl möglicher Merkmale. Die möglichen Merkmale werden im Folgenden näher erläutert. In Anhang D sind Tabellen zur Veranschaulichung der Merkmale dargestellt. Sie zeigen auch die im Framework verwendeten Bezeichner.

Zeit

Der Zeitpunkt der Vorhersage kann in den mittleren drei Formen der oben genannten Aufzählung dargestellt werden. Der Wochentag stellt betreffend der Darstellungsweise eine Ausnahme dar. Er wird als native Form numerisch dargestellt mit 1 repräsentativ für Montage, 2 für Dienstage usw.

Signalgruppen

Die Zustände der Signalgruppen werden primär in ihrer **nativen, binären Form (SG Merkmal 1)** betrachtet. Als SG-Nummer werden die im OCIT-Protokoll verwendeten ID-Nummern verwendet. Diese stimmen nicht mit den SG-Nummern in den Lageplänen überein. Wie in Kapitel 2 beschrieben, wird bezüglich SG vereinfachend binär zwischen Freigabe (1) und Sperrung (0) unterschieden. Übergangssignale werden Sperrungen zugerechnet. Hieraus können für jede SG weitere Merkmale **GA und GE getrennt** berechnet werden, welche eine Darstellung mit Integration von temporärer Information ermöglichen. Dies kann beispielsweise entsprechend obiger Aufzählung in Abschnitt 6.1.2 zu **Repräsentationen 2**, die **Angabe der Umlaufsekunde (SG Merkmal 2 und 3)** des letzten GA (erste 1 einer Folge) oder 3. GE (erste 0 einer Folge) oder die **Angabe der Zeit seit GA oder GE (SG Merkmal 4 und 5)** sein. Des weiteren neben 2. und 3. obiger Aufzählung die SG Merkmale 6 bis 9 entsprechend 4. und 5. Entsprechend ergeben sich für jede Signalgruppe **ein natives Merkmal** sowie für **GE und GA jeweils vier** mit der Zeit verknüpfte Merkmale (SG Merkmal 2 bis 9).

Native Darstellungsweisen mittels Codierungen nach OCIT werden nicht in Betracht gezogen, da angenommen wird, dass die zusätzliche Information zu Übergangssignalen der Qualität der Datenmodelle nicht zuträglich ist, da die Informationen zu Gelb- beziehungsweise Rot-Gelbsignalen für die Prognose nur in geringem Maße unterstützend wirken können aufgrund ihrer starren Verwendung im Kontext von Freigabe und Sperrung. Ebenso wird kein fixer Prognosehorizont angenommen und hierfür der Signalisierungszustand vorhergesagt, wie es in [Ke21] vorgenommen wurde, da dies den Horizont einschränkt und nicht unmittelbar zu einer Prognose der gewünschten Information führt.

Aus der Menge der SG-Zustände lassen sich die Phasen(-übergänge) ableiten, wie in [Wei19a] beschrieben, und somit weitere Merkmale berechnen. Dies wird in einem eigenen Kapitel (7) gesondert betrachtet.

Detektoren

Die Detektordaten liegen binär vor und sind mit Zeitstempeln versehen, welche zentisekundengenau erfasst wurden. Für die sekundlich bestimmten Merkmale wird jeweils der letzte Detektorzustand vor einer vollen Sekunde herangezogen. Dies wird als **natives Merkmal (DET Merkmal 1)** verstanden. **Verknüpfungen mit Zeitangaben**

des nativen Merkmals entsprechend der obigen Aufzählung zu Repräsentationen 2. bis 5. übertragen auf **jeweils Aktivierung sowie Deaktivierung** des Detektors ergeben **DET Merkmal 2 bis 9**. Dies erfolgte äquivalent zu den entsprechend SG Merkmalen.

In Verbindung mit den Zeitstempeln können weitere Merkmale berechnet werden, wie die **Verkehrsstärke (DET Merkmal 10)** als Menge der Detektoraktivierungen innerhalb des letzten Umlaufes. Auch die binär codierte Erfüllung von **Mindestbelegungsdauern (DET Merkmal 11 bis 13)** oder **Zeitlückenkriterien (DET Merkmal 14 bis 16)** als ununterbrochene (Nicht-)Belegungen in den zentisekundengenauen Daten werden als Merkmale berechnet. Als Kriterien für die Mindestbelegungsdauern sowie die Zeitlückenkriterien wurden drei verschiedene Dauern von jeweils 5 s, 10 s und 30 s Dauer angenommen. Für die Zeitlückenkriterien wurden längere Dauern als üblich angenommen, da die hier betrachteten Programme mit Belegungsgraden arbeiten. So wurde versucht sich an das reale Verhalten anzunähern ohne die den einzelnen Programmen hinterlegten Berechnungen der Belegungsgrade, welche den Herstellern proprietär sind, erraten zu müssen.

Die Merkmale zu **Belegdauern und Zeitlücken** können ebenfalls mit den **Zeitangaben** gemäß der obigen Aufzählung zu Repräsentationen 2. bis 5. **verknüpft** werden. Je Kriterium ergeben sich so vier zusätzliche Merkmale, die jeweils eine Zeitangabe für den Zeitpunkt der Kriteriumserfüllung enthält. Dies sind **DET Merkmale 17 bis 30**.

Abhängig von der Art und Verwendung des Detektors sind nicht alle Merkmale hierfür sinnvoll, da die Ermittlung von Zeitlücken mittels Tastern für FG-Ströme nicht möglich ist. Zur Selektion dieser wird versucht, über eine automatisierte Merkmalswahl zu integrieren.

Zusätzlich zu den oben genannten Darstellungsformen können Gültigkeitsbereiche innerhalb von Umläufen für Informationen definiert werden. Entsprechend können bestimmte Zustände beziehungsweise die Veränderung der Information nur innerhalb dieser Bereiche in Merkmale integriert werden. Beispielsweise können Zeitlückenkriterien nur zu Freigaben respektive Belegungszeitkriterien nur während Sperrungen erfüllt werden.

Dies lässt sich auf Datenmodelle übertragen, so dass nur Informationen, welche während Freigaben erfasst wurden, für die Vorhersage von GE respektive während Sperrungen für GA verwendet werden. Hiervon wird hier Abstand genommen, da insbesondere für GE Kriterien bereits während der vorhergehenden Sperrung erfüllt sein können wie beispielsweise für eine Wartezeit.

ÖPNV-Meldungen

Die ÖPNV-Meldungen werden nach Fahrtrichtungen („Richtung“) gruppiert. Jeder Gruppe werden zeitliche Merkmale entsprechend der oben genannten Repräsentationen 2. bis 5. zugeordnet. Als weitere Merkmale werden zu jeder Gruppe die in Abschnitt 2.4.1 genannten Informationen zugeordnet.

Zielgrößen

Als Ziel- beziehungsweise Ausgangsgrößen werden das nächste GE und der nächste GA jeder Signalgruppe betrachtet. Repräsentationen dieser können, angelehnt an die obige Aufzählung in Abschnitt 6.1.2,

1. die Dauer bis zum nächsten Eintreten (Countdown/time until beziehungsweise „Zeit bis“ mit der vorangestellten Markierung t_u),
2. die Umlaufsekunde (Markierung t_x),
3. die relative Umlaufsekunde (Markierung t_{x_w}) und
4. der Zeitpunkt in Minuten gerundet ab 0:00 Uhr (Markierung min)

als Zeitpunkte des nächsten Eintretens sein.

Des Weiteren kann zusätzlich die Eintretenswahrscheinlichkeit ($P(x)$) für die Vorhersage(n) angegeben werden. So könnten alternativ Vorhersagen, zu einem Möglichkeitsraum gebündelt, angegeben werden. Für diesen könnte eine Mindestwahrscheinlichkeit von zum Beispiel 95 % definiert werden. Statt der wahrscheinlichsten Vorhersage, deren Eintretenswahrscheinlichkeit unter 95 % liegen kann, kann so eine Vorhersagequalität gewährleistet werden, insofern der Raum einen definierten Toleranzbereich nicht überschreitet. Dies benötigt zusätzliche Experimente, die im Rahmen dieser Arbeit nicht durchgeführt werden können.

6.1.3 Überlegungen zu Merkern

Die hohe Zahl möglicher Merkmale legt nahe, Einbettungen im mathematischen Sinne wie zum Beispiel PCA, siehe Abschnitt 3.3, zu verwenden, um die Dimension der verwendeten Datensätze zu reduzieren. An dieser Stelle werden manuelle Einbettungen, welche aus Sicht von LSA-Vorlogiken als Merker bezeichnet werden können, vorgestellt. Diese Idee wird hier diskutiert, in den Experimenten jedoch nicht verwendet.

Grundsätzlich wird eine Unterscheidung von drei Typen von Merkern vorgeschlagen: einfache logische Verknüpfungen, Einbettungen wie die Berechnung von Phasen und die Nachbildung von Merkern der Vorlogik. Die Merker der Vorlogik sind in den Merkmalen integriert. Es handelt sich beispielsweise um Zeitlückenkriterien. Die Ausgaben einer Vorlogik, die in die Hauptlogik zur Entscheidungsfindung gegeben werden, können somit vollständig in die Merkmale eines Datenmodells für eine SZP integriert werden. Die Integration erfolgt über die Algorithmen zur Merkmalsberechnung.

Bei einfachen logischen Verknüpfungen werden Merkmale über Und- oder Oder-Logiken miteinander verknüpft: Beispielsweise existieren mehrere Taster für den FG-Verkehr für eine SG, welche mit logischem ODER zu einem Merkmal zusammengeführt werden könnten.

Auch Merkmale von SG zu GA und GE könnten zusammengefasst werden, beispielsweise indem Informationen für GA positiv und GE negativ im Merkmalsvektor für den zuletzt eingetretenen Zustand (GA respektive GE) verzeichnet werden. Vorversuche haben gezeigt, dass diese Verknüpfung vergleichbare Ergebnisse wie die oben beschriebenen Merkmale für SG zeigen, weshalb dies nicht näher dargestellt wird.

Angelehnt an die Codierungen nach OCIT für SG können Detektordaten in verdichtete Darstellungsweisen überführt werden, indem in einem Merkmal numerische Codierungen für die verschiedenen Zustände beziehungsweise Kriterien zu den entsprechenden Zeitpunkten eingetragen werden. Eine solche Darstellungsweise hat den Vorteil, dass weniger Rechenzeit benötigt wird und Merkmale, die gleiche beziehungsweise ähnliche Information in anderer Repräsentation darstellen, in einem Merkmal gebündelt werden. Da hier zunächst die Relevanz einzelner Merkmale untersucht werden soll, um im Rahmen einer Voranalyse einzelne als sinnvoll für Verdichtungen benennen zu können, werden keine Verdichtungen vorgenommen.

Detektormerkmale könnten wie die SG zu Phasen in Einbettungen zusammengefasst werden. Zum Beispiel könnte ein Gesamtzustand für alle Rückstaus oder Wartezeiten für alle relevanten Detektoren gebildet werden. In Abschnitt 7.1 ist dies für SG-Zustände gezeigt. Alternativ können auch gemischte Kombinationen der Merkmale adaptiv in Abhängigkeit zur aktuellen Phase beziehungsweise zum Phasenübergang zusammengefasst werden. Beispielsweise zu einem Zeitpunkt die Zeitlücken für die aktuell freigegebenen SG sowie die Wartezeiten und Rückstaus an den gesperrten SG. Abhängig vom betrachteten Zeitpunkt würde sich die Zusammenstellung der Detektormerkmale verändern, da andere SG freigegeben beziehungsweise gesperrt sein können.

6.2 Versuchsaufbau

In diesem Abschnitt ist die Zusammenstellung der im Vorherigen beschriebenen Merkmale zu Datenmodellen beschrieben, um entsprechend Frage drei geeignete Datenmodelle zu identifizieren. Zusätzlich wird die Auswahl und Parametrierung der verwendeten ML-Methoden aufgezeigt.

6.2.1 Datenmodelle

Als Zielgröße werden alle GA und GE aller Signalgruppen - $SG_{s,GA/GE} \forall s \in S$ - der Knotenpunkte Druseltalstraße/Heinrich-Schütz-Allee (722) und Katzensprung (002) betrachtet. Es wurden Daten der Monate September und zum Teil Februar 2020 mit den Programmen für die Haupt- und Nebenverkehrszeit (SP 2 und 3) ausgewählt. Darstellungen der Zielgrößen für Klassenhäufigkeiten der Test- und Trainingssets sowie Veranschaulichungen der Verkehrsabhängigkeiten sind in Anhang C dargestellt. Die Eingangsdatenmodelle werden für jede Zielgröße separat getestet.

Entsprechend der hohen Zahl möglicher Merkmale und ihrer Zusammenstellung in Datenmodellen, welche sich auf mehr als eine Milliarde pro Knotenpunkt beläuft, können nicht alle möglichen Datenmodelle getestet werden. Deshalb werden die Merkmale in Gruppen getestet. Die Gruppen der Merkmale sind: Signalgruppen (SG), Detektoren (DET) und ÖPNV-Meldungen (OEV). Tabelle 6.1 zeigt die betrachteten Datenmodelle getrennt nach Gruppen.

Für eine vereinfachte **Schreibweise der Eingangsdatenmodelle** wird eine eigene Notation verwendet: Falls keine Indizes angegeben sind, sind alle Merkmale einer Merkmalsgruppe gemeint, zum Beispiel SG für alle Merkmale aller SG des Knotenpunktes. Wenn nachfolgende Indizes angegeben sind, ist nur das Merkmal ausgewählt, welches zur Zielgröße gehört; das heißt beispielsweise für die Eingangsdaten SG_s nur die Merkmale $SG_{s,GA}$ und $SG_{s,GE}$ enthalten sind, wenn als Zielgröße $SG_{s,GA}$ oder $SG_{s,GE}$ betrachtet werden. Ein hochgestelltes n kennzeichnet die Verwendung der zugehörigen nativen Merkmale, zum Beispiel SG_s^n , siehe Abschnitt 6.1.2.

Vorangestellte Indizes zeigen entsprechend der Aufzählung in Abschnitt 6.1.2 die für die Daten gewählte Repräsentation an, beispielsweise ${}_t SG$. Ein vorangestelltes $_a$ zeigt an, dass alle Repräsentationen für die Ein- und Ausgangsdaten des Datenmodells in separaten Experimenten getestet wurden. Ansonsten werden die entsprechenden dort beschriebenen Indizes vorangestellt.

Das für die Selektion gewählte Perzentil an zu behaltenden Merkmalen wird mit P gekennzeichnet. Entsprechend beschreibt $SG_s^n, P = 60$ beispielsweise das optimierte Datenmodell, in dem die wichtigsten 60% der Merkmale der zur Zielgröße gehörigen SG inklusive der natürlichen Darstellungsweise behalten wurden.

In den Datenmodellen sind, falls nicht anders gekennzeichnet, die Merkmale zur aktuellen Umlaufsekunde t_x , zur Uhrzeit in Minuten seit Mitternacht und zum Wochentag enthalten. Die Experimente mit Merkmalen benachbarter Knotenpunkte werden separat in Abschnitt 6.6 beschrieben.

Tabelle 6.1: Getestete Eingangsdatenmodelle, Zielgrößen: $SG_{s,GA/GE} \forall s \in S$

Experimentreihe	SG	DET	OEV
Repräsentation	${}_a SG_s^n$		
Eigene vs. alle Merkmal(e)	${}_{ts} SG_s^n$		
Merkmalsselektion	${}_{ts} SG_s^n, P = 25$	${}_{ts} DET^n, P = 25$	${}_{ts} OEV^n, P = 25$

6.2.2 Sonstige Einstellungen der Experimente

Die Merkmalsgruppen werden in klassischer Vorgehensweise mit der ML-Methode XG-Boost auf ihre Eignung getestet, siehe Abschnitte 3.2.1 und 3.5. Es werden, wenn nicht explizit anders beschrieben, ca. 100.000 Stichproben (Sekunden) der Wochentage Dienstag, Mittwoch und Donnerstag betrachtet. Die Aufteilung zwischen Test- und Trainingsdaten beträgt 1:2.

Als Evaluationsmetriken werden KKR und RMSE verwendet; zusätzlich werden die speziellen Metriken Q_i verwendet, siehe Abschnitt 3.4. Für den Vergleich der Ergebnisse wird auf den Friedman-Nemenji-post-hoc-Signifikanztest [Fri37, Fri40, Fri99] zurückgegriffen. Das Konfidenzniveau beträgt 95%. Alle Datenmodelle der Gruppen werden mit den Selektionsmethoden aus Abschnitt 3.2.5 untersucht, um Zusammenhänge zwischen den Daten aufzuzeigen und irrelevante Merkmale aus den Datenmodellen automatisiert zu entfernen.

Die Regularisierung der Methoden, siehe Abschnitt 3.2.3, wird mittels Bayes'scher Suche, siehe Abschnitt 3.3, automatisiert. Aufgrund des Rechenaufwands werden in Vortests nur Hyperparameter für ausgewählte Zielgrößen optimiert. Es wurden jeweils 15 Suchpunkte (`n_iter`) verwendet. Der gewählte Suchraum (`search space`) der Hyperparameter ist in Anhang F.1 zu finden. Die Hyperparameter wurden aufgrund dieser Experimente für alle Zielgrößen auf die in Anhang F.7 festgelegt, um Rechenzeit zu sparen. Weitere Vorversuche haben gezeigt, dass diese Hyperparameter für die meisten Zielgrößen zielführend sind für die Fragestellung dieses Kapitels. Es wurden hierfür weitere Hyperparameteroptimierungen durchgeführt.

6.3 Vergleich der Repräsentationen

Um geeignete Repräsentationen zu identifizieren, wurden mit ${}_aSG_s^n$ Experimente durchgeführt. Die Ergebnisse für die einzelnen Kombinationen der Repräsentationen wurden mittels Friedman-Nemenji-post-hoc-Test miteinander verglichen. Es wurden signifikante Unterschiede zwischen den Repräsentationen für beide betrachteten LSA festgestellt. Diese sind jedoch nicht eindeutig. Die Ergebnisse widersprechen sich bezüglich der beiden LSA und bezüglich KKR und RMSE. Die Ergebnisse erscheinen abhängig von der jeweiligen SG. Die Tabellen 6.2 und 6.3 sowie in Anhang H zeigen die Ergebnisse mit der selben Auswahl an SG wie in Kapitel 5. Die Spalte „Ein“ respektive „Ziel“ geben die betrachtete Repräsentationen der Eingangsgrößen respektive Zielgröße an, welche in der Kurzschreibweise tiefgestellt der zugehörigen Gruppe vorangestellt würde.

Gemeinsam für beide Evaluationsmetriken und LSA ist, dass die Vorhersage der Uhrzeit in Minuten seit Mitternacht (min) zwar zu einer vergleichsweise hohen KKR allerdings auch einem hohen RMSE geführt hat. Repräsentativ sind nur die Ergebnisse der Experimente mit der gleichen Repräsentation für Eingangssignale dargestellt. Ebenfalls gemeinsam ist allen SG, dass die Repräsentation in t_{xw} nur zu vergleichbaren beziehungsweise schlechteren Ergebnissen als andere Repräsentationen führt, was den zusätzlichen Rechenaufwand zur Erstellung dieser Merkmale nicht rechtfertigt.

Bei Betrachtung der Tabellen 6.2 und 6.3 entsteht der Eindruck, dass die Vorhersage der Umlaufsekunden (t_x) mit anschließender Berechnung der verbleibenden Zeit bis zum Schaltzeitpunkt zu besseren Prognosen führt als die direkte Vorhersage der Zeit bis zum nächsten Schaltzeitpunkt (t_u). Dies kann bei genauerer Betrachtung nicht bestätigt werden. Zum einen ist zu erwarten, dass die KKR für t_x höher ausfällt, da weniger unterschiedlicher Klassen vorherzusagen sind und damit die Wahrscheinlichkeit einer Fehlprognose sinkt. Zum anderen implizieren die Unterschiede im RMSE für die SG, dass eine Abhängigkeit vom Schaltverhalten der Signale besteht. Dies kann beispielsweise über die Betrachtung der Abbildungen 6.1 bis 6.4 bestätigt werden.

Die Abbildungen 6.1 bis 6.4 zeigen Ausschnitte aus den Ergebnissen. Sie veranschaulichen die Regelmäßigkeiten, die für die Vorhersage der Umlaufsekunden vorhanden sind. Die Konfusionsmatrizen zeigen zudem über die Achsenskalierungen die kleineren Lösungsräume mit ca. 50 beziehungsweise 20 Klassen auf. Zudem treten einzelne Klassen häufiger auf als andere, wie insbesondere Abbildung 6.2 zeigt. Es ist zu sehen, dass Klasse Nummer 16 deutlich häufiger auftritt als die anderen Klassen. Des Weiteren zeigt diese Abbildung, dass durch diese Häufung, der Algorithmus andere Klassen nicht mehr korrekt präzidieren kann, was die hellgrauen Felder verdeutlichen.

Tabelle 6.2: Ausgewählte Ergebnisse ${}_aSG_s^m$, KKR [%], jeweils zuerst GA dann GE je Repräsentationsart, KP 722, Monat Sep., SP 2

Ein	Ziel	SG 3	SG 5	SG 7	SG 13	SG 14	SG 24
<i>min</i>	<i>tx</i>	7,0	24,6	12,7	6,2	6,2	25,9
<i>min</i>	<i>tx</i>	43,3	11,4	7,4	43,7	43,0	11,1
<i>tx</i>	<i>tx</i>	8,3	11,0	8,6	8,5	8,4	10,4
<i>tx</i>	<i>tx</i>	14,7	9,0	7,2	8,6	10,6	8,9
<i>ts</i>	<i>tx</i>	8,2	8,2	8,2	8,2	8,3	8,2
<i>ts</i>	<i>tx</i>	8,2	8,2	8,3	8,8	8,9	8,2
<i>t_{xw}</i>	<i>t_{xw}</i>	9,3	9,2	9,1	9,2	9,2	9,2
<i>t_{xw}</i>	<i>t_{xw}</i>	9,7	9,3	10,2	9,2	9,2	9,3
<i>min</i>	<i>tu</i>	10,0	32,4	16,9	7,0	7,1	33,2
<i>min</i>	<i>tu</i>	56,1	20,2	9,9	55,2	55,3	19,3
<i>tx</i>	<i>tu</i>	10,1	9,2	11,1	9,1	9,2	9,3
<i>tx</i>	<i>tu</i>	20,3	10,0	8,4	11,0	10,5	9,7
<i>ts</i>	<i>tu</i>	9,3	9,2	9,1	9,2	9,2	9,2
<i>ts</i>	<i>tu</i>	9,7	9,3	10,2	9,2	9,2	9,3
<i>min</i>	<i>min</i>	34,9	32,9	27,4	35,4	35,9	32,8
<i>min</i>	<i>min</i>	38,8	32,3	23,5	38,8	39,2	31,2

Tabelle 6.3: Ausgewählte Ergebnisse ${}_aSG_s^m$, RMSE [s], jeweils zuerst GA dann GE je Repräsentationsart, KP 722, Monat Sep., SP 2

Ein	Ziel	SG 3	SG 5	SG 7	SG 13	SG 14	SG 24
<i>min</i>	<i>tx</i>	21,1	10,4	16,8	25,6	26,6	10,6
<i>min</i>	<i>tx</i>	45,4	14,3	19,8	46,0	46,3	14,5
<i>tx</i>	<i>tx</i>	43,2	13,1	20,4	56,4	56,9	12,6
<i>tx</i>	<i>tx</i>	69,7	23,6	26,2	75,5	74,3	23,8
<i>ts</i>	<i>tx</i>	51,7	13,9	33,1	59,1	59,2	13,5
<i>ts</i>	<i>tx</i>	75,9	26,3	42,7	75,9	75,9	26,3
<i>t_{xw}</i>	<i>t_{xw}</i>	50,7	50,1	65,4	52,5	52,5	49,9
<i>t_{xw}</i>	<i>t_{xw}</i>	49,1	50,4	65,5	50,9	50,9	50,4
<i>min</i>	<i>tu</i>	27,2	22,6	44,6	31,5	31,4	22,3
<i>min</i>	<i>tu</i>	18,7	23,8	44,1	23,1	23,0	24,3
<i>tx</i>	<i>tu</i>	49,1	50,0	48,4	52,5	52,5	49,8
<i>tx</i>	<i>tu</i>	44,2	50,4	49,7	50,2	50,5	50,4
<i>ts</i>	<i>tu</i>	50,7	50,1	65,4	52,5	52,5	49,9
<i>ts</i>	<i>tu</i>	49,1	50,4	65,5	50,9	50,9	50,4
<i>min</i>	<i>min</i>	330,1	342,9	352,9	339,6	334,9	340,8
<i>min</i>	<i>min</i>	326,9	344,0	353,9	339,0	340,0	351,4

Die verstärkte Häufung von Klassen für die Vorhersage der Umlaufsekunde t_x reduziert die Vorhersagequalität für stärker verkehrsabhängiges Schaltverhalten. Für den GA dieser SG, siehe Abbildungen 6.1, scheint Ähnliches in einer geringeren Ausprägung zuzutreffen. Dies erklärt die höhere KKR und auch die Ergebnisse für die RMSE, welche abhängig von der Stärke der Verkehrsabhängigkeit ansteigt. Die Verkehrsabhängigkeit wird hier über die Häufigkeitsverteilung der Klassen geschätzt, siehe Anhang C. Dies erklärt die Unterschiede zwischen den Ergebnissen für den RMSE der verschiedenen SG.

Für die Vorhersage der verbleibenden Zeit bis zum Schaltzeitpunkt bestehen aufgrund der Angabe als Countdown statt einer konstanten Umlaufsekunde mehr Klassen und weniger Häufungen dieser. Dies veranschaulichen die Abbildungen 6.5 bis 6.8 für LSA 002. Links ist die Vorhersage der Umlaufsekunde und rechts des Countdowns zu sehen.

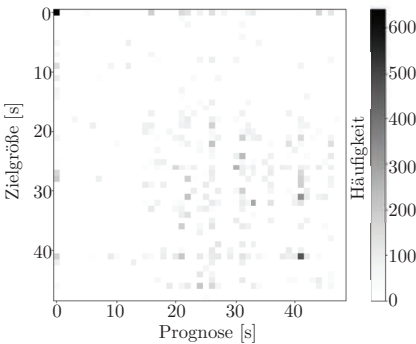


Abbildung 6.1: Konfusionsmatrix der Vorhersage von $t_x SG_{3,GA}$ KP 722 mit der Repräsentation min

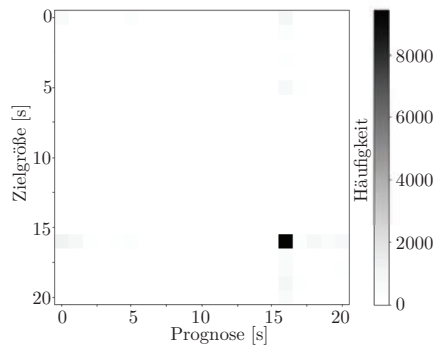


Abbildung 6.2: Konfusionsmatrix der Vorhersage von $t_x SG_{3,GE}$ KP 722 mit der Repräsentation min

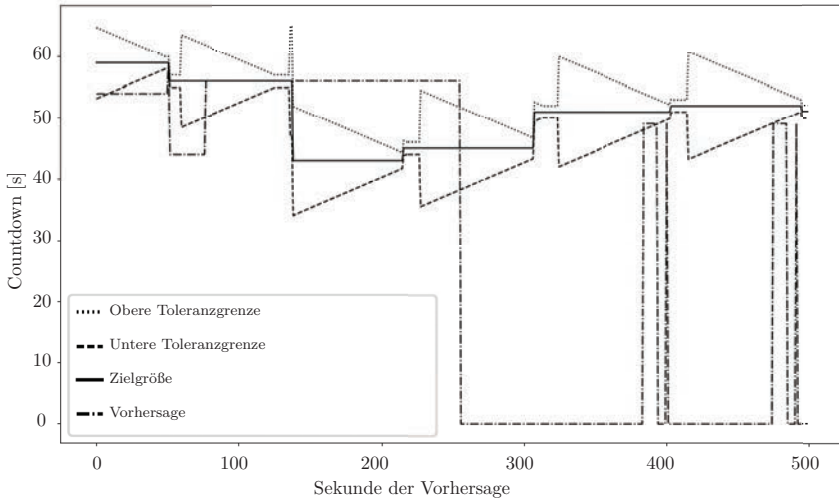


Abbildung 6.3: Vorhersage von $t_x SG_{3,GA}$ KP 722 von Sekunde 19380 bis 19880 mit der Repräsentation min

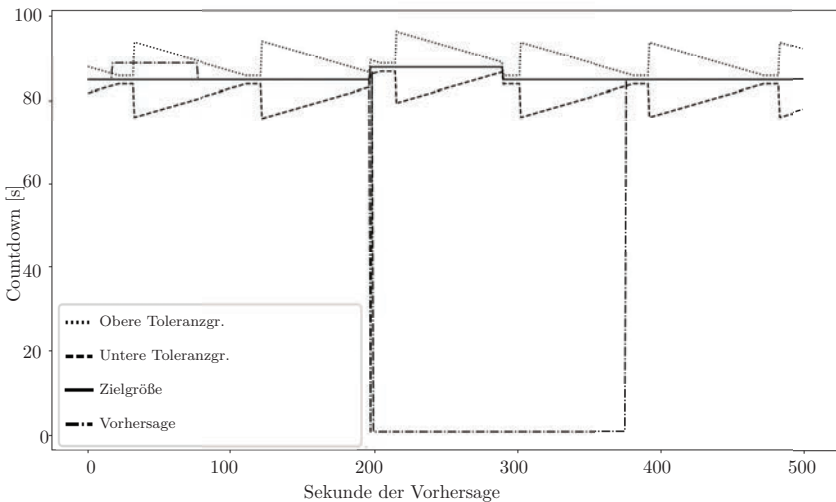


Abbildung 6.4: Vorhersage von $t_x SG_{3,GE}$ KP 722 von Sekunde 9690 bis 10190 mit der Repräsentation min

Für beide Konfusionsmatrizen ist eine unverwaschene Diagonale ein optimales Ergebnis. Für die Umlaufsekunden (t_x) ist eine Häufung von der Prognose 13s zu sehen. Auf der Diagonalen befinden sich nur wenige Vorhersagen. Im Gegensatz hierzu finden sich für die Repräsentation der Zeit bis zum nächsten Ereignis (t_u) die meisten Vorhersagen auf der Diagonalen, was die besonders dunkle Diagonale zeigt. Das Ausgrauen beziehungsweise Verwaschen zu den Seiten zeigt, dass Fehlprognosen im Regelfall nur um wenige Sekunden von der Zielgröße abweichen. Entsprechend wird die Prognose der Zeit bis zum Schaltzeitpunkt als zielführender angesehen als die Prognose der Umlaufsekunde.

Welche der Repräsentationen (min , t_s , t_x) für die Eingangsgrößen besonders sinnvoll ist, konnte aus den Ergebnissen nicht ermittelt werden. Dies scheint eine Optimierung für jede Zielgröße einzeln zu erfordern. Dieser Schritt kann zur Merkmalsauswahl gezählt werden, insofern die unterschiedlich formatierten Merkmale als einzelne Merkmale gezählt werden.

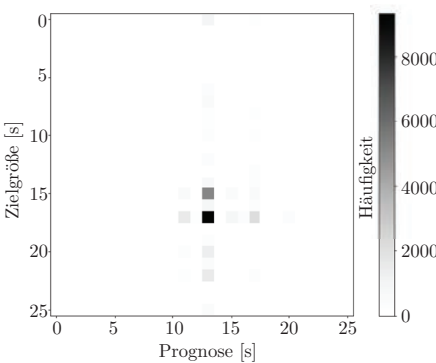


Abbildung 6.5: Konfusionsmatrix der Vorhersage von $t_x SG_{3,GA}$ KP 002 mit der Repräsentation t_u

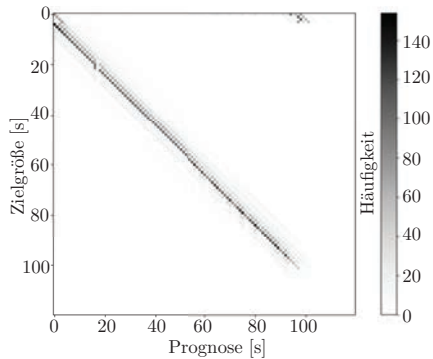


Abbildung 6.6: Konfusionsmatrix der Vorhersage von $t_u SG_{3,GA}$ KP 002 mit der Repräsentation t_x

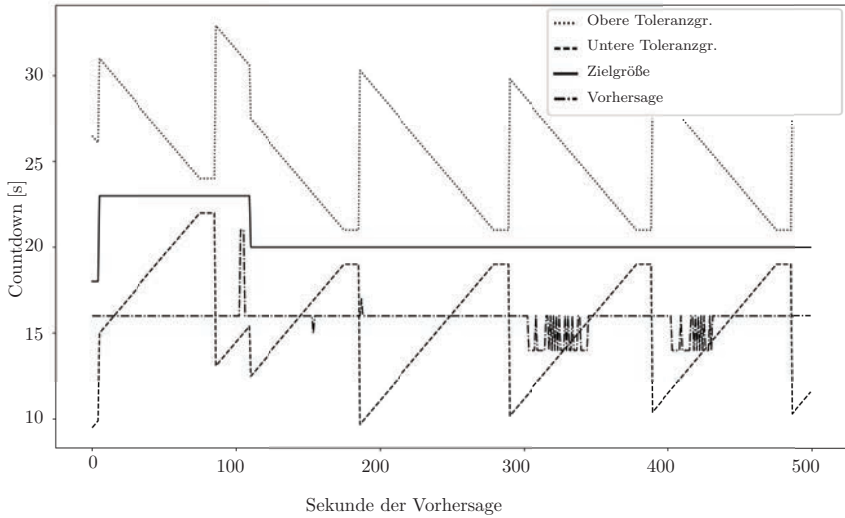


Abbildung 6.7: Vorhersage von $t_x SG_{3,GA}$ KP 002 von Sekunde 23211 bis 23711 mit der Repräsentation t_s

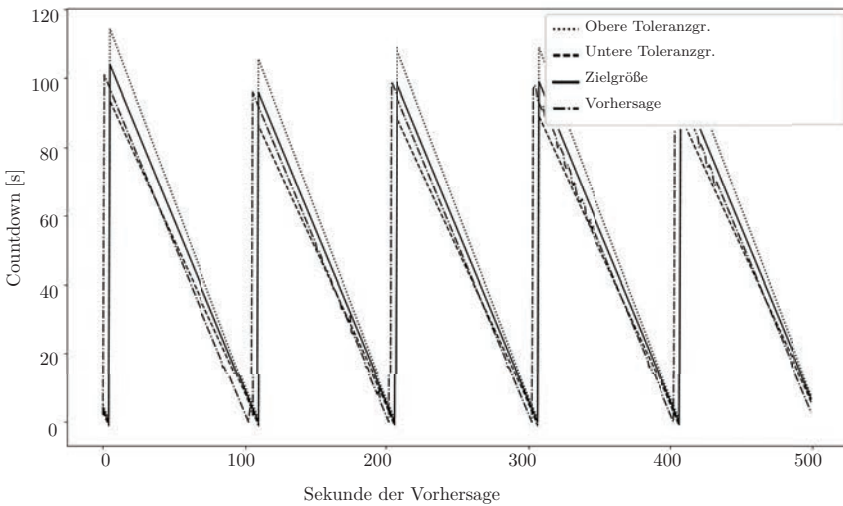


Abbildung 6.8: Vorhersage von $t_u SG_{3,GA}$ KP 002 von Sekunde 23211 bis 23711 mit der Repräsentation t_x

6.4 Vergangene Signale der eigenen SG vs. allen Signalen als Eingangsgrößen

In Annäherung an die Antwort zur Frage, wie viele und welche Informationen zu SG sinnvoll sind, in eine SZP integriert zu werden, wurden zunächst Experimente mit ${}_{ts}SG_s$ und ${}_{ts}SG$ miteinander verglichen. Eigene bedeutet hier, dass die Daten der vorhergesagten SG auch als einzige SG-Daten als Eingangsdaten verwendet werden im Gegensatz zur Integration der Daten aller SG als Eingangsdaten. Die Tabellen 6.4 und 6.5 für KP 722 sowie die Tabellen E.1 und E.2 im Anhang für den KP 002 zeigen die Ergebnisse. Besonders gute respektive schlechte Ergebnisse sind grün respektive rot gedruckt.

Tabelle 6.4: Ausgewählte Ergebnisse zu KP 722, Monat Sep., SP 2, ${}_{ts}SG_s$ (eigene), XG-Boost

SG	KKR [%]	RMSE [s]	Q_{PR}
1 GA	20	9	47
1 GE	22	10	37
2 GA	9	1298	66
2 GE	7	1131	64
3 GA	11	12	59
3 GE	43	5	23
4 GA	8	17	60
4 GE	10	16	56
5 GA	20	9	47
5 GE	22	10	37
6 GA	11	10	52
6 GE	44	5	22
7 GA	9	1385	65
7 GE	8	1253	63
9 GA	9	14	60
9 GE	42	8	24
10 GA	9	14	60
10 GE	42	8	24
13 GA	9	15	57
13 GE	43	10	23
14 GA	9	16	58
14 GE	43	10	23
15 GA	9	2562	93
15 GE	7	2227	91
16 GA	9	2562	93
16 GE	7	2227	91
24 GA	22	8	40
24 GE	21	10	41

Tabelle 6.5: Ausgewählte Ergebnisse zu KP 722, Monat Sep., SP 2, ${}_{ts}SG$ (alle), XG-Boost

SG	KKR [%]	RMSE [s]	Q_{PR}
1 GA	31	8	35
1 GE	29	9	32
2 GA	37	1298	30
2 GE	21	1131	41
3 GA	28	9	28
3 GE	46	4	21
4 GA	35	15	27
4 GE	20	14	43
5 GA	31	8	35
5 GE	29	9	32
6 GA	29	8	26
6 GE	47	4	21
7 GA	34	1385	33
7 GE	16	1253	48
9 GA	26	12	32
9 GE	47	7	21
10 GA	26	12	32
10 GE	47	7	21
13 GA	24	15	36
13 GE	47	9	22
14 GA	24	15	36
14 GE	47	9	22
15 GA	9	2557	92
15 GE	9	2234	91
16 GA	9	2557	92
16 GE	9	2234	91
24 GA	30	7	30
24 GE	29	9	32

Für beide KP ist zu sehen, dass die KKR für alle gezeigten Zielgrößen (SG) bei Integration aller SG höher ausfällt, während der RMSE und Q_{PR} sinken. Nur vereinzelt bleiben die Werte gleich. Dies betrifft insbesondere an KP 002 den RSME, da jedoch Q_{PR} sinkt, zeigt dies, dass zwar die gemittelte Abweichung gleich bleibt, der Prognoseverlauf allerdings häufiger innerhalb der Toleranzgrenzen verläuft.

Die hohen RMSE an KP 722 für SG 2, 7, 15 und 16 (OCIT-ID, nicht Nummer aus dem Lageplan) lassen sich über ihre geringen Verkehrsbelastungen erklären. SG 2 signalisiert einen rechtsabbiegenden Verkehrsstrom der Nebenrichtung, welcher aufgrund einer parallel verlaufenden Straße verhältnismäßig selten verwendet wird und SG 7 den entsprechend parallel entgegenkommenden Linksabbieger der Hauptrichtung. SG 15 und 16 sind gering frequentierte FG-Furten, die die Hauptrichtung queren. Parallel zur Hauptrichtung verlaufen die FG-Furten der SG 10, 13 und 14. Ähnlich wie für SG 2 an KP 722 verhält es sich mit SG 2 von KP 002, dem am östlichen Arm befindlichen Linksabbieger. Für diese Fahrtrichtung besteht ein alternativer Streckenverlauf über den Kreisel. Die SG 16 bis 19 betreffen die längste FG-Furt.

Der Qualitätsunterschied in der Prognose von GA und GE könnte auf vergleichsweise feste Gründauern zurückführbar sein. Nach dem GA einer SG steht ihr GE damit fest und auch der GA der wahrscheinlich folgenden SG bzw. Phase. Dies erklärt auch, weshalb nicht immer einheitlich die GA oder GE besser vorhergesagt werden können für einen KP.

Der Vergleich zwischen den Abbildungen 6.9 bis 6.14 veranschaulicht die Beobachtungen zu den Tabellen. Die farblich dargestellten Konfusionsmatrizen zeigen eine optimale Prognose, wenn eine gleichmäßig gefärbte Diagonale von unten rechts nach oben links zu sehen ist. Je weiter von der Diagonale die Datenpunkte entfernt sind, desto eher befinden sie sich außerhalb des Toleranzbereichs von 10%. Dieser ist nach oben links zulaufend. Vor diesem Hintergrund ist in den Abbildungen zu sehen, dass die Datenmodelle auf der rechten Seite, welche mehr Informationen enthalten, bessere Prognosen darstellen.

Mit Hilfe des Friedman-Nemenji-post-hoc-Tests konnte nur ein signifikanter Unterschied zwischen $tsSG_s$ und $tsSG$ für KP 002 bei Q_{PR} festgestellt werden. Weder für KKR noch für RMSE konnte für die beiden KP eine Signifikanz zwischen den beiden Datenmodellen festgestellt werden trotz der in den Tabellen zu sehenden Unterschiede. Dies lässt darauf schließen, dass sich zwar eine Verbesserung durch die Verwendung von mehr Informationen zu verschiedenen SG bewirken lässt, jedoch in den Datenmodellen noch zu viele doppelte bzw. irrelevante Informationen trotz der Robustheit des ML-Modells gegenüber solchen Informationen enthalten sind. Dass Informationen anderer SG ausschlaggebend für SZP sein können, ist naheliegend, da beispielsweise Nebenrichtungen erst freigegeben werden, wenn die Mindestfreigabezeit der Hauptrichtung erfüllt ist. Deshalb werden in folgenden Experimenten automatisierte Auswahlmethoden getestet, die diese Unterscheidung bezüglich (nicht) relevanter Information treffen sollen.

Die Abbildungen von E.1 bis E.8 decken sich mit den obigen Beschreibungen. Es ist zu sehen, dass die Prognosen häufiger innerhalb des Toleranzbereichs verlaufen. Für SG 2, dem Linksabbieger der Hauptverkehrsrichtung, ist zu sehen, dass starke Unregelmäßigkeiten wie beispielsweise das Auslassen eines Umlaufs vom ML-Modell mit beiden Datenmodellen nicht abgebildet werden können ebenso wie selten freigegebene SG wie beispielsweise SG 16. Die Vorhersage von SG 16 einer Furt für FG zeigt, dass mit dem umfangreicheren Datenmodell eher Prognosen im Bereich der Zielgröße erzielt werden, dass diese allerdings sehr stark schwanken. Selbige Beobachtungen lassen sich für SG 24 eines Radverkehrsweges treffen. Zudem können die Beobachtungen aller hier diskutierten Abbildungen ebenfalls für KP 002 gemacht werden.

Ein Unterschied zwischen der Prognosequalität bei der Verwendung von Daten von allen Wochentagen und nur Dienstag, Mittwoch, Donnerstag bzw. Wochenenden konnte bestimmt werden. Ein signifikanter Unterschied wurde nicht nachgewiesen. Die Ergebnisse von Vorexperimenten zu Merkmalen von Detektoren und ÖPNV-Meldungen haben zu vergleichbaren Erkenntnissen geführt, weshalb auf ihre Darstellung an dieser Stelle verzichtet wird.

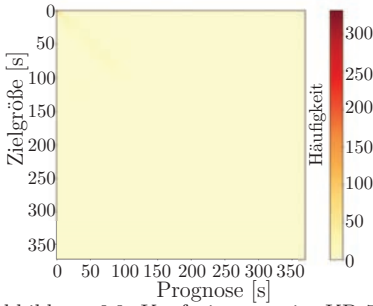


Abbildung 6.9: Konfusionsmatrix, KP 722, SG_s , SG 2 GE, SP 2

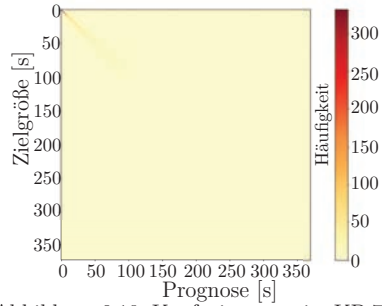


Abbildung 6.10: Konfusionsmatrix, KP 722, SG , SG 2 GE, SP 2

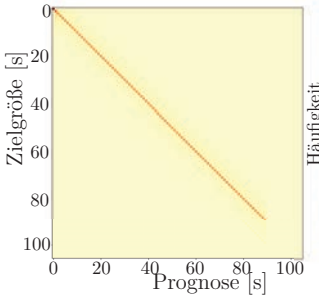


Abbildung 6.11: Konfusionsmatrix, KP 722, SG_s , SG 3 GE, SP 2

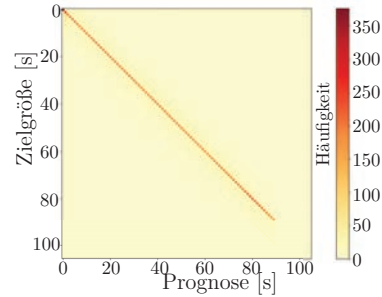


Abbildung 6.12: Konfusionsmatrix, KP 722, SG , SG 3 GE, SP 2

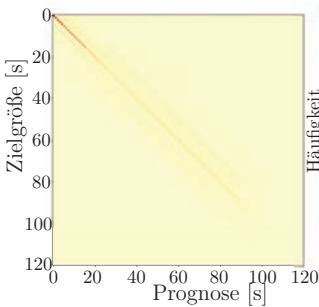


Abbildung 6.13: Konfusionsmatrix, KP722, SG_s , SG24 GE, SP2

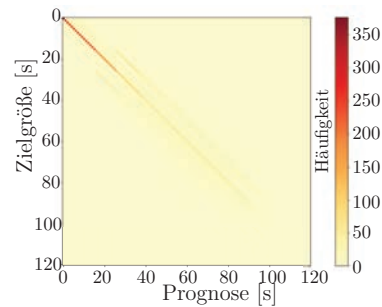


Abbildung 6.14: Konfusionsmatrix, KP722, SG , SG24 GE, SP2

6.5 Erkenntnisse zu den Experimenten der Merkmals-selektion

Zentrale Erkenntnisse der (Vor-)Experimente zur Merkmalsselektion der SG-, Detektor- und ÖPNV-Meldungs-Merkmale sind, dass für jede Zielgröße eine individuelle Auswahl an Merkmalen getroffen werden muss. Die Anzahl der benötigten Merkmale und damit das Perzentil sind für jede Zielgröße individuell und iterativ zu ermitteln. Hierbei spielen die Gruppen getrennt nach Verkehrsmittel weniger eine Rolle als der logische Zusammenhang der SG innerhalb der Phasen. So sind für jede Zielgröße umfangreiche, zeitaufwändige Versuchsreihen notwendig, um zielführende Datenmodelle bestimmen zu können. Dies ist bei gleichbleibendem Rechenaufwand mit Hilfe von Optimierungsalgorithmen automatisierbar.

Alle Selektionsmethoden haben für einzelne Zielgrößen Verbesserungen gegenüber dem keiner Selektion unterzogenen Datenmodell mit SG und Detektoren zeigen können. Es konnte jedoch keine Methode als besonders geeignet bestimmt werden. Besonders häufig wurden durch die Verwendung der PCA Verbesserungen die Ergebnisse erzielt. Auch Relief, der Fisher-Score, die Permutationswichtigkeit und CFS haben zum Teil Verbesserungen erwirken können. Mutual information, Chi-Quadrat und reine Korrelationsanalysen haben keine Verbesserungen gezeigt.

Welche Selektionsmethode für welche Zielgröße die größten Verbesserungen bewirken kann, wurde experimentell ermittelt, systematische Zusammenhänge wurden hierbei nicht aufgedeckt. Jedoch konnte festgestellt werden, dass die Verwendung von mehreren Selektionsmethoden zur iterativen Eingrenzung der Merkmalsauswahl zielführend ist. So wurden beispielsweise jeweils 50% der Merkmale ausgewählt, zunächst mittels Permutationswichtigkeit Merkmale ausgewählt und die Auswahl mittels mutual information verfeinert, um abschließend das Datenmodell mittels PCA oder Fisher-Score zu reduzieren. Auch in diesem Fall erscheint die optimale Abfolge der Methoden zielgrößen-spezifisch.

Vor dem Hintergrund der Ergebnisse aus Kapitel 7 wird der Ansatz, Detektorinformationen zielführend für SZP zu nutzen, darin gesehen, wie für die SG die Detektormerkmale mittels Einbettungen in die Datenmodelle zu integrieren, siehe auch Abschnitt 6.1.3. Sie können beispielsweise in einem Merkmal pro Knotenpunkt, Zufahrt oder Phase gebündelt werden.

Im Gegensatz hierzu zeigt die Integration der ÖPNV-Merkmale keine eindeutigen Ergebnisse bezüglich einer systematischen Verbesserung der SZP. Die Vorexperimente zu ÖPNV-Meldungen haben gezeigt, dass nicht alle Merkmale der ÖPNV-Meldungen gebraucht werden. Es wurden gleiche bis bessere Ergebnisse erzielt bei Beibehaltung von 25% der Merkmale: Eine KKR von ca. 40% bis 50% und ein RMSE von 5s bis 10s. Hierbei waren die AMLI-Informationen den Filtermethoden zufolge wichtiger als die anderen Merkmale der ÖPNV-Meldungen. Immer ausgewählt wurde die Zeit seit der Meldung. Es wird vermutet, dass eine Integration einer Ankunftszeitprognose des ÖPNV in das Datenmodell die Prognosequalität steigern könnte. Die Auswahl der Restfahrzeit zum Zeitpunkt

der Meldung in Kombination mit der Zeit seit dem Eingang der Meldung bestärken diese Vermutung. So wurden häufig zusätzlich zur Zeit seit der Meldung die Fahrplanabweichung, die gewünschte Phase und die Fahrzeit bis zur Haltlinie ausgewählt.

Auf die Darstellung der Ergebnisse von SZP unter Verwendung der Merkmalsselektion wird an dieser Stelle verzichtet, da keine signifikanten Verbesserungen ohne den zeitaufwändigen individuellen Iterationsprozess erzielt werden konnten und die oben beschriebenen Merkmale vor dem Hintergrund von Kapitel 7 vielversprechender erscheinen.

6.6 Experimente zu Merkmalen benachbarter Knoten-punkte

In diesem Abschnitt steht Forschungsfrage vier (Abschnitt 1.2), nach einer möglichen Verbesserung der Prognosequalität durch die Integration von Merkmalen benachbarter Knotenpunkte, im Vordergrund. Es werden Versuche für die SZP der Verkehrsströme des KP Katzensprung, welche vom KP Altmarkt zufließen, vorgestellt, welche durchnummeriert sind. Nach ihrer Entwicklung wurden die Experimente im Rahmen einer Bachelorarbeit durchgeführt, siehe [Bau21]. Abbildungen und Tabellen, die automatisiert von der in Abschnitt 4.3 vorgestellten Implementierung erstellt wurden, werden entsprechend ohne Quelle dargestellt.

6.6.1 Versuchsaufbau

Die erste Experimentreihe, um Frage vier zu beantworten, betrachtet den KP Katzensprung sowie Daten des benachbarten KP Altmarkt, siehe [Bau21]. Als Merkmale wurden SG-, DET- und ÖPNV-Merkmale von Verkehrsströmen, die auf den KP Katzensprung zufließen, integriert. Alle im Weiteren nicht erwähnten Einstellungen stimmen mit denen aus Abschnitt 6.2 überein. Aufgrund der geringen Menge an Experimenten wird auf die Verwendung eines Friedman-Nemenji-post-hoc-Tests verzichtet.

Tabelle 6.7 zeigt die Eingangsdatenmodelle der durchgeführten Experimente mit den grundsätzlich verwendeten Bezeichnern entsprechend den OCIT-Daten. Tabelle 6.6 zeigt den Zusammenhang mit den Bezeichnern der Signallagepläne auf, siehe Abbildung B.4 und B.10. Die ÖPNV-Meldungen werden über ihre Fahrtrichtung mit einbezogen; „slim“ bezeichnet in diesem Zusammenhang, dass Meldepunkt-, Linien- und Kursnummer sowie Zuglänge und Richtung Hand als Merkmale ausgeschlossen werden. Einzelne Experimente wurden mit Signalprogramm Nr. 1 wiederholt. Als Zielgrößen werden SG_3 und SG_4 am Katzensprung prognostiziert.

Tabelle 6.6: Zusammenhang zwischen OCIT-Bezeichnern und Signallageplänen für KP 002 und 003 [Bau21]

Katzensprung			Altmarkt		
Gruppe	OCIT	Signallageplan	Gruppe	OCIT	Signallageplan
SG	3	3_3a	SG	2	1R
SG	4	4_4a	SG	5	4_4a
			SG	8	7_7a
			SG	21	44
DET	16	D8_1B	DET	7	D3_1B
DET	17	D8_2B	DET	12	D4_2B
DET	18	D8_3B	DET	13	D4_3B
			DET	3	D1_3B

Tabelle 6.7: Eingangsdatenmodelle zur Vorhersage der Zielgrößen GA und GE für SG_3 und SG_4 des KP 003 [Bau21]

Versuch Nr.	Katzensprung			Altmarkt		
	SG	DET	SG	DET	ÖPNV	Prg.- Nr.
0	3, 4	16, 17, 18				2
0.1	3, 4					2
1	3, 4		2, 5, 8			2
2	3, 4		2, 5, 8	3, 7, 12, 13		2
3	3, 4	16, 17, 18	2, 5, 8	3, 7, 12, 13		2
3.1	3, 4	16, 17, 18	2, 5, 8			2
4	3, 4			3, 7, 12, 13		2
4.1	3, 4	16, 17, 18		3, 7, 12, 13		2
5	3, 4		21			2
6	3, 4				alle Richtungen	2
6.0	3, 4				0 - slim	2
6.1	3, 4				1 - slim	2
6.2	3, 4				2 - slim	2
6.3	3, 4				3 - slim	2
7.0	3, 4		2, 5, 8		0	2
7.1	3, 4		2, 5, 8		1	2
7.2	3, 4		2, 5, 8		2	2
7.3	3, 4		2, 5, 8		3	2
7.4	3, 4		21 ⁿ			2
8	3, 4	16, 17, 18				1
9	3, 4		2, 5, 8	3, 7, 12, 13		1
10	3, 4	16, 17, 18	2, 5, 8	3, 7, 12, 13		1

6.6.2 Ergebnisse der Vorhersage von SG_3

In Tabelle 6.8 sind die Ergebnisse zur Vorhersage von SG_3 zu sehen. SG_3 ist vom ÖPNV insoweit unbeeinflusst, als ihre Freigabe nicht von Phaseneinschüben unterbrochen wird. Die Fahrtrassen der Straßenbahn und von SG_3 verlaufen parallel.

Tabelle 6.8: Ergebnisse der Experimente zu SG_3

Versuch Nr.	GA		GE	
	KKR [%]	RMSE [s]	KKR [%]	RMSE [s]
0	17,17	5,45	13,13	11,06
0.1	15,88	5,54	11,58	7,87
1	19,98	10,65	12,78	7,95
2	21,91	6,93	16,31	11,03
3	21,29	10,23	16,54	10,81
3.1	16,59	6,13	13,41	8,01
4	16,76	5,47	13,48	11,38
4.1	16,70	5,43	13,73	10,80
5	21,30	5,07	12,81	7,71
6	18,94	11,62	16,69	10,23
6.0	16,29	6,47	13,42	12,98
6.1	15,42	5,39	12,03	11,48
6.2	16,20	5,61	12,27	11,46
6.3	16,24	7,52	19,57	13,57
7.0	20,01	14,00	22,64	15,70
7.1	23,32	13,77	4,13	28,93
7.2	22,94	13,76	8,95	24,17
7.3	23,36	14,22	10,29	26,38
7.4	19,35	13,84	24,83	15,25
8	17,86	5,96	40,72	6,55
9	29,43	6,23	41,65	6,67
10	29,46	6,09	41,62	6,67

Die Experimente zeigen, dass schlechtere Ergebnisse erzielt werden, wenn Detektordaten des KP Katzensprung verwendet werden, sie Versuch Nr. 0, als in Versuch Nr. 0.1, für welchen keine Detektordaten berücksichtigt werden. Dies kann darauf zurückzuführen sein, dass die Schaltzeitpunkte von SG_3 die Hauptverkehrsrichtung regeln beziehungsweise steuern und daher mit besonders hoher Verkehrsauslastung und wenig mittels Detektoren geregelt beziehungsweise gesteuert wird. Aufgrund der hierdurch vorliegenden hohen Gleichmäßigkeit der Freigaben und Sperrungen werden bereits bei der Verwendung der SG-Merkmale der prognostizierten SG des Knotenpunktes ein geringer RMSE und eine KKR erzielt, die nicht zufällig sein kann. Abbildung 6.15 zeigt den zugehörigen Prognoseverlauf.

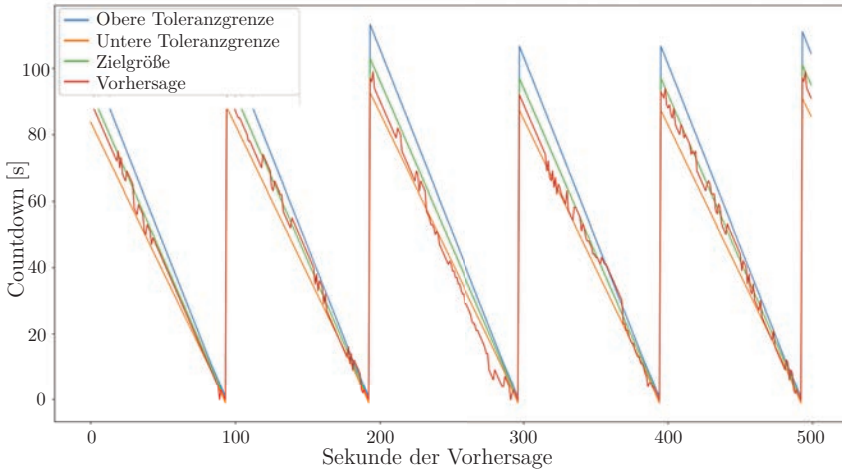
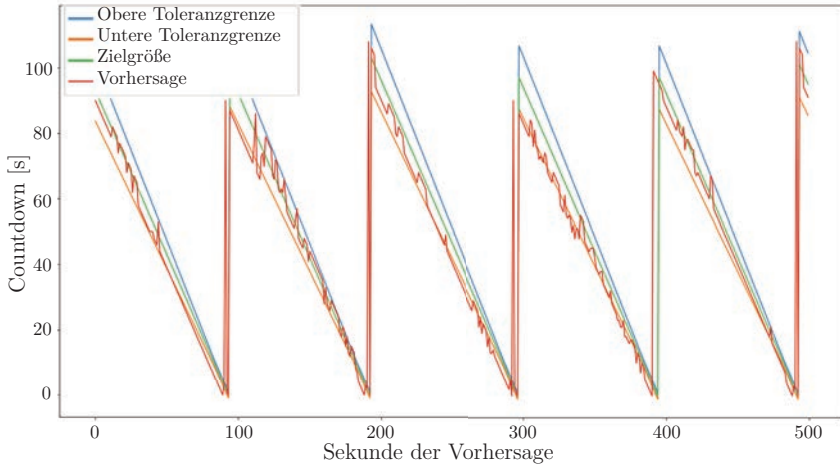


Abbildung 6.15: Vorhersage von $SG_{3,GA}$ von Sekunde 7000 bis 7500 zu Versuch 0

Bei Hinzunahme von SG-Daten des Altmarktes zeigt sich ein indifferentes Bild, wie die Ergebnisse der Versuche 1 bis 3.1 zeigen. Die KKR verbessert sich für GA insbesondere bei Hinzunahme zusätzlicher Detektordaten des Altmarkts, jedoch zeigt sich eine Verschlechterung der Prognosequalität im RMSE, siehe Abbildung 6.16.

Die Vorhersage wird mit den zusätzlichen Merkmalen zu früh getroffen. Entsprechend verschiebt sich die Prognose, siehe Abbildungen 6.16 und 6.17. Die größeren Häufigkeiten in den Nebendiagonalen, siehe Abbildung 6.17, sichtbar durch die stärkere Rotfärbung, veranschaulichen dies. Eine Interpretationsmöglichkeit hierfür ist, dass zwar der Zusammenhang erfasst wird, allerdings die Schätzung der Progressionszeit beim Lernen fehlschlägt. Das zuführende ÖPNV-Signal des Altmarktes, vergleiche mit Versuch 5, hat mit einer Verbesserung um mehr als 20% einen positiven Einfluss auf die Vorhersagequalität von $SG_{3,GA}$ des Katzensprungs, jedoch nicht auf die des GE.

Die Integration der Detektordaten des Katzensprungs in den Versuchen 2, 3, 4 und 4.1 zeigen in Verbindung mit Merkmalen der Signalgruppen in der KKR eine Verbesserung. Dies zeigt sich jedoch nicht im RMSE, welcher sich verschlechtert. Da es sich bei der Zielgröße um einen Hauptverkehrsstrom des KP während der Hauptverkehrszeit handelt, ist keine Verbesserung der Prognosequalität durch die Integration der Detektormerkmale zu erwarten, was die Ergebnisse bestätigen. Die ermittelten Wichtigkeiten, die in den Abbildungen I.1 und I.2 zu sehen sind, zeigen, dass die eigene SG und nachfolgend die Merkmale zu zuführenden SG wichtiger sind als die Merkmale der Detektoren.

Abbildung 6.16: Vorhersage von $SG_{3,GA}$ von Sekunde 7000 bis 7500 zu Versuch 1

Um diese Vermutung zu bestätigen, wurden in den Versuchen 8 bis 10, die Experimente mit Signalprogramm Nr. 1 wiederholt. Für den GA konnte durch die Integration der Detektormerkmale die Prognosequalität verbessert werden, ohne dass sich der RMSE verschlechtert. Allerdings wird auch hier den SG-Merkmalen der größte Einfluss zugeschrieben, siehe Abbildung I.3.

Meldungen des ÖPNV bewirken unabhängig von der Fahrtrichtung bei vollständiger Integration (Versuche 7.x) eine Verbesserung der KKR. Es ist jedoch zu beobachten, dass sich der RMSE verschlechtert. Das zuvor beschriebene Vorseilen der Prognose um ca. 5 Sekunden zeigt sich in diesen Versuchen ebenfalls. Bei unvollständiger Integration der ÖPNV-Meldungen (Versuche 6.x) ist keine Verbesserung der KKR zu beobachten, sondern lediglich ein höherer RMSE.

Auch in diesen Versuchen sind einzelne SG die wichtigsten Merkmale, siehe Abbildungen 6.18 und 6.19. Auffällig ist, dass einige Merkmale der Prognosequalität sogar abträglich sind, was sich in negativen Wichtigkeiten zeigt.

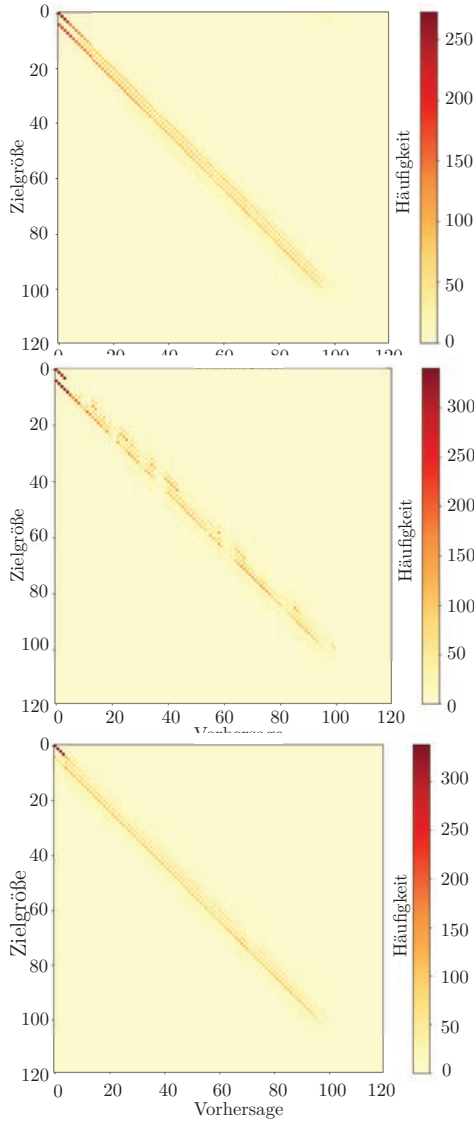


Abbildung 6.17: Konfusionsmatrizen zu Versuch 1, 3.1 und 3 zur Vorhersage von $SG_{3,GA}$

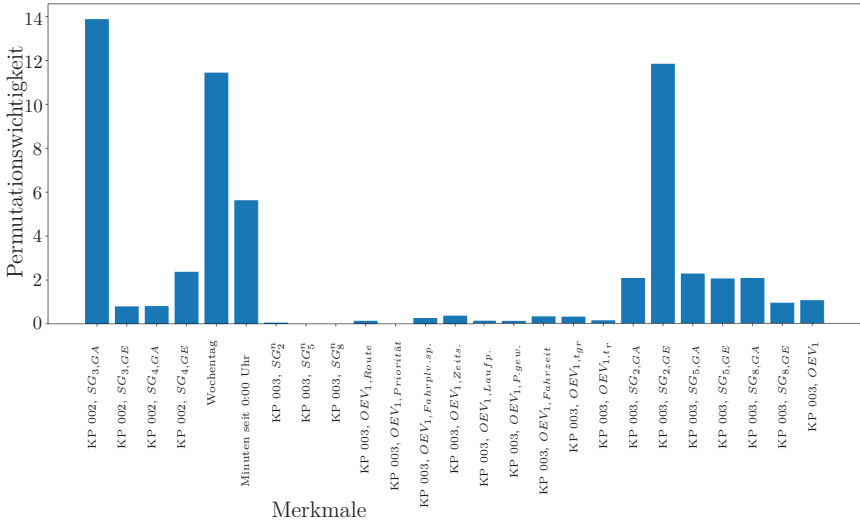


Abbildung 6.18: Permutationswichtigkeit zu Versuch 7.1 zur Vorhersage von $SG_{3,GA}$

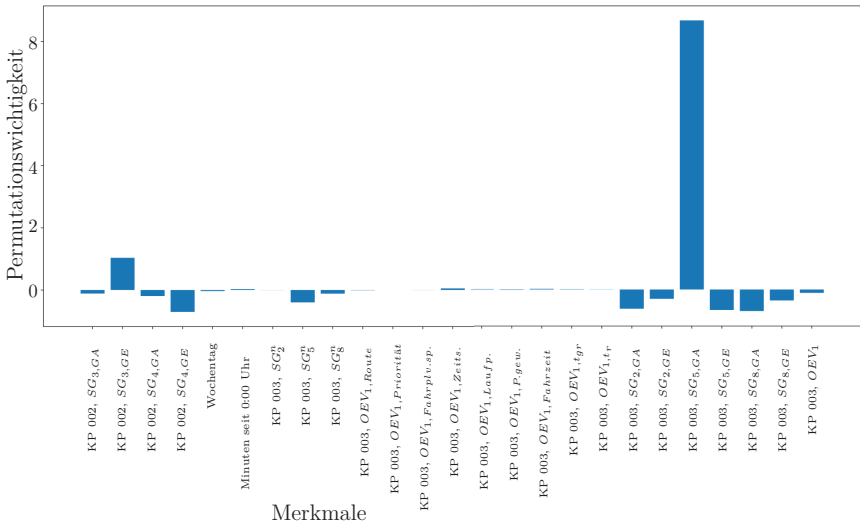


Abbildung 6.19: Permutationswichtigkeit zu Versuch 7.1 zur Vorhersage von $SG_{3,GE}$

6.6.3 Ergebnisse der Vorhersage von SG_4

Die Ergebnisse für die Vorhersage von SG_4 zeigt Tabelle 6.9. In der Tabelle ist zu sehen, dass in Versuch 0 mit Detektordaten geringfügig schlechtere Ergebnisse erzielt werden als ohne Detektordaten. Dies kann darauf zurückzuführen sein, dass die Schaltzeitpunkte von SG_4 vermutlich eher von anderen Phasen am Katzensprung und ÖPNV-Anforderungen abhängig sind als von den Anforderungen vor der eigenen Haltlinie.

Tabelle 6.9: Ergebnisse der Experimente zu SG_4

Versuch Nr.	GA		GE	
	KKR [%]	RMSE [s]	KKR [%]	RMSE [s]
0	18,51	8,83	30,23	12,45
0.1	14,71	9,48	19,37	7,85
1	18,50	9,20	22,05	13,23
2	20,32	11,39	23,13	12,96
3	20,27	12,67	23,29	13,23
3.1	20,71	12,78	23,51	12,76
4	18,24	10,28	20,65	11,29
4.1	18,75	10,51	20,58	11,94
5	17,28	9,55	19,52	9,70
6	14,69	5,84	18,48	12,35
6.0	14,79	10,30	21,83	11,47
6.1	15,53	9,90	24,55	12,96
6.2	14,39	10,60	19,00	9,58
6.3	17,29	10,48	27,69	12,66
7.0	23,39	12,90	32,34	14,20
7.1	22,27	14,32	33,84	14,15
7.2	24,09	14,15	33,52	14,18
7.3	23,41	14,92	32,89	15,03
7.4	23,76	13,90	32,56	14,18
8	37,47	5,20	48,33	6,55
9	43,77	5,14	50,00	4,95
10	43,31	5,08	49,98	4,87

Die Versuche zeigen zudem auf, dass Unregelmäßigkeiten, wie beispielsweise Phaseneinschübe, siehe Abbildung 6.20, nicht erkannt werden, sondern die Vorhersage wie für die häufiger auftretenden, vergleichsweise regelmäßigen Schaltzeitpunkte beibehalten wird. Der Phaseneinschub wird durch die Anforderung eines ÖPNV-Fahrzeugs erzeugt. SG_4 quert den Fahrweg der Straßenbahn.

Bei Hinzunahme von SG-Daten des Altmarktes verschlechtert sich die Prognosequalität, siehe Versuche 1 bis 3.1, obwohl eine Koordinierung vorliegt. Nur die KKR verbessert sich für GA, alle weiteren Evaluationsgrößen zeigen eine Verschlechterung. Dies zeigt sich

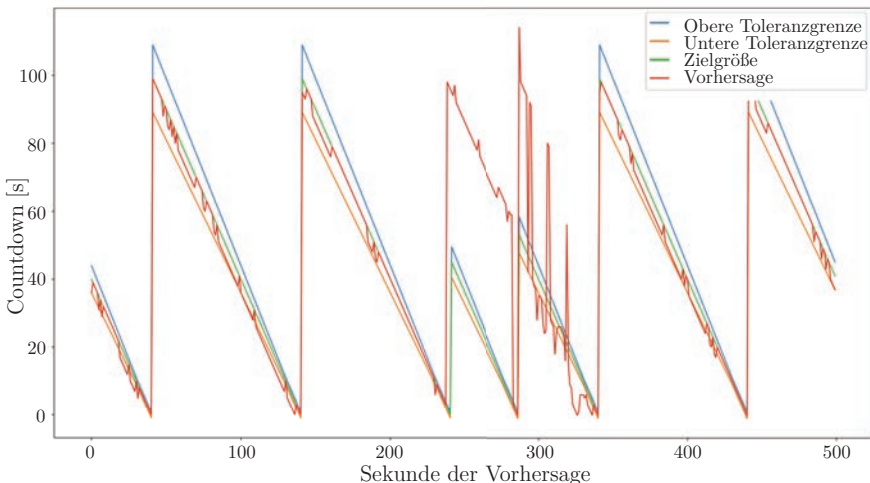


Abbildung 6.20: Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 0

insbesondere darin, dass der GA häufiger um ca. 5 Sekunden zu früh vorhergesagt wird. Entsprechend verschiebt sich die gesamte Prognose, siehe auch Abbildung 6.21.

Das zuführende ÖPNV-Signal des Altmarktes, vergleiche Versuch 5, hat einen Einfluss auf die Vorhersage von SG_4 des Katzensprungs, siehe Abbildung 6.22. Die Prognose verbleibt deutlicher innerhalb des Toleranzbereichs von 10%; Phaseneinschübe werden weiterhin nicht erkannt. Jedoch ist für KRR und RMSE keine Verbesserung zu vermerken. Die geringe Verbesserung kann auf die vor dem Katzensprung gelegene Haltestelle zurückzuführen sein. Die Haltedauer an der Haltestelle und damit die Progressionszeit zwischen den ÖPNV-Signalen zwischen Altmarkt und Katzensprung sind nicht konstant.

Die Integration der Detektordaten des Katzensprungs in den Versuchen 2, 3, 4 und 4.1 zeigen vergleichbare Ergebnisse wie Versuch 0, wobei der RMSE für das GE steigt, siehe Abbildung I.7. Der geringe Einfluss der Detektordaten kann auf das hohe Verkehrsaufkommen zurückgeführt werden, wodurch regelmäßig das Erreichen der maximalen Wartezeiten an den Signalgebern den Schaltzeitpunkt bestimmt und weniger Zeitlückenkriterien und Belegungen. Der Einfluss der Merkmale, ermittelt mittels Permutation, ist in den Abbildungen I.4, I.5 und I.6 zu sehen.

Meldungen des ÖPNV bewirken unabhängig von der Fahrtrichtung bei vollständiger Integration (Versuche 7.X) eine Verbesserung der Prognosequalität, vergleiche Abbildungen 6.23 und I.8. Phaseneinschübe können hiermit besser erkannt werden. Die Prognose verläuft mit erheblich weniger Schwankungen, siehe Abbildung 6.23 und 6.24. Das zuvor beschriebene Vorauseilen der Prognose um ca. 5 Sekunden zeigt sich teilweise auch in diesen Versuchen, was den erhöhten RMSE erklären könnte. Verbesserungen durch eine teilweise Integration der ÖPNV-Meldungen des Altmarktes zeigen sich nicht in den Metriken.

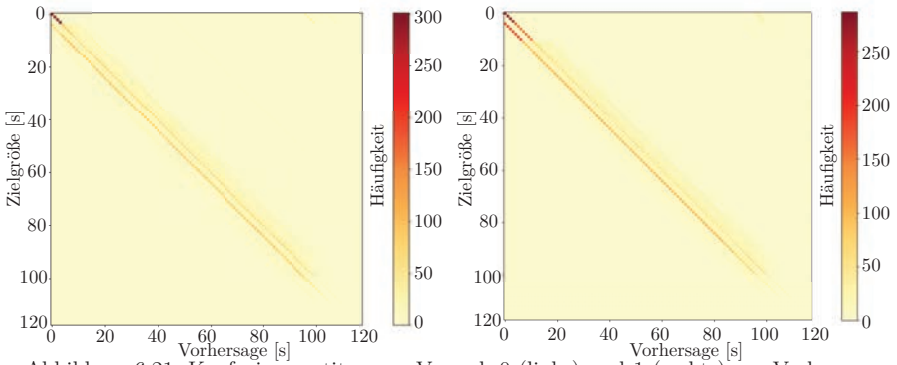


Abbildung 6.21: Konfusionsmatrizen zu Versuch 0 (links) und 1 (rechts) zur Vorhersage von $SG_{4,GA}$

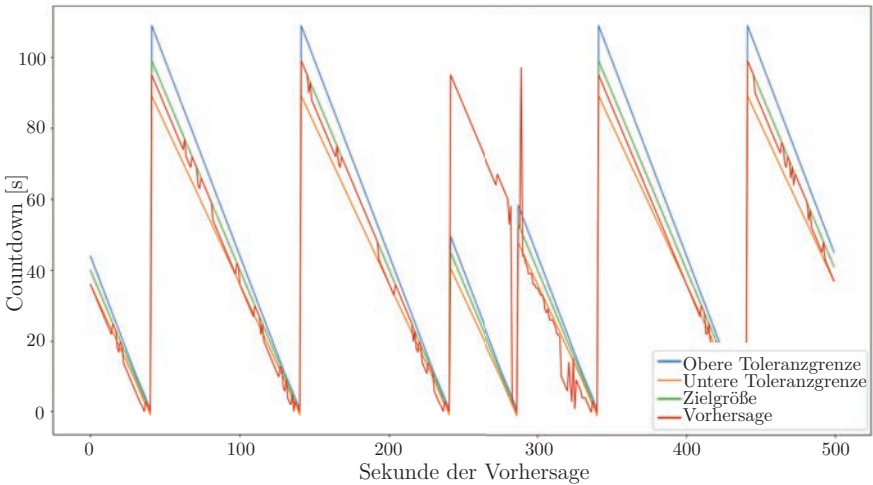
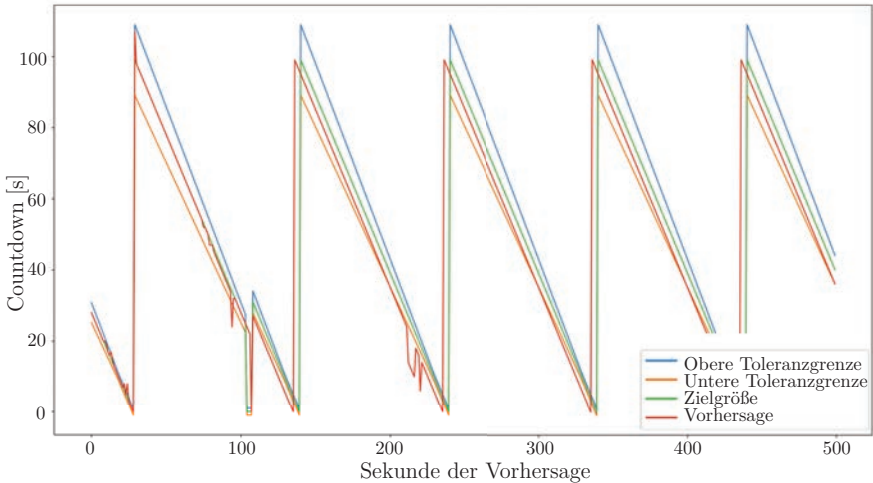
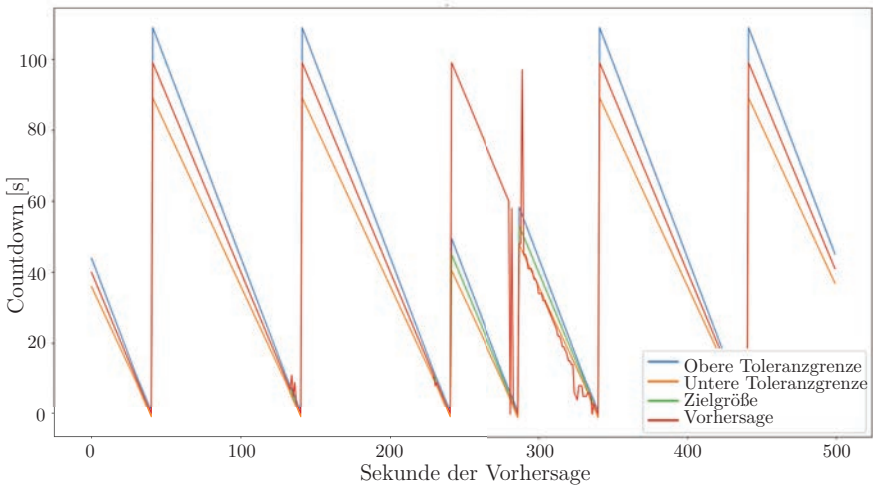


Abbildung 6.22: Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 5

Abbildung 6.23: Vorhersage von $SG_{4,GE}$ von Sekunde 18500 bis 19000 zu Versuch 7.1Abbildung 6.24: Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 7.4

6.7 Zwischenfazit

Die Prognose mit der Repräsentation als Zeit bis zum Schaltzeitpunkt als Zielgröße wird als zielführender angesehen als die Prognose der Umlaufsekunde oder der Uhrzeit. Bezüglich der Eingangsgrößen ist die optimale Repräsentation der Merkmale signalgruppenspezifisch zu ermitteln. Zwecks Vereinfachung werden die Sekunden seit dem letzten Signalwechsel (ts) für die weiteren Experimente verwendet.

Betreffend der automatisierten Merkmalsselektion sind die Datenmodelle iterativ knotenpunktspezifisch und zielgrößenspezifisch zu selektieren. Die beste Methode zur Merkmalsselektion ist experimentell zu ermitteln. Als oft auftretende beste Selektionsmethoden haben sich PCA, Permutationswichtigkeit sowie Relief und Fisher-Score gezeigt. Besonders zielführend war es für die verschiedenen Iterationen, verschiedene Selektionsmethoden zu verwenden.

Es lässt sich feststellen, dass die Integration einzelner Informationen von benachbarten Knotenpunkten zwar punktuelle Verbesserungen erwirken kann, jedoch generelle Verbesserungen nicht erzielt werden können. Die höchste Relevanz zeigen hierbei Signalgruppendaten. Sie verbessern die Vorhersagequalität insbesondere bei stark variierenden Freigabezeitpunkten. Hingegen erscheinen Detektor- und ÖPNV-Daten meist nicht hilfreich oder gar kontraproduktiv.

Da sich in mehreren Versuchen gezeigt hat, dass der zeitliche Versatz bedingt durch die zugehörigen Reisezeiten zwischen den beiden Knotenpunkten ein Problem darstellen könnte, erscheint es sinnvoll, diesen zu beheben und einige der Versuche erneut durchzuführen, um zu überprüfen, ob das zu einer Verbesserung der Ergebnisse führt. Ferner bleibt offen, ob die Wahl eines anderen Signalprogramms die Relevanz der Detektordaten steigert.

Insgesamt hat sich gezeigt, dass die im folgenden Kapitel 7 gezeigten Einbettungen nicht nur für SG, sondern auch für Detektormerkmale, die am betrachteten KP aufgenommen werden, besonders zielführend für die Qualitätssteigerung von SZP erscheinen, unabhängig von den Erkenntnissen zu Detektormerkmalen benachbarter Knotenpunkte. Dies betrifft auch die Integration von Merkmalen benachbarter Knotenpunkte. Die zusätzliche Integration dediziert zielgrößenspezifisch ermittelter Merkmale steigert die Prognosequalität im Gegensatz zur Integration aller verfügbaren Merkmale.

Bezüglich der Systemanalyse lässt sich festhalten, dass es sinnvoll erscheint, den Grad der Verkehrsabhängigkeit beziehungsweise des Festzeitschaltverhaltens jeder SG zu ermitteln, um die Eignung einer Vorhersage über die Umlaufsekunde einschätzen zu können. Hierfür könnten beispielsweise die Lagepläne des betrachteten sowie der benachbarten KP in Verbindung mit bekannten Verkehrsbelastungen hinzugezogen werden. Hilfreich hierbei könnte sein, die Haupt- und Nebenrichtung sowie SG des ÖPNV oder andere SG, welche nur auf Anforderung schalten, zu identifizieren.

Kapitel 7

Experimente zur phasenorientierten Vorhersage

In diesem Kapitel werden Antworten auf die sechste Forschungsfrage aus Abschnitt 1.2 diskutiert, welche die Vorhersage von Phasen beziehungsweise Phasenfolgen adressiert. Zudem werden weitere Antworten bezüglich der vierten Frage, die Integration von Informationen von benachbarten KPen betreffend, vorgestellt. Im ersten Schritt zur Beantwortung dieser Fragen werden in Abschnitt 7.1 Merkmale zu Phasen und Phasenübergängen vorgestellt. In Abschnitt 7.3 ist der Versuchsaufbau mit Datenmodellen dokumentiert. Ihre Ergebnisse werden in den Abschnitten 7.4 bis 7.6 diskutiert. Zugehörige Schlussfolgerungen sind in Abschnitt 7.7 beschrieben.

7.1 Ermittlung der Phasenmerkmale

Die in Abschnitt 6.1.2 vorgestellten Merkmale werden an dieser Stelle um Phasenmerkmale erweitert. In den OCIT-Protokollen ist die Phaseninformation nicht enthalten. Entsprechend ist sie für die Phasenmerkmale zu rekonstruieren. Im Merkmal für den Signalisierungszustand (engl. *state*) wird der Gesamtzustand der SG in einem eindeutigen Skalar zusammengefasst, welches mittels weiteren Algorithmen in ein Merkmal der Phasenfolge überführt werden kann, siehe auch Abbildung 7.1.

Die Algorithmen basieren auf den Erkenntnissen aus [Wei19a] und [HHS06] zur automatisierten Identifikation von Phasen aus KP-Prozessdaten. Die dort beschriebene Methodik fokussiert zweiphasige Systeme, weshalb in diesem Abschnitt Erweiterungen der Methodik aufgeführt sind. Zentrale Ideen der Erweiterungen sind

- der Abgleich mit der Phasendefinition der Dokumentationen,
- die Einführung einer Konsistenzprüfung und
- das Entwickeln einer statistischen Prüfung der erkannten Phasen(-übergänge).

Eine ausführliche Dokumentation zur Umsetzung dieser Ideen findet sich in der im Rahmen dieser Dissertation betreuten Masterarbeit und zugehörigen Veröffentlichungen [Hec21, HSH22, HSH23]. Abbildungen und Tabellen, die automatisiert von der in Abschnitt 4.3 vorgestellten Implementierung erstellt wurden, werden entsprechend ohne Quelle dargestellt, obwohl sie bereits von Heckmann verwendet wurden. Evaluationen der Merkmalsidentifikation sind in Abschnitt 7.2 beschrieben.

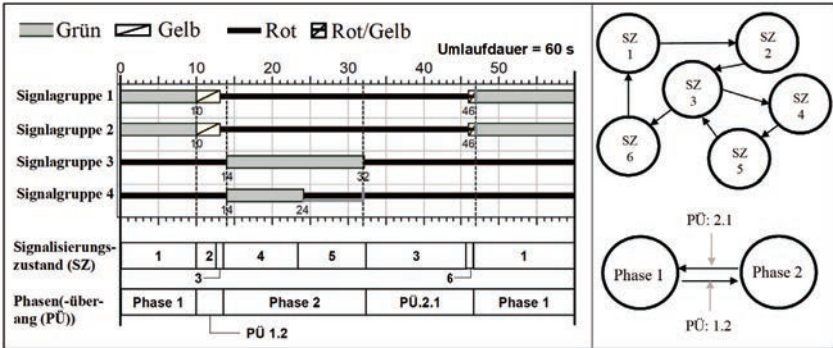


Abbildung 7.1: Veranschaulichung des Zusammenhangs zwischen SG-Signalen, Signalisierungszustand (SZ) und Phasen(-übergängen), gemeinsam erarbeitet für [HSH22]

Eine zentrale Herausforderung besteht darin, Phasen und Phasenübergänge voneinander zu unterscheiden. Phasen kennzeichnen einen für längere Zeit gehaltenen Signalisierungszustand, wobei die minimale Phasendauer in Kassel 0 Sekunden beträgt, da die Mindestfreigabezeit von SG oft in die Phasenübergänge integriert ist. Die Empfehlung der RiLSA für die Mindestfreigaben von SG ist 5 s. Phasenübergänge können mit kurzzeitig aktiven Festzeitprogramme verglichen werden, da sie über eine feste Abfolge von vergleichsweise kurz gehaltenen Signalisierungszuständen, deren Dauer innerhalb eines Phasenübergangs wiederkehrend konstant ist, verfügen. In Kassel ist es möglich, dass Phasenübergänge zugunsten anderer abgebrochen werden. Der Beginn eines Phasenübergangs kann mit anderen Phasenübergängen übereinstimmen, weshalb zu Beginn eines Phasenübergangs dieser nicht eindeutig identifiziert werden kann.

Es wird angenommen, dass für eine hohe Prognosequalität die aus den Prozessdaten rekonstruierten Phasen und Phasenübergänge die tatsächlichen Phasen und Phasenübergänge möglichst genau nachbilden sollten.

Um die Phasen identifizieren zu können, ist zunächst der gesamte Signalisierungszustand der KP zu erfassen. Hierzu wird zunächst ein Merkmal für diesen Signalisierungszustand berechnet auf dessen Basis die Differenzierung in Phasen und Phasenübergänge stattfinden kann. Diese Differenzierung findet unter anderen mittels des in [We17] erläuterten Algorithmus nach Weidemann statt, für dessen Evaluation ein Vergleichsalgorithmus vorgestellt wird. Zudem werden der bekannte Algorithmus optimiert, ein Phasenspeicher sowie ein statistischer Filter ergänzt.

7.1.1 Merkmal des Signalisierungszustands

Das skalare Merkmal für den Signalisierungszustand (engl. *state*) wird ermittelt, indem, wie bereits in [HHS04] vorgeschlagen, zunächst ein Vektor für den Signalisierungszustand gebildet wird. Die Anzahl aller Signalgruppen ist S . Dieser Vektor $\underline{SG}(k)$ umfasst den Zustand aller $SG_s | s \in S$ zum Zeitpunkt k :

$$\underline{SG}(k) = \begin{bmatrix} SG_1(k) \\ \dots \\ SG_s(k) \\ \dots \\ SG_S(k) \end{bmatrix} \quad | s \in 1, \dots, S \quad SG_i(k) \in 0, 1. \quad (7.1)$$

Der Vektor des Signalisierungszustands wird mit einem Signalgruppenvektor s multipliziert:

$$z(k) = \underline{SG}(k) \cdot \underline{EX}^s, \quad (7.2)$$

um einen Skalar für das Merkmal des Signalisierungszustands $z(k)$ zu erhalten. Der Signalgruppenvektor \underline{EX}^s setzt sich aus den Exponenten der Indizes s der Signalgruppen zur Basis 2 zusammen:

$$\underline{EX}^s = \begin{bmatrix} 2^1 \\ \dots \\ 2^s \\ \dots \\ 2^S \end{bmatrix} \quad | s \in 1, \dots, S. \quad (7.3)$$

Abbildung 7.2 zeigt hierfür eine Beispielrechnung. Zum Zeitpunkt $k = 56$ sind SG_1 und SG_2 freigegeben und SG_3 sowie SG_4 gesperrt. Es ergibt sich der Skalar 6 als Signalisierungszustandsmerkmal zum Zeitpunkt k .

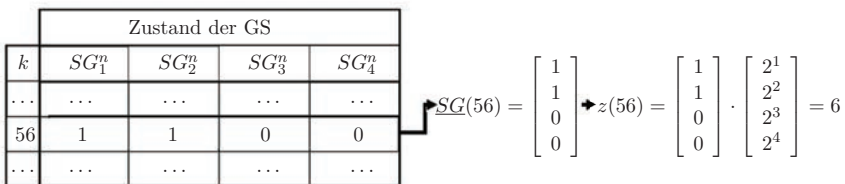


Abbildung 7.2: Beispielrechnung für das Merkmal Signalisierungszustand

7.1.2 Phasenmerkmal nach Weidemann

Als Grundlage für die im Folgenden entwickelte Methode zur Identifikation des Phasenmerkmals wird der Algorithmus nach [We17] (Abkürzung: Wei/weide) verwendet und an dieser Stelle kurz dargestellt. Abweichend von diesem Vorgehen werden die Merkmale des Signalisierungszustands bereits in den ersten Schritten berechnet. Er wird in Abbildung 7.3 veranschaulicht. Vom Algorithmus als Phasen identifizierte Signalisierungszustände werden in den Vektor des Phasenmerkmals übertragen. Alle anderen Zeitpunkte des Phasenmerkmals werden mit einer Zahlenkombination für den Phasenübergang belegt: $[z(k - a), 00100, z(k + b)]$, wobei $k - a$ beziehungsweise $k + b$ für den Endzeitpunkt der letzten respektive Anfangszeitpunkt der nächsten Phase stehen. Die Zustandsabfolgen werden zusammen mit der Zahlenkombination für den Phasenübergang separat vermerkt.

Im Folgenden wird das Vorgehen näher erläutert. Zuerst wird in Schritt 2 für den Zeitpunkt der Zustandsvektor $\underline{SG}(k)$ und das zugehörige Merkmal $z(k)$ für alle Zeitpunkte k (Schritt 1) ermittelt und in Schritt 3 der Merker für einen Phasenbeginn zurückgesetzt.

Hiernach werden die Stichproben rekursiv durchgegangen, siehe Schritt 4. Ist das Merkmal für den aktuellen und letzten Zeitpunkt identisch, wird in Schritt 5 zur nächsten Iteration übergegangen (Schritt 11). Falls sie sich unterscheiden, wird in Schritt 6 geprüft, ob die zugehörigen Vektoren sich um mindestens eine Freigabe unterscheiden. Ist dies der Fall wird der Zeitpunkt in Schritt 7 im Merker für den Phasenbeginn vermerkt und eine neue Iteration begonnen.

Wenn die Prüfung in Schritt 6 keine Freigaben bestätigt, zuvor allerdings eine Zustandsänderung festgestellt wurde, wird in Schritt 8 geprüft, ob bereits eine Phase identifiziert wurde. In diesem Fall werden in Schritt 9 für den vorangegangenen Zeitpunkt ein Phasenende sowie Phasenbeginn und das Merkmal vermerkt. In Schritt 10 wird deshalb der Merker für Phasen zurückgesetzt. Ansonsten wird der nächste Zeitpunkt betrachtet.

Im Zusammenhang dieses Algorithmus kann ein Phasenende, also ein Beginn eines Phasenübergangs keine Freigaben, sondern ausschließlich Sperrungen enthalten, um erkannt zu werden. Freigaben werden als möglicher Phasenanfang identifiziert.

Der Algorithmus von Weidemann erkennt Phasenübergänge. Direkt aufeinander folgende Phasen werden nicht erkannt, da das Phasenende der ersten Phase nicht identifiziert werden kann, siehe oben. Es werden zur ersten der beiden Phasen keine Daten gespeichert, da das Phasenende durch eine Freigabe gekennzeichnet wurde. Sie wird so einem Phasenübergang zugeordnet, da die Prüfung in Schritt 6 bereits bejaht und die Schritte 8 und 9 für diesen Zeitpunkt nicht erreicht werden. Diese werden erst am Ende der zweiten Phase erreicht, falls Zustände eines Phasenübergangs folgen. Ein solcher Phasenwechsel kann beispielsweise durch eine Anforderung durch FG erfolgen, deren Freigabe in der aktuellen Phase „hinzugeschaltet“ werden kann, was in der Logik meist einem Phasenwechsel entspricht. Ebenso werden Phasen, welche durch Signalfreigaben beginnen, nicht erkannt.

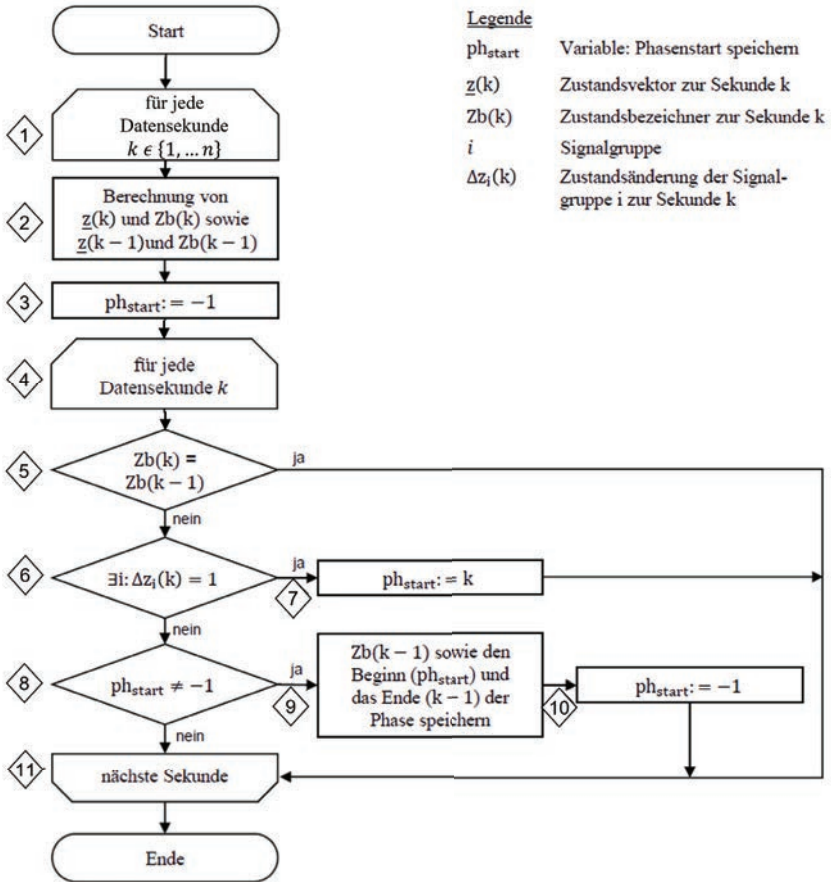


Abbildung 7.3: Programmablauf für die Phasenidentifikation nach [We17] [Hec21]

7.1.3 Phasenmerkmal nach Weidemann, ergänzt

Die im vorangegangenen Absatz besprochenen Schwächen sollen durch die Ergänzung von drei Schritten behoben werden, siehe Abbildung 7.4. Dieser erweiterte Algorithmus wird im Folgenden auch mit Wei+ bezeichnet. Der Ablauf ist bis auf eine Änderung gleich: Wo in Abbildung 7.3 in Schritt 7 der Startzeitpunkt einer Phase vermerkt wurde, werden in die neuen Schritte 7 bis 9 nun zusätzlich SG-Freigaben (Schritt 7) und eine minimale Phasendauer von 4 Sekunden (Schritt 8) überprüft. Wird durch einen solchen gleichzeitigen Freigabe und Sperrung von SG ein direkter Phasenwechsel identifiziert, werden in Schritt 10 die entsprechenden Daten der Phase gespeichert und der Beginn der nächsten Phase vermerkt. Dieser Schritt 7 aus Abbildung 7.3 entspricht nun Schritt 10.

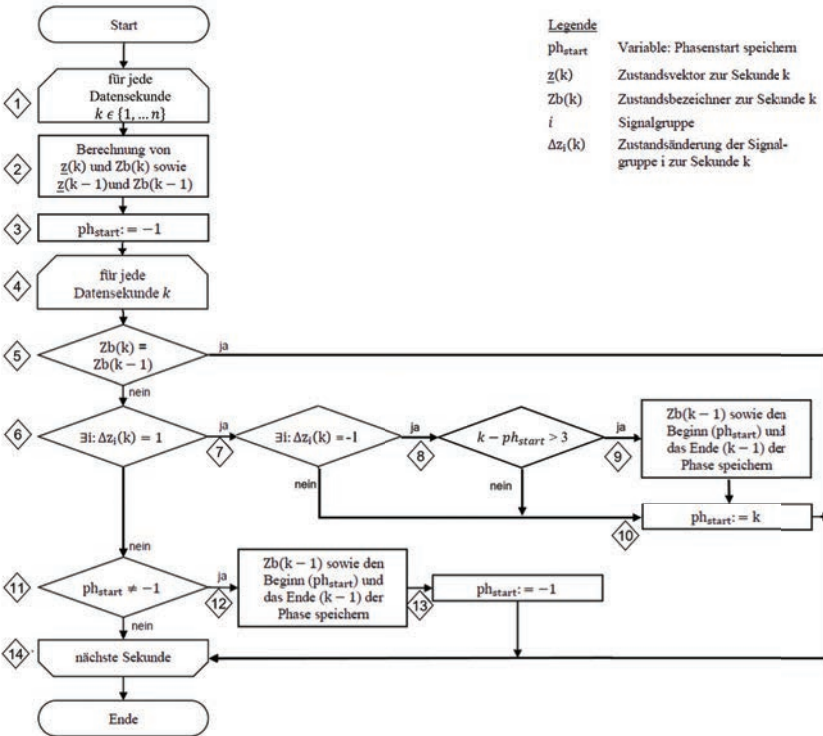


Abbildung 7.4: Programmablauf für die Phasenidentifikation mit Ergänzung von [Hec21]

7.1.4 Phasenspeicher

Zwecks Konsistenzprüfung können in einer Erweiterung die identifizierten Phasen mittels ihres Skalars für den Signalisierungszustand in einer Liste gespeichert werden. Daraufhin werden alle Signalisierungszustände mit dieser Liste abgeglichen und bei Übereinstimmung automatisch als Phase identifiziert. Abbildung 7.5 zeigt den zugehörigen Programmablauf. Voraussetzung ist, dass alle Signalisierungszustände und Phasenmerkmale (mittels einer der zwei vorangehend beschriebenen Algorithmen) berechnet wurden. In den Schritten 1 bis 6 wird eine Liste der bekannten Phasen ermittelt, welche in dieser Arbeit auch als Phasenspeicher bezeichnet wird. In den Schritten 7 bis 10 wird die Konsistenz der Phasenerkennung geprüft. Abweichungen werden im Phasenmerkmal nachgetragen.

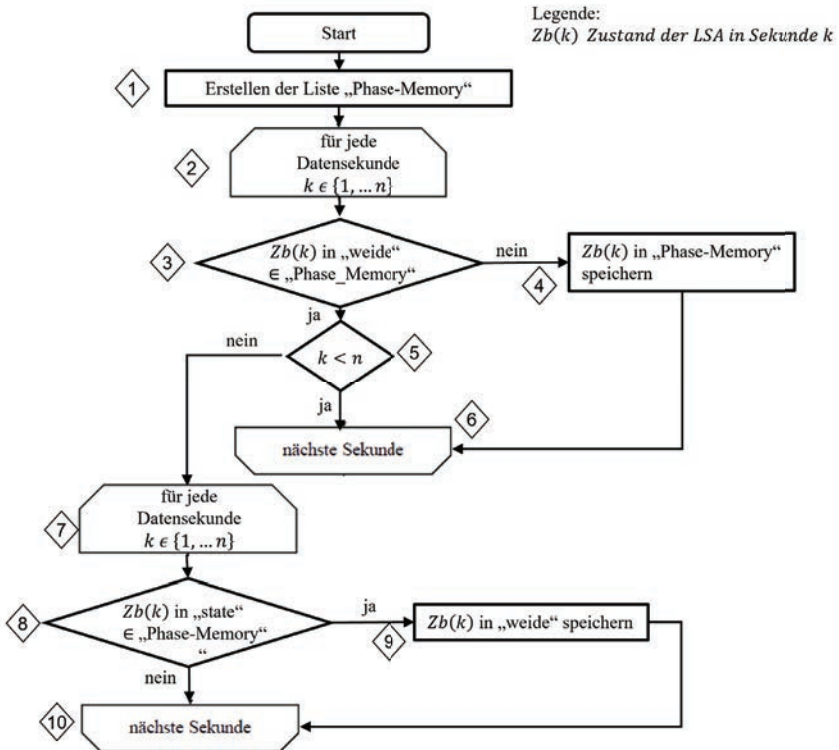


Abbildung 7.5: Programmablauf für den Phasenspeicher [Hec21]

7.1.5 Phasenmerkmal bei Phasenwechseln ohne Phasenübergang

In Abschnitt 7.1.3 wurde der zugrunde liegende Algorithmus ergänzt. Durch die Erweiterung entsteht die Frage, ob es sich bei einem direkten Phasenwechsel ohne Phasenübergang (P2P) bei der zweiten Phase tatsächlich um eine Phase oder den Beginn eines Phasenübergangs handelt. Um solche „Pseudophasen“ zu vermeiden, werden nur Phasen als solche erkannt, wenn sie mindestens 5 Sekunden dauern. Ansonsten werden sie einem Phasenübergang zugeordnet.

Pseudophasen können auch Zustände in Phasenübergängen sein, die vom Algorithmus als Phasen gekennzeichnet werden. Die Auswirkung von Pseudophasen für die Vorhersage veranschaulicht Abbildung 7.6. Gezeigt werden die Skalare der Signalisierungszustände sowie des Phasenmerkmals. Da die Vorhersage den Beginn und das Ende der nächsten Phase prädiziert und Pseudophasen hierfür als Phasen gelten, verkürzt sich der Prognosehorizont entsprechend. Im Beispiel verkürzt sich der Beginn der nächsten Phase um 6 Sekunden und ihres Endes um 12 Sekunden. Für jede Pseudophase ergeben sich zwei Pseudophasenübergänge.

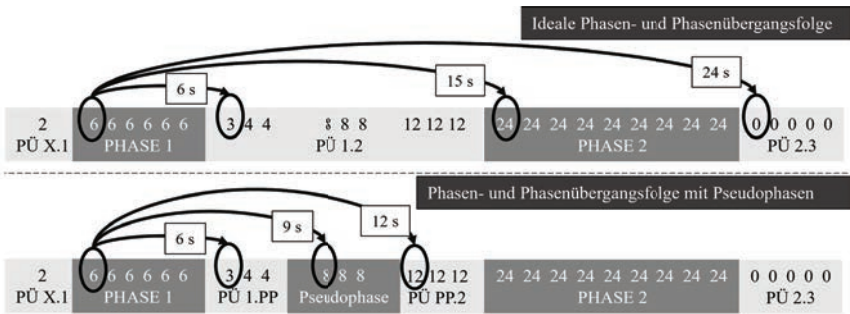


Abbildung 7.6: Beispiel für die Konsequenz von Pseudophasen für den Prognosehorizont [Hee21]

7.1.6 Statistische Bereinigung

Die Menge der erkannten Phasen kann auf statistische Eigenschaften geprüft werden. So können Pseudophasen – Teile von Phasenübergängen, welche fälschlicherweise als Phasen erkannt wurden – aus der Menge der erkannten Phasen entfernt werden. Zudem können bisher nicht identifizierte Phasen ermittelt werden. Ersteres wird im Folgenden als Stat-raus und zweiteres als Stat-rein bezeichnet. Für Stat-raus werden für den Vektor des Phasenmerkmals die Häufigkeit des Phaseneintritts, minimale und maximale Phasendauer sowie Mittelwert, Standardabweichung und Median der Phasendauer ermittelt

beziehungsweise selbige für Stat-rein bezüglich des Vektors der Skalare des Signalisierungszustands.

Stat-raus

Statistische Bedingungen, um eine Phase als einen Teil eines Phasenübergangs umzudefinieren, sind:

1. die Phase tritt weniger häufig auf als x -mal, x ist an das jeweilige SP anzupassen
2. die Phasendauer ist stets gleich (Standardabweichung von Null), und sie ist im Median kürzer als 5 s
3. die Phasendauer dauert im Median weniger als 5 Sekunden und maximal 9 s

Mindestens eine der Bedingungen ist für die Umgruppierung zu erfüllen. Diese Bedingungen und Grenzwerte wurden empirisch ermittelt.

Die erste Bedingung entfernt insbesondere Signalisierungszustände, welche zumeist im Zusammenhang mit einem Signalprogrammwechsel auftreten. Die zweite Bedingung entfernt kurze Phasen welche keine variable Phasendauer aufweisen. Es wird angenommen, dass diese zu jeweils einem bestimmten Phasenübergang gehören. Die dritte Bedingung entfernt kurze Phasen, welche zwecks Einhaltung von Zwischenzeiten in verschiedenen Phasenübergängen notwendig sind. Diese Bedingungen sind ggf. an die betrachtete Logik anzupassen.

Stat-rein

Die Menge der Signalisierungszustände kann ebenfalls auf statistische Eigenschaften geprüft werden. Hiermit können zuvor nicht identifizierte Phasen identifiziert werden. Um auszuschließen, dass Phasen, welche durch Stat-raus entfernt wurden, wieder integriert werden, müssen alle der folgenden Bedingungen erfüllt werden, um den Signalisierungszustand als Phase aufzunehmen:

1. der Median der Phasendauer muss mindestens 5 s betragen,
2. die Standardabweichung mindestens 1 Sekunde und die maximale Phasendauer sollte mindestens 10 s betragen und
3. die Phase muss mindestens x -mal im Datensatz Verwendung finden, x ist an das jeweilige SP anzupassen.

7.2 Merkmalsevaluation

Die Qualität der mit diesen Algorithmen berechneten Merkmale wird im Vorfeld der Vorhersageexperimente ermittelt, um die Prognosequalität erklären zu können. Der Algorithmus Wei wurde zunächst separat evaluiert. Hiernach wurde dessen Optimierung betrachtet. Im nächsten Abschnitt wird hierzu zunächst die Durchführung der Evaluation beschrieben.

7.2.1 Evaluationsdurchführung

Ob die Optimierung des Algorithmus nach Weidemann durch die Erweiterungen einen positiven Einfluss auf die Phasenidentifikation hat, wurde in mehreren Testläufen evaluiert. Für die Evaluation der Merkmale wurde ein Vergleichsalgorithmus formuliert, welcher die identifizierten Signalzustände und Phasen der Phasendefinition mit den Phasendefinitionen der technischen Dokumentation abgleicht. Der Algorithmus lehnt sich an den in Abbildung 7.5 gezeigten Algorithmus nach Heckmann an. Die Liste des Phasenspeichers wird als Referenz händisch mit Hilfe der Dokumentation befüllt. Zudem wird der Merkmalsvektor nicht verändert.

Die Unterschiede zwischen Merkmalsvektor und Sollzustand dieses Vektors anhand des Referenzspeichers werden separat protokolliert und analysiert. Bewertungsgrößen der Algorithmus-Evaluation sind:

- AP_{Doku} Anzahl der Phasen aus der Dokumentation
- AP Anzahl der identifizierten Phasen
- AS Anzahl der identifizierten Signalisierungszustände
- PP relative Übereinstimmung der identifizierten Phasen mit der Dokumentation relativ zu den dokumentierten Phasen
- AP_{pseudo} Anzahl der erfassten Pseudophasen (Anzahl der identifizierten Phasenanzahl korrekt identifizierter Phasen)
- PS Prozentuale Übereinstimmung der Abfolge der Signalisierungszustände mit der durch die Phasendefinition der Dokumentation implizierten Abfolge
- PPu_P Prozentuale Übereinstimmung der rekonstruierten Phasenabfolge mit der durch die Phasendefinition der technischen Dokumentation implizierten Phasenabfolge
- APu Anzahl der identifizierten PÜ
- APu_{var} . Absolute Anzahl der Variationen aller PÜ.

Es werden Stichproben der KP an der Druseltalstraße (KP 722) und Ysenburg-/Schützenstraße (KP 020) betrachtet. Der KP an der Ysenburgstraße ist wie der KP

Trompete dreiarstig und verfügt über einen Teilknoten, der eine LSA für den FG-Verkehr ist. Evaluiert werden die Monate September und November 2019 für alle Signalprogramme außer dem Nachtprogramm.

Die erste Evaluation betrachtet den Algorithmus Wei und umfasst 150.000 Stichproben das heißt circa 41 Stunden. Weitere Evaluationen fassen alle Erweiterungen zu fünf optimierten Phasenerkennungen zusammen und betrachten 150.000 oder 300.000 Stichproben; Evaluationen mit 150.000 Stichproben sind mit $-$ gekennzeichnet. Die eine Hälfte basiert auf Wei die andere auf Wei+, denen sich nachfolgend A beziehungsweise B aus Abbildung 7.7 anschließen. Der Unterschied zwischen A und B besteht in der zusätzlichen Verwendung von Stat-raus vor P2P.

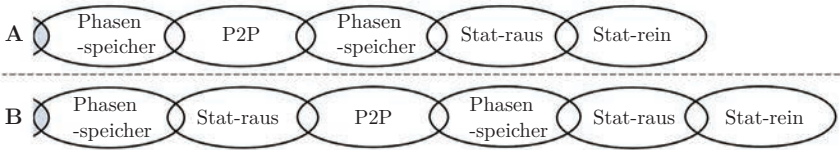


Abbildung 7.7: Abfolge der Algorithmen für ihre Evaluation

7.2.2 Evaluation der Phasenmerkmale nach Weidemann

Die Ergebnisse der Evaluation zeigen die Schwächen des Algorithmus Wei auf. Die Tabellen 7.1 und 7.2 zeigen die ermittelten Bewertungsmaße mit Wei ohne Wei+, A und B.

Tabelle 7.1: Evaluationsergebnisse für Wei der grundlegenden Merkmalsidentifikation zu KP 722

Mo SP	AS_{Doku}	AP_{Doku}	PP	AP	PS	APu_{Doku}	$APu_{var.,Doku}$	APu	$APu_{var.}$
9 1	58	11	90	24	65	54	456	122	200
9 2	61	11	90	27	60	22	232	93	212
9* 2*	61	11	90	27	66	-	-	-	-
9 3	109	11	90	43	67	26	234	163	291
11 1	105	11	90	45	66	28	218	190	310
11 2	112	11	90	46	62	25	220	156	305
11 3	104	11	90	41	68	23	179	144	230

An KP 722 werden von 11 Phasen 10 korrekt identifiziert ($PP_{Doku} = 90\%$). Jedoch werden 24 bis 46 Phasen insgesamt identifiziert. Diese hohe Zahl der Pseudophasen von 14 bis

36 hat Konsequenzen für die Zahl der Phasenübergänge (AP_u). Sie ist entsprechend ein Vielfaches höher, als in der abgebildeten Logik implementiert ist. In Kassel ist es möglich, dass Phasenübergänge unter Einhaltung der Zwischenzeiten abgebrochen werden zugunsten eines Wechsels in eine andere Phase. Dies in Zusammenhang mit der Integration der Mindestfreigabezeiten in Phasenübergänge kann die hohe Zahl an Pseudophasen erklären.

Tabelle 7.2: Evaluationsergebnisse der grundlegenden Merkmalsidentifikation zu KP 020

M	SP	AS_{Doku}	AP_{Doku}	PP	AP	PS	AP_u_{Doku}	$AP_u_{var.,Doku}$	AP_u	$AP_u_{var.}$
9	1	48	17	88	22	81	46	164	81	265
9*	1*	48	17	88	22	92	-	-	-	-
9	2	45	17	64	16	82	49	180	67	258
9	3	50	17	59	17	77	52	205	82	330
11	1	45	17	59	17	80	46	165	70	248
11	2	44	17	88	21	80	51	166	74	271
11	3	48	17	59	16	76	51	165	74	289

Hierdurch gibt es für jeden möglichen Phasenübergang mehrere Abfolgen von Signalisierungszuständen. Die Zahl der möglichen Zustandsfolgen für Phasenübergänge wird durch Pseudophasen reduziert, da ein Teil ihrer Varianz über die Pseudophase aus der Menge der Phasenübergänge entnommen wird. Die Zahl der Phasenübergänge steigt jedoch. Zudem verkürzen die Pseudophasen, wie oben beschrieben, den Prognosehorizont.

Die mit * gekennzeichneten Zeilen zeigen Ergebnisse, welche sich durch die einfache Verwendung des Phasenspeichers aus Abschnitt 7.1.4 ergeben. Für beide KP ergibt sich eine Verbesserung um mehr als 5% für die Übereinstimmung der Signalzustandsfolgen durch diese reine Konsistenzprüfung.

Auch betreffend der weiteren Punkte ist für KP 020 ein ähnliches Bild zu sehen. Es werden lediglich 59% bis 88% der vorgesehenen Phasen korrekt identifiziert und insgesamt erheblich mehr Phasen vermeintlich identifiziert als tatsächlich existieren. Eine Übereinstimmung von mehr als 90% für die Signalzustandsfolgen kann nur unter Verwendung des Phasenspeichers erzielt werden. Der Dokumentation zufolge gibt es weniger mögliche Phasenübergänge. Die identifizierte Anzahl an Phasenübergängen ist um den Faktor 1,5 höher, jedoch werden im Verhältnis mehr Variationen der Phasenübergänge verzeichnet. Wo für KP 722 durchschnittlich weniger als zwei mögliche Variationen eines Phasenübergangs identifiziert wurden, sind es für KP 020 durchschnittlich drei bis vier.

7.2.3 Evaluation der Optimierung

Die gemeinsamen Evaluationen aller Erweiterungen wurde mit acht verschiedenen Zusammenstellungen der Erweiterungen durchgeführt: Wei oder Wei+ ohne weitere Erweiterungen, mit Phasenspeicher und jeweils mit Erweiterungsfolge A oder B aus Abbildung 7.7.

Die durchgeführten Evaluationen mit Ergebnissen sind für den KP in der Druselstraße in Tabelle 7.3 und für den KP an der Ysenburgstraße in Tabelle 7.4 zusammengefasst.

Tabelle 7.3: Evaluationsergebnisse von Wei+ der Merkmalsidentifikation an KP 722

M	SP	Algorithmus	PP	AP	AP_{pseudo}	PPu_P	PS	APu	APu_{var}
11	2	Wei,A	90	-	17	99,9	78	132	263
11	2	Wei,B	90	-	17	99,9	78	132	263
11	2	Wei+,A	90	-	18	99,9	78	137	263
11	2	Wei+,B	90	-	18	99,9	78	137	263
9	2	Wei+	90	50	40	62	62	169	363
9	2	Wei+*	90	50	40	66	58	210	356
9	2	Wei+,A	90	38	28	99,9	78	78	263
9 ⁻	2	Wei	91	27	17	60	62	92	212
9 ⁻	2	Wei*	91	27	17	66	60	111	196
9 ⁻	2	Wei,A	81	23	14	99,9	81	70	139
9	3	Wei	90	48	38	62	62	174	367
9	3	Wei*	90	48	38	66	59	178	353
9	3	Wei,A	90	37	27	99,9	78	75	263

Tabelle 7.3 zeigt in den ersten vier Zeilen, dass die Erweiterungsfolgen A und B keinen qualitativen Unterschied in der Phasenidentifikation aufweisen. Die Ergebnisse für die Ysenburgstraße (Tabelle 7.4) bestätigen dies. Zwischen Wei und Wei+ sind Unterschiede erkennbar, diese sind für die beiden KP konträr. Für KP 722 erkennt Wei+ mehr Pseudophasen (AP_{pseudo}), weshalb mehr Phasenübergänge (APu) identifiziert werden. Für KP 020 deutet sich der gegenteilige Effekt an.

Tabelle 7.4: Evaluationsergebnisse der Merkmalsidentifikation an KP 020

M	SP	Algorithmus	PP	AP	AP_{pseudo}	PPu_P	PS	APu	APu_{var}
9	3	Wei,A	65	-	5	98	72	45	190
9	3	Wei,B	65	-	5	98	72	45	190
9	3	Wei+,A	76	-	5	99	73	43	183
9	3	Wei+,B	76	-	5	99	73	43	183
9	2	Wei	59	17	7	77	83	83	337
9	2	Wei*	59	17	7	92	74	58	185
9	2	Wei,A	65	16	5	99	72	45	190
9 ⁻	3	Wei	59	17	11	77	83	81	330
9 ⁻	3	Wei*	59	17	11	92	74	75	182
9 ⁻	3	Wei,A	65	16	5	98	72	68	184

Eindeutig ist, dass bei Verwendung der Erweiterungsfolgen die Raten für eine korrekte Abbildung bzw. Identifikation – PPu_P , PS (und PP) – erzielt werden können. Auch die Zahl der Pseudophasen (AP) und somit auch Phasenübergänge (APu) ist geringer. Dies ist so ausgeprägt, dass sogar die Zahl der Variationen der Phasenübergänge (APu_{var}) sinkt oder nur geringfügig steigt.

7.2.4 Zwischenfazit

Ob mit Wei oder Wei+ höhere Raten in der Abbildung der Phasen(-übergänge) von KP erzielt werden können, ist den Evaluationsergebnissen zufolge KP-abhängig. Jedoch erscheint es sinnvoll, Erweiterungsfolgen zu verwenden. Im Weiteren wird deshalb nur noch mit Folge A gearbeitet, da diese weniger Rechenzeit benötigt als Folge B und die beiden Folgen gleiche Ergebnisse zeigen. Welche Bewertungsgrößen dieser Evaluationen für die Modellierung zwecks Phasenvorhersage besondere Relevanz haben, zeigen die im Folgenden beschriebenen Prognoseexperimente auf.

7.3 Versuchsaufbau

In Abschnitt 4 wurde beschrieben, dass es sinnvoll erscheint, zunächst die nächste Phase vorherzusagen und die Vorhersage als zusätzliches Merkmal für SZP zu verwenden. Dies soll mit den im Folgenden beschriebenen Experimenten überprüft werden.

Die Vorhersage der Phase teilt sich in zwei Teile auf: Zum einen ist der nächste Signalisierungszustand einer Phase (Phasen-Signalzustand-Prognose (PSP)) und zum anderen ist der Zeitpunkt, zu dem dieser Signalisierungszustand geschaltet wird (Phasen-Schalt-Zeitpunkt-Prognose (PSZP)), zu prognostizieren. Da es sich um eine vorangestellte Vorhersage handelt, auf der weitere Prognosen basieren, wird angenommen, dass eine hohe Prognosequalität notwendig ist, um die Prognosequalität der folgenden Vorhersagen stützen zu können.

Die verwendeten ML-Methoden sind XGBoost und Bayes'sche Netze. Vorexperimente haben gezeigt, dass die (Bedingten-)Wahrscheinlichkeiten sowie die Netzstruktur für die Bayes'schen Netze vorzugeben sind. Die implementierten Algorithmen zur automatisierten Bestimmung dieser versagen anhand der Datengrundlage. Entsprechend werden diese im Vorhinein auf Basis von jeweils 100.000 Stichproben pro Signalprogramm und KP für die Wochentage Dienstag, Mittwoch und Donnerstag ermittelt. Abbildung 7.8 zeigt die beiden verwendeten Bayes'schen Netze. Der Unterschied zwischen den Netzen ist, dass das 1. Netz das Merkmal der letzten Phase nutzt und das 2. Netz nicht. Weitere Merkmale können für die Bayes'schen Netze nicht verwendet werden.

Als Phasenmerkmale für die letzte, aktuelle und nächste Phase können mehrere Merkmale herangezogen werden. Diese sind der Signalisierungszustand (SZ) sowie die Phasenmerkmale der verschiedenen Identifikationen: Doku, Wei, Wei+ und Wei,A.

Zusätzlich werden für XGBoost mindestens die Zeit seit und die Umlaufsekunde der letzten Zustandswechsel der Phasenmerkmale sowie die aktuelle Umlaufzeit und der aktuelle Zeitpunkt verwendet. Dieses Datenmodell wird mit einem vorangestellten m für minimales Datenmodell gekennzeichnet. In einem erweiterten Datenmodell werden zusätzlich SZ

und die letzten beiden Phasen als Merkmale hinzugenommen. Dies wird nicht gesondert gekennzeichnet. Darüber hinaus werden für XGBoost teilweise SG-, Det- und ÖPNV-Richtungs-Merkmale verwendet. Falls diese Verwendung finden, werden alle Merkmale zu SG-, Det- bzw. ÖPNV-Richtungen in der Darstellungsweise Zeit seit verwendet.

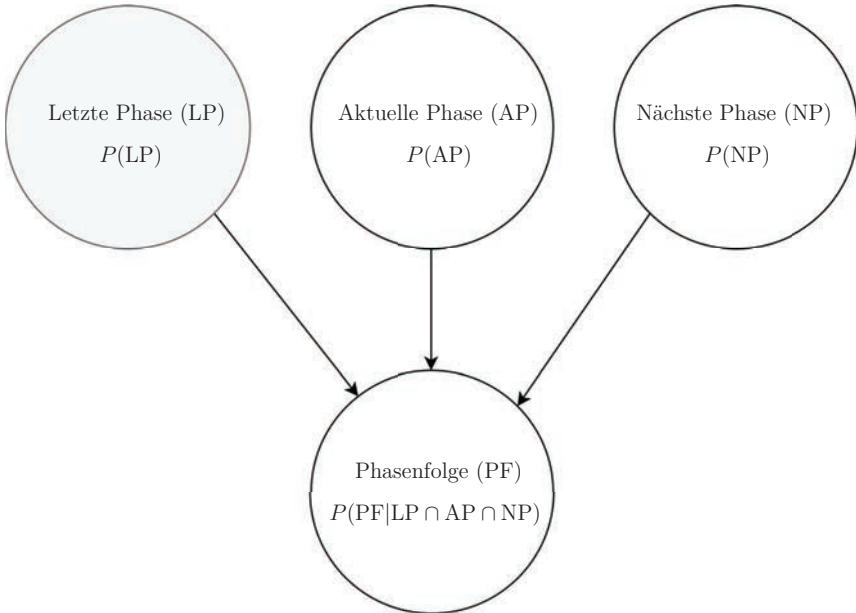


Abbildung 7.8: Bayes'sche Netze zur Phasenvorhersage [Hec21]

Mit den beiden Methoden werden aufeinander aufbauend die nächsten drei zukünftigen Phasen vorhergesagt. Das heißt, das Ergebnis der vorhergehenden Phasenvorhersage wird für die Prognose der folgenden verwendet, außer für m XGB. Eine Anpassung der Wahrscheinlichkeiten und Netzarchitektur bei unveränderter Datenbasis ist für Bayes'sche Netze nicht notwendig und damit auch kein erneutes Training. Die Zeitpunkte von Phasenanfängen werden nur mit XGBoost (mit Bayes'scher Suche) vorhergesagt, da für die Verwendung von Bayes'schen Netzen nicht ausreichend Rechenkapazitäten zur Verfügung standen.

Es wurden 100.000 Stichproben der drei am Tag laufenden Signalprogramme 2019 von KP 002 und 722 jeweils für die Vorhersagen verwendet. Die Test- und Trainingsdaten wurden im Verhältnis 1:2 geteilt. Für XGBoost wurden die Hyperparameter aus Anhang F.7 und für die Bayes'schen Netze $Inertia = 1$ verwendet. $Inertia = 1$ erlaubt dem Training, die Wahrscheinlichkeiten anzupassen. Dieser Wert wurde in Vorexperimenten ermittelt.

Ob die ermittelte Prognosequalität für PSP und PSZP für eine SZP ausreichend ist, kann erst bei der Ermittlung der Qualität der anschließenden SZP evaluiert werden. Entsprechend werden Experimente zur SZP mit Merkmalen der PSP- und PSZP-Prognosen durchgeführt.

Als leistungsfähigste Methode aus Kapitel 5 wird XGBoost zur SZP verwendet. Die Hyperparameter wurden im Vorhinein mit Bayes'scher Suche ermittelt. Für die eigentlichen Experimente erfolgt keine weitere Optimierung, um die Ergebnisse schneller zu erhalten.

Das als Referenz (R) dienende Datenmodell ohne Phasenmerkmale setzt sich aus dem Signalzustand ausgewählter SG sowie der Zeit seit ihrem letzten Grünanfang (GA) bzw. -ende (GE) ($_{ts}SG^m$) zusammen. Darüber hinaus werden alle Detektordaten ($_{ts}DET^m$), ebenfalls mit der Zeit seit der letzten (Nicht-)Belegung und die von den ÖPNV-Fahrzeugen übermittelten Informationen ($_{ts}OEV^n$) verwendet. Das Referenzdatenmodell umfasst folglich drei Merkmale für jede SG und alle Detektoren sowie ÖPNV-Meldungen. Außerdem werden die Zeit in Minuten seit Mitternacht, die Taktsekunde und die Nummer des Wochentages berechnet und einbezogen.

Um die Eignung der PSZ und PSZP für die SZP zu evaluieren, wird das Datenmodell R um die Merkmale SZ bzw. das Phasenmerkmal Doku erweitert. Es wird immer die Information zum letzten und aktuellen Zustand des jeweiligen Merkmals ergänzt. Dies wird in einer Kurzschreibweise mit SZ+0 bzw. Doku+0 vermerkt. Zusätzlich werden in jeweils zwei weiteren Datenmodellen der nächste bzw. übernächste Zustand der beiden Merkmale verwendet, was als SZ+1 bzw. Doku+1 und SZ+2 bzw. Doku+2 usw. Ebenfalls immer verwendet werden die Sekunden seit der letzten Veränderung der Merkmale bzw. die prognostizierte Zeit bis zum nächsten Zustandswechsel der Merkmale einbezogen.

Insgesamt wurden für die SZP neun Datenmodelle einschließlich der Referenz R evaluiert. Es handelt sich um jeweils zwei Datenmodelle für den aktuellen SZ bzw. die aktuelle Phase einschließlich des letzten Merkmalzustands, sowie für den nächsten, übernächsten bzw. überübernächsten Zustand der beiden Merkmale. Zielvorgaben für alle Experimente sind die diskrete Zeit bis zum nächsten grünen Start (GS) und Ende (GE) für jedes der dedizierten SG ($_{tu}SG_{GAbzw.GE}$). Die Auswertung erfolgt mit einem Friedman-Nemenji-post-hoc-Test.

7.4 Methoden- und Datenmodellvergleich mit grundlegenden Datenmodellen: Phasenprognose

Im Folgenden werden die Ergebnisse der Phasenprognose vorgestellt. Zunächst werden die Ergebnisse für die Phasenprognose der nächsten Phase erläutert, hiernach die für die der übernächsten Phase. Die Ergebnisse für die überübernächste Phase sind in Anhang J zu finden.

7.4.1 Phasenprognose der nächsten Phase

Für die Phasenprognose der nächsten Phase werden hohe KKR erzielt. Wie Tabelle 7.5 für Signalprogramm (SP) 2 zeigt, werden für jeden KP KKR von über 85 % erzielt, häufig auch über 95 %. Für die anderen beiden SP werden KKR über 80 % erreicht, siehe Tabellen J.1 und J.2.

Für welche ML-Methode und welches Datenmodell jeweils die höchste KKR erzielt werden konnte, ist abhängig vom KP. Diese Beobachtung war unabhängig von Monat und SP, was die ohnehin bereits hohe Prognosequalität weiter stützt, da über die Monate hinweg die gleichen Logiken in den SP Verwendung finden und sich die Logiken zwischen den SP aufgrund der gleichen KP-Geometrie nicht in wesentlichen Punkten unterscheiden können.

Für KP 020 werden die besten Ergebnisse mit Wei(+),A erreicht. Für KP 722 mit den Phasendefinitionen entsprechend der technischen Dokumentation, was für eine mangelhafte Phasenidentifikation spricht, wobei sich die Ergebnisse mit den automatisierten Phasenidentifikationen im gleichen Wertebereich befinden und nur minimal geringer ausfallen. Überraschend ist, dass für KP 002 das Merkmal für den SZ des KP die besten Prognoseergebnisse zeigt. Da für diesen KP keine Dokumentation der Phasendefinition vorlag, konnten keine Doku-Phasenmerkmale evaluiert werden. Es ist jedoch anzunehmen, dass diese dem SZ-Merkmal sehr ähnlich wären, da dies für die anderen KP der Fall ist und damit auch hier die Phasenidentifikation über Verbesserungspotentiale verfügt.

Allgemein erzielen die SZ- und Doku-Merkmale sowie die Phasenmerkmale, die mit Phasenfolge A bestimmt wurden, bessere KKR als nur Wei(+). Wei und Wei+ mit und ohne A zeigen auch in diesen Versuchen vergleichbare Ergebnisse. Vor dem Hintergrund der Vergleichbarkeit von SZ mit beispielsweise Wei+,A ist in weiteren Experimenten zu validieren, ob die Phasenidentifikation, für die ohnehin SZ zu berechnen ist, für die Vorhersage der Kennzahl der nächsten Phase ausreichend ist.

Tabelle 7.5: Erste Prognoseergebnisse für die nächste Phase (KKR [%]), SP 2

KP	M	SP	Phasenidentifikation	BN1	BN2	^m XGB	XGB	XG+DET+SG	^m XGB+ÖPNV	XGB+ÖPNV	XGB+SG-DET+ÖPNV
020	9	2	SZ	93	93	95	95	97			
020	9	2	Doku	94	94	94	95	97			
020	9	2	Wei	89	90	90	90	93			
020	9	2	Wei+	89	89	90	90	93			
020	9	2	Wei,A	94	94	96	96	98			
020	9	2	Wei+,A	94	94	96	96	98			
020	11	2	SZ	92	92	94	95	97			
020	11	2	Doku	92		93	93	96			
020	11	2	Wei	89		91	92	94			
020	11	2	Wei+	89		91	91	94			
020	11	2	Wei,A	92	93	95	96	97			
020	11	2	Wei+,A	92	93	95	96	97			
722	9	2	SZ	77	86	86	87	94	75	83	93
722	9	2	Doku	81	90	90	91	96	80	85	96
722	9	2	Wei	81		82	81	87	78	77	86
722	9	2	Wei+	81		82	81	87	77	76	86
722	9	2	Wei,A	73	83	84	85	91	74	82	91
722	9	2	Wei+,A	73	84	84	85	92	74	82	92
722	11	2	SZ	74		83	84	91	81	84	91
722	11	2	Doku	79		88	89	94	86	89	93
722	11	2	Wei	78		76	73	82	73	73	82
722	11	2	Wei+	77		76	73	81	72	73	83
722	11	2	Wei,A	71		81	83	90	80	83	89
722	11	2	Wei+,A	71		81	83	90	79	83	90
002	9	2	SZ	74		78	65	85	78	78	86
002	9	2	Wei	81		72	73	82	71	73	82
002	9	2	Wei+	81		77	78	84	76	77	84
002	9	2	Wei,A	67		60	62	71	59		70
002	9	2	Wei+,A	71		66	68	76	64	67	76
002	11	2	SZ	77		81	79	87	80	81	88
002	11	2	Wei	84		76	78	84	75	77	85
002	11	2	Wei+	83		80	81	86	80	81	87
002	11	2	Wei,A	69		64	66	74	62	63	73
002	11	2	Wei+,A	73		72	74	78	69	81	76

Betreffend der ML-Methoden sind das einfache BN und geringfügig komplexere BN miteinander vergleichbar bezüglich ihrer Ergebnisse. Deshalb und weil dieses Modell im Grenzbereich der Rechenkapazität arbeitet, wurde BN2 in den weiteren Versuchen nicht verwendet. BN1 benötigt erheblich weniger Ressourcen für die Berechnung einer Prognose.

Generell lässt sich feststellen, dass XGB um 5 % bis 20 % bessere Ergebnisse für die Vorhersage der nächsten Phase erzielt als die Bayes'schen Netze. Dies kann auf die umfangreicheren Datenmodelle, welche zum Teil zusätzliche aktuelle Informationen zu SG-, DET- und ÖPNV-Merkmalen enthalten, zurückgeführt werden. Die Modelle, die diese zusätzliche Information enthalten, erzielen die höchsten KKR. Die Modelle XGB+DET+SG und XGB+SG+DET+ÖPNV sind diesbezüglich vergleichbar. KP mit höherer Frequentierung durch den ÖPNV zeigen höhere KKR mit ÖPNV-Merkmalen. Für KP mit geringer ÖPNV-Belastung erscheinen ÖPNV-Merkmale hingegen nicht maßgeblich für die Ergebnisse.

Der Vergleich zwischen dem reduzierten und vollen – ohne und mit Informationen zu vergangenen Phasen – Umfang der Phasenmerkmale für XGB zeigt, dass die Verwendung des vollen Umfangs der Phasenmerkmale bessere Ergebnisse bewirkt. Folglich steht die Dauer einer Phase und der Umlaufzeitpunkt, zu dem sie beginnt, in Zusammenhang mit der Auswahl der nächsten Phase.

Für eine optimierte Verkettung der Ergebnisse für die Phasenprognose der drei nächsten Phasen wurden weitere Versuche mit den drei besten Datenmodellen unternommen. Tabelle 7.6 zeigt die zugehörigen Ergebnisse. BN1 und das beste XGB-Modell sind einander gegenübergestellt. Für die Phasenprognose der nächsten Phase ist XGB besser. Teil der Optimierung ist es, die jeweils beste Methode und nicht die gleiche wie im vorherigen für den jeweiligen Vorhersageschritt zu verwenden, wie es für die ersten Experimente getan wurde. Da als Referenz keine mit den Daten stimmige Phasendokumentation zur Verfügung stand, wurde für KP 002 keine Optimierung vorgenommen.

Tabelle 7.6: Prognoseergebnisse für die nächste Phase, SP 1, 2, und 3 (KKR [%])

KP	M	SP	Phasenidentifikation	BN1	XG+DET+SG(+ÖPNV)
020	9	1	SZ	91	97
020	9	1	Doku	92	97
020	9	1	Wei+,A	93	97
020	11	1	SZ	92	97
020	11	1	Doku	93	97
020	11	1	Wei,A	94	97
722	9	1	SZ	59	83
722	9	1	Doku	77	83
722	9	1	Wei	67	78
722	11	1	SZ	63	80
722	11	1	Doku	67	83
722	11	1	Wei+,A	61	78
020	9	2	SZ	93	97
020	9	2	Doku	94	97
020	9	2	Wei,A	94	98
020	11	2	SZ	92	97
020	11	2	Doku	92	96
020	11	2	Wei,A	92	97
722	9	2	SZ	77	94
722	9	2	Doku	81	96
722	9	2	Wei+,A	73	92
722	11	2	SZ	74	91
722	11	2	Doku	79	94
722	11	2	Wei+,A	71	90
020	9	3	SZ	91	93
020	9	3	Doku	91	94
020	9	3	Wei,A	92	94
020	11	3	SZ	92	96
020	11	3	Doku	93	96
020	11	3	Wei,A	93	96
722	9	3	SZ	66	(83)
722	9	3	Doku	71	(85)
722	9	3	Wei	73	77
722	11	3	SZ	65	(83)
722	11	3	Doku	69	(85)
722	11	3	Wei,A	59	(80)

7.4.2 Phasenprognose der übernächsten Phase

Tabelle 7.7: Erste Prognoseergebnisse für die übernächste Phase, SP 2 (KKR [%])

KP	M	SP	Phasenidentifikation	BN1	BN2	m XGB	XGB	XG+DET+SG	m XGB+ÖPNV	XGB+ÖPNV	XGB+SG-DET+ÖPNV
020	9	2	SZ	81	81	12	13	15			
020	9	2	Doku	84	84	2	4	11			
020	9	2	Wei	77	78	6	11	33			
020	9	2	Wei+	78	78	10	17	38			
020	9	2	Wei,A	84	84	14	27	55			
020	9	2	Wei+,A	84	84	16	22	25			
020	11	2	SZ	80	81	1	2	10			
020	11	2	Doku	82		2	6	15			
020	11	2	Wei	77		8	12	15			
020	11	2	Wei+	77		7	10	19			
020	11	2	Wei,A	81	82	10	3	28			
020	11	2	Wei+,A	81	82	6	4	28			
722	9	2	SZ	46	70	36	38	49	36	38	49
722	9	2	Doku	49	75	39	37	46	37	37	46
722	9	2	Wei	70		10	13	22	11	15	22
722	9	2	Wei+	70		10	11	22	12	16	21
722	9	2	Wei,A	44	67	34	40	52	33	41	52
722	9	2	Wei+,A	44	68	34	36	50	34	39	50
722	11	2	SZ	49		12	31	41	11	30	41
722	11	2	Doku	54		37	35	45	34	36	45
722	11	2	Wei	66		10	14	21	13	7	21
722	11	2	Wei+	66		14	17	19	14	13	19
722	11	2	Wei,A	47		7	30	31	18	33	30
722	11	2	Wei+,A	48		15	32	31	22	32	31
002	9	2	SZ	60		14	25	38	12	29	36
002	9	2	Wei	53		10	15	24	9	14	21
002	9	2	Wei+	61		13	26	37	17	23	29
002	9	2	Wei,A	36		4	7	9	5		9
002	9	2	Wei+,A	42		2	6	11	4	10	11
002	11	2	SZ	32		11	25	50	15	25	51
002	11	2	Wei	58		10	16	13	11	18	14
002	11	2	Wei+	63		13	24	28	11	24	25
002	11	2	Wei,A	25		6	9	11	5	8	11
002	11	2	Wei+,A	46		7	7	12	14	24	12

Die Ergebnisse für die übernächste Phase unterscheiden sich deutlich im Vergleich zu denen der nächsten Phase. Die erzielten KKR sinken um ca. 10 % bis 20 % für BN und um ca. 80 % für XGB. Nun sind BN geeigneter als XGB. Die deutliche Verschlechterung der Ergebnisse mit XGB als Modell wird darauf zurückgeführt, dass die aktuellen Informationen zu SG, DET und ÖPNV auf die übernächste Phase keinen maßgeblichen Einfluss mehr haben im Vergleich zur nächsten Phase. Eine Überlegung ist, DET- und ÖPNV-Merkmale vorherzusagen, um Phasenprognose und SZP zu verbessern. Es ist zweifelhaft, ob die Vorhersage dieser Merkmale zu einer vergleichbaren Qualität wie der der BN führen würde.

Bezüglich der Datenmodelle zeigt sich ein vergleichbares Muster wie für die Phasenprognose der nächsten Phase: die meist besten Ergebnisse werden von Wei(+),A erzielt, Wei(+), ohne A oder B erreichen die geringsten KKR. Die KKR für SZ und Doku unterscheiden sich zumeist nur in weniger als 5%-Punkten von Wei(+),A.

Die optimierte Verkettung der Ergebnisse zeigt eine Verbesserung der Prognosequalität um ca. 5 %, wie in Tabelle 7.8 zu sehen ist. Im Rahmen der Optimierung wurde der Prognose der übernächsten Phase mit BN1 nicht wie bisher die Ergebnisse der Phasenprognose für die nächste Phase mit BN1 sondern mit XGB+DET+SG+ÖPNV zugrunde gelegt.

Die Ergebnisse für die Phasenprognose der überübernächsten Phase, siehe Tabelle J.5 im Anhang zeigten einen weiteren Abfall der Prognosequalität, jedoch in einem geringeren Maß als zwischen der Phasenprognose der nächsten und übernächsten Phase. Eine Verwendung dieser Prognosen erscheint deshalb nicht sinnvoll.

Tabelle 7.8: Prognoseergebnisse für die übernächste Phase (KKR [%])

KP	M	SP	Phasenidentifikation	Phasenprognose		
				BN1	XG+DET+SG(+ÖPNV)	XGB+BN1
020	9	1	SZ	75	46	81
020	9	1	Doku	76	9	80
020	9	1	Wei+,A	80	42	86
020	11	1	SZ	77	19	81
020	11	1	Doku	79	19	82
020	11	1	Wei,A	82	46	87
722	9	1	SZ	28	31	52
722	9	1	Doku	50	9	57
722	9	1	Wei	48	11	60
722	11	1	SZ	39	9	52
722	11	1	Doku	40	32	49
722	11	1	Wei+,A	38	6	50
020	9	2	SZ	81	15	84
020	9	2	Doku	84	11	86
020	9	2	Wei,A	84	55	87
020	11	2	SZ	80	10	84
020	11	2	Doku	82	15	86
020	11	2	Wei,A	81	28	85
722	9	2	SZ	46	49	61
722	9	2	Doku	49	46	64
722	9	2	Wei+,A	44	50	58
722	11	2	SZ	49	41	52
722	11	2	Doku	54	45	54
722	11	2	Wei+,A	48	31	50
020	9	3	SZ	76	9	78
020	9	3	Doku	78	8	80
020	9	3	Wei,A	79	44	81
020	11	3	SZ	77	8	81
020	11	3	Doku	79	6	83
020	11	3	Wei,A	80	47	83
722	9	3	SZ	38	(22)	46
722	9	3	Doku	33	(38)	46
722	9	3	Wei	55	17	59
722	11	3	SZ	41	(11)	50
722	11	3	Doku	40	(41)	48
722	11	3	Wei,A	38	(12)	48

7.5 Methoden- und Datenmodellvergleich mit grundlegenden Datenmodellen: Phasenschaltzeitprognose

Die zweite Zielgröße, die für SZP zu prognostizieren ist, ist der Zeitpunkt, zu dem eine Phase beginnen wird (Phasenschaltzeitprognose). Die Zielgröße wird in den Sekunden bis zum Phasenbeginn, also als Countdown dargestellt. Basierend auf dem vorangegangenen Abschnitt haben Voruntersuchungen die Auswahl XGB+SG für den KP 722 und XGB+SG+DET für den KP 020 als (Daten-)Modelle bestätigt.

Die Tabellen 7.9 und 7.10 zeigen die Ergebnisse für den nächsten und übernächsten Phasenbeginn für SP 2. Die Ergebnisse für SP 1 und 3 sind in den Tabellen J.9 und J.10 sowie für den überübernächsten Phasenbeginn in Anhang J.11 zu sehen. Zusätzlich zur KKR wird die Abweichung vom Toleranztrichter Q_{Pr} mit berücksichtigt. Je kleiner Q_{Pr} , desto weniger Prognosen liegen außerhalb des Trichters.

Tabelle 7.9: Prognoseergebnisse für die Schaltzeitpunkte der nächsten Phase, SP 2 (KKR [%])

KP	M	SP	P-Ident.	KKR	Q_{Pr}
020	9	2	SZ	81	9
020	9	2	Doku	82	8
020	9	2	Wei,A	81	9
020	11	2	SZ	86	7
020	11	2	Doku	87	7
020	11	2	Wei,A	86	8
722	9	2	SZ	77	8
722	9	2	Doku	75	8
722	9	2	Wei	78	8
722	11	2	SZ	69	11
722	11	2	Wei+,A	69	11
002	9	2	SZ	86	7
002	9	2	Wei+	84	7
002	11	2	SZ	88	6
002	11	2	Wei+	87	6

Tabelle 7.10: Prognoseergebnisse für die Schaltzeitpunkte der übernächsten Phase, SP 2 (KKR [%])

KP	M	SP	P-Ident.	KKR	Q_{Pr}
020	9	2	SZ	28	53
020	9	2	Doku	30	49
020	9	2	Wei,A	27	55
020	11	2	SZ	28	53
020	11	2	Doku	32	50
020	11	2	Wei,A	28	55
722	9	2	SZ	15	55
722	9	2	Doku	12	58
722	9	2	Wei	14	56
722	11	2	SZ	17	60
722	11	2	Wei+,A	14	57
002	9	2	SZ	44	35
002	9	2	Wei+	38	39
002	11	2	SZ	46	31
002	11	2	Wei+	44	32

Insgesamt werden für die Phasenschaltzeitprognose geringere KKR erzielt als für die Phasenprognose. Die geeignetsten Phasenmerkmale zeigen trotzdem eine korrekte Vorhersage in mindestens drei Vierteln der Fälle für die nächste Phase. Zudem wird der Toleranztrichter für die Vorhersage der Zeit bis zum Phasenbeginn bis auf wenige Ausnahmen eingehalten, was die geringen Werte für Q_{Pr} zeigen.

Wie im vorangegangenen Abschnitt zeigen sich für die Vorhersage der übernächsten und überübernächsten Phase jeweils ein deutlicher Qualitätsverlust in KKR und Q_{Pr} . Es werden nur noch etwa ein Viertel der übernächsten und nur sehr wenige Phasen der überübernächsten Phase korrekt vorhergesagt und die Prognosen befinden sich zum Großteil außerhalb der Toleranztrichter. Die Werte im unteren einstelligen %-Bereich implizieren sogar Zufallstreffer. Die insgesamt höhere Vorhersagequalität für KP 002 kann beispielsweise durch FG-Freigaben, die fest in den Phasenablauf integriert sind, ohne ein Anforderung für benötigten, begründet werden.

7.6 SZP mit Phaseninformation

Wie sich die Prognosequalität verändert, wenn die Datenmodelle um Phasenmerkmale erweitert werden, zeigen die folgenden Ergebnisse. Ihre Gesamtheit von mehr als 1.000 Experimenten zeigt, dass für den KP Katzensprung die KKR ca. 50% beträgt und einen RMSE von ca. 20 Sekunden aufweist. Betreffend der Druseltalstraße liegen die KKR bei 20% bis 25% und der RMSE bei ca. 35 Sekunden. Im Bereich der Toleranz von 10%, welche für die Akzeptanz angenommen werden, liegen etwa zwei Drittel der Prognosen innerhalb der Toleranz. Im Folgenden werden die Ergebnisse beider Knotenpunkte besprochen. Die Ergebnisse des Katzensprungs sind in Anhang K zu finden. Die Ergebnisse der Druseltalstraße werden hier dargestellt.

In den Tabellen 7.11 und 7.12 sind die KKR und der RMSE für alle Datenmodelle und ausgewählte Zielgrößen dargestellt. Die aktuelle Phase bzw. der Signalisierungszustand wird mit +0 abgekürzt, die für die nächste Phase bzw. den nächsten Signalisierungszustand mit +1 usw. Ein geringes Verkehrsaufkommen und selten aktuierte SG vermindern die Vorhersagequalität. Außerdem ist bei diesen SG eine höhere Empfindlichkeit gegenüber redundanten Merkmalen zu beobachten. An der Druseltalstraße betrifft dies SG 5 und SG 7. Sie biegen in die Hauptrichtung ein respektive von ihr ab. SG 3 ist eine der Hauptrichtungen. SG 13 signalisiert FG und SG 24 den Radverkehr.

Die Schätzungen von GA und GE zeigen unterschiedliche KKR, je nach der Abfolge der SZ in Bezug auf die Signalisierung der Hauptrichtungen, siehe Abbildung 7.9. Der RMSE ist von diesem Effekt nicht betroffen. Die allgemeine Vorhersagequalität wird dadurch kaum beeinträchtigt.

Am Katzensprung stellen alle Richtungen hoch frequentierte Hauptrouten dar. SG 13 signalisiert FG und SG 20 die Straßenbahn. Daher ist der Unterschied zwischen GA und GE geringer, und die allgemeine Vorhersagequalität höher.

Die Tabellen 7.11 und 7.12 zeigen, dass die vier Datenmodelle SZ+0, Doku+0 und SZ+1, Doku+1 bessere Ergebnisse erzielen als SZ+2, Doku+2 und SZ+3, Doku+3 und alle in der Regel besser sind als R. Der Friedman-Nemenji-post-hoc-Test bestätigt diese Beob-

Tabelle 7.11: KKR [%] der SZP mit Phaseninformation zu KP 722

SG	Doku+0	SZ+0	Doku+1	SZ+1	Doku+2	Doku+3	R
13 GA	17,10	17,10	18,90	17,80	16,80	16,20	7,80
13 GE	41,80	41,70	38,10	40,80	36,50	35,80	34,60
24 GA	29,00	29,10	29,00	28,20	28,50	28,30	22,60
24 GE	18,50	18,70	18,60	16,50	15,40	14,70	9,80
3 GA	21,60	21,80	23,50	20,50	22,10	20,20	8,30
3 GE	40,80	40,50	37,60	42,90	29,40	35,20	32,40
5 GA	29,70	29,70	29,50	32,10	28,80	28,50	21,50
5 GE	18,40	18,70	18,80	15,30	15,40	14,70	9,60
7 GA	17,30	17,60	17,60	17,90	15,10	14,10	7,50
7 GE	14,70	15,10	14,50	13,40	12,70	12,20	6,90

Tabelle 7.12: RMSE [s] der SZP mit Phaseninformation zu KP 722

SG	Doku+0	SZ+0	Doku+1	SZ+1	Doku+2	Doku+3	R
13 GA	21,60	21,80	21,40	21,00	21,50	21,90	24,50
13 GE	20,60	20,50	20,50	19,60	20,50	20,60	23,20
24 GA	24,50	24,30	24,50	22,70	24,50	24,40	26,30
24 GE	24,80	24,90	24,70	23,00	23,90	24,40	27,20
3 GA	20,40	20,40	20,10	20,10	21,10	19,80	22,70
3 GE	19,60	19,50	20,10	19,80	22,20	19,90	22,00
5 GA	24,40	24,20	24,60	25,60	24,50	24,50	27,30
5 GE	24,80	25,30	24,80	26,10	24,20	24,50	27,30
7 GA	76,40	76,30	76,10	68,50	77,80	83,50	79,90
7 GE	75,60	76,10	74,50	67,00	75,70	81,50	79,30

achtung, siehe Abbildungen 7.10 und 7.11 sowie die Tabellen 7.13 und 7.14. Die Abbildungen 7.10 und 7.11 zeigen die Vorhersagequalität in der relativ zueinander gesetzten Rangfolge des Friedman-Nemenji-post-hoc-Tests. SZ+0 schneidet am besten und R am schlechtesten ab.

Die Tabellen 7.13 und 7.14 zeigen die Ergebnisse des Friedman-Nemenji-post-hoc-Tests in je einer Tabelle. Je höher die Zahl ist, desto höher ist die Ähnlichkeit der Vorhersagequalität. Bei einem Signifikanzniveau von 5%. SZ+0, Doku+0 und SZ+1, Doku+1 verbessern die Vorhersagequalität meist signifikant in Bezug zu R. Die Vorhersagequalität wird also durch die Verwendung von SZ- bzw. Phasenmerkmalen erhöht.

Ob SZ- oder Phasenmerkmale hilfreicher sind, hängt von der Signalgruppe und dem Verkehrsaufkommen ab. Durch die Merkmalswahl, wird die Qualität der Schätzung, einschließlich des Horizonts, beeinflusst. Für relativ kurze Horizonte liefern SZ bessere Ergebnisse als Phasen.

Dies ist auf die längere Dauer der Phasen zurückzuführen. Da die Datenmodelle alle Detektor- und ÖPNV-Daten enthalten, ist davon auszugehen, dass die Vorhersagequalität durch eine geeignete Merkmalsselektion, bei der redundante Daten eliminiert werden, insgesamt steigt, siehe auch Kapitel 6.

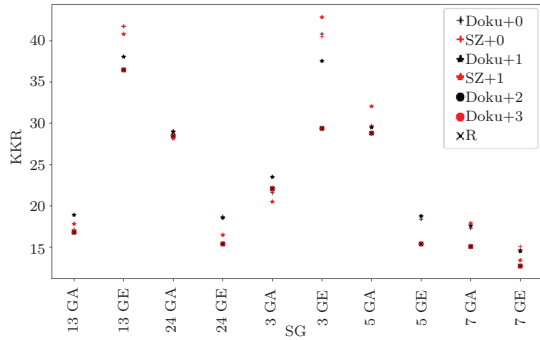


Abbildung 7.9: KKR [%] der Signalgruppen von 722 für verschiedene Datenmodelle



Abbildung 7.10: Skaliertes Ranking des Friedman-Nemenji-post-hoc-Tests für KKR [%] mit LSA 722

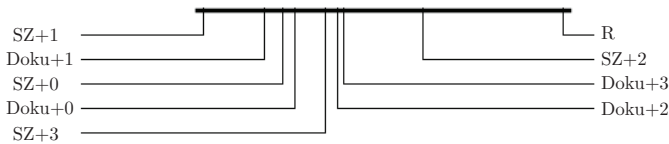


Abbildung 7.11: Skaliertes Ranking des Friedman-Nemenji-post-hoc-Tests für RMSE [s] mit LSA 722

Tabelle 7.13: Ergebnisse des Friedman-Nemenji-post-hoc-Tests zu KP 722KKR [%]

	Doku+0	SZ+0	Doku+1	SZ+1	Doku+2	SZ+2	Doku+3	R
Doku+0	1	0,90	0,90	0,90	0,66	0,08	0,08	0,00
SZ+0	0,90	1	0,90	0,90	0,35	0,02	0,02	0,00
Doku+1	0,90	0,90	1	0,90	0,46	0,03	0,03	0,00
SZ+1	0,90	0,90	0,90	1	0,86	0,18	0,18	0,00
Doku+2	0,66	0,35	0,46	0,86	1	0,90	0,90	0,04
SZ+2	0,08	0,02	0,03	0,18	0,90	1	0,90	0,51
Doku+3	0,08	0,02	0,03	0,18	0,90	0,90	1	0,51
R	0,00	0,00	0,00	0,00	0,04	0,51	0,51	1

Tabelle 7.14: Ergebnisse des Friedman-Nemenji-post-hoc-Tests zu KP 722, RMSE [s]

	Doku+0	SZ+0	Doku+1	SZ+1	Doku+2	SZ+2	Doku+3	R
Doku+0	1	0,90	0,90	0,90	0,90	0,71	0,90	0,01
SZ+0	0,90	1	0,90	0,90	0,90	0,61	0,90	0,01
Doku+1	0,90	0,90	1	0,90	0,90	0,46	0,90	0,00
SZ+1	0,90	0,90	0,90	1	0,66	0,08	0,61	0,00
Doku+2	0,90	0,90	0,90	0,66	1	0,90	0,90	0,06
SZ+2	0,71	0,61	0,46	0,08	0,90	1	0,90	0,61
Doku+3	0,90	0,90	0,90	0,61	0,90	0,90	1	0,08
R	0,01	0,01	0,00	0,00	0,06	0,61	0,08	1

7.7 Schlussfolgerungen

Zusammenfassend lässt sich feststellen, dass die Einbeziehung von Signalzuständen respektive Phasen die Vorhersagequalität der Schaltzeitschätzung von Lichtsignalanlagen deutlich verbessert. Ob das Merkmal des aktuellen oder nachfolgenden Signalzustandes am besten ist, hängt von der jeweiligen Signalgruppe ab. Dies spart aufgrund der geringeren Zahl an Merkmalen gegenüber der Zahl der einzelnen SG-Merkmale Rechenzeit für die Prognose. Eine Integration der Wahrscheinlichkeit beziehungsweise Unsicherheit der Vorhersage der Phase als Eingangsmerkmal erscheint zusätzlich sinnvoll. Konsequenz für das Konzept ist, dass es basierend auf den hier dargestellten Experimenten sinnvoll erscheint (prognostizierte) Phasenmerkmale in SZP stets zu integrieren. Vor diesem Hintergrund erscheint die Integration von Detektorinformation über ähnlich den Phasen gebündelte Merkmale als sinnvoll.

Literaturverzeichnis

- [ABR64] AIZERMAN, M. A. ; BRAVERMAN, E. M. ; ROZONOER, L. I.: The probability problem of pattern recognition learning and the method of potential functions. In: *Automation and Remote Control* 25 (1964), 1175–1190. <https://ci.nii.ac.jp/naid/10022422820/en/>
- [Ae10] AHMED, N. K. ; ET AL.: An Empirical Comparison of Machine Learning Models for Time Series Forecasting. In: *Econometric Reviews* 29 (2010), Nr. 5-6, 594–621. <https://doi.org/10.1080/07474938.2010.481556>
- [Ae16] ALAM, M. ; ET AL. ; KACPRZYK, J. (Hrsg.): *Studies in Systems, Decision and Control*. Bd. 52: *Intelligent Transportation Systems*. Springer, Cham, 2016. – ISBN 978-3-319-28181-0
- [AF16] AXER, S. ; FRIEDRICH, B.: A Methodology for Signal Timing Estimation Based on Low Frequency Floating Car Data: Analysis of Needed Sample Sizes and Influencing Factors. In: *Transportation Research Procedia* 15 (2016), 12, 220–232. <https://doi.org/10.1016/j.trpro.2016.06.019>
- [Alp19] ALPAYDIN, E.: *Maschinelles Lernen*. De Gruyter Oldenbourg, 2019. – ISBN 978-3-110-74019-6
- [Alt92] ALTMAN, N. S.: An introduction to kernel and nearest-neighbor nonparametric regression. In: *The American Statistician* 46 (1992), Nr. 3, 175–185. <https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879>
- [Ana20] ANACONDA SOFTWARE TEAM: *Anaconda Software Distribution*. <https://docs.anaconda.com/>. Version: Vers. 2-2.4.0 2020
- [APF15] AXER, S. ; PASCUCCI, F. ; FRIEDRICH, B.: Estimation of traffic signal timing data and total delay for urban intersections based on low frequency floating car data. In: *Proceedings Mobil.TUM 2015, 30. Juni bis 1. Juli, München, 2015*
- [Aun21] AUNKOFER, B.: *Maschinelles Lernen mit Entscheidungsbaumverfahren – Artikelserie*. <https://data-science-blog.com/blog/2017/02/13/entscheidungsbaumverfahren-artikelserie/>. Version: November 2021

- [AV10] ASADI, B. ; VAHIDI, A.: Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. In: *IEEE transactions on control systems technology* 19 (2010), Nr. 3, 707–714. <https://ieeexplore.ieee.org/document/5454336>
- [Bau21] BAUMGART, L.: *Schaltzeitprognosen für verkehrsabhängige Lichtsignalanlagen mit Methoden des maschinellen Lernens unter Berücksichtigung benachbarter Knotenpunkte*, Bachelorarbeit, Universität Kassel. Dezember 2021
- [BC20] BREIMAN, L. ; CUTLER, A.: *Random Forests*. https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. Version: Oktober 2020
- [Be84] BREIMAN, L. ; ET AL.: Classification and Regression Trees. In: *Wadsworth, Belmont, CA* (1984)
- [Be06a] BOLTZE, M. ; ET AL.: Analyse und Bewertung neuer Forschungserkenntnisse zur Lichtsignalsteuerung / Technische Universität Darmstadt, Fachgebiet Verkehrsplanung und Verkehrstechnik. 2006 (V 149). – Forschungsbericht. – Berichte der Bundesanstalt für Straßenwesen, Verkehrstechnik, Bremerhaven: Wirtschaftsverlag N. W. Verlag für neue Wissenschaft, ISSN: 0943-933
- [Be06b] BOLTZE, M. ; ET AL.: Lichtsignalsteuerung in Japan. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* (2006), Nr. 9, 531-537. <https://www.researchgate.net/profile/Manfred-Boltze/publication/275956234>
- [Be09] BRAUN, R. ; ET AL.: TRAVOLUTION-Netzweite Optimierung der Lichtsignalsteuerung und LSA-Fahrzeug-Kommunikation. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* (2009), Nr. 6, 365-374. <https://mediatum.ub.tum.de/doc/963490/file.pdf>
- [Bel1a] BLEY, O. ; ET AL.: Kooperative Optimierung von Lichtsignalsteuerung und Fahrzeugführung. In: *AAET 2011 Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel*, 2011
- [Bel1b] BOLTZE, M. ; ET AL.: AMONES: Anwendung und Analyse modellbasierter Netzsteuerungsverfahren in städtischen Straßennetzen-Teil 2: Problemstellung und Methodik. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* (2011), Nr. 5, S. 293
- [Ben92] BENNETT, K. P.: Robust linear programming discrimination of two linearly separable sets. In: *Optimization Methods and Software* 1 (1992), 23–34. <https://doi.org/10.1080/10556789208805504>
- [BF07] BOLTZE, M. ; FRIEDRICH, B.: Innovation in der Lichtsignalsteuerung: Die Neufassung der Richtlinien für Lichtsignalanlagen (RiLSA). In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* (2007), Nr. 4, 192–

197. https://www.verkehr.tu-darmstadt.de/media/verkehr/fgvv/prof_boltze/BoVeroeff91.pdf
- [BF14] BARTHAUER, M. ; FRIEDRICH, B.: Evaluation of a signal state prediction algorithm for car to infrastructure applications. In: *17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 Juli 2014, Sevilla, Spanien* Bd. 3, 2014. – ISSN 2352–1465, 982-991
- [BGV92] BOSER, B. E. ; GUYON, I. M. ; VAPNIK, V. N.: A training algorithm for optimal margin classifiers. In: *D. Haussler (Ed.), Proceedings Fifth Annual Workshop on Computational Learning Theory (COLT)*, 1992, S. 144–152
- [Bis09] BISHOP, C. M. ; JORDAN, M. (Hrsg.) ; KLEINBERG, J. (Hrsg.) ; SCHÖLKOPF, B. (Hrsg.): *pattern recognition and machine learning*. 8. Springer, 2009 (Information Science and Statistics). – ISBN 978-0387-31073-2
- [BK93] BUSCH, F. ; KRUSE, G.: MOTION - Ein neues Verfahren für die städtische Lichtsignalsteuerung in Netzen. In: *Tagungsband Heureka 93, Karlsruhe*, 1993
- [Bod17] BODENHEIMER, R.: *Entwicklung und Analyse eines Prognoseverfahrens für verkehradaptive Lichtsignalanlagen*, Technische Fakultät, Friedrich-Alexander Universität Erlangen-Nürnberg, Diss., 2017
- [Bre01] BREIMAN, L.: Random Forests. In: *Machine Learning* 45(1) (2001), S. 5–32
- [Bun19] BUNDESANSTALT F. STRASSENWESEN: *Mobilitäts Daten Marktplatz. mdm-portal.de/*. Version: April 2019
- [Ce13a] CAMPONOGARA, E. ; ET AL.: Distributed Model Predictive Control. In: *IEEE Control Systems Magazine* 22 (2013), Nr. 1, 44-52. <https://ieeexplore.ieee.org/document/980246/>
- [Ce13b] CHRISTOFIDES, P. D. ; ET AL.: Distributed model predictive control: A tutorial review and future research directions. In: *Computers & Chemical Engineering* 51 (2013), 21-41. <https://doi.org/10.1016/j.compchemeng.2012.05.011>
- [CG16] CHEN, T. ; GUESTRIN, C.: XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA : ACM, 2016 (KDD '16). – ISBN 978-1-4503-4232-2, 785–794
- [CV95] CORTES, C. ; VAPNIK, V. N.: Support vector networks. In: *Machine Learning* 20 (1995), 273–297. <https://doi.org/10.1007/BF00994018>
- [DC10] DE OLIVEIRA, L. B. ; CAMPONOGARA, E.: Multi-agent model predictive control of signaling split in urban traffic networks. In: *Transportation Research Part C: Emerging Technologies* 18 (2010), feb, Nr. 1, 120–139. <https://doi.org/10.1016/j.trc.2009.04.022>

- [De 91] DE MÁNTARAS, R. L.: *A Distance-Based Attribute Selection Measure for Decision Tree Induction*. In: *Machine Learning*. 1991 <https://doi.org/10.1023/A:1022694001379>. – ISBN 0885-6125. – S. 81–92
- [De13] DINKEL, A. ; ET AL.: Cooperative optimization of traffic signal control and driver assistance outside urban areas. In: *mobil. TUM* (2013). <https://www.researchgate.net/publication/275716558>
- [De19] DU ; ET AL.: Signal Timing Parameters Estimation for Intersections using Floating Car Data. In: *Journal of the transportation research board* (2019), Nr. 2673(6), 189-201. <https://doi.org/10.1177/0361198119844756>
- [Dre90] DREYFUS, S. E.: Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure. In: *Journal of Guidance, Control, and Dynamics* 13 (1990), Nr. 5, 926–928. <https://doi.org/10.2514/3.25422>. – ISSN 0731–5090
- [DS17] DEY, R. ; SALEM, F. M.: Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks. In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 6-9 Aug., Boston, MA, USA, 2017*
- [Elm90] ELMAN, J. L.: Finding Structure in Time. In: *Cognitive Science* 14 (2) (1990), 179–211. https://doi.org/10.1207/s15516709cog1402_1
- [Eur10] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE: *ETSI EN 302 665 Intelligent Transport Systems (ITS): Communications Architecture*. 2010
- [FC12] FREJO, J. R. D. ; CAMACHO, E. F.: Global Versus Local MPC Algorithms in Freeway Traffic Control With Ramp Metering and Variable Speed Limits. In: *IEEE Transactions on Intelligent Transportation Systems* 13 (2012), dec, Nr. 4, 1556–1565. <http://dx.doi.org/10.1109/tits.2012.2195493>
- [Fe13] FRIEDRICH, B. ; ET AL.: KO-LINE – Kooperative und optimierte Lichtsignalsteuerung in städtischen Netzen / Schlussbericht zum Forschungsprojekt, Förderkennzeichen 19P9002, gefördert vom Bundesministerium für Wirtschaft und Technologie. 2013. – Forschungsbericht
- [Fe14] FAKLER, O. ; ET AL.: Structures of Traffic Management and Control in German Cities. In: *10th ITS European Congress, 16–19 June, 2014*
- [FGS91] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Merkblatt über Detektoren für den Straßenverkehr*. FGSV-Nr. 312, FGSV Verlag GmbH, Köln, 1991
- [FGS93] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Hinweise zur Bevorrechtigung des ÖPNV bei der Lichtsignalsteuerung*. FGSV-Nr. 361, FGSV Verlag GmbH, Köln, 1993

- [FGS99] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Merkblatt für Maßnahmen zur Beschleunigung des ÖPNV mit Straßenbahnen und Bussen*. FGSV-Nr. 114, FGSV Verlag GmbH, Köln, 1999
- [FGS10] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Richtlinien für Lichtsignalanlagen (RILSA)*. FGSV-Nr. 321, FGSV Verlag GmbH, Köln, 2010
- [FGS14a] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Hinweise für die Lichtsignalsteuerung in Straßennetzen (H LiS)*. FGSV-Nr. 312/2, FGSV Verlag GmbH, Köln, 2014
- [FGS14b] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Hinweise für Qualitätsmanagement an Lichtsignalanlagen*. FGSV Verlag GmbH, Köln, 2014
- [FGS15] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Handbuch für die Bemessung von Straßenverkehrsanlagen (HBS) Teil Stadtstraßen*. FGSV Verlag GmbH, Köln, 2015
- [FGS17] FGSV - FORSCHUNGSGESELLSCHAFT STRASSEN-UND VERKEHRSWESEN: *Beispielsammlung zu den Richtlinien für Lichtsignalanlagen (Rilsa-Beispielsammlung)*. FGSV Verlag GmbH, Köln, 2017
- [FH51] FIX, E. ; HODGES, J.: Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties / USAF School of Aviation Medicine, Randolph Field (Texas), USA. 1951. – Forschungsbericht
- [Fri37] FRIEDMAN, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. In: *Journal of the American Statistical Association* 32 (1937), Nr. 200, 675-701. <https://www.tandfonline.com/doi/abs/10.1080/01621459.1937.10503522>
- [Fri40] FRIEDMAN, M.: A Comparison of Alternative Tests of Significance for the Problem of m Rankings. In: *The Annals of Mathematical Statistics* 11 (1940), Nr. 1, 86 – 92. <https://doi.org/10.1214/aoms/1177731944>
- [Fri99] FRIEDRICH, B.: *Ein verkehrsadaptives Verfahren zur Steuerung von Lichtsignalanlagen*, TU München, Fachgebiet Verkehrstechnik und Verkehrsplanung, Diss., 1999
- [Fro19] FROCHTE, J.: *Maschinelles Lernen: Grundlagen und Algorithmen in Python*. Carl Hanser Verlag GmbH Co KG, 2019. – ISBN 978–3–446–45996–0
- [FS97] FREUND, Y. ; SCHAPIRE, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: *Journal of computer and system*

- sciences* 55 (1997), Nr. 1, 119–139. <https://doi.org/10.1006/jcss.1997.1504>
- [Gar90] GARTNER, N.: OPAC: Strategy for Demand-responsive Decentralized Traffic Signal Control. In: *IFAC Proceedings Volumes* (1990), 241–244. [https://doi.org/10.1016/S1474-6670\(17\)52677-4](https://doi.org/10.1016/S1474-6670(17)52677-4)
- [GB10] GLOROT, X. ; BENGIO, Y.: Understanding the difficulty of training deep feed-forward neural networks. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, 249–256
- [Ge88] GABREN, M. ; ET AL.: SIGMA-Ein Optimierungsverfahren zur koordinierten Lichtsignalsteuerung. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* (1988), Nr. 4
- [Ge20] GENSER, A. ; ET AL.: Enhancement of SPaT-messages with machine learning based time-to-green predictions. In: *9th Symposium of the European Association for Research in Transportation (hEART 2020), online, 3.-4. Feb.*, ETH Zurich, 2020, 1-6
- [Ges87] GESETZGEBER: *Verordnung über den Bau und Betrieb der Straßenbahnen (Straßenbahn-Bau- und Betriebsordnung - BOStrab)*. Bundesministerium der Justiz und für Verbraucherschutz sowie des Bundesamts für Justiz, 1987
- [He81] HUNT, P. B. ; ET AL.: SCOOT-a traffic responsive method of coordinating signals. 1981. – Forschungsbericht
- [He12] HOYER, R. ; ET AL.: Optimierungspotenziale des Verkehrsablaufs durch kooperative Lichtsignalanlagen. In: *Straßenverkehrstechnik, Heft 6/2012, FGSV (Hrsg), Kirschbaum Verlag, Bonn* (2012)
- [He20] HARRIS, C. R. ; ET AL.: Array programming with NumPy. In: *Nature* 585 (2020), September, Nr. 7825, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- [Hec21] HECKMANN, K.: *Erstellung von Datenmodellen zur Phasenprognose von Lichtsignalanlagen mit Methoden des maschinellen Lernens, Masterarbeit, Universität Kassel*. Dezember 2021
- [HF89] HENRY, J. ; FARGES, J.: PROLYN. In: *Proceedings of the 6th IFAC-IFIP-FORS Symposium on Transportation*, 1989, S. 505–507
- [HHS04] HOYER, R. ; HERRMANN, A. ; SCHÖNRÖCK, R.: Modelling of vehicle actuated traffic lights. In: SCHULZE, Th. et a. (Hrsg.): *Simulation und Visualisierung*. Society for Modeling and Simulation International SCS-European Publishing House, Erlangen, San Diego, 2004. – ISBN 3–936150–30–3, S. 285–297

- [HHS06] HOYER, R ; HERRMAN, A. ; SCHOENROCK, R.: Vehicle actuated traffic lights - A challenge to modelling of inner-city networks. In: *ITS World Congress, London, 2006*
- [Hin89] HINTON, G. E.: Connectionist learning procedures. In: *Artificial intelligence* 40.1 (1989), 185-234. <https://doi.org/10.1016/B978-0-08-051055-2.50029-8>
- [HMSN20] HOYER, R ; MILTNER, T. ; SCHÄFER, M. ; NOLL, B.: Vernetzung von Fahrzeugen des ÖPNV mit verkehrabhängigen Lichtsignalanlagen auf der Basis des Kommunikationsstandards ETSI ITS G5. In: *Straßenverkehrstechnik, Kirschbaum Verlag, Bonn* 64 (2020), Nr. 11, S. 747–757
- [Hof89] HOFFMANN, G.: Geschwindigkeitsempfehlungen im Kraftfahrzeug - ein Beitrag zur Kraftstoffeinsparung durch das Informationssystem „Wolfsburger Welle“. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* 33 (1989)
- [Hot33] HOTELLING, H.: Analysis of a complex of statistical variables into principal components. In: *Journal of educational psychology* 24 (1933), Nr. 6, 417. <https://doi.org/10.1037/h0071325>
- [Hot92] HOTELLING, H.: Relations between two sets of variates. Version: 1992. https://doi.org/10.1007/978-1-4612-4380-9_14. In: *Breakthroughs in statistics*. Springer, 1992, 162–190
- [HRJ09] HASTIE, T. ; R., Tibshirani ; J., Friedman: *The Elements of Statistical Learning*. Springer-Verlang, 2009. – ISBN 978–1–4899–0519–2
- [HRZZ09] HASTIE, T. ; ROSSET, S. ; ZHU, J. ; ZOU, H.: Multi-class adaboost. In: *Statistics and its Interface* 2 (2009), Nr. 3, 349–360. <https://dx.doi.org/10.4310/SII.2009.v2.n3.a8>
- [HS97] HOCHREITER, S. ; SCHMIDHUBER, J.: LSTM can solve hard long time lag problems. In: *Advances in neural information processing systems*, 1997, S. 473–479
- [HS17] HECK, J. C. ; SALEM, F. M.: Simplified Minimal Gated Unit Variations for Recurrent Neural Networks. In: *CoRR* (2017). <http://arxiv.org/abs/1701.03452>
- [HS21] HOYER, R. ; SCHNEGANS, L. E.: Prognose von Schaltzeiten verkehrabhängiger Lichtsignalanlagen in Deutschland–Herausforderungen und Ansätze. In: *at-Automatisierungstechnik* 69 (2021), Nr. 11, 931–939. <https://doi.org/10.1515/auto-2021-0097>

- [HSH22] HECKMANN, Kevin ; SCHNEEGANS, Lena E. ; HOYER, Robert: Estimating Future Signal States and Switching Times of Traffic Actuated Lights: In proceedings of the 20th European Transport Congress and 12th Conference on Transport Sciences, June 9-10., Győr, Hungary, 2022. (2022), S. 80–91
- [HSH23] HECKMANN, Kevin ; SCHNEEGANS, Lena E. ; HOYER, Robert: *Stage Prediction of Traffic Lights Using Machine Learning*. https://doi.org/10.1007/978-3-658-39438-7_36. Version: 2023
- [Ie18] IBRAHIM, S. ; ET AL.: Estimating Phase Duration for SpaT Messages. In: *IEEE transactions on intelligent transportation systems* (2018), S. 1–9
- [Int14] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO 21217:2014: Intelligent transport systems - Communications access for land mobiles (CALM) - Architecture*. 2014
- [Jan22] JANOTTA, L.: *Qualitätsbewertung von OCIT-Daten für Schaltzeitprognosen, Bachelorarbeit, Universität Kassel*. 2022
- [Ke97] KONONENKO, I. ; ET AL.: Overcoming the myopia of inductive learning algorithms with RELIEFF. In: *Applied Intelligence* 7 (1997), Nr. 1, 39–55. <https://link.springer.com/article/10.1023/A:1008280620621>
- [Ke02] KELLER, H. ; ET AL.: Verkehrsadaptive Netzsteuerung von Lichtsignalanlagen. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* 46 (2002), Nr. 7
- [Ke11a] KOUKOU MIDIS, E. ; ET AL.: Leveraging Smartphone Cameras for Collaborative Road Advisories. In: *IEEE Transactions on mobile computing* 11 (2011), Nr. 5, 707–723. <http://dx.doi.org/10.1109/TMC.2011.275>
- [Ke11b] KOUKOU MIDIS, E. ; ET AL.: SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory. In: *Proceedings of the 9th international conference on Mobile systems, applications, and services* ACM, 2011, 127–140
- [Ke14] KRUMNOW, M. ; ET AL.: Schaltzeitprognose verkehrsabhängiger Lichtsignalanlagen im Rahmen des Forschungsprojektes EFA 2014/2. In: *Aspekte der Verkehrstelematik – ausgewählte Veröffentlichungen* (2014)
- [Ke16] KATHS, J. ; ET AL.: Leitfaden für die Einrichtung kooperativer Systeme auf öffentlicher Seite. 2016. – Forschungsbericht
- [Ke21] KUNASHKO, A. ; ET AL.: Application and Comparison of Machine Learning Algorithms in Traffic Signals Prediction. In: *17th Scientific and Technical Conference Transport Systems Theory and Practice, September 20-21, 2021*
- [KJ96] KOHAVI, R. ; JOHN, G.: Wrappers for feature subset selection. In: *Artificial-Intelligence, special issue on relevance* 97 (1996), S. 273–324

- [KL15] KINGMA, D. P. ; LEI BA, J.: Adam : A method for stochastic optimization. In: *3rd International Conference for Learning Representations, San Diego*, 2015
- [Kob07] KOBBELOER, D.: *Dezentrale Steuerung von Lichtsignalanlagen in urbanen Verkehrsnetzen*, Universität Kassel, Schriftenreihe Verkehr der Universität Kassel, kassel university press GmbH, Diss., 2007
- [KR92] KIRA, K. ; RENDELL, L. A.: A practical approach to feature selection. Version: 1992. <https://doi.org/10.1016/B978-1-55860-247-2.50037-1>. In: *Machine learning proceedings*. Elsevier, 1992, 249–256
- [Kri17] KRIMMLING, J.: *Ampelsteuerung: Warum die grüne Welle nicht immer funktioniert*. Springer-Verlag, 2017. – ISBN 978–3–658–17320–3
- [Le18] LI, J. ; ET AL.: Feature selection: A data perspective. In: *ACM Computing Surveys (CSUR)* 50 (2018), Nr. 6, 1-45. <https://doi.org/10.1145/3136625>
- [LM08] LI, Y. ; MATHIAS, P.: Net Wide Traffic Signal Control With Recurrent Neural Networks. In: *15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting/ITS America/ERTICO/ITS Japan/TransCore*, 2008
- [LPZG15] LINNHOF-POPIEN, C. ; ZADDACH, M. ; GRAHL, A.: *Marktplätze im Umbruch: Digitale Strategien für Services im Mobilen Internet*. Springer-Verlag, 2015 <http://dx.doi.org/10.1007/978-3-662-43782-7>
- [LS94] LANGLEY, P. ; SAGE, S.: Induction of selective Bayesian classifiers. In: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Seattle, W.A* (1994). <https://doi.org/10.1016/B978-1-55860-332-5.50055-9>
- [Lä19] LÄMMER, S.: *Selbst-Steuerung*. stefanlaemmer.de/de/?content=Selbst-Steuerung. Version: Oktober 2019
- [Lü15] LÜSSMANN, J.: *Koordinierung von Lichtsignalanlagen unter Berücksichtigung der Progressionsgeschwindigkeit und Pulkmanagement auf Basis der C2X-Kommunikation zur Reduzierung von CO2 Emissionen*, Technische Universität München, Diss., 2015
- [Mac02] MACIEJOWSKI, J. M.: *Predictive control: with constraints*. Edinburgh : Pearson education, 2002. – ISBN 0–201–39823–0
- [May14] MAYNE, D. Q.: Model predictive control: Recent developments and future promise. In: *Automatica* (2014), Nr. 50, 2967-2986. <https://doi.org/10.1016/j.automatica.2014.10.128>

- [MD89] MAURO, V. ; DITARANO, C.: UTOPIA. In: *Proc. IFAC/IFORS Conf. on Control, Computers and Communications in Transport*, 1989, 245-255
- [Me08] MENIG, C. ; ET AL.: Der informierte Fahrer-Optimierung des Verkehrsablaufs durch LSA-Fahrzeug-Kommunikation. In: *Heureka '08 Stuttgart. Optimierung in Verkehr und Transport* (2008), März
- [Me18a] MAKRIDAKIS, S. ; ET AL.: The M4 Competition: Results, findings, conclusion and way forward. In: *International Journal of Forecasting* (2018), Oktober, Nr. 34(4), 802-808. <https://doi.org/10.1016/j.ijforecast.2018.06.001>
- [Me18b] MAKRIDAKIS, S. ; ET AL.: Statistical and Machine Learning forecasting methods: Concerns and ways forward. In: *PLoS ONE* 13 (2018), 03. <http://dx.doi.org/10.1371/journal.pone.0194889>
- [MV12] MAHLER, G. ; VAHIDI, A.: Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In: *American Control Conference (ACC) IEEE*, 2012, 6557-6562
- [MW43] MCCULLOCH, W. ; W.PITTS: A Logical Calculus of Ideas Immanent in Nervous Activity. In: *Bulletin of Mathematical Biophysics* 5 (1943), Nr. 4, 115-133. <http://dx.doi.org/10.1007/BF02478259>
- [Nie16] NIELSEN, D.: *Tree boosting with xgboost-why does xgboost win „every“ machine learning competition?*, Masterarbeit. 2016
- [NM14] NEGENBORN, R. R. ; MAESTRE, J. M.: Distributed Model Predictive Control: An Overview and Roadmap of Future Research Opportunities. In: *IEEE Control Systems* 34 (2014), August, Nr. 4, 87-97. <http://dx.doi.org/10.1109/mcs.2014.2320397>
- [Nol19] NOLL, B.: Die Neufassung der Hinweise zu Lichtsignal-Steuerungszentralen und die Digitalisierung der Verkehrssteuerung. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* 63 (2019), Nr. 3
- [NZ21] NGUYEN, C. N. ; ZEIGERMANN, O.: *Machine Learning-kurz & gut: Eine Einführung mit Python, Pandas und Scikit-Learn*. o'Reilly, 2021. – ISBN 978-3-96009-161-5
- [OCI22a] OCIT: *Downloadbereich für alle Dokumente Bezüglich der OCIT-Protokolle*. ocit.org/de/ocit/downloads/. Version: September 2022
- [OCI22b] OCIT: *OCIT Homepage*. ocit.org/. Version: September 2022
- [OCI22c] OCIT: *Übersicht über die OCIT-Schnittstellen*. ocit.org/de/ocit/schnittstellen/. Version: September 2022

- [Ott11] OTTO, T.: *Kooperative Verkehrsbeeinflussung und Verkehrssteuerung an signalisierten Knotenpunkten*. Schriftenreihe Verkehr der Universität Kassel, kassel university press GmbH, 2011
- [OW13] OTTO, T. ; WEICHENMEIER, F.: Self-Learning Algorithm and Signal State Prognosis at Traffic Lights for V2I Applications. In: *20th ITS World Congress ITS Japan*, 2013
- [pan20] PANDAS DEVELOPMENT TEAM: pandas-dev/pandas: Pandas. (2020), Februar. <https://doi.org/10.5281/zenodo.3509134>
- [Pe11] PEDREGOSA, F. ; ET AL.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830
- [Pe19] PASZKE, A. ; ET AL.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. Version: 2019. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, 8024–8035
- [PL01] PEARSON, K ; LINES, L. O.: Planes of Closest Fit to Systems of Points in Space, London Edinburgh Dublin Philos. Mag. In: *J. Sci* 2 (1901), Nr. 11, S. 559–572
- [Pla99] PLATT, J. C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: *ADVANCES IN LARGE MARGIN CLASSIFIERS* Bd. 10, MIT Press, 1999, S. 61–74
- [Pro13] PROTSCHKY, V.: Verfahren zur Prognose des Verhaltens von verkehrsadaptiven Ampelverkehrsknoten mithilfe von Historiendaten. 2013. – Forschungsbericht
- [Pro16] PROTSCHKY, V.: *Schaltzeitprädiktion und Routenoptimierung für die Ampelassistentz in Smart Cities*, Ludwig-Maximiliansuniversität München, Diss., 2016
- [Pro17] PROFF, H.: *Innovative Produkte und Dienstleistungen in der Mobilität*. Springer, 2017. – ISBN 978–3–658–18613–5
- [PWF14] PROTSCHKY, V. ; WIESNER, K. ; FEIT, S.: Adaptive Traffic and Light Prediction and via Kalman and Filtering. In: *IEEE Intelligent Vehicles Symposium (IV)*. Dearborn, Michigan, USA, IEEE, 2014
- [QB03] QIN, S. J. ; BADGWELL, T. A.: A survey of industrial model predictive control technology. In: *Control engineering practice* 11 (2003), Nr. 7, 733–764. [https://doi.org/10.1016/S0967-0661\(02\)00186-7](https://doi.org/10.1016/S0967-0661(02)00186-7)

- [Re11] RAUBITSCHKE, C. ; ET AL.: Predictive driving strategies under urban conditions for reducing fuel consumption based on vehicle environment information. In: *2011 IEEE Forum on Integrated and Sustainable Transportation Systems, 29 Juni-1 Juli IEEE*, 2011, 13–19
- [RH14] RUDOLPH, F. ; HOYER, R.: Performance Analysis of coordinated Signalization using Switching Time Data. In: *10th European Congress on Intelligent Transport Systems – Helsinki*, 2014
- [Ric05] RICHTER, A.: *Geschwindigkeitsvorgabe an Lichtsignalanlagen: technische Aspekte und volkswirtschaftlicher Nutzen*. Deutscher Universitäts-Verlag, 2005. – ISBN 3-8244-0828-7
- [Rob69] ROBERTSON, D.: TRANSYT: A Traffic Network Study Tool. (1969)
- [Roe10] ROESSLER, B.: Status of european project intersafe-2 on cooperative intersection safety. In: *Proceedings of the IEEE 6th International Conference on Intelligent Computer Communication and Processing*, 2010, 381–386
- [Ros57] ROSENBLATT, F.: The Perceptron—a perceiving and recognizing automaton, Report 85-460-1 / Cornell Aeronautical Laboratory. 1957. – Forschungsbericht
- [Sca09] SCATTOLINI, R.: Architectures for distributed and hierarchical Model Predictive Control – A review. In: *Journal of Process Control* 19 (2009), Mai, Nr. 5, 723–731. <https://doi.org/10.1016/j.jprocont.2009.02.003>
- [Sch07] SCHOLL, W.: *Adaptive und kooperative Technologien für den intelligenten Verkehr, Online. aktiv-online.org/index.html*. Version: Februar 2007
- [Sch18] SCHNEEGANS, L. E.: Survey on Urbantraffic Control and Management. In: *Organic Computing Doctoral Dissertation Colloquium 2018*, 2018
- [Sch19] SCHLOTHAUER & WAUER GMBH: *INES+ Startpage. schlothauer.de/softwaressysteme/ines/*. Version: Oktober 2019
- [Sch21] SCHÖPFF, M.: *Vergleich von Schaltzeitprognosen von tts Europe mit der Universität Kassel mit Hilfe von Methoden des maschinellen Lernens, Bachelorarbeit, Universität Kassel*. September 2021
- [sci20] SCIKIT-OPTIMIZE-COMMUNITY: *scikit-optimize. https://scikit-optimize.github.io/stable/*. Version: Oktober 2020
- [SD79] SIMS, A. ; DOBINSON, K.: SCAT the Sydney coordinated adaptive traffic system: philosophy and benefits. In: *Band 2, International Symposium on Traffic Control Systems, Berkeley, California, USA*, 1979, S. 22
- [Se10] STEINER, W. ; ET AL.: Entwurfsgrundlagen für Lichtsignalanlagen und Leitfaden. (2010). <https://trid.trb.org/view/1085057>

- [Se14] SANTA, C. ; ET AL.: Potenziale kooperativer Lichtsignalsteuerung zur Steigerung der Verkehrseffizienz und-sicherheit. In: *Straßenverkehrstechnik, Kirschbaum Verlag GmbH, Bonn* (2014), Nr. 10, 676-683. <https://mediatum.uni-tum.de/1081243>
- [Se16] SCHLINGENSIEPEN, J. ; ET AL.: Autonomic transport management systems—enabler for smart cities, personalized medicine, participation and industry grid/industry 4.0. In: *Intelligent transportation systems-problems and perspectives*. Springer, 2016. – ISBN 978-3-319-19150-8, S. 3-35
- [Sha48] SHANNON, C. E.: A mathematical theory of communication. In: *The Bell System Technical Journal* 27 (1948), Nr. 3, 379-423. <http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- [SP10] SEABOLD, S. ; PERKTOLD, J.: statsmodels: Econometric and statistical modeling with python. In: *9th Python in Science Conference*, 2010
- [SP16] ŚLADKOWSKI, A. ; PAMUŁA, W.: *Intelligent transportation systems-problems and perspectives*. Bd. 303. Springer, 2016. – ISBN 978-3-319-19150-8
- [Sta14] STAMATAKIS, I.: *Traffic light prediction and analysis of the related data availability and usability for different types of traffic light assistance services.*, 2014
- [Sta18] STATISTA GMBH: *Anzahl der gemeldeten Pkw in Deutschland in den Jahren 1960 bis 2018 (Bestand in 1.000)*. [de.statista.com/statistik/daten/studie/12131/umfrage/pkw-bestand-in-deutschland/https://trid.trb.org/view/1320902](https://trid.trb.org/view/1320902). Version: November 2018
- [Sta19] STADT INGOLSTADT, AMT FÜR VERKEHRSMANAGEMENT UND GEOINFORMATION - VERKEHRSMANAGEMENT: *TRAVOLUTION - Grüne Welle*. ingolstadt.de/Rathaus/Verkehr/Verkehrsmanagement/Verkehrssteuerung/index.php?La=1&object=tx,465.923.1&sub=0. Version: Dezember 2019
- [Str19] STRASSENVERKEHRS UND TIEFBAUAMT DER STADT KASSEL: Technische Dokumentationen der LSA / Straßenverkehrs und Tiefbauamt der Stadt Kassel. 2019. – Forschungsbericht
- [Te08] TETTAMANTI, T. ; ET AL.: Model predictive control in urban traffic network management. In: *16th Mediterranean Conference on Control and Automation* IEEE, 2008, 1538-1543
- [Te17] TIETZ, M. ; ET AL.: *skorch: A scikit-learn compatible neural network library that wraps PyTorch*, Juli 2017. <https://skorch.readthedocs.io/en/stable/>

- [TK00] TONG, S. ; KOLLER, D.: Restricted Bayes optimal classifiers. In: *Proceedings 17th National Conference on Artificial Intelligence (2000)*, 658–664. https://www.aaai.org/Papers/AAAI/2000/AAAI00-101.pdf?TB_iframe=true&width=370.8&height=658.8
- [Ue14] UEBEL, S. ; ET AL.: Energieeffizientes Fahren 2014 (EFA2014)-2. Projektphase Erhöhung der Reichweite von Elektrofahrzeugen / Technische Universität Dresden. 2014. – Forschungsbericht
- [Umw21] UMWELTBUNDESAMT, (GERMAN ENVIRONMENT AGENCY): *Entwicklung der energiebedingten Treibhausgasemissionen nach Quellgruppen*. <https://www.umweltbundesamt.de/daten/energie/energiebedingte-emissionen#energiebedingtetreibhausgas-emissionen>. Version: September 2021
- [Uni19] UNIVERSITÄT KASSEL, FACHGEBIET FÜR VERKEHRSTECHNIK UND TRANSPORTLOGISTIK: *VERONIKA*. www.veronika.uni-kassel.de. Version: Dezember 2019
- [Uni20] UNIVERSITÄT KASSEL: *Rechencluster der Universität kassel*. <https://www.uni-kassel.de/its/dienstleistungen/wissenschaftliche-datenverarbeitung/anleitung/zugang-und-nutzung-des-clusters/zugang>. Version: Oktober 2020
- [Vap98] VAPNIK, V. N.: *Statistical learning theory*. Wiley, 1998
- [Vap06] VAPNIK, V. N.: *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006
- [VF20] VINGELMANN, P. ; FITZEK, F. H. P.: *CUDA*. NVIDIA. <https://developer.nvidia.com/cuda-toolkit>. Version: release: 10.2.89, 2020
- [WB12] WEIDL, G. ; BREUEL, G.: Overall probabilistic framework for modeling and analysis of intersection situations. Version: 2012. http://dx.doi.org/10.1007/978-3-642-29673-4_24. In: *Advanced Microsystems for Automotive Applications*. Springer, 2012, 257–268
- [We17] WEIDEMANN, T. ; ET AL.: Optimizing Public Transport by co-operative Traffic Signal Control - Implementation of Signal Programs in Micro Simulation Models. In: *Mobil-TUM-International Conference on Intelligent Transport Systems in Theory and Practice, Munich, July, 2017*, S. 4–5
- [Wei17] WEISHEIT, T.: *Ein Verfahren zur Prognose verkehrabhängiger Schaltzeiten von Lichtsignalanlagen*, Universität Kassel, Schriftenreihe Verkehr der Universität Kassel, kassel university press GmbH, Diss., 2017

- [Wei19a] WEIDEMANN, T.: *Prozessdatenanalyse verkehrsabhängiger Lichtsignalsteuerungen zur Modellierung von Ersatzsteuerungslogiken*, Universität Kassel, Schriftenreihe Verkehr der Universität Kassel, kassel university press GmbH, Diss., 2019
- [Wei19b] WEISHEIT, T.: *SZP von TTS für das Projekt Veronika*. 2019
- [WH17] WEIDEMANN, T. ; HOYER, R.: Lichtsignalsteuerungen - Phasenerkennung und Analyse von Phasenwechselbedingungen mittels LSA-Prozessdaten. In: *Tagungsband HEUREKA17 – Optimierung in Transport und Verkehr, Stuttgart, 22.-23. Mrz.*, 2017
- [WHR86] WILLIAMS, R. J. ; HINTON, G. E. ; RUMELHART, D. E.: Learning representations by back-propagating errors. In: *Nature* 323 (6088) (1986), Oktober, S. 533–536. – ISSN 1476–4687
- [Wit19] WITTPAHL, V.: *Künstliche Intelligenz: Technologien/ Anwendung/ Gesellschaft*. Springer Nature, 2019 <http://library.oapen.org/handle/20.500.12657/23312>
- [Wu99] WU, N.: *Optimierung von Signalzeitenplänen nach dem Gleichgewichtsprinzip*, Ruhr-Universität Bochum, Diss., 1999
- [Wut21] WUTTKE, L.: *Künstliche Neuronale Netze*. <https://datasolut.com/neuronale-netzwerke-einfuehrung/>. Version: November 2021
- [Ze10] ZHAO, Z. ; ET AL.: Advancing feature selection research. In: *ASU feature selection repository* (2010), 1-28. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.642.5862&rep=rep1&type=pdf>
- [Ze17] ZHAO, Y. ; ET AL.: GreenDrive: A Smartphone-based Intelligent Speed AdaptationSystem with Real-time Traffic Signal Prediction. In: *ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPs), 18-21 April , Pittsburgh, PA USA, 2017*, 229-238
- [Zim83] ZIMDAHL, W.: *Wolfsburger Welle Ein Projekt der Volkswagen Forschung / Wolfsburg, Volkswagen Forschung*. 1983. – Forschungsbericht
- [ŻK16] ŻOCHOWSKA, R. ; KAROŃ, G.: ITS services packages as a tool for managing traffic congestion in cities. Version: 2016. http://dx.doi.org/10.1007/978-3-319-19150-8_3. In: *Intelligent Transportation Systems–Problems and Perspectives*. Springer, 2016, 81–103

Symbolverzeichnis

Symbole für Merkmale

\underline{EX}^s	Signalgruppenvektor
$\underline{SG}(k)$	Vektor des gesamten Signalisierungszustands einer KP zum Zeitpunkt k
n	[s] nachgestellte Kennzeichnung für die Verwendung nativer Merkmale
a	vorangestellte Kennzeichnung eines Datenmodells was in allen Darstellungsweisen in jeweils separaten Experimenten getestet wurde
min	sekundengenaue Angabe, vorangestellte Kennzeichnung der Repräsentation in Minuten seit Mitternacht
t_{xw}	[s] vorangestellte Kennzeichnung der Repräsentation in relativen Umlaufsekunden
t_X	[s] vorangestellte Kennzeichnung der Repräsentation in Umlaufsekunden
t_s	[s] vorangestellte Kennzeichnung der Repräsentation in der Zeit seit der letzten Änderung bzw. bis zur nächsten Änderung der Information
AP	[-] Bewertungsgröße für die Phasenidentifikation: Anzahl identifizierter Phasen
AP_{Doku}	[-] Bewertungsgröße für die Phasenidentifikation: Anzahl der Phasen nach Dokumentation
AP_{pseudo}	[-] Bewertungsgröße für die Phasenidentifikation: Anzahl der erfassten Pseudophasen (Anzahl der identifizierten Phasen - Anzahl korrekt identifiziert Phasen)

AP_u	[–] Bewertungsgröße für die Phasenidentifikation: Anzahl der identifizierten PÜ
$AP_{u_{var}}$	[–] Bewertungsgröße für die Phasenidentifikation: Absolute Anzahl der Variationen aller PÜ
AS	[–] Bewertungsgröße für die Phasenidentifikation: Anzahl identifizierter Signalisierungszustände
DET_d	Merkmal von Detektor DET_d
OE_{V_r}	Merkmal von ÖPNV-Richtung $_r$
P	Perzentil der Merkmalsselektion
PP	[%] Bewertungsgröße für die Phasenidentifikation: Prozentuale Übereinstimmung der identifizierten Phasen mit der Dokumentation relativ zu den Dokumentierten Phasen
PP_{u_P}	[%] Bewertungsgröße für die Phasenidentifikation: AProzentuale Übereinstimmung der rekonstruierten Phasenabfolge mit der durch die Phasendefinition der technische Dokumentation implizierten
PS	[%] Bewertungsgröße für die Phasenidentifikation: Prozentuale Übereinstimmung der Abfolge der Signalisierungszustände mit der durch die Phasendefinition der Dokumentation implizierten
SG_s	Merkmal der Signalgruppe SG_s
t_{x_w}	[s] relative Umlaufzeit in der Definition nach Weisheit
$z(k)$	Skalares Merkmal des Signalisierungszustands einer KP zum Zeitpunkt k
Doku	Merkmal des Phasenzustands einer KP entsprechend der technischen Dokumentation
SZ	Merkmal des Gesamtsignalisierungszustands einer KP
We	Merkmal des Phasenzustands einer KP entsprechend der Phasenidentifikation nach Weidemann
We+	Merkmal des Phasenzustands einer KP entsprechend der erweiterten Phasenidentifikation nach Weidemann
We+,A	Merkmal des Phasenzustands einer KP entsprechend der erweiterten Phasenidentifikation nach Weidemann mit der Identifikationsfolge A

Wei,A Merkmal des Phasenzustands einer KP entsprechend der Phasenidentifikation nach Weidemann mit der Identifikationsfolge A

Indizes

*	Index für mögliche Eingangsgrößen für SZP
°	Index für mögliche Ausgangsgrößen für SZP
d	[-] Detektornummer
i, j, n, m	[-] Bedeutung ist vom Bezugssystem abhängig
k	[s] Diskrete Zeit
n_d	[-] Anzahl der Detektoren
n_r	[-] Anzahl der Routen für ÖPNV-Meldungen
n_s	[-] Anzahl der Signalgruppen
r	[-] Routennummer bezüglich ÖPNV-Meldungen
s	[-] Signalgruppennummer

Symbole des maschinellen Lernens

\hat{y}	$[\hat{y}_1, \dots, \hat{y}_k, \dots, \hat{y}_m]^T$ (berechnete/geschätzte) Ausgangs-/Zielgröße (prediction)
l	Lineares Kernel für SVM
r	Radialbasisfunktionen (RBF) Kernel für SVM
$E(x_{i,j})$	[-] Erwartungswert der Klassenhäufigkeit der Klasse j des Merkmals i
f_i	[-] f_i -Maß, gewichtet zwischen R und P
f_n	[-] Menge in Klasse und falsch prognostiziert (fälschlich nicht Teil der Klassenmenge)
f_p	[-] Menge nicht in Klasse und falsch prognostiziert (fälschlich Teil der Klassenmenge)

G_j	[-] Gini-Index
H	[-] Entropie; Erwartungswert des Informationsgehalts
H^p	Prädiktionshorizont von MPR
K	[%] Korrektklassifikationsrate (KKR) (en. accuracy)
$L_{\log}(y)$	[-] Kreuzentropie
$O_{i,j}$	[-] Anzahl der Klassenhäufigkeit eines Merkmals i der Klasse j)
P	[%] Relevanz (en. pecision)
$p_{x_i,y}(x_{i,k}, y_k)$	[%] Wahrscheinlichkeit für das gemeinsame Auftreten von $x_{i,k}$ und y_k bezüglich ihrer gemeinsamen Wahrscheinlichkeitsverteilung $p_{x_i,y}$
$p_{x_i}(x_{i,k})$	[%] Wahrscheinlichkeit für das Auftreten von $x_{i,k}$ bezüglich seiner Wahrscheinlichkeitsverteilung p_{x_i}
R	[%] Sensitivität (en. Recall)
R^2	[-] erklärbare Varianz (engl. <i>explained variance</i>)
r_n	[-] Menge nicht in Klasse und korrekt prognostiziert (nicht als Teil der Klassenmenge)
r_p	[-] Menge in Klasse und korrekt prognostiziert (Teil der Klassenmenge)
X	$X = [\dots, x_i, \dots]$ Matrix eines Datenmodells
x	$[x_1, \dots, x_i, \dots, x_n]$ Stichproben (samples)
x_i	$[x_{i,1}, \dots, x_{i,k}, \dots, x_{i,m}]^T$ Merkmal der Stichprobe i (features)
y	$[y_1, \dots, y_k, \dots, y_m]^T$ (tatsächliche) Ausgangs-/Zielgröße (target-features / label)
$y(x)$	ML-Modell
MAE	[-] mittlere absolute Abweichung (engl. <i>Mean Average Error</i>)
Max error	[-] maximaler Fehler (engl. <i>maximal error</i>)
MSE	[-] mittlere quadratische Abweichung (engl. <i>Mean Squared Error</i>)
MSLE	[-] mittlerer quadratischer, logarithmischer Fehler (engl. <i>Mean Squared Logarithmic Error</i>)
RMSE	[-] Wurzel der mittlere quadratischen Abweichung (engl. <i>Root Mean Squar Error</i>)

Symbole der Verkehrstechnik

α_i	[%] Auslastungs- beziehungsweise Sättigungsgrad eines Knotenpunktes beziehungsweise eines Verkehrsstroms i beziehungsweise eines Fahrstreifens i
τ_B	[s/Fz] Zeitbedarfswert eines Fahrzeugs zum Passieren der Haltlinie
ANF	<i>TRUE/FALSE</i> Anforderung
BEL_d	<i>TRUE/FALSE</i> Belegung von Detektor d
k_i	[Fz/km] Verkehrsdichte einer/mehrerer Fahrstreifen i
n_H	[–] Anzahl der Halte
q_i	[Fz/h] Verkehrsstärke/-belastung eines Verkehrsstroms i beziehungsweise einer/mehrerer Fahrstreifen i
$Q_{ges,s}$	[–] Qualitätsparameter der gesamten Prognosesgüte der Signalgruppe s
$Q_{Pr,s}$	[–] Qualitätsparameter Prognoseabweichung der Signalgruppe s
$Q_{sp,s}$	[–] Qualitätsparameter Prognosesprünge der Signalgruppe s
t_G	[s] Übergangs- beziehungsweise Gelbzeit
$t_{B,d}$	[s] Belegungszeit beziehungsweise Anwesenheitszeit an Detektor d
$t_{F,s}(k)$	[s] nächsten (beziehungsweise zu prognostizierenden) Freigabe- beziehungsweise Grünzeit beziehungsweise Grünphase einer Signalgruppe s betrachtet von Zeitpunkt k
$t_{GA,s}(k)$	[s] zum Beispiel in Umlauf- beziehungsweise UTC-Sekunde; nächsten (beziehungsweise zu prognostizierenden) / letzten Zeitpunkt Grünanfang einer Signalgruppe s betrachtet von Zeitpunkt k
$t_{GE,s}(k)$	[s] zum Beispiel in Umlauf- beziehungsweise UTC-Sekunde; nächsten (beziehungsweise zu prognostizierenden) / letzten Zeitpunkt Grünende einer Signalgruppe s betrachtet von Zeitpunkt k
$t_{L,d}$	[s] (Netto-)Zeitlücke
$t_{R,s}$	[s] Rotzeit, einer Signalgruppe s
t_{RG}	[s] Rot-Gelb-Zeit

$t_{R,s}(k)$	[s] nächsten (beziehungsweise zu prognostizierenden) / letzte Sperrzeit, einer Signalgruppe s betrachtet von Zeitpunkt k
t_U, T_U, TU	[s] Umlaufzeit
$t_{W,d}$	[s] Wartezeit von Verkehrsbeteiligten an Detektor d
t_X	[s] Umlaufsekunde, von einer LSA-Logik vorgegeben
t_Z	[s] Zwischenzeit
v	[km/h] Geschwindigkeit
v_{\max}	[km/h] maximal zulässige Höchstgeschwindigkeit zwischen Knotenpunkten
v_p	[km/h] (mittlere) Progressionsgeschwindigkeit zwischen Knotenpunkten
ZL	{ <i>TRUE</i> , <i>FALSE</i> } Erfüllung von Zeitlückenkriterien
VA, HA, AN, ABF, TÜR	{ <i>TRUE</i> / <i>FALSE</i> /Protokoll} An- und Abmeldungen vom öffentlichen Personennahverkehr und Türsignal sowie Abfahrtsignal

Abkürzungsverzeichnis

AKTIV	Adaptive und Kooperative Technologien für den Intelligenten Verkehr
AdaBoost	adaptive boosting
Ada	AdaBoost
BALANCE	Balancing Adaptive Network Control Method
BN	Bayes'sche Netze
CAM	Cooperative Awareness Message
C-ITS	Cooperative Intelligent Transport System
ITS	Intelligent Transport System
CV	Kreuzvalidierung (<i>engl. Cross Validation</i>)
CFS	Korrelationsbasierte Merkmalswahl (<i>engl. Correlation-based Feature Selection</i>)
CART	Classification And Regression Trees
CCP	Cost Complexity Pruning
DENM	Decentralised Environmental Notification Message
DT	Entscheidungsbäume (<i>engl. Decision Trees</i>)
FG	Zu Fuß gehende Personen
FCD	Floating Car Data
FZ	Festzeit
GA	Grünanfang
GE	Grünende
GRU	Gated Recurrent Unit Network
GLOSA	Green Light Optimal Speed Advisory
mIV	motorisierter Individualverkehr
I2I	Infrastruktur zu Infrastruktur-Kommunikation

KKR	Korrektklassifikationsrate <i>engl. accuracy</i>
kNN	k Nearest Neighbours
KP	Knotenpunkt
LSA	Lichtsignalanlage
LSTM	Long Short Term Memory Network
MLP	Multi Layer Perzeptron
MAP	Straßentopologie
MPR	Modell Prädiktive Regelung
MOTION	Method for the Optimization of Traffic Signals In Online controlled Networks
MB	modellbasiert
ML	Maschinelles Lernen
NB	Naive Bayes
ÖPNV	öffentlicher Personennahverkehr
OCIT	Open Communication Interface for Road Traffic Control Systems
OBU	On-Board-Unit
OPAC	Optimization Policies for Adaptive Control
PCA	Hauptkomponentenanalyse (<i>engl. Principal Component Analysis</i>)
P2P	Direkter Wechsel zwischen Phasen ohne Phasenübergang
PSP	Phasen-Signalzustand-Prognose
PSZP	Phasen-Schalt-Zeitpunkt-Prognose
RiLSA	Richtlinien für Lichtsignalanlagen
RSU	Road Side Unit
RB	regelbasiert
RNN	Recurrent Neural Network
RMSE	Wurzel der mittleren Quadratischen Abweichung (<i>engl. Root Mean Square Error</i>)
RFE	Rekursive Merkmalselimination (<i>engl. Recursive Feature Elimination</i>)
ReLU	Rectified Linear Unit
RBF	Radialbasisfunktionen als SVM Kernel
RF	Random Forest
SZP	Schaltzeitprognose

SVM	Support Vector Machine
SG	Signalgruppe
SPaT	Signal Phase and Timing
SCATS	Sydney Coordinated Adaptive Traffic System
SCOOT	Split Cycle Offset Optimisation Technique
SIGMA	Signal Management
SARIMAX	Seasonal AutoRegressive Integrated Moving Averages with exogenous regressors
SAR	SARIMAX
Stat-raus	Statistische Bereinigung, um Pseudophasen zu entfernen
Stat-rein	Statistische Bereinigung, um bisher nicht erkannte Phasen zu identifizieren
SP	Signalprogramm
TRANSYT	Traffic network Study Tool
TRAVOLUTION	TRAFic eVOLUTION
TTS	Traffic Technology Services Inc. Europe
UTOPIA/SPOT	Urban Traffic Optimization by Integrated Automation / System for Priority and Optimization for Traffic
VA	verkehrsabhängig
V2V	Vehicle to Vehicle Kommunikation
V2I	Vehicle to Infrastructure Kommunikation, umgekehrt I2V
Wei	Phasenidentifikation mit dem Algorithmus nach [Wei19a]
Wei+	Erweitere Phasenidentifikation mit dem Algorithmus nach [Wei19a]
X2X	Systembestandteil-X zu Systembestandteil-X
XGB	eXtreme Gradient Boosting

Abbildungsverzeichnis

1.1	Rest-Rot- und Grünbandanzeige auf Basis einer SZP, Projekt AKTIV [Ott11]	18
2.1	LSA-Programmbestandteile	27
2.2	Schema eines Regelkreises	28
2.3	Beispiel eines Regelkreises für eine LSA	28
2.4	Beispiel für den Zustandsverlauf von Signalgruppe	30
2.5	Beispiel eines Signalzeitenplans mit Detektoren	32
2.6	Steuerungsverfahren von GB und im D-A-CH-Raum	44
2.7	Steuerungsverfahren der USA	45
2.8	Steuerungsverfahren in Deutschland	46
2.9	Umsetzung von regel- und modellbasierten Steuerungsverfahren	48
2.10	GLOSA, Projekt TRAVOLUTION [Me08]	50
3.1	Veranschaulichung von Klassifikation; links: linear; rechts: nichtlinear	60
3.2	Veranschaulichung von Regression; links: linear; rechts: nichtlinear	61
3.3	Vorgehen beim maschinellen Lernen	63
3.4	Vorgehen bei der Datenaufbereitung	64
3.5	Über- und Unteranpassung	66
3.6	Regularisierung von Entscheidungsbäumen	67
3.7	RMSE für das Beispiel aus Abbildung 3.5	67

3.8	Trainingserfolg	67
3.9	Kreuzvalidierung	68
3.10	Raster- (links) und randomisierte Suche (rechts)	69
3.11	Mengen der Konfusionsmatrix, basierend auf der Grundgesamtheit	74
3.12	kNN Beispiel eines zweidimensionalen Merkmalsraumes	78
3.13	Beispiel einer SVM	79
3.14	Beispiel eines Entscheidungsbaumes	80
3.15	Veranschaulichung eine neuronalen Netzes	83
3.16	Veranschaulichung eines Neurons [Wut21]	83
3.17	Plot der logistischen Funktion	84
3.18	Veranschaulichung eines RNN	85
3.19	Veranschaulichung einer Markov-Kette	86
3.20	Veranschaulichung eines Bayes'schen Netzes	87
4.1	Lage der abgelegenen LSA im Stadtgebiet Kassel	94
4.2	Lage der LSA in der Innenstadt von Kassel	95
4.3	Unterbrechungsrate der Umlaufsekunden [Jan22]	99
4.4	Prozentuale Differenzierung der Fehler der Umlaufzeit [Jan22]	99
4.5	Fehler- und Unterbrechungsrate für SG aller betrachteter LSA, SP 2 [Jan22]	101
4.6	Fehler- und Unterbrechungsrate für SG aller betrachteter LSA, SP 4 [Jan22]	101
4.7	Fehler- und Unterbrechungsrate der Detektoren aller betrachteter LSA, SP 2 [Jan22]	103
4.8	Fehler- und Unterbrechungsrate der Detektoren aller betrachteter LSA, SP 4 [Jan22]	103
4.9	Schema des Prognoseverfahrens	105
4.10	Schema der Voranalyse zum Konfigurieren einer Prognose	106
4.11	Ablauf der Datenbereinigung	114

4.12 Ablauf der Merkmalsberechnung 115

4.13 Ablauf der Experimente 118

5.1 Relatives Ranking des Friedman-Nemenji-post-hoc-Test für den KKR, KP 722, SG_s , SP 2, 3 132

5.2 Relatives Ranking des Friedman-Nemenji-post-hoc-Test für den RMSE, KP 722, SG_s , SP 2, 3 132

5.3 Relatives Ranking des Friedman-Nemenji-post-hoc-Test für Q_{PR} , KP 722, SG_s , SP 2, 3 132

5.4 Konfusionsmatrix für SG 24 GE K XGB, KP 722, SG , SP 2 133

5.5 Konfusionsmatrix für SG 24 GE K Ada, KP 722, SG , SP 2 133

5.6 Konfusionsmatrix für SG 24 GE K DT, KP 722, SG , SP 2 133

5.7 Konfusionsmatrix für SG 24 GE K LSTM, KP 722, SG , SP 2 133

5.8 Konfusionsmatrix für SG 24 GE K SVM_r, KP 722, SG , SP 2 134

5.9 Konfusionsmatrix für SG 24 GE R kNN, KP 722, SG , SP 2 134

5.10 Konfusionsmatrix für SG 24 GE K DT, KP 722, SG , SP 2 134

5.11 Konfusionsmatrix für SG 24 GE K LSTM, KP 722, SG , SP 2 134

5.12 Konfusionsmatrix für SG 24 GE K SVM_r, KP 722, SG , SP 2 134

5.13 Konfusionsmatrix für SG 24 GE R kNN, KP 722, SG , SP 2 134

5.14 Modellvergleich KKR und RMSE mit dem TTS-Datenmodell 137

5.15 TTS Params-Modellreferenz vs. original TTS-Modell 138

5.16 Modellvergleich KKR und RMSE mit Rohdaten 139

5.17 Modellvergleich KKR und RMSE mit dem überarbeiteten Datenmodell . . 140

6.1 Konfusionsmatrix der Vorhersage von $t_x SG_{3,GA}$ KP 722 mit der Repräsentation min 154

6.2 Konfusionsmatrix der Vorhersage von $t_x SG_{3,GE}$ KP 722 mit der Repräsentation min 154

6.3	Vorhersage von ${}_{t_x}SG_{3,GA}$ KP 722 von Sekunde 19380 bis 19880 mit der Repräsentation ${}_{min}$	155
6.4	Vorhersage von ${}_{t_x}SG_{3,GE}$ KP 722 von Sekunde 9690 bis 10190 mit der Repräsentation ${}_{min}$	155
6.5	Konfusionsmatrix der Vorhersage von ${}_{t_x}SG_{3,GA}$ KP 002 mit der Repräsentation ${}_{t_s}$	156
6.6	Konfusionsmatrix der Vorhersage von ${}_{t_u}SG_{3,GA}$ KP 002 mit der Repräsentation ${}_{t_x}$	156
6.7	Vorhersage von ${}_{t_x}SG_{3,GA}$ KP 002 von Sekunde 23211 bis 23711 mit der Repräsentation ${}_{t_s}$	157
6.8	Vorhersage von ${}_{t_u}SG_{3,GA}$ KP 002 von Sekunde 23211 bis 23711 mit der Repräsentation ${}_{t_x}$	157
6.9	Konfusionsmatrix, KP 722, SG_s , SG 2 GE, SP 2	161
6.10	Konfusionsmatrix, KP 722, SG , SG 2 GE, SP 2	161
6.11	Konfusionsmatrix, KP 722, SG_s , SG 3 GE, SP 2	161
6.12	Konfusionsmatrix, KP 722, SG , SG 3 GE, SP 2	161
6.13	Konfusionsmatrix, KP722, SG_s , SG24 GE, SP2	161
6.14	Konfusionsmatrix, KP722, SG , SG24 GE, SP2	161
6.15	Vorhersage von $SG_{3,GA}$ von Sekunde 7000 bis 7500 zu Versuch 0	166
6.16	Vorhersage von $SG_{3,GA}$ von Sekunde 7000 bis 7500 zu Versuch 1	167
6.17	Konfusionsmatrizen zu Versuch 1, 3.1 und 3 zur Vorhersage von $SG_{3,GA}$	168
6.18	Permutationswichtigkeit zu Versuch 7.1 zur Vorhersage von $SG_{3,GA}$	169
6.19	Permutationswichtigkeit zu Versuch 7.1 zur Vorhersage von $SG_{3,GE}$	169
6.20	Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 0	171
6.21	Konfusionsmatitzen zu Versuch 0 zur Vorhersage von $SG_{4,GA}$	172
6.22	Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 5	172
6.23	Vorhersage von $SG_{4,GE}$ von Sekunde 18500 bis 19000 zu Versuch 7.1	173
6.24	Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 7.4	173

7.1	Veranschaulichung des Zusammenhangs zwischen SG-Signalen, Signalisierungszustand (SZ) und Phasen(-übergängen), gemeinsam erarbeitet für [HSH22]	176
7.2	Beispielrechnung für das Merkmal Signalisierungszustand	177
7.3	Programmablauf für die Phasenidentifikation nach [We17] [Hec21]	179
7.4	Programmablauf für die Phasenidentifikation mit Ergänzung von [Hec21]	180
7.5	Programmablauf für den Phasenspeicher [Hec21]	181
7.6	Beispiel für die Konsequenz von Pseudophasen für den Prognosehorizont	182
7.7	Abfolge der Algorithmen für ihre Evaluation	185
7.8	Bayes'sche Netze zur Phasenvorhersage [Hec21]	189
7.9	KKR [%] der Signalgruppen von 722 für verschiedene Datenmodelle	199
7.10	Skaliertes Ranking des Friedman-Nemenji-post-hoc-Tests für KKR [%] mit LSA 722	199
7.11	Skaliertes Ranking des Friedman-Nemenji-post-hoc-Tests für RMSE [s] mit LSA 722	199
B.1	Signallageplan der LSA Druseltalstraße / Baunsbergstraße (722) [Str19]	xi
B.2	Signallageplan der LSA Druseltalstraße / Baunsbergstraße (722), Ausschnitt [Str19]	xii
B.3	Phasenfolgeplan der LSA Druseltalstraße /Baunsberstraße (722) [Str19]	xiii
B.4	Signallageplan der LSA Katzensprung (002) [Str19]	xv
B.5	Signallageplan der LSA Katzensprung (002), Ausschnitt [Str19]	xvi
B.6	Phasenfolgeplan der LSA Katzensprung (002) [Str19]	xvii
B.7	Signallageplan der LSA Weser- / Ysenburgstraße (001) [Str19]	xix
B.8	Signallageplan der LSA Weser- / Ysenburgstraße (001), Ausschnitt [Str19]	xx
B.9	Phasenfolgeplan der LSA Weser- / Ysenburgstraße (001) [Str19]	xxi
B.10	Signallageplan der LSA Altmarkt (003) [Str19]	xxiii
B.11	Signallageplan der LSA Altmarkt (003), Ausschnitt [Str19]	xxiv

B.12 Phasenfolgeplan der LSA Altmarkt (003) [Str19]	xxv
B.13 Signallageplan der LSA Holländischer Platz (010) [Str19]	xxvii
B.14 Signallageplan der LSA Holländischer Platz (010), Ausschnitt [Str19]	xxviii
B.15 Phasenfolgeplan der LSA Holländischer Platz (010) [Str19]	xxix
B.16 Signallageplan der LSA Ysenburg- / Schützenstraße (020) [Str19]	xxxi
B.17 Signallageplan der LSA Ysenburg- / Schützenstraße (020), Ausschnitt [Str19]	xxxii
B.18 Phasenfolgeplan der LSA Ysenburg- / Schützenstraße (020) [Str19]	xxxiii
B.19 Signallageplan der LSA Kurt-Wolters- / Artilleriestraße (070) [Str19]	xxxv
B.20 Signallageplan der LSA Kurt-Wolters- / Artilleriestraße (070), Ausschnitt [Str19]	xxxvi
B.21 Phasenfolgeplan der LSA Kurt-Wolters- / Artilleriestraße (070) [Str19]	xxxvii
B.22 Signallageplan der LSA Weser- / Magazinstraße (113) [Str19]	xxxix
B.23 Signallageplan der LSA Weser- / Magazinstraße (113), Ausschnitt [Str19]	xl
B.24 Phasenfolgeplan der LSA Weser- / Magazinstraße (113) [Str19]	xli
B.25 Signallageplan der LSA Schützen- / Am Werr-Wimmelstraße (321) [Str19]	xliii
B.26 Signallageplan der LSA Schützen- / Am Werr-Wimmelstraße (321), Aus- schnitt [Str19]	xliv
B.27 Phasenfolgeplan der LSA Schützen- / Am Werr-Wimmelstraße (321) [Str19]	xlvi
C.1 Signalbild von KP 722 SG 1	xlviii
C.2 Klassenhäufigkeit KP 722, SG 1 GA	xlviii
C.3 Klassenhäufigkeit KP 722, SG 1 GE	xlviii
C.4 Signalbild von KP 722 SG 3	xlviii
C.5 Klassenhäufigkeit KP 722, SG 3 GA	xlviii
C.6 Klassenhäufigkeit KP 722, SG 3 GE	xlviii
C.7 Signalbild von KP 722 SG 4	xlix
C.8 Klassenhäufigkeit KP 722, SG 4 GA	xlix

C.9 Klassenhäufigkeit KP 722, SG 4 GE xlix

C.10 Signalbild von KP 722 SG 5 xlix

C.11 Klassenhäufigkeit KP 722, SG 5 GA xlix

C.12 Klassenhäufigkeit KP 722, SG 5 GE xlix

C.13 Signalbild von KP 722 SG 6 1

C.14 Klassenhäufigkeit KP 722, SG 6 GA 1

C.15 Klassenhäufigkeit KP 722, SG 6 GE 1

C.16 Signalbild von KP 722 SG 9 1

C.17 Klassenhäufigkeit KP 722, SG 9 GA 1

C.18 Klassenhäufigkeit KP 722, SG 9 GE 1

C.19 Signalbild von KP 722 SG 10 li

C.20 Klassenhäufigkeit KP 722, SG 10 GA li

C.21 Klassenhäufigkeit KP 722, SG 10 GE li

C.22 Signalbild von KP 722 SG 13 li

C.23 Klassenhäufigkeit KP 722, SG 13 GA li

C.24 Klassenhäufigkeit KP 722, SG 13 GE li

C.25 Signalbild von KP 722 SG 14 lii

C.26 Klassenhäufigkeit KP 722, SG 14 GA lii

C.27 Klassenhäufigkeit KP 722, SG 14 GE lii

C.28 Signalbild von KP 722 SG 18 lii

C.29 Klassenhäufigkeit KP 722, SG 18 GA lii

C.30 Klassenhäufigkeit KP 722, SG 18 GE lii

C.31 Signalbild von KP 722 SG 20 liii

C.32 Klassenhäufigkeit KP 722, SG 20 GA liii

C.33 Klassenhäufigkeit KP 722, SG 20 GE liii

C.34 Signalbild von KP 722 SG 24	liii
C.35 Klassenhäufigkeit KP 722, SG 24 GA	liii
C.36 Klassenhäufigkeit KP 722, SG 24 GE	liii
C.37 Signalbild von KP 002 SG 1	liv
C.38 Klassenhäufigkeit KP 002, SG 1 GA	liv
C.39 Klassenhäufigkeit KP 002, SG 1 GE	liv
C.40 Signalbild von KP 002 SG 3	liv
C.41 Klassenhäufigkeit KP 002, SG 3 GA	liv
C.42 Klassenhäufigkeit KP 002, SG 3 GE	liv
C.43 Signalbild von KP 002 SG 4	lv
C.44 Klassenhäufigkeit KP 002, SG 4 GA	lv
C.45 Klassenhäufigkeit KP 002, SG 4 GE	lv
C.46 Signalbild von KP 002 SG 5	lv
C.47 Klassenhäufigkeit KP 002, SG 5 GA	lv
C.48 Klassenhäufigkeit KP 002, SG 5 GE	lv
C.49 Signalbild von KP 002 SG 6	lvi
C.50 Klassenhäufigkeit KP 002, SG 6 GA	lvi
C.51 Klassenhäufigkeit KP 002, SG 6 GE	lvi
C.52 Signalbild von KP 002 SG 8	lvi
C.53 Klassenhäufigkeit KP 002, SG 8 GA	lvi
C.54 Klassenhäufigkeit KP 002, SG 8 GE	lvi
C.55 Signalbild von KP 002 SG 10	lvii
C.56 Klassenhäufigkeit KP 002, SG 10 GA	lvii
C.57 Klassenhäufigkeit KP 002, SG 10 GE	lvii
C.58 Signalbild von KP 002 SG 11	lvii

C.59 Klassenhäufigkeit KP 002, SG 11 GA lvii

C.60 Klassenhäufigkeit KP 002, SG 11 GE lvii

C.61 Signalbild von KP 002 SG 12 lviii

C.62 Klassenhäufigkeit KP 002, SG 12 GA lviii

C.63 Klassenhäufigkeit KP 002, SG 12 GE lviii

C.64 Signalbild von KP 002 SG 13 lviii

C.65 Klassenhäufigkeit KP 002, SG 13 GA lviii

C.66 Klassenhäufigkeit KP 002, SG 13 GE lviii

C.67 Signalbild von KP 002 SG 20 lix

C.68 Klassenhäufigkeit KP 002, SG 20 GA lix

C.69 Klassenhäufigkeit KP 002, SG 20 GE lix

C.70 Signalbild von KP 002 SG 23 lix

C.71 Klassenhäufigkeit KP 002, SG 23 GA lix

C.72 Klassenhäufigkeit KP 002, SG 23 GE lix

C.73 Signalbild von KP 002 SG 25 lx

C.74 Klassenhäufigkeit KP 002, SG 25 GA lx

C.75 Klassenhäufigkeit KP 002, SG 25 GE lx

C.76 Signalbild von KP 002 SG 26 lx

C.77 Klassenhäufigkeit KP 002, SG 26 GA lx

C.78 Klassenhäufigkeit KP 002, SG 26 GE lx

C.79 Signalbild von KP 002 SG 27 lxi

C.80 Klassenhäufigkeit KP 002, SG 27 GA lxi

C.81 Klassenhäufigkeit KP 002, SG 27 GE lxi

E.1 Vorhersagequalität, 722, SG_s , SG 2 GE, SP 2 lxxiii

E.2 Vorhersagequalität, KP 722, SG , SG 2 GE, SP 2 lxxiii

E.3	Vorhersagequalität, KP 722, SG_s , SG 3 GE, SP 2	lxxiv
E.4	Vorhersagequalität, KP 722, SG , SG 3 GE, SP 2	lxxiv
E.5	Vorhersagequalität, KP 722, SG_s , SG 16 GE, SP 2	lxxv
E.6	Vorhersagequalität, KP 722, SG_s , SG 16 GE, SP 2	lxxv
E.7	Vorhersagequalität, KP 722, SG_s , SG 24 GE, SP 2	lxxvi
E.8	Vorhersagequalität, KP 722, SG , SG 24 GE, SP 2	lxxvi
G.1	Skaliertes Ranking des Post-hoc-Tests für KKR, KP 722, SG , SP 2, 3	c
G.2	Skaliertes Ranking des Post-hoc-Tests für RMSE, KP 722, SG , SP 2, 3	c
G.3	Skaliertes Ranking des Post-Hoc-Tests für Q_{PR} , KP 722, SG , SP 2, 3	c
G.4	Skaliertes Ranking des Post-hoc-Tests für KKR, KP 002, SG_s , SP 2, 3	ci
G.5	Skaliertes Ranking des Post-hoc-Tests für RMSE, KP 002, SG_s , SP 2, 3	ci
G.6	Skaliertes Ranking des Post-hoc-Tests für Q_{PR} , KP 002, SG_s , SP 2, 3	ci
G.7	Skaliertes Ranking des Post-hoc-Tests für KKR, KP 002, SG , SP 2, 3	cii
G.8	Skaliertes Ranking des Post-Hoc-Tests für RMSE, KP 002, SG , SP 2, 3	cii
G.9	Skaliertes Ranking des Post-hoc-Tests für Q_{PR} , KP 002, SG , SP 2, 3	cii
G.10	Konfusionsmatrix für SG 3 GA K XGB, KP 722, SG , SP 2	ciii
G.11	Konfusionsmatrix für SG 3 GA K Ada, KP 722, SG , SP 2	ciii
G.12	Konfusionsmatrix für SG 3 GA K DT, KP 722, SG , SP 2	ciii
G.13	Konfusionsmatrix für SG 3 GA K LSTM, KP 722, SG , SP 2	ciii
G.14	Konfusionsmatrix für SG 3 GA K SVM _r , KP 722, SG , SP 2	ciii
G.15	Konfusionsmatrix für SG 3 GA R kNN, KP 722, SG , SP 2	ciii
G.16	Konfusionsmatrix für SG 3 GE K XGB, KP 722, SG , SP 2,	civ
G.17	Konfusionsmatrix für SG 3 GE K Ada, KP 722, SG , SP 2	civ
G.18	Konfusionsmatrix für SG 3 GE K DT, KP 722, SG , SP 2	civ
G.19	Konfusionsmatrix für SG 3 GE K LSTM, KP 722, SG , SP 2	civ

G.20 Konfusionsmatrix für SG 3 GE K SVM_r, KP 722, SG, SP 2 civ

G.21 Konfusionsmatrix für SG 3 GE R kNN, KP 722, SG, SP 2 civ

G.22 Konfusionsmatrix für SG 5 GE K XGB, KP 722, SG, SP 2 cv

G.23 Konfusionsmatrix für SG 5 GE K Ada, KP 722, SG, SP 2 cv

G.24 Konfusionsmatrix für SG 5 GE K DT, KP 722, SG, SP 2 cv

G.25 Konfusionsmatrix für SG 5 GE K LSTM, KP 722, SG, SP 2 cv

G.26 Konfusionsmatrix für SG 5 GE K SVM_r, KP 722, SG, SP 2 cv

G.27 Konfusionsmatrix für SG 5 GE R kNN, KP 722, SG, SP 2 cv

I.1 Permutationswichtigkeit zu Versuch 3 zur Vorhersage von $SG_{3,GA}$ cix

I.2 Permutationswichtigkeit zu Versuch 4.1 zur Vorhersage von $SG_{3,GA}$ cx

I.3 Permutationswichtigkeit zu Versuch 10 zur Vorhersage von $SG_{3,GA}$ cx

I.4 Permutationswichtigkeit zu Versuch 1 zur Vorhersage von $SG_{4,GA}$ cxi

I.5 Permutationswichtigkeit zu Versuch 3 zur Vorhersage von $SG_{4,GA}$ cxi

I.6 Permutationswichtigkeit zu Versuch 7.1 zur Vorhersage von $SG_{4,GA}$ cxii

I.7 Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 4.1 cxiii

I.8 Vorhersage von $SG_{4,GE}$ von Sekunde 500 bis 1000 zu Versuch 6.1 cxiii

K.1 Skaliertes Ranking des Post-hoc-Tests für KKR mit LSA 002 cxix

K.2 Skaliertes Ranking des Post-hoc-Tests für RMSE mit LSA 002 cxix

Tabellenverzeichnis

2.1	Übersicht über die Steuerungsverfahren	41
3.1	Konfusionsmatrix mit zugehörigen statistischen Maßen (Zush. = Zusammenhang, en. = englisch)	75
5.1	Ausgewählte Ergebnisse SG_s , GA obere Hälfte, GE untere, 722, Sep., SP 2	128
5.2	Signifikanzen des Friedman-Nemenji-post-hoc-Test-Test SG_s für den RMSE mit KP 722	129
5.3	Signifikanzen des Friedman-Nemenji-post-hoc-Test SG_s für die KKR mit 722	130
5.4	Gemittelte KKR [%], Modellvergleich mit TTS-Datenmodell [Sch21]	137
5.5	Gemittelte KKR [%], Modellvergleich mit Rohdaten [Sch21]	139
5.6	Gemittelte KKR, Modellvergleich mit überarb. Datenmodell	140
6.1	Getestete Eingangsdatenmodelle, Zielgrößen: $SG_{s,GA/GE} \forall s \in S$	151
6.2	Ausgewählte Ergebnisse ${}_aSG_s^n$, KKR, 722, Sep., SP 2	153
6.3	Ausgewählte Ergebnisse ${}_aSG_s^n$, RMSE, 722, Sep., SP 2	153
6.4	Ausgewählte Ergebnisse zu KP 722, Monat Sep., SP 2, ${}_{ts}SG_s$ (eigene), XGBoost	158
6.5	Ausgewählte Ergebnisse zu KP 722, Monat Sep., SP 2, ${}_{ts}SG$ (alle), XGBoost	158
6.6	Zusammenhang zwischen OCIT-Bezeichnern und Signallageplänen für KP 002 und 003	164

6.7	Eingangsdatenmodelle zur Vorhersage der Zielgrößen für SG_3 und SG_4 . . .	164
6.8	Ergebnisse der Experimente zu SG_3	165
6.9	Ergebnisse der Experimente zu SG_4	170
7.1	Evaluationsergebnisse für Wei der grundlegenden Merkmalsidentifikation zu KP 722	185
7.2	Evaluationsergebnisse der grundlegenden Merkmalsidentifikation zu KP 020	186
7.3	Evaluationsergebnisse von Wei+ der Merkmalsidentifikation an KP 722 . . .	187
7.4	Evaluationsergebnisse der Merkmalsidentifikation an KP 020	187
7.5	Erste Prognoseergebnisse für die nächste Phase (KKR [%]), SP 2	192
7.6	Prognoseergebnisse für die nächste Phase, SP 1, 2, und 3 (KKR [%])	193
7.7	Erste Prognoseergebnisse für die übernächste Phase, SP 2 (KKR [%])	194
7.8	Prognoseergebnisse für die übernächste Phase (KKR [%])	195
7.9	Prognoseergebnisse für die Schaltzeitpunkte der nächsten Phase, SP 2 . . .	196
7.10	Prognoseergebnisse für die Schaltzeitpunkte der übernächsten Phase, SP 2	196
7.11	KKR [%] der SZP mit Phaseninformation zu KP 722	198
7.12	RMSE [s] der SZP mit Phaseninformation zu KP 722	198
7.13	Ergebnisse des Friedman-Nemenji-post-hoc-Tests zu KP 722 KKR [%]	200
7.14	Ergebnisse des Friedman-Nemenji-post-hoc-Tests zu KP 722, RMSE [s] . . .	200
A.1	Modellbasierte Verfahren mit Ein- und Ausgangsgrößen	vi
A.2	Anwendungsprojekte zur Schaltzeitprognose	vii
D.1	Beispiel für die Zielgrößen, KP 722, SP 2, Monat Sep.	lxiv
D.2	Beispiel der Zeit- und SG-Merkmale, KP 722, SP 2, Monat Sep.	lxv
D.3	Beispiel der Detektor-Merkmale, KP 722, SP 2, Monat Sep.	lxvi
D.4	Beispiel der Sensor-Merkmale, KP 722, SP 2, Monat Sep.	lxvii

D.5	Beispiel der ÖPNV-Merkmale, KP 2, SP 2, Monat Sep.	lxviii
D.6	Beispiel der Phasen-Merkmale	lxix
E.1	Ergebnisse KP 002, Sep. SP 2 SG_s XGBoost	lxxii
E.2	Ergebnisse zu KP 002, Sep. SP 2 SG XGBoost	lxxii
G.1	Ausgewählte Ergebnisse SG_s , GA obere Hälfte GE untere, KP 002, Monat Sep., SP 2	lxxxviii
G.2	Ausgewählte Ergebnisse SG_s , GA obere Hälfte GE untere, KP 002, Monat Sep., SP 3	lxxxix
G.3	Ausgewählte Ergebnisse SG , GA obere Hälfte GE untere, KP 722, Monat Sep., SP 2	xc
G.4	Ausgewählte Ergebnisse SG , GA obere Hälfte GE untere, KP 722, Monat Sep., SP 3	xcii
G.5	Ausgewählte Ergebnisse SG_s , GA obere Hälfte GE untere, KP 002, Monat Sep., SP 2	xciii
G.6	Ausgewählte Ergebnisse SG_s , GA obere Hälfte GE untere, KP 002, Monat Sep., SP 3	xciii
G.7	Signifikanzen des Post-Hoc Tests SG_s für Q_{Pr} mit KP 722, Monat Sep., SP 2, 3	xciv
G.8	Signifikanzen des Post-Hoc Tests SG für den RMSE mit KP 722, Monat Sep., SP 2, 3	xciv
G.9	Signifikanzen des Post-Hoc Tests SG für die KKR mit KP 722, Monat Sep., SP 2, 3	xcv
G.10	Signifikanzen des Post-Hoc Tests SG für Q_{Pr} mit KP 722, Monat Sep., SP 2, 3	xcv
G.11	Signifikanzen des Post-Hoc Tests SG_s für den RMSE mit KP 002, Monat Sep., SP 2, 3	xcvi
G.12	Signifikanzen des Post-Hoc Tests SG_s für die KKR mit KP 002, Monat Sep., SP 2, 3	xcvii
G.13	Signifikanzen des Post-Hoc Tests SG_s für Q_{Pr} mit KP 002, Monat Sep., SP 2, 3	xcviii

G.14	Signifikanzen des Post-Hoc Tests SG für den RMSE mit KP 002, Monat Sep., SP 2, 3	xviii
G.15	Signifikanzen des Post-Hoc Tests SG für die KKR mit KP 002, Monat Sep., SP 2, 3	xcix
G.16	Signifikanzen des Post-Hoc Tests SG für Q_{Pr} mit KP 002, Monat Sep., SP 2, 3	xcix
H.1	Ausgewählte Ergebnisse ${}_aSG_s^n$, KKR, 002, Sep., SP 2	cvii
H.2	Ausgewählte Ergebnisse ${}_aSG_s^n$, RMSE, 002, Sep., SP 2	cviii
J.1	Erste Prognoseergebnisse für die nächste Phase, SP 1 (KKR [%])	cxvi
J.2	Erste Prognoseergebnisse für die nächste Phase, SP 3 (KKR [%])	cxvii
J.3	Erste Prognoseergebnisse für die übernächste Phase, SP 1 (KKR [%])	cxviii
J.4	Erste Prognoseergebnisse für die übernächste Phase, SP 3 (KKR [%])	cxix
J.5	Erste Prognoseergebnisse für die über-übernächste Phase, SP 1 (KKR [%])	cxx
J.6	Erste Prognoseergebnisse für die über-übernächste Phase, SP 2 (KKR [%])	cxxi
J.7	Erste Prognoseergebnisse für die über-übernächste Phase, SP 3 (KKR [%])	cxxii
J.8	Prognoseergebnisse für die über-übernächste Phase (KKR [%])	cxxiii
J.9	Prognoseergebnisse der Schaltzeitpunkte nächste Phase, SP 1, 3	cxxiv
J.10	Prognoseergebnisse der Schaltzeitpunkte übernächste Phase, SP 1, 3	cxxiv
J.11	Prognoseergebnisse für die Schaltzeitpunkte der über-übernächsten Phase	cxxv
K.1	KKR [%] der PSZP zu KP 002	cxxvii
K.2	RMSE [s] der PSZP zu KP 002	cxxviii
K.3	Ergebnisse des Post-Hoc-Tests der PhasenSZP zu KP 002, KKR	cxxviii
K.4	Ergebnisse des Post-Hoc-Tests der PSZP zu KP 002, RMSE	cxxviii

Anhang

Zu dieser Publikation gehört ein Anhang mit weiteren Informationen zu verwandten Projekten, Beschreibungen der betrachteten Knotenpunkte, Beschreibungen zu einzelnen Details der Versuche und weitere Ergebnisse der Versuche, der online veröffentlicht wurde bei KOBRA (Kasseler OnlineBibliothek, Repository und Archiv), dem digitalen Archiv für die wissenschaftlichen Dokumente der Universität Kassel:

<https://doi.org/10.17170/kobra-202401199387>

Schriftenreihe des Institutes für Verkehrswesen

In der Schriftenreihe Verkehr des Institutes (bis zum Jahr 2002 des Fachgebietes) erschienen bisher folgende Hefte:

Heft Nr.:	Inhalt	Preis [€]
1	Köhler, Uwe: Verkehrsplanung im Widerstreit der Meinungen, 1989	(vergriffen)
2	1. Kasseler Nahverkehrs-Tage: Der ÖPNV zwischen Stadt und Umland, 1990	6,00
3	2. Kasseler Nahverkehrstage: Verknüpfungen von Fern- und Nahverkehr, 1992	6,00
4	Arndt, Karin: Entwicklung eines Verfahrens zur Abschätzung einer potentiellen P+R-Nachfrage im Berufsverkehr, 1993	9,00
5	3. Kasseler Nahverkehrstage: Stößt der öffentliche Verkehr an seine Grenzen ?, 1994	9,00
6	4. Kasseler Nahverkehrstage: Beiträge zum Stadt- und Regionalverkehr, 1996	12,00
7	Strauß, Susanne: City-Logistik - Ein Instrument zur Verringerung des städtischen Güterverkehrs, 1997	(vergriffen)
8	Klein, Angelika: Die ÖPNV Grundgebühr – Ein Instrument zur zukunftsorientierten Gestaltung und Finanzierung des öffentlichen Personennahverkehrs, 1998	12,00
9	5. Kasseler Nahverkehrstage: Nahverkehrspläne - eine erste Bilanz, 1998	12,00
10	6. Kasseler Nahverkehrstage: 10 Jahre Fachgebiet Verkehrssysteme und Verkehrsplanung, 2000	15,00
11	Klotz, Günther: Bahnbaubeteiligung - Eine Untersuchung über die Beteiligung der Bürger bei der Planung von Bahnanlagen am Beispiel der Neubaustrecke der deutschen Bundesbahn im Raum Kassel, 2002	15,00
12	7. Kasseler Nahverkehrstage: Wettbewerb im öffentlichen Verkehr, 2002	15,00
13	Zöllner, Ralf: Einsatzbereiche von Schienenregionalbahnen, 2002	15,00
14	Listl, Gerhard: Anwendung neuer Technologien zur Erfassung des Verkehrsablaufs, 2003	22,00
15	Ober-Sundermeier, Anja: Entwicklung eines Verfahrens zur Stauprognose an Engpässen auf Autobahnen unter besonderer Berücksichtigung von Arbeitsstellen, 2003	18,00

Schriftenreihe des Institutes für Verkehrswesen

Heft Inhalt Nr.:	Preis [€]
16 Winter, Olaf Markus: Analyse und Evaluation von Nahverkehrsplänen und die Aufstellung von Kriterien zur Bewertung von Standards im ÖPNV, 2005	(vergriffen)
17 Institut für Verkehrswesen (Hrsg.): Von den Anfängen bis zur Gegenwart - Verkehrstechnik an der Uni Kassel, Kolloquium anlässlich der Verabschiedung von Prof. Dr.-Ing. Heinz Zackor am 14. Juli 2005	18,00
18 Kobbeloer, Detlef: Dezentrale Steuerung von Lichtsignalanlagen in urbanen Verkehrsnetzen, 2007	29,00
19 Bertocchi, Timo: Einsatzbereiche von ÖPNV-Bedienungsformen im ländlichen Raum, 2009	18,00
20 Herzog, Torsten: Strategien und Potenziale zur Verbrauchsreduzierung bei Verkehrsstaus	29,00
21 Otto, Thomas: Kooperative Verkehrsbeeinflussung und Verkehrssteuerung an signalisierten Knotenpunkten	18,00
22 Leitzke, Christian: Reisezeitermittlung im motorisierten Individualverkehr mit Hilfe drahtloser Kommunikationstechnologien	29,00
23 Kühnel, Carsten: Verkehrsdatenerfassung mittels Floating Car Observer auf zweistreifigen Landstraßen	29,00
24 Nahverkehrs-Tage 2013: Neue Konzepte für Stadt und Land	24,00
25 Nahverkehrs-Tage 2015: Nahverkehr und Freizeit - Neue Kunden durch innovative Konzepte gewinnen	(vergriffen)
26 Rudolph, Felix: Qualitätsanalyse koordinierter Lichtsignalsteuerungen unter Verwendung kooperativ gewonnener Messgrößen, 2016	34,00
27 Weisheit, Toni: Ein Verfahren zur Prognose verkehrsabhängiger Schaltzeiten von Lichtsignalanlagen, 2017	34,00
28 Nahverkehrs-Tage 2017: Digital und Disruptiv - Neue Daten und Methoden für einen kundengerechten ÖPNV	29,00
29 Knese, Dennis: Integration der Elektromobilität in die Stadtplanung und Straßenraum- gestaltung – Lösungsansätze für Strategien, Konzepte und Maßnahmen,	39,00
30 Nahverkehrs-Tage 2019: Finanzierung des öffentlichen Verkehrs: zwischen leistungsorientierten E-Tarifen, preisgünstigen Flats und Drittnutzerfinanzierung	29,00

Schriftenreihe des Institutes für Verkehrswesen

Heft Inhalt Nr.:	Preis [€]
31 Schmitt, Volker: In den ÖPNV integrierte Mitnahmesysteme im ländlichen Raum - Wirkungsanalyse und Bewertung	34,00
32 Weidemann, Tanja: Prozessdatenanalyse verkehrsabhängiger Lichtsignalsteuerungen zur Modellierung von Ersatzsteuerungen für 2-Phasen-Systeme	34,00
33 Saighani, Assadollah: Bewertungsverfahren für einen ökonomischen Vergleich städtischer Verkehrssysteme	39,00
34 Lambrecht, Franz: Entwicklung eines Verkehrssicherheitsprogramms für Radfahrende zwischen 11 und 14 Jahren mit dem Fokus auf metakognitiven Fähigkeiten	39,00
35 Bieland, Dominik: Ableitung von Quelle-Ziel-Matrizen im ÖPNV auf Basis von WLAN- und Bluetooth-Daten	39,00
36 Nahverkehrs-Tage 2023: Verkehrswende und ÖPNV: Wie holen wir alle an Bord, bevor der Zug abgefahren ist?	34,00
37 Schneegans, Lena Elisa: Verfahrensentwicklung für Schaltzeitprognosen an verkehrsabhängigen Lichtsignalanlagen mit Hilfe maschinellen Lernens	39,00

Hefte 1 bis 13 und 16 sind zu bestellen über das:
Fachgebiet Verkehrsplanung und Verkehrssysteme
Universität Kassel
34109 Kassel
Fax-Nr. (0561) 804-3382
E-Mail: k.hager@uni-kassel.de

Hefte 14 und 15, sowie ab Heft 17 sind zu bestellen über die:
kassel university press
E-Mail: order.kup@uni-kassel.de

ISBN 978-3-7376-1162-6



9 783737 611626 >