

Bergische Universität Wuppertal



Fakultät 7 — Maschinenbau und Sicherheitstechnik
Fachgebiet Verkehrssicherheit und Zuverlässigkeit

Masterthesis

Entwicklung heuristischer und auf Methoden des maschinellen Lernens basierender Modelle, zur Prognose des Spurwechsels für autonome Fahrzeuge.

Zur Erreichung des akademischen Grades

Master of Science

Name: Raphael Korbmacher
Matrikelnummer: 1551781
Studiengang: Master Qualitätsingenieurwesen
Anschrift: [REDACTED]

Betreuer: Jun.-Prof. Dr. Antoine Tordeux

Datum der Ausgabe: 17.07.2020
Datum der Abgabe: 14.01.2021

Eidesstattliche Erklärung

Hiermit, erkläre ich, dass ich die von mir eingereichte Abschlussarbeit (Masterthesis) selbstständig verfasst und keine andere als die angegebene Quelle und Hilfsmittel benutzt sowie Stellen der Abschlussarbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen wurden, in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Ich bin damit einverstanden, dass die Arbeit durch Dritte eingesehen und unter Wahrung urheberrechtlicher Grundsätze zitiert werden darf.

Datum und Ort: _____

Unterschrift: _____

Zusammenfassung

In dieser Masterthesis werden Spurwechsel des highD-Datensatzes mit Hilfe unterschiedlicher Ansätze prognostiziert. Der Datensatz wurde im Jahr 2018 durch den Einsatz von Drohnen über deutschen Autobahnen erstellt und enthält Fahrzeugtrajektorien. Auf Grundlage dieser Trajektorien wurden Spurwechselfprognosen mit klassischen- und empirischen Modellen erstellt und verglichen. Dadurch kann ihre Nützlichkeit für autonome Fahrzeuge beurteilt werden.

Es konnte gezeigt werden, dass sich empirische Modelle, die auf Methoden des maschinellen Lernens basieren, besser eignen um den Spurwechsel zu prognostizieren. Die beste Performance erzielten die Neuronalen Netze mit einer Prognose-Genauigkeit von 96 % für Spurwechsel auf die linke- und 95 % für den Wechsel auf die rechte Spur.

Klassische Spurwechselmodelle wie MOBIL konnten den Wechsel auf die linke Spur mit einer Genauigkeit von 93 % prognostizieren.

Abstract

The topic of this master thesis is the prediction of lane changes in the highD-Dataset, using different approaches. The given dataset was created in 2018 using drones flying above german highways and contains vehicle trajectories. Based on these trajectories lane changing prediction are generated and compared by using empirical and classical models. The benefit of these models for autonomous vehicles will be discussed.

It is shown that the empirical models, which are based on methods of machine learning, are better in predicting a lane change. The best performance is offered by the neural networks, reaching a prediction accuracy of 96% for the lane change to the left and 95% for the lane change to the right.

Classical lane changing models like MOBIL could predict the lane change to the left with an accuracy of 93%.

Abkürzungsverzeichnis

AF	Autonomes Fahrzeug
AIC	Akaike-Informationskriterium
IDM	intelligent driver model
IV	Input-Variable
LCM	Spurwechselmodell (Lane changing model)
LKW	Lastkraftwagen
logReg	Logistische Regression
MLM	Maximum-Likelihood-Methode
MW	Mittelwert
M1&4	Situation der Manöver 1 und 4
M2&3	Situation der Manöver 2 und 3
NN	Neuronale Netze
NV	Normalverteilung
OV	Output-Variable
PKW	Personenkraftwagen
Sd	Standardabweichung
SW	Spurwechsel
SWF	Spur-wechselnde Fahrzeug
ZAM	Zellulare Automaten Modelle

Abbildungsverzeichnis

Abbildung 1: Schematische Darstellung eines Spurwechsels [14]	6
Abbildung 2: Übersicht mikroskopischer LCM [2]	7
Abbildung 3: Beispiel einer Klassifikation mit Hilfe der logReg	14
Abbildung 4: Verteilung der logReg.	16
Abbildung 5: Mathematische Darstellung eines Neurons.	18
Abbildung 6: Nichtlineare Klassifikation mit NN.....	19
Abbildung 7: Die Entstehung des highD-Datensatzes [37].....	26
Abbildung 8: Situation M1&4.....	28
Abbildung 9: Situation M2&3.....	28
Abbildung 10: Kreisdiagramm der OV	31
Abbildung 11: Histogramme ausgewählter IV	33
Abbildung 12: Histogramme ausgewählter IV 2	35
Abbildung 13: Zweidimensionale Häufigkeitsverteilung der Manöver 1&4.....	37
Abbildung 14: Zweidimensionale Häufigkeitsverteilung der Manöver 2&3	38
Abbildung 15: Aufbau von Wahrheitstafeln	41
Abbildung 16: Optimierung der Parameter von MOBIL	44
Abbildung 17: Häufigkeitsverteilung des highD-Modells	47
Abbildung 18: Output der logReg	50
Abbildung 19: Verteilung des Fehlers für M1&4	52
Abbildung 20: Verteilung des Fehlers für M2&3	53
Abbildung 21: Fehler in Abhängigkeit der Anzahl der Neuronen bei M1&4.....	57
Abbildung 22: Fehler in Abhängigkeit der Anzahl der Neuronen M2&3.....	58
Abbildung 23: Verteilung des Fehlers der NN für M1&4.....	59
Abbildung 24: Verteilung des Fehlers der NN für M2&3.....	60
Abbildung 25: Am häufigsten verwendete Methoden des maschinellen Lernens	70

Tabellenverzeichnis

Tabelle 1: Übersicht der Input Variablen.....	30
Tabelle 2: Ergebnisse des zellularen Automaten Modells	42
Tabelle 3: Ergebnisse von MOBIL	45
Tabelle 4: Ergebnisse des highD- Modells	48
Tabelle 5: Ergebnisse der logReg	54
Tabelle 6: Ergebnisse der NN	61
Tabelle 7: Übersicht über alle Ergebnisse.....	63

Inhaltsverzeichnis

Eidesstattliche Erklärung	II
Zusammenfassung	III
Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Inhaltsverzeichnis	VII
1.0 Einleitung	1
1.1 Motivation und Zielsetzung	2
1.2 Aufbau	3
2.0 Theoretische Grundlagen	4
2.1 Einführung in die Spurwechselmodelle	4
2.1.1 Mikroskopische Spurwechselmodelle	5
2.2 Klassische Spurwechselmodelle	8
2.2.1 Zellulares Automaten Modell	8
2.2.2 MOBIL	10
2.3 Empirischen Modelle	12
2.3.1 Maschinelles Lernen	12
2.3.2 Logistische Regression	13
2.3.3 Neuronale Netze	18
3.0 Der Datensatz	26
3.1 Beschreibung des Datensatzes	26
3.1.2 Die Output-Variable	27
3.1.3 Die Input-Variablen	29
3.2 Deskriptive Analyse	31
3.2.1 Deskriptive Untersuchung der Output Variablen	31
3.2.2 Unterscheidung Spurwechsel / kein Spurwechsel	34
3.2.3 Variablenauswahl für die empirischen Modelle	38

4.0 Entwicklung und Anwendung der Modelle	40
4.1 Zellulare Automaten Modell	40
4.2 MOBIL	42
4.2.1 Ergebnisse von MOBIL	44
4.3 Das highD-Modell	46
4.4 Logistische Regression	49
4.4.1 Modell formulieren	49
4.4.2 Parameter schätzen und Modell auswählen	50
4.4.3 Prognose erstellen	51
4.5 Neuronale Netze	55
4.5.1 Modell formulieren	55
4.5.2 Parameter schätzen und Modell auswählen	56
4.5.3 Prognose erstellen	59
5.0 Schlussteil	62
5.1 Bewertung der klassischen Spurwechselmodelle	64
5.2 Bewertung der empirischen Modelle	67
5.3 Bezug zum autonomen Fahren	72
5.4 Fazit	74
Anhang	76
A1: Literatur	76
A2: ZAM R-Skript	81
A3: MOBIL R-Skript	82
A4: HighD-Modell R-Skript	84
A5: LogR R-Skript	85
A6: Modelauswahl NN R-Skript	86
A7: NN-Ergebnis R-Skript	88

1.0 Einleitung

Das Verhalten eines motorisierten Verkehrsteilnehmers lässt sich auf drei grundlegende Operationen reduzieren: Beschleunigen, Bremsen und Lenken [1]. Die vorliegende Masterthesis fokussiert sich ausschließlich auf die dritte Operation, das Lenken. Hierbei wird das Lenken zum Wechseln der Spur, *der Spurwechsel*, gemeint und nicht das Lenken entlang einer kurvigen Straße. Spurwechsel sind für die Verkehrsflusstheorie von hoher Bedeutung, da sie für einen großen Anteil von Verkehrsunfällen und Verkehrsstaus mitverantwortlich sind [2]. Dies gilt insbesondere auch für autonome Fahrzeuge, da zeitnahes Erkennen von Spurwechseln sowohl die Sicherheit als auch den Fahrkomfort erhöhen [3].

Ein zeitnahes Erkennen von Spurwechseln soll in dieser Ausarbeitung über Prognosen erfolgen, die mit Hilfe von zwei verschiedenen Ansätzen entwickelt werden.

Im ersten Ansatz werden klassische Spurwechselmodelle, die sonst hauptsächlich für Verkehrssimulationen eingesetzt werden, verwendet [2]. Bei diesen klassischen Spurwechselmodellen handelt es sich um mathematische Modelle, mit denen versucht wird, die Spurwechselentscheidungen auf einfache Heuristiken zu reduzieren oder wesentliche Parameter zu finden, mit denen sich der Spurwechsel formal beschreiben lässt. Im Rahmen dieses Ansatzes werden MOBIL und ein zelluläres Automaten Modell angewendet.

Im zweiten Ansatz werden empirische Modelle implementiert. Dabei handelt es sich um statistische Modelle, die die Ausgangsinformationen (Input) verarbeiten und den Beobachtungen Wahrscheinlichkeiten zuordnen. Im Rahmen des zweiten Ansatzes werden mit der logistischen Regression und den Neuronalen Netzen zwei empirische Modelle vorgestellt, die auf Methoden des maschinellen Lernens basieren.

Die Güte der vorgestellten Modelle wird anhand des highD-Datensatzes demonstriert. Dieser enthält Fahrzeugtrajektorien, die mit einer Drohne über deutschen Autobahnen aufgenommen wurden. Insgesamt liegen 11783 Beobachtungen vor, und es soll prognostiziert werden, bei welchen Beobachtungen zeitnah ein Spurwechsel eintritt [4].

1.1 Motivation und Zielsetzung

Autonomes Fahren ist nach der Definition des Verbandes der Automobilindustrie (VDA) die letzte Stufe automatisierter Fahrfunktionen [5]. Auf der Stufe 0 fährt nur der Fahrer, auf der Stufe 1 wird dem Fahrer assistiert, und in gewissen Situationen die Längs- *oder* Querführung des Fahrzeugs übernommen (Bsp. ACC). Auf der Stufe 2 liegt teilautomatisiertes Fahren vor, und in gewissen Situationen können Längs- *und* Querführung übernommen werden (Bsp. Stau-assistent). Auf der Stufe 3 liegt hochautomatisiertes Fahren vor, bei dem der Fahrer das System zeitweise (Bsp. Auf der Autobahn) nicht mehr zu überwachen braucht. Bei der Stufe 4 liegt vollautomatisiertes Fahren vor, und der Fahrer überlässt die Kontrolle vollständig dem System. Autonomes Fahren liegt dann auf der Stufe 5 vor, bei der der Fahrer nicht mehr benötigt wird. Das System ist auch ohne Fahrer vollständig funktionsfähig [6].

Für die Verkehrssicherheit ist automatisiertes- und autonomes Fahren von höchster Relevanz. Die Zukunft der Mobilität wird von diesen Technologien wohl entscheidend geprägt werden. Zurzeit beruhen ca. 90% der Verkehrsunfälle mit Toten oder Schwerverletzten auf menschliches Fehlverhalten, und es ist daher zu erwarten, dass automatisiertes Fahren die Zahl der Verkehrsunfälle stark reduzieren wird [6].

Auch im Bereich der Verkehrseffizienz kann mit signifikanten Verbesserungen gerechnet werden. STERN ET AL. (2017) zeigen, dass auch nicht-autonome Fahrzeuge davon profitieren, wenn autonome Fahrzeuge (AF) im Verkehr vorhanden sind. Schon ein Anteil von 5% autonomer Fahrzeuge reicht aus, um den Verkehrsfluss deutlich zu verbessern und das Unfallrisiko sowie den Treibstoffverbrauch zu reduzieren. [7]

Die Prognosen, die in dieser Ausarbeitung erstellt werden, haben das Ziel, Fahrerassistenzsysteme, die beim autonomen Fahren oder auch schon auf niedrigeren Stufen der automatisierten Fahrfunktionen verwendet werden, zu verbessern. Ähnliches erarbeiteten DOU ET AL. (2016), KUMAR ET AL. (2013) oder LEE ET AL. (2017), die Methoden vorstellten, um Prognosen zu erstellen, die in Advanced Driver Assistance Systems (ADAS) oder Adaptive Cruise Control (ACC) Systems verwendet werden können [3, 8, 9].

Durch die Fähigkeit der Assistenzsysteme den Spurwechsel schon zu prognostizieren bevor er stattfindet, sollen abruptes Abbremsen und die Entstehung gefährlicher Situationen reduziert werden. Dadurch wird ein komfortableres Fahrverhalten möglich, der gesamte Verkehrsfluss verbessert sich und Unfälle können verhindert werden [8].

Neben diesem praktischen Nutzen, der sich aus dieser Ausarbeitung ergeben soll, wird auch ein theoretischer Nutzen verfolgt. Durch die Verwendung unterschiedlicher Ansätze ist eine Gegenüberstellung möglich. Letztendlich kann dadurch bewertet werden, welcher der Ansätze besser geeignet ist, diese Herausforderung zu erfüllen.

1.2 Aufbau

Um diese Ziele zu verfolgen, werden in **Kapitel 2** die theoretischen Grundlagen geschaffen. Zunächst wird der erste Ansatz über die Erstellung von klassischen Spurwechselmodelle erläutert. Dafür werden Spurwechselmodelle allgemein sowie die Untergruppe der mikroskopischen Spurwechselmodelle vorgestellt. Anschließend werden mit MOBIL und einem zellularen Automaten Modell zwei klassische Spurwechselmodelle inhaltlich sowie formal beschrieben. Danach wird dann der zweite Ansatz dieser Ausarbeitung erläutert. Bei diesem werden die Prognosen mit Hilfe der empirischen Modelle erstellt. Ein Modell basiert auf der logistischen Regression und das andere auf Neuronalen Netzen. Empirische Modelle benötigen zwingend Daten.

Der in dieser Ausarbeitung verwendete Datensatz wird ausführlich in **Kapitel 3** behandelt. Zunächst werden die Entstehung des Datensatzes und dann die Variablen, die er beinhaltet, erläutert. Außerdem wird in diesem Kapitel eine deskriptive Analyse der Daten durchgeführt, die Erkenntnisse offenbart, welche bei der Erstellung der Prognosen genutzt werden.

Die Entstehung der Prognosen erfolgt schließlich in **Kapitel 4**. Zuerst werden die klassischen Spurwechselmodelle und anschließend die empirischen Modelle entwickelt und die Ergebnisse präsentiert.

Im letzten **Kapitel 5** werden die Prognosen bewertet und die verschiedenen Ansätze verglichen. Zudem wird Bezug auf das autonome Fahren genommen. Die Ausarbeitung schließt mit dem Fazit.

2.0 Theoretische Grundlagen

Die theoretischen Grundlagen, die in diesem Kapitel beschrieben werden, dienen dazu die Unterschiede der Ansätze zu verdeutlichen und die Funktionsweise der Modelle zu erläutern.

Dazu werden zunächst allgemein die Spurwechselmodelle (LCM) beschrieben, mit Fokus auf den mikroskopischen LCM, die für die Prognose des Verhaltens einzelner Fahrzeuge relevant sind. Anschließend wird ein zellulares Automaten Modell und MOBIL vorgestellt.

Es folgen die empirischen Modelle, die zunächst von den klassischen LCM abgegrenzt werden. Die empirischen Modelle verwenden in der Regel Methoden des maschinellen Lernens, um Prognosen des Spurwechsels (SW) zu erstellen [10].¹ Daher wird zunächst das maschinelle Lernen allgemein beschrieben, bevor dann anschließend die folgenden zwei gängigen Methoden vorgestellt werden: Die logistische Regression und die Neuronalen Netze.

2.1 Einführung in die Spurwechselmodelle

Ein SW ist ein Fahrmanöver, bei dem das betrachtete Fahrzeug von der aktuellen Fahrspur auf eine benachbarte Spur wechselt. Die Fachdisziplin, die sich mit Intentionen, Auswirkungen, Modellierungen und der Prognose von SW beschäftigt, ist die Verkehrsdynamik [1]. In der Verkehrsdynamik wird die Dynamik des Straßenverkehrs mit Hilfe von mathematischen Modellen beschrieben, um dadurch Erkenntnisse über Kenngrößen motorisierter oder nicht motorisierter Einheiten zu gewinnen. Dabei wird zwischen mikroskopischen- und makroskopischen Modellen unterschieden.

Bei den mikroskopischen Modellen wird das Verhalten einzelner Verkehrsteilnehmer untersucht. Es werden Messgrößen wie aktuelle Position, Geschwindigkeit, Beschleunigung und der Abstand zu umliegenden Fahrzeugen analysiert. Dadurch soll das Verständnis über gewisse Verkehrssituationen oder Situationsänderungen verbessert werden, um Auswirkungen von Tempolimits oder anderen Interaktionen der Verkehrsteilnehmer abschätzen und simulieren zu können [1].

Bei den makroskopischen Modellen wird nicht das Verhalten einzelner Fahrzeuge, sondern die Aggregation zu einem Fahrzeugkollektiv betrachtet [1]. Es werden makroskopischen Kenngrößen wie Verkehrsdichte, mittlere Geschwindigkeit oder Verkehrsfluss analysiert. Diese makroskopischen Kenngrößen eignen sich zur Untersuchung von Phänomenen wie der Staubildung, aber auch der Ausbreitungsgeschwindigkeit und Entstehung von Stop-and-Go-Wellen [11, 12].

¹ “Large part of those methods use machine learning techniques” [10].

LCM können grundsätzlich sowohl mikroskopisch als auch makroskopisch entwickelt werden, wobei mikroskopische LCM deutlich häufiger verwendet [2]. Da bei den makroskopischen Modellen die Fahrzeuge nicht einzeln betrachtet werden, können dort auch keine individuellen SW Prognosen gemacht werden. Dies ist nur bei den mikroskopischen Modellen möglich, wo der Fokus auf dem Verhalten einzelner Fahrzeuge liegt [13].

Doch neben der Unterscheidung zwischen mikroskopischen- und makroskopischen Modellen gibt es noch weitere Kriterien, worin sich die Modelle unterscheiden. Zum einen ist es wichtig zu wissen, ob ein Modell den Verkehr auf einer Autobahn oder in der Stadt erklären soll. Hier ergeben sich logischerweise einige Diskrepanzen, da die Gründe für einen SW auf der Autobahn zumeist andere sind, als die in der Stadt. Zum andern ist die Unterscheidung zwischen asymmetrischen- und symmetrischen Verkehr grade für diese Ausarbeitung von Bedeutung. [2]. Asymmetrischer Verkehr (auch einfach Rechtsfahrgebot genannt) liegt vor, wenn die Fahrzeuge angehalten sind sich auf der rechten Spur zu halten, solange sie nicht ein vorausfahrendes Fahrzeug überholen wollen [1]. Auf deutschen Autobahnen gilt das Rechtsfahrgebot, wodurch der Datensatz, der später eingeführt wird, den asymmetrischen Verkehr beschreibt. Liegt ein symmetrischer Verkehr vor, sind die Motivationen für den Spurwechsel andere, wie sich im Laufe der Arbeit herausstellen wird. Bei symmetrischen Verkehr darf ein Auto auch die Spur nach rechts wechseln um schneller fahren zu können, wogegen dies bei asymmetrischen Verkehr nur bei einem Wechsel nach links grundsätzlich erlaubt ist.

2.1.1 Mikroskopische Spurwechselmodelle

Mikroskopische LCM gehen hauptsächlich den Fragen nach *wann* ein Fahrzeug die Spur wechselt oder aber *wieso* ein Fahrzeug die Spur in einer definierten Situation wechselt. Die erste Frage ist interessant, da durch ihre Beantwortung die Prognose des SW möglich werden soll. Die Beantwortung der zweiten Frage ermöglicht realistische Simulationen und soll das allgemeine Verständnis des Verkehrs verbessern.

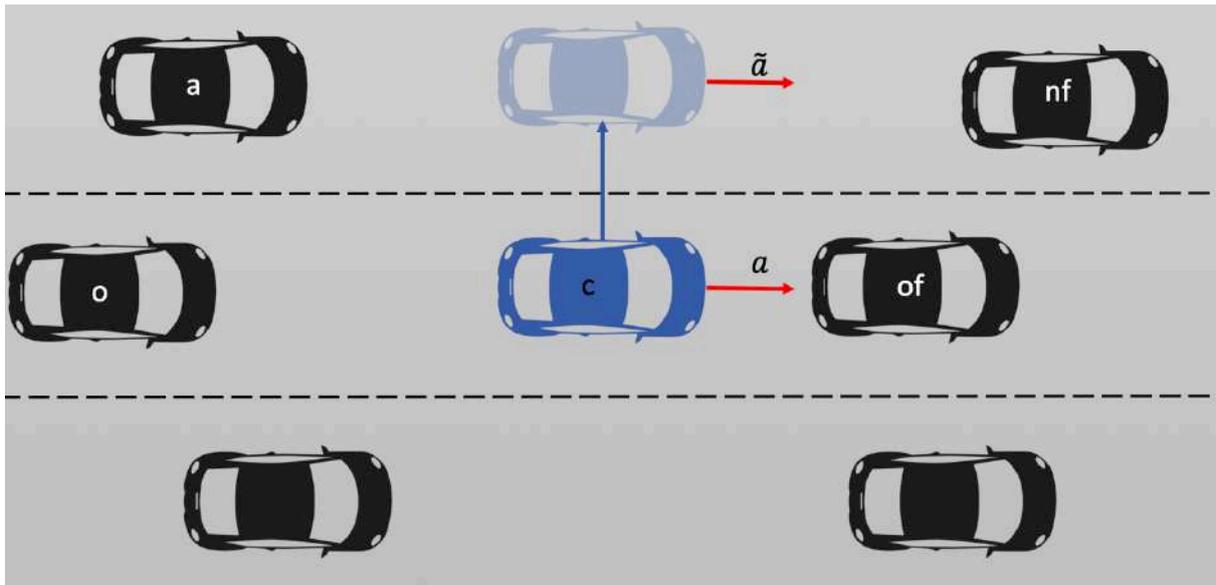


Abbildung 1: Schematische Darstellung eines Spurwechsels [14]

Abbildung 1 zeigt ein Szenario eines mikroskopischen LCM, bei dem das blaue Fahrzeug die Spur nach links wechselt, da seine aktuelle Beschleunigung a nicht mit seiner Wunschbeschleunigung \tilde{a} übereinstimmt [14]. Daher entscheidet sich das Auto im Rahmen dieses Beispiel-Modells zu einem SW, wenn die mögliche Beschleunigung auf der anderen Spur näher an der Wunschbeschleunigung liegt, als auf der aktuellen Spur.

Die Antwort auf die Frage *wieso* das Auto die Spur wechselt wird in diesem Modell also damit beantwortet, dass die mögliche Beschleunigung auf der anliegenden Spur eher der Wunschbeschleunigung entspricht. Diese Art von möglicher Antwort bzw. die damit ausgedrückte Motivation kann einfach modelliert werden. Solche Modelle finden meist Anwendung in Verkehrssimulationen [15].

In Abbildung 2 werden einige mikroskopische LCM, die zur Anwendung in Verkehrssimulationen genutzt werden, dargestellt.²

² „The main focus of this paper is to review and compare microscopic lane-changing models for computer simulation.“ [2]

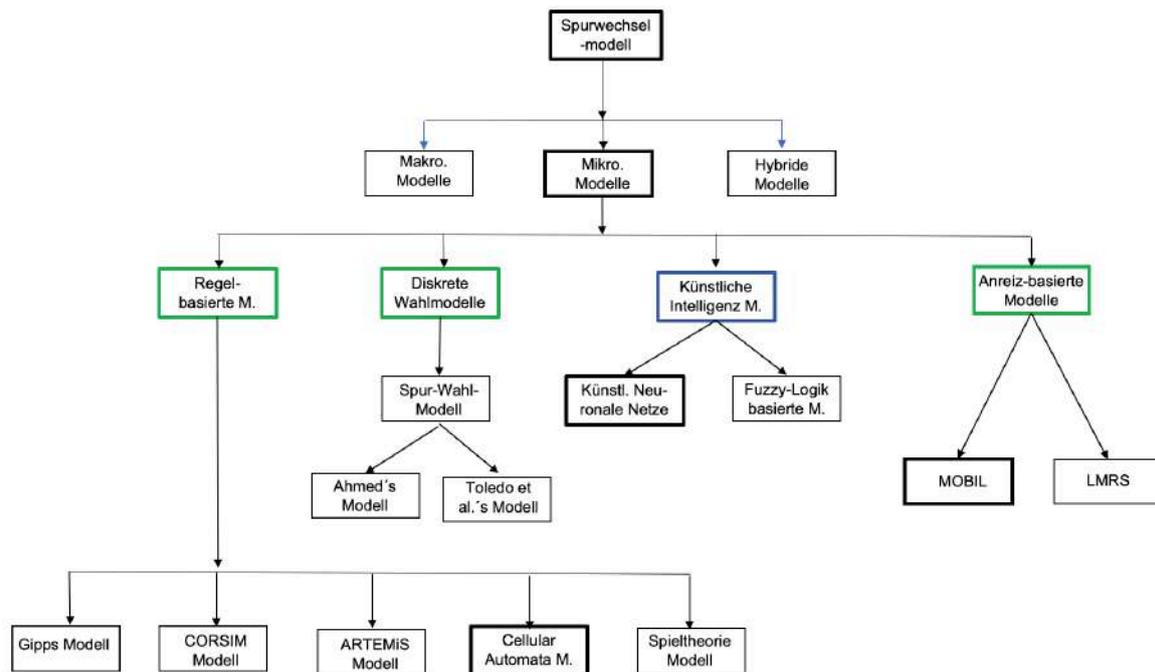


Abbildung 2: Übersicht mikroskopischer LCM [2]

In Abbildung 2 ist zu erkennen, dass diese mikroskopischen LCM in 4 Hauptkategorien unterteilt werden. Außerdem ist zu sehen, dass es neben den mikroskopischen- und makroskopischen- auch noch hybride LCM gibt. Bei diesen hybriden LCM werden sowohl mikroskopische-, als auch makroskopische Kennzahlen verwendet, um das Phänomen zu modellieren.

Unter der ersten Hauptkategorie sind die regel-basierten Modelle zusammengefasst. Dort liegen Regeln oder Heuristiken vor, um das Spurwechselverhalten zu erklären. Beispiele sind der Ablaufplan von Gipps Modell [16] oder das Zellulare Automaten Modell (siehe Abschnitt 2.2.1). In der zweiten Hauptkategorie liegen die diskreten Wahlmodelle und in der dritten Kategorie die Anreiz-basierten Modelle. MOBIL ist ein anreiz-basiertes Modell, welches in dieser Ausarbeitung ausführlich behandelt wird. Diese drei Hauptkategorien sind in Abbildung 2 grün umrandet, da so hervorgehoben werden soll, dass es sich bei Modellen dieser Kategorien um klassische LCM handelt. Die vierte Hauptkategorie wurde blau umrandet und es handelt sich dabei um Modelle die Methoden der künstlichen Intelligenz nutzen. Solche Modelle zählen zu den empirischen Modellen.

2.2 Klassische Spurwechselmodelle

Zunächst werden die klassischen LCM behandelt. Sie unterscheiden sich aufgrund der folgenden drei wichtigen Eigenschaften von den empirischen Modellen: Sie erzielen *deterministische Ergebnisse*, sie erfordern nicht zwingend *Datensätze* und werden eher für Erklärungen als für Prognosen genutzt (eher für die Frage *wieso*, als der Frage *wann*). Die häufigste Form der Anwendung dieser Modelle findet in Verkehrssimulationen statt [1,2].

2.2.1 Zellulares Automaten Modell

Bei Zellulären Automaten Modellen (ZAM) werden dynamische Systeme ausschließlich durch diskrete Variablen beschrieben [2]. Diese Reduzierung komplexer Systeme auf diskrete Variablen ist in der Verkehrsdynamik beliebt. Eines der bekanntesten Modelle, das Nagel-Schreckenberg-Modell, beruht auf dieser Technik [17].

Obwohl die ZAM mikroskopische Modelle sind, beschreiben sie auf Grund ihrer Einfachheit kein realistisches mikroskopischen Fahrverhalten und werden zumeist nur verwendet, um makroskopische Simulationen durchzuführen [18]. Dort werden sie eingesetzt, da sie effizient sind und sich schnell berechnen lassen, aber sich dennoch makroskopische Phänomene gut mit ihnen simulieren lassen. Das Nagels-Schreckenberg-Modell war beispielsweise das erste Modell, dass das Phänomen des *Staus aus dem Nichts* erklären konnte [19].

Auch bei Anwendung der ZAM zur Modellierung des SW waren es unter anderem Nagel und M. Schreckenberg, die Pionierarbeit leisteten [20].

Ihr LCM ist ein typisches Regel-basiertes Modell (siehe Abbildung 2) bei dem drei Regeln (T1, T2 und T3) festgelegt werden, die alle drei erfüllt sein müssen, damit ein SW zustande kommt. Ist eine Regel nicht erfüllt, kommt es zu keinem SW.

Die Regeln entstammen logischen Überlegungen der Autoren, die eine Art Checkliste des Fahrers erstellten. Vor jedem SW geht der Fahrer diese Checkliste durch, die mit T1 beginnt:

T1: Fahrer schaut nach vorne und prüft, ob sich vor ihm ein Fahrzeug befindet

Wird T1 validiert prüft der Fahrer den zweiten Punkt

T2: Fahrer prüft, ob er auf der anliegenden Spur schneller fahren könnte.

Wird auch dieser nicht falsifiziert, folgt noch der letzte Punkt

T3: *Fahrer prüft, ob er, wenn er die Spur wechselt, jemandem hinter sich hat, der schneller sein möchte.*

Mathematisch lassen sich diese drei Regeln nun einfach durch die folgenden drei Ungleichungen in den Formeln (1), (2) und (3) abbilden

$$T1: \text{gap}_n(t) < \min(V_n(t) + 1, V_{max}). \quad (1)$$

Die Variable $\text{gap}_n(t)$ gibt dabei an, wie viele freie Zellen sich zwischen dem Fahrzeug und dem davor fahrenden Fahrzeug befinden. Eine Zelle ist als die Länge eines Fahrzeuges definiert. Der rechte Teil von Formel (1) entspricht dem kleineren Wert zwischen der aktuellen Geschwindigkeit plus eins und der maximal zulässigen Geschwindigkeit.

Die aktuelle bzw. maximal zulässige Geschwindigkeit (in $\frac{m}{s}$) muss größer sein als die Anzahl der freien Zellen, sonst ist ein SW (noch) nicht nötig. Die zweite Regel wird durch Formel (2) beschrieben

$$T2: \text{gap}_{n,o}(t) > \min(V_n(t) + 1, V_{max}). \quad (2)$$

Hier ist nicht die Anzahl der freien Zellen zu dem vorderen Fahrzeug, sondern zu dem vorderen Fahrzeug auf der anliegenden Spur entscheidend. Dieser muss größer sein als die aktuelle/ max. zulässige Geschwindigkeit, wenn ein SW durchgeführt werden soll.

Dies sind die beiden Regeln, die in dem Modell danach fragen, ob ein SW sinnvoll ist. Die dritte Regel geht der Frage nach, ob der SW denn überhaupt möglich ist und wird mit Formel (3) beschrieben [2]

$$T3: \text{gap}_{n,ob}(t) > V_{max}. \quad (3)$$

Die Variable $\text{gap}_{n,ob}(t)$ beschreibt die Anzahl freier Zellen zum nachfolgenden Fahrzeug auf der anliegenden Spur. Diese Anzahl muss größer sein, als die maximale zulässige Geschwindigkeit, da sonst der SW nicht möglich ist.

2.2.2 MOBIL

Das von KESTING, TREIBER und HELBING (2007) entwickelte Spurwechselmodell MOBIL gehört zu der in Abbildung 2 grün markierten Kategorie der Anreiz-basierten Modelle und zeichnet sich durch zwei wichtige Kriterien aus — dem Sicherheits- und dem Anreizkriterium [14].

Sicherheitskriterien werden in vielen klassischen LCM implementiert und sollen dafür sorgen, dass die Spur nur dann gewechselt wird, wenn eine Kollision ausgeschlossen werden kann. Im MOBIL wird dies über Längsbeschleunigungen gewährleistet, in dem ein SW nur dann durchgeführt wird, wenn die Bedingung aus Formel (4)

$$\tilde{a}_a \geq -b_{safe} \quad (4)$$

erfüllt ist.

Es wird angenommen, dass der SW Auswirkungen auf die Fahrzeuge auf der neuen Spur hat, weshalb \tilde{a}_a die Beschleunigung des Fahrzeugs hinter dem Spur-wechselnde Fahrzeug (SWF) beschreibt (Fahrzeug a in Abbildung 1). Der andere Parameter $-b_{safe}$ gibt eine negative Beschleunigung (Bremsen) an, die nicht überschritten werden darf, wenn der SW sicher sein soll. Wenn das Sicherheitskriterium nicht erfüllt ist, muss das nachfolgende Fahrzeug a nachdem SW stärker bremsen, als es möglich ist, weshalb es sicher zur Kollision kommt. Daher darf der SW in diesem Fall nicht durchgeführt werden.

Das weitere Kriterium neben dem Sicherheitskriterium ist das Anreizkriterium. Das Anreizkriterium bestimmt darüber, ob sich die Fahrsituation durch den SW verbessert. In den meisten Modellen bezieht es sich auf die individuelle Situation des Fahrers, der dann die Spur wechselt, wenn sich seine Situation verbessert. Bei MOBIL liegt die Besonderheit vor, dass auch die Situation der umliegenden Fahrzeuge miteinbezogen wird. Wenn sich deren Situation durch den SW verschlechtert, kann es sein, dass der Wechsel nicht durchgeführt wird, obwohl sich die Situation des SWF verbessert hätte. Formal wird dies durch die Ungleichung in Formel (5)

$$\tilde{a}_c - a_c + p(\tilde{a}_o - a_o + \tilde{a}_a - a_a) > \Delta a_{th} \quad (5)$$

beschrieben.

Die ersten beiden Variablen beschreiben die Situationsverbesserung des SWF, denn \tilde{a}_c ist die Beschleunigung nach dem SW und a_c die Beschleunigung davor. Anschließend folgen die Ver-

besserungen des Fahrzeugs o , welches vor dem SW hinter dem SWF fährt und die Verbesserung des Fahrzeugs a das nach dem SW hinter dem SWF fährt. Wie sehr diese beiden Fahrzeuge a und o die Entscheidung des SWF beeinflussen, hängt von dem Parameter p , dem sogenannten Höflichkeitsfaktor ab. Dieser kann Werte zwischen 0 und 1 annehmen und je höher er ist, desto altruistischer ist der Fahrer c . Bei einem hohen Wert von p ist das SWF eher an einer Verbesserung der lokalen Fahrsituation als nur an seiner eigenen Verbesserung interessiert. Bei kleinen Werten nahe null liegt ein „egoistischer“ Fahrer vor, der nur an der Verbesserung seiner eigenen Situation interessiert ist.

Auf der rechten Seite der Formel (5) befindet sich der Schwellenwert (Wechselschwelle) Δa_{th} , welcher verhindert, dass es schon bei der kleinsten Verbesserung der Situation zu einem SW kommt. Wenn ein altruistischer Fahrer abgebildet ($p = 1$) und die Wechselschwelle nicht berücksichtigt wird ($\Delta a_{th} = 0$), dann wird in dem Modell genau dann die Spur gewechselt, wenn sich die Summe aller Beschleunigungen erhöht, bzw. die Summe aller Verzögerungen minimiert. Daher stammt der Name des Modells: *MOBIL – Minimizing overall braking deceleration induced by lane changes* [14].

In dieser Formel (5) wird jedoch nicht spezifiziert auf welcher Spur sich das SWF befindet, also ob ein SW nach links der rechts vorliegt. Häufig spielt dies auch keine Rolle, wenn symmetrischen Verkehr, wie z.B. in den USA vorliegt. In Deutschland, dem Land aus dem der Datensatz stammt, liegt allerdings asymmetrischer Verkehr vor, weshalb MOBIL hier etwas angepasst werden muss.

Wenn ein Wechsel auf die rechte Spur vorliegt wird das Anreizkriterium durch Formel (6)

$$L \rightarrow R: \tilde{a}_c - a_c + p(\tilde{a}_o - a_o) > \Delta a_{th} - \Delta a_{bias} \quad (6)$$

beschrieben und bei einem Wechsel auf die linke Spur durch die Formel (7)

$$R \rightarrow L: \tilde{a}_c - a_c + p(\tilde{a}_n - a_n) > \Delta a_{th} + \Delta a_{bias} \quad (7)$$

beschrieben.

In diesen beiden Formeln ist ein weiterer Parameter Δa_{bias} hinzugefügt worden, welcher den Anreiz auf der rechten Spur zu bleiben erhöhen soll, um so realistischeres Fahrverhalten darstellen zu können [14].

2.3 Empirischen Modelle

Nachdem die klassischen LCM vorgestellt worden sind, werden in diesem Abschnitt die empirischen Modelle der logistischen Regression (logReg) und der Neuronalen Netze (NN) eingeführt. Bei diesen statistischen Modellen wird kein Verständnis über das Phänomen benötigt und auch nicht zwingend geschaffen. Daher sind es auch keine LCM, sondern eher Prognosemodelle bzw. Prognose-Algorithmen. Bei einem Prognose-Algorithmus werden nur der Input und der Output interpretiert, weshalb diese Bezeichnung gut zu den NN passt [21].³ Für die logReg passt dagegen besser die Bezeichnung Prognosemodell, da die Intention klar die Prognose ist, die Parameter aber trotzdem interpretiert werden können. Im Folgenden werden beide Methoden zur Klasse der empirischen Modelle gezählt, da so der Hauptunterschied zu den klassischen LCM betont wird. Denn klassische LCM benötigen a priori Wissen über das Phänomen, aber empirische Modelle nur Beobachtungen des Phänomens. Mit Hilfe der Beobachtungen sollen die Modelle „lernen“ wann ein SW stattfinden kann, um diesen dann prognostizieren zu können. Diese Form des „Lernens“ aus Erfahrungen nennt sich maschinelles Lernen und wird im nächsten Abschnitt erläutert.

2.3.1 Maschinelles Lernen

Mit fortschreitender Computertechnologie sowie enormen Verbesserungen in der Sensoren- und Kamertechnik wurde es in den letzten Jahren möglich Datenmengen zum Verkehrsgeschehen aufzunehmen, wie es vor zehn Jahren noch undenkbar war [6]. Dadurch stehen viele Beobachtungen von SW zur Verfügung, welche, wenn sie richtig analysiert werden Informationen über das Spurwechselverhalten offenbaren könnten.

Dabei wird zunächst davon ausgegangen, dass es nicht genau bekannt ist, wie die Fahrer ihre Spurwechselentscheidung treffen und daher auch nicht genau vorhergesagt werden kann wann der SW erfolgt.

Die Grundidee der Anwendung des maschinellen Lernens ist es, dass es einen Prozess gibt, der zwar nicht direkt beobachtbar ist, aber dennoch das Verhalten der Fahrer erklären kann. Zwar kann dieser Prozess nicht bis ins kleinste Detail identifiziert werden, doch es kann angenommen werden, dass er *nicht rein zufällig* stattfindet.

Durch diese Annahme, dass der Prozess nicht rein zufällig stattfindet, folgt, dass es gewisse *Muster* oder *Regelmäßigkeiten* in den Beobachtungen gibt, die untersucht werden können. Es wird davon ausgegangen, dass sich das zukünftige Spurwechselverhalten nicht gravierend von

³ “Indeed, neural networks are algorithms for prediction, they are not models.” [21]

dem vergangenen Verhalten unterscheidet und somit Muster der Vergangenheit genutzt werden können, um Prognosen über die „Zukunft“ aufstellen zu können [22]. Wie diese Muster zustande kommen, also wieso der Fahrer so handelt, ist dabei nicht von entscheidender Bedeutung.

Um dies umzusetzen, wird beim maschinellen Lernen der Datensatz in Trainings- und Testdaten aufgeteilt. Die Trainingsdaten bilden die Vergangenheit ab und in ihnen sollen die Muster identifiziert werden, die dann auf dem Testdatensatz, welcher quasi die Zukunft darstellt, validiert werden.

Diese Modelle, die auf Methoden des maschinellen Lernens basieren können zwar auch für Verkehrssimulationen genutzt werden (siehe blaue Hauptkategorie in Abbildung 2), doch ihre eigentliche Anwendung liegt in der Prognose des SW und der sich daraus ergebenden Anwendung in Fahrerassistenzsystemen [2].

Allgemein lassen sich drei Kategorien des maschinellen Lernens unterscheiden, dem überwachten Lernen, dem unüberwachten Lernen und dem bestärkenden Lernen. [23]

In dieser Ausarbeitung werden nur Methoden des überwachten Lernens behandelt, die sich dadurch auszeichnen, dass die Ausgabe mit einem wahren Wert verglichen werden kann.

Die erste Methode, welches dieses Vorgehen umsetzen soll, ist die logReg, deren mathematischen Grundlagen nachfolgend erläutert werden.

2.3.2 Logistische Regression

Methoden des maschinellen Lernens können bei der Form des überwachten Lernens für Regressions- oder Klassifikationsprobleme verwendet werden [22].

Auch wenn der Name anderes vermuten lässt, wird die logReg für Klassifikationsprobleme verwendet. In dieser Ausarbeitung für die Klassifikation zwischen SW und kein-SW.

Im Rahmen des logistischen Regressionsmodells werden Klassifikationsprobleme durch eine *Gerade, Ebene oder Hyperebene* in einem mehrdimensionalen Raum getrennt [22]. Die Gerade/Ebene nennt sich lineare-Entscheidungsgrenze und Werte die oberhalb dieser Geraden/Ebene liegen werden anders klassifiziert als Werte unterhalb [24].

In Abbildung 3 wird die Entscheidungsgrenze für ein zweidimensionales Beispiel dargestellt.

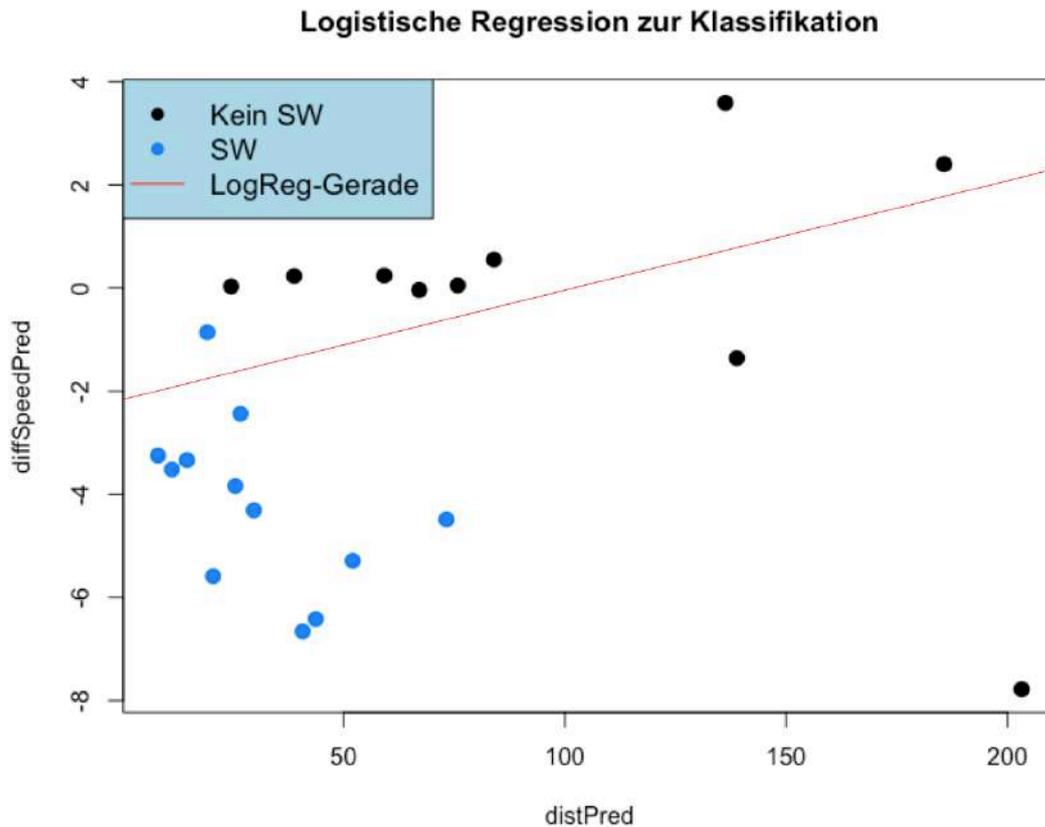


Abbildung 3: Beispiel einer Klassifikation mit Hilfe der logReg

Auf der Ordinate und der Abszisse werden die beiden Input Variablen (IV) diffSpeedPred und distPred^4 abgebildet und die Farbe der Punkte bildet die Output Variable (OV). Die blauen Punkte beschreiben einen SW, wogegen die schwarzen Punkte Situationen beschreiben in denen kein-SW vorliegt. Ziel der Anwendung der logReg ist es, eine Entscheidungsgrenze zu finden, mit der die Ausprägungsmöglichkeiten der OV klassifiziert werden können. In diesem Fall würde es bedeuten, dass alle schwarzen Punkte oberhalb und allen blauen Punkte unterhalb der Grenze liegen würden. Diese Entscheidungsgrenze wird in Abbildung 3 durch die rote Linie dargestellt und es werden dadurch 19 der 22 Beobachtungen richtig klassifiziert.

Auf den ersten Blick scheint es, als sei in Abbildung 3 eine Regressionsgerade, wie sie im linearen Regressionsmodell vorkommt, gezeichnet worden. Bei dem linearen Regressionsmodell handelt es sich um eins der bekanntesten statistischen Methoden überhaupt und sie wird zur Lösung von Regressionsproblemen verwendet [25]. Eine Regressionsgerade sieht ebenfalls so aus wie die Entscheidungsgrenze in Abbildung 3, doch beschreibt diese den Zusammenhang zwischen Input- und Output Variable. In Abbildung 3 sind dagegen auf beiden Achsen IV zu sehen.

⁴ Diese beiden Variablen befinden sich in dem highD-Datensatz, der an späterer Stelle vorgestellt wird. Hier dienen sie nur als Beispiel und müssen nicht inhaltlich interpretiert werden.

Im ersten Schritt werden die IV im logistischen Regressionsmodell, wie auch im linearen Regressionsmodell als linear-Kombination dargestellt

$$y_i = \beta_0 + \beta_1 x_{1i} + \dots + \beta_j x_{ji} = \mathbf{x}'_i \boldsymbol{\beta}, \quad (8)$$

wobei y_i die OV, x_{ji} die IV und β_j die Parameter beschreiben [25]. Der nächste Schritt macht den Unterschied zum linearen Regressionsmodell deutlich.

Denn im logistischen Regressionsmodell ist die OV nicht stetig skaliert, sondern kategorial bzw. im vorliegenden Fall binär. Daher wird die Abhängigkeit der OV mit den IV nicht über eine lineare Funktion, sondern über eine nichtlineare Responsefunktion h approximiert

$$y_i = h(\mathbf{x}'_i \boldsymbol{\beta}). \quad (9)$$

Diese Responsefunktion, die die Form einer logistischen Funktion aufweist, wird durch Formel (10) beschrieben

$$h(\mathbf{x}'_i \boldsymbol{\beta}) = \frac{\exp(\mathbf{x}'_i \boldsymbol{\beta})}{1 + \exp(\mathbf{x}'_i \boldsymbol{\beta})} = \pi_i. \quad (10)$$

Sie ist eine entlang der ganzen Abszisse streng monoton wachsende Verteilungsfunktion, für die immer gilt $h(\mathbf{x}'_i \boldsymbol{\beta}) \in [0, 1]$.⁵

Dieser Wertebereich macht es möglich die Werte der Responsefunktion als Wahrscheinlichkeiten π_i zu interpretieren [25]. Hat die OV also zwei Ausprägungsmöglichkeiten (SW oder kein-SW), so gibt π_i die Wahrscheinlichkeit an, dass die Ausprägungsmöglichkeit „kein-SW“ vorliegt und $1 - \pi_i$ die Wahrscheinlichkeit, dass „SW“ vorliegt.⁶ In Abbildung 4 wird die Verteilung der logistischen Responsefunktion aus Formel (10) graphisch dargestellt.

⁵ Auch wenn die Responsefunktion bei der logistischen Regression nichtlinear ist, können trotzdem nur lineare Klassifikationen durchgeführt werden (lineare Entscheidungsgrenzen), da linear-kombinationen der IV vorliegen. Siehe „Clearly, if either the $\delta_k(x)$ or $\Pr(G = k|X = x)$ are linear in x , then the decision boundaries will be linear.“ [24, p.102]

⁶ Normalerweise werden diese Kategorien einfach mit 0 und 1 beschrieben.

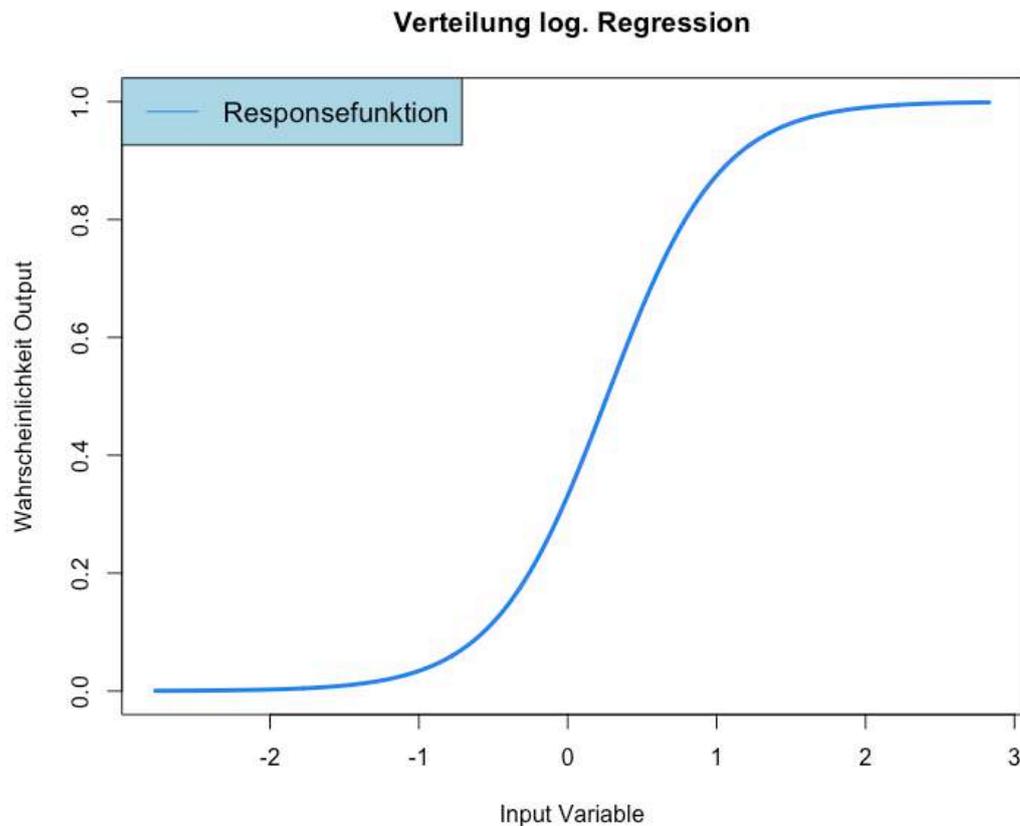


Abbildung 4: Verteilung der logReg.

Auf der Abszisse ist die IV x abgebildet. Je größer die IV in Abbildung 4 ist, desto höher ist die Wahrscheinlichkeit das „kein SW“ vorliegt. Am steilsten verläuft die Kurve im Bereich der Wahrscheinlichkeit um 0,5, wodurch marginale Änderungen der IV die Wahrscheinlichkeit dort am stärksten beeinflussen [25].

Wenn das logistische Regressionsmodell formuliert ist, stellt sich die Frage, wie die „besten“ Parameter β gefunden werden können. Dafür wird bei der logReg oder generell bei den generalisierten linearen Modellen⁷ die *Maximum-Likelihood-Methode (MLM)* verwendet [26]. Dabei handelt es sich um eine Methode bei der die Parameter β so geschätzt werden, dass die Wahrscheinlichkeit die Beobachtungen y_i richtig zu klassifizieren, maximal wird [25]. In Formel (11) wird die Likelihood-Funktion beschrieben, welche mit Hilfe der MLM maximiert werden soll

⁷ Bei den generalisierten linearen Modellen handelt es sich um eine Erweiterung des linearen Regressionsmodells, bei denen der Fehlerterm andere Verteilungen als die Normalverteilung annehmen darf [26].

$$L(\boldsymbol{\beta}) = \prod_{n=1}^i \pi^{y_i} * (1 - \pi)^{1-y_i}. \quad (11)$$

Die Parameter werden so geschätzt, dass die Wahrscheinlichkeit, die durch Formel (10) berechnet wird, möglichst hoch ist, wenn $y_i = 1$ (kein-SW) und möglichst niedrig, wenn $y_i = 0$ (SW). Um die Berechnungen noch zu vereinfachen⁸ wird Formel (11) noch logarithmiert, wodurch sich Formel (12) ergibt

$$\log(L) = l(\boldsymbol{\beta}) = \sum_n^i y_i \log(\pi) + (1 - y_i) * \log(1 - \pi). \quad (12)$$

Von dieser Funktion $\log(L)$ soll das Maximum gefunden werden. Dies ist bei der logistischen Regression nur durch einen iterativ vorgehenden Algorithmus möglich.⁹ Bei den später folgenden Berechnungen wird dafür ein IWLS-Algorithmus verwendet, der auf dem Newton-Raphson-Verfahren aufbaut [27, 28].

Zum Schluss, nachdem das Modell formuliert worden ist, und die Parameter bestimmt sind, muss noch eine Entscheidungsregel festgelegt werden. Diese gibt an ab welcher Wahrscheinlichkeit die Beobachtung zu den kein-SW zugeordnet wird. Häufig wird die Grenze dafür in der Mitte bei der Wahrscheinlichkeit von 50 % gewählt. Wird erneut das Beispiel aus Abbildung 3 verwendet würde sich die Entscheidungsregel in Formel (13)

$$\hat{G} = \begin{cases} \text{Kein SW, wenn } h(\beta_0 + \beta_1 * x_1 + \beta_2 * x_2) > 0,5 \\ \text{SW, wenn } h(\beta_0 + \beta_1 * x_1 + \beta_2 * x_2) < 0,5 \end{cases} \quad (13)$$

ergeben.¹⁰

⁸ Aus einer Multiplikation wird eine Addition.

⁹ Anders als bei der linearen Regression, die mit Hilfe Methode der kleinsten Quadrate analytisch berechnet werden kann.

¹⁰ Mit x_1 als distPred und x_2 als diffSpeedPred

2.3.3 Neuronale Netze

Der Begriff Neuronale Netze wird für eine Klasse unterschiedlicher Modelle benutzt, in denen Informationen auf eine scheinbar ähnliche Weise verarbeitet werden, wie dies im menschlichen Gehirn (den Neuronen) der Fall ist [29]. Mathematisch gibt es drei Hauptrechenoperationen, die bei diesen Modellen durchgeführt werden müssen. Diese laufen auch in ähnlicher Weise in Form von biochemischen Prozessen im menschlichen Gehirn ab, weshalb diese Analogie so beliebt ist. In Abbildung 5 werden die drei Rechenoperationen anhand eines Modells, das ein einzelnes Neuron abbilden soll, dargestellt [23].

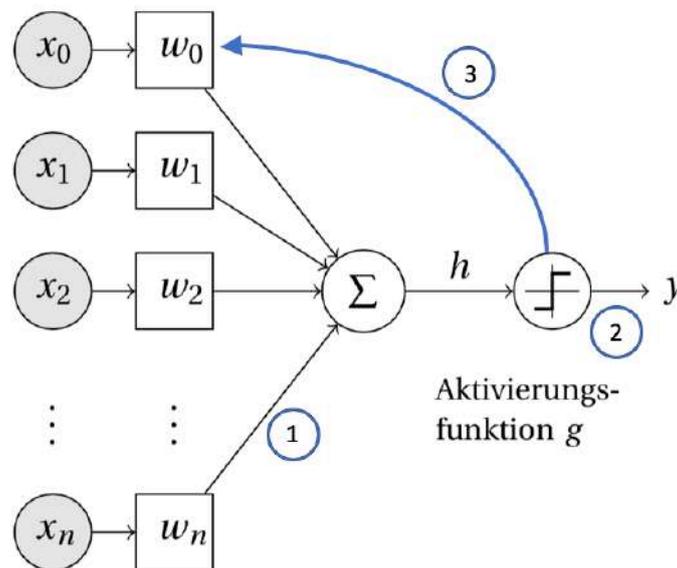


Abbildung 5: Mathematische Darstellung eines Neurons.

Bei diesem in Abbildung 5 dargestellten Modell handelt es sich um das einfache Perzeptron Modell von FRANK ROSEBLATT (1958) [30]. Dieses Modell soll die Signalverarbeitung von menschlichen Neuronen mathematisch beschreiben. Die blauen Kreise sollen die dafür notwendigen drei Hauptrechenoperationen darstellen.

Im ersten Schritt werden die IV mit Parametern verknüpft, welche in der Terminologie der Neuronalen Netze als Gewichte bezeichnet werden (siehe 2.3.3.1). Im menschlichen Gehirn kommen bei diesem Schritt exogene Reizeinwirkungen in den Neuronen an, die dann im zweiten Schritt im Zellkern verarbeitet werden und anschließend an andere Neuronen weitergeleitet werden, sobald ein Schwellwert erreicht worden ist. Bei den NN wird dieser Schritt durch eine Aktivierungsfunktion substituiert (2.3.3.2). Diese beiden Schritte sind auch bei der logReg in Formel (8) und (10) zu finden. Erst der dritte Schritt verdeutlicht den Unterschied zwischen den Methoden, nämlich das Anpassen der Gewichte. Dieses wird durch Algorithmen wie dem

Backpropagation-Algorithmus erreicht und ist dem wachsen bzw. schrumpfen der Verbindungen zwischen den Neuronen im menschlichen Gehirn nachempfunden (siehe 2.3.3.3).

Mit dem in Abbildung 5 dargestellten Modell lassen sich, wie bei den logistischen Regressionsmodellen lineare-Entscheidungsgrenzen zur Klassifikation approximieren [24]. Wenn dieses Modell dann aber um einen oder mehrere verdeckte Schichten (Hidden Layer) ergänzt wird, lassen sich sämtliche stetige Funktionen approximieren, mit denen auch hoch komplexe nicht-lineare Klassifikations- oder Regressionsprobleme gelöst werden können. In Abbildung 6 wird das Beispiel aus dem vorherigen Kapitel erneut aufgegriffen, jedoch wird die Entscheidungsgrenze nicht mit der logReg, sondern mit NN bestimmt.

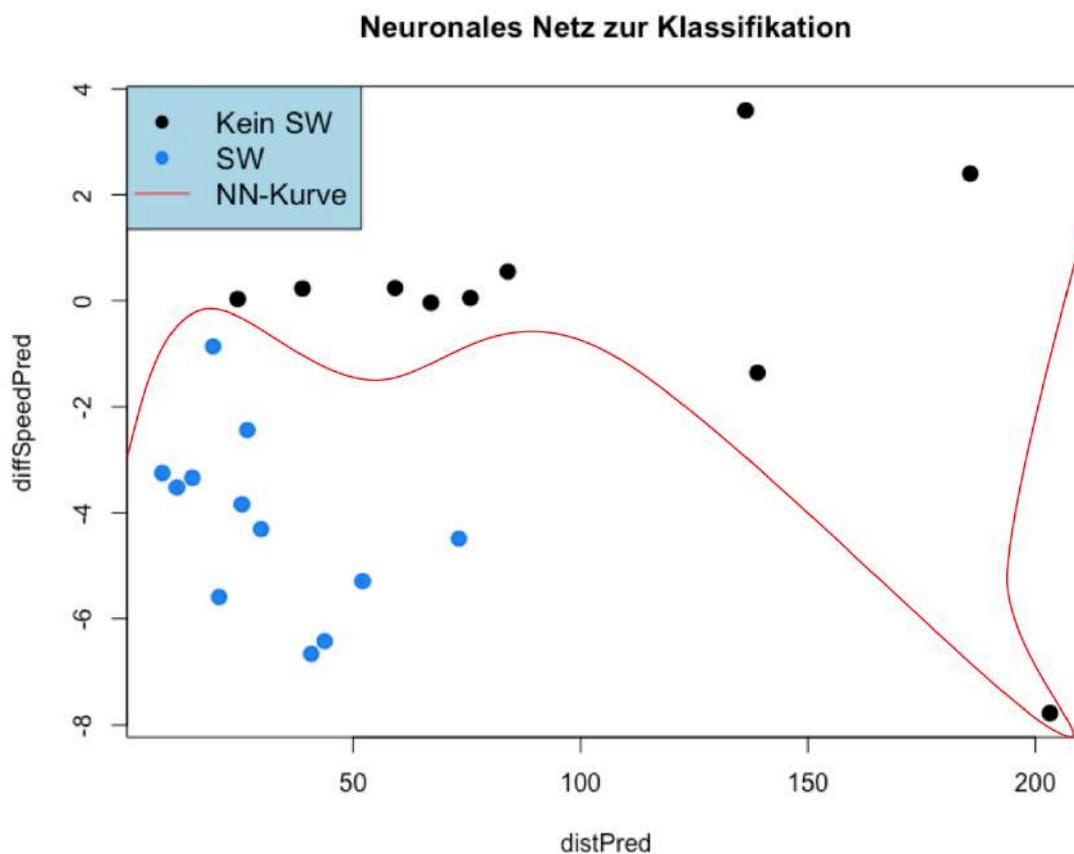


Abbildung 6: Nichtlineare Klassifikation mit NN

Wie in Abbildung 6 zu erkennen ist, beschreibt die Entscheidungsgrenze im Gegensatz zu der in Abbildung 3 einen nichtlinearen Verlauf. Hier wird deutlich welches Potential, aber auch welche Schwierigkeit bei der Anwendung von NN auftreten. Sie können, durch ihr Potential hoch komplexe Funktionen zu approximieren, auch schwierigste Klassifikationsprobleme wie in Abbildung 6 gezeigt erfolgreich lösen, neigen jedoch zu Überanpassungen (Overfitting).

Overfitting liegt im Rahmen des maschinellen Lernens dann vor,¹¹ wenn die approximierende Funktion komplexer ist, als der zugrundeliegende Sachverhalt. Dies wird dann daran ersichtlich, dass die Klassifikation beim Training sehr erfolgreich ist, jedoch nicht auf die Testdaten übertragen werden kann.

In den nächsten Abschnitten werden die mathematischen Grundlagen zur Berechnung und Anpassung dieser Parameter beschrieben. Orientiert wird sich dabei an den 3 Hauptrechenoperationen aus Abbildung 5.

2.3.3.1 1. Schritt: Verknüpfung der Knoten und Gewichte

Zu Beginn werden die in Abbildung 5 grau unterlegten Input Variablen x_n mit den passenden Gewichten w_n multipliziert. Die sich daraus ergebene Summe wird mit s_i bezeichnet und bildet den Input der Neuronen im ersten Hidden Layer. Ein Hidden Layer kann eine beliebige Anzahl an Neuronen beinhalten kann und ein NN kann aus mehreren Hidden Layern aufgebaut sein. Je höher die Anzahl der Hidden Layer und der Neuronen, desto komplexere Muster können erlernt werden,¹² doch umso höher ist auch die Gefahr des Overfittings [31, 32].¹³

Dieser Input s_i wird in eine Aktivierungsfunktion eingesetzt (siehe 2.3.3.2), woraus sich dann der Output h_i ergibt. In Formel (14) wird der Output des iten Neurons im ersten Hidden Layer abgebildet

$$h_i^{(1)} = g^{(1)}(s_i) = g^{(1)}\left(\sum_n^i w_i^{(1)} x_i + b_0^{(1)}\right). \quad (14)$$

Durch die (1) im oberen Index der Parameter in Formel (14) wird verdeutlicht, dass es sich um den ersten Hidden Layer handelt. Zusätzlich zu der Summe aus dem Produkt von Input x_i und den Gewichten $w_i^{(1)}$ sind noch der Konstante Faktor $b_0^{(1)}$ und die Aktivierungsfunktion $g^{(1)}$ zu erkennen.

Falls neben dem ersten Hidden Layer noch ein zweiter Hidden Layer vorhanden ist, wird das Prozedere des ersten Hidden Layer wiederholt, allerdings mit dem Unterschied, dass der Input

¹¹ Die Bedeutung in der Statistik allgemein ist etwas anders. Dort liegt Overfitting vor, wenn zu viele erklärende Variablen in dem Modell vorhanden sind [25].

¹² Lernkapazität nimmt zu [31].

¹³ „Too many neurons in the hidden layers may result in overfitting“.[32]

des zweiten Hidden Layer nicht der Input x_i des Modells ist, sondern der Output des ersten Hidden Layers $h_i^{(1)}$, wie in Formel (15) zu sehen ist

$$h_i^{(2)} = g^{(2)}(s_i) = g^{(2)}\left(\sum_n^i w_i^{(2)} h_i^{(1)} + b_0^{(2)}\right). \quad (15)$$

Wären nun noch weitere Hidden Layer vorhanden, würde der Output von Formel (15) wieder als Input des nächsten Hidden Layers dienen, aber ist dies nicht der Fall, so muss nur noch der Output Layer durch Formel (16)

$$\hat{y}_i = g^{(3)}(s_i) = g^{(3)}\left(\sum_n^i w_i^{(3)} h_i^{(2)} + b_0^{(3)}\right). \quad (16)$$

berechnet werden. Das Ergebnis von Formel (16) ist dann der gesamte Output des Netzes, welcher mit den wahren Werten verglichen werden kann.

2.3.3.2 2. Schritt: Die Aktivierungsfunktion

Ein Beispiel für eine Aktivierungsfunktion bei den Neuronalen Netzen ist die Responsefunktion der logistischen Regression, die in Formel (10) beschrieben ist. In den Anfängen der Entwicklung der künstlichen Intelligenz war auch die Heavyside-Funktion, die nur zwei Zustände annehmen kann, sehr beliebt, da sie dem biologischen Vorbild des Gehirns am nächsten kam [33]. Der später entwickelte Algorithmus, der im nächsten Schritt vorgestellt wird, erfordert jedoch stetige Aktivierungsfunktionen, da nur diese differenziert werden können [34].

Solange die Differenzierung-Bedingung erfüllt wird, kann theoretisch jede Funktion als Aktivierungsfunktion gewählt werden, wobei sich in der Praxis gewisse Funktionen als sinnvoller herausgestellt haben als andere [32].

Um zu entscheiden welche Aktivierungsfunktionen für das Modell verwendet werden soll, kann nur auf heuristische Regeln zurückgegriffen werden und es kann sogar sein, dass innerhalb eines Modells oder sogar eines Layers unterschiedliche Aktivierungsfunktionen verwendet werden [32]. Üblich ist es jedoch bei den Hidden Layers eine Aktivierungsfunktion der Familie der sigmoiden Funktionen zu wählen, welche die vorteilhaften Eigenschaften besitzen, dass sie nur Werte zwischen null und eins annehmen, monoton steigen und differenzierbar sind. Ein Beispiel für eine sigmoide Funktion ist die in Abbildung 4 dargestellte logistische Funktion.

Im Output Layer hängt die Wahl der Aktivierungsfunktion von der Fragestellung ab. Bei Regressionsproblemen kann eine lineare Aktivierungsfunktionen sinnvoll sein während bei Klassifikationsproblemen in der Regel auch beim Output Layer eine sigmoide Funktion gewählt wird.

Das Einsetzen einer Aktivierungsfunktion, wie zum Beispiel der Funktion aus Formel (10) in die Formel (15) des zweiten Hidden Layers, ergibt die Formel (17)

$$h_i^{(2)} = \frac{\exp(\sum_n^i w_i^{(2)} h_i^{(1)} + b_0^{(2)})}{1 + \exp(\sum_n^i w_i^{(2)} h_i^{(1)} + b_0^{(2)})} \quad (17)$$

2.3.3.3 3. Schritt: Anpassung der Gewichte— Das „Lernen“

Dieser dritte Schritt ist fundamental für die Thematik der Neuronalen Netze. Die bisher vorgestellten Grundlagen waren schon in den 1950er Jahren bekannt, als Roseblatt sein Perzeptronen Modell veröffentlichte [30]. Mit diesem Perzeptronen Modell entstand der erste Hype um diese Thematik, der aber schnell wieder abflaute [35].¹⁴ Der Grund war, dass es bis in die 1980er Jahre dauerte, bis ein zuverlässiges Verfahren entwickelt werden konnte, mit dem es möglich wurde, bei mehrschichtigen NN die Gewichte sinnvoll anzupassen. Denn erst diese Anpassung der Gewichte ermöglicht es, Informationen aus den vorherigen Durchläufen zu speichern und das Netz damit zum (maschinellen) Lernen zu bringen. Bei den später folgenden Prognosen wird der Backpropagation-Algorithmus angewendet, weshalb dieser nachfolgend beschrieben wird.

Zu Beginn muss das konfigurierte Netz dafür erst einmal komplett durchlaufen werden, damit die Ausgabe \hat{y} des Output Layers bekannt ist. Diese Ausgabe wird dann mit dem wahren Wert y verglichen, welcher a priori bekannt sein muss.¹⁵ Aus dem Vergleich der beiden Werte kann dann der Fehlerterm hergeleitet werden, der beim Backpropagation-Algorithmus meist durch die *quadratische Differenz* (sse)

¹⁴ Häufig als erster „AI winter“ bezeichnet.

¹⁵ Notwendige Eigenschaft des überwachten Lernens.

$$E(w_{ij}) = \frac{1}{2} \sum_n (y - \hat{y})^2, \quad (18)$$

wenn ein Regressionsproblem vorliegt oder durch die *Kreuzentropie* (ce)

$$E(w_{ij}) = - \sum_n^i y_i \log(\hat{y}) + (1 - y_i) * \log(1 - \hat{y}), \quad (19)$$

wenn ein Klassifikationsproblem vorliegt, berechnet wird [24].¹⁶

Die Kreuzentropie entspricht dabei fast der Formel (12), welche im Rahmen der MLM maximiert wird, nur mit dem Unterschied, dass ein Minuszeichen hinzugefügt worden ist. Der Grund für dieses Vorzeichen ist, dass das Minimum des Fehlers und nicht das Maximum der Wahrscheinlichkeit gesucht wird [22].

Wie auch schon bei der logReg wird nun das Minimum bei dem Backpropagation-Algorithmus mit Hilfe des *Gradientenverfahrens*, auch Verfahren des steilsten Abstiegs genannt, gesucht. Dabei wird untersucht, wie sich der Fehlerterm E in Abhängigkeit der Gewichtseinstellungen verhält, also allgemein $\frac{\partial E}{\partial w_i}$. Um diesen Term abzuleiten wird die Kettenregel angewandt und es ergibt sich Formel (20)

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial s_i} \frac{\partial s_i}{\partial w_{ji}}. \quad (20)$$

Im Folgenden wird gezeigt, wie sich diese drei Brüche ableiten lassen, wobei dies je nach gewählter Aktivierungsfunktion und gewähltem Fehlerterm unterschiedlich ist. Hier wird der Fall einer sigmoiden (logistischen) Funktion mit der Kreuzentropie als Fehlerterm beschrieben, da diese Einstellungen auch später gewählt werden um den SW zu prognostizieren.

Dann ergibt sich für den ersten Bruch

$$\frac{\partial E}{\partial \hat{y}_i} = \frac{-y_i}{\hat{y}_i} + \frac{1 - y_i}{1 - \hat{y}_i} = \frac{\hat{y}_i - y_i}{\hat{y}_i(1 - \hat{y}_i)} \quad (21)$$

¹⁶ Häufig wird es so gehandhabt, aber es ist nicht zwingend. In dieser Ausarbeitung ist es deshalb vorteilhaft, da nur so das AIC der NN mit denen der logReg verglichen werden kann.

in dem Formel (19) nach \hat{y} abgeleitet wird [36].

Als nächstes muss der Output Layer nach s_i abgeleitet werden. Hier zeigt sich erneut, weshalb Sigmoidfunktion beliebt sind: sie lassen sich leicht differenzieren.¹⁷ Dadurch ergibt sich dann Formel (22)

$$\frac{\partial \hat{y}_i}{\partial s_i} = \hat{y}_i * (1 - \hat{y}_i). \quad (22)$$

Der dritte Bruch lässt sich auch leicht bestimmen in dem s_i nach den Gewichten w_{ji} abgeleitet wird, wodurch sich

$$\frac{\partial s_i}{\partial w_{ji}} = h_j \quad (23)$$

ergibt [36].

Werden die drei Brüche zusammengefasst ergibt sich Formel (24)

$$\frac{\partial E}{\partial w_{ji}} = \frac{\hat{y}_i - y_i}{\hat{y}_i(1 - \hat{y}_i)} * \hat{y}_i(1 - \hat{y}_i) * h_j = (\hat{y}_i - y_i) * h_j. \quad (24)$$

Nun muss Formel (24) nur noch um eine Lernrate α ergänzt werden, die dafür sorgt, dass die Gewichtseinstellungen nicht zu stark von dem aktuellen Durchlauf abhängen. Ist dies der Fall verfügt das Netz nur über ein Kurzzeitgedächtnis und vergisst nach jedem Durchlauf vorher gelerntes. Dieser Faktor sorgt dafür, dass aktuelle Durchläufe vorher gelerntes nicht löschen, indem die Anpassungen der Gewichte in kleinen Schritten erfolgen [31]. Dann ergibt sich die Gewichtsveränderung aus Formel (25)

$$\Delta w_{ji} = \alpha (\hat{y}_i - y_i) h_j. \quad (25)$$

Diese Formel (25) ist übersichtlich und leicht zu verstehen. Wenn die Ausgabe \hat{y}_i nicht mit dem wahren Wert y_i übereinstimmt, werden die Gewichte verändert. Sie werden dabei umso stärker verändert, je größer die Eingabe des Layers h_j war und je größer die gewünschte Lernrate ist. Die Formel (25) hat jedoch einen Nachteil: Sie gilt nur für die Gewichte vor dem Output Layer! Gewichte vor oder zwischen den Hidden Layern können so nicht angepasst werden, da deren Output nicht mit dem wahren Output verglichen werden kann. Der Output eines Hidden Layers

¹⁷ $\text{sig}'(x) = \text{sig}(x)(1-\text{sig}(x))$

entspricht nur dem Input des nächsten Layers und kann somit nicht mit einem Referenzwert verglichen werden, wie dies beim Output Layer der Fall ist. Daher muss sich hier eines Hilfsmittels bedienen werden, welches verdeutlicht, woher der Algorithmus seinen Namen hat, nämlich die „Fehlerrückführung“.

Dazu muss zunächst s_i^{j+1} definiert werden, welches den Input des Layers entspricht, der vor dem Layer, dessen Gewichte verändert werden sollen.

Nun kann nicht der Fehler zwischen Output und wahren Wert direkt verwendet werden, sondern der Fehler der vorherigen Schicht wird auf diese Schicht in Abhängigkeit ihrer „Verantwortung“, also ihrer Gewichte w_{ji} , zurückgeführt [23]. Mathematisch wird diese Fehlerzurückführung durch Formel (26)

$$\frac{\partial E}{\partial s_i^{j+1}} = \sum_{i=1} \frac{\partial E}{\partial s_i} \frac{\partial s_i}{\partial h_j} \frac{\partial h_j}{\partial s_i^{j+1}} = \sum_{i=1} (\hat{y}_i - y_i)(w_{ji})(h_j(1 - h_j)) \quad (26)$$

ausgedrückt.

Nun kann durch erneute Anwendung der Kettenregel die Veränderung des Fehlerterms in Abhängigkeit der Gewichte des Hidden Layers durch

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial s_j^1} \frac{\partial s_j^1}{\partial w_{ji}} = \sum_{i=1} (\hat{y}_i - y_i)(w_{ji})(h_j(1 - h_j))(x_i) \quad (27)$$

beschrieben werden [36]. Der hinterste Teil x_i beschreibt den Input des Layers.

Durch Formel (24) und Formel (27) kann der Fehler, beginnend mit den Gewichten des Output Layers, bis zu den ersten Gewichten des ersten Hidden Layers, zurückgeführt werden und somit erfolgreich die Gewichte angepasst werden. Allerdings muss auch hier noch die Lernrate α ergänzt werden und es ergibt sich

$$\Delta w_{ji} = \alpha * \sum_{i=1} (\hat{y}_i - y_i)(w_{ji})(h_j(1 - h_j))(x_i) \quad (28)$$

für die Veränderung der Gewichte eines Hidden Layers.

3.0 Der Datensatz

In diesem Kapitel geht es um die Beobachtungen die zur Prognose herangezogen werden. Diese entstammen dem „highway drone dataset“ (highD-Datensatz) [4]. Zunächst wird der Datensatz allgemein vorgestellt und anschließend erst die Output Variable und anschließend die Input Variablen beschrieben. Darauf folgt eine deskriptive Analyse des Datensatzes, bei der zunächst Besonderheiten der OV und anschließend der Einfluss sowie die Auswahl der IV beschrieben wird. Die Ergebnisse der deskriptiven Analyse werden in dem darauffolgenden Kapitel für die Entwicklung der Modelle genutzt.

3.1 Beschreibung des Datensatzes

Der highD-Datensatz besteht aus Fahrzeug–Trajektorien, die mit Hilfe einer Drohne auf deutschen Autobahnen aufgenommen worden sind. Die Drohnen flogen dafür mindestens 100 m über der Straße wobei sie so einen Bereich von 420 m in der Vogelperspektive aufnehmen konnten (siehe Abbildung 7). Aus diesen Aufnahmen wurden dann mittels Computer Visions Algorithmen und manueller Kommentierung die Trajektorien und weitere Informationen berechnet [37].

Insgesamt wurden an sechs verschiedenen Standorten (in der Nähe von Köln) 60 Stunden Autobahnverkehr aufgenommen und verarbeitet.

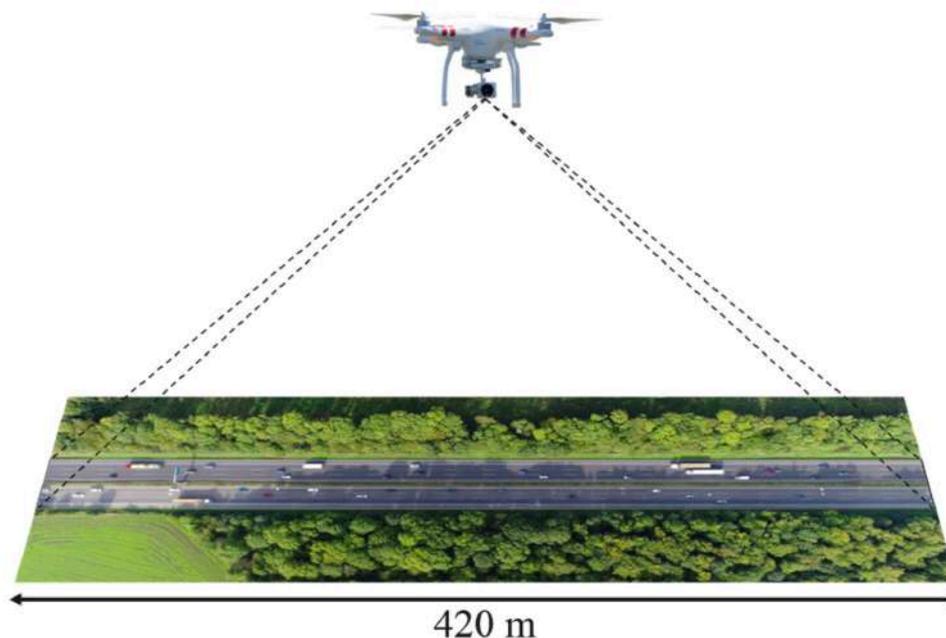


Abbildung 7: Die Entstehung des highD-Datensatzes [37]

In Abbildung 7 ist der Aufbau des Generierungsprozesses der Daten dargestellt, dessen Ziel es war, Daten für die Sicherheitsvalidierung hochautomatisierter Fahrzeuge zur Verfügung zu

stellen [37]. Dabei ist der highD-Datensatz der erste Datensatz, bei dem für diesen Zweck der Verkehr aus der Vogelperspektive mit Hilfe einer Drohne aufgenommen worden war [37]. Bis dato war es üblicher Aufnahmen aus Testfahrzeugen zu machen, welche mit Kameras, GPS, Radars usw. ausgestattet waren (siehe bspw. EuroFOT [38]). Eine weitere Methode beruhte darauf, dass Kameras auf hohen Gebäuden platziert wurden und dadurch Datensätze erstellt wurden (siehe NGSIM [39]).

Die Generierung der Daten mit Hilfe einer Drohne bringt einige Vorteile mit sich. Dadurch, dass die Drohne hoch in der Luft von den Fahrzeugen auf der Straße nicht wahrgenommen wird, kann davon ausgegangen werden, dass ein natürliches unbeeinflusstes Fahrverhalten beobachtet werden kann. Das Weiteren hält sich der Aufwand der für die Generierung betrieben werden muss in Grenzen und sämtliche wichtigen beobachtbaren Informationen können aufgenommen werden [40].

Insgesamt verfügt der verwendete Datensatz über 26 verschiedene Variablen, die Auskunft über die Trajektorien des SWF, sowie vier weiteren umliegenden Fahrzeugen geben. Insgesamt liegen 11783 Beobachtungen vor.

3.1.2 Die Output-Variable

Die kategoriale OV, die später prognostiziert werden soll, wird als „Manöver“ bezeichnet und enthält die folgenden vier Ausprägungsmöglichkeiten:

- Manöver 1: LKR -> Fahrzeug bleibt auf der rechten Spur
- Manöver 2: LKL -> Fahrzeug bleibt auf der linken Spur
- Manöver 3: LCR-> Fahrzeug wechselt auf die rechte Spur
- Manöver 4: LCL -> Fahrzeug wechselt auf die linke Spur

Dabei gibt es zwei unterschiedliche Situationen die nun beschrieben werden.

Die erste Situation wird in Abbildung 8 dargestellt und aus ihr kann entweder das Manöver 1 oder das Manöver 4 entstehen. Im Folgenden wird diese Situation stets M1&4 genannt. Das SWF befindet sich in dieser Situation auf der rechten Fahrbahn und hat somit die Möglichkeit auf dieser zu bleiben (Manöver 1) oder die Spur zu wechseln (Manöver 4).

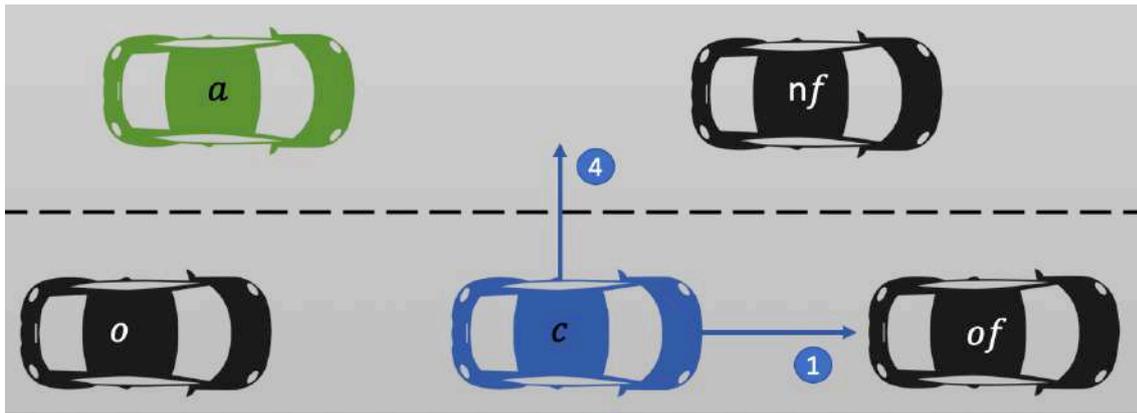


Abbildung 8: Situation M1&4

Das SWF wird in Abbildung 8 blau dargestellt und mit einem c beschriftet. Auf der linken Spur wird in Abbildung 8 ein grünes Fahrzeug a abgebildet. Bei diesem Fahrzeug handelt es sich um das AF, welches die Manöver des blauen Fahrzeuges prognostizieren soll. Die Spurwechselprognosen werden demnach aus der Perspektive dieses grünen Fahrzeuges erstellt.

Die zweite Situation wird in Abbildung 9 dargestellt und aus ihr kann das SWF entweder das Manöver 2 oder das Manöver 3 durchführen. Bei Manöver 2 bleibt es auf der linken Spur und bei Manöver 3 wechselt es auf die rechte Spur aus.

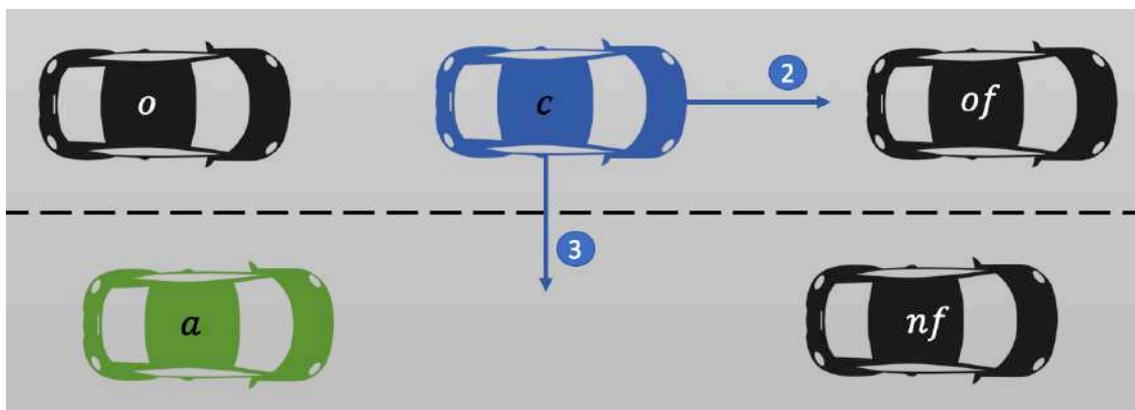


Abbildung 9: Situation M2&3

Dies ist die Situation M2&3. Das AF befindet sich hier auf der rechten Spur und soll mögliches Einschleichen des blauen Fahrzeuges auf die (bei asymmetrischen Verkehr) langsamere rechte Spur prognostizieren.

3.1.3 Die Input-Variablen

Zusätzlich zu der OV sind 25 weitere Variablen in dem highD-Datensatz vorhanden.

Für jedes der fünf dargestellten Fahrzeuge in Abbildung 9 sind fünf weitere Variablen bekannt. Für das SWF sind (neben dem Manöver) noch die laterale sowie die vertikale Geschwindigkeit und die laterale sowie vertikale Beschleunigung angegeben. Zudem beschreibt eine weitere Variable (Class), ob es sich um die Klasse der PKWs oder LKWs handelt.

Bei den anderen vier Fahrzeugen sind für jedes Fahrzeug dieselben fünf Variablen bekannt. Jeweils zwei Variablen die sich nur auf das einzelne Fahrzeug (*o*, *a*, *of* oder *nf*) beziehen, nämlich die Beschleunigung, sowie die Klasse des Fahrzeuges. Die weiteren drei Variablen beziehen sich auf das relative Verhältnis zu dem SWF. Es handelt sich um den Abstand zum SWF, die Geschwindigkeitsdifferenz zum SWF, sowie der zeitliche Abstand zum SWF.

Im Rahmen der Erstellung der Prognosen wurde deutlich, dass es auch bezüglich der Beschleunigung sinnvoll ist, diese nicht isoliert, sondern ebenfalls im Verhältnis zum SWF zu betrachten¹⁸. Daher werden zu den 25 IV des Datensatzes noch vier weitere Variablen berechnet, die die Beschleunigungsdifferenz zum SWF beschreiben. Diese vier Variable heißen *diffaccPred*, *diffaccFollo*, *diffaccPredAdj* und *diffaccFolloAdj*. Sie werden ganz unten in der Tabelle 1 dargestellt.

In Tabelle 1 werden die 25 IV des Datensatzes, sowie die 4 zusätzlich erstellten Variablen mit einer kurzen Beschreibung und den Einheiten der Variablen, dargestellt.

¹⁸ Da dadurch bessere Prognosen möglich sind.

Tabelle 1: Übersicht der Input Variablen

Name	Beschreibung	Einheit
xVelocity	Geschwindigkeit des Fahrzeugs c .	m/s
yVelocity	Seitliche Geschwindigkeit des Fahrzeugs c .	m/s
xAcceleration	Beschleunigung des Fahrzeugs c .	m/s^2
yAcceleration	Seitliche Beschleunigung des Fahrzeugs c .	m/s^2
Class	Fahrzeugart (PKW oder LKW) des Fahrzeugs c	—
distPred	Abstand zwischen Fahrzeug c und of .	m
diffSpeedPred	Geschwindigkeitsdifferenz zwischen Fahrzeug c und of .	m/s
timeGapPred	Zeitlicher Abstand zwischen Fahrzeug c und of .	s
accPred	Beschleunigung des Fahrzeugs of .	m/s^2
classPred	Fahrzeugart (PKW oder LKW) des Fahrzeugs of .	—
distFollo	Abstand zwischen Fahrzeug c und o .	m
diffSpeedFollo	Geschwindigkeitsdifferenz zwischen Fahrzeug c und o .	m/s
timeGapFollo	Zeitlicher Abstand zwischen Fahrzeug c und f .	s
accFollo	Beschleunigung des Fahrzeugs o .	m/s^2
classFollo	Fahrzeugart (PKW oder LKW) des Fahrzeugs o .	—
distPredAdj	Abstand zwischen Fahrzeug c und nf .	m
diffSpeedPredAdj	Geschwindigkeitsdifferenz zwischen Fahrzeug c und nf .	m/s
timeGapPredAdj	Zeitlicher Abstand zwischen Fahrzeug c und nf .	s
accPredAdj	Beschleunigung des Fahrzeugs nf .	m/s^2
classPredAdj	Fahrzeugart (PKW oder LKW) des Fahrzeugs nf .	—
distFolloAdj	Abstand zwischen Fahrzeug c und a .	m
diffSpeedFolloAdj	Geschwindigkeitsdifferenz zwischen Fahrzeug c und a .	m/s
timeGapFolloAdj	Zeitlicher Abstand zwischen Fahrzeug c und a .	s
accFolloAdj	Beschleunigung des Fahrzeugs a .	m/s^2
classFolloAdj	Fahrzeugart (PKW oder LKW) des Fahrzeugs a .	—
diffaccPred	Beschleunigungsdifferenz zwischen Fahrzeug c und of .	m/s^2
diffaccFollo	Beschleunigungsdifferenz zwischen Fahrzeug c und o .	m/s^2
diffaccPredAdj	Beschleunigungsdifferenz zwischen Fahrzeug c und nf .	m/s^2
diffaccPredFollo	Beschleunigungsdifferenz zwischen Fahrzeug c und a .	m/s^2

Mit diesen 29 Variablen aus Tabelle 1 können die klassischen LCM validiert und die empirischen Modelle erstellt werden. Doch bevor dies geschieht, ist es sinnvoll eine deskriptive Analyse der Variablen durchzuführen.

3.2 Deskriptive Analyse

Nach STELAND (2016) besteht das Ziel einer deskriptiven Analyse darin, empirische Daten durch Tabellen und Grafiken übersichtlich darzustellen und zu ordnen, sowie durch geeignete Kenngrößen zahlenmäßig zu beschreiben [41].

Genau dies erfolgt in diesem Abschnitt anhand des highD-Datensatzes.

Zunächst liegt der Fokus auf der OV, bevor dann untersucht wird welche IV sich am besten eignen um die Ausprägungen der OV zu erklären.

Am Ende des Kapitels wird erläutert warum nur elf Input Variablen für die empirischen Modelle verwendet werden.

3.2.1 Deskriptive Untersuchung der Output Variablen

Als ersten Schritt wird die OV genau untersucht und geprüft, ob Auffälligkeiten erkannt werden, die in den später aufzustellenden Modellen berücksichtigt werden müssen.

In Abbildung 10 wird die Verteilung der OV in Form eines Kreisdiagramms dargestellt.

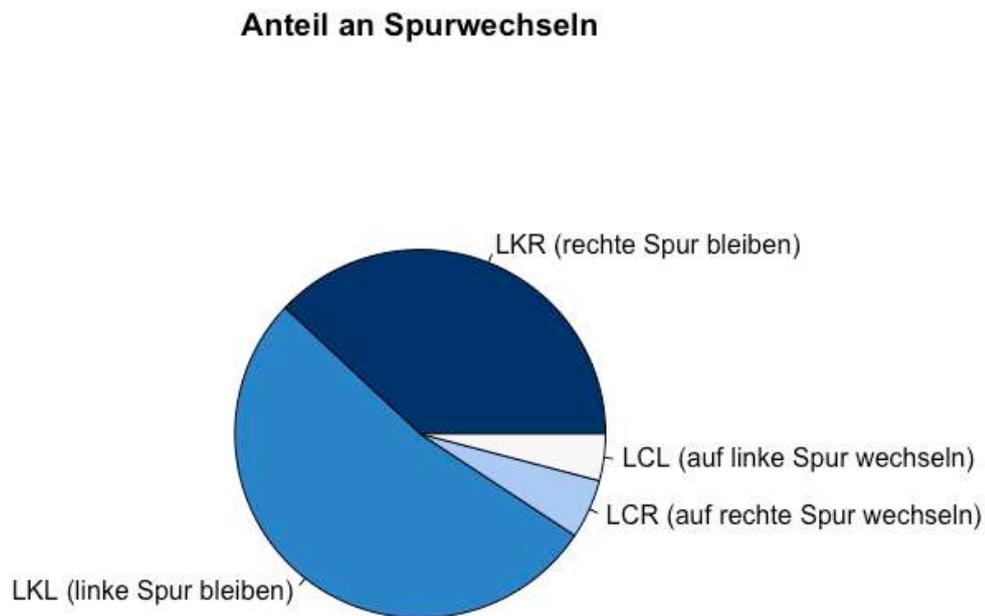


Abbildung 10: Kreisdiagramm der OV

Die beiden großen Flächen des Kreisdiagramms bilden die beiden Manöver bei denen die Spur nicht gewechselt wird (Manöver 1 und 2). In 38% der Beobachtungen bleibt das Fahrzeug auf der rechten Spur und in 53% bleibt es auf der linken Spur. Es kommt also nur bei 9% der Beobachtungen zu einem SW.

Da das Manöver später prognostiziert werden soll, wird hier im Rahmen der deskriptiven Analyse zunächst untersucht, ob sich die IV in den vier Kategorien des Manövers sichtlich unterscheiden. Dazu können die Verteilungen der einzelnen IV in Abhängigkeit der Kategorien der OV in Form von Histogrammen dargestellt und verglichen werden.

Die entsprechenden Histogramme sind in Abbildung 11 abgebildet.

In der ersten Zeile befinden sich nur Histogramme der Variable `distPred`, in der zweiten Zeile Histogramme der Variable `distFolloAdj` und in der dritten Zeile die Histogramme von `diffSpeedAdj`. Es werden pro Variable (pro Zeile) jeweils vier Histogramme dargestellt, das heißt für jede Kategorie ein Histogramm.

In der ersten Spalte ist jeweils das Histogramm des Manövers 1 und in der zweiten Spalte jeweils das Histogramm des Manöver 4 dargestellt. In den Spalten drei und vier werden die Manöver 2 und 3 durch die entsprechenden Histogramme repräsentiert.

Zusätzlich wird in Abbildung 11 noch zu jedem Histogramm der Mittelwert (MW) und die Standardabweichung (SD) der dargestellten Verteilung in dem blauen Kasten angegeben.

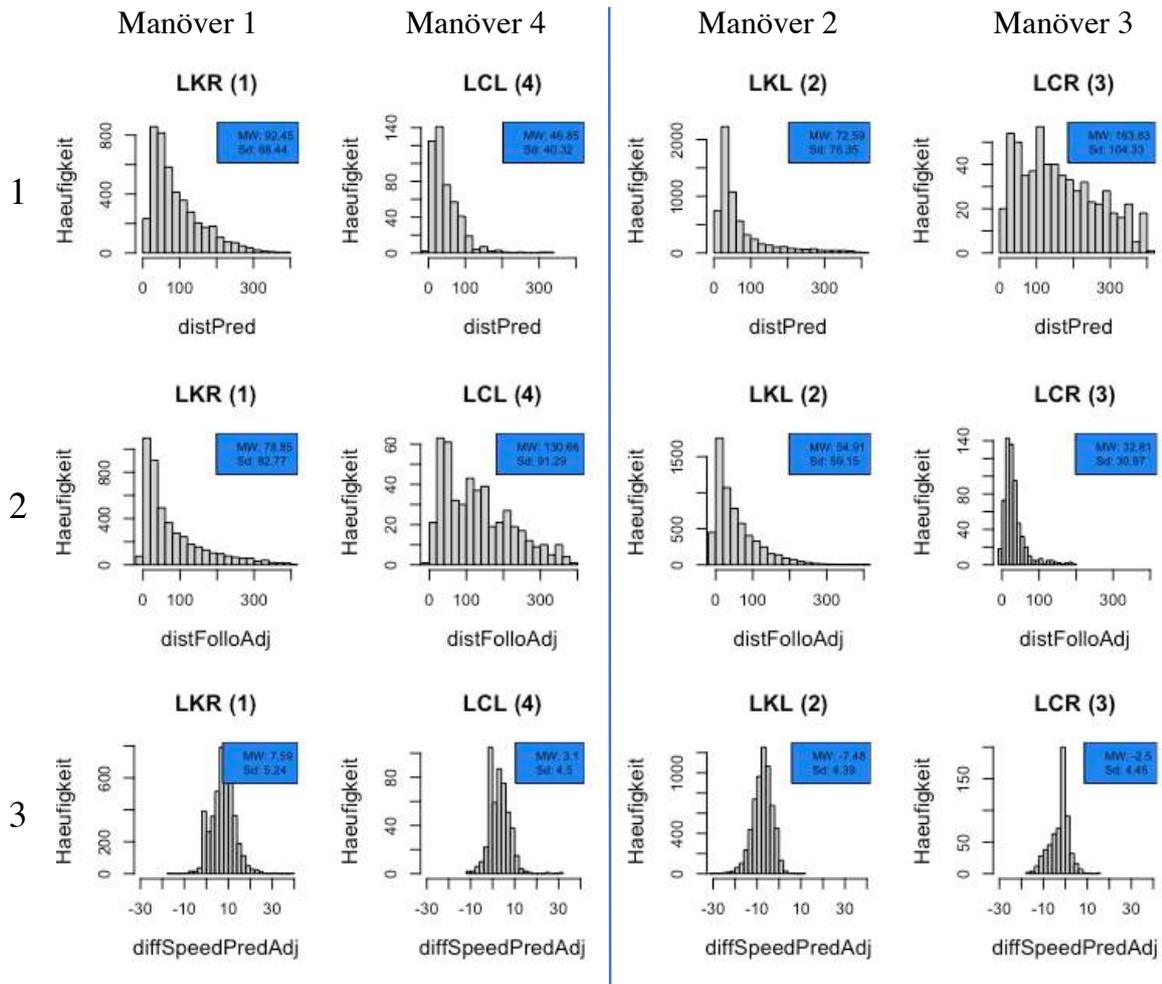


Abbildung 11: Histogramme ausgewählter IV

Die blaue Gerade in der Mitte von Abbildung 11 trennt dann noch die Manöver der Situation aus Abbildung 8 von den Manövern der Situation aus Abbildung 9.

In der ersten Zeile von Abbildung 11 wird der Abstand des SWF zum vorfahrenden Fahrzeug (distPred) dargestellt. Die Verteilung dieser Variable unterscheidet sich bei den kein-SW Manövern (1 und 2) nicht gravierend. Bei den SW Manöver ist dagegen ein deutlicher Unterschied zu erkennen. Der SW auf die rechte Fahrbahn (Manöver 3) wird deutlich häufiger bei Beobachtungen mit einem hohen Abstand zum vorderen Fahrzeug durchgeführt, als dies beim Manöver 4 der Fall ist. Dies wird auch beim Vergleich der beiden MW deutlich. Während der durchschnittliche Abstand beim Manöver 4 bei 46,85 Metern liegt, liegt er beim Manöver 3 bei 163,83 Metern. Der Abstand ist im Durchschnitt also mehr als dreimal so hoch. Auch die SD der beiden Manöver unterscheiden sich bei dieser Variable deutlich.

Ähnliche Resultate wie in Zeile 1 lässt sich auch in Zeile 2 beobachten. Hier ist der Abstand zu dem hinteren Fahrzeug auf der anliegenden Spur in Abhängigkeit der Manöver dargestellt.¹⁹

Auch hier ist kein signifikanter Unterschied zwischen den beiden Manövern, die kein-SW beschreiben zu erkennen. Die Verteilungen sehen fast identisch aus.

Die SW Manövern unterschieden sich jedoch stark!

In diesem Fall wird bei dem Manöver 4 mit 130,66 m ein deutlich höherer Abstand eingehalten als bei Manöver 3, wo der Abstand bei 32,81 m liegt. Der Unterschied diesen Manövern in der Variable `distFolloAdj` ist demnach noch höher als er schon in Zeile 1 von Abbildung 11 zu sehen war.

Die Histogramme der Geschwindigkeitsdifferenzen (`diffSpeedPredAdj`) in der dritten Zeile, unterschieden sich in der Form deutlich von den vorherigen.

Hier fällt auf, dass sich nicht nur die SW Manöver unterschieden, sondern auch die kein-SW Manöver. Beide Manöver der Situation M1&4 weisen einen positiven MW der Geschwindigkeit von $7,59 \frac{m}{s}$ und $3,1 \frac{m}{s}$ auf, während M2&3 einen negative MW von $-7,48 \frac{m}{s}$ und $-2,5 \frac{m}{s}$ ausweisen.

Aus den Erkenntnissen dieser Analyse folgte, dass M1&4 und M2&3 *unterschiedlich behandelt werden müssen*. Dies liegt darin begründet, dass die Intentionen bei dem SW auf die rechte Spur in der Regel andere sind als bei dem SW auf die linke. Für die später folgende Analyse liegt somit nicht mehr nur eine OV mit 4 Kategorien vor, sondern es werden nun zwei OV mit jeweils zwei Kategorien prognostiziert!

3.2.2 Unterscheidung Spurwechsel / kein Spurwechsel

In dem letzten Abschnitt wurde mit Hilfe der Darstellung der Histogramme erläutert, dass es in dem highD-Datensatz sinnvoll ist zwischen zwei Situationen (M1&4 und M2&3) zu unterscheiden.

In diesem Abschnitt soll im Rahmen der deskriptiven Analyse gezeigt werden, wie *innerhalb* dieser Situationen zwischen den Manövern unterschieden werden kann, also zwischen SW und kein-SW.

¹⁹Diese Variable ist wichtig, da es sich bei diesem hinteren Fahrzeug auf der anliegenden Spur um das autonome Fahrzeug handelt, welches die Prognose erstellen soll.

Dazu werden erneut die Histogramme der OV, aufgeteilt in die jeweiligen Manöver, zu betrachten. In Abbildung 12 sind 12 Histogramme von drei unterschiedlichen IV dargestellt. Pro Variable werden Histogramme von allen 4 möglichen Kategorien der OV erstellt. Um zu verdeutlichen, dass diesmal nicht die beiden unterschiedlichen Situationen verglichen werden sollen, sondern stattdessen innerhalb der Situationen, sind diesmal zwei blaue Linien eingetragen. Eine blaue Linie zwischen der ersten M1&4 und eine zwischen M2&3.

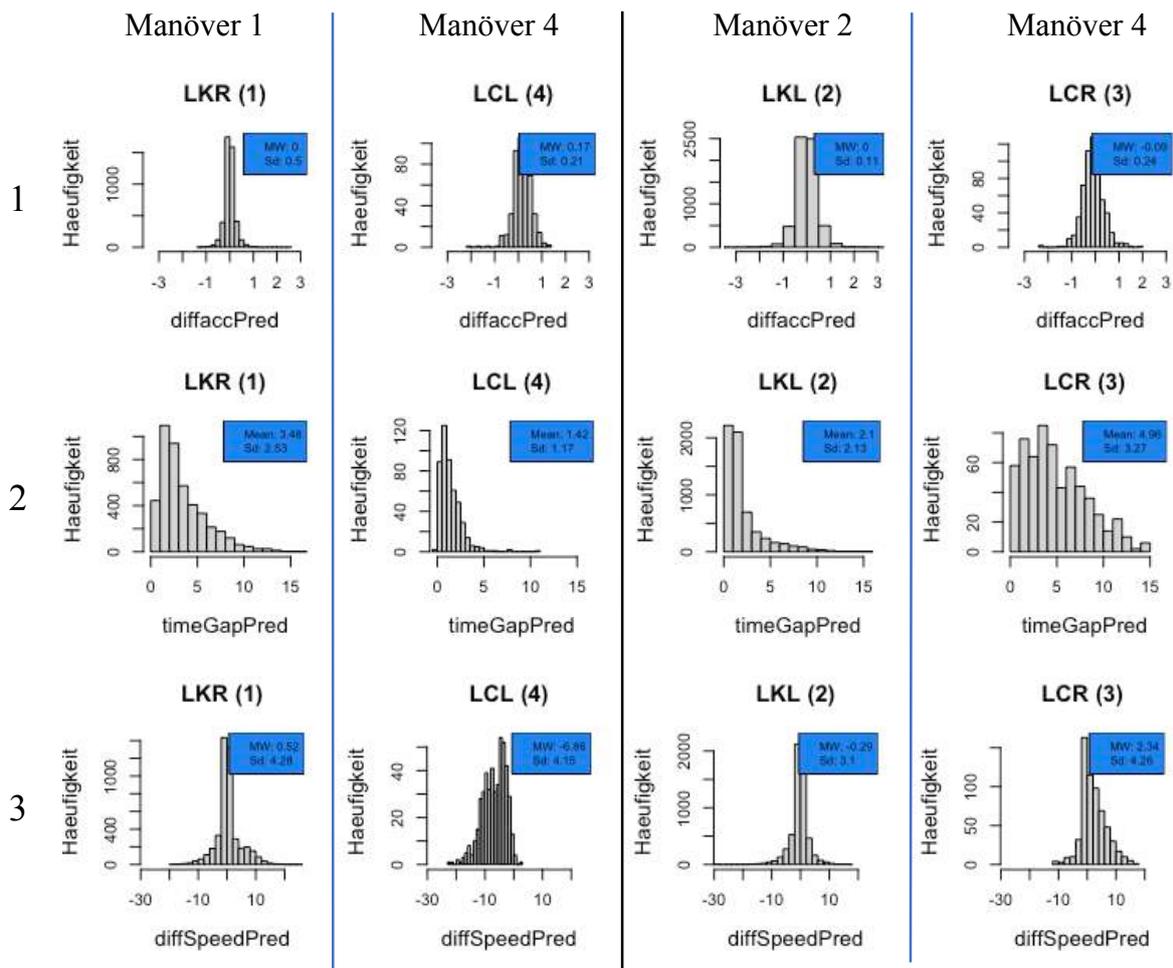


Abbildung 12: Histogramme ausgewählter IV 2

In der ersten Zeile von Abbildung 12 befinden sich die Histogramme der Variable `diffaccPred`, also der Beschleunigungsdifferenz zwischen dem SWF und dem davor fahrenden Fahrzeug. Zwar ist auch hier wieder, wie schon in Abbildung 11, ein Unterschied zwischen den beiden Situationen zu erkennen, doch auffälliger ist noch die Differenz innerhalb der Situationen. Bei Manövern die kein-SW beschreiben ist die Beschleunigungsdifferenz im Mittel genau Null (MW: 0). Bei einem SW auf die linke Spur ist die Beschleunigungsdifferenz im Schritt $0,17 \frac{m}{s^2}$ und bei einem SW auf die rechte Spur liegt der MW bei $-0,09 \frac{m}{s^2}$.

In der zweiten Zeile sind die Histogramme der Variable `timeGapPred` zu sehen, also der zeitliche Abstand zum vorderen Fahrzeug. Auch hier sind deutliche Unterschiede innerhalb der Situationen zu erkennen. Der MW des zeitlichen Abstandes liegt für das Manöver 1 mit 3,48 s mehr als doppelt so hoch wie der des Manövers 4 (1,42 s). Noch deutlicher ist der Unterschied innerhalb von M2&3. Nicht nur, weil die MWs hier noch höhere Differenzen aufweisen, sondern auch, weil sich die Form der Verteilungen sichtbar unterscheidet.

Die letzte Variable in Abbildung 12 ist die Geschwindigkeits-Differenz (`diffSpeedPred`). Bei diesen Histogrammen fällt auf, dass diese Variable bei dem Manöver 1 und dem Manöver 2 im Mittel nahe bei null liegen und symmetrisch um Null schwanken. Bei den SW Manövern sind die Verteilungen klar verschoben. Bei dem Manöver 4 in den negativen Bereich, mit fast allen Werten kleiner als Null und einem MW von $-6,86 \frac{m}{s}$ und bei Manöver 3 in den positiven Bereich mit einem MW von $2,34 \frac{m}{s}$.

Diese Unterschiede in den Variablen werden später genutzt um die Unterscheidung SW/kein-SW prognostizieren zu können.

An diese Erkenntnisse knüpfen nun ergänzend die beiden Abbildungen 13 und 14 an.

Dort werden die zweidimensionalen Häufigkeitsverteilungen der drei IV aus Abbildung 12 dargestellt. Einzelne Beobachtungen werden durch Punkte abgebildet, wobei Beobachtungen eines SW blau und kein-SW Beobachtungen mit schwarz erkenntlich gemacht sind.

Zusätzlich werden noch die Korrelationen zwischen den vorhandenen Variablen dargestellt. Dies geschieht in Form des Korrelationskoeffizienten, welcher nur Werte zwischen -1 und +1 annehmen kann und den Grad des linearen Zusammenhangs der Variablen angibt [42]. Je weiter der Koeffizient von null entfernt ist, desto stärker ist der Zusammenhang.

Die Korrelationskoeffizienten befinden sich unterhalb der Variablennamen, wogegen die zweidimensionalen Häufigkeitsverteilungen über den Variablennamen platziert sind.

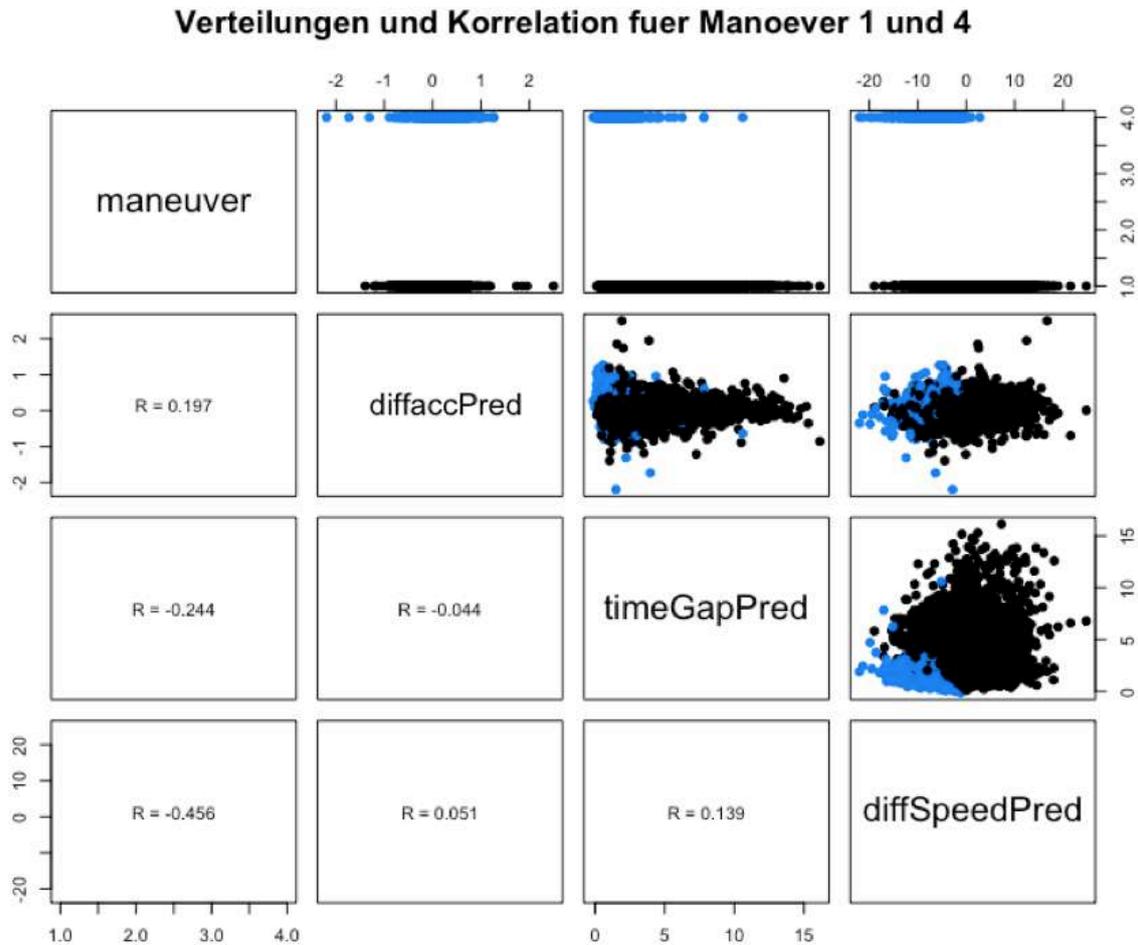


Abbildung 13: Zweidimensionale Häufigkeitsverteilung der Manöver 1&4

Die OV hat einen Korrelationskoeffizienten von 0,197 mit der diffaccPred, einen Korrelationskoeffizienten von $-0,244$ mit timeGapPred und schließlich $-0,456$ mit diffSpeedPred.

Auf der rechten Seite der Abbildung 13 wo die zweidimensionalen Häufigkeitsverteilungen dargestellt werden, fällt auf, dass es teilweise zu einer Art von Clustern von vielen SW in einem kleinen Gebiet kommt. Ganz deutlich zu erkennen ist dies bei der gemeinsamen Verteilung von timeGapPred und diffSpeedPred in der vierten Spalte von Zeile 3. Solche Cluster können für die Prognose des SW wichtig sein, da es so aussieht als wären die Manöver nicht homogenverteilt und somit durch geeignete Werkzeuge unterscheidbar und somit prognostizierbar.

Abbildung 14 ist das Pendant zu mit den zweidimensionalen Häufigkeitsverteilungen und Korrelationen für M2&3.

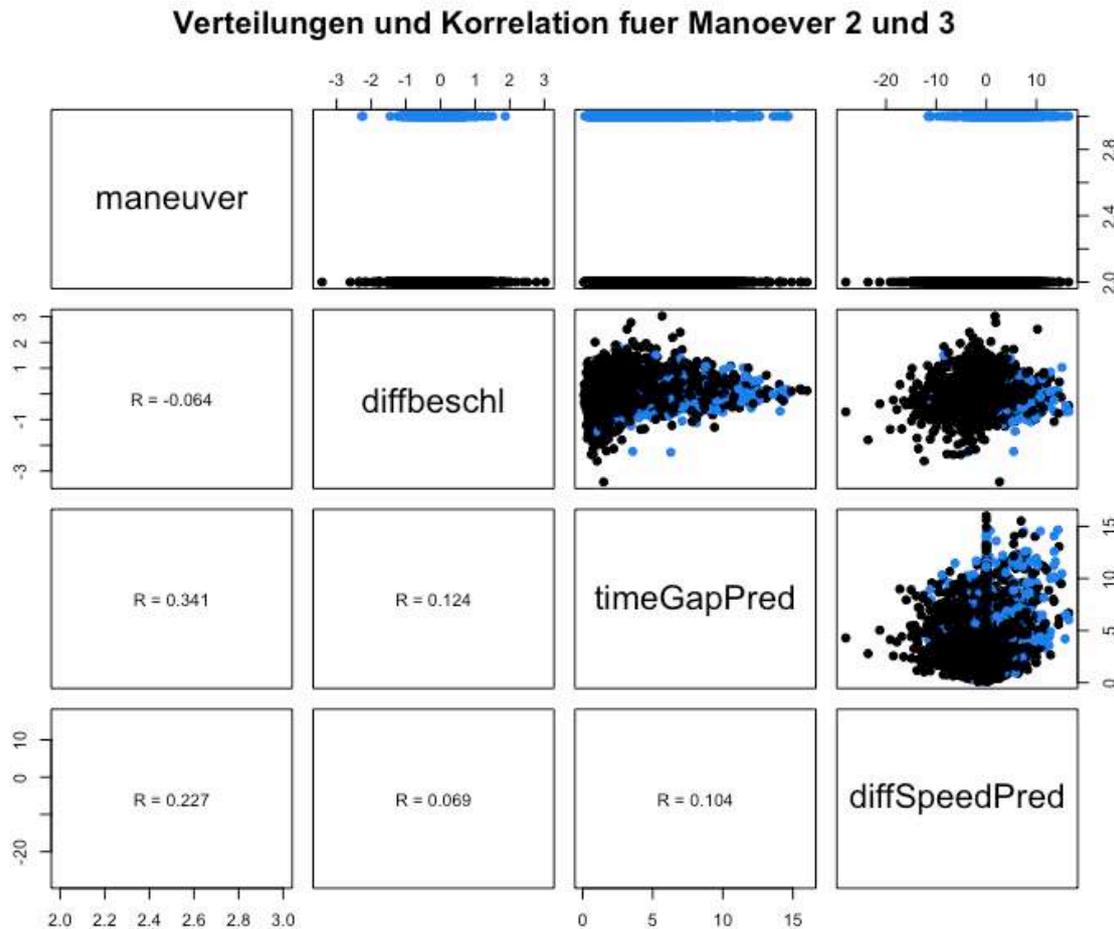


Abbildung 14: Zweidimensionale Häufigkeitsverteilung der Manöver 2&3

Hier lassen sich für die dargestellten Variablen keine so eindeutigen Cluster finden, wie dies in Abbildung 13 zu sehen ist. Dennoch ist Abbildung 14 hilfreich, um nochmal zu verdeutlichen wie unterschiedlich die SW in dieser Situation im Gegensatz zu den SW bei M1&4 verteilt sind. Auch die Korrelationskoeffizienten unterschieden sich deutlich voneinander. Selbst die Vorzeichen sind häufig nicht dieselben, was bedeutet, dass der Zusammenhang in einer umgekehrten Richtung im Vergleich zu dem Zusammenhang aus Abbildung 13 verläuft.

3.2.3 Variablenauswahl für die empirischen Modelle

Bisher konnte durch die deskriptive Analyse gezeigt werden, dass sich die beiden Situationen deutlich unterscheiden und dass es Variablen gibt, innerhalb deren Verteilung deutliche Unterschiede zwischen den SW und kein-SW auftreten. Im letzten Schritt wird hier beschrieben, welche Variablen später für die empirischen Modelle genutzt werden sollten. Dabei ist wichtig, dass für beide Modelle dieselben Variablen genutzt werden, umso die Vergleichbarkeit zu gewährleisten. Das Verwenden einer Variable muss in den empirischen Modellen nicht wirklich begründet werden. Wenn eine Variable einen Erklärungsgehalt besitzt, wird sie verwendet

ohne, dass dies begründet werden muss. Hier wird nun erklärt wieso gewisse Variablen definitiv *nicht* mit in die Analyse einbezogen werden.

Die ersten fünf Variablen in Tabelle 1 beziehen sich alle ausschließlich auf das SWF. Diese fünf Variablen werden alle nicht für die empirischen Modelle genutzt. Die beiden Variablen `yVelocity` und `yAcceleration` beschreiben direkt den Spurwechsel, weshalb sie nicht für die Prognose verwendet werden können.

Die Variable `Class` wird genau wie alle weiteren „class-Variablen“ (`classPred`, `classFollo`, `classPredAdj` und `classFolloAdj`) nicht verwendet. Die später folgenden Modelle sollen nur die SW von PKWs prognostizieren und nicht die von LKWs. Daher hat diese Variable keinen Nutzen für die Prognosen. Des Weiteren werden alle Variablen die den zeitlichen Abstand beschreiben nicht verwendet. Der Grund ist der, dass der zeitliche Abstand hoch mit dem räumlichen Abstand korreliert ist und somit nur eine der beiden Variablengruppe verwendet werden sollte.

Ansonsten werden alle übrigen Variablen, bis auf `diffaccFollo`, genutzt. Es verbleiben elf Variablen die auch in der später folgenden Abbildung 18 nochmal aufgelistet sind.

4.0 Entwicklung und Anwendung der Modelle

In diesem Kapitel werden die Modelle erstellt und die Prognosen berechnet. Es werden zunächst die klassischen LCM kalibriert und validiert. Anschließend werden drei Schritte vorgestellt mit, denen dann die empirischen Modelle trainiert und getestet werden.

Sämtliche Berechnungen wurden in R durchgeführt und die dazugehörigen Skripte befinden sich im Anhang A2 bis A7 [43].

4.1 Zellulare Automaten Modell

Das ZAM von NAGEL und SCHRECKENBERG (1995) ist für diese Ausarbeitung sehr zielführend, da es das Prinzip der Regel-basierten Modelle gut veranschaulicht [17]. Wie in Abschnitt 2.3.1 beschrieben, überlegten sich die Autoren an welche Regeln sich ein Fahrer orientiert, bevor er die Spur wechselt. Aus diesen Überlegungen entstanden die drei Regeln T1, T2 und T3. Da die Regeln ausschließlich Variablen betreffen, die in dem highD-Datensatz aufgeführt werden, lässt sich die Prognosegüte dieses LCM leicht überprüfen.

Für die erste Regel T1 werden der Abstand zum vorderen Fahrzeug (distPred) in die Anzahl der freien Zellen umgewandelt. Dazu wird angenommen, dass die freie Zelle eine Länge von 2 m besitzt. Um von der Variable distPred zu der Anzahl freier Zellen zu kommen muss distPred durch zwei dividiert werden und auf ganze Zahlen gerundet werden²⁰. Bei dem erforderlichen Minimum aus aktueller und maximal zulässiger Geschwindigkeit, wird stets die aktuelle Geschwindigkeit ausgewählt, da eine Geschwindigkeitsbeschränkung nicht bekannt oder wahrscheinlich nicht vorhanden ist, wie dies auf deutschen Autobahnen häufig der Fall ist.

Auch die Geschwindigkeit wird auf ganze Zahlen gerundet, um den Anforderung der diskreten Variablen gerecht zu werden. Wenn die Anzahl an freien Zellen kleiner als die aktuelle Geschwindigkeit ist, ist T1 erfüllt.

Für die Regel T2 ist die Anzahl der freien Zellen zu dem vorderen Fahrzeug auf der anliegenden Spur (distPredAdj), sowie die aktuelle Geschwindigkeit von Bedeutung. Das Vorgehen zur Umsetzung dieser Regel ist das Gleiche wie bei T1. Ist hier die Anzahl an freien Zellen größer als die aktuelle Geschwindigkeit, so ist T2 erfüllt.

²⁰ distPred besitzt die Einheit Meter. Liegt distPred bspw. bei $3,5\text{m}$, so liegt eine freie Zelle vor.

Als letztes folgt Regel T3, bei der die Anzahl der freien Zellen zu dem hinteren Fahrzeug auf der anliegenden Spur (`distFolloAdj`) und die aktuelle Geschwindigkeit die entscheidenden Variablen sind. Ist die Anzahl an freien Zellen größer als die aktuelle Geschwindigkeit, dann ist auch diese Regel erfüllt. Im Anhang 2 wird dargestellt, wie sich diese Regeln in R implementieren lassen.

Sollten alle drei Regeln für ein Fahrzeug in der gegebenen Beobachtung erfüllt sein, wird ein SW prognostiziert. Werden die Regeln des ZAM auf die Beobachtungen des `highD`-Datensatzes angewandt, ergeben sich die Ergebnisse der Tabelle 2.

In der ersten Zeile ist der Fehler in Prozent angegeben. Dieser ergibt sich aus dem Anteil der (falschen) Prognosen die nicht mit den wahren Beobachtungen übereinstimmen. In der zweiten Zeile sind die Wahrheitstafeln abgebildet. Von Wahrheitstafeln kann abgelesen werden, woraus sich der Fehler zusammensetzt [44]. In der ersten Zeile der Wahrheitstafeln werden die Beobachtungen eingetragen, die kein-SW sind. In der ersten Spalte dieser Zeile sind die Fälle, wo kein-SW prognostiziert wurde und auch wirklich kein-SW vorliegt (richtig negativ). In der Spalte daneben wurden fälscherweise SW prognostiziert (falsch positiv). In der zweiten Zeile befinden sich die SW. In der ersten Spalte liegen sie, wenn sie falsch prognostiziert wurden (falsch negativ) und in der zweiten, wenn richtig prognostiziert wurde (richtig positiv).

		Prognose	
		kein-SW	SW
Wahre Beobachtung	kein-SW	richtig negativ	falsch positiv
	SW	falsch negativ	richtig positiv

Abbildung 15: Aufbau von Wahrheitstafeln

In der Abbildung 15 wird der schematische Aufbau der Wahrheitstafeln dargestellt.

In der letzten Zeile der Tabelle 2 befindet sich der wahre Anteil der SW in der vorliegenden Stichprobe. Dieser Wert wird aufgeführt, da er als Referenzgröße dienen kann, um die Güte der Prognosen zu beurteilen. Wenn ein Modell nie einen SW prognostiziert, weiß es einen Fehler auf, der genau dieser Referenzgröße entspricht.

Tabelle 2: Ergebnisse des zellularen Automaten Modells

	M1&4	M2&3
Fehler (in %)	21,64	14,84
Wahrheitstafel	$\begin{bmatrix} 2073 & 204 \\ 388 & 71 \end{bmatrix}$	$\begin{bmatrix} 5700 & 425 \\ 570 & 8 \end{bmatrix}$
Anteil Spurwechsel (in %)	16,78	8,62

Wie in der Tabelle 2 zu erkennen ist, prognostiziert ZAM bei M1&4 über 21 % der Manöver falsch. In den Wahrheitstafeln wird zudem deutlich, dass von den 459 vorliegenden SW (Manöver 4) nur 71 (15 %) richtig prognostiziert worden sind. Das Referenzmodell weist einen geringeren gesamten Fehler auf (16,78 %).

Für M2&3 fällt der Fehler mit 14,84 % etwas kleiner aus. Doch in den Wahrheitstafeln ist zu erkennen, dass der Anteil der prognostizierten SW an den wahren SW (richtig positive) sogar noch kleiner ist. Nur 8 der 578 beobachteten SW konnten prognostiziert werden. Auch hier ist der Referenzwert mit einem Fehler von 8,623 % kleiner als der Fehler des Modells.

4.2 MOBIL

Bei der Implementierung von ZAM lag der Umstand vor, dass keine Parameter benötigt worden und alle nötigen Variablen im highD-Datensatzes vorhanden waren.

Bei MOBIL ist dies nicht der Fall. Wie in Formel (5) dargestellt wird, beinhaltet das Modell beispielsweise die Variable \tilde{a}_c , welche die (Wunsch-) Beschleunigung nach dem SW angibt und somit logischerweise für die meisten Fahrzeuge nicht beobachtbar ist. Demnach ist sie auch nicht im highD-Datensatz vorhanden. Bevor die Prognosen aufgestellt werden können, muss diese Variable erst berechnet werden. Dafür wird ein Fahrzeugfolge-Modell verwendet. Dieses soll den Verkehr simulieren und dadurch ermöglichen, dass die fehlende Variable der Simulation entnommen werden kann. In dem Artikel von TREIBER (2007) nutzen die Autoren das „intelligent driver model (IDM)“ von TREIBER (2002) [14, 45]. In dieser Arbeit wird auch das IDM implementiert.

Die Fahrweise eines Fahrzeuges wird im IDM durch die Variablen: eigene Geschwindigkeit v_α , Abstand zum vorliegenden Fahrzeug s_α und Geschwindigkeitsdifferenz zwischen den beiden Fahrzeugen Δv , determiniert [45].

Diese IV sind alle durch die Variablen $xVelocity$, $distPred$ und $diffSpeedPred$ in dem highD-Datensatz vorhanden. Doch neben diesen IV werden noch fünf Parameter benötigt, um den Verkehr simulieren zu können. Dies ist die gewünschte Geschwindigkeit v_0 , die das Fahrzeug auf einer leeren Straße fahren würde, der sichere zeitliche Abstand T zum vorliegendem Fahrzeug, der geringste mögliche Abstand zum vorliegendem Fahrzeug s_0 , sowie typische Beschleunigungen a und Verlangsamungen b im Verkehr [46].

Mit Hilfe dieser Parameter und der drei IV können die gesuchte Beschleunigungen a_i mit Hilfe von Formel (29)

$$a_i = a \left[1 - \left(\frac{v_\alpha}{v_0} \right)^4 - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2 \right] \quad (29)$$

sowie der Formel (30)

$$s^*(v_\alpha, \Delta v_\alpha) = s_0 + \max \left[0, \left(v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}} \right) \right] \quad (30)$$

berechnet werden.

Damit lassen sich die Beschleunigungen a_c , a_n und a_o der drei relevanten Fahrzeuge bestimmen. Nun fehlen noch die gewünschten/ zukünftigen Beschleunigungen \tilde{a}_c , \tilde{a}_n und \tilde{a}_o nach dem SW. Diese lassen sich auch mit Hilfe von Formel (29) und Formel (30), unter Anpassung der IV, bestimmen. Anstatt des Abstandes zum vorliegendem Fahrzeug ($distPred$) und der Geschwindigkeitsdifferenz zum vorfahrenden Fahrzeug ($diffSpeedPred$), muss dann der Abstand zum vorliegenden Fahrzeug auf der anliegenden Spur ($distPredAdj$), sowie die Geschwindigkeitsdifferenz zum vorderen Fahrzeug auf der anliegenden Spur ($diffSpeedPredAdj$) eingesetzt werden.

Mit Hilfe des IDM können nun die Beschleunigung, sowie die gewünschte Beschleunigung berechnet werden. Zusätzlich verfügt MOBIL noch über vier Parameter, die noch berechnet werden müssen. Wie in Abschnitt 2.3.2 erläutert, sind dies der Höflichkeitsfaktor p , die Wechselschwelle Δa_{th} , das sichere Bremsen b_{safe} , sowie der zusätzlichen Parameter Δa_{bias} für den asymmetrischen Verkehr.

Die letzten beiden Parameter werden aus dem Artikel von TREIBER (2007) übernommen und auch so in R implementiert [14]. Die Parameter p und Δa_{th} werden mit Hilfe einer Optimierungsfunktion ermittelt. Dadurch sollen die beiden Parameter so geschätzt werden, dass der Fehler minimiert wird. Die Ergebnisse der Parameterschätzung und des Fehlers werden nun im nächsten Abschnitt vorgestellt.

4.2.1 Ergebnisse von MOBIL

Für die Berechnung der Prognosen mit MOBIL werden zunächst die IV in das IDM eingesetzt. Die Variablen, die sich aus dem IDM ergeben, werden dann mit den Parametern in Formel (6) oder Formel (7) eingesetzt. Ist der Wert, der sich daraus ergibt, größer als die Wechselschwelle, so wird ein SW prognostiziert

Bisher sind die beiden Parameter Höflichkeitsfaktor und Wechselschwelle noch nicht bestimmt worden. Diese beiden Parameter werden nun mit Hilfe einer Optimierungsfunktion so gewählt, dass der Fehler der Prognosen minimiert wird. Abbildung 16 wurde mit dem R-Skript aus Anhang A3 erstellt und zeigt die Abhängigkeit der Optimierungsfunktion von den beiden Parametern.

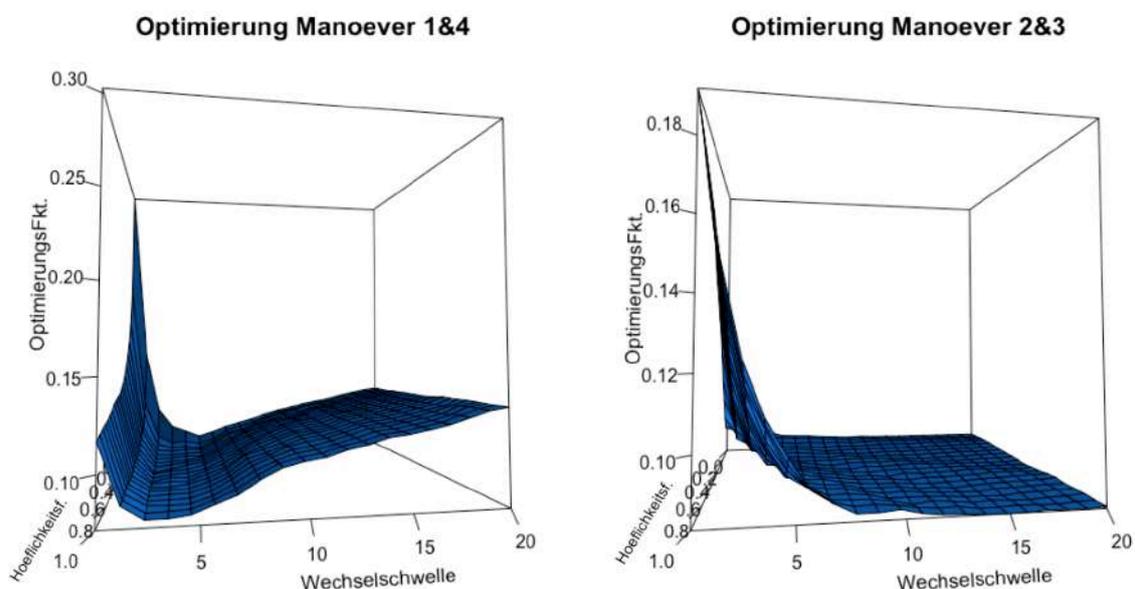


Abbildung 16: Optimierung der Parameter von MOBIL

Die linke Darstellung in Abbildung 16 ist die Optimierungsfunktion von M1&4. Auf den beiden horizontalen Achsen sind die Wechselschwelle und der Höflichkeitsfaktor abgebildet und auf

der vertikalen Achse der Fehler der Optimierungsfunktion. Es soll das globale Minimum dieser in blau dargestellten dreidimensionalen Funktion gefunden werden. Für die Wechselschwelle ist es bei M1&4 möglich das globale Minimum zu finden. Die Funktion besitzt einen U-förmigen Verlauf, wodurch das Minimum im Tal liegen muss. Ungefähr bei einer Wechselschwelle von 2 bis 4 $\frac{m}{s^2}$.

Die Optimierungsfunktion von M2&3, ist in der rechten Darstellung von Abbildung 16 zu sehen. Diese Funktion weist keine U-Form auf, sondern ein Plateau in dem Bereich der Wechselschwelle zwischen 8 bis 20 $\frac{m}{s^2}$. Es ist keine Tendenz zu einem globalen Minimum erkennbar.

Einzig, das starke Ansteigen des Fehlers bei einer Wechselschwelle die kleiner als 8 $\frac{m}{s^2}$ ist, fällt auf.

Die Ergebnisse von MOBIL werden in Tabelle 3 dargestellt. In der ersten Zeile wird der Wert des Höflichkeitsfaktors p angegeben und in der zweiten Zeile der Wert der Wechselschwelle Δa_{th} . Dann folgt, wie schon in Tabelle 2, der Fehler, die Wahrheitstafeln und der Anteil an SW.

Tabelle 3: Ergebnisse von MOBIL

	M1&4	M2&3
Höflichkeitsfaktor p	0,48	0,93
Wechselschwelle Δa_{th} in $\frac{m}{s^2}$	2,3	14,1
Mittlerer Fehler in %	7,27	8,2
Wahrheitstafeln	$\begin{bmatrix} \mathbf{2207} & 70 \\ 129 & \mathbf{330} \end{bmatrix}$	$\begin{bmatrix} \mathbf{6103} & 22 \\ 528 & \mathbf{50} \end{bmatrix}$
Anteil Spurwechsel (in %)	16,78	8,62

Wie bereits in Abbildung 16 zu entnehmen war, liegt das globale Minimum für M1&4 im unterem-einstelligen Bereich bei 2,3 $\frac{m}{s^2}$. Der Höflichkeitsfaktor wurde auf 0,48 geschätzt. Der mittlere Fehler liegt bei 7,27 %, wobei das kein-SW Manöver (Manöver 1), wie in der Wahrheitstafel zu erkennen ist, deutlich besser prognostiziert wurde (Fehler 3 %). Von den 459 vorliegenden SW konnten 330 richtig prognostiziert werden (59 %). Die Referenzgröße liegt mit 16,78 % deutlich über dem Fehler von MOBIL.

Für M2&3 liegt die Wechselschwelle in dem Plateau aus Abbildung 16, nämlich bei $14,1 \frac{m}{s^2}$. Auch der Höflichkeitsfaktor nimmt einen hohen Wert (nahe 1) von 0,93 an. Der Fehler liegt bei 8,2 %, wobei in den Wahrheitstafeln zu erkennen ist, dass weniger als 9 % der SW (50 von 578) richtig prognostiziert worden sind. Der Referenzwert ist mit 8,623 % kleiner als der Fehler von MOBIL.

4.3 Das highD-Modell

Bisher wurden mit ZAM und MOBIL zwei klassische LCM vorgestellt. Das ZAM gehört zu den regel-basierten Modellen, bei dem die Spurwechselentscheidung anhand einfacher elementarer Regeln getroffen werden und MOBIL ist ein anreiz-basiertes Modell, bei dem Anreiz- und Sicherheitskriterien erfüllt sein müssen.

Im Folgenden soll ein Modell entwickelt werden, das auch aus einfachen Regeln besteht. Anders als bei ZAM sollen diese Regeln jedoch an Hand des beobachteten Spurwechselverhaltens des highD-Datensatzes hergeleitet werden. Demnach handelt es sich um ein klassisches LCM, welches einen empirischen Ursprung besitzt.

Bei der Entwicklung von ZAM gingen die Autoren so vor, dass sie sich überlegten, nach welchen elementaren Regeln die Fahrer ihre Spurwechselentscheidungen treffen. In Kapitel 4.1 wurden diese Regeln dann implementiert und geprüft, ob sich damit die SW des highD-Datensatzes prognostizieren lassen. In dem folgenden Modell wird umgekehrte vorgegangen. Zuerst werden die Daten analysiert und nach einem Muster (Cluster) gesucht, welches vielversprechend für die Prognose des SW sein könnte. Anschließend wird versucht auf Basis dieses Musters, einfache Regeln zu entwickeln, die sich dann, ähnlich wie die Regeln des ZAM, implementieren lassen. Somit folgt aus den Daten die Erklärung für einen SW.

Ein Muster, welches dafür prädestiniert erscheint, findet sich in Abbildung 17. Dort ist ein Cluster in der 3. Spalte der 2. Zeile mit vielen SW in einem kleinen Bereich zu erkennen. Dieses Cluster wird Abbildung 17 durch die roten Grenzen markiert.

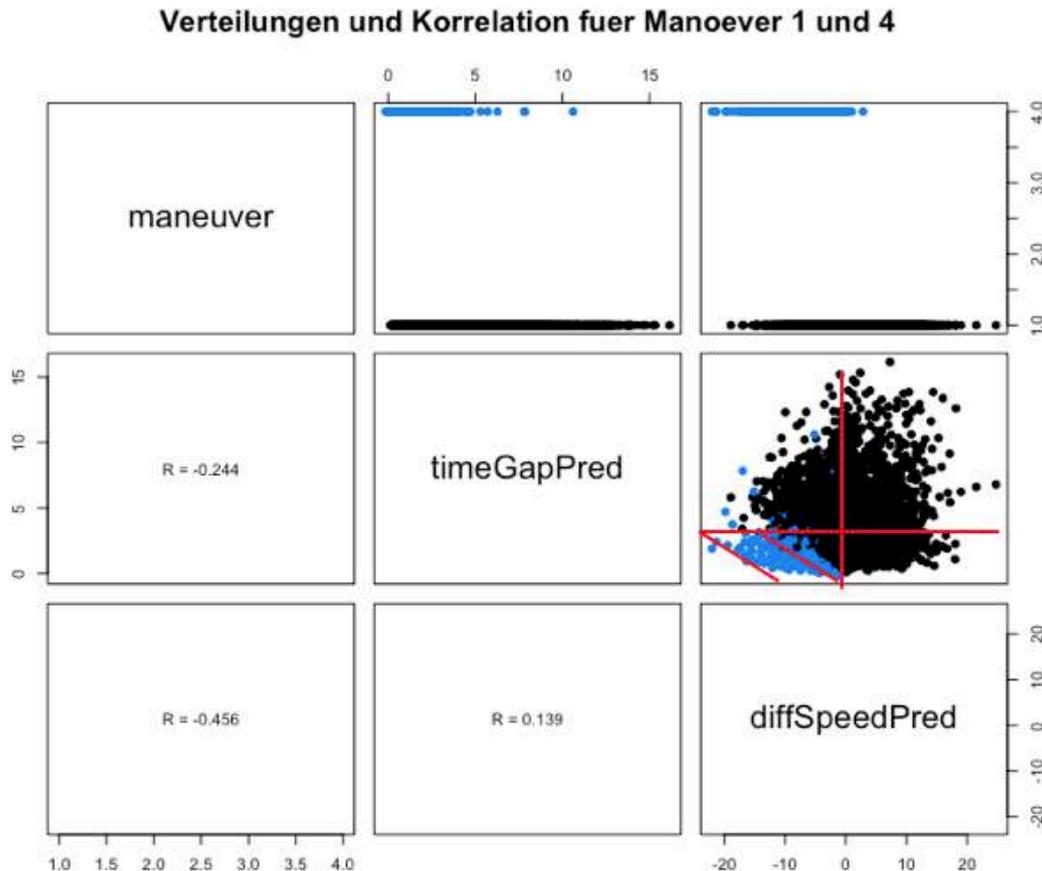


Abbildung 17: Häufigkeitsverteilung des highD-Modells

Dieser eingegrenzte Bereich in Abbildung 17, in dem sich offensichtlich ein Großteil der SW von M1&4 befinden, lässt sich einfach mit einem einfachen Modell abbilden. Dafür werden die Variablen `timeGapPred` und `diffSpeedPred`, sowie zwei Parameter benötigt. Die Parameter werden durch die vertikale- und die horizontale rote Linie in Abbildung 17 repräsentiert. Für die Prognose werden zwei Ungleichungen benötigt, die beide erfüllt sein müssen, damit ein SW prognostiziert wird. Die erste Ungleichung ergibt sich aus dem ersten Parameter und der Variable `timeGapPred`. Dieser Parameter beschreibt die horizontale rote Linie in Abbildung 17. Ist der Wert von `timeGapPred` kleiner als der Parameter ist, so ist die erste Ungleichung erfüllt. Die zweite Ungleichung ergibt sich aus dem zweiten Parameter und der Variable `diffSpeedPred`. Der zweite Parameter wird durch die vertikale rote Linie in Abbildung 17 angegeben und ist `diffSpeedPred` kleiner als dieser Parameter, so ist auch die zweite Ungleichung erfüllt. Die beiden Parameter lassen sich schon grob anhand der roten Linien ablesen und durch Ausprobieren ergaben sich ein Wert von 2,75 s für den ersten und $-2,9 \frac{m}{s}$ für den zweiten Parameter. Diese zwei Ungleichungen ergeben die Spurwechselentscheidung, welche besagt, dass die Spur von der rechten Fahrbahn auf die linke gewechselt wird (Manöver 4), wenn:

der zeitliche Abstand kleiner als 2,75 s ist
und
das vordere Fahrzeug mindestens 2,9 m/s langsamer fährt.

Diese beiden Regeln werden nun wieder in Form einer Bedingten-Anweisung in R umgesetzt (siehe Anhang A4).

Die Ergebnisse der Prognosen für M1&4 sind in Tabelle 4 dargestellt.

Tabelle 4: Ergebnisse des highD- Modells

	M1&4
Fehler (in %)	7,9
Wahrheitstafel	$\begin{bmatrix} 2170 & 107 \\ 109 & 350 \end{bmatrix}$
Anteil Spurwechsel (in %)	16,78

Von den insgesamt 2736 Beobachtungen, die für M1&4 in dem highD-Datensatz vorhanden sind, werden 2520 richtig prognostiziert. Dies ergibt einen Fehler von 7,9 %.

Somit konnten mit Hilfe des Datensatzes die Regel erkannt werden, dass ein *geringer zeitlicher Abstand zu dem vorderen Auto* und *eine gleichzeitig hohe Geschwindigkeitsdifferenz zu dem vorfahrenden Fahrzeug* relativ zuverlässige Indikatoren für einen SW von der rechten auf die linke Spur sind (Manöver 4).

Für M2&3 konnte mit Hilfe der Datenanalyse kein zwei oder drei parametrisches Modell erstellt werden, welches den Referenzwert (8,6 %) dieser Situation übertrifft. Dies wird auch in Abbildung 14 deutlich, in er kein vergleichbares Cluster des Manövers 3 zu finden ist.

4.4 Logistische Regression

In Abschnitt 2.3.2 wurden die mathematischen Grundlagen der logReg vorgestellt. In diesem Kapitel sollen die Implementierung und die Ergebnisse präsentiert werden. Dazu wird sich an drei Schritten orientiert, die an die Veröffentlichung von BOX und JENKINS (1976) angelehnt sind [47, 48]:

1. Modell formulieren
2. Parameter schätzen und Modell auswählen
3. Prognose erstellen

Im ersten Schritt geht es darum, das Modell grob zu definieren und IV sowie Output-Möglichkeiten festzulegen [25]. Im zweiten Schritt werden verschiedene Modelle aufgestellt und das Beste ausgewählt [24]. Im letzten Schritt werden die Ergebnisse der Prognosen dargestellt.

Üblicherweise wird der Datensatz für die letzten beiden Schritte jeweils in Trainingsdaten, Validierungsdaten und Testdaten aufgeteilt, umso die Auswahl und Bewertung durchführen zu können. Im Rahmen dieser Analyse wird der Datensatz nur in zwei Teile aufgeteilt: Trainings- und Testdaten. Die Validierungsdaten werden durch das Bootstrap-Verfahren (BTV) sowie den Akaike-Informationskriterium (AIC) substituiert [24].²¹

4.4.1 Modell formulieren

Die Modellformulierung umfasst Handlungen und Entscheidungen, die vor den Berechnungen der Parameter durchzuführen sind. So müssen zunächst die IV und die Kategorien der OV bestimmt werden.

Wie in Abschnitt 3.2.3 erläutert, werden elf IV für die Prognosen mit den empirischen Modellen genutzt. Außerdem werden die Prognosen für zwei OV bestimmt, welche jeweils zwei Ausprägungsmöglichkeiten besitzen (binäre Variable). Die Aufteilung des Datensatzes erfolgt so, dass 75 % der Daten zum Training und die restlichen 25 % zum Testen verwendet werden. Durch das Training werden die Parameter β_i der logReg berechnet und mit Hilfe des Testens werden die Testdaten in das Modell mit den zuvor bestimmten Parametern eingesetzt, um dann der Output \hat{y} mit dem wahren Output y verglichen.

²¹ „if we are in a data-rich situation, the best approach for both problems is to randomly divide the dataset into three parts: a training set, a validation set and a test set...The methods of this chapter approximate the validation step either analytically (AIC, BIC) or by efficient sample re-use (crossvalidation and the bootstrap)“ [24]

4.4.2 Parameter schätzen und Modell auswählen

In dem zweiten Schritt werden nun normalerweise beim maschinellen Lernen viele verschiedene Modelle aufgestellt und berechnet, um dann das beste Modell auswählen zu können. Bei der logReg ist dieser Schritt ziemlich einfach, da die Parameter mit Hilfe der MLM geschätzt werden. Bei der Likelihood-Funktion handelt es sich um eine konkave Funktion, weshalb stets ein globales Maximum und somit das optimale (*lineare*) Modell gefunden werden kann.²² Dadurch müssen keine unterschiedlichen Startwerte und andere Einstellungen ausprobiert werden. Die Modellauswahl hängt nur von den IV ab und diese wurden bereits in Abschnitt 3.2.3 festgelegt. Dadurch ist die Anwendung des AIC und BTV erst bei der Bewertung des Modells nötig.

Die Parameter können direkt geschätzt werden und es kann davon ausgegangen werden, dass das beste lineare Modell gefunden wird.

Abbildung 18 zeigt beispielhaft den Output eines Trainingsdatensatzes für M1&4.²³ Dieser wurde mit dem R-Skript in Anhang A5 berechnet.

```

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.267425   0.339560  -0.788 0.430952
XdistPred    -0.053785   0.004208 -12.783 < 2e-16 ***
XdistFollo   -0.001947   0.001783  -1.092 0.274771
XdistPredAdj -0.007107   0.002000  -3.553 0.000381 ***
XdistFolloAdj  0.004983   0.001352   3.685 0.000229 ***
XdifffSpeedPred -0.784115   0.053109 -14.764 < 2e-16 ***
XdifffSpeedFollo  0.136464   0.032357   4.217 2.47e-05 ***
XdifffSpeedPredAdj -0.076398   0.029904  -2.555 0.010626 *
XdifffSpeedFolloAdj -0.184589   0.031849  -5.796 6.80e-09 ***
Xbesch1      1.098476   0.501082   2.192 0.028365 *
Xbesch       1.632611   0.327743   4.981 6.31e-07 ***
Xbesch2      0.820447   0.309760   2.649 0.008081 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Abbildung 18: Output der logReg

In der ersten Spalte von Abbildung 18 sind die Namen der elf IV sowie die Konstante (intercept) abgebildet. In der zweiten Spalte befinden sich die geschätzten Parameter β_i und in der dritten Spalte der dazugehörige Standard Fehler [25]. Der so genannte z-Wert in der dritten Spalte gibt

²² „Da... eine konkave Funktion ist, ist $L(\beta)$ konkav in β und besitzt daher ein globales Maximum.“ [49]

²³ Da später das BTV verwendet wird für die Prognosen werden viele solcher Outputs berechnet. Daher ist dieser nur ein Beispiel, wie die Parameter und Test-statistiken aussehen könnten. Bei anderen Trainingsdatensätzen weicht der Output ab.

das Verhältnis zwischen dem Parameter und dem Standard Fehler an. In der letzten Spalte wird das Ergebnis eines Wald-Tests abgebildet. Bei diesem Wald-Test wird geprüft, ob die jeweilige IV einen signifikanten Einfluss auf die OV hat. Wie in der Abbildung 18 zuerkennen ist, haben sieben der elf IV für diesen Trainingsdatensatz einen Einfluss, der selbst bei einem Signifikanzniveau von 0,001 nicht abgelehnt werden kann. Nur der Einfluss von distFollo scheint nicht signifikant zu sein. Alle anderen 10 Variablen haben einen signifikanten Einfluss (bei einem Signifikanzniveau von 0,05).

Die Interpretation dieser Parameter und Teststatistiken aus Abbildung 18 ist beim maschinellen Lernen nicht von großer Bedeutung, wie es in anderen Bereichen der Statistik der Fall ist. Bei anderen Methoden des maschinellen Lernens wie den im nächsten Abschnitt folgenden Neuronalen Netze ist eine Interpretation der Parameter nicht einmal möglich.

Da die Parameter β_i der logReg nun bekannt sind (für M1&4) müssen nur noch die Variablen einer Beobachtung in Formel (10) eingesetzt werden woraus sich schließlich die Wahrscheinlichkeit eines SW ergibt.

4.4.3 Prognose erstellen

In diesem Abschnitt werden Kennzahlen vorgestellt, die zur Bewertung der Prognosen genutzt werden. Dies sind der Fehler, die Verteilung des Fehlers, das AIC und die Wahrheitstafeln.

Diese Kennzahlen werden dabei aber nicht einfach einmal berechnet, sondern es wird das BTW angewendet. Dazu wird eine Stichprobe aus dem Datensatz gezogen. Diese wird dann genutzt, um das Modell zu trainieren und zu testen. Anschließend wird die Stichprobe wieder „zurückgelegt“ und eine neue Stichprobe wird gezogen. Zurückgelegt bedeutet, dass diese Beobachtungen weiterhin im Datensatz vorhanden sind und bei der nächsten zufälligen Stichprobenziehung erneut ausgewählt werden könnten [50].²⁴ Das Ziehen, Trainieren und Testen von Stichproben wird einige tausendmal durchlaufen.

Dieses Vorgehen hat den Vorteil, dass zum einen keine Validierungsdaten nötig sind und zum anderen nicht nur die Kennzahl, sondern gleich die (empirische-) Verteilung der Kennzahl geschätzt wird. Dadurch kann die Sd berechnet und mit dieser dann die Konfidenzintervalle aufgestellt werden. Außerdem lassen sich dadurch Hypothesentest wie z.B. der t-Test durchführen.

²⁴ Man spricht in der Statistik von Resampling.

Abbildung 19 zeigt die Ergebnisse der Berechnung des Fehlers für M1&4 in Form eines Histogramms. Der Fehler wurde 10 000-mal berechnet.

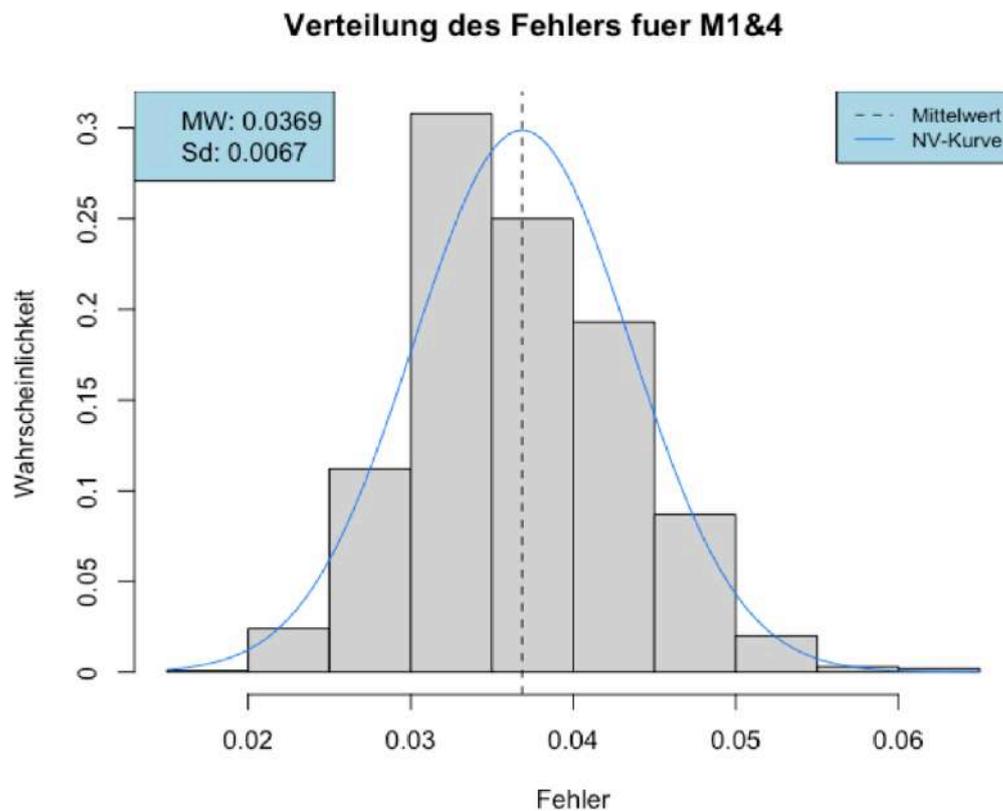


Abbildung 19: Verteilung des Fehlers für M1&4

In Abbildung 19 sind neun Balken dargestellt. Die Höhe der Balken gibt an wie viel Prozent der Fehler in diesem Bereich liegen. In dem Bereich, der die Fehler zwischen 0,03 bis 0,035 beinhaltet, liegen beispielweise ca. 31 % der durchgeführten Berechnungen. Der MW des Fehlers liegt bei 0,0368 (3,68 %), wie durch die gestrichelte Linie dargestellt wird. Neben dieser Linie ist noch eine blaue Kurve in Abbildung 19 zusehen, bei der es sich um eine Normalverteilungskurve (NV-Kurve) handelt. Die NV-Kurve wurde so eingestellt, dass sie den selben MW und dieselbe Sd besitzt wie der Fehler. Die Verteilung des Fehlers, die durch die Balken dargestellt wird, scheint nicht gravierend von der abgebildeten NV-Kurve abzuweichen.

Analog zu M1&4 wird nun das Histogramm des Fehlers bei M2&3 in Abbildung 20 dargestellt. Die Vorgehensweise ist dieselbe wie in Abbildung 19, nur die Anzahl der Balken wurde anders gewählt.

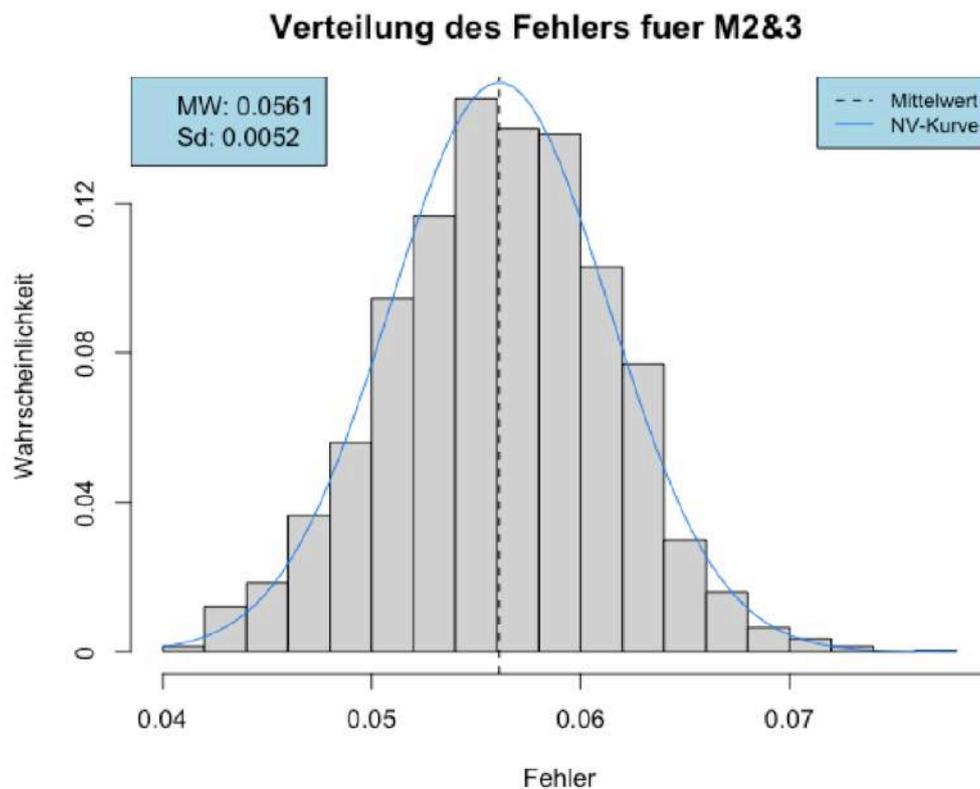


Abbildung 20: Verteilung des Fehlers für M2&3

Der MW des Fehlers liegt für M2&3 bei 5,61 % und die Sd liegt bei 0,52 %. Auch hier wird eine NV-Kurve eingezeichnet, die dieselbe Sd und MW besitzt. Diesmal ist die Ähnlichkeit der Verteilung des Fehlers zu der NV-Kurve noch größer als in Abbildung 19. Insgesamt fällt der Fehler für M2&3 höher aus als für M1&4.

Um später die Ergebnisse der beiden empirischen Modelle besser vergleichen zu können, wird hier noch eine Methode der Modellbewertung vorgestellt. Das AIC lässt sich durch Formel (31) berechnen

$$AIC = 2k - 2l, \quad (31)$$

wobei k die Anzahl der Parameter des Modells und l die logarithmierte Likelihood-Funktion ist (siehe Formel (12)) [50].

Wenn unterschiedliche Modelle auf Basis des AIC verglichen werden sollen gilt generell: je niedriger der AIC, desto besser ist das Modell.²⁵ Allerdings müssen die zu vergleichenden Modelle mit exakt den gleichen Daten berechnet und die Parameter mit derselben Methode geschätzt worden sein (MLM). Die beiden Situationen M1&4 und M2&3 lassen sich somit nicht vergleichen.

In Tabelle 5 werden alle Kennzahlen der logReg übersichtlich dargestellt.

Tabelle 5: Ergebnisse der logReg

	M1&4	M2&3
Fehler (in %)	3,68	5,62
Standard Abweichung (in %)	0,66	0,52
Wahrheitstafeln	$\begin{bmatrix} 519 & 6 \\ 17 & 89 \end{bmatrix}$	$\begin{bmatrix} 1393 & 21 \\ 66 & 67 \end{bmatrix}$
Anzahl Parameter k	12	12
AIC	188,714	503,83
Sd von AIC	29,146	36,26
Anteil Spurwechsel (in %)	16,80	8,60

Der Fehler von M1&4 liegt bei 3,68 % und es werden im Schnitt 89 der 106 beobachteten SW richtig prognostiziert (84 %). Durch die Anwendung des BTV konnte auch die Standardabweichung zu 0,66 % bestimmt werden. Der MW des AIC liegt für diese M1&4 bei 188,714 und variiert mit einer Sd von 29,146.

Der Fehler für M2&3 liegt mit 5,62 % etwas höher und von den (im Mittel) 133 vorliegenden SW konnten 67 richtig prognostiziert werden (50,04 %). Der MW des AIC liegt bei 503,83 und besitzt eine Sd von 36,26. Dieser Wert ist größer, als der AIC von M1&4, doch da unterschiedliche Datensätze verwendet werden, kann dies nicht interpretiert werden.

Tabelle 5 weißt einige Unterschiede zu den vorherigen Ergebnis-Tabellen 2 bis 4 auf. Zum einen ist die Summe der Wahrheitstafeln geringer und zum anderen weicht der Referenzwert (Anteil der SW) von den vorherigen Referenzwerten ab. Der Grund dafür sind die Aufteilung der Daten in Trainings- und Testdaten, sowie die Anwendung des BTV. Dies ist auch der Grund, weshalb es sich bei allen Werten aus Tabelle 5 um Mittelwerte handelt.

²⁵ Modelle mit vielen Parametern, werden „bestraft“, wenn Fehler nicht kleiner ist.

4.5 Neuronale Netze

In diesem Abschnitt wird die Implementierung der Neuronalen Netze vorgestellt. Erneut wird zunächst das Modell formuliert. Dann erfolgt die Parameter-Schätzung und Modellauswahl und schließlich werden die Prognosen erstellt. Gerade der zweite Teil fällt dabei deutlich umfangreicher aus, als dies bei der logReg der Fall war, da auch schon hier das BTV und das AIC verwendet werden, um die beste Modell-Auswahl treffen zu können.

4.5.1 Modell formulieren

Um die Güte der Prognosen vergleichen zu können, werden bei den NN dieselben elf IV gewählt wie auch bei der logReg. Des Weiteren wird als Fehlerterm die Kreuzentropie gewählt, die oft bei Klassifikationsproblemen gewählt wird [24]. Wie aus Formel (19) hervorgeht, entspricht sie bis auf das Vorzeichen der logarithmierten Likelihood-Funktion aus Formel (12). Als Aktivierungsfunktion wird eine Sigmoide-Funktion gewählt, woraus folgt, dass die Rechenoperationen innerhalb der Hidden Layer identisch mit denen der logReg sind. Dies hat zur Folge, dass die Parameter auch durch die MLM berechnet werden können [24].²⁶ Dies ermöglicht es die beiden empirischen Modelle später an Hand des AIC zu vergleichen.

Das Lernen der NN wird durch dem in Abschnitt 2.3.3.3 beschriebenen Backpropagation-Algorithmus bewerkstelligt. Alle Berechnungen erfolgen mit Hilfe des Packetes *neuralnet* in R, wie in Anhang A7 zu sehen ist [51].

Vorweg stellt sich die Frage: Wenn die Parameter doch mit der MLM geschätzt werden und diese wie bereits erläutert, stets das Maximum bzw. das Minimum des Fehlerterms der Formel (19) findet, wieso dann die Modellauswahl bei den NN so viel schwieriger ist als bei der log-Reg?

Die Antwort ist, dass es richtig ist, dass die MLM in der Lage ist das globale Minimum des Fehlerterms zu finden, dieses aber nicht gesucht wird. Der Grund dafür wird in Abbildung 6 deutlich. Dadurch, dass die NN so flexibel sind und so komplexe Funktionen approximieren können, wird das globale Minimum höchst wahrscheinlich nicht das beste Modell sein, sondern ein stark an die Trainingsdaten überangepasstes Modell — Problem des Overfittings [24].

Die Behandlung dieses Problems wird im nächsten Abschnitt beschrieben.

²⁶ „With the softmax activation function and the cross-entropy error function, the NN model is exactly a linear logistic regression model in the hidden units, and all the parameters are estimated by maximum likelihood.” [24]

4.5.2 Parameter schätzen und Modell auswählen

Die Wahl des besten Modells ist bei den NN nicht trivial. Bekannt ist, dass die Netztopologie einen hohen Einfluss auf den Fehler hat [32]. Welche Netztopologie sich besonders eignet kann a priori nicht gesagt werden. Deshalb ist es üblich, verschiedene Modelle mit unterschiedlichen Netztopologien aufzustellen, um zu prüfen welche die besten Prognosen liefern [32]. Die Netztopologie umfasst die Anzahl der Hidden Layer, die Anzahl der Neuronen und die Anzahl der Output Neuronen.

Beim Aufstellen der Modelle wurde schnell deutlich, dass die Wahl einer Netztopologie mit mehr als einem Hidden Layer nicht eignet ist, da starkes Overfitting auftrat. Auch Autoren wie ZHANG (2017) oder HUNT (1994) verwendeten nur einen Hidden Layer für ihre Prognosen des SW [44, 52].

Daher wurden Modelle mit nur einem Hidden Layer untersucht. Wie hoch die Anzahl der Neuronen für diese Modelle mit einem Hidden Layer sein sollte, soll an Hand der Abbildung 21 für M1&4 und der Abbildung 22 für M2&3 entschieden werden.

Es wurden die Kennzahlen Trainingsfehler, Testfehler, sowie AIC für NN mit null bis sechs Neuronen berechnet. Wieder wurde dafür das BTV angewandt, weshalb in Abbildung 21 auch die Sd des Trainings- und Testfehlers in Form der Balken mit angegeben wird.

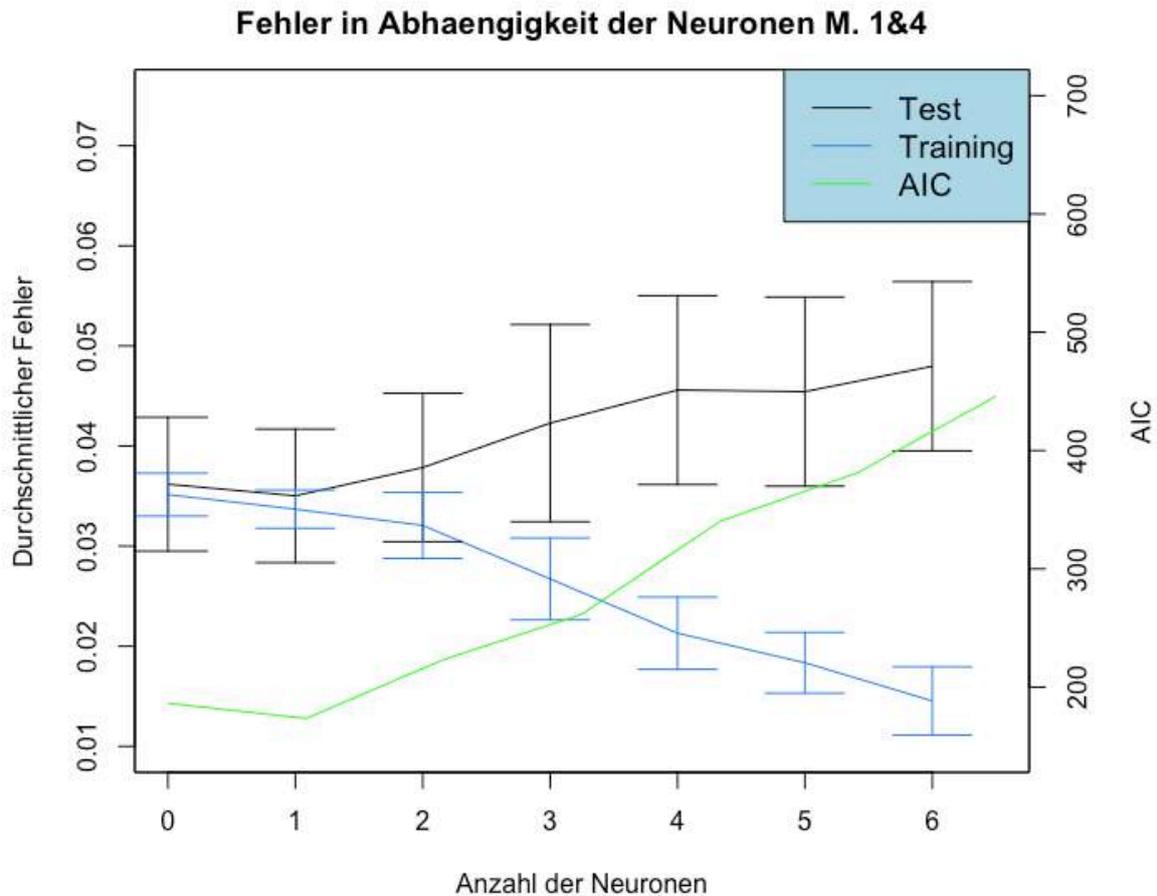


Abbildung 21: Fehler in Abhängigkeit der Anzahl der Neuronen bei M1&4

In Abbildung 21 wird der Fehler auf der linken Ordinate abgebildet und der mittlere AIC auf der rechten Ordinate. Es sind insgesamt drei Kurven abgebildet. Die schwarze Kurve beschreibt den Testfehler in Abhängigkeit der Anzahl der Neuronen und die dazugehörigen schwarzen Balken beschreiben die Sd des Fehlers. Dasselbe gilt auch für die blaue Kurve mit den blauen Balken, nur, dass diese den Trainingsfehler beschreibt. Die grüne Kurve beschreibt den AIC.

Es wird ersichtlich, dass der Testfehler und der AIC einen ähnlichen Verlauf haben. Beide nehmen zunächst ab (0 bis 1 Neuron) und dann ab einer Anzahl von zwei Neuronen zu. Beide haben ihr Minimum bei einer Anzahl von einem Neuron. Der Trainingsfehler nimmt dagegen stetig mit der Anzahl der Neuronen ab.

Zwischen null und zwei Neuronen liegen Trainings- und Testfehler noch nah bei einander, doch ab einer Anzahl von drei Neuronen beginnen sie stark zu divergieren. Dies kann als Anzeichen von Overfitting interpretiert werden.

Der durchschnittliche Testfehler bei einer Anzahl von null Neuronen liegt bei 3,61 % und der bei einem Neuron liegt bei 3,52 %. Ob dieser Unterschied der MW signifikant ist, lässt sich beim Blick auf die Größe der beiden schwarzen Balken anzweifeln. Da aber auch der AIC bei einem Neuron am kleinsten ist, wird für die später folgenden Prognosen eine Netztopologie mit einem Neuron gewählt.

Dieses Vorgehen wird ebenfalls für M2&3 verwendet. Die Ergebnisse sind in Abbildung 22 dargestellt.

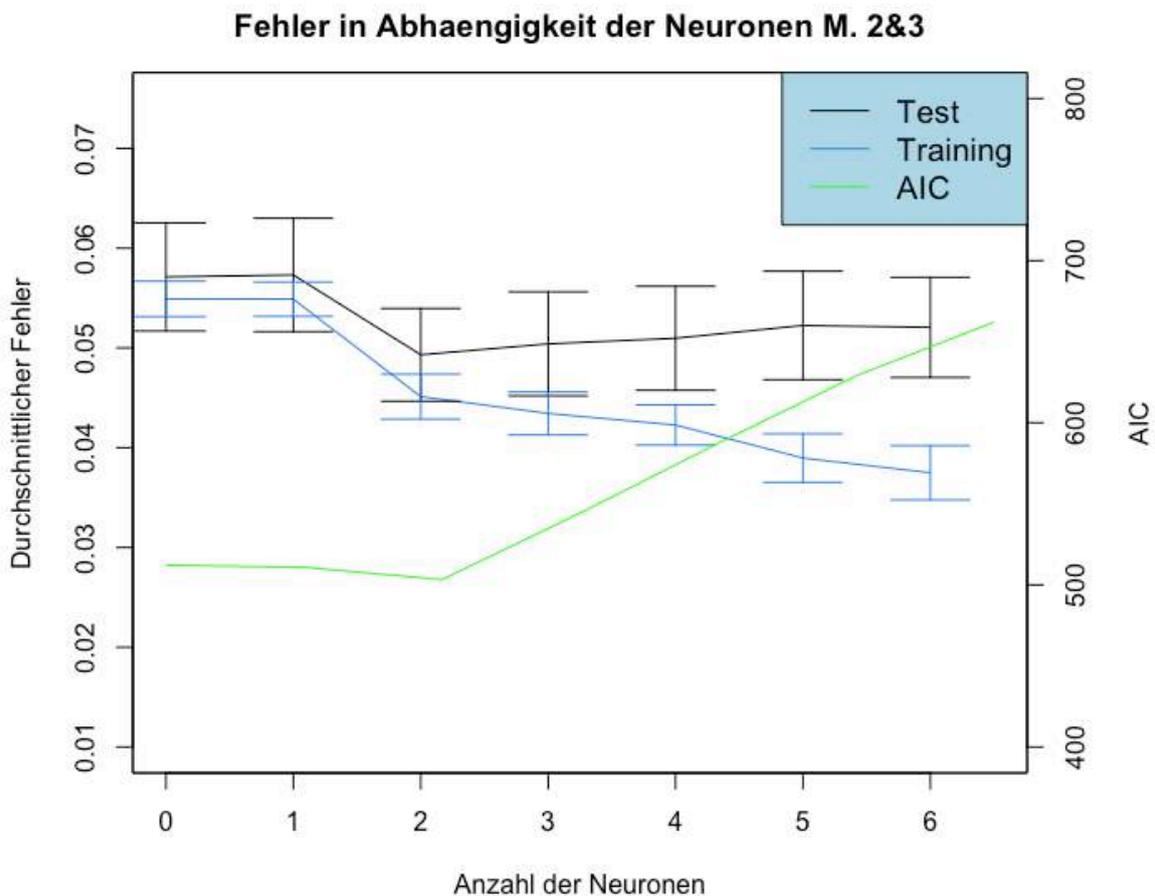


Abbildung 22: Fehler in Abhängigkeit der Anzahl der Neuronen M2&3

Auch hier stellt die blaue Kurve den durchschnittlichen Trainingsfehler in Abhängigkeit der Anzahl der Neuronen dar und die schwarze den durchschnittlichen Testfehler. Die Balken beschreiben die Sd und die grüne Kurve den AIC.

Im Gegensatz zu den Kurven aus Abbildung 21 verlaufen die Kurven in Abbildung 22 zunächst parallel zu einander, wobei der Trainingsfehler stets unterhalb des Testfehlers liegt. Erst ab

einer Anzahl von drei Neuronen steigt der Testfehler und der Trainingsfehler verringert sich und ab da verlaufen die Kurven nicht mehr parallel sondern laufen auseinander.

Auffällig ist die starke Abnahme der Fehler beim Übergang von einem zu zwei Neuronen. Der Testfehler nimmt von 5,6 % auf 4,9 % ab und steigt dann bei drei Neuronen wieder leicht auf 5,1 % an. Auch das AIC hat sein Minimum bei zwei Neuronen, doch ist der Unterschied zum AIC bei null oder einem Neuron sehr gering.

Da AIC und Fehler bei zwei Neuronen am geringsten sind, wird eine Netztopologie mit zwei Neuronen gewählt.

4.5.3 Prognose erstellen

Nachdem die Netztopologie und die weiteren Charakteristika der NN festgelegt worden sind, können nun die Kennzahlen für die Bewertung berechnet werden. Dazu wird zunächst wieder das BTV angewandt, um den Fehler und die Verteilung zu bestimmen. Das Ergebnis ist für M1&4 in Abbildung 23 dargestellt.

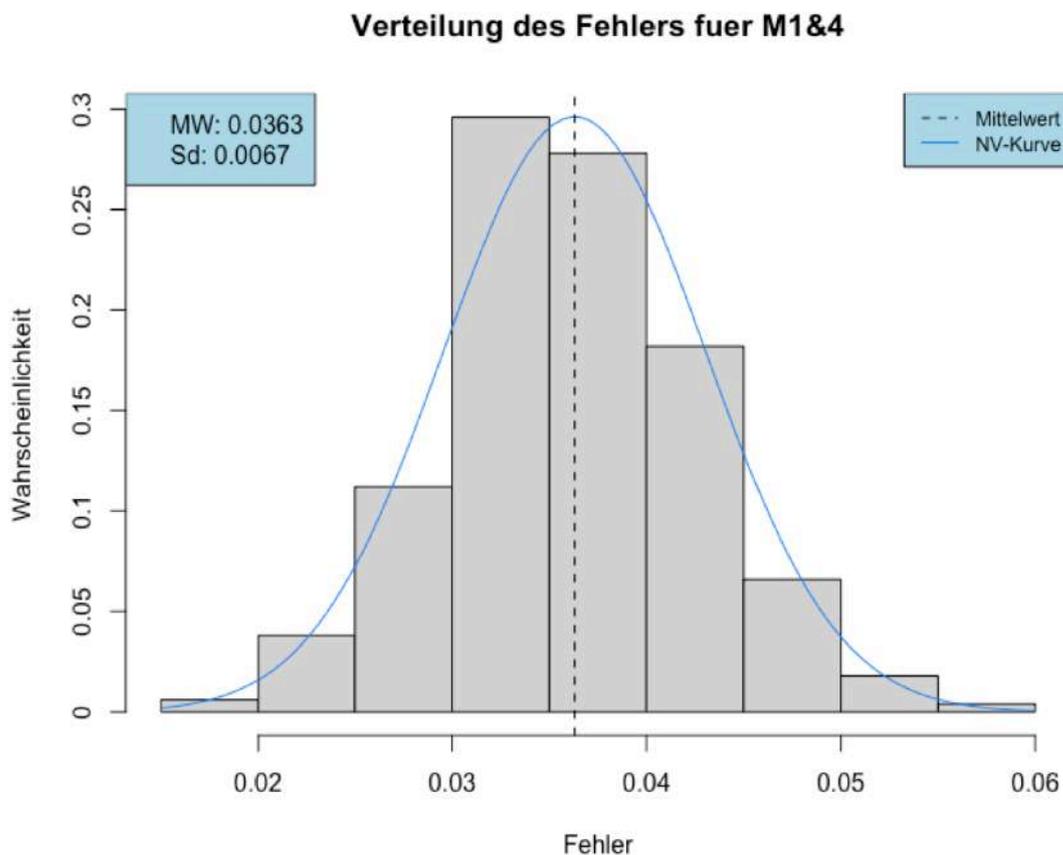


Abbildung 23: Verteilung des Fehlers der NN für M1&4

Die Art der Darstellung ist dieselbe wie im vorherigen Kapitel bei der logReg. Auf der Ordinate wird die Wahrscheinlichkeit, dass der Fehler innerhalb des Bereichs eines der neun Balken liegt, angegeben und auf der Abszisse der Fehler. Außerdem ist auch hier eine NV-Kurve in blau eingezeichnet, die mit demselben MW und Sd approximiert wurde.

Der MW des Fehlers liegt bei 3,63 % und die Sd bei 0,67 %. Diese Werte sind den berechneten Werten und der Verteilung der logReg aus Abbildung 19 ähnlich.

Bei M2&3 ist die Ähnlichkeit nicht so deutlich. Der Fehler und die Form der Verteilung unterscheiden sich von dem Fehler in Abbildung 20, wie in Abbildung 24 deutlich wird.

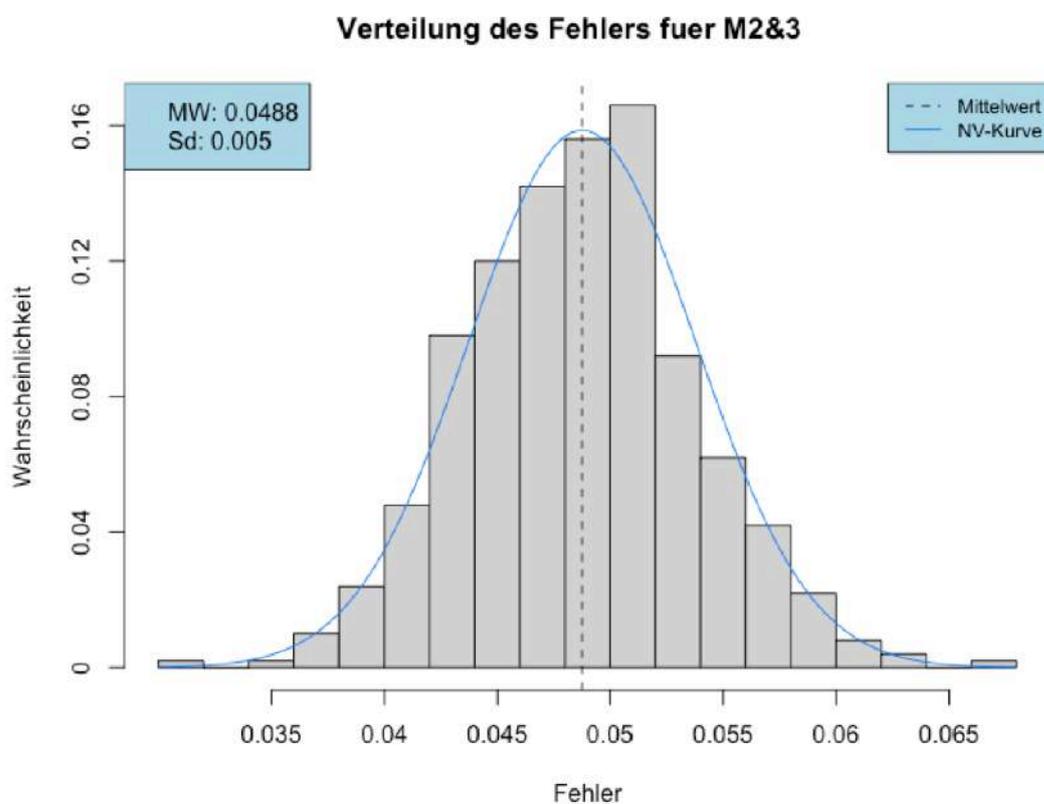


Abbildung 24: Verteilung des Fehlers der NN für M2&3

Der Fehler liegt bei 4,88 % und die Sd bei 0,5 %. Somit liegt der Fehler unter dem der logReg. Auch die Verteilungen unterschieden sich etwas, weil der Fehler bei den NN nach rechts stärker abfällt als dies in Abbildung 20 erkennbar ist.

In Tabelle 6 sind die Kennzahlen zur Bewertung der Prognosen dargestellt. Erneut handelt es sich bei allen dargestellten Werten (wegen des BTV) um MW.

Tabelle 6: Ergebnisse der NN

	M1&4	M2&3
Fehler (in %)	3,63	4,88
Standard Abweichung (in %)	0,67	0,5
Wahrheitstafeln	$\begin{bmatrix} \mathbf{518} & 8 \\ 15 & \mathbf{90} \end{bmatrix}$	$\begin{bmatrix} \mathbf{1388} & 25 \\ 51 & \mathbf{83} \end{bmatrix}$
Anzahl Parameter k	14	27
AIC	176,54	503,87
Sd von AIC	24,83	37,61
Anteil Spurwechsel (in %)	16,8	8,60

Wie bei allen bisher aufgestellten Prognosen, ist auch bei den NN der Fehler für M1&4 kleiner als bei M2&3. Für M1&4 liegt er bei 3,63 % und es werden von den 105 beobachteten SW 90 richtig prognostiziert. Für M2&3 liegt der Fehler bei 4,88 % und es werden 83 der insgesamt 134 SW richtig prognostiziert.

Auffällig in Tabelle 6 ist auch, dass sich die Anzahl der Parameter k zwischen den Situationen unterscheiden. Der Grund dafür liegt in der Wahl von unterschiedlicher Netztopologien.

5.0 Schlussteil

In diesem letzten Kapitel werden die Ergebnisse der Prognosen interpretiert und die unterschiedlichen Ansätze gegenübergestellt.

Zuerst werden dazu die klassischen LCM bewertet und verglichen und dann werden die empirischen Modelle interpretiert und es wird argumentiert, welches Modell bessere Prognosen liefert. Anschließend wird dann die Eignung der Modelle für den Einsatz in Fahrerassistenzsystemen für autonome Fahrzeuge diskutiert. Im letzten Abschnitt erfolgt das Fazit.

Alle Ergebnisse, die mit Hilfe der unterschiedlichen Ansätze berechnet worden sind, sind nun in der folgenden Tabelle 7 dargestellt. Diese Tabelle 7 soll als Interpretationsgrundlage der nächsten Abschnitte dienen.

Tabelle 7: Übersicht über alle Ergebnisse

	Klassische Spurwechselmodelle				Empirisch Modelle →				
	ZAM		MOBIL		HighD-Modell	LogReg		NN	
	M1&4	M2&3	M1&4	M2&3	M1&4	M1&4	M2&3	M1&4	M2&3
Anzahl der Parameter	0	0	4	4	2	12	12	14	27
Fehler	21,64 %	14,84 %	7,27 %	8,2 %	7,9 %	3,68 %	5,62 %	3,63 %	4,88 %
Differenz zum besten Modell	496%	204%	100%	68%	117%	1,4%	15%	0%	0%
Wahrheitstafel	$\begin{bmatrix} 2073 & 204 \\ 388 & 71 \end{bmatrix}$	$\begin{bmatrix} 5700 & 425 \\ 570 & 8 \end{bmatrix}$	$\begin{bmatrix} 2207 & 70 \\ 129 & 330 \end{bmatrix}$	$\begin{bmatrix} 6103 & 22 \\ 528 & 50 \end{bmatrix}$	$\begin{bmatrix} 2170 & 107 \\ 109 & 350 \end{bmatrix}$	$\begin{bmatrix} 519 & 6 \\ 17 & 89 \end{bmatrix}$	$\begin{bmatrix} 1393 & 21 \\ 66 & 67 \end{bmatrix}$	$\begin{bmatrix} 518 & 8 \\ 15 & 90 \end{bmatrix}$	$\begin{bmatrix} 1388 & 25 \\ 51 & 83 \end{bmatrix}$
Falsch-Negativ Rate	84,5 %	98,6 %	28,1 %	91,3 %	23,7 %	16 %	49,6 %	14,3 %	38,1 %
Anteil an Spurwechseln	16,78 %	8,62 %	16,78 %	8,62 %	16,78 %	16,8 %	8,6 %	16,8 %	8,6 %

Bis auf das highD-Modell werden bei allen Modellen beide Situationen M1&4 und M2&3 in Tabelle 7 abgebildet. Außerdem ist die Anzahl der Parameter, der Fehler, die prozentuale Differenz zu dem besten Modell, die Wahrheitstafel, die falsch-negativ Rate, sowie der Anteil der SW aufgelistet. Die falsch negativ Rate wird im Abschnitt 5.3 interpretiert. Sie ergibt sich aus der Anzahl der falsch negativen Prognosen dividiert durch die Summe der falsch negativen (FN) und den richtig positiven (RP) Prognosen (siehe Abbildung 15).²⁷

Der Pfeil der unterhalb von den Bezeichnungen der zwei Ansätze dargestellt ist, soll den Übergang von den klassischen LCM zu den empirischen Modellen verdeutlichen. Bei dem ZAM werden keine Daten benötigt um das Modell aufzustellen. Bei MOBIL ist dies auch nicht nötig, jedoch wurden der Höflichkeitsfaktor und die Wechselschwelle empirisch ermittelt. Das highD-Modell ist nicht eindeutig zuzuordnen, weshalb es zwischen den beiden Ansätzen platziert ist. Es handelt sich um ein klassisches regel-basiertes LCM, doch waren Daten nötig, um diese Regeln zu finden.

Die logReg ist rein daten-basiert, weshalb sie klar den empirischen Modellen zuzuordnen ist. Allerdings lassen sich hier die Parameter interpretieren, was bei den NN nicht möglich ist. Daher sind die NN ganz rechts in der Tabelle zu finden.

Bei dem Vergleich der Ansätze anhand dieser Tabelle 7 muss berücksichtigt werden, dass unterschiedliche Stichproben für die Berechnung der Fehler zu Grunde liegen. Während bei den klassischen LCM alle Beobachtungen des highD-Datensatzes verwendet wird, wird bei den empirischen Modellen der Datensatz in Trainings- und Testdaten aufgeteilt. Dadurch ergeben sich Unterschiede in den Wahrheitstafeln und den Anteil an SW.

5.1 Bewertung der klassischen Spurwechselmodelle

Wie bereits beschrieben, wurden die klassischen LCM hauptsächlich für die Erlangung von Verständnis des Phänomens und für die Anwendung in Verkehrssimulationen erstellt [2]. Ihr primäres Ziel ist demnach gar nicht die Prognose des SW. Dennoch hat eine Verkehrssimulation stets den Anspruch ein realistisches Fahrverhalten zu simulieren, weshalb das Spurwechselverhalten in den Simulationen, welches mit den LCM implementiert wird, nicht so fern von dem Spurwechselverhalten in dem highD-Datensatz sein sollte.

Das erste klassischen LCM, das aufgestellt und berechnet wurde, war das ZAM. Dieses Modell von NAGEL und SCHRECKENBERG (1995) bestand aus drei Regeln, welche an die Entschei-

²⁷ $FNR = \frac{FN}{FN+RP}$

dungsfindung des SWF angelehnt waren [17]. Mit Hilfe dieser einfachen Regeln und der Variation dieser Regeln, erhofften die Autoren das Verständnis des gesamten Verkehrs verbessern zu können. Durch Einstellungen des Spurwechselverhaltens einzelner Fahrer (Mikroebene) sollen Erkenntnisse über die Auswirkungen auf den Verkehr (Makroebene) gewonnen werden.²⁸

Die Anwendung dieser einfachen Regeln zur Prognose des SW fällt jedoch sehr ernüchtert aus. Bei M1&4 liegt der Fehler bei 21,6 % und nur 71 SW der 459 beobachteten SW wurden richtig prognostiziert. Bei M2&3 ist der mittlere Fehler etwas niedriger, jedoch nur auf Grund des geringen Anteils von SW in der Grundgesamtheit. Grade mal 8 der 578 SW wurden richtig prognostiziert (siehe Tabelle 7).

MOBIL ist zwar auch ein klassisches LCM, doch ist es kein regel-basiertes, sondern ein anreiz-basiertes LCM. Es wird durch zwei Kriterien definiert, dem Sicherheitskriterium, das erfüllt ist, wenn ein SW sicher möglich ist und dem Anreiz-Kriterium, das erfüllt ist, wenn der SW reizvoll ist. Sind beide Kriterien erfüllt, prognostiziert das Modell einen SW. Wie auch das ZAM, wurde MOBIL entwickelt, um in Kombination mit anderen Modellen wie dem IDM realistische Verkehrssimulationen zu ermöglichen [53].

Wird zur Prognose des SW MOBIL eingesetzt, liegt dem die Annahme zu Grunde, dass sich Fahrer tatsächlich so Verhalten wie es bei MOBIL vorausgesetzt wird. Die Spur wird genau dann gewechselt, wenn auch MOBIL einen SW in der Simulation vorschlagen würde.

Auf Basis der Ergebnisse aus Kapitel 4 kann gefolgert werden, dass diese Annahme für M1&4 nicht so abwegig ist. Der Fehler für diese Situation liegt bei 7,27 %. Die Wechselschwelle liegt bei $2,3 \frac{m}{s^2}$ und der Höflichkeitsfaktor bei 0,48. Tatsächlich kann durch Anwendung von MOBIL ein Großteil der SW (330 von 459) von der rechten auf die linke Spur richtig prognostiziert werden.

Anderes fallen die Prognosen bei M2&3 aus. Dort konnten von den 578 vorliegenden SW nur 50 richtig prognostiziert werden. Auch die Parameter Wechselschwelle und Höflichkeitsfaktor nehmen Werte an, die nicht realitätsnah sind.

Die Wechselschwelle liegt mit $14,1 \frac{m}{s^2}$ deutlich zu hoch, um eine realitätsnahe Spurwechselentscheidung prognostizieren zu können. Der SW von der linken auf die rechte Spur konnte nicht durch MOBIL erklärt oder prognostiziert werden.

²⁸ „Our approach here is to search for minimal sets of rules which reproduce certain macroscopic facts.“ [17]

Das dritte Modell, das im Rahmen der klassischen LCM vorgestellt wurde, war das highD-Modell. Es handelt sich eigentlich um ein regel-basiertes Modell. Allerdings wurde der Datensatz benötigt um die Regeln zu kalibrieren, weshalb hier von einem empirischen Ursprung gesprochen werden kann. Das Modell wurde ausschließlich für die Situation M1&4 entworfen und basierte auf zwei Bedingungen: Ist der zeitliche Abstand zum vorderen Fahrzeug gering, doch gleichzeitig die Geschwindigkeitsdifferenz hoch, so wird ein SW prognostiziert. Trotz dieser maximalen Einfachheit dieses Modells konnten gute Prognosen erstellt werden. Der Fehler ist kleiner als 8 % und der Anteil der richtig prognostizierten SW liegt mit 350 von 459 sogar höher als der Anteil bei MOBIL!

Abschließend kann zu den klassischen LCMs folgendes festgehalten werden:

Das vorgestellte ZAM eignet sich nicht, um die SW aus dem highD-Datensatz zu prognostizieren. Meiner Einschätzung nach liegt dies daran, dass das Modell zu stark von der wahren Komplexität des Spurwechselverhaltens abstrahiert. Weder berücksichtigt es, ob symmetrischer- oder asymmetrischer Verkehr vorliegt, noch werden Variablen wie Beschleunigung oder Geschwindigkeit anderer Fahrzeuge mit einbezogen.

MOBIL eignet sich nur bedingt. Für die Prognose der Manöver 2&3 kann es nicht eingesetzt werden. Ich denke dies liegt daran, dass bei MOBIL die Spur gewechselt wird, wenn damit die aktuelle Beschleunigung näher an die Wunsch-Beschleunigung kommt. Doch ein SW von der linken auf die rechte Spur wird meiner Erfahrung nach aus anderen Intentionen durchgeführt. Beispielsweise ist es häufig so, dass man auf die rechte Spur wechselt, das nachfolgende Fahrzeug deutlich schneller fährt und sich sichtlich annähert. Der SW erfolgt in diesen Situationen, obwohl sich von der Wunsch-Beschleunigung entfernt wird.

Die Prognosen mit Hilfe des highD-Modells fallen erstaunlich gut aus. Das Modell besteht nur aus zwei simplen Regeln und benötigt nur zwei IV für die Prognose. Dies lässt vermuten, dass es vielleicht doch möglich ist, mit Hilfe von einfachen Heuristiken SW zu prognostizieren. Allerdings müssten für abweichende Situationen andere Regeln gefunden werden. Die Regeln des highD-Modells sind nicht generalisierbar und lassen sich nicht gut auf andere Situationen anwenden. Dass sie, obwohl nur zwei Variablen einbezogen werden, besser prognostizieren als MOBIL ist dennoch erstaunlich.

5.2 Bewertung der empirischen Modelle

Die empirischen Modelle werden im Gegensatz zu den klassischen LCM hauptsächlich angewendet, um zuverlässig den SW zu prognostizieren [2]. Die Prognosen werden dabei nicht auf Basis einer Theorie oder von heuristischen Regeln erstellt, sondern ausschließlich anhand der Daten berechnet.

Es handelt sich bei diesen Modellen nicht um LCM, sondern um statistische Modelle, die ausschließlich zur Prognose dienen. Die berechneten Parameter müssen daher auch nicht interpretiert werden und dienen primär auch nicht für die Erklärung des Phänomens. Bei dem zweiten empirischen Modell, das vorgestellt wurde, den NN, ist es nicht einmal möglich die berechneten Parameter in irgendeiner Weise zu interpretieren.

Dennoch wäre es falsch daraus zu schließen, dass die empirischen Modelle überhaupt keinen Erklärungsgehalt über das Spurwechselverhalten offenbaren.

Ein Vorteil, den die empirischen Modelle bieten ist, dass das Ergebnis probabilistisch ist. Der Output prognostiziert nicht nur SW/kein-SW, sondern liefert einen Wert für die Wahrscheinlichkeit des SW der jeweiligen Beobachtung. Dadurch kann geprüft werden bei welchen Spurwechselsituationen das Modell Schwierigkeiten hat richtige Prognosen zu erstellen. So können beispielsweise kritischen Beobachtungen gefunden werden, die nur schlecht prognostiziert werden können.

Außerdem können empirische Modelle wie z.B. die logReg erklären, welche Variablen das Spurwechselverhalten beeinflussen und sogar in welche Richtung dieser Einfluss wirkt.²⁹

So kann dadurch erkannt werden, dass in der Situation M1&4 andere Variablen für die Prognose nützlich sind, als bei M2&3. Dennoch ist es richtig, dass die Interpretation und das Verständnis nicht das Hauptziel der empirischen Modelle sind. Das Hauptziel ist die Prognose und dadurch die erfolgreiche Anwendung in Fahrerassistenzsystemen [9, 44, 54]. Gerade dafür scheint die probabilistische Herangehensweise vielversprechend zu sein [55].

Das erste empirische Modell, das verwendet wurde, ist die logReg. Dabei handelt es sich um ein generalisiertes lineares Modell, welches die MLM nutzt, um die Parameter so zu schätzen, dass die Wahrscheinlichkeit der Übereinstimmung von geschätztem Wert und wahren Wert maximal wird.

²⁹ Richtung des Einflusses sagt aus, ob die Wahrscheinlichkeit, mit der Zunahme der Variable zu- oder abnimmt.

Wie auch bei ZHANG ET. AL (2017) erzielt die logReg in dieser Ausarbeitung sehr gute Ergebnisse. ZHANG ET. AL (2017) verwendeten ebenfalls die logReg, um Prognosen über SW zu erstellen und kamen zu dem Entschluss: „a lane changing maneuver is indeed predictable.“[44] Bei M1&4 liegt der MW des Fehlers bei 3,68 % und für M2&3 liegt er bei 5,62 %. Beim Blick auf die Wahrheitstafeln in Tabelle 7 fällt jedoch auf, dass der Fehler nicht gleich verteilt ist. Bei M2&3 konnten von den kein-SW Manöver (Manöver 2) 1393 der insgesamt 1414 vorliegenden Beobachtungen richtig prognostiziert werden (98,5 %!). Von dem SW Manöver (Manöver 3) lagen dagegen nur 133 Beobachtungen vor, von denen grade mal 50,4 % richtig prognostiziert worden sind. Die Diskrepanz zwischen den 98,5 % richtigen Prognosen bei Manöver 2 und den 50,4 % richtigen bei Manöver 3 kann nicht vernachlässigt werden. Für M1&4 konnte das SW Manöver (Manöver 4) etwas besser prognostiziert werden, doch auch hier ist ein klarer Unterschied vorhanden. Das Manöver 1 wurde in 98,8 % der Beobachtungen richtig prognostiziert und das Manöver 4 in 84 % der Fälle. Wieso dieses Ergebnis zunächst mal ernüchtert ist und das Zitat von ZHANG ET. AL (2017) etwas revidiert werden muss, wird im nächsten Abschnitt 5.3 erläutert.

Das zweite empirische Modell wurden dann die NN genutzt um den SW zu prognostizieren. Bei dem NN werden die Parameter durch Anwendung des Backpropagation-Algorithmus angepasst. Auf Grund der Hidden Layer ist es ihnen möglich nichtlineare Funktionen zu approximieren [25].

Die Prognoseergebnisse der NN sind nochmal besser als die der logReg. Bei M1&4 ist der Fehler um 1,4 % geringer als bei der logReg und bei M2&3 ist der Fehler um 15 % geringer (siehe Tabelle 7). Da auf Grund der Anwendung des BTV auch die Verteilung des Fehlers bekannt ist, kann hier geprüft werden ob die Unterschiede auch wirklich signifikant sind. Wie im vorherigen Kapitel gezeigt worden ist sind beide Verteilungen annähert NV weshalb sich ein t-Test bei unbekanntem Varianzen beider Grundgesamtheiten anbieten würde [42].³⁰ Bei diesem Test wird die Hypothese aufgestellt, dass die Mittelwerte der Fehler identisch sind. Für M1&4 ergibt sich beim Vergleich der beiden Fehler ein p-Wert von 0,182, was bedeutet, dass die Nullhypothese bei einem Signifikanzniveau von 95 % nicht verworfen werden kann. Es liegt kein signifikanter Unterschied vor. Bei M2&3 liegt ein p-Wert von 0,000 vor, was bedeutet, dass die Nullhypothese zum kleinstmöglichen Signifikanzniveau verworfen wird und somit

³⁰
$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

ein signifikanter Unterschied angenommen wird. Die NN prognostizieren die Situation M2&3 besser als die logReg.

Dennoch weisen auch die NN eine ungleiche Verteilung des Fehlers auf. Bei M1&4 konnte das kein-SW Manöver 1 zu 98,5 % der Fälle richtig prognostiziert werden, doch bei Manöver 4 nur in 85,7 % der Fälle. Noch deutlicher ist der Unterschied bei M2&3. Manöver 2 wird zu 98,2 % und Manöver 3 zu 62 % richtig prognostiziert.

Die Frage, welche der beiden empirischen Modellen besser für die vorliegende Fragestellung geeignet ist, lässt sich anhand der Ergebnisse nicht eindeutig entscheiden. Auch die Berücksichtigung des AIC bringt keine endgültige Aussage. Für M1&4 liegt der AIC bei der logReg bei 188,714 und bei dem NN bei 176,54. Damit ergibt sich nach Durchführung des t-Tests ein hoch signifikanter Unterschied des AIC bei M1&4. Folgt man den Ergebnissen des AIC sollten die NN für M1&4 gewählt werden. Anders sieht es bei M2&3 aus, denn hier liegen der AIC mit 503,83 und 503,87 extrem nah beieinander und obwohl die Fehler signifikant unterschiedlich sind, sind die AIC nahe zu identisch. Dies liegt an der Netztopologie der NN, die bei M2&3 mehr Parameter aufweist, welche den AIC erhöhen.

Weder bei M1&4, noch bei M2&3 weisen die NN *gleichzeitig* einen signifikant kleineren Fehler und auch einen signifikant kleineren AIC auf.

Wird nun berücksichtigt mit wie viel mehr Aufwand die NN verbunden sind, ist es ernüchtert, wie gering die Unterschiede in den Ergebnisse sind. Die logReg ist einfacher anzuwenden, als die NN, was sicher auch ein Grund ist, weshalb die linearen Modelle in der Praxis deutlich häufiger verwendet werden, wie in Abbildung 25 zu sehen ist.

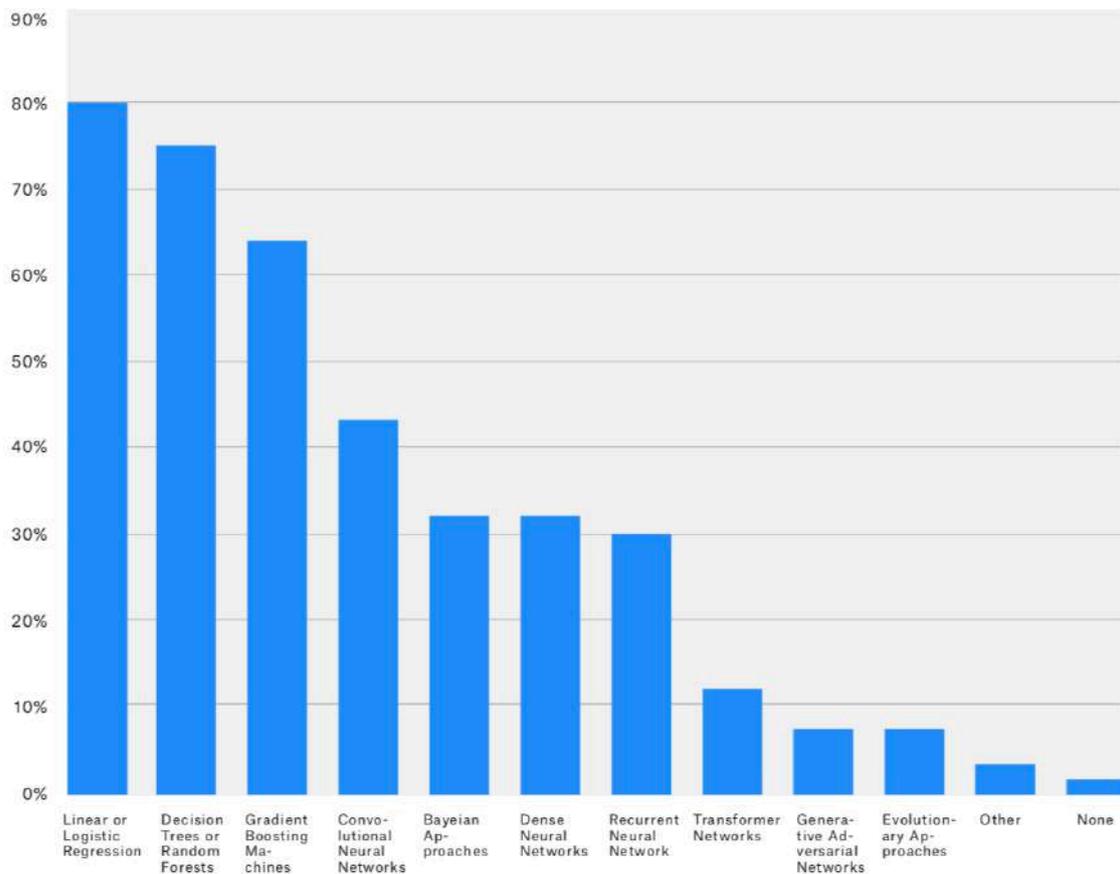


Abbildung 25: Am häufigsten verwendete Methoden des maschinellen Lernens

In Abbildung 25 werden die Ergebnisse des „*Kaggle’s State of Data Science and Machine Learning 2019*“-Reports dargestellt [56]. Dieser Report setzt sich aus den Antworten von 19717 Mitglieder der online-Community von Kaggle zusammen. Auf die Frage welche Methoden die Befragten verwenden, ergaben sich die dargestellten Ergebnisse.

Die Darstellung zeigt, dass die einfachen linearen Modelle von 80 % der Befragten regelmäßig verwendet werden, wogegen die „Dense Neural Networks“, die auch in dem NN verwendet werden, nur von 32 % der Befragten verwendet wurden.³¹

Ich bin der Meinung, dass die Frage, welches der beiden Ansätze besser geeignet ist, um SW zu prognostizieren nicht endgültig anhand der Analyse mit dem highD-Datensatz beantwortet werden kann. Der Datensatz verfügt nur über relativ wenig SW-Beobachtungen (siehe Abbildung 10), was es für die NN schwierig macht ihr Potential voll auszuschöpfen, nämlich das Lernen komplexer Muster.

³¹ „Respondents are big fans of keeping it simple.“ [56]

Ist die Form, in der die verfügbaren Input-Variablen die OV beeinflussen komplex und lassen sich nichtlineare Muster erkennen, so liefern die NN wohl verlässlichere Prognosen. Sind die Zusammenhänge dagegen wenig komplex, dann ist ein einfaches Modell die bessere Wahl. Fest steht, dass das Spurwechselerhalten von Fahrzeugen sehr komplex ist, was auch der Grund ist, dass so viele unterschiedliche Ansätze existieren, um dieses abzubilden oder zu prognostizieren [2]. Doch denke ich nicht, dass die Komplexität durch den Einfluss der beobachtbaren Variablen zustande kommt, sondern grade durch die Faktoren, die sich nicht direkt beobachten lassen. Im nächsten Abschnitt werden die Modelle in Bezug auf das autonome Fahren gegenübergestellt. Dort ergeben sich klare Vorteile der logReg.

5.3 Bezug zum autonomen Fahren

Während der gesamten Ausarbeitung wurden bisher Annahmen über autonomes Fahren gemacht, die nicht weiter begründet worden sind. Dies war zum Beispiel die Annahmen, dass ein AF mit Hilfe seiner Sensoren und Kameras alle Variablen der 5 Fahrzeuge aus Abbildung 8 zuverlässig aufnehmen kann. Für ein AF, wie das Fahrzeug a in Abbildung 8, wird es ohne Probleme möglich sein, Geschwindigkeit, Beschleunigung usw. des vorfahrenden Fahrzeug nf und des anliegenden Fahrzeug o, zu messen. Bei dem SWF, welches einige Meter weiter vorne auf der anderen Spur fährt, könnte dies schon problematisch werden. Die Annahme, dass das AF alle Variablen des Fahrzeuges of genau messen kann, darf bezweifelt werden.

Des Weiteren wurde bei der Bewertung der Modelle der Fokus hauptsächlich auf den Gesamtfehler gelegt, ohne dabei zu berücksichtigen, dass es Unterschiede innerhalb der Fehler gibt, die für ein AF von Bedeutung sind. Es existieren zwei mögliche Arten von Fehlern bei den Prognosen wie in Abbildung 15 dargestellt wird. Zum einen die falsch positiven Prognosen, bei denen SW prognostiziert wird, aber kein-SW vorliegt und zum anderen die falsch negativen Prognosen, bei denen kein-SW prognostiziert wird, aber ein SW beobachtet wird.

Diese Unterscheidung der Fehler ist für das autonome Fahren wichtig, da die Fehler unterschiedliche Auswirkungen haben können. Wird ein SW prognostiziert, der aber nicht eintritt (falsch positiv), so hat das AF seine Geschwindigkeit etwas reduziert, um eine größere Lücke zum nächsten Fahrzeug entstehen zu lassen, ohne dass dies nötig gewesen wäre.

Wird dagegen prognostiziert, dass kein-SW stattfindet, aber dennoch ein SW stattfindet (falsch negativ), so kann dadurch eine gefährliche Situation entstehen, durch die es zu abrupten Abbremsen kommen kann. Daher schätzten ZHANG ET. AL (2017) die Kosten einer falschen negativen Prognose in diesem Zusammenhang höher ein als die einer falsch positiven Prognose [44].³²

In der vorletzten Zeile von Tabelle 7 wurden die falsch negativ Rate eingetragen, die angibt, wie viele der tatsächlich vorliegenden SW falsch prognostiziert wurden. Liegt dieser Wert bei einem Modell bspw. bei 70 %, so wurden 70 % der tatsächlich beobachteten SW als kein-SW erwartet. Somit kam es bei 70 % der SW zu potentiell gefährlichen Situationen. Vor diesem

³² „the cost of false negative is much higher than the cost of false positive because the failure in predicting lane changing event may cause traffic accident.“ [44]

Hintergrund werden erneut zwei wichtigste Erkenntnisse über die Modelle deutlich, die im Fazit genauer erläutert werden: M2&3 lassen sich schlecht prognostizieren und die empirischen Modelle liefern bessere Prognosen als die Klassischen.

Welches der beiden empirischen Modelle besser für Prognosen des SW geeignet ist, konnte anhand der ausführlichen statistischen Untersuchung und Diskussion im letzten Abschnitt, nicht endgültig geklärt werden. Im Bezug zum autonomen Fahren kommen jedoch nochmal zwei Herausforderungen hinzu, die dringend berücksichtigt werden müssen und die die Tendenz weiter in Richtung der logReg verschieben. Dies ist zum einen die *Berechnung in Echtzeit* und zum andern die *Interpretierbarkeit* der Prognosen [6].

Die Rechenzeit zum Erstellen der Prognosen darf in der Praxis nicht zu hoch sein. Je schneller der SW prognostiziert werden kann, desto komfortabler und sicherer kann darauf reagiert werden.³³ Die logReg ist hier gegenüber den NN klar im Vorteil, da die Rechenzeit deutlich geringer ist.³⁴

So verhält es sich auch mit dem zweiten Punkt. Interpretierbarkeit meint in diesem Zusammenhang, dass die Entscheidung (SW/kein-SW) nachvollzogen werden kann. Sollte es auf Grund einer falschen Prognose zu einem Unfall kommen, ist es wichtig, dass im Nachhinein verstanden werden kann, wieso es zur falschen Prognose kam. Bei den NN ist es nicht möglich, die Entscheidungsfindung nachzuvollziehen, was durchaus ein Problem darstellt (auch rechtlich). Bei der logReg kann herausgefunden werden, welche Parameter dazu geführt haben, dass in jener Situation eine falsch prognostiziert wurde.

³³ „When the prediction is performed adequately early, reactive response such as braking or lane changing can be initiated smoothly and comfortably.“[57]

³⁴ Grade gegenüber komplexeren NN mit mehreren Hidden Layern (deep NN).

5.4 Fazit

In dieser Ausarbeitung wurden verschiedene Ansätze zur Prognose des Spurwechsels vorgestellt. Zum einen der Ansatz über die klassischen Spurwechselmodelle (LCM) und zum anderen über die empirischen Modelle.

Wenn die Prognose-Genauigkeit der klassischen Spurwechselmodelle angeschaut wird,³⁵ erhält man einen recht optimistischen Eindruck. MOBIL hat bei M1&4 eine Prognose-Genauigkeit von fast 93 % und bei M2&3 von fast 92 %. Schaut man sich dann aber die Differenz zum besten Modell (Zeile 3, Tabelle 7), die falsch negativ Rate (Zeile 5, Tabelle 7) oder den Anteil an SW an (Zeile 6, Tabelle 7), so wird diese Genauigkeit stark relativiert. Für M1&4 ist der Fehler der klassischen LCM im Vergleich zu den empirischen Modellen mindestens doppelt so hoch und die Situation M2&3 lässt sich, wie an der falsch negativ Rate gut zu erkennen ist, im Grunde überhaupt nicht mit den klassischen Modellen prognostizieren!

Nach BOTSCH (2020) könnte es zwei Gründe geben, weshalb die klassischen LCM schlechte Ergebnisse erzielen [6]. Es könnte sein, dass es heuristische Regeln gibt, die das Phänomen gut erklären könnten, doch diese anhand der vorliegenden Daten, nicht ersichtlich werden. Dieser Fall tritt ein, wenn die „wahren“ Gründe für einen SW nicht-beobachtbar oder einfach nicht im Datensatz vorhanden sind. Es existieren einige Faktoren, die mit Sicherheit einen Einfluss auf das Spurwechselverhalten haben, aber nicht im highD-Datensatz vorhanden sind, wie zum Beispiel die *Umwelteinflüsse*. Die Wetterbedingungen, die Geometrie der Straße oder die Tages- und Jahreszeit sind Beispiele für Umwelteinflüsse, die definitiv einen Einfluss auf das Spurwechselverhalten haben, aber nicht in den klassischen LCM berücksichtigt werden [2].

Der zweite Grund, der nach BOTSCH (2020) vorliegen könnte, ist der, dass alle wichtigen Faktoren beobachtet werden können, das Phänomen jedoch einfach zu komplex ist, um modelliert zu werden. Ich denke, dass auch dieser Grund auf die Problemstellung dieser Ausarbeitung zutrifft, denn im Endeffekt trifft ein menschlicher Fahrer die Entscheidung, ob er die Spur wechselt wird. Das menschliche Verhalten ist sehr komplex, was die Modellierung des Phänomens kompliziert gestaltet [58]. Es gibt viele Wissenschaftsdisziplinen, die sich mit der Modellierung menschlichen Verhaltens beschäftigen, wie z.B. die Sozialwissenschaften, die Wirtschaftswissenschaften oder die Verkehrsdynamik. Bei allen Disziplinen werden Annahmen getroffen, die nicht ganz mit den empirischen Beobachtungen des menschlichen Verhaltens übereinstimmen [11]. Auch bei den LCM wird angenommen, dass sich die Fahrer immer gleich verhalten und in den selben Situationen *immer dieselbe Entscheidung* treffen. Außerdem

³⁵ Prognose-Genauigkeit ergibt sich aus der Differenz des Fehlers zu 100 %.

wird angenommen, dass die Fahrer sämtliche Informationen, die in dem Modell genutzt werden, auch kennen und rational verarbeiten können. Diese Annahmen weichen von der Realität ab, in der Fahrer von ihrem Gemütszustand beeinflusst werden (aggressiv oder ruhig) oder abgelenkt sein können durch Telefonate, Musik oder Beifahrer. In der Realität treffen Fahrer inkonsistente Entscheidungen, was die Modellierung stark erschwert [59].

Die empirischen Modelle haben dieses Problem, dass das menschliche Verhalten schwierig zu verstehen ist, nicht. Die empirischen Modelle brauchen das menschliche Verhalten nicht zu modellieren, sie müssen lediglich Muster³⁶ in dem Verhalten finden, um erfolgreiche Prognosen des SW aufstellen zu können.

Auf Grund der Tatsache, dass automatisierte Fahrzeuge, die generative Eigenschaft besitzen, dass sie mit jeder Sekunde, die sie im Verkehr verbringen, Daten sammeln, ist davon auszugehen, dass sich die Menge an verfügbaren SW-Beobachtungen in der Zukunft verändern wird [6]. Vor allem die Quantität der Daten wird enorm zunehmen. Die Güte von empirischen Modellen hängt stark mit der Qualität und Quantität der Daten zusammen und wenn beides zunimmt, werden auch die Spurwechsel Prognosen mit den empirischen Modellen noch besser werden. Dadurch wird ihre schon jetzt deutliche Überlegenheit gegenüber den klassischen LCM weiter zunehmen.

Ich denke, dass empirische Modelle, die auf Methoden des maschinellen Lernens basieren, in Zukunft vermehrt in Fahrerassistenzsystemen, die für autonomes Fahren benötigt werden, eingesetzt werden. Bezüglich der Prognose des Spurwechsels kann ich mir sogar vorstellen, dass auf die logistische Regression zurückgegriffen wird, da sie vor allem bezüglich Echtzeit Berechnung und Interpretierbarkeit, gegenüber den NN im Vorteil ist.

Doch auch wenn die empirischen Modelle unverzichtbar sind, bedeutet dies nicht zwingend, dass die klassischen LCM gar keinen Nutzen für die Prognose des SW haben. Es ist durchaus vorstellbar, dass sich eine Kombination der Modelle in den autonomen Fahrzeugen etablieren wird. BOTSCH (2020) spricht in diesem Zusammenhang von Diversität und sieht den Vorteil darin, dass die Ansätze so grundverschieden sind, dass im Falle einer Übereinstimmung das Vertrauensmaß hoch ist.

³⁶ oder passender: Regelmäßigkeiten

Anhang

A1: Literatur

- [1] TREIBER, M. & KESTING, A.: *Verkehrsdynamik und –simulation*. Springer Verlag Berlin Heidelberg, 2013.
- [2] RAHMAN, M.; CHOWDHURY, M.; XIE, Y. & HE, Y.: Review of microscopic lane-changing models and future research opportunities. *IEEE Transactions on Intelligent Transportation Systems*, VOL. 14, NO. 4, 1942-1956, 2013.
- [3] DOU, Y.; YAN, F. & FENG, D.: Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. *IEEE International conference on advanced intelligent mechatronics*. Canada, 2016.
- [4] HIGHD-DATASET. Herausgegeben durch das Institut für Kraftfahrzeuge der RWTH Aachen. URL: <https://www.highd-dataset.com/#home>. Aufgerufen: 14.12.2020
- [5] VERBAND DER AUTOMOBILINDUSTRIE (VDA): Automatisierung. Von Fahrerassistenzsystemen zum automatisierten Fahren. *VDA Magazin* 2015. URL: <https://www.vda.de/de/services/Publikationen/automatisierung.html>. Aufgerufen: 05.11.2020
- [6] BOTSCH, M. & UTSCHICK, W.: *Fahrzeugsicherheit und automatisiertes Fahren*. Carl Hanser Verlag München, 2020.
- [7] STERN, R. E.; CUI, S.; MONACHE, M. L. D.; BHADANI, R.; BUNTING, M.; CHURCHILL, M.; HAMILTON, N.; HAULCY, R.; POHLMANN, H.; WU, F.; PICCOLI, B.; SEIBOLD, B.; SPRINKLE, J. & WORK, D. B.: Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C*, 89, 205-221, 2018.
- [8] KUMAR, P.; PERROLLAZ, M.; LEFEVRE, S. & LAUGIER, C.: Learning-based approach for online lane change intention prediction. *IEEE Intelligent Vehicles Symposium*. Australia, 2013.
- [9] LEE, D.; KWON, Y. P.; MCMAINS, S. & HEDRICK, K.: Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC. *IEEE 20th International Conference on Intelligent Transportation Systems*. 2017.
- [10] KRÜGER, M.; NOVO, A. S.; NATTERMANN, T. & BERTRAM, T.: Probabilistic lane change prediction using gaussian process Neural Networks. *IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, NZ, October 27-30, 2019
- [11] TANIMOTO, J.: *Evolutionary games with sociophysics. Analysis of traffic flow and epidemic*. Springer Nature Singapore, 2018.
- [12] TORDEUX, A.: *Vorlesungsskript: Grundlagen des automatisierten Fahrens*, 2019.

-
- [13] ERLEMANN, K.: *Objektorientierte mikroskopische Verkehrsflusssimulation*. Dissertation, Ruhr-Universität Bochum, 2007.
- [14] TREIBER, M.; KESTING, A. & HELBING, D.: General lane-changing model MOBIL for car-following models. *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1999, pp. 86-94, 2007.
- [15] TREIBER, M. & HELBING, D.: *Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell*. Institut für Wirtschaft und Verkehr, TU Dresden. URL: https://www.mtreiber.de/publications/MOBIL_ASIM.pdf, Aufgerufen: 13.12.2020
- [16] GIPPS, P. G.: A Model for the structure of lane-changing decisions. *Transp. Res. Part B, Methodol.*, vol.20, no.5, pp 403-414, Oct. 1986.
- [17] RICKERT, M.; NAGEL, K.; SCHRECKENBERG, M.; LATOUR, A.: Two lane traffic simulations using cellular automata. *Physics Reports*, vol. 95, Dezember 1995.
- [18] MAERIVOET, S.; DE MOOR, B.: Cellular automata models of road traffic. *Physics Reports*, vol. 419, nr.1, pages 1-64, November 2005.
- [19] NAGEL, K. & SCHRECKENBERG, M.: A cellular automaton model for freeway traffic. *Journal de Physique I*, EDP Sciences, 2221-2229, 1992.
- [20] CHOWDHURY, D.; SANTEN, L.; SCHADSCHNEIDER, A.: Statistical Physics of vehicular traffic and some related systems. *Physics Reports*, vol. 329, Juli 2000
- [21] TORDEUX, A.; CHRAIBI, M.; SEYFRIED, A. & SCHADSCHNEIDER, A.: Prediction of pedestrian dynamics in complex architectures with artificial neural networks. *Journal of intelligent transportation systems* 24(6), 556-558, 2020.
- [22] ALPAYDIN, E.: *Maschinelles Lernen*. Oldenbourg Wissenschaftsverlag GmbH, München, 2018.
- [23] FROCHTE, J.: *Maschinelles Lernen*. 2., akt. Auflage, Carl Hanser Verlag München 2019.
- [24] HASTIE, T.; TIBSHIRANI, R. & FRIEDMAN, J.: *The elements of statistical learning: Data Mining, Inference, and Prediction*. Seconde Edition. Springer Series in Statistics. 2008.
- [25] BACKHAUS, B.; B. ERICHSON & WEIBER, R.: *Multivariate Analysemethoden. Eine anwendungsorientierte Einführung*. 14 überarb. und akt. Auflage. Springer Verlag Berlin Heidelberg, 2016.
- [26] DOBSON, A. J.: *An introduction to generalized linear models*. Chapman and Hall. First edition, 1990.
- [27] DUTANG, C.: Some explanations about the IWLS algorithm to fit generalized linear models. 2017 URL: <https://hal.archives-ouvertes.fr/hal-01577698/document>, Aufgerufen: 6.12.2020

- [28] URL: <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/glm>
Aufgerufen: 6.12.2020
- [29] SCHWAIGER, R. & STEINWENDNER, J.: *Neuronale Netze programmieren mit Python*. Rheinwerk Verlag Bonn, 2019.
- [30] ROSEBLATT, F.: The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review Vol 65*, No. 6, 2385-408, 1958.
- [31] RASHID, T.: *Neuronale Netze selbst programmieren*. Ein verständlicher Einstieg mit Python. 1. Auflage, O'Reilly Media, 2017.
- [32] ZHANG, G.; PATUWO, B. E. & HU, M. Y.: Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting 14*. 25-62, 1998
- [33] BACKHAUS, B.; B. ERICHSON UND R. WEIBER: Fortgeschrittene *Multivariate Analysemethoden. Eine anwendungsorientierte Einführung*. 3 überarb. und akt. Auflage. Springer Verlag Berlin Heidelberg, 2015.
- [34] MEYNA, A. & PAULI, B.: Taschenbuch der Zuverlässigkeitstechnik. Quantitative Bewertungsverfahren, 2. überarb. und erw. Aufl., Carl Hanser Verlag, München 2010.
- [35] QUIMBA, F. M. & BARRAL, M. A.: *Exploring Neural Network Models in Understanding Bilateral Trade in APEC: A Review of History and Concepts*. Discussion paper series No 2018-22, 2018.
- [36] SADOWSKI, P.: *Notes on Backpropagation*. Department of Computer Science. University of California Irvine. URL: <https://www.ics.uci.edu/~pjsadows/notes.pdf>.
Aufgerufen: 15.11.2020
- [37] KRAJEWSKI, R.; BOCK, J.; KLOEKER, L. & ECKSTEIN, L.: The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways of Validation of Highly Automated Driving Systems. *21st Internationale Conference on Intelligent Transportation Systems*, Maui, Hawaii, 2018.
- [38] KESSLER, C. ET AL.: SP1 D11.3 Final Report. *EuroFOT Deliverable*, vol. 3, 2012.
- [39] COLYAR, J. & HALKIAS, J.: *Next Generation SIMulation Fact Sheet*. URL: <https://www.fhwa.dot.gov/publications/research/operations/its/06135/index.cfm>.
Aufgerufen: 30.11.2020
- [40] KRAJEWSKI, R.; MOERS, T.; NERGER, D. & ECKSTEIN, L.: Data-driven maneuver modeling using generative adversarial Networks and variational autoencoders for safety validation of highly automated vehicles. *21st Internationale Conference on Intelligent Transportation Systems*, Maui, Hawaii, 2018.
- [41] STELAND, A.: *Basiswissen Statistik. Kompaktkurs für Anwender aus Wirtschaft, Informatik und Technik*. 4., überarb. Auflage Springer-Verlag Berlin 2016.
- [42] HARTUNG, J.: *Statistik*. 15. Überarb. Auflage. Oldenbourg Verlag München, 2009.

-
- [43] R CORE TEAM. R: A language and environment for statistical computing. URL: <http://www.R-project.org/>. Aufgerufen: 05.01.2021
- [44] ZHANG, Y.; LI, S. & ZHANG, M.: *Lane changing prediction modeling on highway ramps: Approches and Analysis*. 2017
- [45] TREIBER, M. & HELBING, D.: *Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell*. URL: https://mtreiber.de/publications/MOBIL_ASIM.pdf. Aufgerufen: 30.12.2020
- [46] TREIBER, M.: Longitudinal Traffic model: The IDM. URL: <https://mtreiber.de/MicroApplet/IDM.html#IDMparameters>. Aufgerufen: 2.12.2020
- [47] BOX, G. E. P. & JENKINS, G. M.: *Time Series Analysis: Forecasting and control*. Holden-Day. San Francisco.1976.
- [48] McCullagh, P. & Nelder, J. A.: *Generalized Linear Model*. Chapman and Hall Second edition. (1990)
- [49] RICHTER, S.: *Statistisches und maschinelles Lernen*. Springer Spektrum, Berlin 2019.
- [50] BURNHAM, K. & ANDERSON, D.: *Model selection and multimodel inference*. Springer-Verlag. 2 Aufl. 2002.
- [51] FRITSCH, S.; GUENTHER, F. & SULING, M.: *Neuralnet: Training of neural networks*. URL: <http://CRAN.R-project.org/package=neuralnet>. Aufgerufen: 28.10.2020
- [52] HUNT, J. G. & LYONS, G. D.: Modelling dual carriageway lane changing using Neural Networks. *Transpn. Res. Vol 2, No. 4, 231-245*, 1994.
- [53] KESTING, A. & TREIBER, M.: Online traffic state estimation based on floating car data. Technische Universität Dresden. 2010.
- [54] DOGAN, Ü.; EDELBRUNNER, J. & IOSSIFIDIS, I.: Autonomous Driving: A comparison of machine learning techniques by means of the prediction of lane change behavior. *IEEE International Conference on Robotics and Biomimetics*. Thailand, Phuket. 2011.
- [55] Mayer-Delius, D., Plagemann, C., Burgard, W.: Probabilistic situation recognition for vehicular traffic scenarios. *IEEE International Conference on Robotics and Automation*. Kobe, Japan, 2009.
- [56] Kaggle's State of Data Science and Machine Learning 2019. URL: <https://www.docdroid.com/qzyxCr4/kaggle-state-of-data-science-and-machine-learning-2019-pdf>. Aufgerufen: 29.12.2020
- [57] Gallitz, O.; De Candido, O.; Botsch, M. & Utschick, W.: Interpretable feature generation using deep neural networks and its application to lane change detection. *IEEE Intelligent Transportation Systems Conference*, Auckland, NZ, 2019

-
- [58] XING, Y. ET AL.: Driver lane change intention inference for intelligent vehicles: Framework, survey, and challenges. *IEEE Transactions on Vehicular Technology*. 2019.
- [59] KAHNEMAN, D. & TVERSKY, A.: Prospect Theory: An analysis of decision under risk. *Econometrica*. Vol. 47, No.2, 263-292. 1979.

A2: ZAM R-Skript

```
rm(list=ls())
data=read.csv("~/TORDEUX_DATA.csv")

names(data)
attach(data)

Y=maneuver>2
cc=(maneuver==1|maneuver==4)&class==1
Zellen1<-round(distPred/2)
Zellen2<-round(distPredAdj/2)
Zellen3<-round(distFolloAdj/2)
Geschw<-round(xVelocity)
b=0
for(i in (1:nrow(data))[cc]){
  if (Zellen1[i] < Geschw[i] && Zellen2[i] > Geschw[i] && Zellen3[i] > Geschw[i]) {
    b<- append(b,1)
  } else {
    b<- append(b,0)
  }
}
result<-b[-1]
mean(abs(Y[cc]-result))
## [1] 0.2163743
table(Y[cc],result)
## result
## 0 1
## FALSE 2073 204
## TRUE 388 71
```

A3: MOBIL R-Skript

```

rm(list=ls())
data=read.csv("~/TORDEUX_DATA.csv")
names(data)
attach(data)

Y=maneuver>2 #Maneuver: 1:LKR 2:LKL 3:LCR 4:LCL
ov=1
if(ov==1)cc=(maneuver==1|maneuver==4)&class==1
if(ov==2)cc=(maneuver==2|maneuver==3)&class==1
if(ov==3)cc=class==1

acc=function(s,v,v1,l,T,v0,a,b,class){
  if(class==0)v0=25
  s0=1+v*T-v*(v1-v)/(2*sqrt(a*b))
  a*(1-(v/v0)^4-(s0/(1+s))^ 2)}

MOBIL=function(i,l=2,T=1.2,v0=50,a=1.5,b=2,p,D0){
  vc=xVelocity[i];vc1=diffSpeedPred[i]+xVelocity[i];vc1p=diffSpeedPredAdj[i]
  +xVelocity[i]
  sc=distPred[i];scp=distPredAdj[i]
  ac=acc(sc,vc,vc1,l,T,v0,a,b,class[i]);acp=acc(scp,vc,vc1p,l,T,v0,a,b,class
  [i])
  vo=diffSpeedFollo[i]+xVelocity[i];vo1=xVelocity[i];vo1p=diffSpeedPred[i]+x
  Velocity[i]
  so=distFollo[i];sop=distFollo[i]+distPred[i]
  ao=acc(so,vo,vo1,l,T,v0,a,b,classFollo[i]);aop=acc(sop,vo,vo1p,l,T,v0,a,b,
  classFollo[i])
  vn=diffSpeedFolloAdj[i]+xVelocity[i];vn1=diffSpeedPredAdj[i]+xVelocity[i];
  vn1p=xVelocity[i]
  sn=distFolloAdj[i]+distPredAdj[i];snp=distFolloAdj[i]
  an=acc(sn,vn,vn1,l,T,v0,a,b,classFolloAdj[i]);anp=acc(snp,vn,vn1p,l,T,v0,a
  ,b,classFolloAdj[i])
  if(ov==3)acp-ac+p*(aop-ao+anp-an)>D0&acp>-4 else{
  if(vc<60/3.6) acp-ac+p*(aop-ao+anp-an)>D0 else{
    if(ov==1&vc>vc1p&vc1p>60/3.6) ac=min(ac,acp)
    if(ov==2&vc>vc1p&vc1p>60/3.6) acp=min(ac,acp)
    if(ov==1) acp-ac+p*(anp-an)>D0&acp>-4 else acp-ac+p*(aop-ao)>D0&acp>-4}}
}

```

```

SSR=function(para){
  s=NULL;for(i in (1:nrow(data))[cc])
    s=c(s,(MOBIL(i,,,,,para[1],para[2])-Y[i])^2)
  return=mean(s)}

down=c(0,.5);up=c(1,8)

if(T){
  Hoeflichkeitsf.=seq(down[1],up[1],length.out=20)
  Wechselschwelle=seq(down[2],up[2],length.out=20)
  OptimierungsFkt.=matrix(-1,length(Hoeflichkeitsf.),length(Wechselschwelle)
)
  for(i in 1:length(Hoeflichkeitsf.))
    for(j in 1:length(Wechselschwelle))
      OptimierungsFkt.[i,j]=SSR(c(Hoeflichkeitsf.[i],Wechselschwelle[j]))
  #plot(Hoeflichkeitsf.,OptimierungsFkt.)
  persp(Hoeflichkeitsf.,Wechselschwelle,OptimierungsFkt. ,xlab = NA,ylab="Ho
eflichkeitsf.",cex.lab=1.1,cex.main=1.4,main="Optimierung Manoever 1&4",the
ta=90,phi=-3,ticktype="detailed",col="dodgerblue2",border = "black",shade=T
,ltheta=-120,lphi=20)
  par(mfrow = c(1, 1))
}
mm=1e5;para0=-1;#set.seed(1)

```

```

for(b in 1:10){
  start=c(down[1]+(up[1]-down[1])*runif(1),down[2]+(up[2]-down[2])*runif(1))
  ;
  op=optim(par=start,fn=SSR,method="L-BFGS-B",lower=down,upper=up)
  if(op$value<mm){
    mm=op$value
    para0=op$par}}
para0=c(0.48,2.3)
Mopt=NULL
for(i in (1:nrow(data))[cc])
  Mopt=c(Mopt,MOBIL(i,,,,,para0[1],para0[2]))

para0

```

```
ccc=Y[cc]==1
```

```
mean(!Mopt==Y[cc])
## [1] 0.07273392
table(Y[cc],Mopt)
##  Mopt
##  FALSE TRUE
## FALSE 2207 70
## TRUE 129 330
```

A4: HighD-Modell R-Skript

```
rm(list=ls())
data=read.csv("~/TORDEUX_DATA.csv")
names(data)
attach(data)

Y=maneuver>2
cc=(maneuver==1|maneuver==4)&class==1
besch<-xAcceleration-accPred
para1=-2.9
para2=2.75
b=0
for(i in (1:nrow(data))[cc]){
  if (diffSpeedPred[i] < para1 && timeGapPred[i] < para2) {
    b<- append(b,1)
  } else {
    b<- append(b,0)
  }
}
result<-b[-1]
mean(abs(Y[cc]-result))
## [1] 0.07894737
table(Y[cc],result)
##  result
##  0 1
## FALSE 2170 107
## TRUE 109 350
```

A5: LogR R-Skript

```

rm(list=ls())
data=read.csv("~/TORDEUX_DATA.csv")
names(data);attach(data)
library(ROCR)
Y=maneuver>2 #Maneuver: 1:LKR 2:LKL 3:LCR 4:LCL
cc=(maneuver==1|maneuver==4)&class==1
besch<-xAcceleration- accFolloAdj
besch1<-xAcceleration- accPred
besch2<-xAcceleration- accPredAdj
X=cbind(distPred,distFollo,distPredAdj,distFolloAdj,diffSpeedPred,diffSpeed
Follo,diffSpeedPredAdj,diffSpeedFolloAdj,besch1,besch,besch2)
Y=Y[cc];X=X[cc,]

logistic=function(x)
  exp(x)/(1+exp(x))
logReg=function() {
  reg=glm(Y~X,family=binomial(logit),subset=ccr)
  print(summary(reg))
  train=logistic(cbind(1,X[ccr,])%*%reg$coeff)
  test=logistic(cbind(1,X[!ccr,])%*%reg$coeff)
  #pl(Y,X,train,test)
  par(mfrow=c(1,1))
  tab<-table(Y[!ccr],test>0.5)
  logli=sum(Y[!ccr]*log(test)+(1-Y[!ccr])*log((1-test)))
  RAIC=-2*logli + 24
  print(logli)
  print(RAIC)
  c(mean(abs(Y[ccr]-(train>.5))),mean(abs(Y[!ccr]-(test>.5))), AIC(reg),tab[
1],tab[2],tab[3],tab[4], RAIC)}

ccr=rep(F,nrow(X));ccr[sample(1:nrow(X),round(nrow(X)/1.3) )]=T
logReg()
B=10000;tr=NULL;te=NULL;AC=NULL;TB1=NULL;TB2=NULL;TB3=NULL;TB4=NULL;llh=NUL
L
for(b in 1:B){
  ccr=rep(F,nrow(X));ccr[sample(1:nrow(X),round(nrow(X)/1.3) )]=T

```

```

lg=logReg();tr[b]=lg[1];te[b]=lg[2];AC[b]=lg[3];TB1[b]=lg[4]; TB2[b]=lg[5]
;TB3[b]=lg[6];TB4[b]=lg[7];llh[b]=lg[8]}

mean(te);sd(te)

mean(llh);sd(llh)

wahrheit=matrix(c(mean(TB1),mean(TB3),mean(TB2),mean(TB4)),byrow=T, nrow=2)

wahrheit

par(mfrow=c(1,1))

par(mar=c(5,5,5,5))

hist(te, main="Verteilung des Fehlers fuer M1&4",freq = T,label = F,xlab="Fehler",ylab= "Wahrscheinlichkeit",axes=T,breaks=18)

curve(dnorm(x, mean = mean(te), sd=sd(te)),add=TRUE, col="dodgerblue2")

abline(v=mean(te), lty=2)

legend("topright",legend = c("Mittelwert","NV-Kurve"), lty=c(2,1),cex=0.8,
col=c("black","dodgerblue2"),bg = "lightblue")

legend("topleft",bg="lightblue",c(paste("MW:",round(mean(te),4)),paste("Sd:",round(sd(te),4))))

```

A6: Modelauswahl NN R-Skript

```

rm(list=ls());require(neuralnet)

data=read.csv("~/TORDEUX_DATA.csv")

names(data);attach(data)

Y=maneuver>2 #Maneuver: 1:LKR 2:LKL 3:LCR 4:LCL

cc=(maneuver==1|maneuver==4)&class==1

besch<-xAcceleration- accFolloAdj

besch1<-xAcceleration- accPred

besch2<-xAcceleration- accPredAdj

X=cbind(distPred,distFollo,distPredAdj,distFolloAdj,diffSpeedPred,diffSpeedFollo,diffSpeedPredAdj,diffSpeedFolloAdj,besch1,besch,besch2)

Y=Y[cc];X=X[cc,]

ANN=function(a){

X=as.data.frame(scale(X,center=apply(X,2,mean),scale=apply(X,2,sd)))

tr=as.data.frame(X[ccr,]);te=as.data.frame(X[!ccr,])

f=as.formula(paste("Y[ccr] ~",paste(names(tr), collapse = " + ")))

nn=neuralnet(f,data=tr,hidden= a,stepmax=1e7,linear.output=F,threshold = 0.04,likelihood = T,err.fct = "ce", act.fct = "logistic")

```

```

#nn=neuralnet(f,data=tr,hidden=(2,a,2),stepmax=1e7,linear.output=F,likelihood=T,err.fct="sse",act.fct="logistic") ##Mehrere Layer

train=as.vector(compute(nn,tr)$net.result)
test=as.vector(compute(nn,te)$net.result)
logli=sum(Y[!ccr]*log(test)+(1-Y[!ccr])*log((1-test)))
print(logli)

AIC=print((nn)$result.matrix[4])
c(mean(abs(Y[ccr]-(train>.5))),mean(abs(Y[!ccr]-(test>.5))),AIC,logli)}

ccr=rep(F,nrow(X));ccr[sample(1:nrow(X),round(nrow(X)/1.3))]=T
ANN(1)
B=500;a=c(0,1,2,3,4,5,6)

str=matrix(-1,B,length(a));ste=matrix(-1,B,length(a));AC=matrix(-1,B,length(a));Logli=matrix(-1,B,length(a))
for(b in 1:B){ccr=rep(F,nrow(X));ccr[sample(1:nrow(X),round(nrow(X)/1.3))]=T
for(i in 1:length(a)){
res_ann=ANN(a[i]);str[b,i]=res_ann[1];ste[b,i]=res_ann[2];AC[b,i]=res_ann[3];Logli[b,i]=res_ann[4]
print(AC)
print(str)
print(ste)
print(Logli)
}
}
mean(ste);sd(ste)
par(mfrow=c(1,1))
par(mar=c(5,5,3,5))
plot(NA,ylab="Durchschnittlicher Fehler",xlab="Anzahl der Neuronen",ylim=c(.010,.075),xlim=c(0,6.5),main="Fehler in Abhaengigkeit der Neuronen M. 1 & 4")
sdtr=apply(str,2,sd);sdte=apply(ste,2,sd)
lines(a,apply(ste,2,mean),col=1);lines(a,apply(str,2,mean),col="dodgerblue2")
for(i in 1:length(a)){
arrows(i-1,apply(ste,2,mean)[i]-sdte[i],i-1,apply(ste,2,mean)[i]+sdte[i],code=3,length=.25,angle=90,col=1)
arrows(i-1,apply(str,2,mean)[i]-sdtr[i],i-1,apply(str,2,mean)[i]+sdtr[i],code=3,length=.25,angle=90,col="dodgerblue2")}

```

```

par(new=T)

plot(asa,type="l", yaxt="n", xaxt="n", ylab = "",xlab="",col="green",ylim =
c(150,700))

axis(side=4)

mtext("AIC", side=4, line=3)

legend("topright",legend = c("Test","Training", "AIC"), lty = c(1,1,1),cex=
1.2, col=c(1,"dodgerblue2","green"),bg = "lightblue")#

```

A7: NN-Ergebnis R-Skript

```

rm(list=ls());require(neuralnet)
data=read.csv("~/TORDEUX_DATA.csv")
names(data);attach(data)
Y=maneuver>2 #Maneuver: 1:LKR 2:LKL 3:LCR 4:LCL
cc=(maneuver==1|maneuver==4)&class==1
besch<-xAcceleration- accFolloAdj
besch1<-xAcceleration- accPred
besch2<-xAcceleration- accPredAdj

X=cbind(distPred,distFollo,distPredAdj,distFolloAdj,diffSpeedPred,diffSpeed
Follo,diffSpeedPredAdj,diffSpeedFolloAdj,besch1,besch,besch2)
Y=Y[cc];X=X[cc,]

ANN=function(a){
  X=as.data.frame(scale(X,center=apply(X,2,mean),scale=apply(X,2,sd)))
  tr=as.data.frame(X[ccr,]);te=as.data.frame(X[!ccr,])
  f=as.formula(paste("Y[ccr] ~",paste(names(tr), collapse = " + ")))
  nn=neuralnet(f,data=tr,hidden= a,stepmax=1e7,linear.output=F, likelihood =
T,err.fct = "ce", act.fct = "logistic")
  #nn=neuralnet(f,data=tr,hidden=(2,a,2) ,stepmax=1e7,linear.output=F, likel
ihood = T,err.fct = "sse", act.fct = "logistic") #####Mehrere Layer
  train=as.vector(compute(nn,tr)$net.result)
  test=as.vector(compute(nn,te)$net.result)
  tab<-table(Y[!ccr],test>0.5)
  logli=sum(Y[!ccr]*log(test)+(1-Y[!ccr])*log((1-test)))
  print(logli)
  RAIC=-2*logli + 28
}

```

```

print(RAIC)
print(nn$result.matrix[2])
AIC=print((nn)$result.matrix[4])
c(mean(abs(Y[ccr]-(train>.5))),mean(abs(Y[!ccr]-(test>.5))),AIC,RAIC,tab[1],tab[2],tab[3],tab[4])}

ccr=rep(F,nrow(X));ccr[sample(1:nrow(X),round(nrow(X)/1.3) )]=T
ANN(1)
B=10000;a=1

str=matrix(-1,B,length(a));ste=matrix(-1,B,length(a));AC=matrix(-1,B,length(a));RAIC=matrix(-1,B,length(a));tab1=matrix(-1,B,length(a));tab2=matrix(-1,B,length(a));tab3=matrix(-1,B,length(a));tab4=matrix(-1,B,length(a)) ###Von Raphael

for(b in 1:B){ccr=rep(F,nrow(X));ccr[sample(1:nrow(X),round(nrow(X)/1.3) )]=T ###Von Raphael Train/Test

for(i in 1:length(a)){
  res_ann=ANN(a[i]);str[b,i]=res_ann[1];ste[b,i]=res_ann[2];AC[b,i]=res_ann[3];RAIC[b,i]=res_ann[4];tab1[b,i]=res_ann[5];tab2[b,i]=res_ann[6];tab3[b,i]=res_ann[7];tab4[b,i]=res_ann[8]

  print(RAIC)
  print(str)
  print(ste)
}
}

wahrheit=matrix(c(mean(tab1),mean(tab3),mean(tab2),mean(tab4)),byrow=T, nrow=2)
wahrheit

hist(ste, main="Verteilung des Fehlers fuer M1&4",freq = F,label = F,xlab="Fehler",ylab= "Wahrscheinlichkeit",axes=F,breaks=12)#,xlim=c(0.025,0.045))
#lines(density(tr),col="blue")
axis(side = 2,at = c(0,10,20,30,40,50,60), labels = c(0, 0.05,0.1,0.15,0.2,0.25,0.3))
axis(side = 1,at = c( 0.02,0.03,0.04,0.05,0.06), labels = c( 0.02,0.03,0.04,0.05,0.06))
curve(dnorm(x, mean = mean(ste), sd=sd(ste)),add=TRUE, col="dodgerblue2")
abline(v=mean(ste), lty=2)

legend("topright",legend = c("Mittelwert","NV-Kurve"), lty=c(2,1),cex=0.8, col=c("black","dodgerblue2"),bg = "lightblue")

legend("topleft",bg="lightblue",c(paste("MW:",round(mean(ste),4)),paste("Sd:",round(sd(ste))

```

