

Fahrsequenz-Clustering zur datenbasierten Szenarienanalyse für die Validierung urbaner automatisierter Fahrfunktionen

Zur Erlangung des akademischen Grades eines

**DOKTORS DER INGENIEURWISSENSCHAFTEN
(Dr.-Ing.)**

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Lennart Ries

geb. in Saulgau

Tag der mündlichen Prüfung:

02.10.2024

Hauptreferent:

Prof. Dr.-Ing. Eric Sax

Korreferent:

Prof. Dr.-Ing. Sören Hohmann

Zusammenfassung

Automatisierte Fahrfunktionen im urbanen Raum stellen eine Schlüsseltechnologie der Mobilität der Zukunft dar. Für den Einsatz dieser Technologie muss ihre Sicherheit nachgewiesen werden. Dieser Nachweis kann erbracht werden, indem die automatisierte Fahrfunktion in allen relevanten Betriebssituationen getestet wurde. Eine zentrale Herausforderung liegt in der Ermittlung und Auflistung dieser Betriebssituationen. Ein prominenter Ansatz zur Validierung automatisierter Fahrfunktionen basiert auf der umfassenden Analyse und Simulation relevanter Verkehrsszenarien.

Zentrale Voraussetzung für die szenariobasierte Validierung ist das Wissen über im urbanen Verkehr vorkommende Situationen. Die vorliegende Dissertation adressiert die Herausforderung der datenbasierten Szenarienanalyse zur Extraktion dieser Szenarien aus Daten. Es wird ein neuartiges Konzept vorgestellt, das auf einer modellfreien Analyse von realen Verkehrsaufzeichnungen basiert. Kernstück des Konzepts ist ein Verfahren zum Clustering semantisch ähnlicher Teile der Verkehrsaufzeichnungen (hier: Fahrsequenzen). Aus jedem Cluster ähnlicher Fahrsequenzen wird dann ein Szenario für die Validierung abgeleitet. Das Fahrsequenz-Clustering wird in einen Prozess zur strukturierten Sammlung und Verarbeitung der notwendigen Daten eingeordnet. Des Weiteren werden statistische Verfahren beschrieben, um anhand der Clustering-Ergebnisse die Auftretenswahrscheinlichkeiten von Szenarien zu ermitteln sowie die Anzahl der noch unbekanntem Szenarien abzuschätzen.

Das Konzept wird anhand von zwei Datensätzen evaluiert – einem simulierten Datensatz und einer realen Verkehrsaufzeichnung einer urbanen Kreuzung. Die Ergebnisse zeigen, dass das Fahrsequenz-Clustering in der Lage ist, ähnliche

Fahrsequenzen zu gruppieren und so auf die Anzahl und Häufigkeitsverteilung der im Datensatz vorhandenen Szenarien zu schließen. Dabei liegt die Zahl der Szenarien deutlich unter der Zahl der aufgezeichneten Fahrsequenzen.

Die Dissertation leistet somit einen Beitrag zur Entwicklung und Zulassung automatisierter Fahrfunktionen im urbanen Raum. Das Fahrsequenz-Clustering sowie die abgeleiteten Auftretenswahrscheinlichkeiten können für die Auswahl zu testender Szenarien sowie bei der Gewichtung und Bewertung von ermittelten Risiken genutzt werden. Die statistischen Betrachtungen dienen als Säule der Sicherheitsargumentation für die Zulassung einer automatisierten Fahrfunktion, da sie den Grad der Vollständigkeit des betrachteten Datensatzes abschätzen können.

Abstract

Automated driving functions in urban areas are a key technology for the mobility of the future. For the use of this, its safety must be proven. This proof can be provided by testing the automated driving function in all relevant operating situations. A key challenge lies in identifying and listing these operating situations. A prominent approach to validating automated driving functions is based on the comprehensive analysis and simulation of relevant traffic scenarios.

The central prerequisite for scenario-based validation is knowledge of situations that occur in urban traffic. This thesis addresses the challenge of data-based scenario analysis to extract these scenarios from data. A novel concept is presented that is based on a model-free analysis of real traffic records. The core of the concept is a method for clustering semantically similar parts of the traffic recordings (here: driving sequences). A scenario for validation is then derived from each cluster of similar driving sequences. Driving sequence clustering is incorporated into a process for the structured collection and processing of the necessary data. Furthermore, statistical methods are described to determine the probability of occurrence of scenarios based on the clustering results and to estimate the number of scenarios that are still unknown.

The concept is evaluated using two data sets - a simulated data set and a real traffic recording of an urban intersection. The results show that driving sequence clustering is capable of grouping similar driving sequences and thus inferring the number and frequency distribution of the scenarios present in the data set. The number of scenarios is significantly lower than the number of recorded driving sequences.

The thesis thus contributes to the development and approval of automated driving functions in urban areas. The driving sequence clustering and the derived probabilities of occurrence can be used for the selection of scenarios to be tested as well as for the weighting and evaluation of identified risks. The statistical considerations serve as a pillar of the safety argumentation for the approval of an automated driving function, as they can estimate the degree of completeness of the data set under consideration.

Danksagung

Diese Dissertation entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am FZI Forschungszentrum Informatik und in enger Zusammenarbeit mit dem Institut für Technik der Informationsverarbeitung (ITIV) am Karlsruher Institut für Technologie. In dieser intensiven Phase haben mich viele Menschen auf unterschiedlichste Weise unterstützt, die ich im folgenden gerne würdigen möchte.

An erster Stelle möchte ich meinem Doktorvater Prof. Dr.-Ing. Eric Sax sehr herzlich danken. Er ermöglichte mir die Arbeit an spannenden und hochaktuellen Themen mit einem direkten Fokus auf die praktische Anwendung. Zudem konnte ich mich durch seine Betreuung sowohl fachlich als auch persönlich stark weiterentwickeln. Auch die Einbindung in seine fachlich und menschlich exzellente, organisationsübergreifende Doktorandengruppe war eine wertvolle Erfahrung.

Ebenso danke ich Prof. Dr.-Ing. Sören Hohmann für die Übernahme des Korreferats, seine zielgenauen inhaltlichen Rückmeldungen sowie die anregenden Diskussionen. Zudem danke ich dem Prüfungsvorsitzenden Prof. Dr.-Ing. Ahmet Cagri Ulusoy sowie den weiteren Prüfern Prof. Dr. rer. nat. Olaf Dössel und Prof. Dr. Ivan Peric.

Teile der Arbeit sind aus einer Zusammenarbeit des FZI mit der ZF Friedrichshafen AG entstanden. Für die jahrelange, sehr gute und fruchtbare Zusammenarbeit möchte ich mich insbesondere bei Franziska Körtke, Johannes Daube, Lara Turner, Wolfgang Scheffer, Thomas Schulik und Martin Sedlacek bedanken.

Maßgeblich für den Erfolg war auch das tägliche Arbeitsumfeld. Dabei gilt mein Dank dem gesamten Forschungsbereich ESS am FZI, der mich jeden Tag mit Rat und Kritik, wertvollen inhaltlichen Impulsen und der manchmal notwendigen

Ablenkung versorgen konnte. Besonders hervorheben möchte ich hier meinen Vorgesetzten Stefan Otten für seine Unterstützung und sein Verständnis, Jacob Langner und Dr.-Ing. Christian King für die inhaltlichen Diskussionen – zeitweise auf griechischen Inseln – sowie Thilo Braun, Markus Lemmer und Laurenz Adolph für die Mitarbeit bei der technischen Realisierung einiger Forschungsergebnisse. Nicht fehlen darf hier auch Ulrike Beideck: vielen Dank für dein grenzenloses Wissen und deine Ansprechbarkeit in jeder Lebenslage.

Gesondert möchte ich mein Team bei ESM erwähnen: vielen Dank für die großartige Hilfe sowie die geduldige Akzeptanz von Zeitnot, Vergesslichkeit und Stimmungen. Dabei auch ganz besonderer Dank für die enge Zusammenarbeit an Jacqueline Henle und Jacob Langner.

Nicht zuletzt danke ich meinen Eltern Uwe und Claudia und meinen Geschwistern Hannah und Robin für Ermutigung und Aufbauhilfe in schwierigen Zeiten sowie die Korrektur der letzten Fehler. Meiner Partnerin Laura danke ich für ihre unerschöpfliche Geduld, ihre Zuneigung und die unbeschreibliche Unterstützung auf allen Ebenen, durch die diese Arbeit erst möglich wurde.

Karlsruhe, im Herbst 2024

Lennart Ries

Inhaltsverzeichnis

Zusammenfassung	i
Danksagung	v
1 Einleitung	1
1.1 Motivation	1
1.2 Aufbau und Forschungsfragen	6
2 Grundlagen der Entwicklung automatisierter Fahrzeuge	7
2.1 Grundbegriffe automatisierter Fahrzeuge	7
2.1.1 Levels des automatisierten Fahrens	7
2.1.2 Rechtliche Situation in Deutschland	9
2.2 Entwicklungsprozess und Testmethoden	11
2.2.1 Das V-Modell	11
2.2.2 Funktionale Sicherheit	13
2.2.3 Testwerkzeuge im Entwicklungsprozess	15
2.3 Szenariobasierte Validierung	19
2.3.1 Motivation	19
2.3.2 Grundbegriffe des szenariobasierten Testens	21
2.3.3 Prozessschritte in der szenariobasierten Validierung	28
2.3.4 Sicherheit der Sollfunktion	33
2.4 Relevante Grundlagen der Datenanalyse	36
2.4.1 Prozess zur Wissensextraktion aus Datensätzen	36
2.4.2 Distanzmaße für Zeitreihen	38
3 Stand der Wissenschaft zur datenbasierten Szenarienanalyse 43	
3.1 Ziele und Herausforderungen	43

3.1.1	Zentrale Annahmen	43
3.1.2	Ziele der datenbasierten Szenarienanalyse	44
3.1.3	Aspekte der Nutzung von Realdaten	47
3.1.4	Formalisierung	48
3.2	Existierende Ansätze und Datensätze	51
3.2.1	Szenarienanalyse auf Autobahnen	51
3.2.2	Szenarienanalyse in urbanen Bereichen	54
3.2.3	Verfügbare Datensätze	56
3.3	Diskussion des Stands der Wissenschaft	58
3.3.1	Konzeptionelle Problematik des manöverbasierten Ansatzes	58
3.3.2	Szenarienanalyse durch Clustering	60
3.4	Fazit	63
4	Konzept und Entwurf einer Methode für datenbasierte Szenarienanalyse	65
4.1	Prozess für Datensammlung und -verarbeitung	65
4.1.1	Randbedingungen	65
4.1.2	Analyse und Auswahl von Aufzeichnungsorten	66
4.1.3	Vorverarbeitung und Segmentierung der Daten	68
4.1.4	Transformation in Fahrsequenzen	69
4.1.5	Fahrsequenz-Clustering	70
4.1.6	Überführung in nutzbare Szenarioformate	71
4.1.7	Verknüpfung mit Funktionsentwicklung	73
4.2	Szenarienanalyse durch Fahrsequenz-Clustering	74
4.2.1	Anforderungen an Methode	74
4.2.2	Grundidee	75
4.2.3	Algorithmus zum Fahrsequenz-Clustering	77
4.2.4	Fahrsequenz-Vergleich	79
4.3	Bestimmung von Szenario-Auftretenswahrscheinlichkeiten	90
4.3.1	Ableitung von Auftretenswahrscheinlichkeiten aus Cluster-Ergebnissen	91
4.3.2	Schätzung der Abdeckung des Datensatzes	93
5	Prototypische Realisierung des Fahrsequenz-Clusterings	97

5.1	Betrachtung anhand simulierter Daten	97
5.1.1	Versuchsaufbau	97
5.1.2	Versuchsdurchführung	100
5.1.3	Versuchsergebnisse	101
5.2	Erprobung auf realen Daten	104
5.2.1	Datensatz	105
5.2.2	Ergebnisse des Fahrsequenz-Clusterings	107
5.2.3	Ableitungen aus Clustering-Ergebnis	109
6	Evaluation und Diskussion des Konzepts für datenbasierte Szenarienanalyse	117
6.1	Diskussion relevanter Aspekte des Fahrsequenz-Clusterings	117
6.1.1	Beschreibung von Fehlertypen im Fahrsequenz-Clustering	117
6.1.2	Auffälligkeiten in realen Daten	122
6.1.3	Rechendauer des Fahrsequenz-Clustering	125
6.1.4	Diskussion des Fahrsequenz-Clusterings	129
6.2	Diskussion der Nutzbarkeit des Konzepts	133
6.2.1	Anwendung im Entwicklungsprozess	133
6.2.2	Beitrag zur Validierung	135
6.3	Fazit	140
7	Zusammenfassung und Ausblick	143
7.1	Zusammenfassung	143
7.2	Ausblick	146
A	Anhang	149
A.1	Fahraufgabe und Assistenzsysteme	149
A.2	Funktionale Komponenten automatisierter Fahrzeuge	150
A.3	Ergänzende Informationen zur Validierung	156
A.3.1	Tabellen zur Bestimmung des ASIL	156
A.3.2	Beispiel für Abstraktionsstufen von Szenarien	158
A.4	Zusätzliche Erläuterung mathematischer Grundlagen	159
A.4.1	Maschinelles Lernen	159
A.4.2	V-Measure als Vergleichsmetrik für Clustering	162
A.4.3	Umrechnung von globalem Koordinatensystem ins Fahrzeugkoordinatensystem	164

A.5 Simulierter Testdatensatz	166
Abbildungsverzeichnis	177
Tabellenverzeichnis	181
Eigene Veröffentlichungen	183
Betreute Abschlussarbeiten	185
Literaturverzeichnis	187

1 Einleitung

1.1 Motivation

Die Einführung von Fahrzeugen mit automatisierten Fahrfunktionen hat das Potential zur Lösung akuter Probleme. In erster Linie ist hier die Sicherheit im Straßenverkehr zu nennen. Im deutschen Straßenverkehr kam es im Jahr 2019 zu 2,7 Mio. polizeilich aufgenommenen Unfällen, davon 300 000 mit Personenschäden [20]. 88,2% dieser Unfälle wurden durch Fehlverhalten des Fahrzeugführers verursacht [20] und wären möglicherweise bei einem automatisierten System nicht passiert.

Die Automatisierung des öffentlichen Verkehrs bietet zusätzlich weitere Potentiale. Durch die Einsparung des Fahrers können bis zu 50% der Kosten gespart werden [25], was die Preise öffentlicher Verkehrsangebote senkt und den wirtschaftlichen Betrieb eines größeren Angebots ermöglicht. Somit würde sowohl für einkommensschwache als auch für mobilitätseingeschränkte Menschen eine höhere Mobilität und damit Teilhabe ermöglicht. Des Weiteren ist die automatisierte Bedienung von Buslinien eine Lösung für den erwarteten gravierenden Fahrermangel im öffentlichen Verkehr [58]. Analoge Entwicklungen zeigen sich auch bei den angrenzenden Logistik-Bereichen: Sowohl bei klassischen Speditionen¹ als auch bei Zustellern² zeichnet sich ein wachsender Personal- und Fahrermangel ab.

¹ <https://www.tagesschau.de/wirtschaft/unternehmen/speditionen-lkw-fachkraeftemangel-100.html>

² <https://www.tagesschau.de/wirtschaft/verbraucher/post-filialen-briefe-pakete-mangel-land-beschwerden-gesetzesnovelle-101.html>

Hinter diesen Überlegungen steht die zentrale Anforderung, dass automatisierte Fahrfunktionen sicher sind und niemals oder nur sehr selten Unfälle verursachen. Bevor automatisierte Fahrfunktionen in einem Regelbetrieb eingesetzt werden können, muss deren Sicherheit durch die Fahrzeughersteller nachgewiesen werden. Dies stellt eine bisher nicht gelöste Herausforderung dar, da sich automatisierte Fahrzeuge in einem sogenannten offenen Kontext (vgl. Definition 2.19) bewegen, also nicht alle möglichen Betriebssituationen systematisch erfasst und vorab getestet werden können.

Gleichzeitig ist die Entwicklung automatisierter Fahrfunktionen kein sprunghafter, sondern ein evolutionärer Vorgang. Seit Jahren existieren Systeme, die Teile der Fahraufgabe automatisieren, beispielsweise Längs- oder Querführung. Zur Systematisierung dieser Systeme beschreibt die Society of Automotive Engineers (SAE) sechs Level der Automatisierung, von Level 0 mit keiner Automatisierung bis zur Vollautomatisierung in Level 5 (vgl. Abschnitt 2.1.1). Systeme auf Level eins und zwei sind heute bereits serienmäßig verbaut. Ein Beispiel ist der im Jahr 1999 erstmals verkaufte adaptive Abstandsregeltempomat (vgl. Abschnitt A.1) zur Regelung der Längsgeschwindigkeit [126]. Derartige Systeme müssen dauerhaft vom Fahrer überwacht werden, was den Sicherheitsnachweis vereinfacht. Anstatt den gesamten offenen Kontext zu betrachten genügt es hier, die sichere Überwachbarkeit und Übersteuerbarkeit durch den Fahrer sicherzustellen.

Die beschriebenen Vorteile automatisierter Fahrzeuge im öffentlichen Verkehr können jedoch keinen Fahrer als Rückfallebene einplanen, womit mindestens eine Automatisierung auf Level vier erforderlich ist. Dieses Level beschreibt den Zustand der vollständigen Automatisierung in einer begrenzten Operational Design Domain (ODD) (vgl. Definition 2.1). Die ODD beschreibt die Bedingungen, unter der eine automatisierte Funktion sicher betrieben werden kann und umfasst verschiedenen Dimensionen, beispielsweise die Tageszeit, das Wetter oder die räumliche Ausdehnung des Betriebsbereichs.

Durch die Möglichkeit der Begrenzung der ODD wird die Problematik des offenen Kontexts verkleinert, was den Sicherheitsnachweis vereinfacht. Diese Aussage

stützt sich auf aktuelle Entwicklungen im Bereich des automatisierten öffentlichen Verkehrs: Die Firma Waymo bietet seit 2020 einen fahrerlosen Taxi-Service in Phoenix, Arizona an [73]. Hier wurde die ODD explizit eingeschränkt. Der Betriebsbereich (vgl. Definition 2.2) erstreckt sich auf die Vororte von Phoenix, die sich durch breite Straßen und eine geringe Verkehrsdichte auszeichnen. Des Weiteren ist der Betrieb nur bei sonnigem und trockenem Wetter zulässig³ [50]. Diese explizite Einschränkung hat einige implizite Folgen. Durch die Beschränkung auf klare Wetterverhältnisse sinken die Anforderungen an die Sensorik zur Umfeldwahrnehmung (vgl. Abschnitt A.2). Zusätzlich sorgt die räumliche Beschränkung auf homogene Vorstadtstraßen für eine Reduktion der beim Betrieb potentiell auftretenden Situationen. Diese Reduktion wird einerseits durch eine begrenzte Varianz des Straßennetzes gegeben, liegt aber andererseits auch in der geringeren Interaktion mit anderen Verkehrsteilnehmern. Situationen, in denen auf viele Verkehrsteilnehmer gleichzeitig geachtet werden muss, sind durch die geringe Verkehrsdichte nicht zu erwarten.

Vororte sind wichtige Anwendungsbereiche automatisierter Fahrfunktionen zur Lösung des Problems der „letzten Meile“, beispielsweise zum Transport von Personen von ihrer Haustür zur S-Bahn-Station [9].

In Abbildung 1.1 sind die Zielorte aller Taxifahrten über sieben Jahre im Großraum von New York City abgebildet⁴. Diese Karte zeigt, dass in urbanen Ballungsgebieten wie Manhattan ein größerer Bedarf für öffentliche Transportangebote besteht, was diese Gebiete besonders attraktiv für Anbieter solcher Services macht. In Ballungsgebieten bewegt sich typischerweise eine Vielzahl von Menschen auf unterschiedlichste Weise, zu Fuß, auf dem Fahrrad oder per E-Scooter. Für die notwendigen Fähigkeiten automatisierter Fahrfunktionen bedeutet dies, dass die Begrenzung der Interaktionen mit Verkehrsteilnehmern nicht sinnvoll haltbar ist. Der Nachweis der Sicherheit muss also auch für Ballungszentren geführt werden können.

³ Phoenix hat im Schnitt 27 Regentage im Jahr.

⁴ Bild entnommen aus <https://towardsdatascience.com/if-taxi-trips-were-fireflies-1-3-billion-nyc-taxi-trips-plotted-b34e89f96cfa>, abgerufen am 22.03.2023.



Abbildung 1.1: Visualisierung der Zielorte von Taxifahrten im New York City Metropolitan Area von Jahr 2009-2016.

Ein Ansatz für den Sicherheitsnachweis automatisierter Fahrfunktionen liegt in der szenariobasierten Validierung (vgl. 2.3.3). Szenarien beschreiben die Verkehrsteilnehmer, den Kontext der Umgebung wie das Straßennetz, Ziele der Verkehrsteilnehmer sowie Abfolgen von Aktionen und Ereignissen, beispielsweise die Durchführung von Fahrmanövern [114]. Analog zu den anderen Einsatzdomänen wie dem strategischem Management dienen sie dazu, aktuelle und zukünftige Realitäten zu beschreiben [57]. Anhand von Szenarien lassen sich zu Beginn der Entwicklung Anforderungen an Systeme und Komponenten ableiten sowie zum Ende der Entwicklung ein Sicherheitsnachweis führen, indem man aus den Szenarien Testfälle ableitet [81]. Dazu müssen die betrachteten Szenarien die realen Eigenschaften einer ODD umfassen.

Für die Bestimmung der Szenarien werden verschiedene Quellen genutzt (vgl. Kapitel 2.3.3). Eine Quelle ist die Definition von Szenarien durch den Menschen. Hier entscheiden Experten anhand von verfügbaren Informationen (bspw. Kartenmaterial) und persönlichen Erfahrungen über die voraussichtlich in der ODD

auftretenden Szenarien. Da menschliche Experten in ihrem Urteil immer kognitiven Verzerrungen unterliegen [113] und somit potentiell fehlerhafte Annahmen treffen, werden noch weitere Szenario-Quellen benötigt (vgl. Abschnitt 2.3.3). Dazu dient die datenbasierte Szenarienanalyse (vgl. Abschnitt 3), die anhand von Verkehrsaufzeichnungen in der ODD auf dort relevante Szenarien schließt.

Für den Sicherheitsnachweis in urbanen Ballungsgebieten müssen also alle Interaktionen mit anderen Verkehrsteilnehmern in Szenarien repräsentiert werden. Durch die gleichzeitig ablaufende Interaktion mit Verkehrsteilnehmern wie Fahrradfahrern und Fußgängern steigt die Zahl der zu erwartenden Szenarien an. Ebenso werden in verschiedenen Betriebsbereichen unterschiedliche Szenarien auftreten. Verschiedene Städte oder Stadtviertel haben beispielsweise unterschiedliche Häufigkeiten von Fahrradfahrern, sodass eine Varianz in der Häufigkeit von Szenarien mit Fahrradbeteiligung zu erwarten ist. Um eine Aussage über die Sicherheit einer automatisierten Funktion in einem spezifischen Betriebsbereich zu treffen, müssen die Auftretenswahrscheinlichkeiten der Szenarien bekannt sein.

Die aktuelle deutsche Gesetzgebung erlaubt eine Zulassung von Fahrzeugen auf SAE-Level vier in begrenzten Betriebsbereichen (vgl. Abschnitt 2.1.2). Zur Zulassung muss die Sicherheit anhand von Testszenarien und „auf Basis empirischer Datenerhebungen“⁵ nachgewiesen werden. Jedoch wurde bisher⁶ noch kein automatisiertes Fahrzeug auf dieser Basis zugelassen.

In der Literatur und Wissenschaft wird die datenbasierte Szenarienanalyse breit diskutiert (vgl. Abschnitt 3.2). Hier liegt der Fokus jedoch häufig auf der ODD der Autobahn, die sich durch hohe Strukturiertheit auszeichnet. Gegenverkehr oder Fußgänger müssen hier nicht betrachtet werden. Erste Ansätze zur urbanen datenbasierten Szenarienanalyse sind zwar vorhanden, sind bisher jedoch fern von praktischer Anwendbarkeit im Entwicklungsprozess. Damit die datenbasierte Szenarienanalyse einen Mehrwert liefert, sind zum einen Prozesse und Vorgehensweisen zur Datenaufzeichnung notwendig, die mit etablierten Entwicklungsprozessen

⁵ Verordnung zur Regelung des Betriebs von Kraftfahrzeugen mit automatisierter und autonomer Fahrfunktion und zur Änderung straßenverkehrsrechtlicher Vorschriften, Anlage 1 Nummer 10

⁶ Stand Herbst 2023

vereinbar sind. Ebenso werden Analysemethoden benötigt, um die aufgezeichneten Daten effizient und vollständig zu verwerten sowie weitere Erkenntnisse wie die Auftretenswahrscheinlichkeiten der Szenarien im Betriebsbereich abzuleiten. Diese Forschungslücke für Verfahren zur datenbasierten Szenarienanalyse in urbanen Betriebsbereichen wird in dieser Dissertation adressiert.

1.2 Aufbau und Forschungsfragen

Aus der Motivation ergeben sich die zentralen Forschungsfragen der Dissertation:

- (FF₁) Wie sieht ein Prozess zur Datensammlung und -verarbeitung in festen urbanen Betriebsbereichen aus, der die szenariobasierte Validierung ermöglicht?
- (FF₂) Mit welcher Methode können Szenarien vollständig aus realen Verkehrsaufzeichnung extrahiert werden?
- (FF₃) Wie kann mithilfe der datenbasierten Szenarienanalyse die Auftretenswahrscheinlichkeiten von Szenarien in einem bestimmten Betriebsbereich abgeleitet werden?

Der Aufbau der Dissertation ist folgender: Zuerst wird ein Überblick über die Grundlagen der Entwicklung automatisierter Fahrzeuge aus technischer sowie organisatorischer Sicht beschrieben (vgl. Kapitel 2). Darauf folgt eine umfassende Beschreibung von Ansätzen zur Validierung automatisierter Fahrfunktionen mit Fokus auf das szenariobasierte Testen (vgl. Kapitel 2.3). In Kapitel 3 werden verschiedene Ansätze zur datenbasierten Szenarienanalyse beschrieben und deren Anwendbarkeit diskutiert, woraufhin das Konzept zur datenbasierten Szenarienanalyse beschrieben wird (vgl. Kapitel 4). Anschließend wird eine exemplarische Umsetzung des Konzepts beschrieben (vgl. Kapitel 5) und das Konzept anhand der Forschungsfragen evaluiert (vgl. Kapitel 6). Zum Schluss wird ein Ausblick über notwendige weitere Forschung gegeben (vgl. Kapitel 7).

2 Grundlagen der Entwicklung automatisierter Fahrzeuge

2.1 Grundbegriffe automatisierter Fahrzeuge

2.1.1 Levels des automatisierten Fahrens

Durch die Kombination verschiedener FAS kann schon heute ein Teil der Fahraufgabe automatisiert werden. Dies ist jedoch begrenzt auf spezifische Umgebungen oder Geschwindigkeitsbereiche, außerhalb derer die Funktionen nicht aktivierbar sind. Die SAE teilt die verschiedenen technischen Systeme in sechs Levels auf, abhängig vom Grad der Automatisierung der Fahraufgabe (vgl. Tabelle 2.1). Die Levels können systematisch durch drei Kategorien definiert und unterschieden werden: Übernahme der Fahraufgabe, Rückfallebene und ODD.

Definition 2.1 (Operational Design Domain (ODD)) *Die ODD beschreibt die Betriebsbedingungen, unter denen eine bestimmte automatisierte Fahrfunktion oder eine bestimmte Teilfunktion davon funktionieren soll, einschließlich, aber nicht beschränkt auf umweltbedingte, geografische und tageszeitliche Einschränkungen und/oder das erforderliche Vorhandensein oder Nichtvorhandensein bestimmter Verkehrs- oder Fahrbahnmerkmalen. [100]*

Die Fahraufgabe bezieht sich hier auf die Ebenen Führung und Stabilisierung, also auf die unmittelbare Steuerung des Fahrzeugs. Sie sind in zwei Aspekte unterteilt: einerseits die Übernahme von longitudinaler und lateraler Fahrzeugkontrolle und andererseits die Beobachtung der Umgebung. Mit letzterem ist beispielsweise die

Wahrnehmung und Reaktion auf eine starke Bremsung des Vorderfahrzeugs oder einen plötzlich erscheinenden Fußgänger gemeint.

Das klassische Fahrzeug, in dem der Fahrer sowohl die laterale als auch die longitudinale Kontrolle ausübt, wird auf Level 0 eingeordnet. Dabei kann der Fahrer durch verschiedene Sicherheitssysteme wie ABS, ESP oder Notbremsassistenten unterstützt werden. Wenn der Fahrer die Längs- oder Querführung an ein System abgeben kann, ist das Fahren auf Level 1 gegeben.

Kann durch eine Kombination von Systemen die komplette Fahrzeugführung an das Fahrzeug umgeben werden, spricht man von Level 2. Ein Beispiel hierfür ist die Kombination eines Adaptiver Abstandsregeltempomat (ACC) und eines LKA. Auf Level 1 und 2 behält der Fahrer jedoch noch immer die Verantwortung und muss im Falle von unerwarteten Ereignissen eingreifen können, fungiert also als Rückfallebene für die Systeme. Das ändert sich mit Systemen auf Level 3. In Fahrsituationen innerhalb der ODD darf der Fahrer sich hier anderen Dingen zuwenden und muss nicht mehr durchgehend den Verkehr und die Systeme überwachen. Sobald die ODD verlassen wird, muss jedoch der Fahrer wieder für eine Übergabe der Steuerung bereitstehen. Dies erfordert jedoch eine rechtzeitige Benachrichtigung des Fahrers [80]. Wie viel Zeit zwischen Benachrichtigung und Übernahme liegen darf, ist Inhalt aktueller Forschungen und von der SAE nicht definiert [34].

Die nächste Steigerung ist das hochautomatisierte Fahren auf Level 4. Hier muss der Mensch nicht mehr als Rückfallebene fungieren, da das System selbstständig in der Lage sein muss, einen sicheren Zustand zu erreichen. Allerdings ist auch hier die ODD begrenzt, beispielsweise auf Autobahnen und Landstraßen. Diese Begrenzung wird auf Level 5 aufgehoben. Hier ist die ODD nicht limitiert, somit müsste ein solches System alle möglichen Situationen selbst und sicher lösen können.

Level	Name	Fahraufgabe		Rückfall- ebene	ODD
		Quer-/Längs- führung	Umgebungs- beobachtung		
0	Keine Automatisierung	Fahrer	Fahrer	Fahrer	nicht definiert
1	Assistenz- systeme	Fahrer & System	Fahrer	Fahrer	begrenzt
2	Teil- automatisierung	System	Fahrer	Fahrer	begrenzt
3	Bedingte Automatisierung	System	System	Fahrer	begrenzt
4	Hoch- automatisierung	System	System	System	begrenzt
5	Voll- automatisierung	System	System	System	unbegrenzt

Tabelle 2.1: Übersicht über die SAE-Levels aus [100].

2.1.2 Rechtliche Situation in Deutschland

Durch das Inkrafttreten des „Gesetzes zum autonomen Fahren“ (geregelt im Straßenverkehrsgesetz (StVG), §1d-§1l) wurde ein Rechtsrahmen zum Betrieb von Fahrzeugen auf Level 4 geschaffen. Dieses Gesetz zielt auf den Betrieb von automatisierten Fahrzeugen im kommerziellen und öffentlichen Kontext und ermöglicht das automatisierte Fahren in einem begrenzten Betriebsbereich, beispielsweise die Bedienung einer bestimmten Buslinie oder den Betrieb von Taxi-Diensten in geographisch umgrenzten Gebieten. Dazu muss für jedes automatisierte Fahrzeug eine technische Aufsicht verfügbar sein. Diese muss reagieren, wenn das Fahrzeug nicht in der Lage ist, eine Verkehrssituation selbst zu lösen. Gründe dafür können Fahrzeug-Fehlfunktionen, oder das Verlassen des Betriebsbereichs

sein. Mögliche Reaktionen darauf sind die Deaktivierung des Fahrzeugs aus der Ferne, oder die Freigabe von Manövern (z.B. Umfahrung von Hindernissen).

Definition 2.2 (Betriebsbereich) *Der Betriebsbereich „bezeichnet den örtlich und räumlich bestimmten öffentlichen Straßenraum, in dem ein Kraftfahrzeug mit autonomer Fahrfunktion [...] betrieben werden darf.“^a*

^a nach StVG §1d Abs. 2

Seit Juni 2022 wurde der Gesetzesinhalt in einer Verordnung ausdetailliert¹. Diese regelt im Wesentlichen die Erteilung einer Betriebserlaubnis in Kombination mit der Genehmigung eines festen, kartographisch umgrenzten Betriebsbereichs. Zur Zulassung einer automatisierten Fahrfunktion muss sichergestellt sein, dass „durch den Betrieb des Kraftfahrzeugs mit autonomer Fahrfunktion in diesem Betriebsbereich weder die Sicherheit und Leichtigkeit des Straßenverkehrs beeinträchtigt noch Leib und Leben von Personen über das allgemeine Risiko einer Beeinträchtigung durch den für den beantragten Betriebsbereich ortsüblichen Straßenverkehr hinaus erheblich gefährdet werden.“²

Zusätzlich sind hier detaillierte Regelungen zur Erprobung der Funktionen, Details zur notwendigen technischen Aufsicht sowie technische Anforderungen an Bau, Beschaffenheit und Ausrüstung der Fahrzeuge beschrieben. Des Weiteren schreibt die Verordnung vor, dass ein Sicherheitsnachweis auf Basis eines Katalogs von Testszenarien (vgl. Abschnitt 2.3.2) vorgelegt wird und das dieser Nachweis konform mit verschiedenen Standards ist, beispielsweise ISO 26262 (vgl. Abschnitt 2.2.2) oder ISO 21448 (vgl. Abschnitt 2.3.4).

¹ „Verordnung zur Regelung des Betriebs von Kraftfahrzeugen mit automatisierter und autonomer Fahrfunktion und zur Änderung straßenverkehrsrechtlicher Vorschriften“ vom 24. Juni 2022

² Verordnung zur Regelung des Betriebs von Kraftfahrzeugen mit automatisierter und autonomer Fahrfunktion und zur Änderung straßenverkehrsrechtlicher Vorschriften, §9 Abs.2 Nr. 3

2.2 Entwicklungsprozess und Testmethoden

2.2.1 Das V-Modell

Definition 2.3 (System) *Eine Menge von Komponenten oder Subsystemen, die mindestens einen Sensor, einen Regler und einen Aktor miteinander in Beziehung setzt. [54]*

Definition 2.4 (Komponente) *Element, dass auf logischer oder technischer Ebene separierbar ist und aus mehreren Hardware-Teilen oder Software-Modulen besteht. [54]*

Definition 2.5 (Software-Modul) *Atomares Level einer Software-Komponente einer Software-Architektur, das eigenständig getestet werden kann. [54]*

Die Entwicklung von Fahrzeugen und insbesondere die Entwicklung und Integration von Komponenten mit Software-Anteil im Fahrzeug ist ein aufwendiger Prozess. Eine wesentlicher Treiber des Aufwands ist die Tatsache, dass die entwickelte Software immer nur ein Teil des Gesamtsystems Fahrzeug ist, das aus mechanischen, elektrischen/elektronischen Komponenten und eben der Software besteht [128]. Zusätzlich werden diese Komponenten gleichzeitig und von verschiedenen Teams weltweit entwickelt.

Um die erfolgreiche Integration aller Komponenten in ein wie spezifiziert funktionierendes Gesamtsystem zu ermöglichen, werden definierte Prozessabläufe, Vorgehensweisen und Schnittstellen benötigt. Diese orientieren sich im Kontext der Fahrzeugentwicklung am V-Modell, welches als Referenzprozess fungiert [101]. Das V-Modell besteht aus einer Abfolge von Phasen, die in Form eines „V“ angeordnet sind (vgl. Abbildung 2.1). Die linke Seite des V zeigt die verschiedenen Entwicklungsphasen von der Anforderungserhebung auf Systemebene hin zur Implementierung einzelner Komponenten. Die rechte Seite stellt die verschiedenen Integrationsebenen der auf dieser Ebene entwickelten Komponenten und Systeme dar und endet mit der Überleitung in die Nutzung. Je „tiefer“ im V eine Phase angeordnet ist, desto niedriger der Abstraktionsgrad dieser Phasen. Dies spiegelt sich

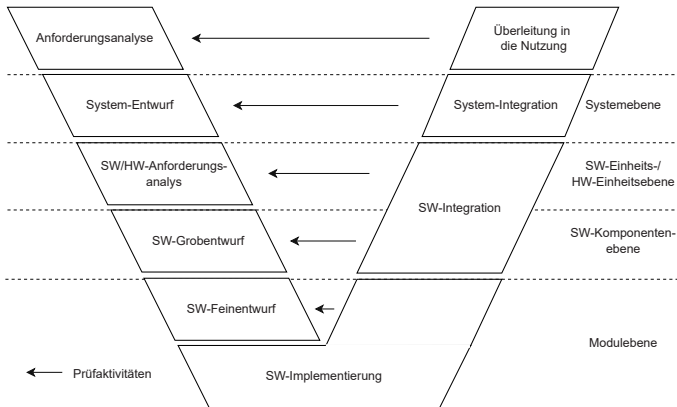


Abbildung 2.1: Das V-Modell, angelehnt an die Darstellung von [30].

in den Ebenenbezeichnungen: Modulebene, SW-Komponentenebene, SW-/HW-Einheitsebene und Systemebene. Nach jedem Schritt auf der rechten Seite des V werden die implementierten bzw. integrierten Komponenten und Systeme gegen die Anforderungen bzw. Entwürfe geprüft. Bei der Prüfung wird unterschieden zwischen Validierung und Verifikation. Eine Definition dieser Begriffe im Umfeld der Fahrzeugentwicklung liefert die ISO26262 [54]:

Definition 2.6 (Validierung) *Die Validierung ist ein Prozessschritt mit zwei Zielen. Zum einen muss der Nachweis erbracht werden, dass die Sicherheitsziele erfüllt und dass die abgeleiteten Sicherheitskonzepte geeignet für den Nachweis der funktionalen Sicherheit des Systems sind. Zum anderen muss der Nachweis erbracht werden, dass die Sicherheitsziele korrekt, vollständig und auf Gesamtsystemebene vollumfänglich erreicht sind.*

Definition 2.7 (Verifikation) *Die Verifikation ist ein Prozessschritt zur Bestimmung der Vollständigkeit und korrekten Spezifikation oder Implementierung der Anforderungen für eine Entwicklungsphase. Am Ende der Entwicklungsphase wird durch die Verifikation geprüft, ob die Ergebnisse den Anforderungen an diese Phase genügen.*

Die bei der Verifikation und Validierung adressierten Spezifikationen können verschiedener Natur sein und beispielsweise die Ausführungsdauer eines konkreten Teilschritts oder den Komfort bei der Nutzung einer Funktion beschreiben (stark abhängig von der aktuellen Komponenten- bzw. Systemebene). Im Kontext dieser Dissertation ist vor allem die Validierung auf Systemebene relevant, da nur auf dieser Ebene die Sicherheit eines Gesamtsystems beurteilt werden kann.

2.2.2 Funktionale Sicherheit

Zentral für die Verifikation und Validierung von elektrischen und elektronischen Systemen (E/E-Systeme) ist die funktionale Sicherheit. Mit dieser befasst sich der Standard ISO 26262 [54] mit dem Titel „*Road vehicles - Functional safety*“. Dabei steht die Sicherheit vor Fehlfunktionen von technischen Systemen auf Hardware-, Software- und Systemebene im Fokus. Die folgenden Definitionen sind aus der Norm entnommen [54] und ins Deutsche übersetzt.

Definition 2.8 (Funktionale Sicherheit) *Funktionale Sicherheit beschreibt die Abwesenheit von unzumutbarem Risiko einer Gefahr, die durch fehlerhaftes Verhalten eines E/E-Systems auftreten kann.*

E/E-Systeme können hier Komponenten oder auch gesamte Systeme sein, und ein fehlerhaftes Verhalten beschreibt das Versagen eines Systems in Bezug auf das angestrebte Systemdesign.

Definition 2.9 (Risiko) *Die Kombination der Wahrscheinlichkeit des Eintritts eines Schadens und der Schwere dieses Schadens.*

Definition 2.10 (Unzumutbares Risiko) *Risiko, das in einem bestimmten Kontext nach den geltenden gesellschaftlichen Moralvorstellungen als inakzeptabel eingestuft wird.*

Definition 2.11 (Systemversagen (engl. failure)) *Beendigung des erwünschten Verhaltens eines Systems aufgrund einer Störung.*

Definition 2.12 (Störung (engl. *fault*)) *Ein abnormaler Zustand, der das Versagen eines Systems verursacht.*

Ein Systemversagen kann durch einen Fehler im System ausgelöst werden.

Definition 2.13 (Fehler (engl. *error*)) *Abweichung zwischen einem berechneten, beobachteten oder gemessenen Wert bzw. einer Bedingung, und dem wahren, spezifizierten und theoretisch korrekten Wert bzw. der Bedingung.*

In der Entwicklung von Software unterscheidet man üblicherweise zwischen zwei verschiedenen Arten von Fehlern [103], den Spezifikationsfehlern, die zu fehlerhaftem Verhalten aufgrund von fehlerhaften Spezifikationen führen und den Implementierungsfehlern, die während der Umsetzung der Spezifikationen entstehen.

Automotive Safety Integrity Level (ASIL)

Zur Bewertung des notwendigen funktionalen Sicherheitslevels liefert die ISO 26262 ein Klassifikationssystem für Risiken von Systemen und Komponenten, die Automotive Safety Integrity Level (ASIL). Diese teilen sich in vier Levels (A-D) auf, die jeweils verschiedene Anforderungen, beispielsweise an Entwicklungsprozesse, Testing oder Ausfallraten von Hardware-Bauteilen, mit sich bringen. Die Stringenz der Anforderungen steigt dabei von A bis D. Heutige Software im Fahrzeug-Cockpit ist üblicherweise als ASIL B klassifiziert, während Fahrerassistenzsysteme und automatisierte Fahrfunktionen in ASIL D eingruppiert werden [78]. Ausgangspunkt der Systematik ist die Formulierung eines Sicherheitsziels.

Definition 2.14 (Sicherheitsziel (engl. *safety goal*)) *Eine Sicherheitsanforderung auf höchster Abstraktionsebene als Ergebnis einer Gefahren- und Risikoanalyse auf Systemebene.*

Definition 2.15 (Gefährdungereignis (engl. *hazardous event*)) *Der Eintritt eines Schadens aufgrund von einem fehlerhaften Verhalten eines Systems in einem Nutzungsszenario des Systems.*

Im Kontext der ISO 26262 wird Schaden (*engl. harm*) als körperliche Verletzung oder Gesundheitsbeeinträchtigung definiert. Um Sicherheitsziele in verschiedene ASIL einzustufen, muss eine Beurteilung der möglichen Schadensereignisse bei der Verletzung eines Sicherheitsziels (also beim Eintritt eines Gefährdungereignisses) in drei Kategorien aufgestellt werden:

- Schwere des potentiellen Schadens eines Gefährdungereignisses (*engl. severity*)
- Auftretenswahrscheinlichkeit der Fahrsituation, in der der Fehler auftreten kann (*engl. probability of exposure*)
- Beherrschbarkeit eines Gefährdungereignisses (*engl. controllability*)

Jede dieser Kategorien ist in verschiedene Klassen eingeteilt und nummeriert, wobei kleine Zahlen für ungefährliche Ereignisse („kein Schaden“, „extrem unwahrscheinlich“, „Generell kontrollierbar“) und hohe Zahlen für gefährliche Ereignisse („lebensgefährlicher Schaden“, „sehr wahrscheinlich“, „nicht kontrollierbar“). Aus der Kombination dieser verschiedenen Klassen wird eine Einstufung der ASIL vorgenommen (vgl. Tabellen in Anhang A.3).

2.2.3 Testwerkzeuge im Entwicklungsprozess

Die rechte Seite im V-Modell (siehe Abbildung 2.1)) befasst sich mit Testen, in der ISO 26262 definiert als:

Definition 2.16 (Testen (engl. *testing*)) *Der Prozess des Planens, Vorbereitens und Betriebens eines Systems, um zu verifizieren, dass es den Anforderungen genügt, um sicherheitsrelevante Anomalien zu detektieren, um zu*

validieren, dass die Anforderungen einem gegebenen Kontext genügen und um Vertrauen in sein Verhalten zu erzeugen.

Ein Test ist also ganz allgemein ein Teil des Entwicklungsprozesses und kann verschiedenen Zwecken dienen, Wolf [128] nennt hier den Nachweis von Fehlerwirkungen beim Ausführen der Systems, die Bestimmung der Qualität des Systems und die Erhöhung des Vertrauens in das System.

Definition 2.17 (System Under Test (SUT)) *SUT bezeichnet das zu testende System in einem Test. [102].*

Anforderungsbasiertes Testen

Das anforderungsbasierte Testen wird in der ISO26262 als geeignete Testmethode bei der Entwicklung von E/E-Systemen vorgeschlagen [54], sowohl auf Komponenten- als auch auf Systemebene. Testfälle werden dabei aus den Anforderungen an ein System oder eine Komponente generiert und in Testkatalogen gesammelt. In diesen Anforderungen sind die grundlegenden Funktionalitäten spezifiziert [101]. Das anforderungsbasierte Testen dient der Verifikation eines Systems, da durch eine Prüfung des Systemverhaltens getestet wird, ob die Anforderungen erfüllt wurden.

Definition 2.18 (Testfall) *Eine Menge von Eingangswerten, Ausführungsbedingungen und erwarteten Ergebnissen für eine bestimmte Zielvorgabe, mit der die Übereinstimmung mit einer spezifischen Anforderung überprüft werden kann [1].*

Simulation

Um Tests nicht nur mit einer voll entwickelten und integrierten Komponente oder einem fertigen Gesamtsystem ausführen zu können werden Simulationen genutzt. Dabei werden die relevanten Teile der Umwelt und die zur Systemausführung benötigten Teile des Fahrzeugs virtuell nachgebildet und erlauben somit eine

gezielte Ausführung von Systemen unter exakt steuerbaren Bedingungen. Im Vergleich zum Test eines Systems in der realen Welt hat die Simulation verschiedene Vorteile.

Entwicklungsbegleitendes Testen Das SUT kann in frühen Entwicklungsphasen unter Laborbedingungen getestet werden, was ein frühes Auffinden von Fehlern erlaubt.

Skalierbarkeit Tests können einfacher parallelisiert und beschleunigt werden.

Ungefährliches Testen Kritische Testszenarien können durchgeführt werden ohne die Gefahr der Beschädigung von Systemen.

Verschiedene Integrationsstufen Simulationen können sowohl für einzelne, in der Funktion beschränkte Komponenten als auch für gesamte Systeme und Wirkketten durchgeführt werden.

Zur konkreten Durchführung eines Tests in der Simulation können verschiedene Technologien verwendet werden. Zentral im Kontext dieser Arbeit ist der X-in-the-Loop-Gedanke (XiL) [3]. Dabei steht X für das konkrete SUT, das „in-the-Loop“ ausgeführt wird. Abhängig vom SUT sowie seinem Integrationslevel wird das für den Test notwendige Rest-Fahrzeug simuliert. Eine Kategorisierung verschiedener XiL-Stufen kann über folgende Parameter erfolgen: der Form des SUT sowie die Form der äußeren Logik des SUT [101]. In frühen Phasen des Systemdesigns wird ein Modell der zu entwickelnden Funktion getestet. Dies wird als *Model-in-the-Loop (MiL)* bezeichnet [19]. Dies geschieht ohne Hardware und vollständig auf einem Entwicklungssystem, beispielsweise mithilfe von Simulationsplattformen kommerzieller Anbieter. Die äußere Logik, also die Eingangs- und Ausgangsbedingungen werden vollständig durch Modelle abgebildet [128]. Dieses Modell wird im nächsten Schritt in Serienelement umgesetzt und kann mit der Methode *Software-in-the-Loop (SiL)* getestet werden. Dabei wird die zukünftige Hardware-Plattform sowie die Umgebung vollständig simuliert. Der entscheidende Unterschied zu MiL ist die Ausführung der konkreten Implementierung in Serienelement [128], nicht nur eines Modells der Funktion. Sobald die Funktion auf einem Steuergerät integriert wurde, kann sie mit einem *Hardware-in-the-Loop (HiL)* Aufbau getestet werden. Somit liegt das SUT in der finalen Form vor und nur

noch die äußere Logik wird durch Simulation abgebildet. Mit dem Komponenten-HiL, dem System-HiL oder dem Fahrzeug-HiL existieren verschiedene Varianten, die sich in der Integrationsstufe des SUT und der Menge der vorhandenen realen Steuergeräte unterscheiden [67].

Erprobung

Erprobung bezeichnet das „Ausprobieren“ des Systems in einer unbestimmten Situation [102]. Das Ziel von Erprobungen ist die Bestimmung des Reifegrads des Endprodukts und umfasst die korrekte Integration der Steuergeräte ins Fahrzeug, die korrekte Funktionalität des Gesamtsystems innerhalb eines realen Umfelds sowie die Erfüllung der Kundenerwartungen. Der entscheidende Vorteil der Erprobung ist der höchstmögliche Realitätsgrad, da sowohl die technische Umgebung als auch die äußeren Faktoren wie andere Verkehrsteilnehmer oder Wetter real vorhanden sind. Diese äußeren Faktoren können jedoch nicht beeinflusst werden, sodass die Erprobung vieler verschiedener Umweltbedingungen nur sehr aufwendig umgesetzt werden kann. Als Reaktion darauf werden beispielsweise Sommer- und Wintererprobungen [46] durchgeführt oder verschiedene Wetterbedingungen wie Regen gezielt angefahren. In Bezug auf die in einer Fahrt erlebten Verkehrssituationen enthält eine Erprobungsfahrt ein hohes Maß an Redundanz, da die erlebten Fahrsituationen überwiegend aus Standard-Fahrsituationen (z.B. freie Fahrt, Folgefahrt) bestehen. Durch ihre Seltenheit kommen spezielle oder kritische Verkehrssituationen nur selten im Rahmen einer Erprobung vor [4].

Für die Ausführung auch früherer Entwicklungsstände im realen Fahrzeug kommen üblicherweise Rapid Prototyping Systeme zum Einsatz. Rapid Prototyping umfasst dabei verschiedene Methoden zur Spezifikation und Ausführung von Software-Funktionen im realen Fahrzeug unter Nutzung von Experimentiersystemen [103]. Als Experimentiersysteme bezeichnet man Echtzeitrechensysteme mit weitaus höherer Rechenleistung als das spätere Zielsteuergerät. Dadurch ist man in der Lage, Code auch ohne Performance-Optimierungen in Echtzeit auszuführen. Durch die Real-Erprobung auf Experimentiersystemen findet eine Validierung

der Software-Spezifikation statt. Mit zunehmendem Entwicklungsfortschritt erfolgt der Übergang vom Experimentiersystem auf ein Seriensteuergerät.

2.3 Szenariobasierte Validierung

2.3.1 Motivation

Bei automatisierten Fahrfunktionen auf SAE-Level 3 und höher ergeben sich zwei Aspekte für die Verifikation und Validierung dieser Funktionen: Erstens steht beim Betrieb dieser Funktionen der Fahrer nicht mehr jederzeit als Rückfallebene zur Verfügung. Zweitens können die möglichen Betriebsituationen nicht auf definierte, testbare Anforderungen heruntergebrochen werden (vgl. Definition 2.19). Die erste Herausforderung stellt sich auch bei etablierten, sehr dynamischen Systemen wie dem Elektronisches Stabilitätsprogramm (ESP) [39]. ESP verfügt jedoch über einen beschreibbaren Raum möglicher Betriebsituationen. Somit ergibt sich die Herausforderung zur Validierung automatisierter Fahrfunktionen aus der Kombination beider Aspekte.

Ein Sicherheitsnachweis kann auf Tests in der Simulation sowie auf Erprobung in der realen Welt basieren (vgl. Abschnitt 2.2.3). Während die verschiedenen Stufen des simulativen Testens in der Entwicklung eine wichtige Rolle spielen, beruht der finale Sicherheitsnachweis von Assistenzsystemen bis einschließlich SAE-Level 2 auf Erprobungen mit realen Testfahrzeugen in der realen Welt [118]. Dieser Nachweis auf Basis der gefahrenen Distanz wird mit zunehmender Automatisierung immer aufwendiger. Als Beleg dafür liefert laut Wachenfeld et al. eine exemplarische Berechnung der zu fahrenden Kilometer, um ein geforderetes Sicherheitsniveau durch Sammlung von unfallfreien Kilometern nachzuweisen [118]. Anhand der Zahl der jährlichen Unfälle mit getöteten Personen und der jährlichen Fahrleistung (Bezugsjahr 2012) wird ein Vergleichswert, der das aktuelle Sicherheitsniveau bei menschlichen Fahrern beschreibt, zu

$$\frac{709 \cdot 10^9 \text{ km}}{3375 \text{ Unfälle}} \approx 210 \cdot 10^6 \frac{\text{km}}{\text{Unfall}} \quad (2.1)$$

bestimmt. Um in einer Erprobung mit ausreichender Konfidenz nachzuweisen, dass eine automatisierte Fahrfunktion mindestens so sicher ist wie der durchschnittliche menschliche Fahrer, muss diese Strecke sogar mehrfach durchfahren werden. Bei nur einmaliger Fahrt der Strecke wäre die Erprobung möglicherweise nur zufällig geglückt³. Die resultierende, zu fahrende Strecke ist daher abhängig vom geforderten Konfidenzlevel und der Leistung des automatisierten Fahrfunktion. Liegt diese weit über der menschlichen Leistung wird der Nachweis entsprechend einfacher. In [118] wird für die Verteilung von Unfällen eine Poisson-Verteilung angenommen und das geforderte Konfidenzniveau des Nachweises auf 95% gesetzt. Daraus ergibt sich für eine automatisierte Fahrfunktion eine notwendige Nachweis-Distanz von $2,1 \cdot 10^9$ km. Für den vollständigen Rechenweg wird auf [118] verwiesen. Auch mit massiver und kostenintensiver Parallelisierung von Realtests durch eine Vielzahl Testfahrzeuge und Testfahrer kann eine solche Menge an Testkilometern nicht wirtschaftlich eingefahren werden.

Alternativ könnte ein Sicherheitsnachweis durch das anforderungsbasierte Testen (vgl. Abschnitt 2.2.3) erfolgen. Dem steht entgegen, dass die ODD (vgl. Definition 2.1) einer automatisierten Fahrfunktion einen offenen Kontext (*engl. open context*) darstellt [93].

Definition 2.19 (Offener Kontext) *Offener Kontext bezeichnet die unendlich vielen Charakteristiken, möglichen Interaktionen und Effekte, die beim Betrieb eines technischen Systems in der unstrukturierten, realen Welt auftreten können. Diese können formal nicht vollständig beschrieben werden.*

Bei der Formulierung von klassischen Anforderungen werden spezifische, benötigte Fähigkeiten des Systems in natürlicher Sprache definiert und für jede Anforderung ein spezifischer Testfall abgeleitet. Der offene Kontext kann jedoch nicht vollständig beschrieben werden, daher kann auch kein vollständiger Satz an Anforderungen formuliert werden. Für den Sicherheitsnachweis automatisierter

³ Beispiel zur Veranschaulichung: wenn durch Versuch bestimmt werden soll, ob eine Münze beim Münzwurf „fair“ ist, also „Kopf“ und „Zahl“ gleiche Wahrscheinlichkeiten liefern, genügt es nicht, die Münze nur zweimal zu werfen. Auch wenn bei zwei Würfeln jeweils einmal „Kopf“ und einmal „Zahl“ vorliegt, ist die Konfidenz in den Nachweis der Fairness sehr klein.

Fahrzeuge muss demnach ein neuer Ansatz gefunden werden. Dazu wurden verschiedene Ansätze vorgeschlagen und diskutiert (z.B. in [110] oder [118]). Ein vielversprechender Ansatz zur Beschreibung vieler Testfälle ist das szenariobasierte Testen, das sich mittlerweile in Standardisierung (vgl. Abschnitt 2.3.2) und Gesetzgebung (vgl. Abschnitt 2.1.2) durchgesetzt hat. Dieses erlaubt einerseits die automatisierte Ableitung von Testfällen durch Definition von Parameterbereichen und ist andererseits in der Lage, die natürliche Varianz im Straßenverkehr durch Variation dieser Parameter abzubilden. Die Repräsentation des offenen Kontexts ist auch für das szenariobasierte Testen eine Herausforderung und wird in dieser Dissertation adressiert.

2.3.2 Grundbegriffe des szenariobasierten Testens

Ein Grund für die hohe Distanz zur Validierung (vgl. Abschnitt 2.3.1) liegt in der Redundanz im Straßenverkehr. Redundanz bedeutet hier, dass über eine Erprobungsfahrt hinweg die Mehrzahl der auftretenden Situationen keinen Beitrag zur Absicherung leisten. Ein Beispiel der Absicherung einer Spurwechselfunktion zeigt, dass in 24 Stunden Versuchsfahrt nur insgesamt 40 Minuten für einen Test relevante Situationen (hier: Spurwechsel) enthalten sind [32]. Über 97% der Testfahrt bestehen somit aus redundanten Folge- und Freifahrten. Der Ansatz des szenariobasierten Testens zur Validierung automatisierter Fahrzeuge versucht, alle relevanten Aspekte und Situationen, die im Straßenverkehr auftreten können, als Szenarien zu repräsentieren und simulativ zu testen. Auf diesem Weg kann die dem Straßenverkehr inhärente Redundanz verringert werden [60]. Das grundsätzliche Vorgehen besteht in der Sammlung von relevanten Szenarien, die in einem Szenariokatalog aggregiert werden. Zur Validierung einer automatisierten Fahrfunktion muss jedes im Katalog enthaltene Szenario erfolgreich getestet werden. Im nationalen Umfeld befassen sich vor allem zwei große Kooperationsprojekte mit szenariobasierten Testen für den Sicherheitsnachweis automatisierter Fahrzeuge. Das 2019 abgeschlossene Projekt PEGASUS hatte dabei den Fokus auf einer Level 3-Funktion für Autobahnen, den sogenannten „Autobahn-Chauffeur“.

Das Nachfolgeprojekt VV Methoden weitet die Betrachtung auf den urbanen Einsatz einer Level 4 oder Level 5 Funktion aus. Viele der folgenden Definitionen und Begriffe wurden im Kontext dieser Förderprojekte entwickelt.

Begriffsdefinitionen

Für die folgenden Begriffe werden die Definitionen aus den Standards ISO 21448⁴ [55] und ISO 34501⁵ [56] verwendet.

Definition 2.20 (Szenario) *Beschreibung der zeitlichen Entwicklung zwischen verschiedenen Szenen in einer Sequenz von Szenen. mit Zielen und Werten innerhalb einer vorgegebenen Situation, beeinflusst durch Aktionen und Events [55].*

Definition 2.21 (Szene) *Momentaufnahme der Umgebung, einschließlich der Szenerie, der dynamischen Elemente sowie der Repräsentationen aller Teilnehmer und Beobachter sowie den wechselseitigen Verhältnissen zwischen ihnen [55].*

Jedes Szenario startet mit einer initialen Szene. Im Gegensatz zur Szene erstreckt sich ein Szenario über eine Zeitspanne.

Definition 2.22 (Szenerie) *Der Teil der Umgebung, der während eines Szenarios unverändert bleibt [56].*

Geyer et al. erklären die Szenerie anhand der Analogie zum Film oder Theater [43]. Demnach ist die Szenerie der „Rahmen, in dem sich eine Handlung abspielt“⁶, beispielsweise der Bühnenhintergrund. Somit ist die Szenerie ein Teil der sich darin abspielenden Szene.

⁴ Road vehicles — Safety of the intended functionality

⁵ Road vehicles — Terms and definitions of test scenarios for automated driving systems

⁶ „Szenerie“ auf Duden online, <https://www.duden.de/rechtschreibung/Szenerie>, abgerufen 11.11.2022

Definition 2.23 (Dynamisches Element) *Element in einem Szenario, das einer Zustandsveränderung unterliegt [56].*

Definition 2.24 (Aktion) *Einzelnes Verhalten eines Teilnehmers einer Szene [55].*

Der Begriff des Teilnehmers ist nicht formal definiert, im Kontext dieser Dissertation wird die Definition von Bach et al. verwendet [5], die mit der ISO 21448 konform geht (siehe [55], *Note 2 to entry 3.2*). Teilnehmer sind eine Teilmenge der dynamischen Elemente.

Definition 2.25 (Teilnehmer) *Der Begriff Teilnehmer umfasst alle dynamischen Elemente innerhalb eines Szenarios, die miteinander interagieren können. Das umfasst alle von Menschen oder automatisierten Systemen gesteuerten Fahrzeuge und Vehikel sowie Fußgänger.*

Ein besonderer Teilnehmer ist das Ego-Fahrzeug:

Definition 2.26 (Ego-Fahrzeug) *Das Fahrzeug bzw. die automatisierte Fahrfunktion, die im Zentrum eines Szenarios steht und darin als SUT fungiert [56].*

Aktionen können nur von Teilnehmern durchgeführt werden und besitzen eine Zeitdauer, während Events zu einem Zeitpunkt definiert sind, und durch äußere Einflüsse oder Zufälle auftreten können. Beispiele für Events sind die Richtungsänderung eines Fußgängers oder ein auf die Straße fallender Baum.

Definition 2.27 (Event) *Auftreten eines Ereignisses zu einem Zeitpunkt [55].*

Um die Bewegung sowohl des Ego-Fahrzeugs als auch der anderen Teilnehmer abstrakt zu beschreiben, wird der Begriff des Manövers eingeführt.

Definition 2.28 (Manöver) *Eine Serie von Aktionen, um die Position eines Teilnehmers zu verändern oder beizubehalten [56].*

Da ein Szenario üblicherweise dazu genutzt wird, um ein spezifisches Fahrzeug bzw. eine spezifische Fahrfunktion zu testen (siehe Definition 2.26), wird mit der Situation noch die Perspektive des Ego-Fahrzeugs definiert:

Definition 2.29 (Situation) *Eine Szene aus der Perspektive eines spezifischen Teilnehmers [56].*

Die definierten Begriffe lassen sich anhand eines Beispielszenarios mit kreuzendem Fußgänger veranschaulichen (vgl. Abbildung 2.2).

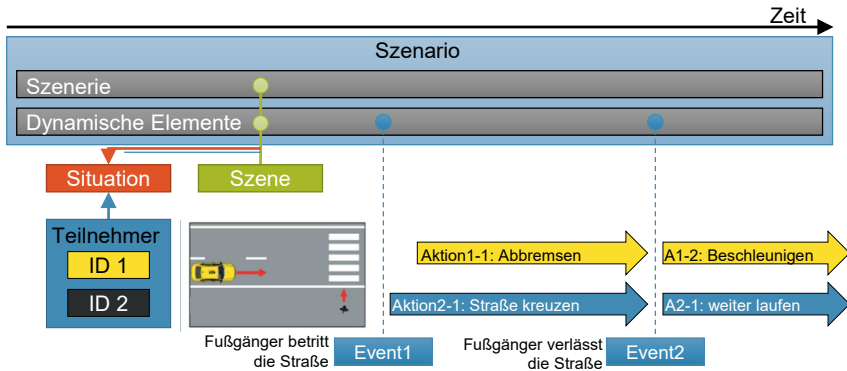


Abbildung 2.2: Darstellung des Beispielszenarios „kreuzender Fußgänger“ zur Einordnung der definierten Begriffe.

Schichten-Modell für die Szenario-Beschreibung

Bagschick et al. definieren die Inhalte eines Szenarios über ein mehrschichtiges Modell [6], um die für ein Szenario relevanten Aspekte strukturieren zu können. Eine weitere Schicht wurde von Bock et al. ergänzt [13].

Schicht 1: Straßentopologie Die erste Schicht beschreibt die Straßenebene und schließt die Topologie der Umgebung, das Straßennetz und die Straßenmarkierungen mit ein.

Schicht 2: Verkehrsinfrastruktur In der zweiten Schicht wird die Verkehrsinfrastruktur aufgeführt. Sie umfasst Leitplanken, Ampeln und Schilder und beschreibt somit die statischen Elemente, die einen Rahmen für den Verkehr setzen.

Schicht 3: Zeitlich begrenzte Veränderungen In Schicht drei werden zeitlich begrenzte Veränderungen der ersten und zweiten Schicht beschrieben, die die dort definierten Elemente überlagern. Ein Beispiel hierfür sind Baustellen, in denen zusätzliche Linien, Schilder und ggf. Ampeln die herkömmliche Verkehrsführung überlagern und temporär gültig sind.

Schicht 4: Dynamische Objekte In der vierten Schicht sind alle dynamischen Objekte beschrieben. Das umfasst im Wesentlichen die Beschreibung anderer Verkehrsteilnehmer und deren Aktionen sowie deren Manöver. Objekte können in verschiedene Klassen eingeteilt werden, beispielsweise Fahrzeuge, Fußgänger oder Fahrradfahrer. Die Beschreibung ihrer Interaktionen erfolgen beispielsweise über Manöver oder Trajektorien.

Schicht 5: Umweltbedingungen In der fünften Schicht sind Umweltbedingungen wie Wetter oder Beleuchtung gegeben. Hier werden also beispielsweise Behinderungen der Sicht durch Regen oder tief stehende Sonne definiert.

Schicht 6: Digitale Informationen Die sechste Schicht deckt die gesamte digitale Kommunikation und Informationsebene ab, um zusätzliches Wissen über Teilnehmer oder Umgebung zu modellieren.

Die verschiedenen Schichten beschreiben die relevanten Aspekte eines Szenarios. Zwischen diesen Aspekten bestehen Abhängigkeiten. Beispielsweise hat die Anwesenheit von starkem Nebel großen Einfluss auf das Verhalten und die Interaktionen von Objekten. Genauso sind die möglichen Bewegungen von Teilnehmern von den Straßen- und Verkehrsbedingungen abhängig.

Das Schichtenmodell wurde im Rahmen des PEGASUS Projekts eingeführt und somit mit Fokus auf Autobahn-Szenarien definiert. Die Adaption auf urbane Szenarien wird von Scholtes et al. [104] vorgenommen, wobei einige Details in der ersten und zweiten Schicht anders sortiert werden. Im Kontext dieser Dissertation liegt jedoch die Interaktion mit anderen Verkehrsteilnehmern im Fokus, die in [6, 13, 104] einheitlich in der vierten Schicht eingeordnet werden.

Abstraktionsstufen der Szenario-Beschreibung

Um die Verwendung von Szenarien im Prozess der Entwicklung und Validierung automatisierter Fahrfunktionen zu ermöglichen, sind verschiedene Abstraktionsstufen nötig. Im Rahmen des Pegasus-Projekts wurden dazu die Abstraktionsstufen der funktionalen, logischen und konkreten Szenarien definiert [7]. In der ISO 34501 wurde dieses Modell um die Stufe der abstrakten Szenarien erweitert [56]. Diese ordnet sich zwischen den funktionalen und den logischen Szenarien ein. In Abbildung A.3 sind die Stufen anhand des Beispielszenarios „kreuzender Fußgänger“ dargestellt, wobei die verschiedenen Attribute anhand des Schichten-Modells (vgl. Abschnitt 2.3.2) geordnet sind.

Funktionale Szenarien stellen eine Beschreibung dar, die auf genaue Parametrisierung verzichtet. Die Beschreibung kann dabei aus anschaulichen Visualisierungen oder aus einer umgangssprachlichen Formulierung bestehen. Dabei können verschiedene Aspekte, abhängig von ihrer Relevanz, in unterschiedlichem Detailgrad beschrieben werden.

Die abstrakten Szenarien bilden eine formale und maschinen-lesbare Beschreibung des Szenarios. Die Beschreibung ist mit einer Ontologie bzw. einer Familie von Ontologien verknüpft. Die Ontologie definiert mögliche Elemente und Ausprägungen eines Szenarios und lässt so eine präzise, formale Beschreibung zu. Damit ist ein Szenario formaler beschrieben als durch die natürlichsprachliche Beschreibung des funktionalen Szenarios.

Definition 2.30 (Ontologie) *Eine Ontologie ist eine formale, explizite Spezifikation einer gemeinsamen, abstrakten und vereinfachten Sicht auf einen Teil der Welt, der zur Erfüllung eines Zwecks repräsentiert werden soll [111] [42].*

Logische Szenarien beschreiben die nächste Abstraktionsstufe. Hier werden die verschiedenen Eigenschaften eines Szenarios mit Parameterbereichen beschrieben. Jedem funktionalen Szenario ist mindestens ein logisches Szenario zugeordnet. Parameterbereiche können im einfachsten Fall durch einen Minimal- und

Maximalwert angegeben werden, nach Bedarf aber auch durch statistische Verteilungen ausgedrückt werden. Zusätzlich sind Verknüpfungen zwischen Parametern möglich, sodass bedingte Parameterbereiche formuliert werden können. Durch ein logisches Szenario werden sinnvolle Variationen der Parameter eines Szenarios möglich.

Definition 2.31 (Parametervariation) *Die Auswahl von konkreten Werten für jeden in einem logischen Szenario definierten Parameter zur Definition von konkreten Szenarios wird Parametervariation genannt. [96]*

Eine exakte Parametrisierung eines logischen Szenarios wird konkretes Szenario genannt. Hier liegt für jeden Parameter ein exakter Wert vor, sodass ein konkretes Szenario in einen Testfall überführt werden kann. Aus einem logischen Szenario können mehrere konkrete Szenarien generiert werden, wobei die Menge der möglichen konkreten Szenarien durch die Anzahl der Parameter sowie der Größe der Abtastschritte definiert ist. Konkrete Szenarien können direkt in verschiedenen Simulationsumgebungen ausgeführt werden. Für die technische Repräsentation eines konkreten Szenarios existieren verschiedene, vom konkreten Simulationstool abhängige Ansätze. Mit OPENScenario⁷ wird ein Industriestandard entwickelt, der perspektivisch die Definition konkreter Szenarien toolunabhängig erlaubt.

Modellbasierte Beschreibung von Szenarien

Eine gemeinsame Beschreibung für die Aspekte der verschiedenen Szenarioschichten liefern modellbasierte Ansätze zur Szenario-Beschreibung [5, 114]. In der Modellierung sind einerseits die statischen Aspekte (vgl. „Szenerie“ in Bild) und andererseits die dynamischen Aspekte über die Teilnehmer (vgl. Definition 2.25) des Szenarios repräsentiert (vgl. Abbildung 2.3). Üblicherweise wird das Verhalten der Teilnehmer durch ihre Manöver beschrieben. Für jeden Typ von Verkehrsteilnehmern wird eine eigene Manöverliste mit allen validen Manövern dieses Typs definiert. Beispielsweise ist das Manöver „per Zebrastreifen die

⁷ <https://www.asam.net/standards/detail/openscenario/>

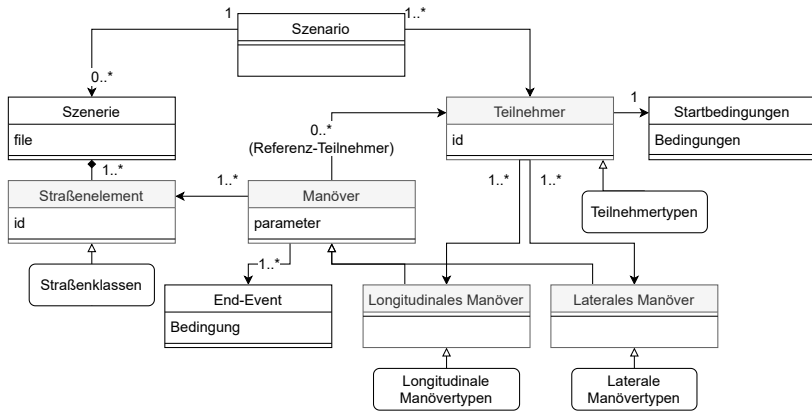


Abbildung 2.3: Formales Modell als Ansatz zur Beschreibung von Szenarien nach [16].

Straße überqueren“ für Fahrzeuge nicht durchführbar, während das Manöver „Ein-sichern“ für Fußgänger keinen Sinn ergibt. Die beschriebene Modellierung von den Szenarien hat folgende Vorteile: da sie eine Abstraktion der verschiedenen Aspekte bereitstellt, vereinfacht sie das menschliche Verständnis für verschiedene Typen von Szenarien und erlaubt eine strukturierte Definition von Szenarien. Des Weiteren können von derartigen Modellierungen maschinenlesbare Dateiformate abgeleitet werden, die dann von Simulationswerkzeugen interpretiert werden können [16]. Ein Beispiel für ein Dateiformat ist OpenSCENARIO⁸.

2.3.3 Prozessschritte in der szenariobasierten Validierung

Das konkrete Vorgehen für eine szenariobasierte Validierung automatisierter Fahr-funktionen wird breit diskutiert [62,85,96]. Aufgrund ihrer Übersichtlichkeit wird in dieser Dissertation die Darstellung von [85] verwendet (vgl. Abbildung 2.4).

⁸ <https://www.asam.net/standards/detail/openscenario/>

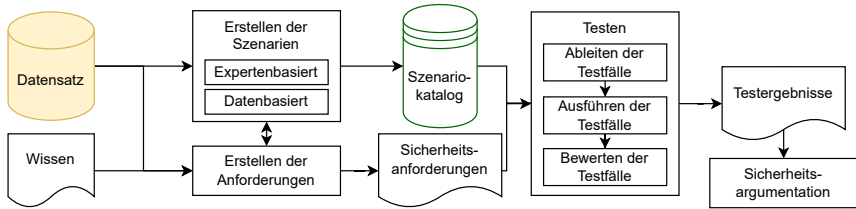


Abbildung 2.4: Visualisierung der relevanten Schritte und der vorhandenen Artefakte beim szenariobasierten Testen, angelehnt an [85].

Erstellen der Szenarien

Der Schritt „Erstellen der Szenarien“ hat zum Ziel, einen möglichst vollständigen Katalog relevanter Szenarien für eine Funktion zu generieren [95]. Im Katalog liegen logische Szenarien vor.

Ein trivialer Ansatz für die Ermittlung der relevanten Szenarien ist die Betrachtung der Kombinatorik von unterschiedlichen Aspekten. Am Beispiel möglicher Kreuzungsszenarien wird dieser Ansatz verdeutlicht: An einer gegebenen, vierarmigen Kreuzung kann ein Ego-Fahrzeug geradeaus fahren, rechts oder links abbiegen (3 Möglichkeiten). Aus jeder Richtung kann ein oder mehrere Verkehrsteilnehmer kommen, die analog zum Ego-Fahrzeug auf drei Arten die Kreuzung überqueren können. Diese Verkehrsteilnehmer können vom Typ Auto, Truck, Fahrradfahrer oder Fußgänger sein (4 Möglichkeiten). Für jeden Verkehrsteilnehmer werden unterschiedliche Geschwindigkeiten berücksichtigt (z.B. hier 10 unterschiedliche Geschwindigkeiten). Für jeden Verkehrsteilnehmer existieren somit $4 (\text{Typ}) \times 10 (\text{Geschwindigkeit}) = 40$ unterschiedliche Möglichkeiten. Beschränkt man die Zahl der Verkehrsteilnehmer auf maximal zwei pro Kreuzungsarm ergeben sich pro Kreuzungsarm bereits 14521 Möglichkeiten⁹. Für drei Kreuzungsarme sowie unterschiedliche Wege des Ego-Fahrzeugs sind somit

⁹ Kein Verkehrsteilnehmer (1) + ein Verkehrsteilnehmer mit 3 Möglichkeiten für die Fahrtrichtung ($3 \cdot 40$) + 2 Verkehrsteilnehmer mit insgesamt 9 Kombinationsmöglichkeiten für die Fahrtrichtungen ($40 \cdot 40 \cdot 9$) = 14521

$14521^3 \times 3 = 9,19 \times 10^{12}$ Möglichkeiten gegeben. Diese Zahl kommt alleine durch die Variation der Verkehrsteilnehmer auf einer spezifischen Kreuzung zustande, wobei die Reihenfolge der Teilnehmer noch nicht berücksichtigt ist. Nimmt man die weiteren Schichten des Schichten-Modells (vgl. Abschnitt 2.3.2) hinzu, beispielsweise Variationen der vorhandenen Verkehrsinfrastruktur, steigt die Zahl potenzieller Szenarien um weitere Größenordnungen an.

Um nicht die ganze Kombinatorik des theoretisch Möglichen zu testen existieren Ansätze zur Definition der relevanten Szenarien. Diese stützen sich entweder auf Wissen von Funktions- und Domänenexperten oder auf die Analyse von Verkehrsdatensätzen.

Ein Startpunkt zur expertenbasierten Erstellung von Szenarien sind gesetzliche Vorgaben für relevante, zu testende Szenarien. Hier liefert beispielsweise das europaweit gültige Regelwerk der UN/ECE¹⁰ einen Katalog für die Validierung von automatisierten Lenksystemen [40]. Des Weiteren können Quellen wie Unfalldatenbanken analysiert werden, um typische kritische Szenarien zu erhalten [119]. Darüber hinaus existieren verschiedene systematische Ansätze: ausgehend von den durch Experten identifizierten und gewichteten Einflussfaktoren einer Funktion werden Szenarien abgeleitet, die aus Kombinationen der Einflussfaktoren bestehen [129]. Eine weitergehende Systematik erlaubt die Nutzung von Ontologien, welche beispielsweise Verkehrsregeln, funktionale Beschreibungen und Expertenwissen in einer Wissensbasis kombinieren [6].

Ein rein expertenbasiert definierter Szenarien-Katalog hat den Nachteil, dass keine Repräsentativität für eine gegebene ODD nachgewiesen werden kann [28]. Außerdem unterliegt das Urteil von Experten potentiell kognitiven Verzerrungen. „Kognitive Verzerrung“ ist der Oberbegriff für systematisch auftretende Denk- und Wahrnehmungsfehler, beispielsweise in Bezug auf Wahrscheinlichkeiten von Ereignissen. Diese sind auch bei Experten in ihren jeweiligen Domänen nachgewiesen [113]. Angewandt auf die Erstellung von Szenarien können sich diese

¹⁰ United Nations/Economic Commission for Europe

Verzerrungen beispielsweise darin äußern, dass Experten gewisse Szenarien ignorieren, die ihnen zufällig nie selbst begegnet sind.

Als Lösung für diese Problematik dient die datenbasierte Szenarioerstellung. Diese fußt auf der Annahme, dass in umfangreichen Verkehrsaufzeichnungen (z.B. von Verkehrsüberwachungskameras) alle relevanten Szenarien vorhanden sind. Mithilfe einer datenbasierten Szenarioanalyse sollen diese Verkehrsaufzeichnungen untersucht werden und eine für den gesamten Datensatz repräsentative Teilmenge für den Test extrahiert werden (vgl. Abschnitt 3).

Ein dritter Weg zur Erstellung von relevanten Szenarien ist die Simulation von Verkehr. Der Ansatz ist hier, dass mittels Simulation realistisch modellierter Verkehrsteilnehmer realitätsnahe Szenarien erzeugt werden. Analog zur datenbasierten Szenarioerstellung werden die simulierten Szenarien für den Test extrahiert. [74] Der Vorteil dieses Ansatzes liegt in der einfach skalierbaren Erzeugung der Daten, da kein realer Verkehr und keine Sensorik benötigt wird. Dem gegenüber steht jedoch die Schwierigkeit, einen ausreichenden Realismus der verwendeten Modelle nachzuweisen.

Unabhängig von der Methode zur Definition der Szenarien ist Resultat der Szenarioerstellung ein Szenariokatalog, in dem die Szenarien aus den beschriebenen Quellen strukturiert abgelegt werden. Das Pegasus-Projekt beschreibt den Szenariokatalog als eine Datenbank mit abgelegten logischen Szenarien (vgl. Abschnitt 2.3.2), aus denen dann konkrete Testfälle abgeleitet werden können (vgl. Abschnitt 2.3.3). Dabei sind die logischen Szenarien verknüpft mit der Datenquelle, aus der sie gewonnen wurden [95]. Diesem Ansatz folgt das Projekt *Safety Pool Scenario Database*¹¹: Hier werden logische Szenarien in einer öffentlich nutzbaren Datenbank abgelegt. Die logischen Szenarien werden hier mit einer modellbasierten Sprache beschrieben [136].

¹¹ <https://www.safetypool.ai/database>, abgerufen am 19.05.2023

Erstellen der Anforderungen

Um anhand der im Szenariokatalog vorliegenden Beschreibungen eine Aussage über die Sicherheit der automatisierten Fahrfunktion treffen zu können, müssen diese mit Anforderungen verknüpft werden. Diese Anforderungen werden im nächsten Prozessschritt „Testen“ als Bestehenskriterien für Szenarien verwendet, womit die Szenarien in Testfälle umgewandelt werden (vgl. Definition 2.18). Im Beispielszenario „kreuzender Fußgänger“ sind mögliche Anforderungen die maximal erlaubte Verzögerung des Ego-Fahrzeugs oder der minimale erlaubte Abstand zum Fußgänger. Weitere, offensichtliche Anforderungen betreffen die Kollisionsfreiheit oder die Vermeidung von unbeabsichtigtem Spurverlassen. Bei der Simulation der Szenarien wird die Erfüllung dieser Anforderungen kontinuierlich überprüft und erlaubt damit eine detaillierte Bewertung der automatisierten Fahrfunktion.

Testen

Das Schritt „Testen“ gliedert sich in drei Teilschritte auf: Ableitung, Ausführung und Evaluation der Testfälle. In der Ableitung der Testfälle werden konkrete Parameterwerte aus den logischen Szenarien gezogen. Um aus einem logischen Szenario möglichst relevante (z.B. kritische) konkrete Szenarien zu generieren können verschiedene Strategien bei der Parameter-Auswahl verwendet werden [10, 12, 27].

In einem zweiten Schritt werden diese konkreten Szenarien ausgeführt. Dies wird zu großen Teilen in verschiedenen Simulationsumgebungen durchgeführt (vgl. Abschnitt 2.2.3), da diese parallelisiert und skaliert werden können. Beispiele für gängige Simulationswerkzeuge sind *IPG Carmaker*, *Virtual Test Drive* oder *Carla*. Die simulierten Szenarien werden anhand der Anforderungen (vgl. Abschnitt 2.3.3) bewertet. Aus einer Aggregation dieser Bewertung kann dann auf den Reifegrad der automatisierten Fahrfunktion geschlossen werden.

Sicherheitsargumentation

Die Sicherheitsargumentation im szenariobasierten Testen basiert auf dem Ansatz, dass eine ausreichend aussagekräftige Anzahl an Szenarien getestet wurde, um ein unzumutbares Risiko beim Betrieb der automatisierten Fahrfunktion auszuschließen [81]. Die Tests werden anhand von konkreten Szenarien durchgeführt. Die Ergebnisse der konkreten Szenarien lassen sich zu ihren zugehörigen logischen Szenarien aggregieren (siehe Abschnitt 2.3.2). Zentral ist dabei die Annahme, dass die relevanten Größen im Szenario-Kontext bei kleinen Änderungen der Parameter keine großen Sprünge aufweisen [85]. Zum Beispiel sollte die Änderung der Geschwindigkeit des Ego-Fahrzeugs um wenige Prozent nur sehr geringen Einfluss auf die geplante Trajektorie eines Fahrzeugs haben.

Das Ziel der Sicherheitsargumentation ist der Nachweis, dass kein unzumutbares Risiko beim Betrieb des automatisierten Fahrfunktion vorliegt. Inwiefern Szenarien mit Risiken verknüpft sind, wird in der ISO 21448 beschrieben (vgl. Abschnitt 2.3.4).

2.3.4 Sicherheit der Sollfunktion

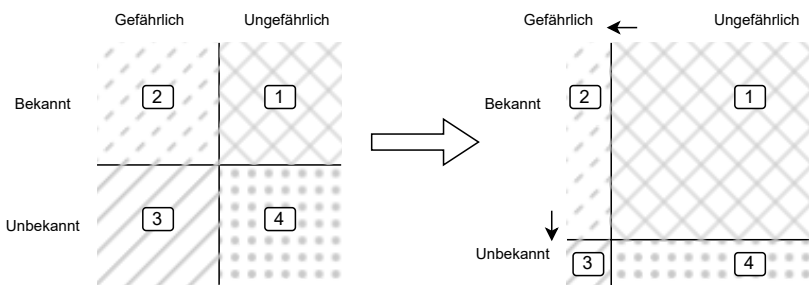


Abbildung 2.5: Der Szenarienraum wird eingeteilt in die Kategorien bekannt/unbekannt sowie gefährlich/ungefährlich, angelehnt an [55].

Der Standard ISO 21448 befasst sich mit dem Nachweis der ausreichenden Sicherheit für automatisierte Fahrzeuge. Dabei fokussiert er auf die „Sicherheit der Sollfunktion“ und meint damit die Sicherheit beim Auftreten von externen Einflüssen, die nicht spezifiziert oder spezifizierbar sind (vgl. Definition 2.19). Zur Strukturierung des Szenarienraums führt der Standard die Aufteilung der Szenarien in vier Kategorien ein (vgl. Abbildung 2.5):

Kategorie 1 bekannte, gefährliche Szenarien

Kategorie 2 bekannte, ungefährliche Szenarien

Kategorie 3 unbekannte, gefährliche Szenarien

Kategorie 4 unbekannte, ungefährliche Szenarien

Das Ziel der Validierung der automatisierten Fahrfunktion muss die Reduktion der gefährlichen Szenarien (Kategorie 2 und 3) sein, während Kategorie 1 maximiert werden muss. Die Szenarien aus Kategorie 4 müssen nicht weiter betrachtet werden, da sie per Definition ungefährlich sind. Sie dienen nur zur Vervollständigung der Darstellung des Szenarienraums. [55]

Bekanntheit von Szenarien

Das Ziel der in dieser Dissertation vorgestellten Methode ist die Vergrößerung der Menge der bekannten Szenarien. Diese Vergrößerung geht mit einer Verschiebung von Szenarien aus Kategorie 3 und 4 in Kategorie 1 und 2 einher. Wie in Prozessschritt „Erstellen der Szenarien“ (vgl. Abschnitt 2.3.3) beschrieben, müssen hier experten- und datenbasierte Ansätze zur Vergrößerung des bekannten Szenarienraums angewendet werden.

Reduktion der Gefährlichkeit

Die Reduktion der Gefährlichkeit liegt nicht im Fokus dieser Dissertation. Die ISO 21448 nennt verschiedene Maßnahmen zur Reduktion der Gefährlichkeit von Szenarien, also zur Verschiebung von Szenarien aus Kategorie 2 und 3 in Kategorie 1

und 4. Die Maßnahmen zielen darauf ab, gefährliche Szenarien entweder ganz zu vermeiden, oder die Auswirkungen abzumildern. Falls während der Funktionsentwicklung beispielsweise erkannt wird, dass die entwickelte Funktion Schwächen bei Nacht hat, können verschiedene Maßnahmen ergriffen werden. Beispielsweise könnte der Betrieb der Funktion bei Nacht untersagt und technisch unmöglich gemacht werden (Vermeiden). Alternativ könnte eine verbesserte Perzeption, beispielsweise durch zusätzliche Sensorik, zu einem Ausgleich der Schwächen führen (Abmildern). [55]

Risikobetrachtung

Die ISO 21448 beschreibt die Aspekte, die bei der Ermittlung des Risikos eine Rolle spielen (vgl. Abbildung 2.6). Als Ausgangspunkt dienen hier die Szenarien, welche potentielle Auslöser für ein gefährliches Verhalten der Funktion sind und somit eine Gefährdung hervorrufen können. Wenn innerhalb des Szenarios Umstände gegeben sind, die zu Schaden führen können (siehe „Auftrittswahrscheinlichkeit“, Abschnitt 2.2.2) entsteht ein gefährliches Event. Dieses wird verknüpft mit der Möglichkeit zur Reaktion von anderen involvierten Personen (siehe „Beherrschbarkeit“, Abschnitt 2.2.2). Wenn diese nicht gegeben ist entsteht ein Schaden mit einer gewissen Schadensschwere (vgl. Abschnitt 2.2.2). Aus der Beurteilung dieser Aspekte kann dann ein Risiko bestimmt werden.

Die Aspekte Beherrschbarkeit und Schadensschwere sind bereits aus der funktionalen Sicherheit bekannt und somit Teil eines etablierten Prozesses in der Automobilindustrie (siehe Abschnitt 2.2.2). Die ISO 21448 verweist für die Bestimmung der Beherrschbarkeit und Schadensschwere auf Methoden zur Systemanalyse aus dem Standard für die funktionale Sicherheit ISO 26262. Auch die Auftrittswahrscheinlichkeit von Szenarien ist aus diesem Standard bekannt, für eine Betrachtung der Sicherheit der Sollfunktion sind die Methoden der funktionalen Sicherheit jedoch nicht ausreichend. Die funktionale Sicherheit befasst sich ausschließlich mit Fehlern und Ausfallszenarien von E/E-Komponenten, wohingegen für die Sicherheit der Sollfunktion potentiell jeder Aspekt eines Szenarios (vgl. Abschnitt 2.3.2) relevant ist.

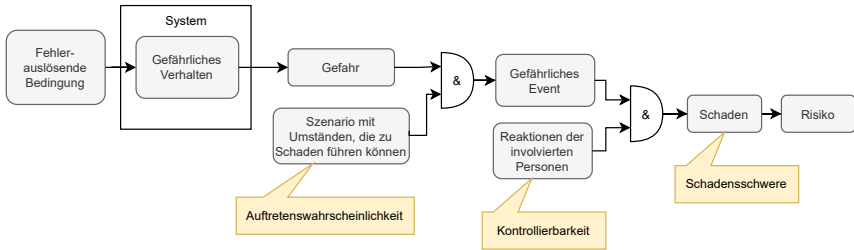


Abbildung 2.6: Überblick über den Ansatz zur Herleitung des Risikos nach [55].

Zur ausreichenden Beurteilung müssen möglichst viele dieser Szenarien bekannt sein. Aus einer Risiko-Sicht ist jedoch zusätzlich nicht nur die bloße Existenz eines Szenarios relevant, sondern auch die Wahrscheinlichkeit, mit der das Szenario in der ODD auftritt. Anschaulich ist ein Risiko wesentlich geringer, wenn es durch ein Szenario ausgelöst wird, das nur sehr selten in der ODD auftritt. Im Gegensatz dazu sind Schäden, die aus sehr häufig auftretenden Szenarien resultieren können, als höheres Risiko zu bewerten.

Für eine erfolgreiche und realistische Abschätzung des Risikos beim Betrieb einer automatisierten Funktion in einer gegebenen ODD sind also Methoden notwendig, um die Auftretenswahrscheinlichkeit von Szenarien zu quantifizieren [26]. Diese müssen im Prozess-Schritt „Erstellen von Szenarien“ des szenariobasierten Testen (siehe Abschnitt 2.3.3) integriert werden, sodass jedes Szenario im Szenario-Katalog mit einer Auftretenswahrscheinlichkeit annotiert ist.

2.4 Relevante Grundlagen der Datenanalyse

2.4.1 Prozess zur Wissensextraktion aus Datensätzen

Der Prozess zur Wissensdetektion in Datensätzen (engl. Knowledge Discovery in Databases (KDD)) ist ein Prozess, um aus Datensätzen Wissen zu extrahieren. Bei den Datensätzen handelt es sich um Rohdaten, welche zu umfassend sind um sie manuell zu analysieren. Abhängig von der Natur der Datensätze und dem Ziel der

Datenanalyse werden die Rohdaten in kompaktere, abstraktere oder nützlichere Formen gebracht, um eine menschliche Interpretation und Analyse zu ermöglichen. Dabei ist der Prozess unabhängig von der Art der Daten sowie von der Art der verwendeten Algorithmen, sondern bietet einen Rahmen für eine strukturierte Durchführung beliebiger Analysen. [38]

Der Begriff wurde erstmalig von Piatetsky-Shapiro 1990 erwähnt [92]. Der KDD umfasst fünf Schritte, welche durch die Entwicklung des initialen Zielverständnisses sowie der Nutzung der Erkenntnisse eingerahmt sind (vgl. Abbildung 2.7).

Zielverständnis Dieser Schritt ist dem eigentlichen Prozess vorgelagert und umfasst die Entwicklung eines Verständnisses der Anwendungsdomäne und des relevanten Vorwissens sowie die Identifikation des Ziels des KDD-Prozesses.

Datenauswahl Der erste Schritt des KDD ist die Erstellung eines Zieldatensatzes: die Auswahl eines Datensatzes oder die Konzentration auf eine Teilmenge von Variablen oder Datenpunkte, für die eine Datenanalyse durchgeführt werden soll.

Vorverarbeitung Der zweite Schritt ist die Datenbereinigung und -vorverarbeitung. Zu den grundlegenden Operationen gehören das Entfernen von Rauschen, das Festlegen von Strategien für den Umgang mit fehlenden Datenfeldern und das Berücksichtigen von Zeitreiheninformationen.

Transformation Der dritte Schritt ist die Datenreduktion und -projektion: die Suche nach nützlichen Merkmalen zur Darstellung der Daten, abhängig vom Ziel der Datenanalyse (Schritt 1). Mit Methoden zur Dimensionsreduktion oder Transformation wird die effektive Anzahl der zu berücksichtigenden Variablen bei Bedarf reduziert.

Datenanalyse Der vierte Schritt ist der Kern des KDD-Prozesses und besteht aus zwei Teilschritten. Abhängig vom Ziel des KDD-Prozesses wird zuerst ein passender Datenanalyse-Algorithmus ausgewählt. Im zweiten Schritt wird der Algorithmus angewendet, um interessante Muster in einer bestimmten Darstellungsform, beispielsweise Klassifikationsregeln, Regressionsmodelle, Cluster oder trainierte Modelle zu erhalten. Dieser Schritt profitiert von der korrekten Durchführung der vorangegangenen Schritte.

Interpretation Der fünfte Schritt ist die Interpretation der gesammelten Muster, wobei möglicherweise zu einem der vorangegangenen Schritte für weitere Iterationen zurückgekehrt wird. Dieser Schritt umfasst z.B. auch die Visualisierung der extrahierten Muster und Modelle oder die Visualisierung der Daten anhand der extrahierten Modelle.

Nutzung Dieser Schritt ist dem eigentlichen KDD-Prozess nachgelagert und besteht darin, das extrahierte Wissen zu nutzen, beispielsweise durch direkte Verwendung, Einbindung in ein anderes System für weitere Maßnahmen oder Dokumentation für interessierte Parteien.

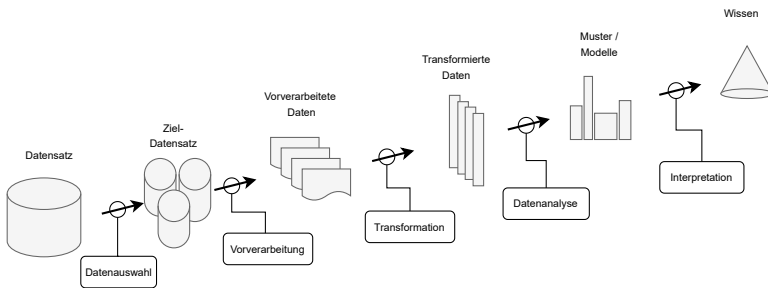


Abbildung 2.7: Prozess zur Entdeckung von Wissen in Datenbanken nach [38].

2.4.2 Distanzmaße für Zeitreihen

Definition 2.32 (Zeitreihe) Eine Zeitreihe $F_i = \{f_{i1}, \dots, f_{iT}\}$ der Länge T_i ist eine Sequenz von realen Zahlen, die mit konstanter Rate aufgezeichnet wurde. Jede reale Zahl drückt einen gemessenen Wert aus. [122]

Ein Zeitreihen-Distanzmaß ρ misst die Unterschiedlichkeit von zwei Zeitreihen F_i, F_j und bildet diese auf einen Wert ab:

$$\rho(F_i, F_j) \rightarrow \mathbb{R} \geq 0. \tag{2.2}$$

Euklidische Distanz

Für die Berechnung des euklidischen Abstands zwischen zwei Zeitreihen müssen beide Zeitreihen F_i, F_j die gleiche Länge besitzen. Gilt $T_i = T_j = T$, wird die euklidische Distanz mit

$$\rho_{\text{euklid}}(F_i, F_j) = \|f_i - f_j\| = \sqrt{\sum_{t=1}^T (f_{it} - f_{jt})^2} \quad (2.3)$$

berechnet.

DTW - Dynamic Time Warping

Dynamic Time Warping (DTW) ist ein Distanzmaß, das zwei gegebene Zeitreihen unabhängig von ihrer Länge miteinander vergleichen kann [106]. Der DTW-Algorithmus erstellt eine Distanzmatrix $C \in \mathbb{R}^{T_i \times T_j}$, die aus allen paarweise bestimmten Distanzen zwischen X und Y besteht:

$$C \in \mathbb{R}^{N \times M} : c_{i,j} = \|f_i - f_j\|, i \in [1 : T_i], j \in [1 : T_j] \quad (2.4)$$

Die Distanzmatrix wird als Kostenmatrix für die Abbildung zweier Zeitreihen bezeichnet. Nach der Erstellung dieser Matrix wird der *Warping-Pfad* $W = w_1, \dots, w_k, \dots, w_K$ bestimmt, der die zwei Zeitreihen mit den geringsten Kosten aufeinander abbildet. Jedes Pfadelement beschreibt ein Feld der Kostenmatrix $w_k = (i, j)_k$. Für den Warping-Pfad gelten drei Bedingungen:

1. $w_1 = (1, 1)$ und $w_K = (T_i, T_j)$. Der Warping-Pfad muss demnach in der linken oberen Ecke der Matrix starten und rechts unten enden.
2. Aufeinander folgende Pfadelemente müssen auf angrenzende Zellen weisen, einschließlich diagonalen Zellen.

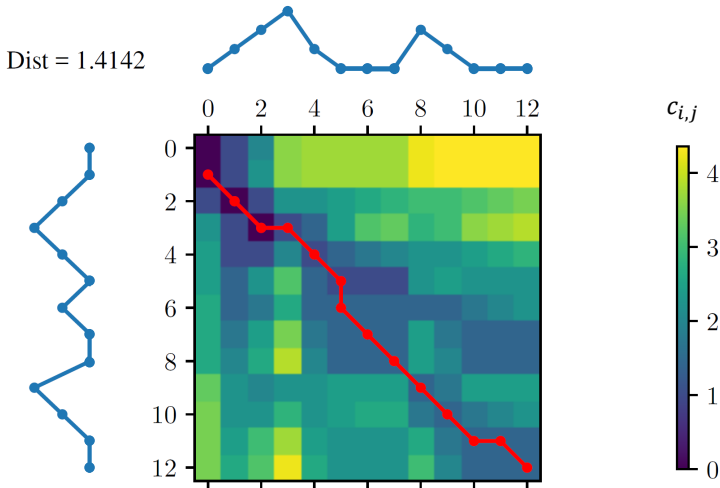


Abbildung 2.8: Kostenmatrix sowie kostengünstigster Pfad für zwei exemplarische Zeitreihen.

- Der Pfad muss monoton in der Zeit verlaufen. Das auf ein Pfadelement $w_k = (i, j)$ folgende Element w_{k+1} darf demnach nur rechts $w_{k+1} = (i, j + 1)$, unterhalb $w_{k+1} = (i + 1, j)$ oder diagonal $w_{k+1} = (i + 1, j + 1)$ liegen.

Da es im Allgemeinen mehr als einen Warping-Pfad gibt, der diese Bedingungen erfüllt, wird der Pfad gesucht, dessen aufsummierte Kosten minimal sind [23]. Die aufsummierten Kosten der Pfadelemente ergeben die DTW-Distanz. Der Matrix C für zwei exemplarische Zeitreihen sowie der Pfad mit den minimalen Kosten ist in Abbildung 2.8 dargestellt. Die resultierende, optimale Zuordnung zwischen beiden Zeitreihen zeigt Abbildung 2.9. Der Effekt kann als abschnittsweises Stauchen oder Strecken der Zeitreihen in zeitlicher Richtung betrachtet werden: Beispielsweise wird der Abschnitt der Zeitschritte 10-12 in der unteren Zeitreihe auf die Zeitschritte 11-12 abgebildet, was ein Stauchen dieses Abschnitts darstellt.

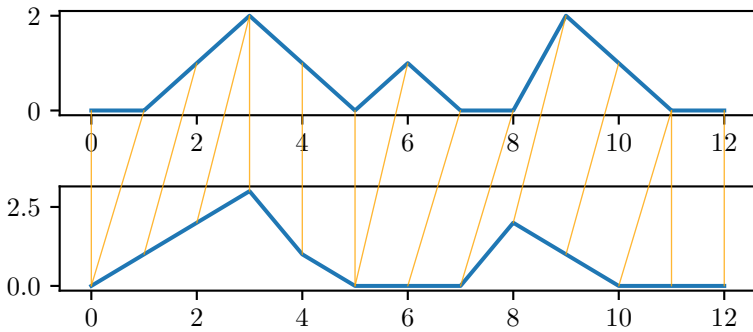


Abbildung 2.9: Resultierende Zuordnung aus der DTW-Berechnung für zwei exemplarische Zeitreihen.

LCSS-Distanz

Eine dritte Möglichkeit zur Bestimmung der Distanz von zwei Zeitreihen ist die Nutzung von Methoden auf Basis der Levensthein-Distanz. Die Levensthein-Distanz ist definiert auf Symbolketten und zählt die Anzahl der nötigen Ersetzungen, Einfügungen oder Löschungen zwischen zwei Symbolketten. Die Levenshtein-Distanz zwischen den Symbolketten „SEITE“ und „SAITEN“ ist 2, da eine Ersetzung („e“ zu „a“) und eine Einfügung („n“) notwendig ist. [132] Eine Variante der Levensthein-Distanz ist Longest Common Subsequence (LCSS)-Distanz, welche die längste gemeinsame Sequenz zweier Symbolketten beschreibt. Sie basiert auf einer Zuordnung der gleichen Symbole in der gegebenen Reihenfolge, lässt aber auch Symbole zu, die nicht zugeordnet sind. Im Beispiel ergibt sich die LCSS zu „SITE“.

Um die LCSS-Distanz auf Zeitreihen anzuwenden, wird ein Parameter ϵ gesetzt, um den maximal erlaubten Unterschied zweier Werte zu definieren, mit dem die Werte noch als gleich bezeichnet werden. Mithilfe dieser Anpassung kann die LCSS-Distanz für zwei Zeitreihen F_i, F_j beliebiger Länge angewendet werden:

dazu wird die längste, übereinstimmende Subsequenz bestimmt. Diese hat maximal die Länge der kürzeren Zeitreihe und kann daher mit dieser normiert werden. Das Distanzmaß ergibt sich dann zu

$$\rho_{LCSS} = 1 - \frac{LCSS}{\min(T_i, T_j)}, \quad (2.5)$$

wobei $LCSS$ die Länge der längsten, übereinstimmenden Subsequenz und T_i, T_j die Länge der Zeitreihen F_i, F_j sind. Um bei größeren Unterschieden auf größere Werte zu erhalten, muss das Ergebnis noch von 1 subtrahiert werden. [117]

3 Stand der Wissenschaft zur datenbasierten Szenarienanalyse

3.1 Ziele und Herausforderungen

3.1.1 Zentrale Annahmen

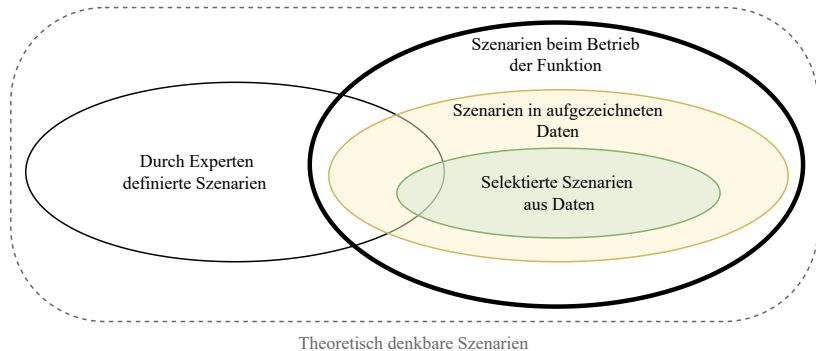


Abbildung 3.1: Mengendiagramm zur Visualisierung der zentralen Annahmen zur datenbasierten Szenarioanalyse. Die dargestellten Größen der Mengen sowie ihrer Vereinigungen und Differenzen stellen keine quantitative Aussage dar.

Die zentralen, der datenbasierten Szenarienanalyse zugrunde liegenden Annahmen lassen sich in einem Mengendiagramm darstellen (vgl. Abbildung 3.1). Den Rahmen bilden die theoretisch denkbaren Szenarien: Diese beinhalten alle grundsätzlich physikalisch möglichen Szenarien. Für die Validierung relevante Szenarien sind die, welche beim Betrieb der automatisierten Fahrfunktion auftreten

und getestet werden müssen (breite schwarze Ellipse). Diese bilden eine Teilmenge der theoretisch denkbaren Szenarien. Die von Experten definierten Szenarien sind ebenso eine Teilmenge, die jedoch nicht deckungsgleich mit den Szenarien im Betrieb ist (vgl. Abschnitt 2.3.3). Die erste Annahme ist, dass die Szenarien beim Betrieb der Funktion aus Szenarien in aufgezeichneten Daten (orange) abgeleitet werden können. Da die Datensätze in Bezug auf die enthaltenen Szenarien redundant sind (vgl. Abschnitt 2.3.2), wird die zweite Annahme getroffen: alle vorhandenen Szenarien in einem gegebenen Datensatz können durch Selektion der richtigen Teildatensätze repräsentiert werden.

3.1.2 Ziele der datenbasierten Szenarienanalyse

Datenbasierte Szenarienanalyse dient dazu, die expertenbasierten Szenarien zu ergänzen (vgl. Abschnitt 2.3.3). Sie hat zwei zentrale Ziele:

1. Die Detektion von neuen, bisher unbekanntem Szenarien
2. Die Extraktion von Wissen über bereits bekannte Szenarien

Detektion neuer Szenarien

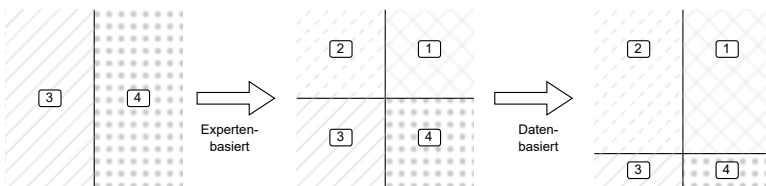


Abbildung 3.2: Szenario-Raum zu Beginn der Entwicklung, nach der expertenbasierten Szenariodefinition sowie nach der zusätzlichen datenbasierten Szenarienanalyse. Die Zahlen stehen hier, für bekannte, ungefährliche Szenarien (1); bekannte, gefährliche Szenarien (2); unbekannte, gefährliche Szenarien (3) sowie unbekannte, ungefährliche Szenarien (4) (vgl. Abschnitt 2.3.4).

Neue Szenarien sind Szenarien, die sich von den bekannten Szenarien auf semantischer Ebene unterscheiden, da sie beispielsweise unbekannte Aktionen oder

Kombinationen von Aktionen von Teilnehmern beinhalten, welche potentiell Einfluss auf das Verhalten einer automatisierten Fahrfunktion haben.

Zu Veranschaulichung dieses ersten Ziels der Szenarienanalyse in Realdaten wird der Raum aller vorhandenen Szenarien mit der zweidimensionalen Einteilung in bekannt/unbekannt sowie gefährlich/ungefährlich abgebildet betrachtet (vgl. Abbildung 3.2).

Zu Beginn der Entwicklung ist weder das zu entwickelnde System ausreichend definiert noch die ODD bekannt, daher sind alle Szenarien unbekannt. Im Rahmen der Anforderungserhebung werden initiale Szenarien definiert, anhand derer die ODD beschrieben werden kann (vgl. Abschnitt 2.2.1) [108]. Parallel zur Systementwicklung werden Daten innerhalb der definierten ODD gesammelt. Durch die datenbasierte Szenarienanalyse werden neue Szenarien detektiert und somit der Raum der bekannten Szenarien vergrößert. Dadurch verschiebt sich die Grenzen zwischen bekannten und unbekannt Szenarien in Abbildung 3.2 weiter nach unten. Durch die Durchführung der Szenarienanalyse in werden die gesammelten Daten laufend auf neue Szenarien untersucht. Somit wächst die Anzahl der bekannten Szenarien mit zunehmender Entwicklungsdauer immer weiter und die Grenze zwischen bekannten und unbekannt Szenarien verschiebt sich weiter nach unten.

Die Achse zwischen gefährlichen und ungefährlichen Szenarien wird hier nicht verschoben, da die datenbasierte Szenarienanalyse keine Aussage über die Sicherheit eines Szenarios trifft. Dies passiert im nächsten Schritt der szenariobasierten Validierung, dem Testen (vgl. Abschnitt 2.3.3). Jedoch ist die Bekanntheit eines Szenarios eine notwendige Vorbedingung, um eine Aussage über die Sicherheit eines Szenarios treffen zu können.

Extraktion von Wissen über Szenarien

Das zweite Ziel der Szenarienanalyse befasst sich mit der Extraktion von Wissen über bereits bekannte Szenarien. Relevantes Wissen in diesem Kontext sind die

Parameterbereiche eines logischen Szenarios (vgl. Abschnitt 2.3.2), die Auftretenswahrscheinlichkeit in der ODD sowie die Kritikalität der konkreten Szenarien.

Definition 3.1 (Kritikalität) *Die Kritikalität eines Szenarios ist das kombinierte Risiko der Teilnehmer eines Szenarios. [84]*

Definition 3.2 (Auftretenswahrscheinlichkeit) *Die Auftretenswahrscheinlichkeit eines Szenarios ist die Wahrscheinlichkeit, mit der ein gegebenes Szenario beim Betrieb eines Systems in einer gegebenen ODD auftritt.*

Für die Parameterbereiche werden reale Szenarien betrachtet und realistische Parameterverteilungen abgeleitet, beispielsweise für die Geschwindigkeitsverteilung von konkreten Szenarien (vgl. Abschnitt 2.3.3).

Die Kritikalität wird anhand gängiger Kritikalitätsmetriken bestimmt. Ein Beispiel für eine Kritikalitätsmetrik ist die Time to Collision (TTC). Diese beschreibt die Zeit, die bis zu einer Kollision vergeht, wenn alle Teilnehmer ihre Geschwindigkeiten beibehalten [51]. Die TTC eignet sich zur Beschreibung von Auffahrunfällen bei einander folgenden Fahrzeugen. Die Kritikalität anderer Verkehrssituationen kann durch weitere Metriken¹ beschrieben werden: Für Interaktionen mit Fußgängern existieren beispielsweise die Time to Zebra (TTZ) [116], welche die Zeit eines Verkehrsteilnehmers bis zu einem Zebrastreifen beschreibt oder der Pedestrian Risk Index (PRI), der anhand von TTC und Fahrzeug-Geschwindigkeit ein Unfallrisiko bestimmt [22]. Die Analyse der Kritikalität dient der Priorisierung von Testaufwänden sowie zur Einordnung von sicheren und unsicheren Szenarien.

Die Auftretenswahrscheinlichkeit eines Szenarios wird genutzt, um Risiken beim Betrieb der Funktion abzuschätzen (vgl. Definition 2.9). Im Rahmen des Schritts „Testen“ werden für Szenarien Fehlerwahrscheinlichkeiten ermittelt. Aus der Kombination von Auftretenswahrscheinlichkeiten der Szenarien und der Fehlerwahrscheinlichkeiten innerhalb der Szenarien wird die globale Wahrscheinlichkeit des Eintritts von Schaden bestimmt. Gemeinsam mit der erwarteten Schwere der möglichen Schäden kann dann ein Risiko beim Betrieb der Funktion bestimmt

¹ Eine Übersicht ist in [125] gegeben.

werden. Im Rahmen der Sicherheitsargumentation (vgl. Abschnitt 2.3.3) muss dann abgeschätzt werden, ob ein unzumutbares Risiko beim Betrieb der Funktion vorliegt (vgl. Definition 2.10). Die Auftretenswahrscheinlichkeit eines Szenarios ist bereits eine gängige Größe in der funktionalen Sicherheit, wo sie in verschiedene Klassen E0-E4 eingeteilt wird (vgl. Abschnitt 2.2.2).

3.1.3 Aspekte der Nutzung von Realdaten

Datensatz-Klassen

Grundsätzlich existieren zwei Möglichkeiten für die Aufzeichnung von Realdaten in einer ODD: Ego-Datensätze und externe Datensätze:

Definition 3.3 (Ego-Datensatz) *Ein Datensatz, der durch die Nutzung eines mit Sensoren zur Wahrnehmung des Umfelds und Detektion der anderen Verkehrsteilnehmer ausgerüsteten Fahrzeugs aufgezeichnet wurde. Ego-Datensätze enthalten die aus den Sensoren bestimmten relativen Trajektorien anderer Verkehrsteilnehmer. Des Weiteren können Informationen über interne Fahrzeugzustände mit aufgezeichnet werden.*

Definition 3.4 (Externer Datensatz) *Ein Datensatz, der durch die Nutzung fahrzeug-externer Sensorik, beispielsweise befestigt an Infrastruktur oder Drohnen aufgezeichnet wurde. Externe Datensätze zeichnen das Verkehrsgeschehen aus der Vogelperspektive^a auf. Durch eine Verarbeitung der Sensordaten werden Verkehrsteilnehmer und deren Bewegungen extrahiert.*

^a „Vogelperspektive“ umfasst hier alle Möglichkeiten der Aufzeichnung aus erhöhter Position, einschließlich Sensoren auf Masten oder Ampeln.

Für beide Arten von Datensätzen gilt, dass nur endliche Ausschnitte der Umgebung erfasst werden können. Ego-Datensätze sind näher am realen Sensoreindruck eines automatisierten Fahrzeugs, beinhalten jedoch auch durch Verkehrsteilnehmer, Vegetation oder Infrastruktur verdeckte Abschnitte der Umgebung. Abgesehen

von Spezialfällen wie Tunneln oder großen Bäumen ist in externen Datensätzen die Umgebung verdeckungsfrei aufgezeichnet. Für beide Ausprägungen sind wissenschaftlich frei verwendbare Datensätze verfügbar (vgl. Abschnitt 3.2.3)

Herausforderung durch Redundanz

Die zentrale Herausforderung bei der Verwendung von Realdaten im Kontext von Verkehrsaufzeichnungen ist ihre Redundanz bezüglich der auftretenden Szenarien (vgl. Abschnitt 2.3.2) [4]. Dies gilt sowohl für die Aufzeichnung von Daten aus einem Ego-Fahrzeug als auch aus externer Infrastruktur an einer festen Position (z.B. einer Kreuzung). Während bei ersterem üblicherweise lange Folge- oder Freifahrten überwiegen, sind es bei letzterem lange Fahrzeug-Kolonnen, welche die Kreuzung ohne Abzubiegen überqueren. Daraus resultiert, dass nur ein geringer Anteil der Aufzeichnungen zur Vergrößerung des bekannten Szenarienraums beiträgt. Die Definition eines neuen logischen Szenarios für jede in den Daten beobachtete Verkehrssituation ist also nicht sinnvoll und würde den resultierenden Szenariokatalog mit vielen ähnlichen Szenarien unnötig vergrößern. Ansätze zur Szenarienanalyse in Realdaten müssen also mit dieser Redundanz geeignet umgehen können.

3.1.4 Formalisierung

Ein Datensatz \mathcal{D} mit n aufgezeichneten Verkehrsteilnehmern kann formal durch

$$\mathcal{D} = \{T_1, \dots, T_i, \dots, T_n\} \quad (3.1)$$

beschrieben werden, wobei die T_i alle aufgezeichneten Signale für einen im Datensatz vorkommenden Verkehrsteilnehmer mit Index $1 \leq i \leq n$ beinhaltet. Jedes Verkehrsobjekt T_i wird durch ein Tupel

$$T_i = (A_i, \mathbf{M}_i) \quad (3.2)$$

beschrieben. Dabei umfasst die Menge A_i alle Attribute des Verkehrsobjekts, die zeitlich nicht veränderlich sind. Hier wird der Typ des Verkehrsteilnehmers (vgl. Definition 3.5) und seine räumliche Ausdehnung (z.B. Fahrzeugbreite, Fahrzeuglänge) sowie mögliche weitere Attribute angegeben.

Definition 3.5 (Teilnehmertyp) *Die einem dynamischen Verkehrsteilnehmer zugewiesene Klasse, die seine dynamischen Eigenschaften beschreibt. Die möglichen Teilnehmertypen sind als Elemente des Tupels O definiert.*

Übliche Teilnehmertypen sind $O = (\text{Auto}, \text{Fußgänger}, \text{Fahrradfahrer}, \text{Truck})$. Die Signale, welche die Trajektorie eines Verkehrsobjekts T_i beschreiben, gibt die Matrix $\mathbf{M}_i \in \mathbb{R}^{m \times d_i}$ an, wobei m die Anzahl der aufgezeichneten Signale und d_i die Anzahl Messpunkte pro Signal angibt. Elementare Signale zur Beschreibung der Trajektorie sind die Positionen und Geschwindigkeiten der Verkehrsobjekte, sodass \mathbf{M}_i als

$$\mathbf{M}_i = [\mathbf{t}, \mathbf{x}^g, \mathbf{y}^g, \mathbf{v}_x^g, \mathbf{v}_y^g]_i \quad (3.3)$$

geschrieben werden kann. \mathbf{t} umfasst die Messzeitpunkte, $\mathbf{x}^g, \mathbf{y}^g$ die Position und $\mathbf{v}_x^g, \mathbf{v}_y^g$ die Geschwindigkeiten von T_i . Die Messzeitpunkte \mathbf{t} sind äquidistant, ein Datensatz wurde also mit einer festen Abtastrate aufgezeichnet. Die räumlichen Informationen sind in einem Koordinatensystem g definiert, das vom jeweiligen Datensatz vorgegeben wird.

Im Fall eines Ego-Datensatzes muss zwischen dem Ego-Fahrzeug E (vgl. Definition 2.26) und anderen Verkehrsobjekten T_i^e unterschieden werden:

$$\mathcal{D}_{\text{ego}} = \{E\} \cup \{T_1^e, \dots, T_i^e, \dots, T_n^e\}. \quad (3.4)$$

Die Bewegungen des Ego-Fahrzeugs werden analog zu (3.2) und (3.3) als Bewegung in einem global definierten Koordinatensystem beschrieben und durch die Eigenlokalisierung des Ego-Fahrzeugs bestimmt (vgl. Abschnitt A.2). In Ego-Datensätzen sind die Verkehrsobjekte $T_i^e = (A_i, \mathbf{M}_i^e)$ relativ zum Ego-Fahrzeug beschrieben mit

$$\mathbf{M}_i^e = [\mathbf{t}, \mathbf{x}^e, \mathbf{y}^e]_i. \quad (3.5)$$

Die Signale der Positionen x^e, y^e sind hier im Fahrzeug-Koordinatensystem des Ego-Fahrzeugs, bezeichnet mit e , gegeben (vgl. A.4.3).

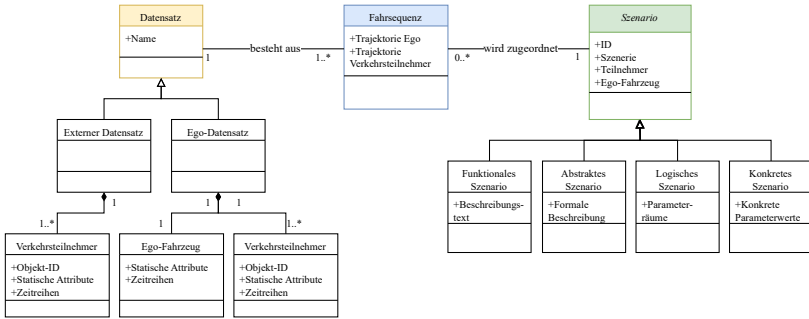


Abbildung 3.3: Klassendiagramm für den Zusammenhang zwischen Datensätzen, Fahrsequenzen und Szenarien.

Da Datensätze im Allgemeinen mehr als ein einziges Szenario beinhalten können, müssen sie zur Szenarienanalyse in Fahrsequenzen zerlegt werden.

Definition 3.6 (Fahrsequenz) Eine Fahrsequenz ist eine Sequenz von aufgezeichneten Trajektorien, welche die Bewegung eines Ego-Fahrzeugs (nach Gleichungen (3.2) und (3.3)) mitsamt den umgebenden Verkehrsobjekten (nach Gleichungen (3.2) und (3.5)) beschreibt. Eine Fahrsequenz kann einem Szenario (z.B. konkreten, logischen, abstrakten und funktionalen) zugeordnet werden.

Im Gegensatz zu konkreten, logischen, abstrakten und funktionalen Szenarien handelt es sich bei einer Fahrsequenz nicht um eine Beschreibung auszuführender Aktivitäten, sondern um die Aufzeichnung der Aktivitäten. Ein Datensatz nach Definition 3.4 kann aus einer variablen Anzahl Fahrsequenzen bestehen. Eine Fahrsequenz S_j kann also als

$$S_j = \{E^j\} \cup \{T_1^j, T_2^j, \dots, T_{n_s}^j\} \quad (3.6)$$

geschrieben werden. E^j sowie T^j bezeichnen hier die auf die Dauer der Fahrsequenz gekürzten Signale des Ego-Fahrzeugs sowie der Teilnehmer (vgl. Abb.

3.3). Die Zuordnung einer Fahrsequenz S_j zu einem Szenario geschieht durch eine Funktion Z mit

$$\Omega = Z(S_j), \quad (3.7)$$

wobei Ω ein spezifisches Szenario bezeichnet. Abhängig davon, ob Σ als funktionales, abstraktes, logisches oder konkretes Szenario vorliegen soll, muss eine entsprechende Funktion $Z(S)$ gewählt werden.

3.2 Existierende Ansätze und Datensätze

3.2.1 Szenarienanalyse auf Autobahnen

Elrofai et al. nutzen in einem frühen Ansatz aus dem Jahr 2016 interne Fahrzeugsensorik zur Aufzeichnung von Signalen wie Geschwindigkeit und Gierrate des Ego-Fahrzeugs. Auf diesen Signalen entwickeln sie dann Detektoren für zwei verschiedene Manöver der Ego-Fahrzeugs: Abbiegen und Spurwechsel. Diese werden anhand von manuell annotierten Abbiege- und Spurwechsel-Manövern innerhalb von Fahrten auf einer niederländischen Autobahn ausgewertet. Dabei kommen sie auf Genauigkeiten von ca. 76%. Für jeden detektierten Spurwechsel bzw. Abbiegevorgang wird dann ein Szenario abgelegt, wobei die weiteren relevanten Parameter der Szenarien (siehe Szenario-Schichtenmodell in Abschnitt 2.3.2) manuell ergänzt werden müssen. [31]

Ein konzeptionell ähnlicher Ansatz wird im selben Jahr von Roesener et al. präsentiert. Auch hier wird eine begrenzte Anzahl Manöver in verschiedenen Signalen untersucht und klassifiziert. Dabei werden aus den Zeitreihen verschiedene Merkmale, beispielsweise die laterale Distanz zum nächsten Verkehrsteilnehmer bestimmt. Bei den ausgewählten Manövern handelt es sich hier um „Spurwechsel“, „Freie Fahrt“ sowie „Einscherer“. Die Klassifikationsansätze liefern für die untersuchten Szenarien Erkennungsraten im Bereich von 72% bis 97% (F-Score). [97]

Die beiden beschriebenen Ansätze befassen sich ausschließlich mit der Klassifikation und Beschreibung der Bewegung des Ego-Fahrzeugs.

Erdogan et al. betrachten erstmals auch Manöver von anderen Verkehrsteilnehmern. Zur Klassifikation von Manövern werden verschiedene Detektoren auf synthetischen generierten Daten trainiert (vgl. Abbildung 3.4). Die Autoren ver-

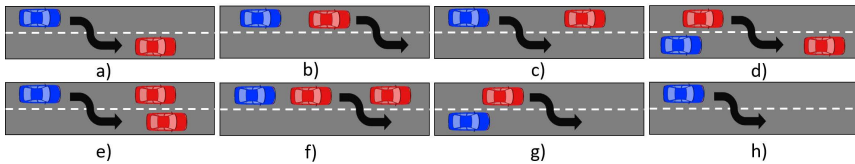


Abbildung 3.4: Betrachtete Szenarien, definiert durch die Manöver des Egos sowie des Vorderfahrzeugs: a) Einscheren Ego-Fahrzeug, b) Ausscheren Vorderfahrzeug, c) Ausscheren Ego Fahrzeug, d) Wechsel des Vorderfahrzeugs durch Einscheren, e) Ego Ausscheren + Einscheren, f) Wechsel des Vorderfahrzeugs durch Einscheren., g) Einscheren Vorderfahrzeug , h) Spurwechsel des Egos [33]

gleichen in ihrer Arbeit drei verschiedene Ansätze zur Detektion der Szenarien: einen regelbasierten Ansatz, einen überwachten Ansatz zur Klassifikation sowie einen unüberwachten Ansatz durch Clustering. Die erzielten Ergebnisse weisen darauf hin, dass Ansätze aus dem maschinellen Lernen bessere Ergebnisse erzielen. Des weiteren finden die Autoren einen Leistungsunterschied zwischen überwachter Klassifikation und unüberwachtem Clustering: Während die Klassifikation eine bessere Precision aufweist, erreicht das Clustering bessere Werte bei der Betrachtung des Sensitivität. [33]

Der Ansatz des Trainings von Klassifikatoren auf synthetischen Daten zur Detektion von Szenarien wurde auch von Pfeffer aufgegriffen. Er beschreibt einen Prozess zur Detektion von Szenarien in realen Verkehrsaufzeichnungen (siehe Abbildung 3.5). Dabei wird die gesamte Verarbeitungskette, beginnend bei der Datenaufzeichnung bis hin zur Ausführung der Szenarien in einer Simulation abgedeckt. Relevant für das Thema der vorliegenden Dissertation ist vor allem der Schritt „der Szenarien“ (Z2). Hierzu wurde eine Tensordarstellung gewählt, mit der sich die Bewegung und die Relationen mit dem Ego-Fahrzeug einer variablen

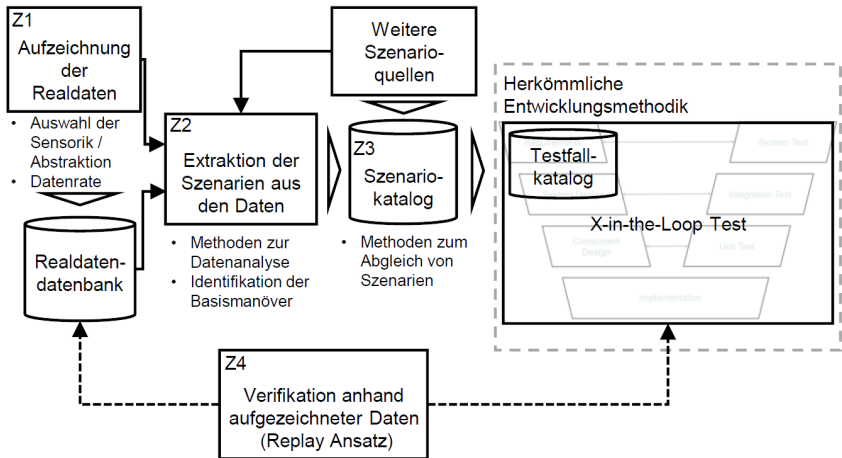


Abbildung 3.5: Prozess zur szenariobasierten simulationsgestützten Absicherung nach [91]

Anzahl von anderen Verkehrsobjekten darstellen lässt. Im Gegensatz zu den vorherigen Ansätzen ist somit eine erhöhte Flexibilität gegeben, da nicht nur vorher definierte Szenarien oder Manöver detektiert werden können. Zur Beschreibung der Manöver verschiedener Objekte wählt Pfeffer keinen fest im Voraus definierten Katalog, sondern beschreibt die Manöver über eine Codierung des Umfelds des Ego-Fahrzeugs (vgl. Abbildung 3.6). In der Detektion von Manövern in den realen Daten werden analog zu [33] Ansätze aus der Klassifikation und dem Clustering genutzt, wobei auch Pfeffer auf synthetische Trainingsdaten zurückgreift. Mit der eingeführten Darstellung liefert Pfeffer eine Beschreibung von beliebigen Szenarien auf mehrspurigen Straßen, die somit extrahiert und in definierter Form in einen Szenarienkatalog überführt werden können. [91]

Während die vorhergehenden Ansätze allesamt auf einer Klassifikation von zuvor definierten Manövern basieren, versuchen Hauer et al. die relevanten Szenarien durch ein unüberwachtes Clustering mit möglichst wenigen Annahmen zu extrahieren. Dabei argumentieren sie, dass mit jeder initialen menschlichen Annahme (beispielsweise der Definition einer Liste relevanter Manöver) mögliche Fehler


[long, lat]	Objekte hinter Egofahrzeug	Gleiche Höhe	Objekte vor Egofahrzeug
Objekte auf linker Spur	(-1,-1)	(0,-1)	(1,-1) 
Objekte auf gleicher Spur	(-1,0)		(1,0)
Objekte auf rechter Spur	(-1,1)	(0,1)	(1,1)

Abbildung 3.6: Codierung der räumlichen Zustände zur Abstraktion der Manöver von anderen Verkehrsteilnehmern [91]

passieren können. Die Idee ist daher, Abschnitte der Daten auf Basis von möglichst objektiven Distanzmaßen durchzuführen. Dazu wird das Gebiet um das Ego-Fahrzeug in verschiedene Teilbereiche aufgeteilt und die Trajektorien der dort vorhandenen Teilnehmer miteinander verglichen. Anhand dieser Trajektorienvergleiche wird das Distanzmaß zwischen zwei Fahrabschnitten bestimmt und eine vollständige Distanzmatrix zwischen allen Fahrabschnitten gefüllt um Merkmalsvektoren zu erhalten. In der Folge werden diese Merkmalsvektoren mit klassischen Verfahren geclustert. [49]

3.2.2 Szenarienanalyse in urbanen Bereichen

Im Gegensatz zu den oben beschriebenen Ansätzen, die die Domäne der Autobahnfahrt betrachten, müssen Ansätze im urbanen Umfeld mit einer größeren Varianz möglicher Verkehrsteilnehmer umgehen können. Einen frühen Ansatz zur Analyse urbaner Verkehrsszenarien präsentieren Barbier et al. im Jahr 2017. Anhand von Daten an einer urbanen Kreuzung (vgl. Abbildung 3.7) sollen die Manöver der Fahrzeuge klassifiziert werden. Dazu nutzen sie einen Random Forest Klassifikator [17], der mit einem synthetisch erzeugten Datensatz trainiert wird und longitudinale Manöver sowie Abbiegemanöver erkennen soll. Damit liefern



Abbildung 3.7: Betrachtete Kreuzung von Barbier et al. zur Erkennung von Manövern und Szenarien in Grenoble, Frankreich, aus [8]

sie einen Ansatz zur Detektion und Klassifikation von Szenarien an urbanen Kreuzungen mit Beschränkung auf Signale aus dem Ego-Fahrzeug. [8] Im Jahr 2019 beschreiben Hartjen et al. ein Verfahren zur Klassifikation von Fahrmanövern im urbanen Raum. Dabei nutzen sie einen umfassenden Manöverkatalog, eingeteilt in Klassen von Manövern. Diese teilen sich in Manöver zur Beschreibung des Zustands des Ego-Fahrzeugs, der Interaktion mit Infrastruktur sowie der Interaktion mit anderen Verkehrsteilnehmern auf (Abbildung 3.8). Mithilfe einer regelbasierten Klassifikation werden Manöver aus realen Verkehrsaufzeichnungen extrahiert und definierten funktionalen Szenarien zugeordnet. Zusätzlich beschreiben die Autoren einen weiteren Schritt, in dem die Parameter der extrahierten Manöver (beispielsweise die Geschwindigkeit des Ego-Fahrzeugs) analysiert und aggregiert werden. Die resultierenden Parameterräume dienen zur Beschreibung von logischen Szenarien. Der Ansatz wird auf Daten aus fest installierter Kamera-Infrastruktur exemplarisch angewendet. [48]

Ein ähnlicher Ansatz wird 2021 von King et al. präsentiert. Hier werden Kreuzungsdaten aus dem INTERACTION Datensatz [134] auf auftretende konkrete Szenarien untersucht, welche analog zu [48] zu logischen Szenarien aggregiert

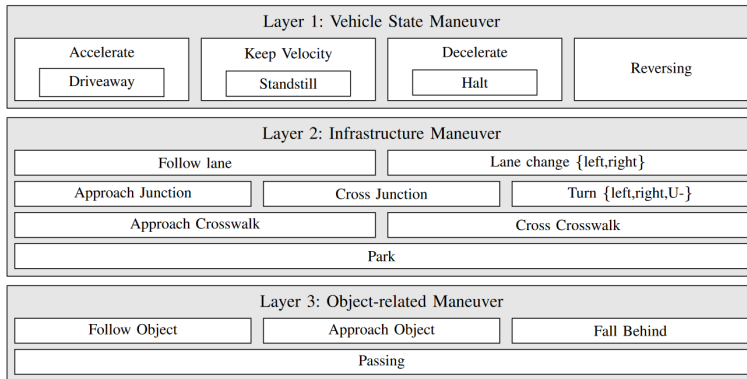


Abbildung 3.8: Darstellung der verwendeten Manöver zur Analyse urbaner Szenarien aus [48]

werden. Zusätzlich wird die Häufigkeit verschiedener Manöver in den Daten analysiert. [66]

In eine ähnliche Richtung wie der Ansatz von Hauer et al. [49] geht die Arbeit von Weber et al.² aus dem Jahr 2022. Auch diese Arbeit zielt auf die Minimierung von potentiell fehlerhaften Annahmen in der Szenarienanalyse ab. Der hier gewählte Ansatz beschränkt sich auf Szenarien, die aus dem Ego-Fahrzeug in Interaktion mit einzelnen relevanten Verkehrsteilnehmern entstehen. Somit bestehen Szenarien immer aus zwei Teilnehmern. Die relevanten Teilnehmer wurden hier definiert als die, welche einen Schnittpunkt mit der Trajektorie des Ego-Fahrzeugs haben. Der Ansatz wird auf dem inD-Datensatz [14] evaluiert, welcher eine große Zahl Fahrzeuge in Interaktion mit Fußgängern und Fahrradfahrern umfasst. [123]

3.2.3 Verfügbare Datensätze

Ego-Datensätze liegen seit Jahren im Fokus der Forschung und sind verfügbar für wissenschaftliche Zwecke. Im Jahr 2012 wurde mit dem „KITTI-Datensatz“

² Weber et al. beziehen sich in ihrer Arbeit auf vorhergehende Arbeiten des Autors, welche im Konzeptkapitel dieser Arbeit referenziert werden.

ein Ego-Datensatz, bestehend aus Daten von zwei Kameras sowie einem Lidar-Sensor, veröffentlicht [41]. Der Datensatz umfasst 39,2 km Fahrt im Umfeld von Karlsruhe. Darauf folgende Datensätze umfassten größere Umfänge, beispielsweise besteht das in den Jahren 2014 bis 2015 aufgezeichnete „Oxford Robotcar Dataset“ aus 1000 km wiederholter Fahrten im Raum Oxford. Dabei wurden 6 Kamera und ein Lidar-Sensor verwendet [76]. Mittlerweile ist eine große Auswahl an Ego-Datensätzen verfügbar, ein Überblick wird in [131] gegeben.

Die Ego-Datensätze umfassen somit Daten von Sensoren zur Wahrnehmung der Umgebung (vgl. Abschnitt A.2). Abhängig vom Ziel des Datensatzes sind zusätzlich Annotationen für verschiedene Aspekte vorhanden (z.B. Fußgänger in Bildern in [64]). Diese Datensätze eignen sich somit zur Entwicklung und dem Test von Wahrnehmungsfunktionen. Für die Ableitung von Szenarien sind Ego-Datensätze nicht optimal: bedingt durch die Aufnahme aus einem Fahrzeug heraus sind nicht alle relevanten Informationen über das Verhalten der anderen Verkehrsteilnehmer vorhanden. Beispielsweise wenn die Aufnahmefahrzeuge nur über Sensorik nach vorne verfügen oder Verdeckungen durch die Umgebung vorliegen.

Externe Datensätze werden aus einer erhöhten Position aufgezeichnet, damit für den Aufzeichnungsbereich keine Verdeckungen vorliegen und die Bewegung aller Verkehrsteilnehmer über die gesamte Zeit erfasst wird. Diese erhöhte Position kann beispielsweise erreicht werden, indem Sensoren an bestehender Verkehrsinfrastruktur wie Ampelmasten angebracht werden. Ein Beispiel für derartige Aufzeichnung liefert das 2016 aufgebaute Testfeld autonomes Fahren in Karlsruhe [139]. Eine weitere Möglichkeit ist die Aufzeichnung durch mit Kameras ausgerüsteten Drohnen. Hier sind verschiedene Datensätze verfügbar, in denen Kreuzungen [14, 134], Autobahnabschnitte [68] oder Kreisverkehre [69] von einer Drohne aufgezeichnet wurden. Bedingt durch die begrenzte Flugdauer von Drohnen sind diese Datensätze jedoch auf eine Dauer von wenigen Stunden begrenzt (z.B. Aufzeichnungsdauer inD-Datensatz: 10 Stunden). Bei beiden Aufzeichnungsarten werden im Algorithmus zur semantischen Segmentierung verwendet, die für jeden Pixel eines Bildes klassifizieren, ob er Teil der Straße, der Umgebung oder Teil eines Verkehrsteilnehmers ist [14].

3.3 Diskussion des Stands der Wissenschaft

3.3.1 Konzeptionelle Problematik des manöverbasierten Ansatzes

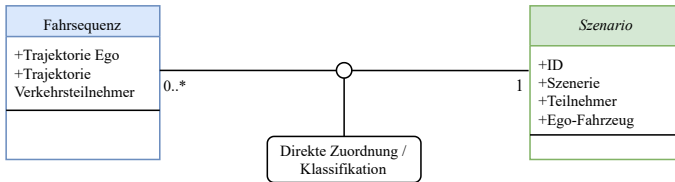


Abbildung 3.9: Direkte Zuordnung von Fahrsequenzen zu Szenarien

Die Mehrzahl der in Abschnitt 3.2 beschriebenen Ansätze basiert auf Manöverlisten und Manöverbeschreibungen zur Analyse der Szenarien in Realdaten. Damit beruhen diese Ansätze auf einer modellbasierten Beschreibung von Szenarien (vgl. Abschnitt 2.3.2). Fahrsequenzen werden somit einer spezifischen Beschreibung direkt zugeordnet (vgl. Abbildung 3.9).

Stachowiak [109] beschreibt drei allgemeine Eigenschaften eines Modells [63]:

Abbildung Ein Modell ist immer ein Abbild oder eine Repräsentation eines natürlichen oder künstlichen Originals.

Verkürzung Ein Modell umfasst nur die Eigenschaften des Originals, die dem Modellersteller oder -nutzer als relevant erscheinen.

Pragmatismus Ein Modell dient einem konkreten Einsatzzweck. Es ist einem Original nicht von sich aus zugeordnet sondern wird in einer bestimmten Zeitspanne für einen bestimmten Zweck eingesetzt.

Für die datenbasierte Szenarienanalyse im urbanen Kontext ist das Original der urbane Straßenverkehr. Diesen soll eine modellbasierten Szenariobeschreibung ausreichend erfassen. Jedes Manöver innerhalb der modellbasierten Szenariobeschreibung ist ein Modell eines gewissen realen Verhaltens eines Verkehrsteilnehmers. Entsprechend der zweiten Modelleigenschaft werden dabei Verkürzungen

vorgenommen, die „dem Modellersteller als relevant erscheinen“. Für die Detektion neuer Szenarien ist diese Modelleigenschaft problematisch: Szenarien oder Manöver, die der Modellersteller nicht vorhergesehen und definiert hat, sind nicht im Modell enthalten.

Ein Beispiel für nicht vorhersehbare Manöver von Verkehrsteilnehmern ist das Verhalten, wenn Einsatzfahrzeuge mit Blaulicht involviert sind. Während das zu erwartende Verhalten auf Autobahnen durch die Pflicht zur Bildung einer Rettungsgasse definiert ist, ist in der urbanen ODD Improvisation sowie Abweichung von Verkehrsregeln (z.B. das Befahren der Fußgängerwege) notwendig.

Da der offene Kontext (vgl. Definition 2.19) formal nicht vollständig beschrieben werden kann, kann auch die Vollständigkeit eines Modells zur Beschreibung dieses Kontexts (durch Szenarien) nicht nachgewiesen werden.

Ein weiteres Problem zeigt sich bei der konkreten Umsetzung von Algorithmen zur Klassifikation von Manövern in Datensätzen. Das Ziel dieser Algorithmen ist die Zuweisung von Manövern für alle Verkehrsteilnehmer in einem Datensatz (vgl. Gleichung (3.1)). Hier müssen zwei verschiedene Vorgehensweisen unterschieden werden: die regelbasierte und die lernbasierte Manöverklassifikation.

Beim regelbasierten Ansatz werden für jedes Manöver spezifische Schwellwerte für gemessene Signale definiert. Für das Manöver „Folgefahrt“ muss

$$v_{\text{ego}} \cong v_{\text{voraus}} \quad (3.8)$$

gelten, wobei v_{ego} die Geschwindigkeit des Ego-Fahrzeugs und v_{voraus} die Geschwindigkeit des vorausfahrenden Fahrzeugs bezeichnet. Da in gemessenen Signalen niemals völlige Gleichheit vorliegt, muss hier ein Toleranzbereich Δv definiert werden, sodass dann (3.8) durch

$$f(v_{\text{ego}}, v_{\text{voraus}}) = \begin{cases} 1 & (v_{\text{ego}} + \Delta v \geq v_{\text{voraus}}) \wedge (v_{\text{ego}} - \Delta v \leq v_{\text{voraus}}) \\ 0 & \text{sonst} \end{cases} \quad (3.9)$$

ersetzt wird. Die Größe des Toleranzbereichs wird durch die Wahl des Parameter Δv definiert. Dieses Vorgehen wird analog für alle relevanten Signale aller Manöver in der Manöverliste durchgeführt. Für die Wahl der Parameterwerte ist Expertenwissen erforderlich. In der Praxis werden die Parameterwerte anhand von Entwicklungsdatensätzen bestimmt [66].

Bei der lernbasierten Manöverklassifikation werden die Parameter nicht manuell anhand von Expertenwissen bestimmt, sondern es wird ein Modell trainiert (vgl. Abschnitt A.4.1). Dazu werden annotierte Trainingsdaten benötigt. Diese können durch manuelle Annotation von realen Daten erzeugt werden. Dazu ist allerdings manueller Aufwand notwendig, daher greifen einige Quellen auf simulativ generierte Trainingsdaten zurück [33, 91]. Simulierte Daten werden erzeugt, indem bekannte Szenarien, die dementsprechend bekannte Manöver beinhalten, mehrfach in einem Simulationsprogramm ausgeführt und aufgezeichnet werden. Dabei werden Details der Szenarien variiert, beispielsweise erhalten die Teilnehmer verschiedene Start-Geschwindigkeiten. Daraus kann, ohne manuellen Annotationsaufwand, ein Datensatz für das Training erzeugt werden.

In beiden Umsetzungsmethoden ergibt sich eine Abhängigkeit von den zur Entwicklung der Manöverklassifikation verwendeten Daten. Entsprechend der dritten Modelleigenschaft ist ein Modell nur in einer bestimmten Zeitspanne für einen bestimmten Zweck definiert. Diese Zeitspanne umfasst jedoch nur die Dauer des Entwicklungsdatensatz sowie den Zweck, die Manöver im Entwicklungsdatensatz richtig zu erkennen. Daher kann sich die resultierende Modellierung als unpassend für einen neuen Datensatz erweisen.

3.3.2 Szenarienanalyse durch Clustering

Im Gegensatz zur Klassifikation gehört Clustering zu den unüberwachten Ansätzen (vgl. Abschnitt A.4.1). In der Literatur finden sich verschiedene Clustering-Ansätze zur datenbasierten Szenarienanalyse (vgl. Abschnitt 3.2). Diese haben

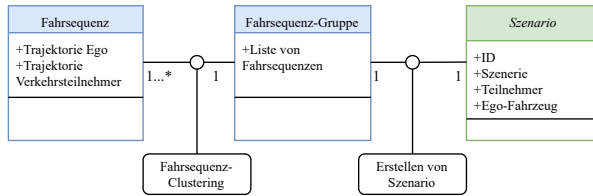


Abbildung 3.10: Clustering zur datenbasierten Szenarienanalyse.

zum Ziel, Gruppen von ähnlichen Fahrsequenzen zu finden ohne vorher Annahmen über die Beschaffenheit der Gruppen (beispielsweise Anzahl der Fahrsequenzen pro Gruppe oder Inhalt der Fahrsequenzen einer Gruppe) zu treffen. Erst nach einer vollständigen Einteilung eines Datensatzes in Gruppen wird der Inhalt der Gruppe in ein Szenario überführt (vgl. Abbildung 3.10).

Um bekannte Clustering-Verfahren wie k-Means oder DBSCAN durchführen zu können, muss ein Distanzmaß zwischen zwei zu clusternden Punkten definiert sein. Die in Abschnitt 3.2 beschriebenen Ansätze nutzen dazu jeweils selbst definierte Distanzmaße.

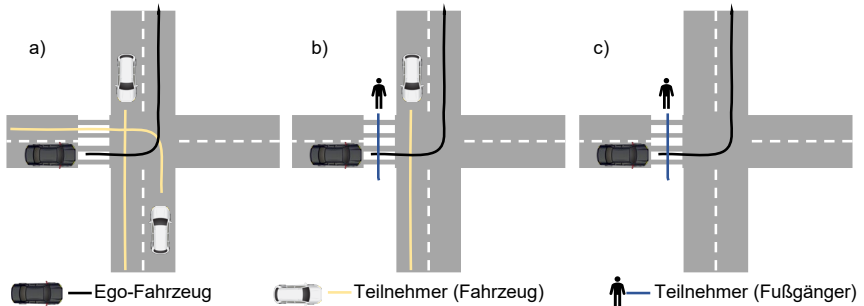


Abbildung 3.11: Drei Beispiele für urbane Verkehrssituationen mit verschiedenen Verkehrsteilnehmertypen. Die Verkehrsteilnehmer sind immer an ihrem jeweiligen Startpunkt mit samt ihrer darauf folgenden Bewegung dargestellt.

Im Fall des Clusterings von Fahrsequenzen mit einer variablen Anzahl von Verkehrsteilnehmern ist die Definition eines solchen Distanzmaßes jedoch nicht trivial: Zur Veranschaulichung sind in Abbildung 3.11 drei verschiedene Fahrsequenzen in einem urbanen Umfeld dargestellt. Dabei befinden sich sowohl Ego-Fahrzeug als auch die Teilnehmer an ihrem jeweiligen Startpunkt und bewegen sich in der Folge entlang ihrer dargestellten Trajektorie. Existierte ein Distanzmaß

$$\delta(S_i, S_j) \rightarrow \mathbb{R}, i, j \in \mathbb{N} \quad (3.10)$$

für zwei Fahrsequenzen S_i und S_j , könnten die drei Distanzen

$$\delta(S_a, S_b) = \Delta_{ab} \quad (3.11)$$

$$\delta(S_b, S_c) = \Delta_{bc} \quad (3.12)$$

$$\delta(S_a, S_c) = \Delta_{ac} \quad (3.13)$$

berechnet werden (vgl. dargestellte Fahrsequenzen a),b),c) in Abbildung 3.11. Da die reellen Zahlen \mathbb{R} mathematisch einen geordneten Körper darstellen, müssten somit Aussagen wie

$$\Delta_{ab} > \Delta_{ac} > \Delta_{bc} \quad (3.14)$$

formulierbar sein, was sprachliche Formulierungen wie „Die Fahrsequenz b) ist ähnlicher zu c) als zu a)“ erlauben würde. Ein Entwickler, der sich mit Notbremsystemen für Fußgänger befasst, würde dieser Aussage vermutlich zustimmen, da in beiden Fahrsequenzen ein Fußgänger anwesend ist. Ein Entwickler für die Wahrnehmungsfunktion von kreuzendem Verkehr würde dieser Aussage wohl widersprechen, da er den Fokus auf die Anwesenheit von kreuzendem Verkehr legen würde. Ähnlichkeit von Fahrsequenzen ist demnach ein subjektiver Begriff und ein von allen anerkanntes Distanzmaß $\delta(S_1, S_2) \rightarrow \mathbb{R}$ kann nicht allgemeingültig definiert werden. Die Betrachtung von Verkehrssituationen als Teil eines Raumes, in dem sinnvolle Distanzmaße oder Metriken anwendbar sind, ist demnach nicht zielführend.

Die in Abschnitt 3.2 skizzierten, unüberwachten Ansätze vermeiden dieses Problem durch eine starke Einschränkung der möglichen Szenarien. Einige Ansätze erreichen dies durch die Einschränkung auf die Domäne der Autobahn und einer damit einhergehenden starken Strukturierung der Umgebung sowie einen einzigen Verkehrsteilnehmertyp „Fahrzeug“ in einer sehr strukturierten Umgebung [33, 49, 91]. Andere Ansätze beschränken die analysierbaren Szenarien auf die Kombination eines Egos mit maximal einem weiteren Teilnehmer [123]. Somit ist keiner der in der Literatur beschriebenen Ansätze für den gesamten Raum aller der Szenarien anwendbar.

3.4 Fazit

Aus dem beschriebenen und diskutierten Stand der Wissenschaft geht hervor, dass im Kontext der Szenarienanalyse in Realdaten eine Forschungslücke besteht: die Methoden nutzen häufig eine Modellierung von Manövern oder Szenarien. Durch die fehlende Möglichkeit, Modelle für den offenen Kontext zu definieren und zu validieren, ist dieser Weg allein nicht zielführend (vgl. Abschnitt 3.3.1). Ansätze aus dem Bereich des Clusterings, die allein anhand der Daten eine abgeschlossene Liste von Fahrsequenz-Gruppen erstellen und daraus Szenarien ableiten vermeiden dieses Problem. Diese Ansätze konzentrieren sich jedoch meist auf einen sehr beschränkten Szenario-Raum, z.B. durch Fokus auf sehr strukturierte ODDs wie die Autobahn. Im Kontext einer urbanen ODD von Fahrfunktionen sind diese Ansätze jedoch nicht anwendbar. Um einen von Experten basierten Szenariokatalog zu ergänzen und die Ziele der datenbasierten Szenarienanalyse (vgl. Abschnitt 3.1.2) zu erfüllen, werden demnach neue Methoden benötigt.

Neben der Erfüllung der formulierten Ziele muss eine Methode auch so entworfen sein, dass sie im realen Entwicklungs- und Validierungsprozess von automatisierten Fahrfunktionen angewendet werden kann. Somit ergeben sich die in Kapitel 1 bereits formulierten Forschungsfragen.

4 Konzept und Entwurf einer Methode für datenbasierte Szenarienanalyse

4.1 Prozess für Datensammlung und -verarbeitung

4.1.1 Randbedingungen

Die aktuellen deutschen Gesetzgebung (vgl. Abschnitt 2.1.2) gibt vor, dass für eine automatisierte Fahrfunktion sowie ihren geplanten Betriebsbereich nachgewiesen werden muss, dass das Risiko einer Beeinträchtigung für Leib und Leben von Personen durch die Fahrfunktion nicht erhöht wird. Des Weiteren wird vorgeschrieben, dass dieser Nachweis auf Szenarien basierende Testfälle beinhalten und die Szenarienauswahl anhand empirischer Daten begründet werden muss. Somit schreibt die Gesetzgebung die Datensammlung im Betriebsbereich indirekt vor.

Zur Erfüllung der Vorschriften wird ein Prozess benötigt, der die Datensammlung und -verarbeitung beschreibt. Dieser Prozess ist eine Voraussetzung für den szenariobasierten Sicherheitsnachweis einer automatisierten Fahrfunktion in Kombination mit dem Betriebsbereich.

Zur Definition der Szenarien müssen grundsätzlich alle Aspekte im Schichtenmodell (vgl. Abschnitt 2.3.2) betrachtet werden. Durch den bekannten Betriebsbereich sind die statischen Aspekte wie die Straßen- und Verkehrsinfrastruktur

der möglichen Szenarien bereits bekannt (Schicht 1 und 2 im Schichten-Modell). Dies gilt ebenso für die Verfügbarkeit von digitalen Informationen, beispielsweise das Vorhandensein von intelligenter Infrastruktur (Schicht 6).

Eine weitere Gesetzesvorgabe ist die Einrichtung einer technischen Aufsicht. Diese kann bei temporärer Veränderung der statischen Umgebung (Schicht 3), beispielsweise durch Baustellen, eingreifen und einen sicheren Pfad für das automatisierte Fahrzeug vorgeben. Ebenso kann die technische Aufsicht bei herausfordernden Wetterverhältnissen (Schicht 5) eingreifen und das Fahrzeug beispielsweise bei starkem Nebel stilllegen.

Dem gegenüber steht die Interaktion mit Verkehrsteilnehmern (Schicht 4). Durch die hohe Dynamik kann diese weder durch Einschränkung des Betriebsbereichs noch durch die technische Aufsicht sinnvoll begrenzt oder vorhergesehen werden. Daher muss der Fokus des szenariobasierten Sicherheitsnachweises auf der Betrachtung von Interaktionen mit Verkehrsteilnehmern gesetzt werden. Dies ist im Prozess zur Datensammlung und -verarbeitung zu berücksichtigen.

Die Nutzung von Daten zum Aufbau von Wissen wird im Prozess zur Wissensdetektion in Datensätzen (vgl. Abschnitt 2.4.1) beschrieben. Dieser Prozess gibt abstrakte Schritte vor, welche für das vorliegende Problem der datenbasierten Szenarioanalyse in urbanen Betriebsbereichen angepasst und erweitert wurden.

Automatisierte Fahrzeuge bestehen aus den drei Komponenten Perzeption, Planung und Aktorik (vgl. Anhang A.2). Der hier beschriebene Prozess bezieht sich auf die datenbasierte Szenarioanalyse für die Planungskomponente. Somit ist die Sensorik zur Umfeldwahrnehmung und die Aktorik zur Fahrzeugführung nicht berücksichtigt.

4.1.2 Analyse und Auswahl von Aufzeichnungsorten

Der erste Prozessschritt des KDD (vgl. Abschnitt 2.4.1) ist die Datenauswahl. Im vorliegenden Problem wurde dieser Schritt um die systematische Auswahl von Aufzeichnungsorten erweitert. Durch die Auswahl der Aufzeichnungsorte wird

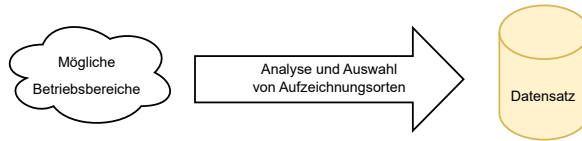


Abbildung 4.1: Darstellung der Artefakte im Schritt „Auswahl von Aufzeichnungsorten“.

der statische Aspekt der Szenarien adressiert. Zur kostengünstigen Parallelisierung der Aufzeichnung werden externe Datensätze (vgl. Def. 3.4) aufgezeichnet. Maximale Abdeckung wird erreicht, wenn im gesamten Betriebsbereich Daten aufgezeichnet werden. In der Praxis ist dieser Schritt aufwendig und kostenintensiv. Daher müssen die Abschnitte zur Aufzeichnung so ausgewählt werden, dass im Datensatz alle relevanten statischen Eigenschaften des Betriebsbereichs vorhanden sind. Beispielsweise sollten alle vorkommenden Typen von Straßenkreuzungen (z.B. T-Kreuzung, Kreuzung mit/ohne Lichtsignalanlage, Kreisverkehr) in den Daten vorhanden sein. Zusätzlich müssen Abschnitte mit erhöhter Unfallgefahr miteinbezogen werden. Diese können aus Unfallstatistiken extrahiert werden.

In den Aufzeichnungsorten wird dann Infrastruktur und Sensorik installiert, um Datensätze aufzuzeichnen. Dabei handelt es sich um externe Datensätze, die mit verschiedenen Technologien aufgezeichnet werden können. Grundsätzlich sind hier sowohl die stationäre Befestigung der Sensorik als auch die Nutzung von ausgerüsteten Drohnen möglich. Jedoch wird mit stationärer Befestigung die durchgehende Datenaufzeichnung ermöglicht, während bei Drohnen bedingt durch Ladeпаusen der Verkehr nicht vollständig aufgezeichnet wird. Daher ist die stationäre Installation der Sensorik zu bevorzugen.

Externe Datensätze haben gegenüber Ego-Datensätzen zwei Vorteile. Erstens kann die Bewegung von Verkehrsteilnehmern auf einem Segment vollständig aufgezeichnet werden, da sich Verkehrsteilnehmer nicht gegenseitig verdecken. Zweitens ist bei Aufnahme durch installierte Sensorik eine durchgängige Aufnahme über lange Zeiträume (z.B. ein Jahr) wirtschaftlich möglich, da nicht dauerhaft ein Testfahrer verfügbar sein muss.

Die initiale Verarbeitung der Rohdaten wird in diesem Schritt ebenso adressiert. Dabei kommen etablierte Algorithmen zur Objektdetektion und -Klassifikation sowie Objektverfolgung zum Einsatz (beschrieben in [14, 68]), sodass der resultierende Datensatz aus Verkehrsteilnehmern mit statischen Attributen (z.B. Teilnehmertyp) und dynamischen Trajektorien von Verkehrsteilnehmern besteht (vgl. (3.1) - (3.3)).

Zusätzlich zu den Trajektoriendaten müssen Umweltinformationen gewonnen werden, da sie wichtig für die Interpretation der im Prozess extrahierten Szenarien sind. Dabei handelt es sich um alle Wetter- (z.B. Schnee, Regen, Glätte, Nebel) und Umweltbedingungen (z.B. Baustellen, Sperrungen), die Einfluss auf den Verkehr haben können. Diese müssen jedoch nicht zwingend vor Ort aufgezeichnet werden, sondern können auch nachträglich angereichert werden.

4.1.3 Vorverarbeitung und Segmentierung der Daten

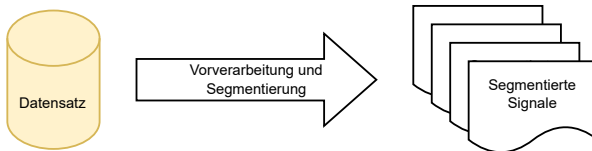


Abbildung 4.2: Darstellung der Artefakte im Schritt „Segmentierung der Daten“.

Der KDD-Schritt „Vorverarbeitung“ umfasst die Aufbereitung der Daten für die folgenden Verarbeitungsschritte mittels Standard-Verfahren, z.B. das Entfernen von Rauschen. Im vorliegenden Problem muss der Schritt um eine Segmentierung der Daten anhand von Betriebsbereichs-Segmenten erweitert werden.

Bei der Aufzeichnung der Daten können Fehler auftreten, beispielsweise durch Ausfälle von Sensoren, fehlerhafte Kalibrierungen oder Speicherfehlern. Durch eine Überwachung der Sensorzustände bei der Aufzeichnung sowie einer Prüfung

der Plausibilität der aufgezeichneten Daten werden die fehlerhaften Daten herausgefiltert. Diese Plausibilitätschecks beinhalten die Überprüfung physikalischer Gesetze und umfassen

- Stetigkeit der Position in x- und y-Richtung,
- Konsistenz der Geschwindigkeit mit der Positionsänderung,
- unplausible Beträge der Geschwindigkeit in Abhängigkeit von Teilnehmer-typ (z.B. Fußgänger mit $v = 80\text{km/h}$).

Die Zerlegung der Daten anhand von Betriebsbereichs-Segmenten ist ein notwendiger Schritt zur Gewinnung von Fahrsequenzen (vgl. Definition 3.6). Der Betriebsbereich wird in statische Segmente zerlegt. Die triviale Lösung für die Segmentierung ist durch die Wahl der Aufzeichnungsorte gegeben: jeder Streckenabschnitt, auf dem lückenlos aufgezeichnet wird, wird als Segment definiert. Im Fall von längeren, lückenlos aufgezeichneten Abschnitten werden diese zerlegt. Die Zerlegung erfolgt anhand der Szenerie (vgl. Definition 2.22), also den Aspekten der ersten drei Schichten im Szenario-Schichtenmodell (vgl. Abschnitt 2.3.2). Dabei werden die Segmente so gewählt, dass innerhalb eines Segments der Kontext konstant ist. Dies resultiert beispielsweise in der Segmentierung von Kreisverkehren, Kreuzungen, geraden Abschnitten oder Kurven.

4.1.4 Transformation in Fahrsequenzen

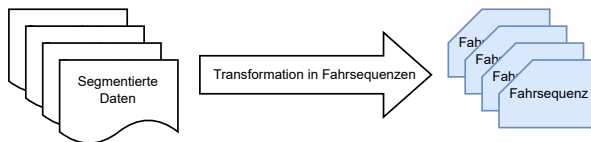


Abbildung 4.3: Darstellung der Artefakte im Schritt „Transformation in Fahrsequenzen“.

Der KDD-Prozess beschreibt in diesem Schritt die Transformation der Daten, beispielsweise durch Methoden der Dimensionsreduktion. Dieser Schritt wird im vorliegenden Problem um die Auswahl geeigneter Ego-Fahrzeuge erweitert,

um Fahrsequenzen (vgl. Definition 3.6) zu erhalten. Bei einem externen Datensatz (vgl. Definition 3.4) sind alle Fahrzeuge im Datensatz gleich repräsentiert und das Ego-Fahrzeug einer Fahrsequenz muss definiert werden. Dazu werden aufgezeichnete Verkehrsteilnehmer selektiert und als Ego interpretiert. Das Ziel ist die Analyse von Szenarien, die für die automatisierte Fahrfunktion relevant sind. Daher werden die Fahrzeuge ausgewählt, deren Verhalten auch für die automatisierte Fahrfunktion realistisch ist. Da Fußgänger und Fahrradfahrer andere Bewegungseigenschaften besitzen als Autos, werden diese nicht als Ego selektiert.

Nachdem ein Ego-Fahrzeug selektiert wurde, werden die Verkehrsteilnehmer, die im gleichen Zeitraum wie das selektierte Ego-Fahrzeug in den Daten vorhanden waren, als Teilnehmer der Fahrsequenz definiert. Für alle Teilnehmer wird dann die relative Bewegung zum selektierten Ego-Fahrzeug mit einer Transformation ins Fahrzeugkoordinatensystem (vgl. Abschnitt A.4.3) berechnet.

Abhängig von der Größe des Segments kann es vorkommen, dass einzelne Verkehrsteilnehmer offensichtlich nicht relevant für das Verhalten des Ego-Fahrzeugs sind. Ein Beispiel hierfür sind Verkehrsteilnehmer, die sehr weit hinter dem Ego-Fahrzeug fahren. Diese Verkehrsteilnehmer werden durch die Definition eines Relevanzbereichs aus den Fahrsequenzen herausgefiltert. Der Relevanzbereich ist im Koordinatensystem des Ego-Fahrzeugs als maximal erlaubte Distanz definiert. Alle Verkehrsteilnehmer, die sich über die Fahrsequenz hinweg nicht in diesem Relevanzbereich befinden, werden in der Fahrsequenz nicht berücksichtigt.

Die Bewegung des Ego-Fahrzeugs in globalen Koordinaten sowie die Bewegung der relevanten Teilnehmer in relativen Koordinaten wird dann als Fahrsequenz abgespeichert.

4.1.5 Fahrsequenz-Clustering

Dieser Schritt entspricht der Mustersuche des KDD-Prozesses. Hier konnte aufgrund der Datenstruktur und der Anforderungen (vgl. Abschnitt 4.2.1) kein vorhandenes Clustering-Verfahren (vgl. Abschnitt A.4.1) angewendet werden. Daher

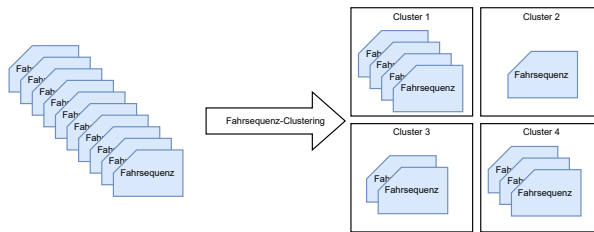


Abbildung 4.4: Darstellung der Artefakte im Schritt „Fahrsequenz-Clustering“.

wurde ein neues Verfahren entwickelt, das in Abschnitt 4.2 umfassend beschrieben ist. Das Ziel des Fahrsequenz-Clustering ist, die Menge der Fahrsequenzen in Gruppen von Fahrsequenzen aufzuteilen, die in jeder Fahrsequenz das gleiche Verhalten der Ego-Fahrzeuge sowie der Teilnehmer aufweisen. Pro Segment wird ein eigenes Fahrsequenz-Clustering durchgeführt. Demnach liegt nach diesem Schritt auf jedem Segment eine vollständige Gruppierung aller vorhandenen Fahrsequenzen vor. Jede Fahrsequenz ist genau einer dieser Gruppen zugeordnet und jede Gruppen besteht aus mindestens einer Fahrsequenz.

4.1.6 Überführung in nutzbare Szenarioformate

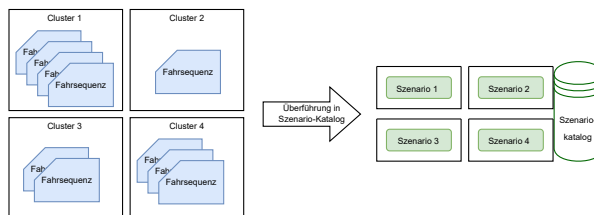


Abbildung 4.5: Darstellung der Artefakte im Schritt „Überführung in Szenario-Katalog“.

Die Gruppen von Fahrsequenzen mit gleichem Verhalten des Ego-Fahrzeugs und der Teilnehmer beinhalten eine implizite Beschreibung der darin enthalten

Szenarien. Im weiteren Entwicklungs- und Testprozess können diese impliziten Beschreibungen nicht direkt genutzt werden, da beispielsweise Simulationswerkzeuge ihre eigenen spezifischen Dateiformate haben. Daher müssen die Fahrsequenz-Gruppen in nutzbare Szenario-Formate konvertiert werden.

Anhand der vorliegenden Daten werden die notwendigen Szenarioinformationen (vgl. Abschnitt 2.3.2) gewonnen. Die Schichten Straßentopologie (Schicht 1), Verkehrsinfrastruktur (Schicht 2) und Digitale Informationen (Schicht 6) werden durch eine Kartierung der Umgebung der Messung gewonnen. Sollte es im Aufzeichnungszeitraum temporäre Veränderungen wie Straßensperrungen oder Baustellen (Schicht 3) gegeben haben, müssen diese durch enge Kommunikation mit Baubehörden identifiziert werden. Bei besonders schwere Veränderungen (bspw. Vollsperrung) werden die in diesen Zeiträumen gesammelten Daten nicht weiter genutzt. Die Umweltbedingungen werden durch die lokale Aufzeichnung von Wetterdaten oder durch die Nutzung von Wetterdatenbanken ermittelt. Für die Beschreibung des in den Fahrsequenzen vorliegenden Verhaltens der dynamischen Objekte (Schicht 4) existieren verschiedene Möglichkeiten: Mithilfe von Ontologien können abstrakte Szenarien aus den Gruppen abgeleitet werden [115]. Diese liegen in einer sprachlichen Beschreibung vor und können früh im Entwicklungsprozess zur Definition von Anforderungen verwendet werden. Zur Durchführung von simulativen Tests müssen die Fahrsequenz-Gruppen in logische Szenarien überführt werden. Dazu existieren Methoden, die ausführbare Szenariobeschreibungen aus den Fahrsequenzen ableiten können [48] (vgl. Abschnitt 2.3.3). Das Zielformat ist hier abhängig von den genutzten Simulationswerkzeugen.

Die konkrete Methode für die Umwandlung in nutzbare Szenarioformate liegt nicht im Fokus dieser Dissertation. Für die breite Nutzung der Szenarien im Entwicklungsprozess (vgl. Abschnitt 4.1.7) sollten die Szenarien in allen Abstraktionsleveln 2.3.2 transformiert werden können.

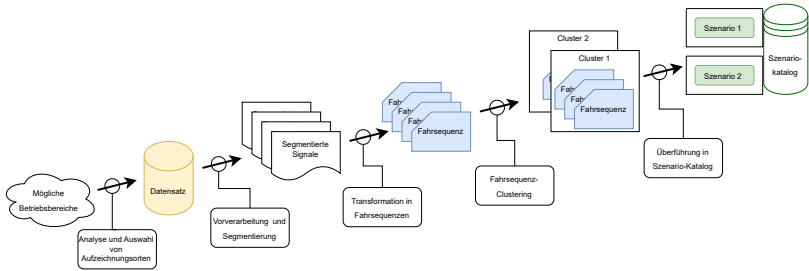


Abbildung 4.6: Vereinigung der in Abschnitt 4.1 dargestellten Schritte zur Datenaufzeichnung und -verarbeitung, angelehnt an den KDD (vgl. Abschnitt 2.4.1)

4.1.7 Verknüpfung mit Funktionsentwicklung

Einen Überblick über den gesamten Prozess zur Datenaufzeichnung und -verarbeitung gibt Abbildung 4.6. Dieser Prozess wird parallel zur Funktionsentwicklung durchgeführt und die resultierenden Szenarien in den verschiedenen Stufen werden entwicklungsbegleitend sowie zur Sicherheitsargumentation gegenüber dem Kraftfahrt-Bundesamt (KBA) genutzt. Zur Maximierung der Aufzeichnungsdauer muss die Datenaufzeichnung so früh wie möglich im Entwicklungsprozess gestartet werden. Daher sollte zu Beginn des Entwicklungsprozess mit der Datenaufzeichnung innerhalb der zukünftigen ODD, idealerweise innerhalb der geplanten Betriebsbereiche (vgl. Definition 2.2) begonnen werden. Zu Beginn der Entwicklung ist keine ausführbare automatisierte Fahrfunktion verfügbar, die zur Analyse von relevanten Szenarien genutzt werden könnte. Daher müssen Methoden zur datenbasierten Szenarienanalyse unabhängig von einer konkreten automatisierten Fahrfunktion arbeiten können.

4.2 Szenarienanalyse durch Fahrsequenz-Clustering

4.2.1 Anforderungen an Methode

Aus dem diskutierten Stand der Wissenschaft (vgl. Abschnitt 3.3) leiten sich die Anforderungen an eine Methode zu datenbasierten Szenarienanalyse ab. Aus den Schwächen des manöverbasierten Ansatzes (vgl. Abschnitt 3.3.1) ergibt sich die Forderung, dass eine Methode nicht auf einer Modellierung von Szenarien und Manövern beruhen sollte.

(A₁) Die Methode zur datenbasierten Szenarienanalyse soll keine Modellierung von Szenarien oder Manövern verwenden.

Um als vollständig zu gelten, muss die Methode geeignet sein, jedes möglicherweise auftretende Szenario in der urbanen ODD verarbeiten zu können. Insbesondere die Vielfalt der möglichen Verkehrsteilnehmer muss berücksichtigt werden.

(A₂) Die Methode zur datenbasierten Szenarienanalyse muss mit verschiedenen Typen und Anzahlen von Verkehrsteilnehmern geeignet umgehen können.

Des Weiteren haben Szenarien in Realdaten eine variable Dauer [32]. Eine Methode zur datenbasierten Szenarienanalyse darf also nicht nur Fahrsequenzen mit einheitlicher Dauer verarbeiten können, da ansonsten zusammengehörige Szenarien beschnitten werden könnten.

(A₃) Die Methode zur datenbasierten Szenarienanalyse muss mit Fahrsequenzen variabler Dauer umgehen können.

Um durch die datenbasierte Szenarienanalyse die Vollständigkeit der resultierenden Szenarien argumentieren zu können, muss ein Datensatz verschiedene Aspekte beinhalten. Im Winter sind andere Szenarien zu erwarten als im Sommer (beispielsweise durch die Anzahl Fahrradfahrer) sowie bei Tag andere Szenarien als bei Nacht. Somit muss sich der Datensatz über alle Jahreszeiten sowie alle

Tageszeiten erstrecken. Die zu erwartende Menge an Daten kann durch folgende Überlegung abgeschätzt werden: Bei der Aufzeichnung mit einer statischen Aufzeichnungsinfrastruktur im urbanen Bereich werden für ein Jahr durchgängig die Trajektorien der Verkehrsteilnehmer aufgezeichnet. Aus Verkehrszählungen ist bekannt, dass pro Tag zwischen 19700 und 44500 Fahrzeuge über eine viel befahrene urbane Kreuzung fahren¹. Pro Jahr fahren also zwischen 7,2 Mio. und 16,2 Mio. Fahrzeuge über diese Kreuzung. Jedes Fahrzeug, das die Kreuzung überquert bedeutet potentiell ein neues Szenario und muss daher von der Szenarienanalyse betrachtet werden.

Diese Zahlen ergeben sich für eine einzige Kreuzung, für reale Betriebsbereiche mit mehreren, unterschiedlichen Fahrtabschnitten sind um ein Vielfaches höhere Zahlen zu erwarten. Um mit den zu erwartenden großen Datenmengen umgehen zu können, muss die zu entwickelnde Methode effizient sein.

- (A₄) Die Rechendauer einer Analyse muss performant genug sein, um auch bei großen Datensätzen (bis zu 16,2 Mio. Fahrsequenzen pro Kreuzung) den Entwicklungsprozess nicht zu verzögern.

4.2.2 Grundidee

Der im folgenden vorgestellten Methode² zur datenbasierten Szenarienanalyse liegt folgendes Gedankenexperiment zugrunde: Ein Autofahrer, der absolut kein Wissen über die Welt hat (hier: die ODD, vgl. Definition 2.1), setzt sich in sein Fahrzeug und beginnt eine Fahrt. Zu Beginn seiner Fahrt ist alles, was ihm im Straßenverkehr begegnet, neu für ihn. Mit der Zeit werden jedoch immer mehr Ereignisse auftreten, die ihm bereits bekannt sind. Ein Ereignis ist beispielsweise eine normale Folgefahrt, ein die Straße kreuzender Fußgänger, eine Kreuzung mit Ampelanlagen oder ein zu überholender Fahrradfahrer³. Jedes erlebte Ereignis

¹ Quelle: Landesanstalt für Umwelt Baden-Württemberg, Verkehrszählungen an Kreuzungen in Freiburg, Karlsruhe und Stuttgart. Mittelwerte für 2021, abgerufen 08.11.2022

² Das Fahrsequenz-Clustering wurde in Teilen bereits in [LR7] veröffentlicht.

³ Oder jede andere, im Straßenverkehr vorkommende Situation.

vergleicht er mit allen ihm bekannten Ereignissen und entscheidet, ob er diese Fahrsequenz bereits kennt oder nicht. Mit fortschreitender Fahrdauer und zunehmender Erfahrung des Fahrers wird die Anzahl der für ihn neuen Ereignisse abnehmen.

Übertragen auf die datenbasierte Szenarienanalyse und die in dieser Dissertation verwendete Nomenklatur entspricht ein Ereignis einer Fahrsequenz. Jedes Ereignis bzw. jede Gruppe von gleichartigen Ereignissen wird einem Szenario zugeordnet (vgl. Abschnitt 3.1.4). Um entscheiden zu können, ob eine neue beobachtete Fahrsequenz bereits bekannt ist, muss sie in der Lage sein, neue Fahrsequenzen mit den bereits bekannten Fahrsequenzen zu vergleichen. Entsprechend Anforderung (A_1) soll für diesen Vergleich keine Modellierung von Szenarien verwendet werden. Daher wird ein Clustering-Ansatz gewählt. Zur Anwendung klassischer Clustering-Ansätze ist eine Distanzfunktion zwischen Fahrsequenzen notwendig, die jedoch nicht allgemeingültig definiert werden kann (vgl. Abschnitt 3.3.2). Die Einordnung von Fahrsequenzen auf einer Ordinalskala ist also nicht möglich. Daher greift die hier vorgestellte Methode auf einem niedrigeren Skalenniveau, der Nominalskala, ein. Anstatt eine Distanzfunktion über Fahrsequenzen zu definieren (vgl. Gleichung (3.10)), genügt eine binäre Funktion, welche bestimmt, ob zwei Fahrsequenzen ähnlich sind:

$$\theta(S_i, S_j) \rightarrow \{0, 1\}, \quad i, j \in \mathbb{N}, \quad (4.1)$$

Mit dieser Funktion werden Fahrsequenzen bezüglich des Verhaltens der Interaktionen von Verkehrsteilnehmern verglichen. Sie ist ein zentraler Baustein der Methode zur vollständigen Szenarienanalyse (vgl. Abschnitt 4.2.4). Der Fahrsequenz-Vergleich wird in eine effiziente Vorgehensweise zur kontinuierlichen Einteilung der Fahrsequenzen in Gruppen eingebettet (vgl. Abschnitt 4.2.3). Diese nutzt den Fahrsequenz-Vergleich und definiert, zu welchem Zeitpunkt welche Fahrsequenzen verglichen werden.

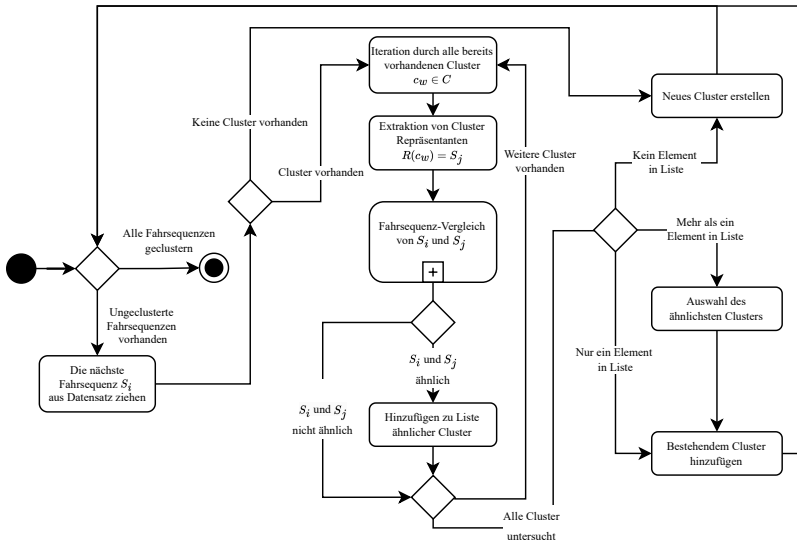


Abbildung 4.7: Diagramm für den Ablauf des kontinuierlichen Clusterings.

4.2.3 Algorithmus zum Fahrsequenz-Clustering

Der Algorithmus für das Fahrsequenz-Clustering spiegelt das beschriebene Gedankenexperiment des Fahrers, der kontinuierlich die Welt entdeckt (vgl. Abschnitt 4.2.2), wider. Das Vorgehen ist in Abbildung 4.7 dargestellt. Ausgangspunkt ist der in Fahrsequenzen segmentierte und transformierte Datensatz. Über diese Fahrsequenzen soll ein Clustering $C = \{c_1, c_2, \dots, c_w, \dots, c_W\}$ mit $1 \leq w \leq W$ gebildet werden. Dabei ist die Anzahl der Fahrsequenzen pro Gruppe c_w variabel. Zusätzlich wird eine Repräsentanten-Funktion $R(c_w) = S_i$ definiert, welche eine repräsentative Fahrsequenz für jedes Cluster beschreibt. Die repräsentative Fahrsequenz ist immer die erste Fahrsequenz eines neuen Clusters. Die Definition einer repräsentativen Fahrsequenz verringert den notwendigen Rechenaufwand, da neue Fahrsequenzen nur mit der repräsentativen Fahrsequenz verglichen werden müssen und nicht mit allen Fahrsequenzen im Cluster. Jedes Cluster muss daher zu jedem Zeitpunkt eine repräsentative Fahrsequenz besitzen.

Die Fahrsequenzen im Datensatz werden dann konsekutiv verarbeitet. Zu Beginn des Clusteringvorgangs existiert noch kein Cluster und es gilt $C = \emptyset$. Die erste Fahrsequenz S_1 wird daher unvermeidlich ein neues Cluster eröffnen, sodass

$$c_1 \leftarrow \{S_1\} \quad R(c_1) \leftarrow S_1 \quad (4.2)$$

definiert wird. Für jede darauf folgende Fahrsequenz S_i wird verglichen, ob es ein Cluster c_w gibt, deren Repräsentant $R(c_w)$ ähnlich zu S_i ist (vgl. Abschnitt 4.2.4). Dazu wird über alle bereits bestehenden Cluster $c_w \in C$ iteriert und alle Cluster mit ähnlichen Repräsentanten werden in einer Liste gespeichert. Sobald durch alle Cluster-Repräsentanten iteriert wurde, wird die resultierende Liste analysiert. Dabei treten drei Fälle auf: im ersten Fall ist die Liste leer, es existiert also kein ähnliches Cluster. Somit wird die Fahrsequenz S_i ein neues Cluster eröffnen. Im zweiten Fall ist exakt ein ähnliches Cluster in der Liste. Somit wird S_i diesem Cluster hinzugefügt:

$$c_w \leftarrow c_w \cup \{S_i\}. \quad (4.3)$$

Im dritten Fall besteht die Liste aus mehr als einem Cluster. In diesem Fall wird der vom Fahrsequenz-Vergleich bestimmte Ähnlichkeitsgrad (vgl. Abschnitt 4.2.4) verwendet, um das am besten passende Cluster zu bestimmen.

Nachdem S_i entweder ein neues Cluster eröffnet hat oder einem bestehenden zugewiesen wurde, wird die nächste Fahrsequenz S_{i+1} gezogen und der die beschriebenen Schritte starten von neuem. Dieser Vorgang wird wiederholt, solange es noch Fahrsequenzen im Datensatz gibt, welche noch nicht geclustert wurden. Ist der Datensatz vollständig geclustert, bricht der Algorithmus ab. Wird der Datensatz vergrößert, kann das Clustering an derselben Stelle wieder fortgesetzt werden und alle vorher bestimmten Ergebnisse weiter verwenden.

4.2.4 Fahrsequenz-Vergleich

Der Fokus des Fahrsequenz-Vergleichs liegt auf der Interaktion von Teilnehmern (vgl. Abschnitt 4.1.1). Es werden demnach die Typen der Teilnehmer (vgl. Definition 3.5) und ihre jeweiligen Bewegungen betrachtet. Zum Vergleich der Fahrsequenzen und der Entscheidung, ob sie als ähnlich aufgefasst werden können, werden drei Bedingungen geprüft, die durch folgende Funktionen beschrieben werden:

1. Ähnlichkeit der Teilnehmertypen: $\theta_{\text{typ}}(S_i, S_j)$. Hier genügt der Vergleich, ob dieselben Teilnehmertypen in einer Fahrsequenz vorhanden sind.
2. Ähnlichkeit des Ego-Verhaltens: $\theta_{\text{ego}}(S_i, S_j)$. Durch ein Zeitreihen-Distanzmaß wird die Bewegung der Ego-Fahrzeuge in den Fahrsequenzen S_i und S_j analysiert und auf Ähnlichkeit überprüft.
3. Ähnlichkeit des Teilnehmer-Verhaltens: $\theta_{\text{teil}}(S_i, S_j)$. Analog zu den Ego-Fahrzeugen wird die Ähnlichkeit des Teilnehmerverhaltens durch einen Vergleich von Zeitreihen bestimmt. Zusätzlich wird die variable Anzahl von Teilnehmern durch eine Vergleichs-Strategie berücksichtigt, sodass Teilnehmer aus S_i und S_j mit ähnlichem Verhalten detektiert werden.

Nur wenn alle drei Funktionen den Ausgabewert 1 haben, werden die beiden Fahrsequenzen S_i und S_j als ähnlich bezeichnet. Dementsprechend wird $\theta(S_i, S_j)$ aus Gleichung 4.1 als

$$\theta(S_i, S_j) = \theta_{\text{typ}}(S_i, S_j) * \theta_{\text{ego}}(S_i, S_j) * \theta_{\text{teil}}(S_i, S_j) \quad (4.4)$$

geschrieben. Die einzelnen Werte für $\theta_{\text{typ}}(S_i, S_j)$, $\theta_{\text{ego}}(S_i, S_j)$ und $\theta_{\text{teil}}(S_i, S_j)$ werden in der angegebenen Reihenfolge bestimmt, beginnend mit dem Schritt mit geringstem Rechenaufwand (vgl. Abbildung 4.8). Sobald eine Funktion 0 ergibt (entspricht „Teilaspekt nicht ähnlich“), werden die dahinter liegenden Schritte nicht ausgeführt, was die Rechenzeit verringert (vgl. Anforderung (A_4)).

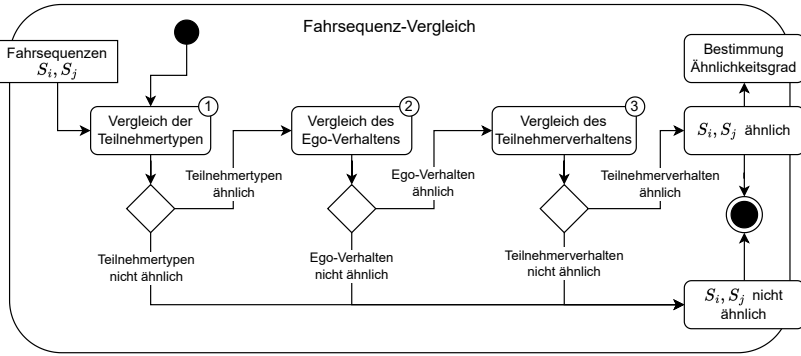


Abbildung 4.8: Ablaufdiagramm für den Fahrsequenz-Vergleich.

Vergleich der Teilnehmertypen

Diesem Vergleich liegt die Annahme zugrunde, dass nur ähnliche Teilnehmertypen (vgl. Definition 3.5) ähnliches Verhalten aufweisen können. Beispielsweise kann ein sich Fahrradfahrer nicht ähnlich zu einem Fahrzeug verhalten, sondern nur zu anderen Fahrradfahrern. Demnach können auch nur zwei Fahrsequenzen ähnlich sein, die dieselben Teilnehmertypen aufweisen. Für den Vergleich der Teilnehmertypen werden alle Teilnehmer einer Fahrsequenz betrachtet. Die einzelnen Teilnehmertypen sind in A_i (vgl. Gleichung (3.2)) codiert. Für jedes S_i kann ein Tupel $\mathbf{o}_i \in \mathbb{N}^{1 \times |O|}$ mit der Anzahl der Teilnehmer eines spezifischen Typs aufgebaut werden, wobei $|O|$ die Anzahl möglichen Teilnehmertypen beschreibt. Dieses Tupel wird im folgenden als Teilnehmertyp-Kombination bezeichnet. Wenn $O = (\text{Auto}, \text{Fußgänger}, \text{Fahrradfahrer}, \text{Truck})$ und in einer Fahrsequenz ausschließlich zwei Autos und ein Fußgänger vorhanden sind, ergibt sich $\mathbf{o}_i = (2, 2, 0, 0)$.

Der Vergleich der Teilnehmertyp-Konfigurationen wird also mit

$$\theta_{\text{typ}}(S_i, S_j) = \begin{cases} 1 & \text{wenn } \mathbf{o}_i = \mathbf{o}_j \\ 0 & \text{sonst} \end{cases} \quad (4.5)$$

durchgeführt. Da dieser Vergleich nicht direkt auf die Zeitreihen zugreift, sondern nur auf Objektattribute, hat er einen geringeren Rechenaufwand als die darauf folgenden Schritte. Im Fall von $\theta_{\text{typ}}(S_i, S_j) = 0$ müssen die darauf folgenden Schritte nicht ausgeführt werden.

Auswahl der Zeitreihen-Distanzfunktion

Die im Verfahren folgenden Schritte „Ähnlichkeit des Ego-Verhaltens“ sowie „Ähnlichkeit des Teilnehmer-Verhaltens“ basieren auf dem direkten Vergleich von Zeitreihen mit einer Distanzfunktion für Zeitreihen. An diese Distanzfunktion (im folgenden ρ genannt) gilt die zentrale Anforderung: sie soll die semantische Ähnlichkeit der Trajektorien bewerten, also bei semantisch ähnlichen Trajektorien geringere Distanzen bestimmen als bei unterschiedlichen Trajektorien. „Semantische Ähnlichkeit“ von zwei Trajektorien bedeutet in dem Kontext, dass die beiden zugehörigen Fahrzeuge durch ihr Verhalten demselben logischen Szenario zugeordnet werden können. Beispielsweise sollen zwei Fahrzeuge, die an einer gegebenen Kreuzung links abbiegen, als ähnlich bezeichnet werden. Die durch ρ bestimmte Distanz ihrer Trajektorien muss demnach minimal sein.

Die gemessenen Trajektorien von Fahrzeugen in realen Daten werden jedoch nicht in exakt gleichen Zeitreihen resultieren. Gründe dafür sind die Varianz verschiedener menschlicher Fahrweisen sowie das Rauschen des Messsystems. Ein Beispiel für die Varianz im Verhalten menschlicher Fahrer ist die Standzeit an einem Stoppschild. Diese ist individuell vom Fahrer abhängig. In Abbildung 4.9 sind zwei exemplarische Verläufe der x-Position⁴ für zwei verschiedene Fahrer an einem Stoppschild gegeben. Während der Fahrt in x-Richtung steigt der Wert konstant an, während des Wartens bleibt er konstant. Auch wenn die Dauer des Anhaltens und die Länge der Zeitreihe bei beiden Fahrzeugen nicht exakt gleich ist, sollte ein Zeitreihen-Distanzmaß für die x-Positionen hier in einer kleinen Distanz resultieren, da beide Verhalten trotz unterschiedliche Anhaltedauer demselben

⁴ Vereinfachte Darstellung. Trägheit beim Anhalte- und Anfahrvorgang ist hier nicht berücksichtigt.

Szenario zugehörig sind. Eine Distanzfunktion muss daher in der Lage sein, die Zeitreihen in zeitlicher Richtung zu strecken oder zu stauchen.

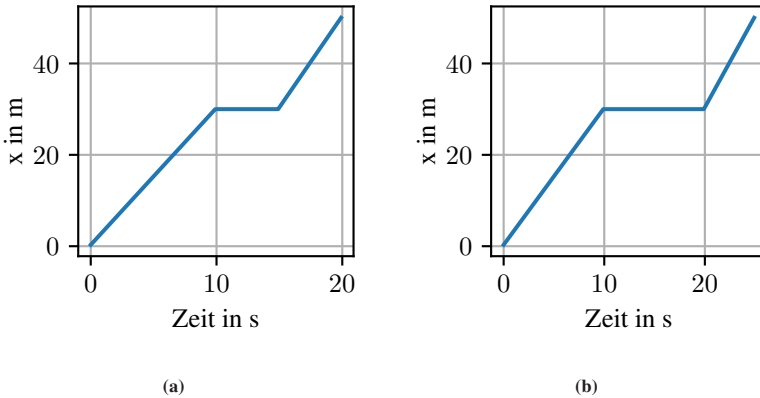


Abbildung 4.9: Beispiel für zwei ähnliche X-Werte einer Trajektorie $x_{\text{ego},1}$ und $x_{\text{ego},1}$.

Die Distanzfunktion muss zusätzlich mit der Tatsache umgehen, dass die Trajektorienlängen der verschiedenen Objekte im Allgemeinen unterschiedlich sind (vgl. Anforderung (A_3)), also auch Distanzen zwischen Zeitreihen mit unterschiedlich vielen Messpunkte berechnen.

Um die Zahl der einstellbaren Parameter im Fahrsequenz-Clustering zu minimieren, soll das Distanzmaß parameterfrei sein. Bekannte Distanzfunktionen für Zeitreihen in der Literatur sind die euklidische Distanz, die LCSS-Distanz oder das Dynamic Time Warping (DTW) (vgl. Abschnitt 2.4.2). In Tabelle 4.1 werden diese Funktionen in Bezug auf die formulierten Anforderungen verglichen: die euklidische Distanz vergleicht die Zeitreihen punktweise, wobei eine feste Zuordnung von Punkt zu Punkt notwendig ist. Somit können die Signale nicht gestreckt oder gestaucht werden. Zudem ist für den Vergleich unterschiedlich langer Zeitreihen eine vorherige Interpolation auf gleiche Signallänge notwendig. Sowohl die LCSS-Distanz als auch DTW erfüllen die Anforderungen. Zur Bestimmung ähnlicher Zeitreihen eignen sich beide Funktionen gleichermaßen, wie Vergleiche auf Basis von 38 Datensätzen zeigen konnten [121]. Bei der LCSS-Distanz ist jedoch

ein Parameter ϵ nötig, der manuell ausgewählt werden muss (vgl. Abschnitt 2.4.2), während DTW vollständig parameterfrei funktioniert. Aus diesem Grund wird im folgenden das Dynamic Time Warping als Distanzfunktion ρ verwendet.

Anforderung	Euklidische Distanz	LCSS-Distanz	DTW
Strecken und Stauchen von Zeitreihen	nicht möglich	möglich	möglich
unterschiedliche Längen von Zeitreihen	möglich (mit Interpolation)	möglich	möglich
Notwendige Parameter	keine	Distanzparameter ϵ	keine

Tabelle 4.1: Vergleich der verschiedenen Distanzfunktionen (vgl. Abschnitt 2.4.2) in Bezug auf Erfüllung der Anforderungen.

Vergleich des Ego-Verhaltens

In diesem Schritt wird das Verhalten der Ego-Fahrzeuge auf Ähnlichkeit überprüft. Für den direkten Vergleich der Ego-Fahrzeuge E werden die Trajektorien $\mathbf{X}_E^i, \mathbf{X}_E^j$ beider Ego-Fahrzeuge betrachtet. Die Trajektorien bestehen aus den Positionensverläufen im globalen Koordinatensystem:

$$\mathbf{X}_E^i = [\mathbf{x}^g, \mathbf{y}^g]^i, \quad \mathbf{X}_E^j = [\mathbf{x}^g, \mathbf{y}^g]^j. \quad (4.6)$$

Mit dem Zeitreihen-Distanzmaß ρ wird die Distanz für die Positionsverläufe

$$P_E = \rho(\mathbf{X}_E^i, \mathbf{X}_E^j) / \kappa_E \quad (4.7)$$

bestimmt. Bei einer exakt gleichen Bewegung zweier Objekte ergibt sich die Distanz $P_E = 0$. Da jedoch auch zwei Trajektorien mit leichten Unterschieden demselben Szenario zugeordnet werden können, wird ein Schwellwert $\Gamma_E, \theta_E(S_i, S_j) > 0$ eingeführt. Der Schwellwert ist so zu wählen, dass die Distanzen

ähnlicher Trajektorien (vgl. Abschnitt 4.2.4) unterhalb des Schwellwerts liegen. Das Ergebnis des Zeitreihendistanzmaßes $\rho = \text{DTW}$ ist abhängig von der Länge der beiden Trajektorien \mathbf{X}_E^i und \mathbf{X}_E^j . Damit ein fester Schwellwert unabhängig von der Trajektorienlänge verwendet werden kann, wird die bestimmte Distanz mit der Länge des Warping-Pfads κ_E (vgl. Abschnitt 2.4.2) normiert. Damit ergibt sich

$$\theta_E(S_i, S_j) = \begin{cases} 1 & \text{wenn } P_E < \Gamma_E \\ 0 & \text{sonst} \end{cases} \quad (4.8)$$

Durch den Schwellwert wird also die mittlere Abweichung zweier Trajektorien pro Warping-Schritt in Meter eingestellt. Somit stellt der Schwellwert einen Parameter der Vergleichsmethode dar, der anhand von Daten eingestellt werden muss.

Die Methodik des Vergleichs von zwei Trajektorien wird analog auch im Vergleich des Teilnehmerverhaltens angewandt. Der Vergleich der Ego-Fahrzeuge wird aus zwei Gründen separat und zuvor durchgeführt:

- Pro Fahrsequenz existiert nur ein Ego-Fahrzeug, somit ist eindeutig, welche Trajektorien verglichen werden müssen. Die variable Anzahl der Teilnehmer verlangt weitere Schritte.
- Der Vergleich der Teilnehmer ist für Fahrsequenzen mit mehr als einem Teilnehmer rechenaufwändiger als der Vergleich der Ego-Fahrzeuge. Sollten die Ego-Fahrzeuge bereits unterschiedliches Verhalten aufweisen, müssen die Teilnehmer nicht betrachtet werden, wodurch Rechenzeit gespart wird.

Vergleich des Teilnehmer-Verhaltens

Die dritte Bedingung befasst sich mit der Ähnlichkeit der Teilnehmer der Fahrsequenzen. Damit zwei Fahrsequenzen als ähnlich bezeichnet werden, muss für jeden Teilnehmer T_u^i in S_i ein passender Teilnehmer T_v^j in S_j existieren, der ein ausreichend ähnliches Verhalten aufweist. Analog muss für jeden Teilnehmer in S_j ein Gegenstück in S_i vorhanden sein. Eine zwingende Voraussetzung für Ähnlichkeit ist, dass die Teilnehmer denselben Teilnehmertyp haben. Die vorhandenen

Teilnehmertypen werden daher separat betrachtet und für jeden Teilnehmertyp beurteilt, ob Ähnlichkeit vorliegt (vgl. Anforderung (A_2)). Dass grundsätzlich in S_i und S_j dieselben Teilnehmertypen vorliegen, wird durch die vorherige Überprüfung von (4.5) sichergestellt. Der im folgenden beschriebene Algorithmus dient dazu zu ermitteln, ob für jeden Teilnehmer in einer Fahrsequenz ein Teilnehmer des gleichen Typs in der anderen Fahrsequenz existiert, dessen Trajektorie als ähnlich angesehen werden kann.

Dazu wird für jeden Teilnehmertyp o_k eine Distanzmatrix

$$\Psi_k = \begin{bmatrix} P_T^{1,1} & \dots & P_T^{1,v_k} & \dots & P_T^{1,V_k} \\ \vdots & \ddots & & & \vdots \\ P_T^{u_k,1} & \dots & P_T^{u_k,v_k} & \dots & P_T^{u_k,V_k} \\ \vdots & & & \ddots & \vdots \\ P_T^{U_k,1} & \dots & P_T^{U_k,v_k} & \dots & P_T^{U_k,V_k} \end{bmatrix} \quad (4.9)$$

berechnet. $P_{T,X}^{u_k,v_k}$ bezeichnet hier analog zu Gleichung (4.7) die Distanz zwischen den Positionen von zwei Teilnehmern mit

$$P_T^{u_k,v_k} = \rho(\mathbf{X}_T^{u_k}, \mathbf{X}_T^{v_k}) / \kappa_{u_o,v_o} \quad (4.10)$$

$$\mathbf{X}_T^{u_k} = [\mathbf{x}^e, \mathbf{y}^e]^{u_k}, \quad \mathbf{X}_T^{v_k} = [\mathbf{x}^e, \mathbf{y}^e]^{v_k}. \quad (4.11)$$

Die Teilnehmer des Typs o_k in S_i sind mit Index u_k , $1 \leq u_k \leq U_k$ und die Teilnehmer in S_j analog mit Index v_k , $1 \leq v_k \leq V_k$ bezeichnet. Entsprechend dem Vorgehen beim Ego-Fahrzeug wird auch hier das Distanzmaß DTW verwendet. Anstelle der absoluten Positionen werden hier die Positionen im Ego-Koordinatensystem e betrachtet, um die relative Bewegung zum Ego-Fahrzeug zu beschreiben.

Analog zum Vergleich des Ego-Fahrzeugs wird auch hier ein Schwellwert Γ_T verwendet um zu prüfen, ob die Distanz zwischen den Trajektorien zweier Teilnehmer ausreichend klein ist. Diese Überprüfung wird auf jedes Element in Ψ_k durch

$$\beta(P_T^{u_o, v_o}, \Gamma_T) = \begin{cases} 1 & \text{wenn } P_T^{u_o, v_o} \leq \Gamma_T \\ 0 & \text{sonst} \end{cases} \quad (4.12)$$

angewendet, sodass die Transformation

$$\Psi_k \in \mathbb{R}^{U_k \times V_k} \rightarrow \Psi_k^b \in \{0, 1\}^{U_k \times V_k} \quad (4.13)$$

durchgeführt wird. Jedes Element der Matrix Ψ_k^b ist entweder 1 oder 0, je nachdem ob die Distanz der Positionen der jeweiligen Teilnehmer kleiner als der Schwellwert ist oder nicht. Die Matrix Ψ_k^b beschreibt also für einen Teilnehmertyp o_k die Ähnlichkeiten zwischen den vorhandenen Teilnehmern, wobei die Teilnehmer in S_i in den Zeilen und die Teilnehmer von S_j in den Spalten dargestellt sind. Auf Basis von Ψ_k^b wird nun entschieden, inwiefern in Bezug auf alle Teilnehmer des Typs o eine Ähnlichkeit der Fahrsequenzen vorliegt.

Um anhand Ψ_k^b zu entscheiden, ob für jeden Teilnehmer ein passender Teilnehmer vorliegt, muss ein Zuordnungsproblem gelöst werden. Dazu wird eine Modifikation der ungarischen Methode⁵ verwendet. Die Matrix Ψ_k^b beschreibt, welche Teilnehmer aus S_i (Zeilen der Matrix) zu welchen Teilnehmern aus S_j (Spalten) passen. Eine Zuordnung ist nur zulässig, wenn der jeweilige Matrixeintrag eine 1 ist. Wenn zwei Teilnehmer einander zugeordnet wurden, kann keiner der Teilnehmer einem anderen zugeordnet werden. Die ungarische Methode wird so modifiziert, dass nur Einträge mit einer 1 eine erlaubte Zuordnung darstellen, während Einträge mit einer 0 unendlich hohe Kosten verursachen. Findet die modifizierte

⁵ Die „ungarische Methode“ ist ein Algorithmus zur Lösung von Zuordnungsproblemen auf bipartiten Graphen, welche durch quadratische Kostenmatrizen repräsentiert werden können. Mit ihrer Hilfe wird eine eindeutige Zuordnung gefunden, bei der die Kosten minimal sind. [15]

ungarische Methode eine gültige Lösung mit Kosten kleiner als unendlich, ist die Ähnlichkeit der Teilnehmer des jeweiligen Teilnehmertyps gegeben.

Die Durchführung der modifizierten ungarischen Methode wird durch $\gamma(\Psi_k^b) \rightarrow 0, 1$ beschrieben und ergibt eine 1, wenn eine gültige Zuordnung gefunden wurde und eine 0, wenn nicht. Diese wird dann für die Ψ_k^b aller Teilnehmertypen o_k ausgeführt. Zusammenfassend kann der Vergleich des Teilnehmer-Verhaltens durch

$$\theta_{\text{teil}}(S_i, S_j) = \begin{cases} 1 & \text{wenn } \gamma(\Psi_k^b) = 1 \text{ für alle } o_k \\ 0 & \text{sonst} \end{cases} \quad (4.14)$$

ausgedrückt werden.

Rechenbeispiel

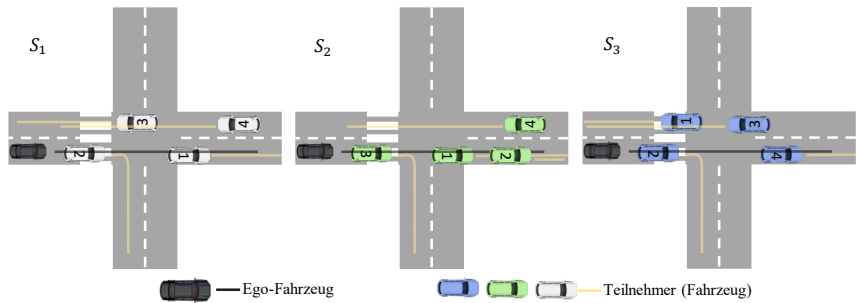


Abbildung 4.10: Drei Fahrsequenzen S_1, S_2, S_3 werden verglichen. Die Verkehrsteilnehmer sind immer an ihrem jeweiligen Startpunkt mitsamt ihrer darauf folgenden Bewegung dargestellt.

Das Vorgehen wird anhand eines Beispiels mit exemplarischen Werten verdeutlicht: Angenommen, die Fahrsequenzen S_1, S_2 beinhalten jeweils vier Teilnehmer vom Typ Auto (vgl. Abbildung 4.10). Da jeweils vier Teilnehmer des gleichen

Typs vorhanden sind, ist $\theta_{\text{typ}}(S_1, S_2) = 1$. Da die Ego-Fahrzeuge beide geradeaus über die Kreuzung fahren, ergibt sich eine niedrige DTW-Distanz beim Vergleich der Ego-Trajektorie: $P_E^{1,2} = 3$. Bei einem Schwellwert von $\Gamma_T = 5$ gilt somit $\theta_E(S_i, S_j) = 1$. Für den Vergleich der Teilnehmer wird mit Gleichung (4.10) für jedes Teilnehmerpaar die Distanz bestimmt und in die Distanzmatrix Ψ_{auto} eingetragen. In den Zeilen sind die Werte für die Teilnehmer aus S_1 und in den Spalten für S_2 . Die Werte selbst sind exemplarisch.

$$\Psi_{\text{auto}} = \begin{bmatrix} 3 & 3 & 20 & 50 \\ 20 & 20 & 2 & 50 \\ 50 & 50 & 50 & 2 \\ 50 & 50 & 50 & 2 \end{bmatrix}. \quad (4.15)$$

Unter Anwendung von Gleichung 4.9 mit Schwellwert $\Gamma_T = 10$ ergibt sich

$$\Psi_{\text{auto}}^b = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.16)$$

Als nächster Schritt wird untersucht, ob eine eindeutige und exklusive Zuordnung zwischen den Teilnehmern aus S_1 und S_2 existiert. Dazu wird die modifizierte ungarische Methode verwendet: sie bestimmt, ob in Ψ_{auto}^b jede Zeile exklusiv einer Spalte zugeordnet werden kann (und umgekehrt).

Im Beispiel liegt diese Zuordnung nicht vor, da z.B. die Teilnehmer 3 und 4 in S_1 beide nur zu Teilnehmer 4 aus S_2 zugeordnet werden müssten. Somit ist $\theta_{\text{teil}}(S_1, S_2) = 0$, woraus $\theta(S_1, S_2) = 0$ folgt. Demnach sind die beiden Fahrsequenzen S_1 und S_2 nicht ähnlich.

Als zweites Beispiel werden die Fahrsequenzen S_1 und S_3 verglichen. Auch hier sind die Teilnehmertypen ähnlich. Analog zum obigen Fall gilt für das Verhalten

der Ego-Fahrzeuge $P_E^{1,3} = 3$ und somit $\theta_E(S_i, S_j) = 1$. Für den Vergleich von S_1 und S_3 ergeben sich die Matrizen unter Anwendung von $\Gamma_T = 10$ zu

$$\Psi_{\text{auto}} = \begin{bmatrix} 50 & 20 & 50 & \mathbf{4} \\ 50 & \mathbf{2} & 50 & 20 \\ \mathbf{3} & 50 & 2 & 50 \\ 2 & 50 & \mathbf{2} & 50 \end{bmatrix}, \quad \Psi_{\text{auto}}^b = \begin{bmatrix} 0 & 0 & 0 & \mathbf{1} \\ 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & 1 & 0 \\ 1 & 0 & \mathbf{1} & 0 \end{bmatrix}. \quad (4.17)$$

Hier kann durch die modifizierte ungarische Methode eine eindeutige Zuordnung gefunden werden (siehe fett gedruckte Matrixeinträge). Somit ist Teilnehmer 1 aus S_1 ähnlich zu Teilnehmer 4 aus S_3 und Teilnehmer 2 aus S_1 zu Teilnehmer 2 aus S_3 .

Ähnlichkeitsgrad von Fahrsequenzen

Durch das Design des Algorithmus des Fahrsequenz Clusterings (vgl. Abschnitt 4.2.3) kann es passieren, dass für eine Fahrsequenz S_i mehr als nur ein passendes Cluster auftritt. Ein zu S_i passendes Cluster ist eines, für dessen repräsentative Fahrsequenz S_j sich $\theta(S_i, S_j) = 1$ ergibt. Um im Falle mehrerer passender Cluster zu ermitteln, welches Cluster am besten zu S_i passt, wird eine Funktion

$$\Lambda(S_i, S_j) \rightarrow \mathbb{R}, \text{ wenn } \theta(S_i, S_j) = 1 \quad (4.18)$$

für die Bestimmung des Ähnlichkeitsgrads definiert. Zu diesem Zweck werden die während des Fahrsequenz-Vergleichs bestimmten, relevanten Ergebnisse des Distanzmaßes zwischengespeichert und ein Mittelwert gebildet.

Betrachtet wird zum einen die Distanz der Ego-Fahrzeuge P_E (vgl. Gleichung (4.8)). Zusätzlich werden für jeden Objekttyp der Teilnehmer alle relevanten Distanzen der Objektmatrix (vgl. Gleichung (4.9)) gespeichert. Relevant sind die Matrixeinträge, die anhand der modifizierten ungarischen Methode ausgewählt wurden (vgl. fett gedruckte 1 in Gleichung (4.17)). Sie bilden die Zuordnung der ähnlichsten Teilnehmer ab.

Somit existiert eine Liste von Werten, die jeweils die Güte der Zuordnung und somit den Ähnlichkeitsgrad zwischen den Ego-Fahrzeugen bzw. den Teilnehmern beschreibt. Der Mittelwert über alle Werte in dieser Liste wird Ähnlichkeitsgrad zwischen zwei Fahrsequenzen genannt. Der Ähnlichkeitsgrad kann nur für zwei Fahrsequenzen bestimmt werden, bei denen die drei Ähnlichkeitskriterien (Teilnehmer-Typen, Ego-Verhalten und Teilnehmer-Verhalten) erfüllt sind.

Im oben beschriebenen Beispiel kann der Ähnlichkeitsgrad $\Lambda_{1,2}$ für den Vergleich von S_1 und S_3 bestimmt werden. Dazu wird der Mittelwert zwischen der Distanz der Ego-Trajektorien sowie den fett gedruckten Distanzen der Teilnehmer (vgl. Gleichung (4.17)) gebildet:

$$\Lambda(S_1, S_3) = (P_E + P_T^{1,4} + P_T^{2,2} + P_T^{3,1} + P_T^{4,3})/5 \quad (4.19)$$

$$= (3 + 4 + 2 + 3 + 2)/5 \quad (4.20)$$

$$= 2,8. \quad (4.21)$$

Für den Vergleich von S_1 und S_2 ist die Bestimmung des Ähnlichkeitsgrads nicht möglich, da diese Fahrsequenzen nicht als ähnlich identifiziert wurden.

4.3 Bestimmung von Szenario-Auftretenswahrscheinlichkeiten

Für die Sicherheitsargumentation gemäß den Standards der funktionalen Sicherheit (vgl. Abschnitt 2.2.2) und der Sicherheit der Sollfunktion (vgl. Abschnitt 2.3.4) muss schlüssig dargelegt werden, dass der Betrieb des Systems kein unzumutbares Risiko (vgl. Definition 2.10) für alle beteiligten Akteure darstellt. Entsprechend der Automotive Safety Integrity Level (ASIL) basiert die Bewertung und Kategorisierung potenzieller Risiken auf den Faktoren Controllability, Severity oder Exposure (vgl. Abschnitt 2.2.2). Der Prozess zur Datensammlung und -verarbeitung (vgl. Abschnitt 4.1) sowie das Fahrsequenz-Clustering (vgl. Abschnitt 4.2) ermöglichen eine durch Daten belegte Bestimmung der Auftretenswahrscheinlichkeit von Szenarien. Dabei wird die Größe der entstandenen

Fahrsequenz-Cluster genutzt, um eine Auftretenswahrscheinlichkeit des zugehörigen Szenarios zu schätzen (vgl. Abschnitt 4.3.1).

Grundsätzlich muss die Anzahl aller theoretisch denkbaren und physikalisch möglichen Szenarien als unendlich groß aufgefasst werden. Auch intuitiv sehr unwahrscheinliche Ereignisse können nicht vollkommen ausgeschlossen werden. Ereignisse wie auf der Straße notlandende Flugzeuge oder auf der Straße freilauende Giraffen können vorkommen. Durch das hier vorgestellte Konzept kann für solche sehr seltenen Ereignisse jedoch belastbar argumentiert werden, dass die Auftretenswahrscheinlichkeit klein genug ist um kein unzumutbares Risiko darzustellen. Das Problem der unendlich großen Zahl an Szenarien wird also reduziert in einen endlichen Raum von Szenarien, deren Wahrscheinlichkeit oberhalb einer definierten Schwelle ist. Der Standard ISO 26262 definiert verschiedenen Klassen von Auftretenswahrscheinlichkeiten (vgl. Abschnitt 2.2.2). Szenarien mit der Klassifikation E0 müssen für einen Sicherheitsnachweis nicht betrachtet werden. In den übrigen Klassifikationen E1-E4 richtet sich das resultierende Sicherheitslevel zusätzlich nach der Schadensschwere sowie der Kontrollierbarkeit. Diese Dimensionen müssen anhand von Systemanalysen und Tests ermittelt werden und sind nicht Teil dieses Konzepts.

4.3.1 Ableitung von Auftretenswahrscheinlichkeiten aus Cluster-Ergebnissen

Die nach dem Clustering vorliegenden Gruppen von Fahrsequenzen können genutzt werden, um Auftretenswahrscheinlichkeiten von Szenarien in einem Betriebsbereich zu schätzen.

Wahrscheinlichkeitstheoretisch kann die Fahrt im Betriebsbereich durch eine Abfolge von Versuchen repräsentiert werden. Bei einer Überfahrt über einen spezifischen Streckenabschnitt q_j (z.B. eine Kreuzung) wird in jedem Fall ein Szenario auftreten. Gesucht ist also die Wahrscheinlichkeit $p_{\Omega_w} = P(\Omega_w)$, dass ein spezifisches Szenario Ω_w im Betriebsbereich auftritt. Zum Szenario gehört die statische

Umgebung sowie das dynamische Verhalten der Teilnehmer. Die statische Umgebung ist durch das konkrete Segment q_j gegeben, daher gilt

$$p_{\Omega_w} = P(\Omega_w) = P(\Omega_w|q_j). \quad (4.22)$$

Das dynamische Verhalten wurde im Fahrsequenz-Clustering analysiert. Jedes Szenario ist hier einem Cluster c verknüpft. Die Anzahl der Fahrsequenzen in diesem Cluster ist $|c|$. Die Wahrscheinlichkeiten der Szenarien auf einem definierten Segment werden über eine Multinomialverteilung mit der Wahrscheinlichkeitsdichte

$$f(\Omega_1, \dots, \Omega_w) = \frac{K!}{\prod_{w=1}^W |c_w|!} \prod_{w=1}^W p_{\Omega_w} \quad (4.23)$$

modelliert [36], wobei W die Anzahl aller Cluster beschreibt und K die Summe aller Cluster-Größen $|c_1| + \dots + |c_W| = K$ und damit auch die Anzahl der Fahrsequenzen auf diesem Segment beschreibt. Der Erwartungswert für die Auftretenswahrscheinlichkeit eines Szenarios ergibt sich dann zu

$$E(|c_w|) = K \cdot p_{\Omega_w}. \quad (4.24)$$

Die wahre Auftretenswahrscheinlichkeit des Szenarios p_{Ω_w} ist unbekannt, wird aber anhand der aufgezeichneten Daten geschätzt. Die Schätzung kann mit

$$\hat{p}_{\Omega_w} = \frac{|c_w|}{K} \quad (4.25)$$

berechnet werden.

Die Wahrscheinlichkeit gibt eine Aussage darüber, mit welcher Häufigkeit ein bestimmtes Szenario im Betriebsbereich auf dem definierten Segment auftritt. Je größer der Datensatz, desto sicherer ist die Schätzung. Die Sicherheit der Schätzung äußert sich in der Größe des Konfidenzintervalls: kleine Konfidenzintervalle bedeuten eine hohe Sicherheit der Schätzung. Das geforderte Konfidenzniveau $1 - \alpha$ ist ein Parameter, der vom Anwender vorgegeben werden muss, üblich sind hier beispielsweise Werte von $1 - \alpha = 0.95$ oder $1 - \alpha = 0.99$ [124]. Ein Wert

von $1 - \alpha = 0.95$ bedeutet, dass der wahre Wert p_{Ω_w} mit einer Sicherheit von 95% innerhalb des angegebenen Konfidenzintervalls liegt.

4.3.2 Schätzung der Abdeckung des Datensatzes

Zusätzlich zur Auftretenswahrscheinlichkeit kann anhand des Ergebnisses des Fahrsequenz-Clusterings der Grad der Abdeckung der bekannten Szenarien auf einem bestimmten Segment abgeschätzt werden. Abdeckung beschreibt hier den Anteil der gefundenen Szenarien an allen Szenarien, die auf einem Segment auftreten können. Konkret dient diese Abschätzung der Entscheidung, ob zu einem gegebenen Zeitpunkt (z.B. nach einem Jahr Datensammlung) die Datensammlung beendet werden kann oder fortgesetzt werden sollte.

Beschreibung über Wachstumsmodelle

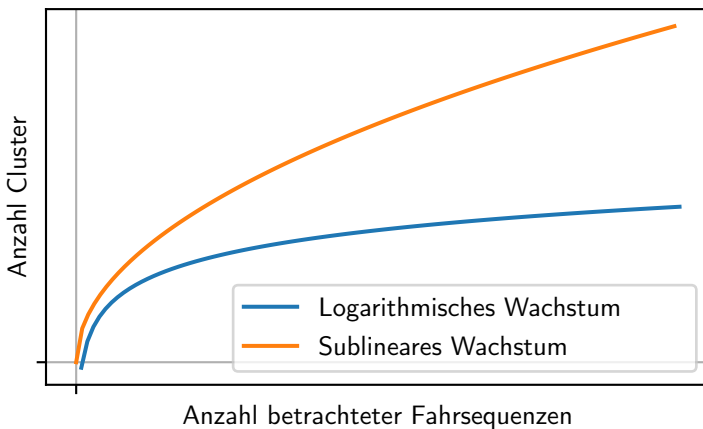


Abbildung 4.11: Mögliche Modelle für Wachstumsverhalten der Clusteranzahl über der Anzahl der betrachteten Fahrsequenzen.

Der erste Weg leitet sich aus dem Gedankenexperiment des Fahrers, der kontinuierlich die Welt entdeckt (vgl. Abschnitt 4.2.2) her. Zu Beginn wird der Fahrer viele neue und zuvor unbekannte Ereignisse erleben, mit der Zeit werden unbekannte Ereignisse immer seltener, bis er nahezu alles gesehen hat und neue Ereignisse nur noch selten auftreten. Abgebildet auf das Clustering ist die Rate der Bildung neuer Cluster zu Beginn am höchsten und sinkt dann ab. Die Anzahl der gefundenen Cluster über der Anzahl analysierter Fahrsequenzen wird somit durch ein Wachstumsmodell beschrieben. Im Grenzfall, dass für jede Fahrsequenz ein neues Cluster entsteht, liegt lineares Wachstum vor. Aufgrund der beschriebenen Effekte wird aber eher langsames Wachstum erwartet. Gängige Wachstumsmodelle mit geringerem als linearem Wachstum sind das logarithmische und das sublineare Wachstum.

$$|C|(|S|) = a_{\log} \log |S| + b_{\log} \quad (\text{Logarithmisches Wachstum}) \quad (4.26)$$

$$|C|(|S|) = a_{\text{sub}} \sqrt{|S|} + b_{\text{sub}} \quad (\text{Sublineares Wachstum}) \quad (4.27)$$

a und b sind Parameter, die durch Beobachtungsdaten ermittelt werden müssen. Nachdem ein Datensatz aufgezeichnet wurde, beispielsweise mit der Dauer von einem Jahr, kann die Kurve der Anzahl Cluster $|C|$ über der Anzahl der Fahrsequenzen $|S|$ aufgetragen werden. Durch gängige Kurvenanpassungsverfahren (engl. *curve fitting*) werden die Parameter a, b ermittelt. Da nicht bekannt ist, welches Wachstumsmodell zutreffend ist, muss anhand der Daten entschieden werden. Dazu wird die Güte der angepassten Wachstumsmodelle im Vergleich zu den Beobachtungsdaten mit dem R2-Score bestimmt. Der R2-Score wird wie folgt berechnet:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.28)$$

wobei y_i die beobachteten Werte, also die reale Clusteranzahl, \hat{y}_i die vorhergesagten Werte und \bar{y} der Durchschnitt der beobachteten Werte sind. Das Modell mit dem höchsten R2-Score wird dann dazu verwendet, den weiteren Anstieg des Clusteranzahl vorherzusagen.

Modellfreie Schätzung

Für den Fall, dass kein passendes Wachstumsmodell gefunden werden kann, wird ein zweiter Weg zur Schätzung der noch unbekanntem Szenarien vorgeschlagen: Dazu wird ein von I. Good und G. Toulmin entwickeltes Schätzverfahren [45] angewendet. Das Verfahren wurde 1956 entwickelt, um anhand der Zählung gefundener Schmetterlingsarten auf einer Insel auf die Anzahl noch nicht beobachteter Arten auf dieser Insel zu schließen. Als Eingangswerte werden zwei Informationen benötigt:

1. die Anzahl der entdeckten Schmetterlingsexemplare pro Art sowie
2. die Häufigkeit, mit der Arten einmal, zweimal oder n -mal beobachtet wurden.

Eine Intuition für das Schätzverfahren lässt sich anhand des folgenden Beispiels entwickeln: wurden im Zeitraum $[0, T]$ insgesamt R Schmetterlingsexemplare gefunden und jeder Schmetterling gehört einer neuen Art an, ist zu erwarten, dass auch im Zeitraum $[T, 2T]$ weiterhin viele neue Arten gefunden entdeckt werden. Wurden stattdessen alle R gefundenen Exemplare nur zwei verschiedenen Arten zugeordnet, so ist es plausibler, im Zeitraum $[T, 2T]$ keine oder kaum zusätzliche Arten neu zu detektieren. Das Schätzverfahren ist also in der Lage, die erwartete Anzahl neu entdeckter Arten bei einer Verdopplung des Beobachtungszeitraums zu bestimmen.

$ c_w $	1	2	3	4	5	...	$ c_w _{\max}$
Häufigkeit	Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	...	$\Phi_{ c_w _{\max}}$

Tabelle 4.2: Tabellarische Auflistung der Anzahl der Größen der Cluster.

Übertragen auf das Fahrsequenz-Clustering entsprechen die Anzahl der Schmetterlingsarten der Anzahl der Cluster, die Exemplare pro Art sind die entsprechenden Cluster-Größen. Das Verfahren basiert also auf einer Zählung der auftretenden Cluster-Größen (vgl. Tabelle 4.2). Die Cluster-Größe $||c_w||$ bezeichnet die Anzahl

Fahrsequenzen pro Cluster. Φ_i bezeichnet die Anzahl von Clustern der Größe i in C .

Die geschätzte Zahl der noch zu erwartenden Cluster $|C|_{\text{est}}$ kann dann durch

$$|C|_{\text{est}} \hat{=} - \sum_{i=0}^{\infty} (-t)^i \Phi_i \quad (4.29)$$

bestimmt werden. Dabei gilt $t = \frac{m}{n}$, wobei m die Anzahl der zukünftigen und n die Zahl der bereits beobachteten Fahrsequenzen ist. Mit $m = n$ und somit $t = 1$ wird demnach die Anzahl der erwarteten, neuen Cluster geschätzt, wenn der Datensatz verdoppelt wird. Die Qualität der Schätzung steigt, je größer die Anzahl der bereits analysierten Fahrsequenzen n ist und sinkt für große Werte von t [88]. Der maximale Wert für t ist $t = 1$, bei größeren Werten divergiert der Schätzwert und läuft, abhängig von den konkreten Eingangswerten, gegen Unendlich bzw. Null [88]. Somit kann mit dem Good-Toulmin Schätzverfahren maximal die Zahl der Cluster bei einer Verdopplung der beobachteten Fahrsequenzen abgeschätzt werden.

5 Prototypische Realisierung des Fahrsequenz-Clusterings

5.1 Betrachtung anhand simulierter Daten

Zur qualitativen Bewertung des Clusterings wird es mit einer Grundwahrheit als Referenz verglichen. Da diese in realen Datensätzen nicht verfügbar ist, wurden in einem ersten Versuchsaufbau mithilfe einer Simulationsumgebung Fahrsequenzen generiert. Dazu werden logische Szenarien mehrfach variiert und die konkreten Szenarien in einem Simulationsprogramm ausgeführt. Die Zuordnung der Fahrsequenzen zum erzeugenden logischen Szenario wird dann als Vergleichs-Referenz für das Clustering-Ergebnis verwendet.

5.1.1 Versuchsaufbau

Zur Generierung des simulierten Datensatzes wird der Euro NCAP Szenario-Katalog für den Schutz von verletzlichen Verkehrsteilnehmern verwendet [94]. Euro NCAP ist eine Gesellschaft europäischer Verkehrsministerien, Automobil-Clubs und Versicherungsverbände, die eine Sicherheitsbewertung von Fahrzeugsystemen anhand von standardisierten Szenarien vornimmt. Für den Test von automatisierten Brems- und Spurhalteassistenten sind in [94] Szenarien definiert. Für die Erprobung des Fahrsequenz-Clusterings wurden fünf Szenarien ausgewählt (vgl. Tabelle 5.1). Sie bilden im urbanen Umfeld typische Szenarien ab, die potentiell zu Unfällen führen können. Somit sind urbane, potentiell kritische Szenarien im Datensatz enthalten. Da nur jeweils ein Ego-Fahrzeug und ein weiterer

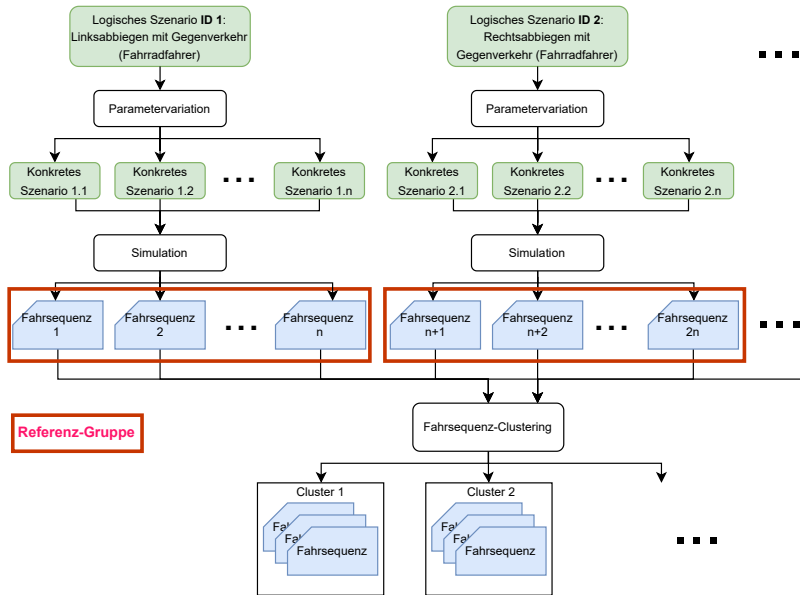


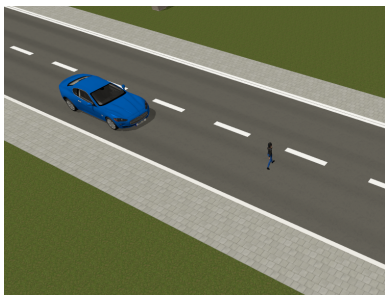
Abbildung 5.1: Darstellung des Vorgehens zur quantitativen Erprobung anhand eines Datensatzes von simulierten NCAP-Szenarien. Hier exemplarisch dargestellt anhand der logischen Szenarien mit ID 1 und ID 2.

Teilnehmer enthalten ist, bilden die Szenarien nur einen Teil des realen urbanen Verkehrs ab, in dem auch gleichzeitig stattfindende Interaktionen mit mehreren Verkehrsteilnehmern auftreten. Trotzdem kann der Datensatz für eine Analyse der grundsätzlichen Eignung des vorgeschlagenen Fahrsequenz-Clusterings verwendet werden. Die Fähigkeit, die simulierten Fahrsequenzen in sinnvolle Gruppen einzuteilen, ist eine notwendige Bedingung für die Funktionalität des Clusterings mit realen Daten.

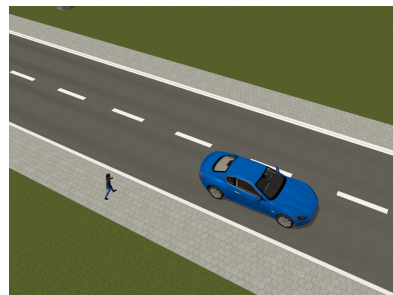
Im Euro NCAP sind für jedes Szenario auch Parameterbereiche für die Ego-Geschwindigkeit angegeben, somit liegen logische Szenarien vor. Für die Simulation werden konkrete Szenarien benötigt. Dazu wurde der Wert der initialen Geschwindigkeit der Ego-Fahrzeuge entsprechend der Vorgaben aus [94] variiert.

ID	Szenario
1	Linksabbiegen mit Gegenverkehr (Fahrradfahrer)
2	Rechtsabbiegen mit Gegenverkehr (Fahrradfahrer)
3	Kreuzender Fußgänger von links
4	Kreuzender Fußgänger von rechts
5	Kreuzender Fußgänger von links mit Verdeckung

Tabelle 5.1: Tabelle der gewählten Szenarien aus dem Euro NCAP.



(a) Fußgänger vor Ego-Fahrzeug



(b) Fußgänger hinter Ego-Fahrzeug

Abbildung 5.2: Momentaufnahme von zwei Fällen im simuliertem Szenario mit ID 4 „Kreuzender Fußgänger von rechts“.

Zusätzlich wurde durch eine zufällige Variation der Zeitpunkt verändert, bei dem der Fußgänger bzw. der Fahrradfahrer seine Bewegung startet. In Anhang A.5 findet sich eine Auflistung der simulierten Szenarien inkl. der variierten Parameter. Als Resultat ergeben sich für vier der fünf logischen Szenarien jeweils zwei verschiedene Semantiken: in einem Teil der Fälle läuft das Szenario ab wie spezifiziert und das Ego-Fahrzeug muss auf den Fußgänger oder Fahrradfahrer reagieren. Durch die Variation von Geschwindigkeiten und Start-Zeitpunkten existieren jedoch ebenso Fälle, in denen das Ego-Fahrzeug nicht reagieren muss, da der Fußgänger oder Radfahrer erst nach dem Ego-Fahrzeug die Straße betritt bzw. befährt (vgl. Abbildung 5.2). Somit werden die simulierten Fahrsequenzen

in 9 Fälle mit unterschiedlicher Semantik eingeteilt (vgl. Tabelle 5.2). Zur Simulation wurde eine Gesamtfahrzeugsimulation¹ verwendet, bei der das Ego-Fahrzeug von einem einfachen Fahrermodell gesteuert wurde.

ID	Logisches Szenario aus Euro NCAP	Unterteilung	Anzahl konkreter Szenarien
1a	Linksabbiegen mit Gegenverkehr (Fahrradfahrer)	Radfahrer vor Ego	11
1b		Radfahrer nach Ego	11
2a	Rechtsabbiegen mit Gegenverkehr (Fahrradfahrer)	Radfahrer vor Ego	11
2b		Radfahrer nach Ego	11
3a	Kreuzender Fußgänger von links	Fußgänger vor Ego	22
3b		Fußgänger nach Ego	20
4a	Kreuzender Fußgänger von rechts	Fußgänger vor Ego	24
4b		Fußgänger nach Ego	19
5	Kreuzender Fußgänger von links mit Verdeckung	Fußgänger vor Ego	20
			149

Tabelle 5.2: Auflistung der logischen Szenarien sowie deren Unterteilung und jeweilige Anzahl konkreter Szenarien aus dem Euro NCAP Szenarienkatalog [94].

5.1.2 Versuchsdurchführung

Im Rahmen der Versuchsdurchführung wird das Fahrsequenz-Clustering mehrfach mit verschiedenen Parametern ausgeführt. In jeder Versuchsdurchführung werden die 149 Fahrsequenzen vollständig geclustert. Die Parameter sind die Schwellwerte Γ_E und Γ_T (vgl. Abschnitten 4.2.4). Die Schwellwerte beschreiben die maximal erlaubte mittlere Abweichung in Meter pro Warping-Schritt des DTW-Algorithmus beim Vergleich von zwei Trajektorien. Dabei beschreibt

¹ dSpace MotionDesk Version 4.6

Γ_E die erlaubte mittlere Abweichung beim Vergleich von Ego-Fahrzeugen und Γ_T die erlaubte mittlere Abweichung beim Teilnehmer-Vergleich. Die Versuchsdurchführung hat also das Ziel, den Einfluss der Parameterwahl auf das Ergebnis des Fahrsequenz-Clustering zu untersuchen. Dazu wird untersucht, inwieweit die Einteilung des Fahrsequenz-Clustering mit der bekannten Zuordnung der Fahrsequenzen zu ihrem erzeugenden logischen Szenario übereinstimmt. Zu diesem Zweck wird das Clustering-Ergebnis mithilfe der Cluster-Vergleichsmetrik V-Measure (vgl. Abschnitt A.4.2) mit den durch die logischen Szenarien vorgegebenen Referenzgruppen verglichen (vgl. Abbildung 5.1). Das Fahrsequenz-Clustering wurde mit jeder Kombination der Werte

$$\Gamma_E, \Gamma_T \in \{x \in \mathbb{R} \mid 0 \leq x \leq 16, x = n \times 0,5, n \in \mathbb{Z}\} \quad (5.1)$$

durchgeführt. Jeder Parameter wurde also von 0m bis 16m in 0,5m-Schritten variiert. Somit wurde der Datensatz von 149 Fahrsequenzen 1089 mal geclustert und das Ergebnis mit den Referenzgruppen verglichen.

5.1.3 Versuchsergebnisse

Die Schwellwert-Wahl zeigt großen Einfluss auf die Anzahl der resultierenden Cluster. Während für kleine Werte $\Gamma_E = 0$ und $\Gamma_T = 0$ jede der 149 Fahrsequenzen in ein eigenes Cluster eingeteilt wird, werden für große Schwellwerte $\Gamma_E \geq 1,5$ und $\Gamma_T \geq 13,5$ nur 4 Cluster erzeugt (vgl. Abbildung 5.3). Die Qualität eines spezifischen Fahrsequenz-Clustering wird durch die V-Measure beschrieben. Je höher der Wert der V-Measure, desto stärker stimmen die Gruppeneinteilungen zwischen erzeugenden Referenz-Gruppen und Fahrsequenz-Clustering überein. Dies entspricht dem erwarteten Verhalten, da in den aufgezeichneten Daten niemals zwei exakt gleiche Trajektorien auftreten werden. Dies gilt hier aufgrund der Variation der Ego-Geschwindigkeit auch für die simulierten Fahrsequenzen.

Mit 0.966 werden die höchsten V-Measure-Werte mit den Parameterkombinationen $\Gamma_E \times \Gamma_T = [1,5\text{m}, 16\text{m}] \times 4\text{m}$ erreicht (vgl. Abbildung 5.4). Somit hat

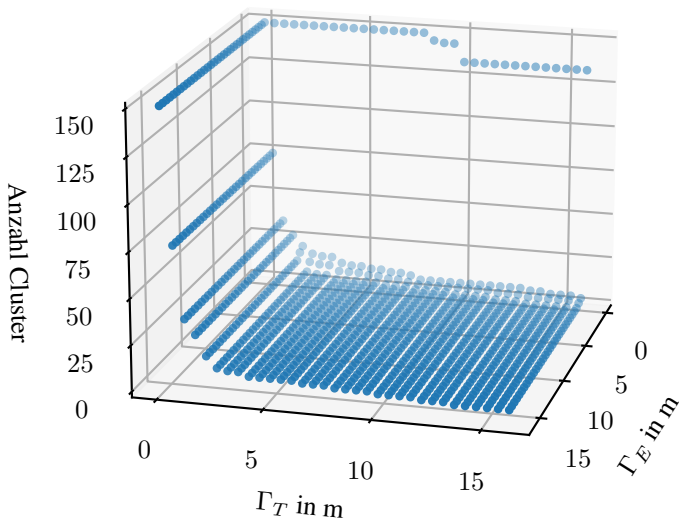


Abbildung 5.3: Clusteranzahl abhängig von Parametern Γ_E, Γ_T .

der Parameter zur Unterscheidung des Teilnehmerverhaltens Γ_T einen größeren Einfluss auf das Clustering-Ergebnis als der Parameter zur Unterscheidung der Ego-Fahrzeuge. Dies ist jedoch bedingt durch den spezifischen betrachteten Datensatz: Die verwendeten logischen Szenarien unterscheiden sich vorwiegend durch die Richtung, aus der ein Fußgänger oder Fahrradfahrer kommt.

Ein Clustering mit der höchsten erreichten V-Measure (hier $\Gamma_E = 4\text{m}, \Gamma_T = 4\text{m}$) wird mit den 9 Referenz-Gruppen in einem Parallele-Kategorien-Diagramm verglichen (vgl. Abbildung 5.5). Jeder vertikale Balken symbolisiert ein Cluster oder eine Referenzgruppe, wobei die Höhe des Balkens die Anzahl der Fahrsequenzen in diesem Cluster darstellt. Auf der linken Seite sind die 9 Referenz-Gruppen aus Tabelle 5.2 und auf der rechten Seite die Ergebnisse des Clusterings aufgetragen.

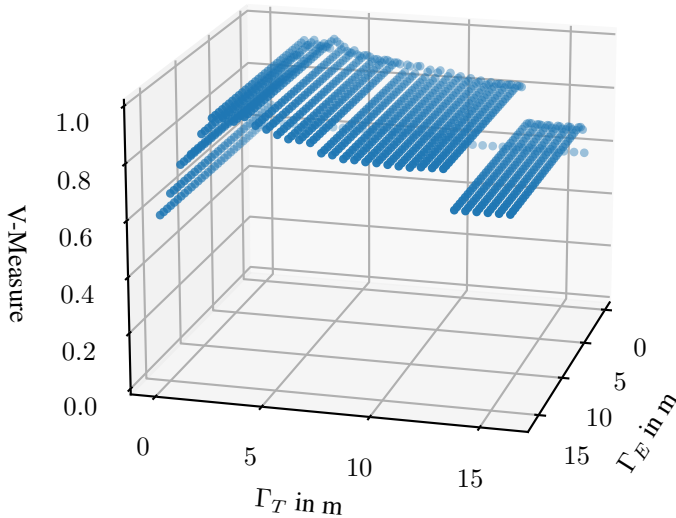


Abbildung 5.4: V-Measure für den Vergleich der Clustering-Ergebnisse mit der Semantik der Fahrsequenzen (vgl. Tabelle 5.2).

Durch einen horizontalen Balken zwischen Referenzgruppe (z.B. 1a) und Cluster (z.B. A) wird symbolisiert, welcher Anteil der Fahrsequenzen der Referenzgruppe im Cluster einsortiert wurde (hier: alle). Dass das Diagramm im wesentlichen aus horizontalen Balken besteht, zeigt, dass die Referenzgruppen vom Clustering im Großteil der Fälle reproduziert wurden. Nur in zwei Fällen weichen die Cluster von den Referenzgruppen ab (siehe rote Markierung): eine Fahrsequenz aus Referenzgruppe 3b findet sich in Cluster E, während alle anderen Fahrsequenzen aus 3b in Cluster F auftreten. Ebenso sind zwei Fahrsequenzen aus Referenzgruppe 4b in Cluster G einsortiert und alle anderen in Cluster H. Somit weicht das Clustering bei 3 der insgesamt 149 Fahrsequenzen von der Referenz ab. Eine genaue Betrachtung und Diskussion dieses Falls findet sich in Abschnitt 6.1.1.

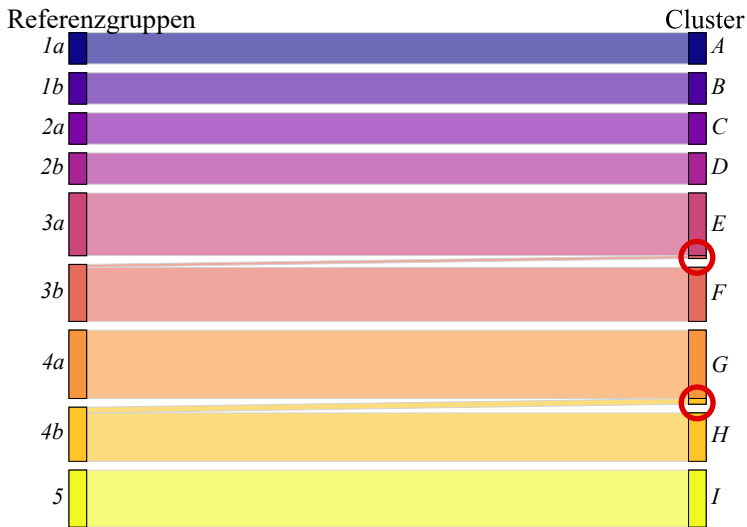


Abbildung 5.5: Parallele-Kategorien-Diagramm für den detaillierten Vergleich des besten Clustering-Ergebnisses mit den Referenz-Gruppen.

Die Auswertung zeigt, dass das Fahrsequenz-Clustering in der Lage ist, semantisch zusammengehörige Fahrsequenzen zu clustern. Jedoch zeigt sich auch, dass keine perfekte Zuordnung erreicht wird. Die Gründe dafür werden in Abschnitt 6.1.1 erläutert.

5.2 Erprobung auf realen Daten

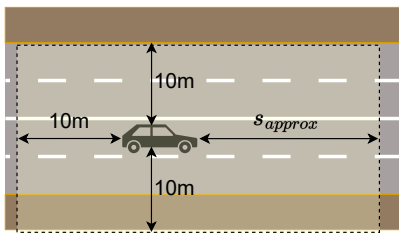
Anhand des simulierten Datensatzes wurde gezeigt, dass das Fahrsequenz-Clustering grundsätzlich in der Lage ist, ähnliche Fahrsequenzen zu clustern. Die Aussagekraft dieses Ergebnisses ist aufgrund der verwendeten Daten jedoch eingeschränkt. Das hat drei Gründe: erstens umfassen die Fahrsequenzen nur maximal zwei Teilnehmer. Somit sind im realen Straßenverkehr auftretende Interaktionen zwischen drei und mehr Verkehrsteilnehmern nicht abgebildet. Zweitens

sind die Bewegungen sowohl des Ego-Fahrzeugs als auch der Teilnehmer von Simulationsmodellen gesteuert, die Trajektorien entsprechen also nicht zwingend den Trajektorien von realen Verkehrsteilnehmern. Drittens ist der Datensatz mit 149 Fahrsequenzen sehr klein im Vergleich zur angestrebten Beobachtungsdauer von einem Jahr (vgl. Abschnitt 4.2.1). Aus diesem Grund wird das Fahrsequenz-Clustering im folgenden anhand eines realen Datensatzes erprobt.

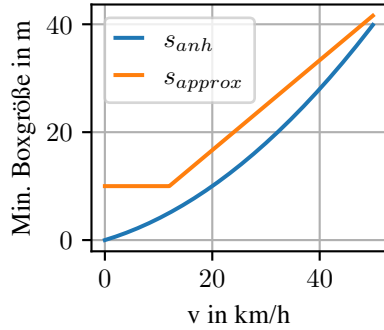
5.2.1 Datensatz

Für die qualitative Erprobung des Fahrsequenz-Clusterings wird die Aufzeichnung von realem Verkehr, aufgezeichnet an einer T-Kreuzung in einem Wohngebiet einer süddeutschen Stadt verwendet. Die Aufzeichnung umfasst einen vollständigen Monat und wurde im Sommer 2022 durchgeführt. Dazu wurde Sensorik an einem Mast in ca. 10m Höhe angebracht. Entsprechend der beschriebenen Schritte (vgl. Abschnitt 4.1) wurden die Daten vorverarbeitet und in Fahrsequenzen transformiert. Jede Fahrsequenz besteht aus Ego-Fahrzeugen sowie den sich gleichzeitig auf der Kreuzung befindlichen Teilnehmern. Ein Teil der Teilnehmer einer Fahrsequenz hat keine Relevanz für das Verhalten des Ego-Fahrzeugs, beispielsweise wenn sich der Teilnehmer weit hinter dem Ego-Fahrzeug befindet. Diese Teilnehmer wurden aus den Fahrsequenzen durch die Definition eines Relevanzbereichs herausgefiltert (vgl. Abschnitt 4.1.4).

Dazu wurde ein Bereich um das Ego-Fahrzeug definiert, der sich mit dem Ego-Fahrzeug mitbewegt. Teilnehmer, die sich zu keinem Zeitpunkt innerhalb dieses Bereichs befinden, werden nicht in die Fahrsequenz mit aufgenommen. Der Relevanzbereich um das Ego-Fahrzeug wird als Rechteck mit einer Größe abhängig von der Ego-Geschwindigkeit definiert. Dabei werden in lateraler Richtung und longitudinal nach hinten jeweils exakt 10m, nach vorne mindestens 10m definiert. Die Größe des Bereichs nach vorne ergibt sich aus dem notwendigen Anhalteweg



(a) Relevanzbereich



(b) Boxgröße in Fahrtrichtung

Abbildung 5.6: Geschwindigkeitsabhängiger Relevanzbereich um Ego-Fahrzeug.

bei verschiedenen Geschwindigkeiten. Für eine konservative Abschätzung von Anhalte und Reaktionsweg existieren die folgenden Formeln²:

$$s_{\text{reaktion}} = v/10 * 3, \tag{5.2}$$

$$s_{\text{brems}} = (v/10) * (v/10). \tag{5.3}$$

Dabei werden die Werte für die Geschwindigkeit v in km/h verwendet. Für den Anhalteweg gilt dann $s_{\text{anhalte}} = s_{\text{reaktion}} + s_{\text{brems}}$. In Abbildung 5.6 ist der Zusammenhang zwischen Anhalteweg und Geschwindigkeit gezeigt. Für eine einfachere Berechnung wird der Anhalteweg nach oben linear abgeschätzt, indem eine TTC (vgl. Abschnitt 3.1.2) von 3 als Grenze gesetzt wird. Der Relevanzbereich nach vorne ist dann die Distanz, die bei ungebremster Fahrt in drei Sekunden zurückgelegt würde.

² Faustformeln für die Berechnung des Bremswegs und des Reaktionswegs, Quelle ADAC, <https://www.adac.de/verkehr/rund-um-den-fuehrerschein/erwerb/anhalteweg-berechnen/>

Der resultierende Datensatz enthält 105704 Fahrsequenzen. Aufgezeichnet wurden die Positionen sowie die Geschwindigkeiten der verschiedenen Objekte. Die Signale haben eine Zeitauflösung von 10 Hz.

5.2.2 Ergebnisse des Fahrsequenz-Clusterings

Das Fahrsequenz-Clustering wurde mit zwei Parameterkombinationen ausgeführt. Dabei wurden einmal die Schwellwerte $\Gamma_E = 4m$ und $\Gamma_T = 4m$ selektiert, welche auf simulierten Daten die höchste V-Measure (0,966) erreicht hatten (vgl. Abschnitt 5.1.3). Um den Einfluss der Schwellwert-Parameter in realen Daten zu untersuchen, wurde zusätzlich noch ein Clustering mit den Schwellwerten $\Gamma_E = 6m$ und $\Gamma_T = 6m$ (V-Measure von 0,901) durchgeführt. Im Gegensatz zum simulierten Datensatz ist für die Fahrsequenzen im realen Datensatz keine korrekte Zuordnung zu Referenzgruppen bekannt. Somit kann kein quantitatives Gütemaß (z.B. V-Measure) bestimmt werden. Daher werden im Folgenden die Verteilungen der Cluster sowie die Inhalte ausgewählter Cluster betrachtet.

Parameter	Anzahl Fahrsequenzen	Anzahl Cluster	Mittlere Cluster-Größe	Anteil $ c == 1$
$\Gamma_E = 4m, \Gamma_T = 4m$	105704	26950	3,92	80%
$\Gamma_E = 6m, \Gamma_T = 6m$	105704	19645	5,38	74%

Tabelle 5.3: Ergebnisse des Fahrsequenz-Clusterings auf realen Daten.

Die Ergebnisse zeigen, dass das Clustering mit niedrigeren Schwellwerten eine höhere Zahl an Clustern und demnach eine geringere durchschnittliche Anzahl Fahrsequenzen pro Cluster aufweist (vgl. Tabelle 5.3). Dies entspricht den Ergebnissen mit dem simulierten Datensatz (vgl. Abschnitt 5.1.3). Die Verteilung der Cluster-Größen zeigt für beide Parameterkombinationen, dass zwei große Cluster existieren, die jeweils mehr als 10000 Fahrsequenzen umfassen (vgl. Abbildung 5.7). Danach fällt die Größe der Cluster schnell ab, beispielsweise

umfasst das zehntgrößte Cluster nur 616 ($\Gamma_E = 4m, \Gamma_T = 4m$) bzw. 1025 ($\Gamma_E = 6m, \Gamma_T = 6m$) Fahrsequenzen und das 20. nur 306 bzw. 457 Fahrsequenzen. Den mit Abstand höchsten Anteil an allen Clustern haben die Cluster mit nur einer einzigen Fahrsequenz (80% bzw. 74% aller Cluster, vgl. Abbildung 5.8). Somit konnten 21% bzw. 14% aller Fahrsequenzen keiner anderen Fahrsequenz im Datensatz zugeordnet werden.

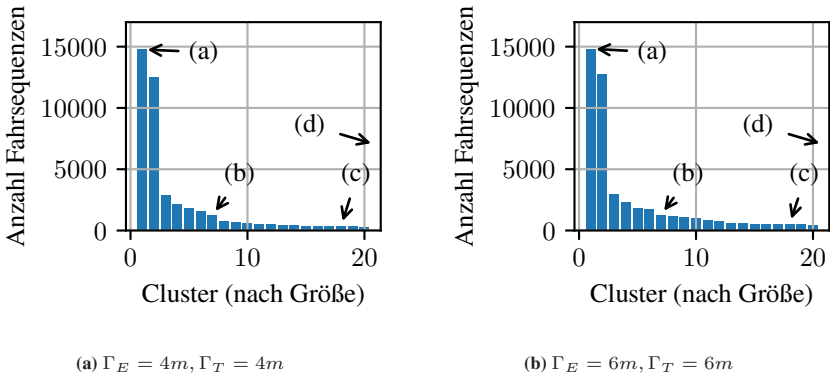


Abbildung 5.7: Anzahl Fahrsequenzen in den größten 20 Clustern für zwei Schwellwert-Parameter.

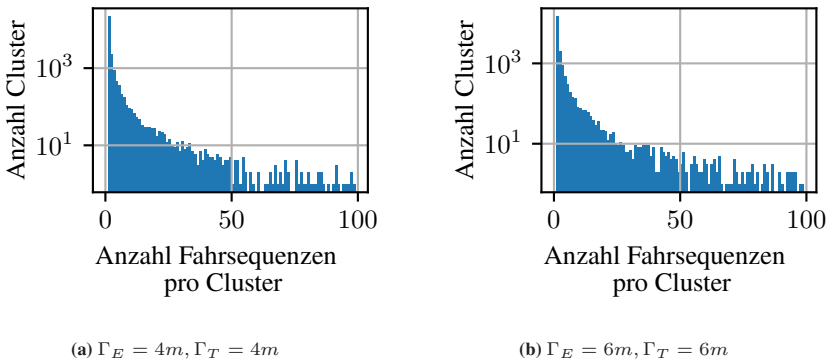


Abbildung 5.8: Häufigkeit von Clustern mit einer spezifischen Größe, Cluster mit mehr als 100 Fahrsequenzen sind abgeschnitten. Aufgrund der Größenunterschiede ist die Y-Achse logarithmisch skaliert.

In den Clustern mit den meisten Fahrsequenzen befinden sich die Fälle, bei denen nur ein Ego-Fahrzeug die T-Kreuzung überquert, ohne dass weitere Teilnehmer vorhanden sind. Dabei können die Ego-Fahrzeuge sechs verschiedene Wege über die Kreuzung nehmen. Die größten sechs Cluster entsprechen diesen sechs Wegen. Am häufigsten wurde dabei von rechts kommend links abgebogen (vgl. Abbildung 5.9a). Die Cluster im Bereich von ca. 1000 Fahrsequenzen umfassen größtenteils Fahrsequenzen mit einem Teilnehmer, der entweder dem Ego-Fahrzeug folgt oder in der Gegenspur fährt (vgl. Abbildung 5.9b). Die Cluster im Bereich von ca. 100 Fahrsequenzen beinhalten Fahrsequenzen mit Kombinationen mehrerer Teilnehmern (z.B. ein folgendes und ein entgegenkommendes Fahrzeug) oder Interaktionen mit Fußgängern (vgl. Abbildung 5.9c). Die Fahrsequenzen der Cluster mit nur einer Fahrsequenz sind aus zwei Gründen einzigartig: meistens sind gleichzeitig eine hohe Anzahl von Teilnehmern verschiedener Typen vorhanden (vgl. Abbildung 5.9d), außerdem sorgen Wende- oder Rangiermanöver des Ego-Fahrzeugs für Einzigartigkeit. Die beschriebenen Inhalte der Cluster verschiedener Größe gelten für beide untersuchten Parameterkombinationen. Die exemplarisch aufgeführten Cluster existieren in beiden Clusterings, nur die exakte Größe weicht leicht ab. Gründe für die Unterschiedlichkeit werden in Abschnitt 6.1.2 diskutiert.

5.2.3 Ableitungen aus Clustering-Ergebnis

Anhand des Ergebnisses des Fahrsequenz-Clusterings wird die Auftretenswahrscheinlichkeit der Cluster geschätzt (vgl. Abschnitt 4.3.1). Dazu werden die Ergebnisse des Fahrsequenz-Clusterings mit der Parameterkombination $\Gamma_E = 6m, \Gamma_T = 6m$ verwendet. In Abbildung 5.10 sind die auf Basis der Cluster geschätzten Wahrscheinlichkeiten mitsamt ihres Konfidenzintervalls, hier gewählt zu $1 - \alpha = 0.95$, dargestellt. Die geschätzte Wahrscheinlichkeit für das größte Cluster mit 14831 Fahrsequenzen liegt somit zwischen 0,138 und 0,142, die Wahrscheinlichkeit eines Clusters mittlerer Größe mit 1260 Fahrsequenzen zwischen 0.011 und 0.013. Ein Cluster mit einer einzelnen Fahrsequenz entspricht einer geschätzten Wahrscheinlichkeit zwischen 0 und 0.0000284.

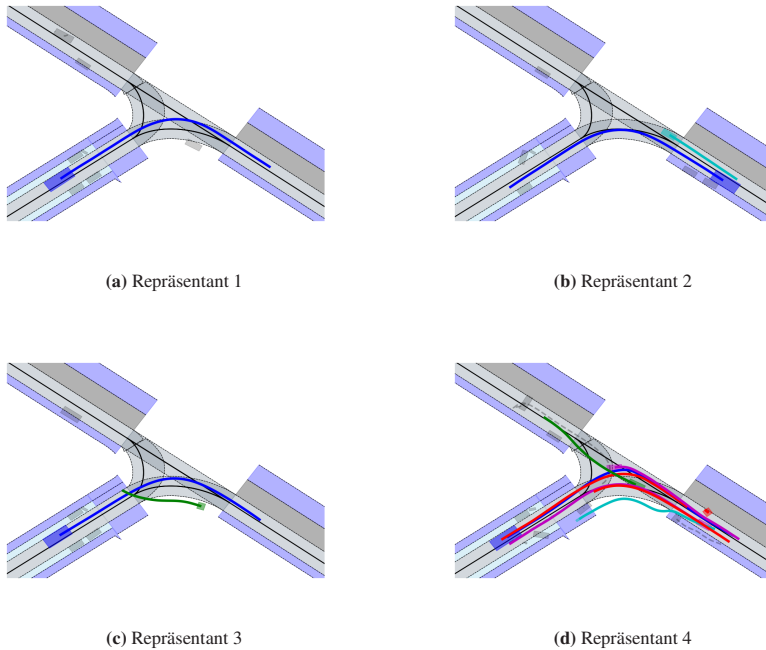


Abbildung 5.9: Repräsentative Fahrsequenzen von vier exemplarischen Clustern. Dabei ist das Ego-Fahrzeug in blau dargestellt. Der dargestellte Zeitpunkt ist der Endpunkt der Ego-Trajektorie. Als irrelevant markierte Teilnehmer sind ausgegraut.

Des Weiteren wird die Abdeckung des analysierten Datensatzes über alle aufgetretenen Szenarien untersucht, indem die Anzahl der noch zu erwartenden Cluster geschätzt wird (vgl. Abschnitt 4.3.2). Dazu wird die Entwicklung der Anzahl Cluster über mit steigender Anzahl analysierter Fahrsequenzen betrachtet (vgl. Abbildung 5.11). Beide Kurven weisen eine über die Zeit absinkende Steigung auf. Dies entspricht dem erwarteten Verhalten: mit zunehmender Anzahl analysierter Fahrsequenzen wächst die Anzahl der Cluster und somit das Wissen über mögliche Szenarien auf einem gegebenen Streckensegment. Demzufolge treten weniger Fahrsequenzen mit unbekanntem Inhalt auf.

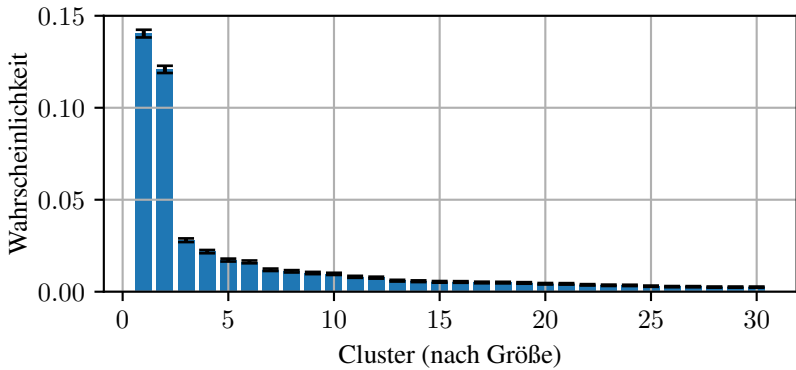


Abbildung 5.10: Schätzung der Auftretenswahrscheinlichkeiten mitsamt ihrer Konfidenzbereiche

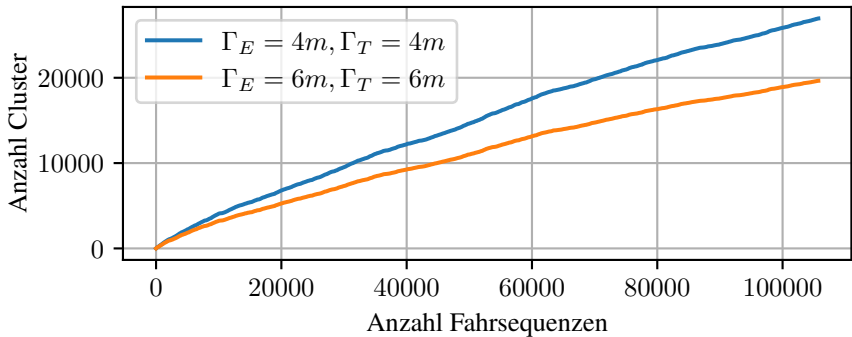


Abbildung 5.11: Wachstumsverhalten der Clusteranzahl über der Anzahl der untersuchten Fahrsequenzen.

Um anhand der Kurve auf die Anzahl noch unbekannter Cluster zu schließen werden Wachstumsmodelle eingesetzt. Die Parameter dieser Wachstumsmodelle werden anhand der Daten geschätzt. Dazu werden die beiden Wachstumsmodelle logarithmisches Wachstum $|C|(|S|) = a_{\log} \log |S| + b_{\log}$ (vgl. Gleichung (4.26)) und sublineares Wachstum $|C|(|S|) = a_{\text{sub}} \sqrt{|S|} + b_{\text{sub}}$ (vgl. Gleichung (4.27)) verwendet. Für die Anpassung der Parameter a_{\log} , b_{\log} und a_{sub} , b_{sub}

wird ein Standard-Algorithmus zur Kurvenanpassung verwendet³. Für die Untersuchung der Wachstumsmodelle wird die Kurve der Parameterkombination $\Gamma_E = 6m, \Gamma_T = 6m$ gewählt. Anhand des R2-Scores (vgl. Gleichung (4.28)) wird die Güte des angepassten Modells überprüft. Zusätzlich wird die Qualität der Vorhersage der zukünftigen Cluster-Anzahl überprüft. Um das zu ermöglichen werden nur 50% der untersuchten Daten für die Kurvenanpassung genutzt, die Clusteranzahl bei 100% der Fahrsequenzen wird dann anhand der Wachstumsmodelle geschätzt.

Modell	Parameter	R2-Score	Vorhersage	Relativer Fehler
Logarithmisch	$a_{\log} = 2823, b_{\log} = -21474$	0.8	11194	0.43
Sublinear	$a_{\text{sub}} = 57, b_{\text{sub}} = -2408$	0.98	16305	0.17

Tabelle 5.4: Ergebnisse der angepassten Wachstumsmodelle bei der Schätzung der zukünftigen Clusteranzahl.

Sowohl der grafische Eindruck (vgl. Abbildung 5.12) als auch die berechneten R2-Scores (vgl. Tabelle 5.4) weisen darauf hin, dass das sublineare Wachstumsmodell das reale Wachstum der Clusteranzahl besser abbildet als das logarithmische. Auf Basis von 50% der Fahrsequenzen werden vom sublinearen Wachstumsmodell 16305 Cluster vorhergesagt. Damit wird die gefundene Clusteranzahl von 19645 um 17% unterschätzt. Der relative Fehler berechnet sich hier zu $\frac{|C_{100\%}| - |C_{\text{est}}|}{|C_{100\%}|}$, wobei $|C_{100\%}|$ die reale Clusteranzahl bei 100% der Daten und $|C_{\text{est}}|$ die geschätzte ist. Das logarithmische Wachstum ergibt deutlich ungenauere Werte und unterschätzt die gefundene Clusteranzahl um 43%. Die grafische Darstellung zeigt, dass beide Wachstumsmodelle das reale Wachstumsverhalten nur unzureichend erklären und den realen Kurvenverlauf nicht treffen.

³ Non-linear least squares, siehe [59]

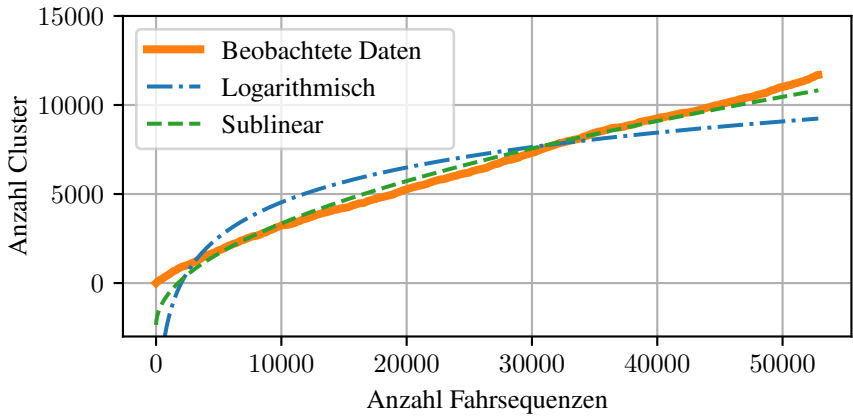


Abbildung 5.12: Darstellung der Wachstumskurven im für die Parameteranpassung verwendeten Bereich bis 50% des gesamten Datensatzes.

Als zweiter Weg zur Schätzung der unbekannt, zukünftigen Clusteranzahl wurde der Good-Toulmin-Schätzer beschrieben (vgl. Gleichung (4.29)). Dazu wird die Häufigkeit der verschiedenen Cluster-Größen betrachtet (Tabelle 5.5). Um auch hier die Qualität dieser Schätzung beurteilen zu können, wurden analog zur Auswertung oben nur 50% der Daten verwendet.

$ c_w $	1	2	3	4	5	6	...	7045	7046
Häufigkeit Φ	8797	1154	500	249	188	114	...	0	1

Tabelle 5.5: Tabellarische Auflistung der Anzahl der Clustergrößen für die erste Hälfte des Datensatzes mit $\Gamma_E = 6m$ und $\Gamma_T = 6m$.

Anhand von Gleichung (4.29) mit $t = \frac{m}{n}$ und n bereits beobachteten Fahrsequenzen sowie m zukünftig zu beobachtenden Fahrsequenzen wird die Clusteranzahl bei Verdopplung der Daten geschätzt, indem $m = n$ und somit $t = 1$ gesetzt

werden. Werden t sowie die Werte aus der Tabelle in Gleichung (4.29) eingesetzt, ergibt sich

$$|C|_{\text{est},2} \hat{=} - \sum_{i=0}^{\infty} (-1)^i \Phi_i \quad (5.4)$$

$$= 8797 - 1154 + 500 - 249 + 188 - 114 + \dots - 0 + 1 \quad (5.5)$$

$$= 8036 \quad (5.6)$$

$|C|_{\text{est},2}$ beschreibt hier die Anzahl zu erwartender, neuer Fahrsequenzen bei Analyse der zweiten Hälfte des Datensatzes. Die Gesamtzahl der Cluster nach 100% des Datensatzes errechnet sich demnach aus $|c_{50\%}| + |C|_{\text{est},2} = 11688 + 8036 = 19724$. Der relative Fehler zur gefundenen Clusteranzahl 19645 von liegt hier nur bei 0,4%.

Zur Überprüfung dieser Ergebnisse wurden zwei weitere Versuche durchgeführt. Dazu wurden 25% des Datensatzes für die Kurvenanpassung bzw. die Schätzung des Good-Toulmin-Schätzers verwendet. Mit dieser Datenbasis wurde die Clusteranzahl bei 50% des Datensatzes (Versuch B) und die Clusteranzahl bei 100% des Datensatzes (Versuch C) bestimmt (vgl. Tabelle 5.6).

Verfahren	A: 50% Basis Vorhersage 100%	B: 25% Basis Vorhersage 50%	C: 25% Basis Vorhersage 100%
Wahrer Wert	19645 Cluster	11688 Cluster	19645 Cluster
Logarithmisch	11194 Cluster, Fehler 0.43	6426 Cluster, Fehler 0.45	7542 Cluster, Fehler 0.61
Sublinear	16305 Cluster, Fehler 0.17	9292 Cluster, Fehler 0.20	13674 Cluster, Fehler 0.30
Good-Toulmin	19724 Cluster, Fehler 0.004	10936 Cluster, Fehler 0.06	∞ Cluster, Fehler ∞

Tabelle 5.6: Zusammenfassung der drei Versuche zur Prädiktion der Clusteranzahl.

Die Ergebnisse der zusätzlichen Experimente verstärken die Tendenz: Das logarithmische Wachstumsmodell unterschätzt die reale Clusteranzahl stärker als das sublineare Wachstumsmodell. Zusätzlich zeigt sich: mehr Daten (absolute Zahl der Fahrsequenzen) als Grundlage für die Kurvenanpassung führen zu besseren Schätzungen, auch wenn in beiden Fällen die Clusteranzahl bei einer Verdopplung der Daten geschätzt wird (vgl. Versuche A,B). Der Good-Toulmin Schätzer zeigt bei einer Verdopplung der Daten jeweils die besten Ergebnisse. Wird der Schätzer jedoch verwendet, um mehr als eine Verdopplung der Daten vorherzusagen, wird das Ergebnis extrem groß⁴ (Versuch C). Der Grund dafür liegt darin, dass der Exponent in Gleichung (4.29) größer als 1 wird, was zu einem divergenten Verhalten (hier: gegen ∞) führt.

Soll der Datensatz maximal verdoppelt werden, eignet sich somit der Good-Toulmin-Schätzer am besten, da er in beiden Versuchen den geringsten relativen Fehler aufweist. Bei Vorhersagen mit mehr als einer Verdopplung des Datensatzes wird er jedoch invalide und es müssen Wachstumsmodelle verwendet werden. Die hier untersuchten neigen zu einer Unterschätzung der Daten, die jedoch bei einer Vergrößerung der Datenmenge geringer wird.

⁴ hier notiert als ∞ , da eine 64-Bit Fließkommazahl nicht ausreicht um den realen Wert zu codieren.

6 Evaluation und Diskussion des Konzepts für datenbasierte Szenarienanalyse

6.1 Diskussion relevanter Aspekte des Fahrsequenz-Clusterings

6.1.1 Beschreibung von Fehlertypen im Fahrsequenz-Clustering

Die Fehler, die im Fahrsequenz-Clustering auftreten können, lassen sich in zwei Kategorien einteilen. Dazu werden die aus der Klassifikation bekannten Fehlertypen (vgl. Abschnitt A.4.1) adaptiert:

Fehler 1. Art Zwei Fahrsequenzen, die semantisch unterschiedlich sind werden vom Fahrsequenz-Clustering in dasselbe Cluster eingeordnet (falsch positiv).

Fehler 2. Art Zwei Fahrsequenzen, die semantisch ähnlich sind werden vom Fahrsequenz-Clustering in verschiedene Cluster eingeordnet (falsch negativ).

Im Kontext dieser Dissertation ist der Fehler 1. Art als kritischer zu betrachten: werden zwei semantisch unterschiedliche Fahrsequenzen in dasselbe Cluster sortiert, kann ein möglicherweise relevantes Szenario übersehen werden. Umgekehrt führt ein Fehler 2. Art nur dazu, dass ein Szenario doppelt in der Datenbank vorhanden ist (vgl. Abschnitt 6.1.4)

Die Auswahl der Parameter sowie die Reihenfolge der Fahrsequenzen können zu Fehlern im Clustering führen. Sie werden im folgenden am Beispiel des simulierten Datensatzes besprochen. Der simulierte Datensatz eignet sich durch die geringe Teilnehmerzahl von maximal 2 sowie die bekannte Referenz-Zuordnung besser für die Diskussion der Fehler, die Fehlertypen und -ursachen sind aber datensatzunabhängig und auch für den Realdatensatz relevant.

Effekte durch Parameterwahl

Wie die Versuchsergebnisse zeigen, existieren Fälle, in denen das Fahrsequenz-Clustering entgegen semantischer Ähnlichkeit gruppiert (vgl. Abschnitt 5.1.3). Beispielsweise wird eine Fahrsequenz mit kreuzendem Fußgänger *hinter* dem Fahrzeug zu einem Cluster mit ansonsten ausschließlich Fahrsequenzen mit dem Fußgänger *vor* dem Fahrzeug geclustert (vgl. Gruppe 3b und Cluster E, Abbildung 5.5). Somit liegt hier ein Fehler 1. Art vor.

Zur Untersuchung dieses Verhaltens werden im folgenden die einzelnen, im Fahrsequenz-Clustering erfolgten Schritte rekapituliert (vgl. Abbildung 4.7). Zu Beginn der Untersuchung liegen fünf Cluster (A,B,C,D,E) vor. Nun soll die Fahrsequenz S_{66} , die den hinter dem Fahrzeug kreuzenden Fußgänger beinhaltet, einem dieser Cluster zugeordnet werden. Da in den Clustern A-D jeweils ein Fahrradfahrer vorkommt, liegen andere Teilnehmertypen vor und somit gilt für diese vier Cluster $\theta_{\text{typ}} = 0$. In Cluster E befindet sich ein Fußgänger, somit gilt hier $\theta_{\text{typ}} = 1$. Anhand der Teilnehmertypen kann S_{66} also nur zu Cluster E zugeordnet werden oder in ein neues Cluster einsortiert werden.

Daher wird als nächster Schritt S_{66} mit der repräsentativen Fahrsequenz von Cluster E, $R(E) = S_{44}$ verglichen. Der erste Schritt ist hier der Vergleich der Ego-Fahrzeuge: Nach den Formeln (4.7)-(4.8) wird die DTW-Distanz P_E berechnet, mit der Länge des Warping-Pfads κ normiert und mit dem Schwellwert $\Gamma_E = 4\text{m}$ verglichen. Dabei ergibt sich $P_E/\kappa = 0,066\text{m} < \Gamma_E = 4\text{m}$ und somit $\theta_E(S_{44}, S_{66}) = 1$. Die Trajektorien der Ego-Fahrzeuge werden demnach als ähnlich identifiziert.

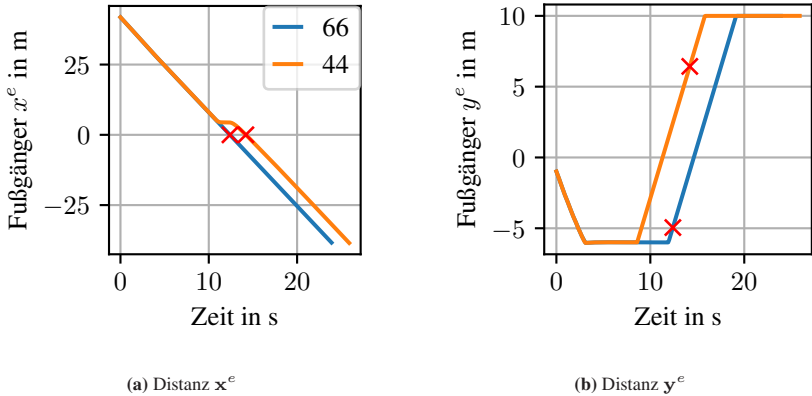


Abbildung 6.1: Relative Abstände aus Ego-Perspektive zu einem über die Straße laufendem Fußgänger für zwei Fahrsequenzen inkl. Markierung der Vorbeifahr-Zeitpunkte (rotes Kreuz).

Somit wird im nächsten Schritt das Verhalten der Teilnehmer verglichen. Im Fall der Fahrsequenzen S_{44} und S_{66} handelt es sich dabei um jeweils einen Fußgänger. Die relativen Trajektorien zum Ego-Fahrzeug zeigen semantisch relevante Unterschiede (vgl. Abbildung 6.1): Durch die roten Kreuze ist für beide Signale markiert, an welchem Zeitpunkt das Ego-Fahrzeug den Fußgänger passiert (Null-durchgang von x^e). Dabei zeigt der Vorzeichenunterschied in y^e , dass sich der Fußgänger zu diesem Zeitpunkt einmal noch links (S_{66}) und einmal schon rechts (S_{44}) des Ego-Fahrzeugs befindet.

Wird nun die normierte DTW-Distanz entsprechend der Gleichungen (4.9)-(4.14)¹ bestimmt, ergibt sich $P_T/\kappa = 3,266\text{m} < \Gamma_T = 4\text{m}$ und somit $\theta_{\text{teil}}(S_{44}, S_{66}) = 1$. Da somit alle drei Bedingungen für Ähnlichkeit im Fahrsequenz-Vergleich erfüllt sind, wird S_{66} dem Cluster E zugeordnet. Der Schwellwert Γ_T ist demnach für die Unterscheidung von S_{66} und S_{44} zu hoch gewählt.

¹ Die Schritte (4.9) sowie (4.13) sind hier trivial, da nur jeweils ein Objekt pro Fahrsequenz vorhanden ist.

Dieser konkrete Fehlerfall kann durch eine Verringerung des Schwellwerts auf $\Gamma_T = 3m$ behoben werden. Jedoch verursacht dieser Schwellwert an anderer Stelle neue Fehler: die Referenz-Gruppe 3b (vgl. Tabelle 5.2) würde in zwei Cluster aufgeteilt. Folglich läge ein Fehler 2. Art vor.

Dieses Beispiel legt dar, dass sich keine Schwellwert-Kombination eignet, um alle Fahrsequenzen perfekt entsprechend der Referenzgruppen zu clustern. Durch die Selektion des Schwellwerts muss also immer eine Abwägung zwischen Fehlern 1. Art (hohe Schwellwerte) und Fehlern 2. Art (kleine Schwellwerte) getroffen werden.

Effekt der Fahrsequenz-Reihenfolge

Das Ergebnis des Fahrsequenz-Clusterings ist abhängig von der Reihenfolge, in der die Fahrsequenzen verarbeitet werden. Dieses Verhalten wird an einer Erweiterung des eben beschriebenen Beispiels dargestellt. Zusätzlich zu S_{44} und S_{66} wird nun auch S_{67} betrachtet. Auch hier handelt es sich um einen kreuzenden Fußgänger, der hinter dem Ego-Fahrzeug die Straße kreuzt (vgl. Abbildung 6.2).

Das Ego-Verhalten ist für diese drei Fahrsequenzen unterhalb des Schwellwerts Γ_E . Die Auswertung des Vergleichs des Teilnehmerverhaltens mit $\Gamma_T = 4$ ergibt sich

$$\theta_{\text{teil}}(S_{44}, S_{66}) = 1 \Rightarrow \theta(S_{44}, S_{66}) = 1, \quad (6.1)$$

$$\theta_{\text{teil}}(S_{66}, S_{67}) = 1 \Rightarrow \theta(S_{66}, S_{67}) = 1, \quad (6.2)$$

$$\theta_{\text{teil}}(S_{44}, S_{67}) = 0 \Rightarrow \theta(S_{44}, S_{67}) = 0. \quad (6.3)$$

In dieser Situation ergeben sich unterschiedliche Ergebnisse, je nachdem in welcher Reihenfolge die Fahrsequenzen verarbeitet werden (vgl. Abbildung 6.3). In

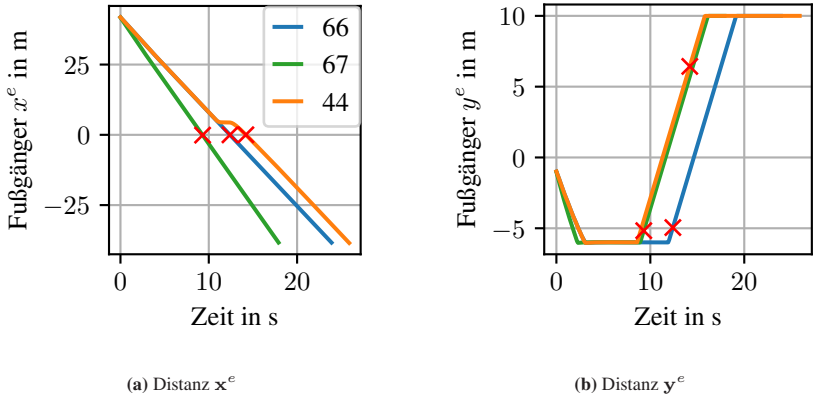


Abbildung 6.2: Relative Abstände aus Ego-Perspektive zu einem über die Straße laufendem Fußgänger für drei Fahrsequenzen S_{44} , S_{66} und S_{67} inkl. Markierung der Vorbeifahr-Zeitpunkte (rotes Kreuz).

einem Fall sorgt, wie oben beschrieben, der Schwellwert $\Gamma_T = 4\text{m}$ für die Zuordnung von S_{44} in Cluster E. Da für Fahrsequenz S_{67} $\theta_{\text{teil}}(S_{44}, S_{67}) = 0$ gilt², passt sie nicht in Cluster E und eröffnet ein neues Cluster F (vgl. Abbildung 6.3a).

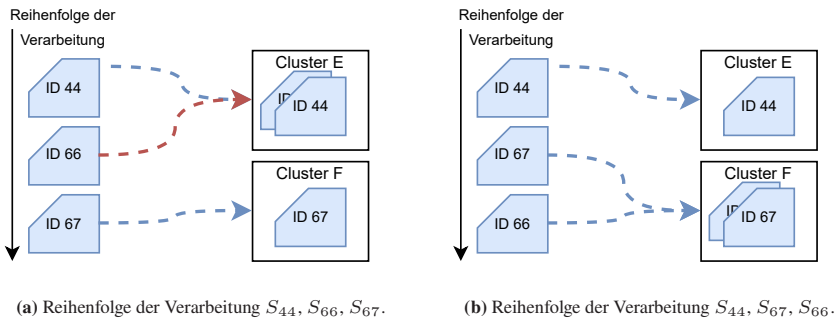


Abbildung 6.3: Zwei verschiedene Clusterergebnisse abhängig von der Verarbeitungsreihenfolge. Korrekte Zuordnungen sind in blau markiert, falsche in rot.

² Die normierte DTW-Distanz der Teilnehmer von S_{44} und S_{67} ergibt $P_T/\kappa = 4,073\text{m} > \Gamma_T = 4\text{m}$.

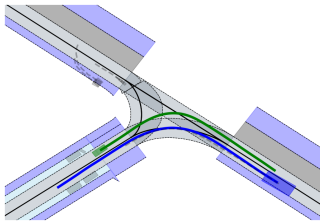
Wird nun jedoch die Reihenfolge der Verarbeitung geändert, sodass S_{67} vor S_{66} verarbeitet wird, ändert sich das Ergebnis: S_{67} passt nicht in Cluster E und eröffnet Cluster F. Für S_{66} ergeben sich nun zwei Möglichkeiten, da sie zu beiden repräsentativen Fahrsequenzen aus Cluster E (vgl. Gleichung (6.1)) und Cluster F (vgl. Gleichung (6.2)) passt. Um die Entscheidung zu treffen, welchem Cluster die Fahrsequenz zugewiesen wird, wird der Ähnlichkeitsgrad von Fahrsequenzen verwendet (vgl. Abschnitt 4.2.4). Da die normierte DTW-Distanz zwischen den Teilnehmern von S_{67} und S_{66} geringer ist als zwischen den Teilnehmern in S_{44} und S_{66} ³, wird S_{66} zu Cluster F hinzugefügt (vgl. Abbildung 6.3b). Die Reihenfolge der Fahrsequenzen hat also einen Einfluss auf das Ergebnis des Clusterings.

Zusammenfassend existieren also zwei Aspekte, die fehlerhafte Zuordnungen im Clustering verursachen. Diese sind jedoch unterschiedlich gewichtet. Die Schwierigkeit, einen für einen gesamten Datensatz passenden Schwellwert einzustellen ist die notwendige Bedingung für das Auftreten eines Fehlers. Ein Fehler im Ergebnis des Fahrsequenz-Clusterings resultiert jedoch nur dann, wenn auch die Reihenfolge der Fahrsequenzen entsprechend gestaltet ist. Das beschriebene Beispiel zeigt, dass auch trotz erfüllter notwendiger Bedingung („ S_{44} und S_{66} werden fälschlicherweise als ähnlich gesehen“) eine korrekte Zuordnung möglich ist, wenn ein noch besser passendes Cluster vorhanden ist (vgl. Abbildung 6.3b). Somit ist die fehler-verursachende Reihenfolge eine hinreichende Bedingung für das Auftreten von Fehlern.

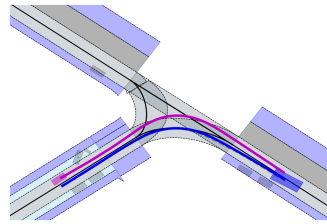
6.1.2 Auffälligkeiten in realen Daten

Aufgrund der Referenzgruppen können die Fehler beim Clustering des simulierten Datensatzes eindeutig erkannt und analysiert werden. Im Gegensatz dazu ist die eindeutige Bestimmung einer korrekten Zuordnung bei Realdaten schwieriger. Die manuelle Analyse des Clusterergebnisses auf der urbanen Kreuzung zeigt

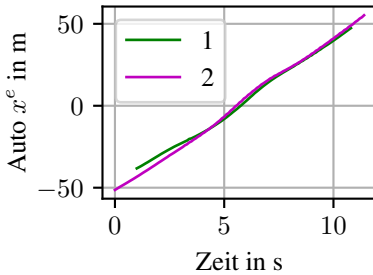
³ Für S_{44} , S_{66} : $P_T/\kappa = 4,073\text{m}$. Für S_{67} und S_{66} : $P_T/\kappa = 0,910\text{m}$



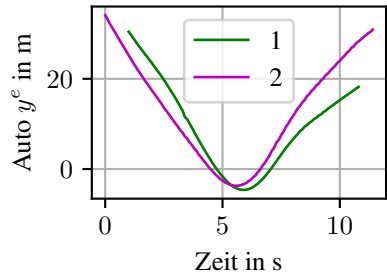
(a) Fahrsequenz 1



(b) Fahrsequenz 2



(c) Relativer Abstand (gemessen im Ego-Koordinatensystem) in longitudinaler Richtung



(d) Relativer Abstand (gemessen im Ego-Koordinatensystem) in lateraler Richtung

Abbildung 6.4: Beispiele für Fahrsequenzen mit einem Teilnehmer. In (a) und (b) ist das Ego-Fahrzeug in blau dargestellt, die entgegen kommenden Fahrzeuge in grün bzw. magenta. Der dargestellte Zeitpunkt ist der Endpunkt der Ego-Trajektorie.

plausible Ergebnisse. Jedoch beinhalten die Ergebnisse auch unerwartete Auffälligkeiten, die im Folgenden anhand von zwei Beispielen beschrieben werden.

Ein Beispiel für Auffälligkeiten sind die Fahrsequenzen, in der das Ego-Fahrzeug rechts abbiegt und ein Fahrzeug entgegen kommt. Hier zeigt sich, dass diese Fahrsequenzen nicht in ein einziges Cluster, sondern in 10 bzw. 14 Cluster eingeteilt wurden.

Zur Analyse der Gründe für dieses Verhalten werden zwei Fahrsequenzen aus unterschiedlichen Clustern auf Trajektorienebene betrachtet (vgl. Abbildung 6.4).

Aus einer semantischen Sicht können die Fahrsequenzen als ähnlich bezeichnet werden, in beiden Fällen kommt dem Ego-Fahrzeug im Bereich der Kurve ein Teilnehmer vom Typ Auto entgegen. Jedoch gibt es einen Unterschied in den exakten Zeitpunkten, an denen der Gegenverkehr auftritt: in Fahrsequenz 1 fährt der Gegenverkehr (grün) erst eine Sekunde nach dem Ego-Fahrzeug auf die Kreuzung, während in Fahrsequenz 2 der Gegenverkehr (magenta) gleichzeitig startet. Betrachtet man die Trajektorien relativ zum Ego-Fahrzeug, so zeigen sich zwei augenscheinlich ähnliche Kurvenverläufe, die leicht zueinander verschoben sind. Die Verschiebung in Zeitrichtung hat bei der Anwendung des DTW-Distanzmaßes keinen Einfluss. Jedoch liegt, insbesondere beim lateralen Abstand zum Ego (vgl. Abbildung 6.4d), auch ein Unterschied im Wertebereich vor. Dieser führt zu einer Distanz oberhalb des Schwellwerts für beide untersuchten Parameter-Kombinationen⁴ und resultiert somit in unterschiedlichen Clustern.

Im zweiten Beispiel wurden Fahrsequenzen 3 und 4 ins selbe Cluster einsortiert. Hier handelt es sich jeweils um ein links abbiegendes Ego-Fahrzeug, bei dem sich auf der linken Seite ein Fußgänger bewegt. In Fahrsequenz 3 bewegt sich der Fußgänger in entgegengesetzter Richtung des Ego-Fahrzeugs (vgl. Abbildung 6.5a), in Fahrsequenz 4 in die gleiche Richtung (vgl. Abbildung 6.5b). Die DTW-Distanz zwischen diesen beiden Fußgängern ist jedoch mit $P_T/\kappa = 3,717\text{m}$ unterhalb des Schwellwerts für beide Parameter-Kombinationen. Dies erklärt sich dadurch, dass für den relativen Abstand zum Ego-Fahrzeug die Bewegung des Ego-Fahrzeugs mehr Einfluss hat als die wesentlich langsamere Bewegung des Fußgängers. Im ersten Beispiel der entgegenkommenden Fahrzeuge liegt möglicherweise ein Fehler 2. Art vor, da semantisch ähnliche Fahrsequenzen in unterschiedliche Cluster einsortiert wurden. Ebenso könnte im zweiten Beispiel der Fußgänger ein Fehler der ersten Art vorliegen, denn die beiden Fußgänger wurden trotz unterschiedlicher Bewegungsrichtung demselben Cluster zugewiesen. Inwiefern wirklich Fehler vorliegen oder nicht, entscheidet sich entlang der Frage, inwiefern sich durch die semantischen Unterschiede ein für die Entwicklung und Absicherung relevantes Szenario ergibt oder nicht. Im ersten Beispiel könnte es

⁴ $P_T/\kappa = 6,653\text{m}$

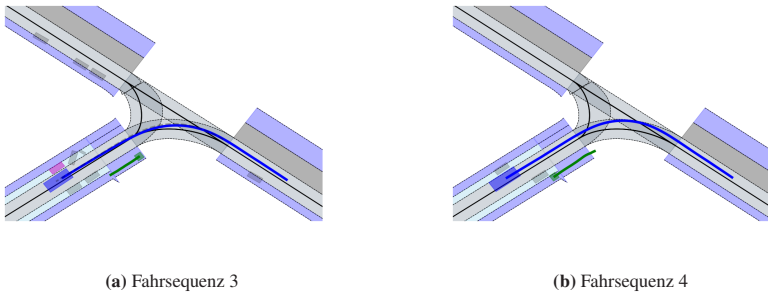


Abbildung 6.5: Beispiele für Fahrsequenzen mit jeweils einem Fußgänger. Das Ego-Fahrzeug ist in blau dargestellt, die Fußgänger in grün. Der dargestellte Zeitpunkt ist der Endpunkt der Ego-Trajektorie. Als irrelevant markierte Teilnehmer sind ausgegraut.

für eine Fahrfunktion durchaus einen Unterschied machen, ob der Gegenverkehr vor, nach oder in der Kurve entsteht. Umfasst die Fahrfunktion eine Posenschätzung für Fußgänger, wäre eine Unterscheidung der Bewegungsrichtung im zweiten Beispiel wünschenswert. Abhängig von derartigen Informationen über die Funktion müssen die Schwellwerte eingestellt werden, da sie das gewünschte Level der Ähnlichkeit definieren.

6.1.3 Rechendauer des Fahrsequenz-Clustering

Die Rechendauer des Fahrsequenz-Clusterings ist relevant, da die Anzahl Fahrsequenzen im Laufe eines Jahres auf bis zu 16 Mio. ansteigen kann (vgl. Abschnitt 4.2.1). Die Rechendauer des Fahrsequenz-Clustering ist von zwei Aspekten abhängig. Zum einen ist das die Anzahl der notwendigen Vergleiche zwischen zwei Fahrsequenzen, zum anderen die Dauer eines solchen Vergleichs. Die Dauer eines Fahrsequenz-Vergleichs hängt von der Anzahl der Teilnehmer und der Länge der Trajektorien innerhalb der beiden Fahrsequenzen ab. Im Mittel über das Clustering hinweg liegt die Dauer bei 0,01s.

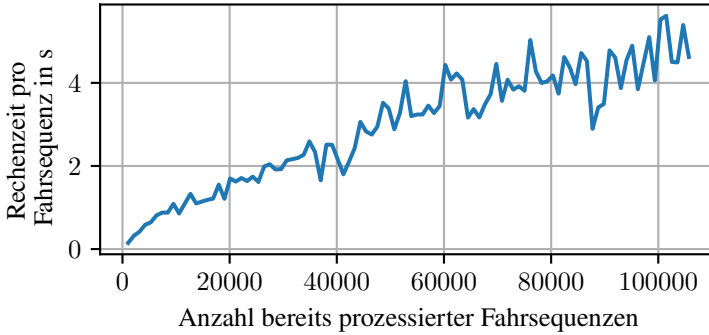


Abbildung 6.6: Entwicklung der Rechenzeit pro verarbeiteter Fahrsequenz während des Clustering von 105704 Fahrsequenzen.

Die Anzahl der notwendigen Vergleiche hängt von der gesamten Anzahl Fahrsequenzen sowie der Anzahl Cluster ab. Da jede Fahrsequenz mit allen, zum Zeitpunkt der Verarbeitung vorhandenen Clustern verglichen wird, steigt die Dauer der Prozessierung pro neuer Fahrsequenz mit zunehmender Clusteranzahl an (vgl. Abbildung 6.6⁵). Während zu Beginn die Fahrsequenzen mit einem Bruchteil einer Sekunde prozessiert wurden, benötigten die letzten Fahrsequenzen mehr als 4 Sekunden⁶.

Da die finale Clusteranzahl bei der Analyse von 16 Mio. Fahrsequenzen unbekannt ist, wird die benötigte Rechenzeit abgeschätzt. Dazu wird das Clustering mit der Parametrierung $\Gamma_T = 6m$ und $\Gamma_E = 6m$ betrachtet: Zum Ende der Prozessierung liegen 19645 Cluster vor. Bei dieser Anzahl Cluster benötigt der Vergleich der Fahrsequenz mit allen anderen Clustern im Mittel eine Zeit von 5,3s. Wird nun angenommen, dass über ein gesamtes Jahr 100000 Cluster gefunden werden, wird die Dauer pro Fahrsequenz linear auf 13,7s geschätzt. Mit der Abschätzung von 16 Mio. Fahrsequenzen für ein Jahr ergibt sich eine Gesamt-Rechendauer von 121942 Stunden oder 5080 Tagen. Somit würde die Rechendauer für ein

⁵ Ermittelt auf einer Workstation, Prozessor AMD S-AM5 Ryzen 9 7950X.

⁶ Mit den Parametern $\Gamma_T = 6m$ und $\Gamma_E = 6m$.

Jahr Datenaufnahme ca. 14 Jahre kosten und ist somit für die Entwicklung einer automatisierten Fahrfunktion nicht anwendbar.

Eine Möglichkeit, die Rechenzeit deutlich zu verkürzen, liegt in der parallelen Ausführung des Algorithmus. Dazu wird der erste Schritt im Fahrsequenz-Vergleich „Vergleich der Teilnehmertypen“ genutzt (vgl. Abschnitt 4.2.4). Nur wenn in zwei Fahrsequenzen die gleichen Teilnehmertypen sowie pro Typ dieselbe Anzahl Teilnehmer vorliegt, können die Fahrsequenzen ähnlich sein. Somit lässt sich ein Datensatz in mehrere Teil-Datensätze mit gleicher Teilnehmertyp-Kombination aufteilen und parallel berechnen (vgl. Abbildung 6.7). Da Fahrsequenzen mit verschiedenen Teilnehmertyp-Kombinationen per Definition nicht in gleiche Cluster sortiert werden können, hat diese Aufteilung keinen Einfluss auf das Ergebnis.

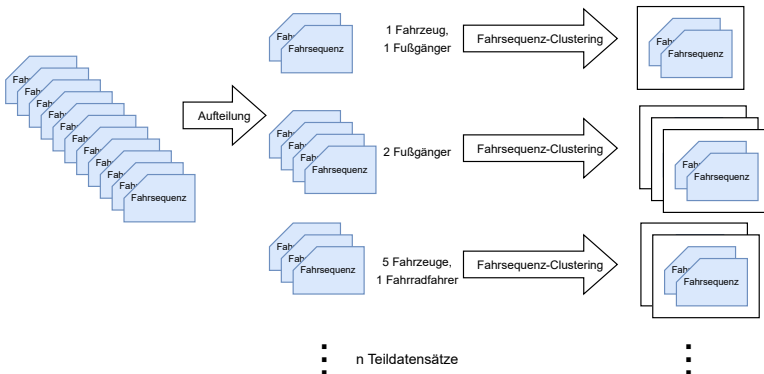


Abbildung 6.7: Aufteilung des Datensatzes in verschiedene Teilnehmertyp-Konfigurationen und anschließende parallele Ausführung des Fahrsequenz-Clusterings.

Eine Analyse der vorhandenen Teilnehmertypen im realen Datensatz (vgl. Abschnitt 5.2.1) ergibt in einem Monat $n = 373$ einzigartige Teilnehmertyp-Kombinationen. Demnach kann er in 373 Teildatensätze aufgeteilt werden, die parallel berechnet verarbeitet werden. Bei größeren Datensätzen ist zu erwarten, dass sich die Zahl der unterschiedlichen Konfigurationen weiter vergrößert. Als

exemplarische Abschätzung für die zu erwartende Anzahl Teilnehmerkonfigurationen werden vier Teilnehmertypen (Fußgänger, Fahrradfahrer, Truck, PKW) mit jeweils 0-9 Teilnehmern des Typs pro Fahrsequenz angenommen. Somit ergeben sich $n = 10^4 = 10000$ verschiedene Konfigurationen, die parallelisiert ausgeführt werden können. Würde jede Teilnehmertyp-Konfiguration gleich häufig auftreten und 10000 parallele Rechenprozesse gestartet, würden die 16 Mio Fahrsequenzen in 0,5 Tagen prozessiert. Diese Gleichverteilung ist jedoch in der Praxis unwahrscheinlich. Die fünf am häufigsten vorkommenden Teilnehmertyp-Kombinationen umfassen bereits 76% aller Fahrsequenzen im Datensatz. Dabei handelt es sich um Fahrsequenzen mit maximal zwei Teilnehmern. Vier der fünf am häufigsten vorkommenden Teilnehmertyp-Kombinationen werden in größere Cluster eingeteilt als der Durchschnitt von 5,38 Fahrsequenzen pro Cluster (vgl. Tabelle 6.1).

Teilnehmertyp-Kombinationen	Anzahl Fahrsequenzen	Anzahl Cluster	Fahrsequenzen pro Cluster	Abgeschätzte Rechenzeit
Kein Teilnehmer	37935	48	790,3	5,1h (Monat), 8 Tage (Jahr)
1 Auto	23210	779	29,8	50,2h (Monat), 75 Tage (Jahr)
2 Autos	9615	1692	5,7	45,2h (Monat), 68 Tage (Jahr)
1 Fußgänger	5972	391	15,3	6,5h (Monat), 10 Tage (Jahr)
1 Auto, 1 Fußgänger	4236	1356	3,1	16,0h (Monat), 24 Tage (Jahr)

Tabelle 6.1: Anzahl Cluster für häufigste Teilnehmertyp-Konfigurationen im verwendeten Realdatensatz.

Diese Tatsache wirkt sich positiv auf die Rechenzeit im parallelisierten Fall aus: Die Gesamtdauer der Verarbeitung eines Teildatensatzes lässt sich durch die Anzahl Fahrsequenzen N_{FS} , die mittlere Dauer pro Fahrsequenz-Vergleich

$t_{vgl} = 0,01s$ sowie der Anzahl Cluster N_{Cl} mit $N_{FS} \times t_{vgl} \times N_{Cl}$ nach oben abschätzen. Die Ergebnisse (vgl. Spalte „Abgeschätzte Rechenzeit“) zeigen, dass auch große Teildatensätze in maximal 50,2h verarbeitet werden. Für eine Hochrechnung der Rechenzeit bei einem ganzen Jahr aufgezeichneter Daten werden folgende Annahmen getroffen: Die Zahl der Fahrsequenzen wird für jeden Monat als konstant angenommen, die Zahl der Cluster wird verdreifacht. Dann ergeben sich Rechenzeiten im Bereich von bis zu 75 Tagen.

Somit zeigt sich, dass durch parallelisierte Berechnung der gesamte Datensatz in weniger als 3 Monaten berechnet werden kann. Sollte sich die Anzahl der gefundenen Cluster nicht nur verdreifachen, sondern beispielsweise verzehnfachen, ergeben sich Rechenzeiten von bis zu 252 Tagen. Da das Fahrsequenz-Clustering die Fahrsequenzen konsekutiv verarbeitet, kann es bereits parallel zur Datenaufnahme gestartet werden. Somit sind auch Rechendauern von 252 Tagen bei einem Jahr Aufzeichnungsdauer noch akzeptabel.

6.1.4 Diskussion des Fahrsequenz-Clusterings

Anhand der experimentellen Ergebnisse (vgl. Abschnitt 5.1.3) wurde dargelegt, dass das Fahrsequenz-Clustering in der Lage ist, semantisch ähnliche Fahrsequenzen zu clustern. Allerdings zeigen sich auch Schwächen: ausgelöst durch die Parameterwahl sowie die Reihenfolge kommt es zu Fehlern, bei denen entweder semantisch unterschiedliche Fahrsequenzen in einem Cluster (Fehler 1. Art) oder semantisch ähnliche Fahrsequenzen in verschiedenen Clustern eingeordnet (Fehler 2. Art) werden.

Durch die konsekutive Verarbeitung der Fahrsequenzen ergibt sich eine Abhängigkeit des Clustering-Ergebnisses von der Fahrsequenz-Reihenfolge. Das Fahrsequenz-Clustering ist so entwickelt, dass es einen wachsenden Datensatz verarbeiten kann und so bereits während der Datenaufzeichnung Ergebnisse liefert. Somit unterscheidet sich das Fahrsequenz-Clustering von herkömmlichen Clustering-Methoden (z.B. k-Means) in dem zentralen Punkt, dass in herkömmlichen Clustering-Ansätzen alle Datenpunkte zu Beginn des Clusterings bekannt

sind. Würde diese Eigenschaft für das Fahrsequenz-Clustering gefordert, könnte das Fahrsequenz-Clustering erst nach vollständiger Datenaufzeichnung durchgeführt werden. Zusätzlich müsste nach jeder Vergrößerung des Datensatzes ein komplett neues Clustering gestartet werden.

Aus den beschriebenen Fehlern in den ausgewerteten Datensätzen lässt sich schließen, dass auch bei der Auswertung der Aufzeichnung eines ganzen Jahres Fehler auftreten werden. Zusätzlich zu offensichtlichen semantischen Unterschieden werden dort - analog zu den untersuchten Realdaten - auch Grenzfälle auftreten, bei denen die Korrektheit der Zuordnung nicht eindeutig ist (vgl. Abschnitt 6.1.2). Der Grund für diese Unschärfe liegt darin, dass vorab keine Beschreibung von Szenario-Unterscheidungskriterien oder gewünschten Eigenschaften von Szenarien erfolgt ist. Dies ist einerseits ein Vorteil, da durch den Verzicht auf eine Modellierung diese als Fehlerquelle ausgeschlossen ist (vgl. Anforderung (A_1)). Jedoch zeigt sich, dass durch diese Entscheidung andere Arten von Fehlern auftreten.

Somit stellt sich die Frage, inwiefern das Fahrsequenz-Clustering trotz möglicher Fehler einen Mehrwert für die Entwicklung und Validierung von automatisierten Fahrfunktionen liefert. Auch wenn die Ergebnisse nicht vollständig fehlerfrei sind, liefert das Fahrsequenz-Clustering eine Möglichkeit, den offenen Kontext in Realdaten zu strukturieren. Für die Sicherheitsargumentation ist insbesondere die Vermeidung von Fehlern 1. Art wichtig, da so einzelne, möglicherweise für die Absicherung relevante Fahrsequenzen übersehen werden können.

Dazu wird ein theoretisches Beispiel anhand von 7 Fahrsequenzen untersucht (vgl. Abbildung 6.8). Die 7 Fahrsequenzen sind als Punkte dargestellt. Semantisch ähnliche Fahrsequenzen sind durch die gleiche Farbe gekennzeichnet. Das optimale Clustering als Zielbild clustert die Fahrsequenzen anhand ihrer Semantik (vgl. Abbildung 6.8b). Bei der Wahl von zu hohen Schwellwerten tritt ein Fehler 1. Art auf (vgl. Abbildung 6.8c): die blaue Fahrsequenz wird zu den schwarzen zugeordnet und würde in der weiteren Verarbeitung nicht berücksichtigt. Im Gegensatz dazu wird die blaue Fahrsequenz bei einem zu niedrigen Schwellwert korrekt in ein eigenes Cluster einsortiert. Es entsteht jedoch auch ein Fehler 2.

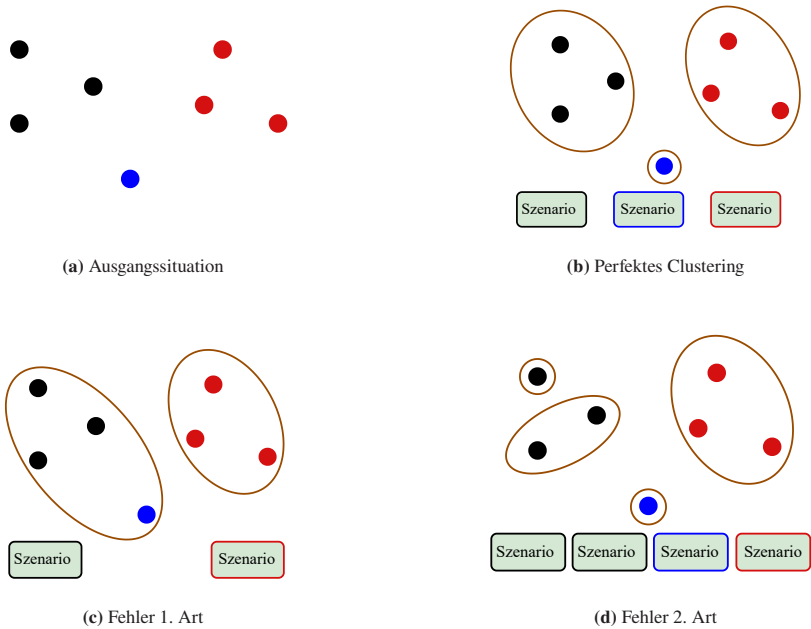


Abbildung 6.8: Darstellung der Fehlertypen beim Clustering.

Art, denn die schwarzen Fahrsequenzen werden in zwei Cluster unterteilt (vgl. Abbildung 6.8d). Dieser Fall ist für die Vollständigkeit des Szenarioraums jedoch weniger kritisch: es wird keine möglicherweise relevante Fahrsequenz und somit kein für die Absicherung wichtiges Szenario übersehen. Durch die zwei schwarzen Cluster werden jedoch auch zwei Szenarien abgeleitet, wodurch redundante Szenarien vorliegen. Dieser Effekt hat jedoch keine Auswirkungen auf die Abdeckung des Szenarioraums und kann in späteren Verarbeitungsschritten einfach detektiert werden.

Auch wenn im Fahrsequenz-Clustering Fehler auftreten können, zeigen die Versuche dass die Mehrzahl der Fahrsequenzen anhand ihrer Semantik und wie erwartet geclustert werden (im simulierten Datensatz 146 von 149). Auch bei der manuellen Untersuchung der Cluster in realen Daten zeigten sich größtenteils erwartete

Ergebnisse. Nach der Verarbeitung der Daten von einem einzigen Monat liegen bereits 19645 Cluster⁷ vor. Im Gegensatz zur Gesamtheit aller Fahrsequenzen wurde der Datensatz um 81,5 % reduziert.

74% der Cluster bestehen jedoch aus nur einer einzigen Fahrsequenz. Bei der Betrachtung der Anzahl Teilnehmer pro Cluster zeigt sich, dass die einzeln geclusterten Fahrsequenzen tendenziell aus deutlich mehr Teilnehmern bestehen als die Fahrsequenzen in größeren Clustern (vgl. Abbildung 6.9). Mit steigender An-

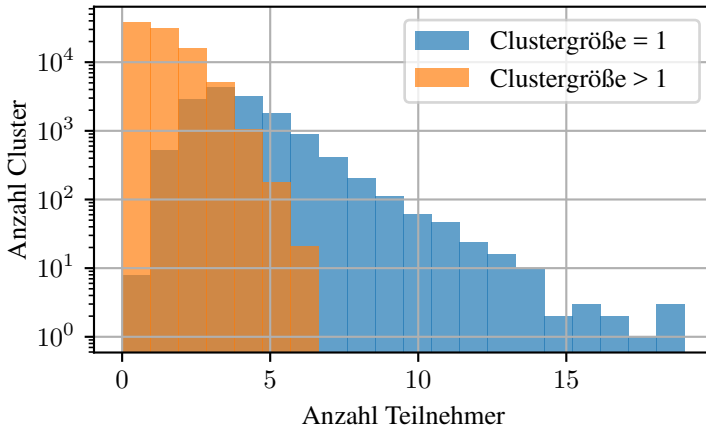


Abbildung 6.9: Histogramm der Anzahl Cluster über der Anzahl Teilnehmer pro Cluster, aufgeteilt in Clustern der Größe 1 (blau) und größeren Clustern (orange). Aufgrund der großen Unterschiede in der Gesamtanzahl ist die Y-Achse logarithmiert.

zahl Teilnehmer wird die Wahrscheinlichkeit, zwei ähnliche Fahrsequenzen zu finden, immer kleiner. Dies zeigt sich anhand folgender Abschätzung: über eine T-Kreuzung kann jedes Fahrzeug auf 6 verschiedenen Wegen fahren. Somit ergeben sich bei einem Ego-Fahrzeug und drei Teilnehmern $6^4 = 1296$ verschiedene Möglichkeiten. Durch Fußgänger, die zusätzliche Aktionen ausführen oder durch

⁷ Mit den Parametern $\Gamma_T = 6m$ und $\Gamma_E = 6m$.

nicht regelkonformes Verhalten wird die Zahl der Möglichkeiten weiter vergrößert. Somit entspricht die hohe Anzahl von Clustern mit nur einer Fahrsequenz dem erwarteten Verhalten.

6.2 Diskussion der Nutzbarkeit des Konzepts

6.2.1 Anwendung im Entwicklungsprozess

Das vorgestellte Konzept liefert während der Funktionsentwicklung verschiedene Beiträge. Es ist so gestaltet, dass der Beginn der Datenaufzeichnung und die Szenarienanalyse zu frühen Entwicklungszeitpunkten ermöglicht wird. Durch die Nutzung externer Datensätze kann der Verkehr im Betriebsbereich durchgängig aufgezeichnet werden (vgl. Abschnitt 4.1.2). Ebenso werden die Fahrsequenzen, Cluster und daraus abgeleiteten Szenarien unabhängig von einer konkreten Funktionsdefinition gebildet. Die Gestaltung des Fahrsequenz-Clusterings erlaubt die gleichzeitige Datenaufzeichnung und Auswertung. Somit liegen schon nach wenigen Tagen oder Wochen der Datenaufzeichnung erste Ergebnisse vor.

Eine Herausforderung für die praktische Anwendung des Fahrsequenz-Clusterings ist die Einstellung von passenden Parameterwerten, also den Schwellwerten Γ_T und Γ_E . Dabei sollten die Schwellwerte so eingestellt werden, dass Fehler 1. Art minimiert werden (vgl. Abschnitt 6.1.4). Eine grobe Einstellung kann anhand von kleinen Datensätzen mit einer verfügbaren Referenz-Zuordnung vorgenommen werden. Die Referenz-Zuordnung kann durch Nutzung von Simulation (vgl. Abschnitt 5.1) oder durch manuelles Labeling erzeugt werden. Inwiefern zwei Fahrsequenzen als ähnlich zu bewerten sind oder nicht, ist allerdings nicht in jedem Fall eindeutig und hängt ggf. von Details der zu entwickelnden Funktion ab (vgl. Abschnitt 6.1.2). Da die Datensammlung und -auswertung parallel zur Funktionsentwicklung verläuft, kann keine fertige und ausführbare Funktion analysiert werden. Um die Funktionsperspektive trotzdem einzubeziehen, sollten zwei Maßnahmen ergriffen werden: erstens sollten Funktionsexperten, die die Anforderungen an die zu entwickelnde Funktion kennen, mit einbezogen werden. Durch

Betrachtung unterschiedlicher Grenzfälle können Entscheidungen getroffen werden, welcher Grad der Ähnlichkeit anvisiert werden soll. Da dieses Vorgehen zeit- und personalaufwändig ist, kann es nur für wenige Fälle durchgeführt werden. Daher sollten zweitens nach Möglichkeit frühe, ausführbare Funktionsversionen in den Grenzfällen simuliert werden, während untersucht wird, inwiefern zwei unterschiedliche Cluster auch einen Unterschied für die Funktion machen. Durch diese Maßnahmen kann die Konfidenz in eine passende Schwellwertauswahl deutlich erhöht werden.

Unabhängig von der exakten Schwellwertauswahl wird die zu erwartende, große Zahl Szenarien (vgl. Tabelle 5.3) zu einer weiteren Herausforderung: möglicherweise im Entwicklungsprozess vorhandene, manuelle Schritte sind nicht auf alle Szenarien anwendbar. Somit müssen für die Entwicklung besonders relevante Szenarien selektiert werden. Für verschiedene Entwicklungsziele werden dazu unterschiedliche Größen herangezogen. Für das Entwicklungsziel „Komfort“ können beispielsweise die Fahrsequenzen mit hohen Auftretenswahrscheinlichkeiten selektiert und betrachtet werden. Ebenso können durch einfache Analysen der Trajektorien die Szenarien mit hohen Quer- bzw. Längsbeschleunigungen ausgewählt werden. Für das Entwicklungsziel „Sicherheit“ werden die in den Fahrsequenzen eines Clusters aufgetretenen Kritikalitäten (vgl. Definition 3.1) bestimmt und ausgewertet. Somit wird der Raum der relevanten Szenarien innerhalb einzelner Schritte der Funktionsentwicklung verkleinert.

Eine weitere Herausforderung ist die Umwandlung der Fahrsequenz-Cluster in spezifische Datenformate für die Nutzung im Entwicklungsprozess: Für die manuelle Ableitung von Anforderungen werden die Fahrsequenzen innerhalb eines Clusters direkt betrachtet und analysiert, eine einfache Visualisierung der Bewegungen von Teilnehmern reicht also aus. Für die Nachbildung eines Clusters in der Simulation sind jedoch ausführbare Formate notwendig, die mit gängigen Simulationswerkzeugen kompatibel sind (vgl. Abschnitt 2.3.3). Diese Formate beruhen auf einer Modellierung von Szenarien durch Abfolgen von Manövern. Die Repräsentanten der einzelnen Cluster können dann mit Konvertierungsmethoden in gewünschte Formate umgewandelt werden, beispielsweise in OpenScenario (vgl. Abschnitt 2.3.2).

6.2.2 Beitrag zur Validierung

Auftretenswahrscheinlichkeiten

Zwei maßgebliche Standards für die Durchführung der Validierung automatisierter Fahrfunktionen sind die ISO 26262 (vgl. Abschnitt 2.2.2) und die ISO 21448 (vgl. Abschnitt 2.3.4). In beiden Standards ist ein zentraler Aspekt, dass mögliche Gefährdungsereignisse beim Betrieb einer Fahrfunktion anhand ihrer Schadensschwere, Auftretenswahrscheinlichkeit sowie Beherrschbarkeit bewertet werden müssen. Das vorgestellte Konzept ermöglicht die datengetriebene Berechnung von Auftretenswahrscheinlichkeiten von Szenarien. Diese können in die Sicherheitsargumentation einfließen und diese an verschiedenen Stellen vereinfachen. Beispielsweise hat ein Szenario, das über ein Jahr hinweg nur in einer einzigen Fahrsequenz beobachtet wird, maximal eine Wahrscheinlichkeit von $1,875 \cdot 10^{-5}\%$ ⁸. Wird nun in diesem Szenario ein potentiell Gefährdungsereignis festgestellt, bei dem eine hohe Schadensschwere vorliegt, wird trotzdem ein geringeres Sicherheitslevel vergeben (vgl. Tabelle A.5). Doch auch umgekehrt kann die datenbasierte Szenarienanalyse dabei helfen, unerwartet häufig auftretende Szenarien besonders abzusichern. Wird an einer unübersichtlichen Kreuzung oft die Vorfahrt nicht gewährt, wodurch häufigere starke Bremsungen erforderlich werden, muss auf diese erhöhte Auftretenswahrscheinlichkeit durch andere Maßnahmen reagiert werden. Eine Maßnahme, welche die Schadensschwere verringern und die Kontrollierbarkeit erhöhen würde, ist die Beschränkung der maximalen Geschwindigkeit des Ego-Fahrzeugs auf dieser Kreuzung.

Schätzung der Vollständigkeit

Der Standard ISO 21448 betrachtet die Sicherheit der Sollfunktion auch in Szenarien, die sich außerhalb der Spezifikation bewegen [55]. Eine wichtige Betrachtung zur Erfüllung dieses Standards ist die Argumentation, dass die Menge der

⁸ Annahme: 16 Mio. Fahrsequenzen pro Jahr und ein Konfidenzlevel von $1 - \alpha = 0,95$.

unbekannten Szenarien klein genug ist, um kein unzumutbares Risiko beim Betrieb darzustellen. Um diese Argumentation auf Basis von Daten zu unterstützen, werden die Verfahren zur Abschätzung der Vollständigkeit (vgl. Abschnitt 4.3.2) verwendet. Sowohl durch die Analyse der Wachstumsmodelle als auch durch die Nutzung des Good-Toulmin-Schätzers wird ein Hinweis gegeben, wie groß die Zahl unbekannter Szenarien im Vergleich zu den bereits bekannten Szenarien ist. Es ist jedoch zu berücksichtigen, dass beide Verfahren Einschränkungen unterliegen.

Die erste Einschränkung betrifft die Qualität und den Horizont der Vorhersagen: Die Experimente zeigen, dass die untersuchten Wachstumsmodelle Abweichungen vom realen Kurvenverlauf aufweisen und auch in der Schätzung der zukünftigen Clusteranzahl einen relativen Fehler von mindestens 17% aufweisen. Der Trend des untersuchten Fehlers deutet darauf hin, dass eine weitere Vergrößerung des zur Kurvenanpassung verwendeten Datensatzes den relativen Fehler verringern kann. Der Good-Toulmin-Schätzer dagegen sagt die zukünftige Clusteranzahl besser vorher, kann jedoch maximal für eine Verdopplung des Datensatzes angewendet werden.

Die zweite Einschränkung, die für beide Verfahren gilt, ist das Fehlen einer Konfidenzaussage. In beiden Fällen wird eine Zahl der zu erwartenden Anzahl neuer Cluster bzw. Szenarien bei Verlängerung der Beobachtung, also Vergrößerung des Datensatzes, geliefert. Beide Verfahren sind aber nicht in der Lage, eine Bewertung der Konfidenz dieser Einschätzung mitzuliefern. Somit sind die Verfahren anfällig für zufällige Muster oder Ausreißer-Datenpunkte und liefern unter Umständen falsche Ergebnisse, welche nicht als unsicher erkannt werden.

Aufgrund der beschriebenen Einschränkungen eignen sich die Schätzverfahren angreifbar und eignen sich nicht als einziges Verfahren zur Begründung, dass kein unzumutbares Risiko durch noch unbekannte Szenarien gegeben ist. Trotzdem können sie als eine Säule für diesen Nachweis verwendet werden. Weitere mögliche Säulen werden von der ISO 21448 gegeben, beispielsweise die Simulation von Szenarien mit zufällig variierten Eingangswerten für die Erzeugung von (noch) nicht beobachteten Fahrsequenzen. Zusätzlich können existierende, vergleichbare

Systeme zur Argumentation einer ausreichenden Beobachtungsdauer herangezogen werden [55]. Dazu könnten beispielsweise Statistiken über Unfälle beim Betrieb von fahrerlosen Taxis in den San Francisco oder Phoenix verwendet werden. Zur Zulassung muss eine nachvollziehbare Argumentation vorliegen, bei der die Schwachpunkte der einzelnen verwendeten Verfahren durch andere Verfahren ausgeglichen werden. Beispielsweise kann ein Vergleich mit ähnlichen Systemen in anderen Betriebsbereichen die die Aussage der beschriebenen Schätzverfahren stützen und somit ihre Konfidenz vergrößern.

Schätzung und Vergleich des Aufwands

Die Ergebnisse zeigen, dass schon in einem Monat Datenaufzeichnung bis zu 25000 Cluster und somit Szenarien gefunden werden. Auch wenn die Rate neu gebildeter Cluster abnimmt (vgl. Abbildung 5.11), ist über ein gesamtes Jahr Datenaufzeichnung eine Zahl über von 100000 Clustern realistisch. Damit liegt auch die Anzahl der unterschiedlichen Szenarien auf einer Kreuzung in diesem Bereich. Dabei handelt es sich nur um die auf einer einzigen Kreuzung auftretenden Szenarien. Entsprechend des Prozesses zur szenariobasierten Validierung (vgl. Abschnitt 2.3.3) muss jedes dieser Szenarien in eine Beschreibung überführt und simuliert werden. Dabei wird jedes Szenario mehrfach mit leicht unterschiedlichen Verhaltensweisen der Teilnehmer simuliert. In der Literatur wird pro Szenario von 160 bis 444 konkreten, im Simulator ausgeführten Szenarien ausgegangen [65]. Um alle gefundenen Szenarien vollständig zu testen, müssen somit bis zu 44,4 Mio. Simulationen durchgeführt werden. Daraus leiten sich zwei Voraussetzungen für eine erfolgreiche Durchführung einer szenariobasierten Validierung ab:

Automatisierung Jeder Schritt im szenariobasierten Testen muss automatisiert ablaufen. Das umfasst die Ableitung ausführbarer Szenarien, die Bewertung der Funktionsleistung innerhalb eines Szenarios sowie die Detektion von Auffälligkeiten bei der Ausführung.

Parallelisierung Bei einer angenommenen Simulationszeit von 5s pro konkretem Szenario summiert sich die Gesamtsimulationsdauer bei 44,4 Mio.

konkreten Szenarien zu 7 Jahren⁹. Analog zum Fahrsequenz-Clustering müssen demnach auch die Simulationen parallelisiert werden. Eine parallele Simulation auf 100 Rechnern würde die Simulationszeit auf 26 Tage reduzieren.

Zur Bewertung des notwendigen Aufwands wird das beschriebene Vorgehen mit dem aktuell vorherrschenden, distanzbasierten Ansatz verglichen (vgl. Abschnitt 2.3.1). Diesem liegt die Annahme zugrunde, dass im Rahmen von umfassenden Erprobungsfahrten¹⁰ eine ausreichende und aussagekräftige Anzahl verschiedener Fahrsituationen für den Sicherheitsnachweis auftritt. Zur Beurteilung des Aufwands werden öffentlich verfügbare Zahlen der Firma Waymo herangezogen: im Jahr 2020 wurde ein Taxi-Service mit fahrerlosen Fahrzeugen in einem Vorort von Phoenix eingeführt [73]. Waymo gibt zwar auf einer methodischen Ebene Informationen zum Vorgehen bei der Absicherung heraus (z.B. [37]), quantitative Daten werden jedoch nicht veröffentlicht. In Kalifornien werden jedoch jedes Jahr Berichte über die gefahrene Distanz mit automatisierten Fahrzeugen auf öffentlichen Straßen veröffentlicht, sogenannte Disengagement Reports [86]. Um den Aufwand für die Absicherung der 2020 freigegebenen automatisierten Fahrfunktion der Shuttles abzuschätzen, wird die Zahl der von Waymo gefahrenen Testkilometer in den Jahren 2015-2019 betrachtet: die Gesamtdistanz betrug 6,64 Mio. Kilometern, gefahren mit bis zu 148 Fahrzeugen (in 2019) [86]. Bei einer angenommenen Durchschnittsgeschwindigkeit von 50 km/h sind das 132703 Stunden Arbeitszeit für Sicherheitsfahrer. Damit liegt Waymo drei Größenordnungen unter den theoretisch bestimmten 2,1 Milliarden Kilometern, die als notwendig für den Nachweis der Sicherheit berechnet wurden (vgl. Abschnitt 2.3.1).

Dem gegenüber steht der szenariobasierte Ansatz, bei dem alle beim Betrieb einer Funktion auftretenden Szenarien systematisch definiert und getestet werden. Dieser eröffnet die Frage, welche Szenarien konkret getestet werden müssen. Diese Frage wird vom Fahrsequenz-Clustering für einen definierten Betriebsbereich

⁹ 444 konkrete Szenarien \times 100000 Szenarien \times 5s Ausführungszeit

¹⁰ z.B. 24 Mio. km für Erprobung der C-Klasse in 2010 [18]

beantwortet, indem alle auftretenden Szenarien aus einem Datensatz extrahiert werden.

Die Datenaufnahme im Betriebsbereich benötigt ein Jahr. Die Durchführung des Fahrsequenz-Clusterings wird mit drei Monaten für die Rechenzeit abgeschätzt (vgl. Abschnitt 6.1.3). Da sie aber bereits während der Datenaufnahme stattfinden kann, erhöht sie die Gesamtzeit nicht. An das Fahrsequenz-Clustering schließt sich die Simulation der Szenarien an, deren Dauer oben zu 26 Tagen abgeschätzt wurde. Die gesamte Zeit für die Validierung mitsamt Datenaufzeichnung beträgt damit weniger als 13 Monate, liegt also deutlich unter den 5 Jahren Erprobungsfahrt von 2015-2019. Die beschriebenen Schritte gelten jedoch nur für ein Streckensegment, für die Freigabe des Betriebsbereichs müssen mehrere Streckensegmente betrachtet werden. Dies kann jedoch vollständig parallel geschehen, es muss nur eine entsprechende Anzahl an parallelen Rechenprozessen verfügbar sein (z.B. 100 pro Aufzeichnungsort).

Zusätzlich zur Beschleunigung wird die Qualität der Sicherheitsaussage größer. Waymo hat in 5 Jahren 6,64 Mio. Kilometer im realen Straßenverkehr gesammelt. Das Fahrsequenz-Clustering hat im untersuchten Segment in einem Monat bereits 5,3 Mio. Kilometer¹¹ realen Verkehr untersucht. In einem Jahr werden also 63,4 Mio. km realer Straßenverkehr pro Streckensegment untersucht. Werden in einem Betriebsbereich 34 Aufzeichnungsorte eingerichtet und somit 34 Streckensegmente ein Jahr lang beobachtet, summiert sich die beobachtete Distanz im Straßenverkehr zu 2,15 Milliarden Kilometern. Damit ist die in [118] beschriebene, notwendige Zahl von 2,1 Milliarden Kilometern mit dem beschriebenen Vorgehen erreichbar und kann mit groß angelegter Verkehrsbeobachtung (z.B. Großstädte mit mehreren hundert Aufzeichnungsorten) auch deutlich überschritten werden.

Es muss jedoch hinzugefügt werden, dass die Aussagekraft des Vergleichs mit Waymo in zwei Punkten eingeschränkt ist: Erstens wurden die von Waymo gefahrenen Kilometern mit einer aktivierten Fahrfunktion in der realen Welt getestet.

¹¹ 105704 Fahrsequenzen, im Mittel 50m zurückgelegter Strecke des Ego-Fahrzeugs pro Fahrsequenz

Beim Fahrsequenz-Clustering werden beliebige, menschliche Verkehrsteilnehmer als Ego-Fahrzeug definiert und die Funktionsausführung findet in der Simulation statt. Um einen vergleichbaren Test der Fahrfunktion zu erhalten, muss die Simulation realitätsnah sein, also alle funktionsrelevanten Elemente der Realität abbilden. Zweitens ist über die von Waymo gefahrenen Kilometer nicht exakt bekannt, an welchen Orten diese gefahren wurden. Für das Fahrsequenz-Clustering sind diese Orte exakt bekannt und definiert. Demnach ist die Vergleichbarkeit der Aussage der gesammelten Kilometer eingeschränkt.

Eine weitere Einschränkung existiert für den Vergleich der erreichbaren Distanzen mit der notwendigen Anzahl zu fahrenden Kilometern für einen distanzbasierten Nachweis. Zur Bestimmung dieser Zahl wurde berechnet, welche Distanz mindestens gefahren werden müsste, um eine Funktionsgüte besser als der menschliche Fahrer nachzuweisen [118]. Die Funktionsgüte wird hier anhand der Anzahl von Unfällen mit Getöteten gemessen. Inwiefern innerhalb der bestimmten Distanz alle relevanten Szenarien für den Test auftreten, wurde nicht diskutiert. Da die Zahl der einzige bekannte Ansatz zur Bestimmung einer notwendigen Größenordnung betrachteter Kilometer ist, wurde sie trotzdem in den Vergleich miteinbezogen.

6.3 Fazit

Bilanzierend kann konstatiert werden, dass das vorgestellte Konzept mehrere offene Punkte in der Entwicklung und Validierung von automatisierten Fahrfunktionen adressiert.

Das Fahrsequenz-Clustering ist eine Methode, die zu testenden Szenarien für einen urbanen Betriebsbereich aus Daten zu extrahieren. Damit stellt es einen zentralen Baustein für die szenariobasierte Validierung dar. In den durchgeführten Versuchen war das Fahrsequenz-Clustering in der Lage, einen Großteil (im simulierten Datensatz 146 von 149) der Fahrsequenzen entsprechend ihrer Semantik in Cluster einzuteilen. Jedoch zeigt sich auch, dass die Einteilung der

nicht in allen Fällen semantisch korrekt ist. Durch eine gezielte Auswahl von Schwellwerten kann der Einfluss des kritischen Fehlers 1. Art minimiert werden.

Die Anzahl der gefundenen Szenarien (bis zu 25000 in einem Monat) zeigt, dass die Durchführung einer szenariobasierten Validierung nur weitgehend automatisiert und parallelisiert gelingen kann. Dazu passt der beschriebene Prozess zur Datensammlung und -verarbeitung im gegebenen Betriebsbereich, welcher manuelle Schritte nur für die Auswahl der Datenorte sowie die Aufstellung der Sensorik benötigt.

Da das Fahrsequenz-Clustering gleichzeitig mit der Datenaufzeichnung ausgeführt werden kann, können auch stark befahrene Streckensegmente mit einer großen Zahl an Fahrsequenzen (im Bereich von 16 Mio.) effizient verarbeitet werden. Für das Fahrsequenz-Clustering sowie die daran anschließende Simulation ist jedoch ein hohes Maß an Parallelisierung notwendig: pro Aufzeichnungsort sollte eine dreistellige Anzahl an parallelen Rechenprozessen verfügbar sein.

Neben der Vergrößerung des Szenario-Katalogs wird aus den Ergebnissen des Fahrsequenz-Clusterings mit den Auftretenswahrscheinlichkeiten auch Wissen über bereits bekannte Szenarien generiert. Gemeinsam mit den Verfahren zur Schätzung der Vollständigkeit des Szenariokatalogs liefern sie einen wichtigen Beitrag zur Sicherheitsargumentation und bilden damit einen Baustein für die Erfüllung der „Verordnung zur Regelung des Betriebs von Kraftfahrzeugen mit automatisierter und autonomer Fahrfunktion“ und somit zur Zulassung sowie dem sicheren Betrieb.

Zur Einordnung muss jedoch erwähnt werden, dass die hier gezogenen Schlussfolgerungen nur eingeschränkt Gültigkeit besitzen. Dies begründet sich im Wesentlichen auf die folgenden Aspekte:

- Der zur quantitativen Evaluation verwendete, simulierte Datensatz besteht zwar aus einem etablierten Szenariokatalog, dem Euro NCAP-Katalog, kann jedoch die real vorhandene Variation im Straßenverkehr nicht abbilden. Dies liegt zum einen darin, dass die enthaltenen Szenarien nur aus einem Ego-Fahrzeug und sowie maximal zwei anderen Teilnehmern

bestehen. Interaktionen mit mehreren Verkehrsteilnehmern werden somit nicht untersucht. Zum anderen wurden die Daten durch ein Simulationswerkzeug generiert, wobei das Ego-Fahrzeug und die Teilnehmer durch Standard-Bewegungsmodelle generiert wurden. Somit liegt hier auch kein realistisches Verhalten vor.

- Der Datensatz der urbanen Kreuzung, der reale Aufzeichnungen des Verkehrs umfasst, korrigiert die Nachteile des simulierten Datensatzes. Jedoch kann hier keine exakte Aussage zur Qualität des Fahrsequenz-Clusterings getroffen werden, da keine korrekte Zuordnung bekannt ist. Somit werden in der manuellen Auswertung nicht unbedingt alle Fehler der 105704 Fahrsequenzen detektiert.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Die Zulassung automatisierter Fahrfunktionen im urbanen Raum ist in Deutschland seit 2021 gesetzlich möglich und seit 2022 wird die Durchführung der Zulassung durch eine Verordnung beschrieben. Diese schreibt einen Sicherheitsnachweis auf Basis von Szenarien vor, welche durch empirische Datenerhebung und -analyse gestützt werden muss. Es existieren jedoch noch keine allgemein akzeptierten Methoden, diese empirische Datenerhebung durchzuführen.

In der Forschung existieren verschiedene Ansätze, um die relevanten Szenarien für den Test automatisierter Fahrfunktionen in realen Verkehrsaufzeichnungen zu analysieren. Viele dieser Verfahren betrachten die ODD der Autobahn und nutzen die dort herrschende Strukturiertheit der Umgebung aus. Diese Strukturiertheit umfasst beispielsweise die Abwesenheit von relevantem Gegenverkehr durch bauliche Trennung sowie die Abwesenheit von Verkehrsobjekten wie Fußgängern oder Fahrradfahrern. Diese Ansätze sind durch die so erfolgten Vereinfachungen nicht übertragbar auf die urbane ODD.

Andere Ansätze versuchen, die aufgezeichneten Verkehrsdaten modellbasiert zu analysieren. Dabei liegt der Analyse ein Modell von Szenarien zugrunde, welches verschiedene Manöver und Manöverfolgen für die verschiedenen Szenario-Teilnehmer beschreibt. Die Modelle beruhen üblicherweise auf Expertenwissen. Dabei werden Manöverkataloge und Schwellwerte für den Übergang zwischen zwei Manövern definiert und die Daten auf diese Manöver hin überprüft. Durch die große Zahl zu definierender Manöver und Schwellwerte ergeben sich zahlreiche Möglichkeiten für Fehler, die nur aufwändig detektiert werden können.

In dieser Dissertation wurde daher ein Konzept für die datenbasierte Szenarienanalyse beschrieben, das urbane Verkehrsaufzeichnungen ohne die Nutzung von Modellen oder Manövern verarbeitet. Das Konzept beschreibt einen Prozess zur Datensammlung und -verarbeitung in einem definierten Betriebsbereich, von der Auswahl der Aufzeichnungsorte bis hin zur Integration der detektierten Szenarien in einen Szenariokatalog. Kernstück dieses Prozesses ist das Fahrsequenz-Clustering. Hier wird mit einer Methode zum Vergleich von zwei Fahrsequenzen sowie einer effizienten Vorgehensweise ein Clustering semantisch ähnlicher Fahrsequenzen durchgeführt. Nach dem Clustering liegen Gruppen von semantisch ähnlichen Fahrsequenzen vor. Aus diesen Gruppen werden dann Szenarien abgeleitet und in verschiedenen Schritten während der Funktionsentwicklung genutzt.

Zusätzlich zu den Szenarien selbst kann durch die datenbasierte Szenarienanalyse auch die Auftretenswahrscheinlichkeit der Szenarien bezogen auf die untersuchten Abschnitte des Betriebsbereichs ermittelt werden. Zusätzlich lassen sich anhand der Ergebnisse des Clusterings Abschätzungen treffen, wie groß die Zahl der noch unbekanntenen Szenarien auf einem untersuchten Abschnitt des Betriebsbereichs ist. Diese Erkenntnisse sind Bausteine einer Sicherheitsargumentation.

Das Fahrsequenz-Clustering wurde anhand zwei verschiedener Datensätze evaluiert. Ein simulierter Datensatz bestehend aus wenigen Fahrsequenzen, aber mit einer Referenz-Zuordnung zu festen Gruppen, wurde für eine qualitative Auswertung verwendet. Ergänzend wurde eine Auswertung anhand von einem Monat durchgehender realer Verkehrsaufzeichnung einer urbanen Kreuzung durchgeführt.

Die Ergebnisse der exemplarischen Umsetzung wurden evaluiert und diskutiert. Dabei zeigte sich, dass das Konzept einen Mehrwert in der Entwicklung und Validierung automatisierter Fahrfunktionen liefert: neben den durch die Cluster vorgegebenen Szenarien selbst wird die Auftretenswahrscheinlichkeit der Szenarien ermittelt. Zusätzlich wird durch zwei Verfahren abgeschätzt, wie viele weitere Szenarien bei weiterer Datensammlung zu erwarten sind. Dabei ergeben die Versuche, dass die Zahl der zu erwartenden Szenarien auf einer einzigen Kreuzung bei einer Monat Datenauswertung im Bereich von 19000 bis 25000 liegt, bei

längerer Beobachtungsdauer wird die Zahl weiter steigen. Anhand des Verlaufs der Cluster-Anzahl lässt sich jedoch feststellen, dass die Rate neu entdeckter Szenarien langsam aber stetig abnimmt.

Zusammenfassend stellen folgende Ergebnisse den wissenschaftlichen Beitrag dieser Dissertation dar:

- Durch die Diskussion bestehender Ansätze zur datenbasierten Szenarienanalyse wurde herausgearbeitet, dass eine für die Sicherheitsargumentation geeignete Methode zur datenbasierten Szenarienanalyse nicht allein auf Szenario-Modellen mit integrierten Manöverkatalogen beruhen darf. Um die dort möglicherweise auftretenden Verzerrungen durch menschliche Annahmen zu vermeiden, muss die Analyse der Daten durch modellfreie Ansätze vorgenommen werden. Dies gilt insbesondere für den Fall urbaner Betriebsbereiche.
- Es wurde eine Methode vorgestellt, die den Vergleich von zwei beliebigen Fahrsequenzen auf einer semantischen Ebene erlaubt. Diese bildet das Kernstück für ein Clustering semantisch ähnlicher Fahrsequenzen. Da die Vergleichsmethode auf beliebige Fahrsequenzen, unabhängig von Länge oder Anzahl der Teilnehmer, anwendbar ist, hebt sie sich vom Stand der Wissenschaft ab. Vergleichbare Methoden basieren immer auf einer Form der Vereinfachung, beispielsweise durch die Betrachtung von maximal einem anderen Verkehrsteilnehmer.
- Die Dissertation beschreibt ein Vorgehen, wie ein Katalog zu testender Szenarien auf einem spezifischen urbanen Betriebsbereich generiert der durch empirische Datenauswertung belegt wird. Somit kann eine zentrale Anforderung zur Zulassung eines Level-4-Systems in Deutschland erfüllt werden.
- Die vorgelegten Ergebnisse des Clusterings zeigen, dass die Zahl semantisch unterschiedlicher Szenarien auf einer urbanen Kreuzung sehr groß ist und eine manuelle Verarbeitung und Auswertung aller Szenarien unmöglich macht. Daraus wurde abgeleitet, dass eine szenariobasierte Validierung nur vollständig automatisiert ablaufen kann. Zusätzlich ist ein hohes Maß an

Parallelisierung notwendig, um für mehrere Aufzeichnungsorte und sowohl das Fahrsequenz-Clustering als auch die nachgelagerte Simulation der Szenarien durchzuführen. Die notwendige Anzahl paralleler Rechenprozesse liegt in einer Größenordnung von 100 Prozessen pro Aufzeichnungsort.

7.2 Ausblick

In der vorliegenden Dissertation wurde ein Konzept zur datenbasierten Szenarienanalyse vorgestellt. Ausgehend von den Ergebnissen ergeben sich unterschiedliche, anknüpfende Forschungsfragen.

Zum einen sollte das Konzept umfassender validiert werden. Dabei sollte die Datenmenge größer sein als der in dieser Dissertation verwendete Umfang von einem Monat. Durch verschiedene Maßnahmen muss der Einfluss der Fehler quantifiziert werden: dies kann manuelle Analysen einzelner Cluster oder eine parallele, regelbasierte Auswertung darstellen. Zu beachten ist, dass auch bei der regelbasierten Auswertung eine teilweise manuelle Auswertung notwendig ist, um mögliche Ungenauigkeiten der definierten Regeln zu vermeiden. Anhand der Analysen kann der Einfluss der fehlerhaft zugeordneten Fahrsequenzen quantifiziert werden und als zusätzliche Unsicherheit mit in die Auftretenswahrscheinlichkeiten eingehen. Somit kann das vorgestellte Konzept weitaus belastbarere Ergebnisse liefern und stellt einen größeren Mehrwert in der Sicherheitsargumentation dar.

Das vorgestellte Fahrsequenz-Clustering hat die zentrale Eigenschaft, unabhängig von einer konkreten Funktion alle Szenarien in einem Datensatz zu extrahieren. Das hat zur Folge, dass auch Szenarien unterschieden werden, die für eine konkrete Fahrfunktion keinen Unterschied machen. Sind beispielsweise Fußgänger abseits der Fahrspur für eine Fahrfunktion irrelevant und haben keinen Einfluss auf Entscheidungen der Fahrfunktion, so müssen sie nicht in der szenariobasierten Betrachtung berücksichtigt werden. Im Fahrsequenz-Clustering könnten die Szenarien mit und ohne Fußgänger also in das gleiche Cluster sortiert werden. Wären derart zielgenaue Kriterien für die Relevanz oder Irrelevanz von Teilnehmern für

die Fahrfunktion verfügbar, könnte die Zahl der Cluster und somit der Szenarien möglicherweise stark reduziert werden. Die Herausforderung liegt jedoch darin, diese Kriterien herauszuarbeiten. Dazu ist umfassendes Wissen über das Design der Funktion selbst notwendig. Zusätzlich entsteht, je mehr Kriterien angewandt werden, wieder die Gefahr, Fehler durch die Modellbildung selbst zu erzeugen. Die Suche nach einem Punkt, an dem durch die Zusammenführung von Funktionswissen und einer Weiterentwicklung des in dieser Dissertation präsentierten Konzepts ein optimaler und zielgenauer Szenariokatalog aus Daten extrahiert wird, kann durch den Zugewinn an Effizienz lohnenswert sein.

Eine weitere vielversprechende Forschungsrichtung ist die Betrachtung der effizienten Skalierbarkeit über eine wachsende Anzahl an Betriebsbereichen. Das vorgestellte Konzept befasst sich mit der Analyse von Szenarien in einem, spezifischen Betriebsbereich. Aus wirtschaftlicher und gesellschaftlicher Sicht sollten automatisierte Fahrfunktionen jedoch nicht auf einige wenige Betriebsbereiche beschränkt bleiben, sondern schnell in der Breite skalieren. Der Absicherungsaufwand zur Absicherung eines zweiten Betriebsbereichs sollte geringer sein als der Aufwand für den ersten Betriebsbereich und mit jedem weiteren Betriebsbereich weiter sinken. Um dies zu erreichen muss ein strukturiertes Vorgehen gewählt werden: Für alle Szenario-Schichten (vgl. Abschnitt 2.3.2) muss untersucht werden, wie sich ein neuer Betriebsbereich von einem bekannten, bereits abgesicherten Bereich unterscheidet. Dazu müssen Methoden entwickelt werden, um sowohl statische Aspekte (Straßentopologie und Verkehrsinfrastruktur) als auch dynamische Aspekte (Verkehrsteilnehmer) zu vergleichen. Für die dynamischen Aspekte können Weiterentwicklungen des Fahrsequenz-Clusterings Informationen über Auftretenswahrscheinlichkeiten von Szenarien in verschiedenen Betriebsbereichen geben. Für die statischen Aspekte werden vollständig neue Methoden benötigt, um die Ähnlichkeit von zwei verschiedenen Kreuzungen zu bewerten. Diese Methoden können auf Basis von etablierten Kartenformaten arbeiten und perspektivisch den Absicherungsaufwand für ein neues Streckensegment (z.B. eine Kreuzung) anhand der bereits abgesicherten Kreuzungen bewerten. Der Fokus dieser Dissertation liegt auf einer Betrachtung auf Objektebene und vernachlässigt den Einfluss von Fehlern durch die Umfeldwahrnehmung. Für die

Skalierung von Betriebsbereichen ist aber auch dieser Aspekt relevant: die Prädiktion des Absicherungsaufwands für einen neuen Betriebsbereich muss demnach auch wahrnehmungsrelevante Eigenschaften umfassen, beispielsweise inwiefern stark reflektierende Oberflächen vorhanden sind.

A Anhang

A.1 Fahraufgabe und Assistenzsysteme

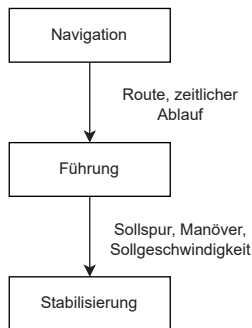


Abbildung A.1: Darstellung der drei Ebenen der Fahraufgabe nach Donges [29].

Nach Edmund Donges wird die Fahraufgabe in die drei Ebenen Navigation, Führung und Stabilisierung unterteilt (vgl. Abbildung A.1) [29]. Die Navigationsebene bezieht sich auf die zu fahrende Route und umfasst dabei Randbedingungen wie Zeitbedarf und Verkehrsbedingungen. Diese Ebene erfordert vorausschauende Planung und durchgehende Überwachung, ob die aktuelle Route mit der geplanten übereinstimmt. Die Fahrzeugführung findet auf den Ebenen der Führung und Stabilisierung statt. Die Führungsebene umfasst die Einstellung von sinnvollen Lenkwinkeln und Geschwindigkeiten durch den Fahrer, um auf der durch die Navigationsebene vorgegebenen Route zu bleiben. Dabei wird zusätzlich das Verhalten von anderen Verkehrsteilnehmern miteinbezogen. Darunter liegt die Stabilisierungsebene, in der der Fahrer durch Regeleinriffe dafür sorgt,

dass die durch die Führungsebene vorgegebenen Größen mit möglichst geringer Abweichung gehalten wird.

Auch wenn die Idee komplett automatisiert fahrender Fahrzeuge bereits in der ersten Hälfte des 20. Jahrhunderts aufkam [70], ist die tatsächliche technische Umsetzung geprägt von einer iterativen Entwicklung von Fahrerassistenzsystemen (FAS), die den Fahrer auf einer oder mehreren der drei Ebenen der Fahraufgabe unterstützen. Prominente Vertreter auf Stabilisierungsebene sind die heute in allen Neufahrzeugen vorgeschriebenen ABS und elektronischen Stabilitätsprogramme (ESP), welche den Bremsweg signifikant verkürzen bzw. das Ausbrechen des Fahrzeugs verhindern. Ein Beispiel für die Unterstützung auf der Navigationsebene sind die seit 1990 verfügbaren Navigationssysteme, welche durch die Unterstützung von globalen satellitengestützten Navigationssystemen (GNSS) und Kartendaten in der Lage sind, eine optimale Route zu bestimmen [82]. Auf der Führungsebene gehören mittlerweile Systeme wie der Adaptive Abstandregeltempomat (Adaptive Cruise Control (ACC) [126]) oder Spurhalteassistenten (LKA [2]) zur Ausstattung vieler moderner Fahrzeuge.

A.2 Funktionale Komponenten automatisierter Fahrzeuge

Die funktionalen Komponenten können nach Behere [11] in drei Kategorien eingeteilt werden (vgl. Abbildung A.2):

- die Wahrnehmung des Umfelds sowie die Lokalisierung des Fahrzeugs innerhalb dieses Umfelds,
- die Entscheidungsfindung und Planung von Aktionen und Fahrwegen,
- die Aktorik zur Ausführung der geplanten Aktionen.

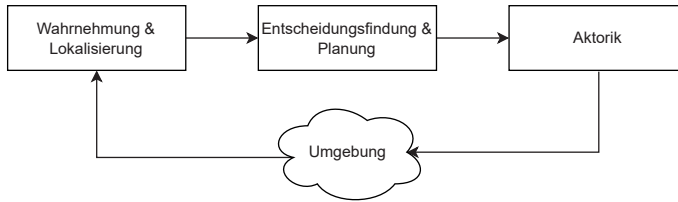


Abbildung A.2: Funktionale Komponenten eines automatisierten Fahrzeugs, Darstellung angelehnt an [52].

Umfeldwahrnehmung

Die Wahrnehmung des Umfelds erlaubt die Erkennung von Fahrspuren, die Detektion und Interpretation von Verkehrshinweisen wie Schildern und Ampeln oder die Erfassung von Hindernissen auf der Strecke. Dazu muss das Fahrzeug sowohl das eigene Umfeld erfassen als auch sich selbst darin verorten können. Die Aufgabe der Umfeldwahrnehmung lässt sich in die Teilaufgaben Erfassung des Umfelds sowie Eigenlokalisierung aufteilen [90].

Die Umfeldwahrnehmung besteht aus verschiedenen Teilaufgaben. Beispielsweise befasst sich die Objektdetektion mit der Wahrnehmung von dynamischen Objekten, beispielsweise Verkehrsteilnehmern. Die Objektklassifikation zielt darauf ab, den Typ eines detektierten Objekts zu erkennen. Die Aufgabe der Spurerkennung verarbeitet die verfügbaren Fahrbahn- und Spurmarkierungen und bestimmt daraus eine valide Fahrspur für das Fahrzeug. Diese und ähnliche Teilaufgaben werden von Algorithmen auf Basis von Sensordaten adressiert [107].

Zur Erfassung des Umfelds können verschiedene Sensorkonzepte mit unterschiedlichen Stärken und Schwächen genutzt werden. Die folgende Auflistung gibt einen Überblick über die Funktionsprinzipien sowie Stärken und Schwächen der verschiedenen Sensorkonzepte.

Radar Die Radar (*radio detection and ranging*) Technologie nutzt hochfrequente elektromagnetische Wellen um die Distanz zu Objekten zu messen. Dazu wird die sogenannte time of flight (ToF), also die Dauer, bis die Reflexion einer ausgesendeten Welle wieder am Sensor aufgefangen wird, gemessen

und in eine Distanz umgerechnet. Mit Radar werden sowohl Distanzen als auch relative Geschwindigkeiten von anderen Objekten gemessen, wobei Reichweiten von bis zu 250m erreicht werden [79].

Kamera Kamerasensoren sind von zentraler Bedeutung für automatisierte Fahrzeuge und sind in verschiedensten Auflösungen, Bildraten und Sensorgrößen verfügbar. Sie bieten die Möglichkeit, eine visuelle Repräsentation der Welt zu erstellen [21] und eignen sich daher für die Detektion und Klassifikation von Objekten, Schildern oder Straßenmarkierungen. Sie werden dazu mit Algorithmen kombiniert, um die Bilder geeignet zu interpretieren, beispielsweise auf Basis neuronaler Netze [87].

Lidar Lidarsensoren (*light detection and ranging*) verwenden analog zu Radarsensoren das ToF-Prinzip, um Distanzen zu Objekten messen zu können. Aktuelle Sensoren¹ erreichen Sichtweiten von bis zu 300m mit einer Genauigkeit von $0,1^\circ - 0,2^\circ$. Bei Regen und Nebel nimmt die Genauigkeit eines Lidarsensors ab, da die in der Luft vorhandenen Tröpfchen eine Streuung der ausgesendeten Strahlen erzeugen [71], [120].

	Radar	Kamera	Lidar
Objektdetektion	+	o	+
Objektklassifikation	-	+	o
Distanzbestimmung	+	o	+
Spurerkennung	-	+	-
Sichtweite	+	o	o
Robustheit geg. Wetter	+	o	o
Robustheit geg. Beleuchtung	+	-	+

Tabelle A.1: Bewertung von Stärken und Schwächen verschiedener Sensoren nach [130].

In einer empirischen Studie von Xique et al. wurden die Sensoren Radar, Kamera und Lidar in zentralen Aufgaben der Umfeldwahrnehmung und verschiedenen

¹ Velodyne Alpha Prime, <https://velodynelidar.com/products/alpha-prime/>, abgerufen am 11.08.2022

Umfeldbedingungen getestet [130], siehe Abbildung A.1. Daraus geht hervor, dass eine robuste technische Umsetzung einer automatisierten Fahrfunktion eine Kombination der verschiedenen Sensorkonzepte erfordert.

Eigenlokalisierung

Um die Fahraufgabe lösen zu können, müssen automatisierte Fahrzeuge ihre eigene Position kennen. Diese Anforderung erwächst einerseits aus der Notwendigkeit, eine korrekte Route für die Navigation zum Ziel zu berechnen. Andererseits muss die Eigenlokalisierung für hochautomatisierte Fahrzeuge noch genauer sein, da das Fahrzeug die eigene Position innerhalb der Fahrspur genau kennen muss. Als Grundlage für die Eigenlokalisierung werden globale Navigationssatellitensysteme (GNSS) verwendet, welche für die Navigationsebene ausreichen. Aufgrund von der nicht immer vorhandenen Verfügbarkeit und fehlender Genauigkeit von GNSS-Signalen genügt die satellitengestützte Lokalisierung jedoch nicht für eine hochautomatisierte Fahrfunktion. Daher existieren verschiedene Ansätze, diese Genauigkeit der Lokalisierung zu erhöhen. Im Fahrzeug verbaute Kameras können dazu verwendet werden, visuelle Landmarken auf existierenden Karten zu detektieren, wobei die Karten im Fahrzeug verfügbar sein müssen [11]. Zudem spielen die Ansätze des Simultaneous Localization and Mapping (SLAM) eine Rolle, bei dem das Fahrzeug durch die eigene Sensorik eine Karte aufbaut, in der es sich dann sehr genau verorten kann [75]. Sowohl die Kommunikation mit anderen Verkehrsteilnehmern (*Vehicle-to-Vehicle (V2V)*) als auch mit der Infrastruktur (*Vehicle-to-Infrastructure (V2I)*) kann für eine Verbesserung der Lokalisierung verwendet werden. Für einen umfassenden Überblick über aktuelle Ansätze der Eigenlokalisierung wird auf [72] verwiesen.

Entscheidungsfindung und Planung

Durch die Umfeldwahrnehmung und Lokalisierung wird eine interne Repräsentation der Umwelt geschaffen. Die Komponenten zur Entscheidungsfindung und

Planung befassen sich mit der fehlerlosen Navigation durch diese Welt. Die Aufgabe der Planung teilt sich in globale und lokale Planung auf [133].

Die globale Planung, welche auch als Routenplanung bezeichnet wird [135], ist verantwortlich für die Frage der zu fahrenden Route vom Startpunkt zum anvisierten Endpunkt der Fahrt und löst somit die Fahraufgabe auf Navigationsebene. Sie befasst sich dabei mit Zeitdauern von Minuten oder Stunden. Die Planung betrachtet statische Informationen wie Kartendaten, aber keine Interaktionen mit anderen Verkehrsteilnehmern.

Die lokale Planung wird in Manöverplanung und Trajektorienplanung unterteilt, welche die Fahraufgabe auf Führungs- und Stabilisierungsebene adressieren [44]. Sie stellt sicher, dass die vorgegebene Route fehlerfrei abgefahren wird. Die Manöverplanung findet auf einem taktischen Level statt und liefert Entscheidungen, welche Manöver als nächstes gefahren werden. Typische Manöver sind beispielsweise „Spur wechseln“ oder „links abbiegen“. Der Zeithorizont einer Manöverplanung liegt im Bereich weniger Sekunden [135]. Notwendig für die lokale Planung sind die Informationen über die statische Umgebung sowie über die Bewegung und das Verhalten anderer Verkehrsteilnehmer. Die statische Umgebung kann über hochgenaue Karten bereitgestellt werden, in denen sich das Fahrzeug mit oben beschriebenen Methoden zur Eigenlokalisierung selbst verortet. Andere Verkehrsteilnehmer können als dynamische Objekte beschrieben werden, die als rechteckig angenommen werden. Für jedes dieser Objekte ist dann die Größe (also die Kantenlängen des Rechtecks), die Position auf der Karte sowie ein Geschwindigkeitsvektor bekannt. Die Summe der in einer Fahrsituation vorhandenen Objekte wird Objektliste genannt [138].

Unterhalb der lokalen Planung liegt die Trajektorienplanung, die im Bereich von Millisekunden arbeitet und einen exakten Pfad bestimmt, um das gewünschte Manöver erfolgreich auszuführen. Dieser Pfad besteht aus Ortsinformationen und Zeitstempeln und wird Trajektorie genannt.

Regelung und Aktorik

Um die Vorgaben der Planungskomponente umzusetzen, muss eine Komponente zur konkreten Ausführung vorhanden sein. Dabei werden die berechneten Trajektorien in konkrete Stellgrößen umgerechnet und eingestellt. Hierzu müssen Beschleunigung, Bremsen sowie Lenkung elektrisch ansteuerbar sein und unabhängig von menschlicher Kraft einstellbar sein. Derartige Systeme werden als Drive-by-Wire bzw. Brake-by-Wire bezeichnet. Anstatt über das Gaspedal mechanische Kraft auf eine Drosselklappe oder über ein Bremspedal Kraft auf ein hydraulisches Bremssystem zu übertragen, wird über Gas- bzw. Bremspedalstellung eine elektrische Stellgröße eingestellt. Die Kraftausübung geschieht über Servomotoren oder elektromechanische Aktoren [127].

Für die Lenkung eines automatisierten Fahrzeugs kann eine schon in herkömmlichen Fahrzeugen verbaute elektromechanische Hilfskraftlenkung (Electric Power Drive, EPS) verwendet werden [77], [83]. Zukünftige Fahrzeuge mit Automatisierungsstufen ab Level 4, beispielsweise vollautomatisierte Shuttles, werden voraussichtlich kein Lenkrad besitzen, sondern direkt über einen elektrisch regelbaren Lenkmotor gesteuert werden [53].

A.3 Ergänzende Informationen zur Validierung

A.3.1 Tabellen zur Bestimmung des ASIL

Klasse			
S0	S1	S2	S3
Keine Verletzungen	Leichte oder moderate Verletzungen	Ernste und lebensbedrohliche Verletzungen (überleben wahrscheinlich)	Lebensbedrohliche Verletzungen (überleben unwahrscheinlich), tödliche Verletzungen

Tabelle A.2: Definition der verschiedenen Klassen der Schwere des potentiellen Schadens eines Gefährdungsereignisses nach ISO 26262. [54]

Klasse				
E0	E1	E2	E3	E4
Undenkbar	Sehr niedrige Wahrscheinlichkeit	Niedrige Wahrscheinlichkeit	Mittlere Wahrscheinlichkeit	Hohe Wahrscheinlichkeit

Tabelle A.3: Definition der verschiedenen Klassen der Wahrscheinlichkeit des Eintritts eines Gefährdungsereignisses nach ISO 26262. [54]

Klasse			
C0	C1	C2	C3
Generell kontrollierbar	Einfach kontrollierbar	Normalerweise kontrollierbar	Schwierig zu kontrollieren oder unkontrollierbar

Tabelle A.4: Definition der verschiedenen Klassen der Beherrschbarkeit eines Gefährdungsereignisses nach ISO 26262. [54]

Schadens- schwere	Auftritts- wahrscheinlichkeit	Kontrollierbarkeit		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

Tabelle A.5: Bestimmung eines konkreten ASIL anhand von Klassen, nach ISO 26262. [54]

A.3.2 Beispiel für Abstraktionsstufen von Szenarien

Funktionales Szenario	Abstraktes Szenario	Logisches Szenario	Konkretes Szenario
Schicht 1: Straßentopologie			
Einspurige gerade Straße innerorts mit Zebrastreifen	Straßenklasse: urban einspurig	Spurweite: [2,5, 3,5]m Gleichvert.	Spurweite: 2,75m
	Straßen-Geometrie: Gerade Zebrastr.	Kurvenradius: [500, ∞)m Gleichvert.	Kurvenradius: 2000m
	Geschw.-Begrenzung: innerorts	Geschw.-Begrenzung: [20,30,50] disk.-Vert.	Geschw.-Begrenzung: 30 km/h
Schicht 2: Verkehrsinfrastruktur			
Keine Schilder	Keine Schilder	Keine Schilder	Keine Schilder
Schicht 3: Zeitlich begrenzte Veränderungen			
Keine zeitl. begrenzten Veränderungen	Keine zeitl. begrenzten Veränderungen	Keine zeitl. begrenzten Veränderungen	Keine zeitl. begrenzten Veränderungen
Schicht 4: Dynamische Objekte			
Fahrzeug fährt auf Zebrastreifen zu, während Fußgänger die Straße kreuzt	Fahrzeug geradeaus fährt	Fahrzeug Geschw.: (20,60) Norm.Vert.	Fahrzeug Geschw.: 32 km/h
	Fußgänger Zebrastr. geht	Fußgänger Geschw.: (3,7) Norm.Vert.	Fußgänger Geschw.: 4,5 km/h
Schicht 5: Umweltbedingungen			
Sonnig Sommer Tagsüber	Wetter Information: Sommer Sonnig Tagsüber wird gesetzt auf	Helligkeit: [3k, 10k]x Gleichverteilt	Helligkeit: 7000lx
		Sicht: [15,25]km Gleichverteilt	Sicht: 18 km
		Temperatur: [15,35]°C Gleichverteilt	Temperatur: 28°C
Schicht 6: Digitale Informationen			
Keine dig. Information	Keine dig. Information	Keine dig. Information	Keine dig. Information

Abbildung A.3: Darstellung der verschiedenen Abstraktionsstufen anhand eines Beispielszenarios mit kreuzendem Fußgänger. Darstellung angelehnt an [56].

A.4 Zusätzliche Erläuterung mathematischer Grundlagen

A.4.1 Maschinelles Lernen

Eine wichtige Klasse von Methoden zur Datenanalyse im KDD-Prozess ist das Maschinelle Lernen. Eine grundlegende Definition für maschinelles Lernen gibt ist in [112] gegeben:

Definition A.1 (Maschinelles Lernen) *Ein System lernt aus Erfahrung E im Hinblick auf eine Klasse von Aufgaben T und einem Performanzmaß P , wenn seine Leistungen bei Aufgaben aus T gemessen mit P durch Erfahrung aus E steigt.*

Problemstellungen und Algorithmen des maschinellen Lernens werden in die Kategorien überwachtes und unüberwachtes Lernen eingeteilt.

Definition A.2 (Überwachtes Lernen) *Anwendungen, bei denen die Trainingsdaten aus Eingangsdatenvektor und korrespondierendem Zielvektor bestehen, der für jedes Beispiel aus den Trainingsdaten die gewünschte Lösung beschreibt.*

Definition A.3 (Unüberwachtes Lernen) *Problemstellungen, bei denen die Trainingsdaten nur aus Eingangsdaten ohne zugehörigem Zielvektor bestehen, also für kein Beispiel der Trainingsdaten eine Lösung vorliegt.*

Bezogen auf Definition A.1 unterscheiden sich die beiden Kategorien durch die Form ihrer Erfahrung: ihrer Trainingsdaten. Neben dem überwachten und unüberwachten Lernen existieren weitere Formen, beispielsweise Mischformen („semi-überwachtes Lernen“ [137]) oder andere Trainingsverfahren („bestärkendes Lernen“ [61]), die für diese Dissertation nicht relevant sind.

Vorhersage	Wahrheit	
	Positive Klasse	Negative Klasse
Positive Klasse	Richtig positiv (RP)	Falsch positiv (FP)
Negative Klasse	Falsch negativ (FN)	Richtig negativ (RN)

Tabelle A.6: Konfusionsmatrix für einfache binäre Klassifikation.

Klassifikation

Die Aufgabe der Klassifikation gehört zum Bereich des überwachten Lernens und ist die Zuordnung von durch Merkmalen definierten Objekten zu einer definierten Kategorie oder Klasse. Ein Merkmal ist eine beobachtbare oder physikalisch messbare Eigenschaft von Objekten. Die Trainingsdaten bestehen aus Merkmalsvektoren für jedes Objekt als Eingangsdaten und einer korrekten Klasse als Zielvektor. Anhand dieser Trainingsdaten kann ein Klassifikator trainiert werden, der für unbekannte Objekte die zugehörige Klasse findet. Beispiele für Klassifikationsalgorithmen sind die „Support Vector Machine“ [24] oder der „Random Forest Klassifikator“ [89].

Zur Messung der Güte eines Klassifikators werden verschiedene Maße genutzt. Grundlage für diese Maße ist die Konfusionsmatrix in Tabelle A.6, welche für einen ausgewerteten Datensatz die Anzahl der richtig und falsch klassifizierten Datenpunkte für eine binäre Klassifikation darstellt. Es existieren demnach zwei Arten von Fehlern. Die falsche Zuordnung eines negativen Datenpunkts zur positiven Klasse (falsch positiv) wird als Fehler 1. Art und die umgekehrte Zuordnung eines positiven Datenpunkts zur negativen Klasse als Fehler 2. Art bezeichnet. Aus der Anzahl der Datenpunkte in den vier Feldern der Konfusionsmatrix berechnen sich die folgenden Gütemaße: [35]

$$\text{Fehlerrate} = \frac{FN + FP}{RP + FP + FN + RN} = 1 - \text{Korrektheit}, \quad (\text{A.1})$$

$$\text{Sensitivität} = \frac{RP}{RP + FN}, \quad (\text{A.2})$$

$$\text{Spezifität} = \frac{FN}{FP + RN}, \quad (\text{A.3})$$

$$\text{Precision} = \frac{RP}{FP + RP}, \quad (\text{A.4})$$

$$\text{F-Score} = \frac{2 \times \text{Precision} \times \text{Sensitivität}}{\text{Precision} + \text{Sensitivität}}. \quad (\text{A.5})$$

Clustering

Ein Beispiel für unüberwachtes Lernen ist das *Clustering*. Hier wird versucht, Gruppierungen (im folgenden: Cluster) in den Daten zu erkennen. Die Zuordnung zu einem Cluster kann beispielsweise durch Distanzen bezüglich definierten Merkmalen bestimmt werden. Beispiele für gängige Cluster-Algorithmen sind „k-Means“ [47] oder „DBSCAN“ [105]. Im Gegensatz zur Klassifikation, wo der Zielvektor direkt mit der Klassifikation des lernenden Systems verglichen wird, ist beim Clustering die Performanz schwieriger zu bestimmen. Zur Bewertung der Performanz können verschiedene Methoden verwendet werden:

1. Durch manuelle Interpretation der Cluster wird mithilfe von Expertenwissen bewertet, ob „sinnvolle“ Cluster gebildet wurden. Inwiefern ein Cluster als „sinnvoll“ bezeichnet wird, hängt vom Anwendungsfall ab.
2. Durch Analyse der Distanzen von Datenpunkten innerhalb eines Clusters sowie die Distanz der Cluster zueinander können verschiedene Performanz-Maße bestimmt werden. Der Silhouettenkoeffizient vergleicht beispielsweise für den mittleren Abstand der Datenpunkte innerhalb eines Clusters mit dem mittleren Abstand zum nächstgelegenen anderen Cluster. Je höher der Silhouettenkoeffizient desto höhere Qualität hat das Clustering. [99]
3. Ist durch zusätzliches Wissen über den Datensatz doch eine korrekte Zuordnung der Cluster bekannt, können die gefundenen Cluster damit verglichen werden. Im Unterschied zur Klassifikation haben die Cluster jedoch keine eindeutige Identifikation („Klasse“). Daher existieren Verfahren, die den

Überschneidungsgrad von zwei Clusterings bestimmen, beispielsweise das in dieser Dissertation genutzte V-Measure (vgl. Abschnitt A.4.2).

A.4.2 V-Measure als Vergleichsmetrik für Clustering

Die V-Measure² [98] ist eine Metrik für den Vergleich eines Clusterings mit einer anderen Zuordnung, beispielsweise zu Klassen. Da das Clustering-Verfahren keine Zuordnung zu Klassen beschreibt, sondern nur eine Gleichartigkeit oder Andersartigkeit von Datenpunkten evaluiert, kann die Zuordnung nicht direkt evaluiert werden. Stattdessen wird betrachtet, inwiefern ein Datenpunkt in einem Cluster mit anderen Datenpunkten seiner Klasse ist. Die V-Measure berechnet sich aus zwei Größen, der Homogenität und der Vollständigkeit.

Definition A.4 (Homogenität) *Ein perfekt homogenes Clustering ist ein Clustering, bei dem jedes Cluster nur Datenpunkte enthält, die derselben Klasse angehören. Die Homogenität beschreibt, wie nah der Clustering-Algorithmus an dieser Perfektion ist.*

Definition A.5 (Vollständigkeit) *Ein perfekt vollständiges Clustering ist ein Clustering, bei dem alle Datenpunkte, die zur selben Klasse gehören, im selben Cluster zusammengefasst sind. Die Vollständigkeit beschreibt, wie nah der Clustering-Algorithmus an dieser Perfektion ist.*

Im trivialen Fall für perfekte Homogenität (vgl. Abbildung A.4a) ist jeder Datenpunkt in ein separates Cluster eingeteilt. Somit sind niemals Datenpunkte verschiedener Klassen innerhalb eines Clusters zu finden. Umgekehrt ist der triviale Fall für perfekte Vollständigkeit (vgl. Abbildung A.4b) die Zuordnung aller Datenpunkte zu einem einzigen Cluster. Demnach sind per Definition auch alle

² „V“ steht hier für Validität

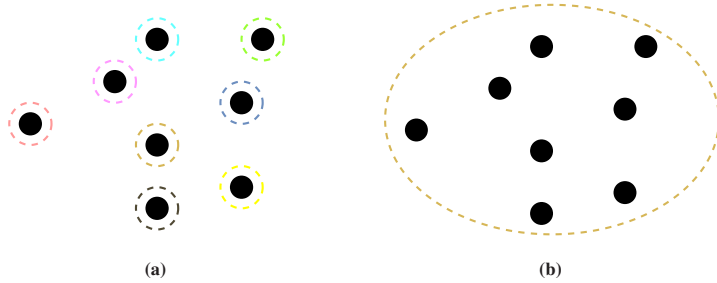


Abbildung A.4: Trivialer Fall für perfekte Homogenität (a) und perfekte Vollständigkeit (b).

Datenpunkte derselben Klasse im selben Cluster. Angenommen, es gibt N Datenpunkte, C verschiedene Klassenlabels, K Cluster und a_{ck} Datenpunkte, die zur Klasse c und zum Cluster k gehören. Dann berechnet sich die Homogenität zu

$$h = 1 - \frac{H(C|K)}{H(C)} \quad (\text{A.6})$$

mit

$$H(C|K) = - \sum_{k=1}^K \sum_{c=1}^C \frac{a_{ck}}{N} \log\left(\frac{a_{ck}}{\sum_{c=1}^C a_{ck}}\right), \quad (\text{A.7})$$

$$H(C) = - \sum_{c=1}^C \frac{\sum_{k=1}^K a_{ck}}{C} \log\left(\frac{\sum_{k=1}^K a_{ck}}{C}\right), \quad (\text{A.8})$$

und die Vollständigkeit zu

$$c = 1 - \frac{H(K|C)}{H(K)} \quad (\text{A.9})$$

mit

$$H(K|C) = - \sum_{c=1}^C \sum_{k=1}^K \frac{a_{ck}}{N} \log\left(\frac{a_{ck}}{\sum_{k=1}^K a_{ck}}\right), \quad (\text{A.10})$$

$$H(K) = - \sum_{k=1}^K \frac{\sum_{c=1}^C a_{ck}}{C} \log\left(\frac{\sum_{c=1}^C a_{ck}}{C}\right). \quad (\text{A.11})$$

$H(C)$, $H(K)$ bezeichnen hier die Entropie von C bzw. K , $H(C|K)$, $H(K|C)$ jeweils die abhängige Entropie von C und K . Die V-Messure berechnet sich dann durch einen mit β gewichteten, harmonischen Mittelwert aus h und c :

$$V_\beta = \frac{(1 + \beta)hc}{(b\beta h) + c}. \quad (\text{A.12})$$

A.4.3 Umrechnung von globalem Koordinatensystem ins Fahrzeugkoordinatensystem

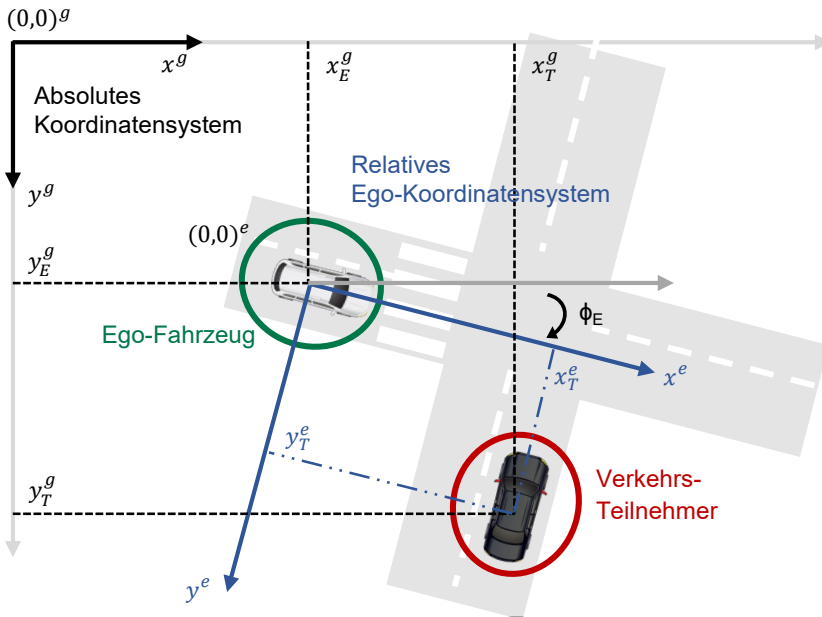


Abbildung A.5: Koordinatentransformation des absoluten Koordinatensystems der Kreuzung in das relative Koordinatensystem des Ego-Fahrzeugs zur Bestimmung der relativen Koordinaten der Szenarioteilnehmer.

Um Daten aus einem globalen Koordinatensystem in ein fahrzeugspezifisches Koordinatensystem umzurechnen, muss eine Koordinatentransformation durchgeführt werden (vgl. Abbildung A.5). Hier sollen die Koordinaten des Verkehrsteilnehmers in das Koordinatensystem des Ego-Fahrzeugs umgerechnet werden. Zur Umrechnung ist sowohl eine Translation als auch eine Rotation notwendig. Für die Translation wird die Differenz der Koordinaten des Verkehrsteilnehmers x_T^g, y_T^g mit denen des Ego-Fahrzeugs x_E^g, y_E^g im globalen Koordinatensystem berechnet:

$$\begin{pmatrix} x_{\text{diff}} \\ y_{\text{diff}} \end{pmatrix} = \begin{pmatrix} x_T^g \\ y_T^g \end{pmatrix} - \begin{pmatrix} x_E^g \\ y_E^g \end{pmatrix}.$$

Der Einfluss der Drehung des Koordinatensystems lässt sich durch Multiplikation des berechneten Differenzvektors mit einer Rotationsmatrix ermitteln. Für die Rotationsmatrix gilt

$$R(\Phi) = \begin{pmatrix} \cos \Phi & -\sin \Phi \\ \sin \Phi & \cos \Phi \end{pmatrix}. \quad (\text{A.13})$$

Für die Koordinaten eines Szenarioteilnehmers im fahrzeugspezifischen Koordinatensystem des Ego-Fahrzeugs gilt somit

$$\begin{pmatrix} x_T^e \\ y_T^e \end{pmatrix} = \begin{pmatrix} \cos \Phi & -\sin \Phi \\ \sin \Phi & \cos \Phi \end{pmatrix} * \begin{pmatrix} x_{\text{diff}} \\ y_{\text{diff}} \end{pmatrix}. \quad (\text{A.14})$$

A.5 Simulierter Testdatensatz

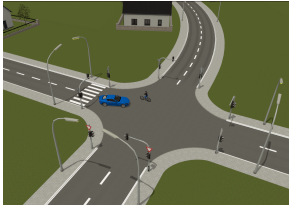
ID 1a	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Linksabbiegen mit Gegenverkehr: Radfahrer vor Ego-Fahrzeug		10.0	45301
		10.0	45301
		11.0	40900
		12.0	37400
		13.0	34400
		14.0	31801
		15.0	29600
		16.0	27701
		17.0	26001
		18.0	24500
		19.0	23201
		20.0	22001

Tabelle A.7: Liste der konkreten Szenarien in 1a mit variierten Parameterwerten.

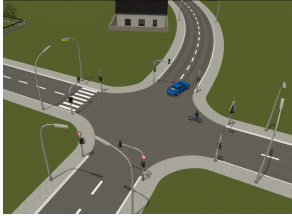
ID 1b	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Linksabbiegen mit Gegenverkehr: Radfahrer nach Ego-Fahrzeug		10.0	53101
		11.0	48300
		12.0	44501
		13.0	41201
		14.0	38400
		15.0	36000
		16.0	33901
		17.0	32000
		18.0	30401
		19.0	29001
20.0	27701		

Tabelle A.8: Liste der konkreten Szenarien in 1b mit variierten Parameterwerten.

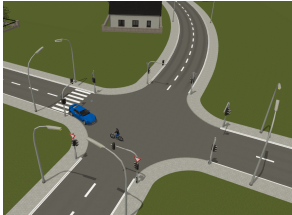
ID 2a	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Rechtsabbiegen mit Gegenverkehr: Radfahrer vor Ego-Fahrzeug		10.0	43701
		11.0	39601
		12.0	36101
		13.0	33201
		14.0	30701
		15.0	28601
		16.0	26701
		17.0	25101
		18.0	23701
		19.0	22401
20.0	21201		

Tabelle A.9: Liste der konkreten Szenarien in 2a mit variierten Parameterwerten.


ID 2b	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Rechtsabbiegen mit Gegenverkehr: Radfahrer nach Ego-Fahrzeug		10.0	51101
		11.0	46501
		12.0	42800
		13.0	39701
		14.0	37000
		15.0	34701
		16.0	32701
		17.0	31101
		18.0	29601
		19.0	28300
		20.0	27201

Tabelle A.10: Liste der konkreten Szenarien 2b mit variierten Parameterwerten.

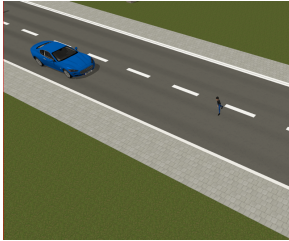
ID 3a	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Kreuzender Fußgänger von links: Fußgänger vor Ego-Fahrzeug		12.0	31701
		16.0	24300
		18.0	21701
		20.0	19501
		22.0	17701
		24.0	16300
		26.0	15101
		30.0	13201
		32.0	12500
		34.0	11901
		36.0	11300
		38.0	10901
		40.0	10501
		42.0	10201
		43.0	9800
		46.0	9300
		48.0	9101
49.0	9101		
50.0	9001		
51.0	8901		
52.0	8900		
53.0	8901		

Tabelle A.11: Liste der konkreten Szenarien in 3a mit variierten Parameterwerten.

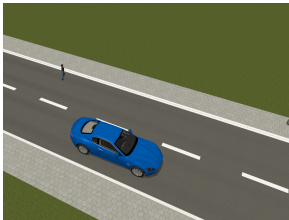
ID 3b	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Kreuzender Fußgänger von links: Fußgänger nach Ego-Fahrzeug		12.0	35001
		16.0	26800
		18.0	24101
		20.0	22001
		22.0	20300
		26.0	17701
		30.0	15901
		32.0	15201
		34.0	14601
		38.0	13701
		40.0	13300
		42.0	13001
		43.0	12701
		45.0	12401
		46.0	12201
		48.0	12001
		49.0	11900
51.0	11601		
52.0	11500		
53.0	11400		

Tabelle A.12: Liste der konkreten Szenarien in 3b mit variierten Parameterwerten.

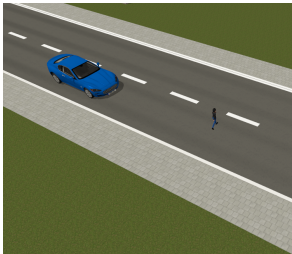
ID 4a	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Kreuzender Fußgänger von rechts: Fußgänger vor Ego-Fahrzeug		10.0	38101
		12.0	31601
		16.0	23501
		18.0	20901
		20.0	18800
		22.0	17101
		24.0	15800
		26.0	14601
		28.0	13701
		30.0	12901
		34.0	11601
		36.0	11101
		38.0	10701
		40.0	10300
		42.0	10001
		43.0	9701
		45.0	9500
46.0	9201		
48.0	9100		
49.0	9100		
50.0	9000		
51.0	8900		
52.0	8900		
53.0	8900		

Tabelle A.13: Liste der konkreten Szenarien in 4a mit variierten Parameterwerten.

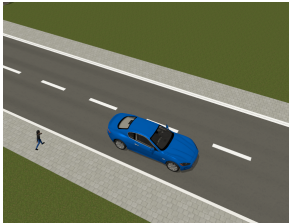
ID 4b	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Kreuzender Fußgänger von rechts: Fußgänger nach Ego-Fahrzeug		12.0	34800
		16.0	26701
		18.0	24000
		20.0	21900
		22.0	20201
		26.0	17701
		28.0	16701
		32.0	15201
		36.0	14101
		38.0	13601
		40.0	13300
		42.0	12901
		43.0	12701
		45.0	12401
		48.0	12001
		49.0	11800
50.0	11701		
51.0	11601		
53.0	11401		

Tabelle A.14: Liste der konkreten Szenarien in 4b mit variierten Parameterwerten.

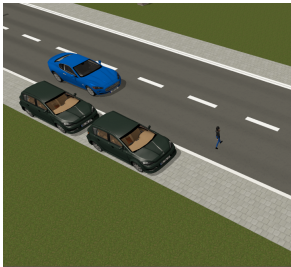
ID 5	Momentaufnahme	v_{ego} in m/s	t_{start} in ms
Kreuzender Fußgänger von links mit Verdeckung: Fußgänger vor Ego-Fahrzeug		10.0	38101
		12.0	32101
		16.0	24300
		20.0	19401
		22.0	17600
		24.0	16201
		28.0	14001
		32.0	12500
		34.0	11801
		36.0	11300
		38.0	10901
		40.0	10501
		42.0	10100
		43.0	9800
		45.0	9600
		46.0	9300
49.0	9101		
51.0	9001		
52.0	8900		
53.0	8901		

Tabelle A.15: Liste der konkreten Szenarien in 5 mit variierten Parameterwerten.

Abkürzungsverzeichnis

ABS	Antiblockiersystem
ACC	Adaptiver Abstandsregeltempomat
ASIL	Automotive Safety Integrity Level
DTW	Dynamic Time Warping
ESP	Elektronisches Stabilitätsprogramm
FAS	Fahrerassistenzsystem
GNSS	globales Navigationssatellitensystem
HiL	Hardware-in-the-Loop
KBA	Kraftfahrt-Bundesamt
KDD	Knowledge Discovery in Databases
LCSS	Longest Common Subsequence
LKA	Lane Keeping Assistant
MiL	Model-in-the-Loop
ODD	Operational Design Domain
PRI	Pedestrian Risk Index
SAE	Society of Automotive Engineers
SiL	Software-in-the-Loop
SLAM	Simultaneous Localization and Mapping
StVG	Straßenverkehrsgesetz
SUT	System Under Test
ToF	time of flight
TTC	Time to Collision
TTZ	Time to Zebra
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle

XiL X-in-the-Loop

Abbildungsverzeichnis

1.1	Visualisierung der Zielorte von Taxifahrten im New York City Metropolitan Area von Jahr 2009-2016.	4
2.1	Das V-Modell, angelehnt an die Darstellung von [30].	12
2.2	Darstellung des Beispielszenarios „kreuzender Fußgänger“ zur Einordnung der definierten Begriffe.	24
2.3	Formales Modell als Ansatz zur Beschreibung von Szenarien nach [16].	28
2.4	Visualisierung der relevanten Schritte und der vorhandenen Artefakte beim szenariobasierten Testen, angelehnt an [85].	29
2.5	Der Szenarienraum wird eingeteilt in die Kategorien bekannt/unbekannt sowie gefährlich/ungefährlich, angelehnt an [55].	33
2.6	Überblick über den Ansatz zur Herleitung des Risikos nach [55].	36
2.7	Prozess zur Entdeckung von Wissen in Datenbanken nach [38].	38
2.8	Kostenmatrix sowie kostengünstigster Pfad für zwei exemplarische Zeitreihen.	40
2.9	Resultierende Zuordnung aus der DTW-Berechnung für zwei exemplarische Zeitreihen.	41
3.1	Mengendiagramm zur Visualisierung der zentralen Annahmen zur datenbasierten Szenarioanalyse	43
3.2	Szenario-Raum zu unterschiedlichen Entwicklungsschritten	44
3.3	Klassendiagramm für den Zusammenhang zwischen Datensätzen, Fahrsequenzen und Szenarien.	50
3.4	Betrachtete Manöver bei Erdogan et al.	52
3.5	Prozess zur szenariobasierten simulationsgestützten Absicherung nach [91]	53

3.6	Codierung der räumlichen Zustände zur Abstraktion der Manöver von anderen Verkehrsteilnehmern [91]	54
3.7	Betrachtete Kreuzung von Barbier et al. zur Erkennung von Manövern und Szenarien in Grenoble, Frankreich, aus [8]	55
3.8	Darstellung der verwendeten Manöver zur Analyse urbaner Szenarien aus [48]	56
3.9	Direkte Zuordnung von Fahrsequenzen zu Szenarien	58
3.10	Clustering zur datenbasierten Szenarienanalyse.	61
3.11	Drei Beispiele für urbane Verkehrssituationen mit verschiedenen Verkehrsteilnehmertypen. Die Verkehrsteilnehmer sind immer an ihrem jeweiligen Startpunkt mitsamt ihrer darauf folgenden Bewegung dargestellt.	61
4.1	Darstellung der Artefakte im Schritt „Auswahl von Aufzeichnungsorten“.	67
4.2	Darstellung der Artefakte im Schritt „Segmentierung der Daten“.	68
4.3	Darstellung der Artefakte im Schritt „Transformation in Fahrsequenzen“.	69
4.4	Darstellung der Artefakte im Schritt „Fahrsequenz-Clustering“.	71
4.5	Darstellung der Artefakte im Schritt „Überführung in Szenario-Katalog“.	71
4.6	Vereinigung der in Abschnitt 4.1 dargestellten Schritte zur Datenaufzeichnung und -verarbeitung, angelehnt an den KDD (vgl. Abschnitt 2.4.1)	73
4.7	Diagramm für den Ablauf des kontinuierlichen Clusterings.	77
4.8	Ablaufdiagramm für den Fahrsequenz-Vergleich.	80
4.9	Beispiel für zwei ähnliche X-Werte einer Trajektorie $x_{ego,1}$ und $x_{ego,1}$	82
4.10	Drei Fahrsequenzen S_1, S_2, S_3 werden verglichen. Die Verkehrsteilnehmer sind immer an ihrem jeweiligen Startpunkt mitsamt ihrer darauf folgenden Bewegung dargestellt.	87
4.11	Mögliche Modelle für Wachstumsverhalten der Clusteranzahl über der Anzahl der betrachteten Fahrsequenzen.	93

5.1	Exemplarische Darstellung des Vorgehens zur quantitativen Erprobung anhand eines Datensatzes von simulierten NCAP-Szenarien.	98
5.2	Momentaufnahme von zwei Fällen im simuliertem Szenario mit ID 4 „Kreuzender Fußgänger von rechts“.	99
5.3	Clusteranzahl abhängig von Parametern Γ_E, Γ_T	102
5.4	V-Measure für den Vergleich der Clustering-Ergebnisse mit der Semantik der Fahrsequenzen (vgl. Tabelle 5.2).	103
5.5	Parallele-Kategorien-Diagramm für den detaillierten Vergleich des besten Clustering-Ergebnisses mit den Referenz-Gruppen.	104
5.6	Geschwindigkeitsabhängiger Relevanzbereich um Ego-Fahrzeug.	106
5.7	Anzahl Fahrsequenzen in den größten 20 Clustern für zwei Schwellwert-Parameter.	108
5.8	Häufigkeit von Clustern mit einer spezifischen Größe, Cluster mit mehr als 100 Fahrsequenzen sind abgeschnitten. Aufgrund der Größenunterschiede ist die Y-Achse logarithmisch skaliert.	108
5.9	Repräsentative Fahrsequenzen von vier exemplarischen Clustern.	110
5.10	Schätzung der Auftretenswahrscheinlichkeiten mitsamt ihrer Konfidenzbereiche	111
5.11	Wachstumsverhalten der Clusteranzahl über der Anzahl der untersuchten Fahrsequenzen.	111
5.12	Darstellung der Wachstumskurven im für die Parameteranpassung verwendeten Bereich bis 50% des gesamten Datensatzes.	113
6.1	Relative Abstände aus Ego-Perspektive zu einem über die Straße laufendem Fußgänger für zwei Fahrsequenzen inkl. Markierung der Vorbeifahr-Zeitpunkte (rotes Kreuz).	119
6.2	Relative Abstände aus Ego-Perspektive zu einem über die Straße laufendem Fußgänger für drei Fahrsequenzen S_{44}, S_{66} und S_{67} inkl. Markierung der Vorbeifahr-Zeitpunkte (rotes Kreuz).	121
6.3	Zwei verschiedene Clusterergebnisse abhängig von der Verarbeitungsreihenfolge. Korrekte Zuordnungen sind in blau markiert, falsche in rot.	121
6.4	Vergleich von zwei Fahrsequenzen mit einem Teilnehmer in Realdaten.	123

6.5	Beispiele für Fahrsequenzen mit jeweils einem Fußgänger. Das Ego-Fahrzeug ist in blau dargestellt, die Fußgänger in grün. Der dargestellte Zeitpunkt ist der Endpunkt der Ego-Trajektorie. Als irrelevant markierte Teilnehmer sind ausgegraut.	125
6.6	Entwicklung der Rechenzeit pro verarbeiteter Fahrsequenz während des Clusterings von 105704 Fahrsequenzen.	126
6.7	Aufteilung des Datensatzes in verschiedene Teilnehmertyp-Konfigurationen und anschließende parallele Ausführung des Fahrsequenz-Clusterings.	127
6.8	Darstellung der Fehlertypen beim Clustering.	131
6.9	Histogramm der Anzahl Cluster über der Anzahl Teilnehmer pro Cluster	132
A.1	Darstellung der drei Ebenen der Fahraufgabe nach Donges [29]. . . .	149
A.2	Funktionale Komponenten eines automatisierten Fahrzeugs, Darstellung angelehnt an [52].	151
A.3	Darstellung der verschiedenen Abstraktionsstufen anhand eines Beispielszenarios mit kreuzendem Fußgänger. Darstellung angelehnt an [56].	158
A.4	Trivialer Fall für perfekte Homogenität (a) und perfekte Vollständigkeit (b).	163
A.5	Koordinatentransformation des absoluten Koordinatensystems der Kreuzung in das relative Koordinatensystem des Ego-Fahrzeugs zur Bestimmung der relativen Koordinaten der Szenarioteilnehmer.	164

Tabellenverzeichnis

2.1	Übersicht über die SAE-Levels aus [100].	9
4.1	Vergleich der verschiedenen Distanzfunktionen (vgl. Abschnitt 2.4.2) in Bezug auf Erfüllung der Anforderungen.	83
4.2	Tabellarische Auflistung der Anzahl der Größen der Cluster.	95
5.1	Tabelle der gewählten Szenarien aus dem Euro NCAP.	99
5.2	Auflistung der logischen Szenarien sowie deren Unterteilung und jeweilige Anzahl konkreter Szenarien aus dem Euro NCAP Szenarienkatalog [94].	100
5.3	Ergebnisse des Fahrsequenz-Clusterings auf realen Daten.	107
5.4	Ergebnisse der angepassten Wachstumsmodelle bei der Schätzung der zukünftigen Clusteranzahl.	112
5.5	Tabellarische Auflistung der Anzahl der Clustergrößen für die erste Hälfte des Datensatzes mit $\Gamma_E = 6m$ und $\Gamma_T = 6m$	113
5.6	Zusammenfassung der drei Versuche zur Prädiktion der Clusteranzahl.	114
6.1	Anzahl Cluster für häufigste Teilnehmertyp-Konfigurationen im verwendeten Realdatensatz.	128
A.1	Bewertung von Stärken und Schwächen verschiedener Sensoren nach [130].	152
A.2	Definition der verschiedenen Klassen der Schwere des potentiellen Schadens eines Gefährdungereignisses nach ISO 26262. [54]	156
A.3	Definition der verschiedenen Klassen der Wahrscheinlichkeit des Eintritts eines Gefährdungereignisses nach ISO 26262. [54]	156

A.4 Definition der verschiedenen Klassen der Beherrschbarkeit eines
Gefährdungsereignisses nach ISO 26262. [54] 156

A.5 Bestimmung eines konkreten ASIL anhand von Klassen, nach
ISO 26262. [54] 157

A.6 Konfusionsmatrix für einfache binäre Klassifikation. 160

A.7 Liste der konkreten Szenarien in 1a mit variierten Parameterwerten. . 166

A.8 Liste der konkreten Szenarien in 1b mit variierten Parameterwerten. . 167

A.9 Liste der konkreten Szenarien in 2a mit variierten Parameterwerten. . 167

A.10 Liste der konkreten Szenarien 2b mit variierten Parameterwerten. . . 168

A.11 Liste der konkreten Szenarien in 3a mit variierten Parameterwerten. . 169

A.12 Liste der konkreten Szenarien in 3b mit variierten Parameterwerten. . 170

A.13 Liste der konkreten Szenarien in 4a mit variierten Parameterwerten. . 171

A.14 Liste der konkreten Szenarien in 4b mit variierten Parameterwerten. . 172

A.15 Liste der konkreten Szenarien in 5 mit variierten Parameterwerten. . . 173

Eigene Veröffentlichungen

- [LR1] Thilo Braun, Julian Fuchs, Felix Reisgys, Lennart Ries, Johannes Plaum, Barbara Schütt, and Eric Sax. A review of scenario similarity measures for validation of highly automated driving. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 689–696, 2023.
- [LR2] Thilo Braun, Lennart Ries, Franziska Körtke, Lara Ruth Turner, Stefan Otten, and Eric Sax. Collection of requirements and model-based approach for scenario description. In *VEHITS*, pages 634–645, 2021.
- [LR3] Christian King, Lennart Ries, Christopher Kober, Christoph Wohlfahrt, and Eric Sax. Automated function assessment in driving scenarios. In *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)*, pages 414–419. IEEE, 2019.
- [LR4] Christian King, Lennart Ries, Jacob Langner, and Eric Sax. A taxonomy and survey on validation approaches for automated driving systems. In *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–8. IEEE, 2020.
- [LR5] Jacob Langner, Johannes Bach, Lennart Ries, Stefan Otten, Marc Holzäpfel, and Eric Sax. Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1860–1866. IEEE, 2018.
- [LR6] Lennart Ries, Jacob Langner, Stefan Otten, Johannes Bach, and Eric Sax. A driving scenario representation for scalable real-data analytics with neural

networks. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2215–2222. IEEE, 2019.

- [LR7] Lennart Ries, Philipp Rigoll, Thilo Braun, Thomas Schulik, Johannes Daube, and Eric Sax. Trajectory-based clustering of real-world urban driving sequences with multiple traffic objects. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1251–1258. IEEE, 2021.
- [LR8] Lennart Ries, Maximilian Stumpf, Johannes Bach, and Eric Sax. Semantic comparison of driving sequences by adaptation of word embeddings. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020.
- [LR9] Philipp Rigoll, Lennart Ries, and Eric Sax. Scalable data set distillation for the development of automated driving functions. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3139–3145. IEEE, 2022.

Betreute Abschlussarbeiten

- [S1] Pascal Dorn. Weiterentwicklung einer methode zur bewertung der neuheit von testfahrten mit fokus auf der nachvollziehbarkeit von ki-entscheidungen, Mai 2019.
- [S2] Rouven Karp. Ähnlichkeitsanalyse von fahrszenarien auf basis des visuellen eindrucks, September 2020.
- [S3] Sabrina Kost. Entwurf und umsetzung einer methode zur analyse von kameradaten aus fahrtaufzeichnungen mithilfe einer ki-basierten anomalie-detektion, Mai 2019.
- [S4] Felix Niemeier. Nutzung von drohnen Datensätzen für die kontextbasierte analyse von urbanen fahrszenarien mit word embeddings, Mai 2021.
- [S5] Sihan Ren. Untersuchung zur nutzbarkeit von autoencodern zur szenarien-basierten analyse von fahrdaten, Januar 2019.
- [S6] Max Stumpf. Entwicklung einer methode zur kontextbasierten analyse von fahrdaten mittels adaption von sprachverarbeitungsmethoden aus der künstlichen intelligenz, November 2019.
- [S7] Alexander Uhl. Extraktion und clustering von ego-fahrszenarien aus drohnen-aufnahmen für die validierung autonomer fahrfunktionen, Januar 2021.
- [S8] Dan Wu. Clustering von multi-objekt-szenarien mit machine-learning-techniken zur validierung von automatisierten fahrfunktionen, Juli 2020.

Literaturverzeichnis

- [1] Iso/iec/ieee international standard - systems and software engineering – vocabulary. *ISO/IEC/IEEE 24765:2010(E)*, pages 1–418, 2010.
- [2] Tokihiko Akita. Lane keeping assist device for vehicle, May 8 2007. US Patent 7,216,023.
- [3] Alber Albers, Tobias Düser, and S Ott. X-in-the-loop als integrierte entwicklungs-umgebung von komplexen antriebssystemen. *Tagung Hardware-in-the-loop-Simulation, Kassel*, 16:17, 2008.
- [4] Johannes Bach, Jacob Langner, Stefan Otten, Eric Sax, and Marc Holzäpfel. Test scenario selection for system-level verification and validation of geolocation-dependent automotive control systems. In *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 203–210. IEEE, 2017.
- [5] Johannes Bach, Stefan Otten, and Eric Sax. Model based scenario specification for development and test of automated driving functions. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1149–1155. IEEE, 2016.
- [6] Gerrit Bagschik, Till Menzel, and Markus Maurer. Ontology based scene creation for the development of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1813–1820. IEEE, 2018.
- [7] Reschka Andreas Bagschik Gerrit, Menzel Till and Maurer Markus. Szenarien für entwicklung, absicherung und test von automatisierten fahrzeugen. *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, 2017.

- [8] Mathieu Barbier, Christian Laugier, Olivier Simonin, and Javier Ibanez-Guzman. Classification of drivers manoeuvre for road intersection crossing with synthetic and real data. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 224–230. IEEE, 2017.
- [9] Lukas Barthelmes, Gabriel Wilkes, Martin Kagerbauer, and Peter Vortisch. Ein on-demand-und level 4-kleinbus auf dem testfeld autonomes fahren bw-erkenntnisse aus der begleitenden haushaltsbefragung zu eva-shuttle. *Journal für Mobilität und Verkehr*, (12):36–46, 2022.
- [10] Felix Batsch, Alireza Daneshkhah, Vasile Palade, and Madeline Cheah. Scenario optimisation and sensitivity analysis for safe automated driving using gaussian processes. *Applied Sciences*, 11(2):775, 2021.
- [11] Sagar Behere and Martin Törngren. A functional architecture for autonomous driving. *WASA 2015 - Proceedings of the 2015 ACM Workshop on Automotive Software Architecture, Part of CompArch 2015*, pages 3–10, 2015.
- [12] Lukas Birkemeyer. *Sampling strategies for generating scenarios for simulation-based validation of advanced driver assistance systems*. PhD thesis, Master’s thesis. Braunschweig. <https://doi.org/10.24355/dbbs.084...>, 2021.
- [13] Julian Bock, R Krajewski, L Eckstein, J Klimke, J Sauerbier, and A Zlocki. Data basis for scenario-based validation of had on highways. In *27th Aachen colloquium automobile and engine technology*, 2018.
- [14] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1929–1934. IEEE, 2020.
- [15] Ulrik Brandes. Graphentheorie. In *Handbuch Netzwerkforschung*, pages 345–353. Springer, 2010.

- [16] Thilo Braun, Lennart Ries, Franziska Körtke, Lara Ruth Turner, Stefan Otten, and Eric Sax. Collection of requirements and model-based approach for scenario description. In *VEHITS*, pages 634–645, 2021.
- [17] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [18] Christian Brinkmann. Über 24 millionen testkilometer: Die erprobung der neuen mercedes c-klasse, 2010.
- [19] Bart Broekman and Edwin Notenboom. *Testing embedded software*. Pearson Education, 2003.
- [20] Statistisches Bundesamt. Verkehr verkehrsunfälle 2019. <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-jahr-2080700197004.pdf>, 2020.
- [21] Katie Burke. How does a self-driving car see? camera, radar and lidar sensors give autonomous vehicles superhuman vision.
- [22] Salvatore Cafiso, A Garcia Garcia, Rosario Cavarra, and MA Romero Rojas. Crosswalk safety evaluation using a pedestrian risk index as traffic conflict measure. In *3rd International Conference on Road safety and Simulation*, volume 15, 2011.
- [23] Selina Chu, Eamonn Keogh, David Hart, and Michael Pazzani. Iterative deepening dynamic time warping for time series. In *Proceedings of the 2002 SIAM International Conference on Data Mining*, pages 195–212. SIAM, 2002.
- [24] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.
- [25] Scott Corwin, Joe Vitale, Eamonn Kelly, and Elizabeth Cathles. The future of mobility: How transportation technology and social trends are creating a new business ecosystem. *Pobrane z: www2. deloitte*.

com/content/dam/Deloitte/br/Documents/manufacturing/Future_of_mobility.pdf (12.09. 2017), 2015.

- [26] Krzysztof Czarnecki. „operational design domain for automated driving systems. *Taxonomy of Basic Terms*“, *Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo, Canada*, 2018.
- [27] Erwin de Gelder, Jasper Hof, Eric Cator, Jan-Pieter Paardekooper, Olaf Op den Camp, Jeroen Ploeg, and Bart De Schutter. Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):18794–18807, 2022.
- [28] Erwin de Gelder and Jan-Pieter Paardekooper. Assessment of automated driving systems using real-life scenarios. *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.
- [29] Edmund Donges. Aspekte der aktiven sicherheit bei der führung von personenkraftwagen. *Automob-Ind*, 27(2), 1982.
- [30] Wolfgang Dröschel and Manuela Wiemers. *Das V-Modell 97: der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz*. Walter de Gruyter GmbH & Co KG, 2015.
- [31] Hala Elrofai, Daniël Worm, and Olaf Op den Camp. Scenario identification for validation of automated driving functions. In *Advanced Microsystems for Automotive Applications 2016*, pages 153–163. Springer, 2016.
- [32] Philip Elspas, Jacob Langner, Michael Aydinbas, Johannes Bach, and Eric Sax. Leveraging regular expressions for flexible scenario detection in recorded driving data. In *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–8. IEEE, 2020.
- [33] Ahmetcan Erdogan, Burak Ugranli, Erkan Adali, Ali Sentas, Eren Mungan, Emre Kaplan, and Andrea Leitner. Real-world maneuver extraction for autonomous vehicle validation: A comparative study. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 267–272. IEEE, 2019.

- [34] Alexander Eriksson and Neville A Stanton. Takeover time in highly automated vehicles: noncritical transitions to and from manual control. *Human factors*, 59(4):689–705, 2017.
- [35] Wolfgang Ertel and Wolfgang Ertel. Maschinelles lernen und data mining. *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*, pages 201–283, 2021.
- [36] Ludwig Fahrmeir, Hans Wolfgang Brachinger, Alfred Hamerle, and Gerhard Tutz. *Multivariate statistische verfahren*. Walter de Gruyter, 1996.
- [37] Francesca Favaro, Laura Fraade-Blanan, Scott Schnelle, Trent Victor, Mauricio Peña, Johan Engstrom, John Scanlon, Kris Kusano, and Dan Smith. Building a credible case for safety: Waymo’s approach for the determination of absence of unreasonable risk. *arXiv preprint arXiv:2306.01917*, 2023.
- [38] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.
- [39] H Fennel, R Gutwein, A Kohl, M Latarnik, and G Roll. Das modulare regler-und regelkonzept beim esp von itt automotive. In *7. AACHENER KOLLOQUIUM FAHRZEUG-UND MOTORENTECHNIK, 5.-7. OKTOBER 1998*, 1998.
- [40] Economic Commission for Europe. Regulation no 79 of the economic commission for europe of the united nations (un/ece) — uniform provisions concerning the approval of vehicles with regard to steering equipment. Technical report, Economic Commission for Europe, 2018.
- [41] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.
- [42] Michael R Genesereth and Nils J Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann, 2012.

- [43] Sebastian Geyer, Marcel Baltzer, Benjamin Franz, Stephan Hakuli, Michaela Kauer, Martin Kienle, Sonja Meier, Thomas Weißgerber, Klaus Bengler, Ralph Bruder, et al. Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. *IET Intelligent Transport Systems*, 8(3):183–189, 2014.
- [44] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2015.
- [45] Irving J Good and George H Toulmin. The number of new species, and the increase in population coverage, when a sample is increased. *Biometrika*, 43(1-2):45–63, 1956.
- [46] Stefan Grundhoff. Wintererprobung am polarkreis: Winterparadies, 2008.
- [47] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.
- [48] Lukas Hartjen, Robin Philipp, Fabian Schuldt, Bernhard Friedrich, and Falk Howar. Classification of driving maneuvers in urban traffic for parametrization of test scenarios. In *9. Tagung Automatisiertes Fahren*, 2019.
- [49] Florian Hauer, Ilias Gerostathopoulos, Tabea Schmidt, and Alexander Pretschner. Clustering traffic scenarios using mental models as little as possible. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1007–1012. IEEE, 2020.
- [50] Andrew Hawins. Waymo’s driverless car: ghost-riding in the back seat of a robot taxi, 2019.
- [51] John C Hayward. Near miss determination through use of a scale of danger. 1972.
- [52] Joachim Hertzberg, Kai Lingemann, and Andreas Nüchter. *Mobile Roboter: Eine Einführung aus Sicht der Informatik*. Springer-Verlag, 2012.

- [53] Calin Iclodean, Nicolae Cordos, and Bogdan Ovidiu Varga. Autonomous shuttle bus for public transportation: A review. *Energies*, 13(11):2917, 2020.
- [54] ISO. *ISO 26262 - Road Vehicles - Functional Safety*. International Organization for Standardization (ISO), 2011.
- [55] ISO. *ISO/DIS 21448 - Road vehicles - Safety of the intended functionality*. International Organization for Standardization (ISO), 2022.
- [56] ISO. *ISO/DIS 34501 - Road vehicles — Terms and definitions of test scenarios for automated driving systems*. International Organization for Standardization (ISO), 2022.
- [57] Matthias Jarke, X Tung Bui, and John M Carroll. Scenario management: An interdisciplinary approach. *Requirements Engineering*, 3:155–173, 1998.
- [58] Nina Jaschke. Konjunkturumfrage 2019/2020. Technical report, Bundesverband Deutscher Omnibusunternehmer (bdo) e.V., 2020.
- [59] Michael L Johnson and Susan G Frasier. [16] nonlinear least-squares analysis. In *Methods in enzymology*, volume 117, pages 301–342. Elsevier, 1985.
- [60] Philipp Junietz, Jan Schneider, and Hermann Winner. Metrik zur bewertung der kritikalität von verkehrssituationen und-szenarien. In *11. Workshop Fahrerassistenzsysteme*, 2017.
- [61] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
- [62] Sytze Kalisvaart, Zora Slavik, and Olaf Op den Camp. Using scenarios in safety validation of automated systems. In *Validation and Verification of Automated Systems*, pages 27–44. Springer, 2020.
- [63] Albert Karer. Modellkonzept und methodik. *Optimale Prozessorganisation im IT-Management: Ein Prozessreferenzmodell für die Praxis*, pages 21–58, 2007.

- [64] Christoph Gustav Keller, MarkusENZweiler, and Dariu M Gavrila. A new benchmark for stereo-based pedestrian detection. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 691–696. IEEE, 2011.
- [65] Christian King. *Bewertung von Fahrerassistenzsystemen im Umfeld des szenariobasierten Testens*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2023.
- [66] Christian King, Thilo Braun, Constantin Braess, Jacob Langner, and Eric Sax. Capturing the variety of urban logical scenarios from bird-view trajectories. In *VEHITS*, pages 471–480, 2021.
- [67] Christian King, Lennart Ries, Jacob Langner, and Eric Sax. A taxonomy and survey on validation approaches for automated driving systems. In *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–8. IEEE.
- [68] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125. IEEE, 2018.
- [69] Robert Krajewski, Tobias Moers, Julian Bock, Lennart Vater, and Lutz Eckstein. The round dataset: A drone dataset of road user trajectories at roundabouts in germany. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.
- [70] Fabian Kröger. Automated driving in its social, historical and cultural contexts. In *Autonomous Driving*, pages 41–68. Springer, 2016.
- [71] Vipin Kumar Kukkala, Jordan Tunnell, Sudeep Pasricha, and Thomas Bradley. Advanced driver-assistance systems: A path toward autonomous vehicles. *IEEE Consumer Electronics Magazine*, 7(5):18–25, 2018.
- [72] Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis Mccullough, and Alexandros Mouzakitis. A survey of the state-of-the-art

- localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2):829–846, 2018.
- [73] Timothy Lee. This map perfectly explains why waymo hasn't expanded faster, 2022.
- [74] Markus Lemmer, Stefan Schwab, and Sören Hohmann. The role of driver models in testing highly-automated driving: a survey. *at-Automatisierungstechnik*, 71(1):16–27, 2023.
- [75] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: science and systems*, volume 4, page 1. Citeseer, 2007.
- [76] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.
- [77] Raffaele Manca, Salvatore Circosta, Irfan Khan, Stefano Feraco, Sara Luciani, Nicola Amati, Angelo Bonfitto, and Renato Galluzzi. Performance assessment of an electric power steering system for driverless formula student vehicles. In *Actuators*, volume 10, page 165. Multidisciplinary Digital Publishing Institute, 2021.
- [78] Riccardo Mariani. An overview of autonomous vehicles safety. In *2018 IEEE International Reliability Physics Symposium (IRPS)*, pages 6A–1. IEEE, 2018.
- [79] Enrique Martí, Miguel Ángel De Miguel, Fernando García, and Joshué Pérez. A Review of Sensor Technologies for Perception in Automated Driving. *IEEE Intelligent Transportation Systems Magazine*, 11(4):94–108, 2019.
- [80] Rod McCall, Fintan McGee, Alexander Mirnig, Alexander Meschtscherjakov, Nicolas Louveton, Thomas Engel, and Manfred Tscheligi. A taxonomy of autonomous vehicle handover situations. *Transportation research part A: policy and practice*, 124:507–522, 2019.

- [81] Till Menzel, Gerrit Bagschik, and Markus Maurer. Scenarios for Development, Test and Validation of Automated Vehicles. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June:1821–1827, 2018.
- [82] Koichi Nagaki. Evolution of in-car navigation systems. *Handbook of Intelligent Vehicles*, pages 463–487, 2012.
- [83] José Eugenio Naranjo, Carlos González, Ricardo García, and Teresa de Pedro. Electric power steering automation for autonomous driving. In *International Conference on Computer Aided Systems Theory*, pages 519–524. Springer, 2005.
- [84] Christian Neurohr, Lukas Westhofen, Martin Butz, Martin Herbert Bollmann, Ulrich Eberle, and Roland Galbas. Criticality analysis for the verification and validation of automated vehicles. *IEEE Access*, 9:18016–18041, 2021.
- [85] Christian Neurohr, Lukas Westhofen, Tabea Henning, Thies de Graaff, Eike Möhlmann, and Eckard Böde. Fundamental considerations around scenario-based testing for automated driving. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 121–127. IEEE, 2020.
- [86] State of California: Department of Motor Vehicles. Disengagement reports, 2015-2019.
- [87] Niall OMahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. In *Science and Information Conference*, pages 128–144. Springer, 2019.
- [88] Alon Orlitsky, Ananda Theertha Suresh, and Yihong Wu. Optimal prediction of the number of unseen species. *Proceedings of the National Academy of Sciences*, 113(47):13283–13288, 2016.
- [89] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. A review on random forest: An ensemble classifier. In *International Conference on Intelligent*

-
- Data Communication Technologies and Internet of Things (ICICI) 2018*, pages 758–763. Springer, 2019.
- [90] Scott Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Megh-jani, You Eng, Daniela Rus, and Marcelo Ang. Perception, Planning, Control, and Coordination for Autonomous Vehicles. *Machines*, 5(1):6, 2017.
- [91] Raphael Pfeffer. *Szenariobasierte simulationsgestuetzte funktionale Absicherung hochautomatisierter Fahrfunktionen durch Nutzung von Realdaten*. PhD thesis, Dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), 2020, 2020.
- [92] Gregory Piatetsky-Shapiro. Knowledge discovery in real databases: A report on the ijcai-89 workshop. *AI magazine*, 11(4):68–68, 1990.
- [93] Alexander Poddey, Tino Brade, Jan Erik Stellet, and Wolfgang Branz. On the validation of complex systems operating in open contexts. *arXiv preprint arXiv:1902.10517*, 2019.
- [94] EUROPEAN NEW CAR ASSESSMENT PROGRAMME. Euro ncap test protocol - aeb/lss vru systems. Technical report, Euro NCAP, 2023.
- [95] Andreas Pütz, Adrian Zlocki, Julian Bock, and Lutz Eckstein. System validation of highly automated vehicles with a database of relevant traffic scenarios. *situations*, 1:E5, 2017.
- [96] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles. *IEEE access*, 8:87456–87477, 2020.
- [97] Christian Roesener, Felix Fahrenkrog, Axel Uhlig, and Lutz Eckstein. A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pages 1360–1365. IEEE, 2016.

- [98] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.
- [99] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [100] SAE. Sae j3016:201806 - surface vehicle recommended practice - taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE International*, 4970(724):1–5, 2018.
- [101] Eric Sax, editor. *Automatisiertes Testen eingebetteter Systeme in der Automobilindustrie*. Hanser, München Wien, 2008.
- [102] Eric Sax. *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*. 2008.
- [103] Jörg Schäuffele and Thomas Zurawka. *Automotive software engineering*. Springer, 2010.
- [104] Maike Scholtes, Lukas Westhofen, Lara Ruth Turner, Katrin Lotto, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Christian Neurohr, Martin Herbert Bollmann, Franziska Körtke, et al. 6-layer model for a structured description and categorization of urban traffic and environment. *IEEE Access*, 9:59131–59147, 2021.
- [105] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DbSCAN revisited, revisited: why and how you should (still) use dbSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.
- [106] Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40, 2008.

- [107] Weijing Shi, Mohamed Baker Alawieh, Xin Li, and Huafeng Yu. Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey. *Integration*, 59:148–156, 2017.
- [108] Christoph Sippl, Florian Bock, Christoph Lauer, Aaron Heinz, Thomas Neumayer, and Reinhard German. Scenario-based systems engineering: An approach towards automated driving function development. In *2019 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2019.
- [109] Herbert Stachowiak. *Allgemeine modelltheorie*. Springer, 1973.
- [110] Jan Erik Stellet, Matthias Woehrle, Tino Brade, Alexander Poddey, and Wolfgang Branz. Validation of automated driving a structured analysis and survey of approaches. In *13. Uni-DAS eV Workshop Fahrerassistenz und automatisiertes Fahren*, page 10, 2020.
- [111] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.
- [112] M Tom. Mitchell: Machine learning. *1997 Burr Ridge*, 45(37):870–877, 1997.
- [113] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, 185(4157):1124–1131, 1974.
- [114] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. Defining and substantiating the terms scene, situation, and scenario for automated driving. *IEEE 18th International Conference on Intelligent Transportation Systems*, 2015.
- [115] Itziar Urbieto, Marcos Nieto, Mikel García, and Oihana Otaegui. Design and implementation of an ontology for semantic labeling and testing: automotive global ontology (ago). *Applied Sciences*, 11(17):7782, 2021.

- [116] Andras Varhelyi. Drivers' speed behaviour at a zebra crossing: a case study. *Accident Analysis & Prevention*, 30(6):731–743, 1998.
- [117] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings 18th international conference on data engineering*, pages 673–684. IEEE, 2002.
- [118] Walther Wachenfeld and Hermann Winner. Die freigabe des autonomen fahrens. In *Autonomes Fahren*, pages 439–464. Springer Vieweg, Berlin, Heidelberg, 2015.
- [119] Lei Wang, Felix Fahrenkrog, Timo Vogt, Olaf Jung, and Ronald Kates. Prospective safety assessment of highly automated driving functions using stochastic traffic simulation. In *25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration, Detroit*, 2017.
- [120] Mian Wang, WQ Liu, YH Lu, XS Zhao, BC Song, YJ Zhang, YP Wang, CH Lian, Jun Chen, Yin Cheng, et al. Study on the measurement of the atmospheric extinction of fog and rain by forward-scattering near infrared spectroscopy. *Guang pu xue yu guang pu fen xi= Guang pu*, 28(8):1776, 2008.
- [121] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26:275–309, 2013.
- [122] Xiaozhe Wang, Kate A Smith, Rob Hyndman, and Daminda Alahakoon. A scalable method for time series clustering. *Unrefereed research papers*, 1, 2004.
- [123] Nico Weber, Christoph Thiem, and Ulrich Konigorski. Unscene: Toward unsupervised scenario extraction for automated driving systems from urban naturalistic road traffic data. *arXiv preprint arXiv:2202.06608*, 2022.

- [124] Ulrike Weinrich. *Methoden zur Bestimmung der Ausfallraten von elektrischen und elektronischen Systemen am Beispiel der Lenkungelektronik*. Springer-Verlag, 2019.
- [125] Lukas Westhofen, Christian Neurohr, Tjark Koopmann, Martin Butz, Barbara Schütt, Fabian Utesch, Birte Neurohr, Christian Gutenkunst, and Eckard Böde. Criticality metrics for automated driving: A review and suitability analysis of the state of the art. *Archives of Computational Methods in Engineering*, 30(1):1–35, 2023.
- [126] Hermann Winner, Bernd Danner, and Joachim Steinle. Adaptive cruise control. In *Handbuch Fahrerassistenzsysteme*, pages 478–521. Springer, 2009.
- [127] Hermann Winner, Stephan Hakuli, and Gabriele Wolf. *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und System für aktive Sicherheit und Komfort*. 2012.
- [128] Fabian Wolf. *Fahrzeuginformatik*. Springer, 2018.
- [129] Qin Xia, Jianli Duan, Feng Gao, Qiuxia Hu, and Yingdong He. Test scenario design for intelligent driving system ensuring coverage and effectiveness. *International Journal of Automotive Technology*, 19(4):751–758, 2018.
- [130] Ismael J Xique, William Buller, Zahra Bahrani Fard, Eric Dennis, and Benjamin Hart. Evaluating complementary strengths and weaknesses of adas sensors. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2018.
- [131] Hang Yin and Christian Berger. When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017.
- [132] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

- [133] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access*, 8:58443–58469, 2020.
- [134] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.
- [135] Sumin Zhang, Weiwen Deng, Qingrong Zhao, Hao Sun, and Bakhtiar Litkouhi. Dynamic trajectory planning for vehicle autonomous driving. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4161–4166. IEEE, 2013.
- [136] Xizhe Zhang, Siddhartha Khastgir, and Paul Jennings. Scenario description language for automated driving systems: a two level abstraction approach. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 973–980. IEEE, 2020.
- [137] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.
- [138] Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. Trajectory planning for bertha—a local, continuous method. In *2014 IEEE intelligent vehicles symposium proceedings*, pages 450–457. IEEE, 2014.
- [139] Maximilian Zipfl, Tobias Fleck, Marc René Zofka, and J Marius Zöllner. From traffic sensor data to semantic traffic descriptions: The test area autonomous driving baden-württemberg dataset (taf-bw dataset). In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020.