Bachelorthesis

Quantitative Untersuchung des Einflusses von Software-Fehlern auf überwachte und angeordnete Fahrzeugrückrufe mithilfe maschineller Lernmethoden

im Studiengang Fahrzeugtechnik
der Fakultät Mobilität und Technik

Tomasz Wojnicki-Wartenberg

Matrikelnummer:
Zeitraum: 01.04.2023 bis 31.08.2023
Erstprüfer: Prof. Dr.-Ing. Tobias Leopold
Zweitprüfer: Dipl.-Ing. Jonas Wolf

Firma: Vector Informatik GmbH
Betreuer: Dipl.-Ing. Jonas Wolf

Eigenständigkeitserklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Stuttgart, den 31.08.2023

Unterschrift

Vorwort

Die vorliegende Bachelorarbeit wurde von April 2023 bis August 2023 bei der Vector Informatik GmbH in Stuttgart bearbeitet. Die Ergebnisse der Arbeit konnten in Kooperation mit der Abteilung PES9.3 (Product Line Embedded Software & Systems) und der Hochschule Esslingen erzielt werden.
Es wurde das Thema „Quantitative Untersuchung des Einflusses von Software-Fehlern auf angeordnete und überwachte Fahrzeugrückrufe mittels maschineller Lernmethoden“ bearbeitet.

Die Aufgabenstellung wurde von Herrn Dipl.-Ing. Jonas Wolf aus eigenem Interesse entworfen, von mir übernommen und ausgearbeitet. Erste Eindrücke der Softwareentwicklung, der maschinellen Lernmethoden und dem Alltag im Hause Vector Informatik konnte ich bereits in meinem vorausgegangenen einmonatigen Praktikum sammeln. Nach dieser Erfahrung war ich von den Möglichkeiten des maschinellen Lernens begeistert. Da ich auch privat ein großes Interesse an innovativen und zukunftsorientierten Themen habe, war die Motivation, mich mit dem Thema zu beschäftigen, sehr groß.

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Erstellung meiner Bachelorarbeit unterstützt und somit maßgeblich zum Gelingen meiner Abschlussarbeit beigetragen haben.
Insbesondere danke ich Herrn Dipl.-Ing. Jonas Wolf für die sehr engagierte Betreuung und Unterstützung während der gesamten Abschlussarbeit und darüber hinaus. Mein Dank gilt auch Herrn Prof. Dr.-Ing. Tobias Leopold, der diese Arbeit von Seiten der Hochschule Esslingen betreut hat. Des Weiteren möchte ich mich bei allen Mitarbeitern der Abteilung PES9 für die freundliche Aufnahme in das Unternehmen und die kollegiale Zusammenarbeit bedanken.

Tomasz Wojnicki-Wartenberg
Stuttgart, den 13. August 2023

Kurzfassung

Gegenstand der hier vorgestellten Arbeit ist die quantitative Untersuchung des Einflusses von Software-Fehlern auf angeordnete und überwachte Fahrzeugrückrufe mithilfe maschineller Lernmethoden. Der Anteil von immer komplexerer Software in Kraftfahrzeugen nimmt seit Jahren stets zu. Damit steigt auch der Anteil an Software, die einen erheblichen Einfluss auf die Fahrzeugsicherheit hat – aber steigen damit auch die von Software verursachten Rückrufe proportional dazu an? Ziel der vorliegenden Arbeit ist es, genau diese Frage zu untersuchen und den genauen Einfluss zu quantifizieren. Dazu werden Daten aus internationalen Rückrufdatenbanken herangezogen und näher untersucht. Aufgrund der großen Datenmenge werden geeignete Algorithmen des maschinellen Lernens ausgewählt und mit deren Hilfe die Daten aufbereitet und analysiert.

Schlagwörter: Fahrzeugrückrufe, Software, Software-Fehler, KBA, RAPEX, NHTSA, Maschinelles Lernen, Entscheidungsbaum, Neuronale Netzwerke, Datenanalyse

Abstract

The subject of the work presented here is a quantitative investigation of the influence of software faults causing ordered and monitored vehicle recalls using machine learning methods. The proportion of increasingly complex software in motor vehicles has been growing steadily for years. This means that the proportion of software that has a significant influence on vehicle safety is also increasing – but are the recalls caused by software also increasing proportionally? The aim of this thesis is to investigate this question and to quantify the exact influence. For this purpose, data from international recall databases are used and examined in more detail. Due to the large amount of data, suitable machine learning algorithms are selected and used to process and analyse the data.

Keywords: vehicle recalls, software, software error, KBA, RAPEX, NHTSA, machine learning, clustering, decision tree, neural networks, data analysis

Inhaltsverzeichnis

Eigenständigkeitserklärung ..... **I**

Vorwort ..... **II**

Kurzfassung ..... **III**

Abstract ..... **III**

Inhaltsverzeichnis ..... **IV**

Abbildungsverzeichnis ..... **VII**

Tabellenverzeichnis ..... **IX**

Abkürzungsverzeichnis ..... **X**

1. Einleitung **1**

1.1. Motivation 1

1.2. Zielsetzung 2

1.3. Aufbau der Arbeit 3

2. Einführung in das Thema **4**

2.1. Fahrzeugrückrufe 4

2.1.1. KBA 4

2.1.2. RAPEX 7

2.1.3. NHTSA 8

2.2. Machine Learning, Deep Learning, Artificial Intelligence 10

2.2.1. Unsupervised Learning 11

2.2.2. Supervised Learning 13

2.2.3. Reinforcement Learning 14

2.2.4. Neuronale Netze 15

3. Methodik und Auswahl geeigneter Verfahren 18

3.1. Literatur- und Forschungsrecherche zur Leitfrage 18

3.2. Auswahl geeigneter Machine Learning-Verfahren 21

4. Deskriptive Datenanalyse und -aufbereitung 24

4.1. Datendeskription 24

4.1.1. Deskription RAPEX 24

4.1.2. Deskription NHTSA 25

4.2. Datenaufbereitung 27

4.2.1. Rauschentfernung/Datenbereinigung 27

4.2.2. Tokenisierung 28

4.2.3. Normalisierung 28

4.2.4. Vektorisierung – Variante für Clustering 29

4.2.5. Aufteilung der Daten 30

4.2.6. Padding 30

5. Machine Learning-Prozess und Kategorisierung 33

5.1. Clustering-Analyse 33

5.2. Klassifikationsverfahren – Decision Tree 43

5.3. Neuronale Netzwerke 47

6. Ergebnisse und Diskussion 50

6.1. Vergleich angewandter Verfahren 50

6.2. Kategorisierung RAPEX-Datenbank 52

6.3. Kategorisierung NHTSA-Datenbank 54

6.3.1. Historische Entwicklung 56

6.3.2. Auswertung der Hersteller 61

6.3.3. Differenz Rückrufjahr - Modelljahr 65

6.3.4. Diskussion der Auswertungen 68

7. Zusammenfassung und Ausblick 70

7.1. Fazit 70

7.2. Ausblick 72

Literatur 75

Abbildungsverzeichnis

Abbildung 1: Graphische Darstellung Meldeweg KBA 6
Abbildung 2: Graphische Darstellung Meldeweg RAPEX 7
Abbildung 3: Graphische Darstellung Meldeweg NHTSA 9
Abbildung 4: Schematische Abbildung Unterkategorien AI 10
Abbildung 5: "Ein Software-Agent mit Benutzerinteraktion" [15, S. 18] 11
Abbildung 6: „Zeichnungen verschiedener Pflanzen“ [16, S. 25] 12
Abbildung 7: Aufbau eines formalen Neurons [15, S. 269]..... 15
Abbildung 8: Neuron eines FFNN..... 16
Abbildung 9: Neuron eines RNN..... 16
Abbildung 10: Erste Initialisierung k-means [44] 34
Abbildung 11: Bilden der Entscheidungsgrenze k-means [44] 34
Abbildung 12: Verschieben der Zentroiden zum Mittelpunkt k-means [44]..... 35
Abbildung 13: Zwischenschritt Iterationen und Endergebnis k-means [44] 35
Abbildung 14: Flussdiagramm Clustering-Analyse 36
Abbildung 15: Dreidimensionale Darstellung 2 Beispielssätze ungewichtet..... 39
Abbildung 16: Dreidimensionale Darstellung 2 Beispielssätze gewichtet..... 40
Abbildung 17: Dreidimensionale Darstellung 6 Beispielssätze ungewichtet..... 41
Abbildung 18: Dreidimensionale Darstellung 6 Beispielssätze gewichtet..... 41
Abbildung 19: Flussdiagramm überwachtes Lernverfahren 43
Abbildung 20: Trainierter Entscheidungsbaum 45
Abbildung 21: Flussdiagramm CNN 49
Abbildung 22: Balkendiagramm absolute Zahlen Clustering Rückrufe RAPEX..... 52
Abbildung 23: Kreisdiagramm relative Zahlen Clustering Rückrufe RAPEX 53
Abbildung 24: Balkendiagramm absolut Rückrufe NHTSA 54
Abbildung 25: Kreisdiagramm relativ Rückrufe NHTSA 55
Abbildung 26: Balkendiagramm absolut betroffene Fahrzeuge NHTSA..... 55
Abbildung 27: Kreisdiagramm relativ betroffene Fahrzeuge NHTSA 56
Abbildung 28: Absolute Anzahl softwarebedingte Rückrufe pro Modelljahr 56

Abbildung 29: Relativer Anteil softwarebedingte Rückrufe pro Modelljahr 57
Abbildung 30: Absolute Anzahl softwarebedingte Rückrufe in Fahrzeugen pro Modelljahr 58
Abbildung 31: Relativer Anteil softwarebedingte Rückrufe in Fahrzeugen pro Modelljahr 59
Abbildung 32: Relativer Anteil Software-Fehler-Rückrufe pro Marke 62
Abbildung 33: Relativer Anteil Software-Fehler-Fahrzeuge pro Marke..... 62
Abbildung 34: Anteil softwarebedingte Rückrufe an Gesamtrückrufen in Fahrzeugen pro Hersteller 64
Abbildung 35: Differenz Rückrufjahr – Modelljahr 66
Abbildung 36: Normierte Differenz logarithmisch mit Trendlinie - Rückrufe..... 67
Abbildung 37: Normierte Differenz logarithmisch mit Trendlinie – Fahrzeuge 68

Tabellenverzeichnis

Tabelle 1: Paarvergleichsmatrix Anforderungskriterien ML-Verfahren 22
Tabelle 2: Entscheidungsmatrix ML-Verfahren 23
Tabelle 3: Vektorisierter Beispielssatz 29
Tabelle 4: Vektorisierte Beispielssätze 29
Tabelle 5: Vokabular Beispielsätze 31
Tabelle 6: Tokenisierte Beispielsätze (Padding) 32
Tabelle 7: Gepaddete Beispielsätze 32
Tabelle 8: Gewichtete vektorisierte Beispielssätze 38
Tabelle 9: Reduzierte Darstellung ungewichtete und vektorisierte Beispielssätze 39
Tabelle 10: Reduzierte Darstellung gewichtete und vektorisierte Beispielssätze .. 40
Tabelle 11: Konfusionsmatrix 1000 Stichproben gewichteter k-means 42
Tabelle 12: Stichprobenauswertung ML-Verfahren 50
Tabelle 13: Lineare Extrapolation Historie Software-Rückrufe 57
Tabelle 14: Durchschnitt, Standardabweichung und Median Anzahl Fahrzeuge pro Rückruf HW/SW 60
Tabelle 15: Lineare Extrapolation Historie Software-Fahrzeuge 61

Abkürzungsverzeichnis

ADAC	Allgemeiner Deutscher Automobil-Club
AI	Artificial Intelligence
BAuA	Bundesanstalt für Arbeitsschutz und Arbeitsmedizin
BVL	Bundesamt für Verbraucherschutz und Lebensmittelsicherheit
CNN	Convolutional Neural Network
FDA	Food and Drug Administration
FFNN	Feedforward Neural Network
FMVSS	Federal Motor Vehicle Safety Standard
HW	Hardware
KBA	Kraftfahrt-Bundesamt
KI	Künstliche Intelligenz
ML	Machine Learning
NHTSA	National Highway Traffic Safety Administration
OTA	Over-the-Air
ProdSG	Produktsicherheitsgesetz
PS	Produktsumme
RAPEX	Rapid Exchange of Information System
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
StVG	Straßenverkehrsgesetz
SW	Software
TW	Technischer Wert

1. Einleitung

1.1. Motivation

In Zeiten von Level-3-Autopiloten in serienreifen Fahrzeugen – wie dem seit April 2022 verfügbaren Drive Pilot von Mercedes-Benz [1] – und immer weiter fortschreitender Digitalisierung und Automatisierung ist Software (SW) fortlaufend zu einem unverzichtbaren Teil des Automobils sowie einem wichtigen Aspekt für die aktive als auch passive Sicherheit der Insassen geworden. Gab es vor etwa 50 Jahren nur eine Handvoll Fahrzeuge, in denen Software überhaupt zum Einsatz kam, so sind heute durchaus mehr als 100 Millionen Zeilen Code [2, S. 215] in einem Fahrzeug enthalten, um den hohen technologischen Anforderungen der Kunden sowie der anspruchsvollen Gesetzgebung gerecht zu werden. Da diese Zahlen sogar die von Flugzeugen übertreffen und der Code auch im Fahrzeug im Ernstfall über Leib und Leben entscheiden kann, sind sowohl strenge Kontrollen bei der Zulassung als auch die Überwachung der Fahrzeuge im Feld notwendig.

Die Überwachung im Feld erfolgt durch den Hersteller sowie durch verschiedene Behörden. Bei auftretenden und gemeldeten sicherheitsrelevanten Mängeln werden betroffene Fahrzeuge durch die Behörde oder den jeweiligen Hersteller identifiziert, die Halter mit den jeweiligen Rückrufdaten kontaktiert und um einen Werkstattaufenthalt gebeten.

Laut Prognose aus dem Jahr 2019 soll sich der Umsatzanteil von Software in der Automobilindustrie innerhalb des Zeitraumes 2020 bis 2030 von 20 auf 50 Milliarden US-Dollar mehr als verdoppeln [3]. Somit steigt der Marktanteil und der Einfluss von Software im Automobilbereich auch in Zukunft weiterhin stark an.

Auch ein bemerkenswerter Aspekt ist, dass die Rückrufzahlen der Automobilindustrie stark steigen. Allein 2021 waren es 3,4 Millionen Fahrzeuge in Deutschland und somit 12 % mehr als im Vorjahr 2020 [4, S. 22]. Folglich waren also allein im Jahr 2021 statistisch etwa 5 % aller in Deutschland zugelassenen Fahrzeuge von einem Rückruf betroffen. Innerhalb Deutschlands waren die meisten Fahrzeuge von Rückrufaktionen in Bezug auf Assistenzsysteme, einem stark von Software abhängigen Teil des Fahrzeuges, betroffen [4, S. 24].

1.2. Zielsetzung

Rückrufe stellen für das Image als auch für das Budget der Automobilhersteller einen großen Schaden dar. Dementsprechend ist es von großem Interesse und auch ein Teil dieser Arbeit, aufzuarbeiten, ob der Anstieg der komplexen Software auch mit den Zahlen der Rückrufe korreliert. Die Rückrufdaten der Behörden sollen untersucht und der Einfluss von Software-Fehlern quantifiziert werden.
Die genannten Themen werden mithilfe der Datenbanken von internationalen Behörden untersucht. Aufgrund der enormen Datenmengen wird auf maschinelle Lernmethoden zurückgegriffen.
Um dies zu bewerkstelligen, wird im Zuge dieser Arbeit die Anwendbarkeit unterschiedlicher Machine-Learning-Modelle betrachtet. Dabei wird sowohl auf Methoden des sogenannten Supervised Learning als auch des Unsupervised Learning näher eingegangen.
All die genannten Themenbereiche sollen unter zwei Leitfragen bearbeitet werden:

Die erste Leitfrage lautet, ob und wie maschinelle Lernmethoden für die geforderte Kategorisierung eingesetzt werden können.

Die zweite Leitfrage beschäftigt sich damit, wie häufig Software-Fehler die Ursache von Fahrzeugrückrufen sind, wie der historische Verlauf aussieht und ob ein Zusammenhang mit Fahrzeugherstellern hergestellt und begründet werden kann.

1.3. Aufbau der Arbeit

Im ersten Teil der Arbeit werden grundlegende Themen wie Begriffsdefinitionen, der wirtschaftliche Einfluss von Rückrufen, der behördliche Ablauf von Rückrufaktionen und der Aufbau der Rückrufdatenbanken thematisiert. Außerdem werden Machine Learning Algorithmen genauer betrachtet. Differenziert wird dabei zwischen den vier Gruppen Unsupervised Learning, Supervised Learning, Reinforcement Learning und Neuronalen Netzen. Hier wird ein grober Überblick über das Thema aufgebaut.

Der zweite Teil beinhaltet einen Literatur- und Forschungsergebnisabgleich, um aufzuzeigen, welche Bereiche des Themas schon abgedeckt und welche Bereiche es noch nicht sind. Zudem werden die zuvor vorgestellten Machine Learning-Algorithmen auf ihre Eignung hinsichtlich dieser Arbeit untersucht und auch ausgewählt.

Im dritten Abschnitt wird mithilfe einer deskriptiven Datenanalyse eine Beschreibung der vorhandenen Daten durchgeführt. Im Anschluss wird eine sinnvolle Vorbereitung der Daten beschrieben und auf die vorliegenden Daten angewendet.

Der vierte Teil der Arbeit beschreibt die genaue Funktionsweise und erforderliche Anpassungen und Verwendungen der gewählten maschinellen Lernmethoden. Hierbei wird insbesondere auf Clustering, Entscheidungsbäume und das Convolutional Neural Network eingegangen.

Im fünften Teil werden die Ergebnisse diskutiert und das am besten funktionierende Verfahren zur Kategorisierung gewählt. Im Anschluss daran werden die kategorisierten Daten nun auf unterschiedliche Arten ausgewertet und die Auswertungen diskutiert.

Zuletzt wird das Gesamtergebnis der Arbeit zusammengefasst. Auch werden die Leitfragen hier nun beantwortet. Zudem werden anhand der aktuellen Ergebnisse ein Ausblick in die Zukunft und mögliche Handlungsempfehlungen für die zukünftige Entwicklung getroffen.

2. Einführung in das Thema

2.1. Fahrzeugrückrufe

Der Rückruf von Produkten als solches ist kein neues Erscheinungsbild. Die ersten Produktrückrufe datieren zurück auf das Jahr 1903 und betrafen den damaligen Automobilhersteller Packard [5, S. 16].
Schon in diesem Beispiel ist sichtbar, dass die ersten Produktrückrufe Kraftfahrzeuge betrafen. Bis heute sind Rückrufe aufgrund der hohen Komplexität kostspielige Angelegenheiten für Automobilhersteller.
Historisch betrachtet waren Rückrufe anfangs keine groß publizierte Maßnahme. Erst 1966 begannen die amerikanischen Behörden der Verkehrssicherheit, Rückrufe offiziell zu speichern und zu veröffentlichen.

2.1.1. KBA

Das Kraftfahrt-Bundesamt (KBA) ist die deutsche Behörde, die für alle Produkte, die dem Straßenverkehrsgesetz (StVG) unterliegen, zuständig ist. Dementsprechend fallen laut § 1 StVG Kraftfahrzeuge und Anhänger darunter, die auf öffentlichen Straßen in Betrieb genommen werden sollen. Ausgenommen sind hierbei schienengebundene Fahrzeuge, Fahrräder und E-Bikes mit einer Nenndauerleistung von höchstens 0,25 kW, dessen Unterstützung sich mit zunehmender Geschwindigkeit progressiv verringert und ab spätestens 25 km/h unterbrochen wird als auch Gegenstände, die zwar in Fahrzeugen verwendet werden, deren Sicherheitsanforderungen aber schon in eigenen Rechtsvorschriften festgelegt wurden [6, S. 5].
In den Zuständigkeitsbereich des KBA fallen unter anderem Rückrufe und andere Maßnahmen nach dem Produktsicherheitsgesetz (ProdSG). Ein Rückruf zielt auf eine vorübergehende oder endgültige Rückgabe eines bereits in den Verkehr gebrachten Produkts durch den Verwender [6, S. 7].
Bei den durch das KBA verwalteten Rückrufarten finden sich drei Kategorien:

1. Freiwilliger Rückruf:
   Dieser Rückruf basiert begriffsnotwendig auf der Freiwilligkeit des jeweiligen Wirtschaftsakteurs (Hersteller, Bevollmächtigte, Einführer oder Händler

[6, S. 6]). Der Wirtschaftsakteur kann selbstständig eine Maßnahme einleiten, wenn ihm Informationen darüber vorliegen, dass ein Produkt ein Risiko darstellt oder fehlerhaft ist. Sobald das KBA von einem freiwilligen Rückruf erfährt, prüft es, ob der Verdacht auf ein ernsthaftes Risiko besteht und ob in diesem Fall die Maßnahmen des Wirtschaftsakteurs ausreichend sind. Sollte ein ernstes Risiko vorliegen, wird der freiwillige Rückruf zu einem überwachten Rückruf.

Bei einem Verbraucherprodukt wird zusätzlich geprüft, ob der Wirtschaftsakteur gegen die Meldepflicht nach Art. 5 Abs. 3 der Richtlinie 2007/95/EG [7, 63ff.] verstoßen hat.

2. Überwachter Rückruf:

Die Form des überwachten Rückrufes stellt einen ernsteren Fall des Rückrufes dar. Zunächst informiert der Wirtschaftsakteur entsprechend seiner Meldepflicht das KBA über ein mangelhaftes Produkt, das ein ernstes Risiko darstellt. Diese Meldung prüft das KBA und wenn die Prüfung ergibt, dass ein ernsthaftes Risiko besteht, werden die getroffenen Maßnahmen dahingehend kontrolliert, ob sie ausreichen. Im Regelfall wird darauffolgend eine Überwachung des Rückrufes angeordnet. Hierbei wird die Wirksamkeit der Maßnahmen sowie deren Verlauf anhand der Übermittlung von Abarbeitungszahlen überprüft [6, S. 8].

Im Falle des Vorliegens eines ernsten Risikos wird seitens des KBA eine sogenannte maximale Erfüllungsrate gefordert. Diese ist gegeben, wenn der verantwortliche Wirtschaftsakteur einen Rückruf durchführt und unter Anwendung aller notwendigen und angemessenen Mittel zu erwarten ist, dass alle Verwender erreicht werden [6, S. 9]. Sie kann – bei kennzeichnungspflichtigen Fahrzeugen im Falle eines Unterlassens der Forderung sowie einer Fristüberschreitung – zur Betriebsuntersagung des Fahrzeuges führen [6, S. 8].

3. Angeordneter Rückruf:

Der schwerwiegendste und seltenste Fall ist der angeordnete Rückruf, welcher von dem KBA dann angewandt wird, wenn der verantwortliche Wirtschaftsakteur keine oder keine geeigneten Maßnahmen vorschlägt, um das ernste Risiko schnell und wirksam zu beseitigen [6, S. 8].

Auch in diesem Fall wird die oben genannte maximale Erfüllungsrate erwartet.

Sowohl überwachte als auch angeordnete Rückrufe werden durch das KBA auf das Vorliegen eines ernsten Risikos geprüft. Ein ernstes Risiko liegt nach ProdSG dann vor, wenn das von einem Produkt ausgehende Risiko ein rasches Eingreifen der Marktüberwachungsbehörde erfordert [6, S. 6]. Nach positiver Prüfung werden die vorliegenden Informationen an die Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (BAuA) gemeldet [6].

Diese Daten werden nach dortiger Prüfung in der Datenbank des Rapid Exchange of Information System (RAPEX) veröffentlicht.

Hinweise für die Aufnahme von Untersuchungen und gegebenenfalls Anordnungen von Rückrufen durch das KBA können durch den Wirtschaftsakteur selbst aber auch durch den sogenannten Mangelmelder des KBA – ein Online-Tool für die Meldung durch Endnutzer – gesammelt werden. Zudem greift das KBA auf eigene als auch beauftragte Produktuntersuchungen, Unfälle, Medienberichte und ähnliches zurück [8].

Das KBA veröffentlicht die Daten der nationalen Rückrufaktionen auch selbst in einer eigenen Datenbank. Eine vereinfachte Darstellung des Ablaufes ist in Abbildung 1 zu sehen.

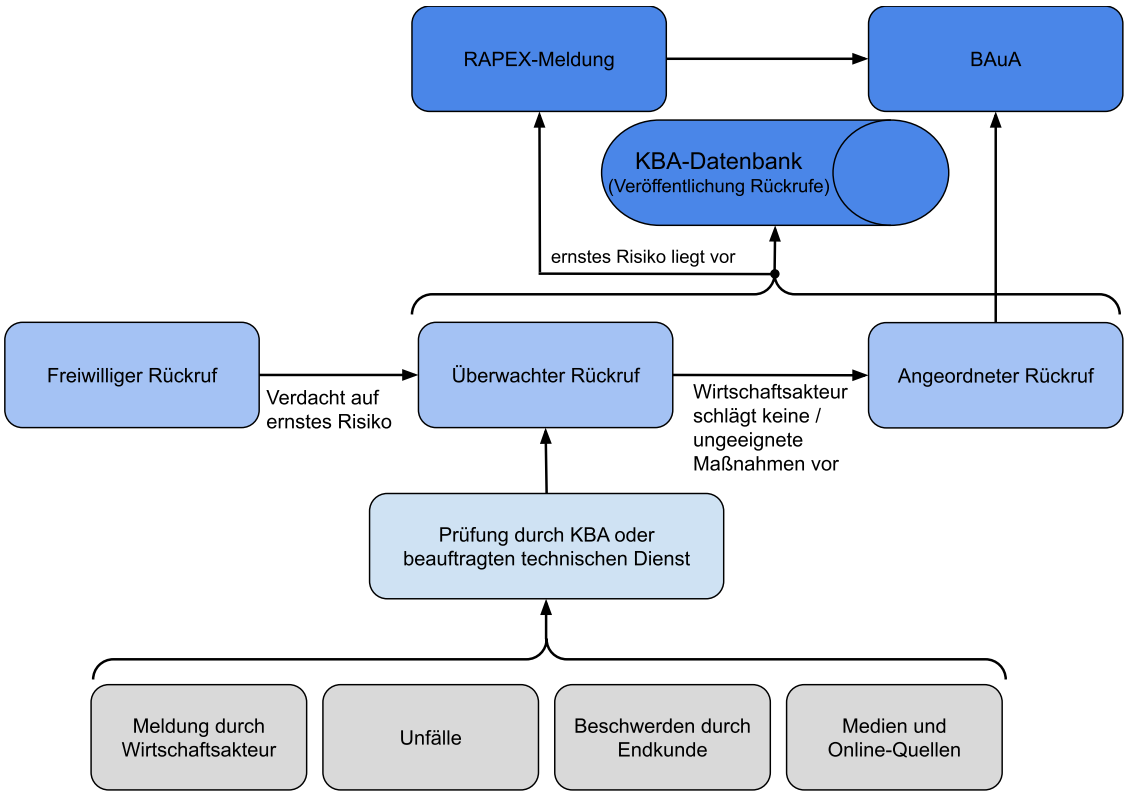

Abbildung 1: Graphische Darstellung Meldeweg KBA

Allerdings sind – auch auf schriftliche Anfrage bei dem KBA – die Daten der KBA-Datenbank nicht gebündelt abrufbar und somit für den weiteren Verlauf der Arbeit nicht direkt nutzbar. Da die Rückrufe des KBA, bei denen ein ernsthaftes Risiko besteht, auch über das RAPEX-System veröffentlicht werden, können diese aber indirekt über die RAPEX-Datenbank berücksichtigt werden.

2.1.2. RAPEX

Um ein internationales Rückrufsystem im europäischen Wirtschaftsraum aufzubauen und Verbraucher möglichst schnell und weitreichend vor Gesundheitsgefährdungen zu warnen, wurde das europäische Schnellwarnsystem für Verbraucherprodukte RAPEX in das Leben gerufen. Jährlich werden laut Bundesamt für Verbraucherschutz und Lebensmittelsicherheit (BVL) hierbei ca. 2000 Meldungen [9] eingetragen und Endnutzer vor Gefahren gewarnt.

RAPEX dient hierbei als Sammelstelle der europäischen Behörden, die für Produktsicherheit und -rückrufe zuständig sind. Jedes Land hat eine festgelegte Behörde, die für die Meldungen an RAPEX verantwortlich ist. In Deutschland ist beispielsweise die meldende Behörde das BAuA [siehe 10].

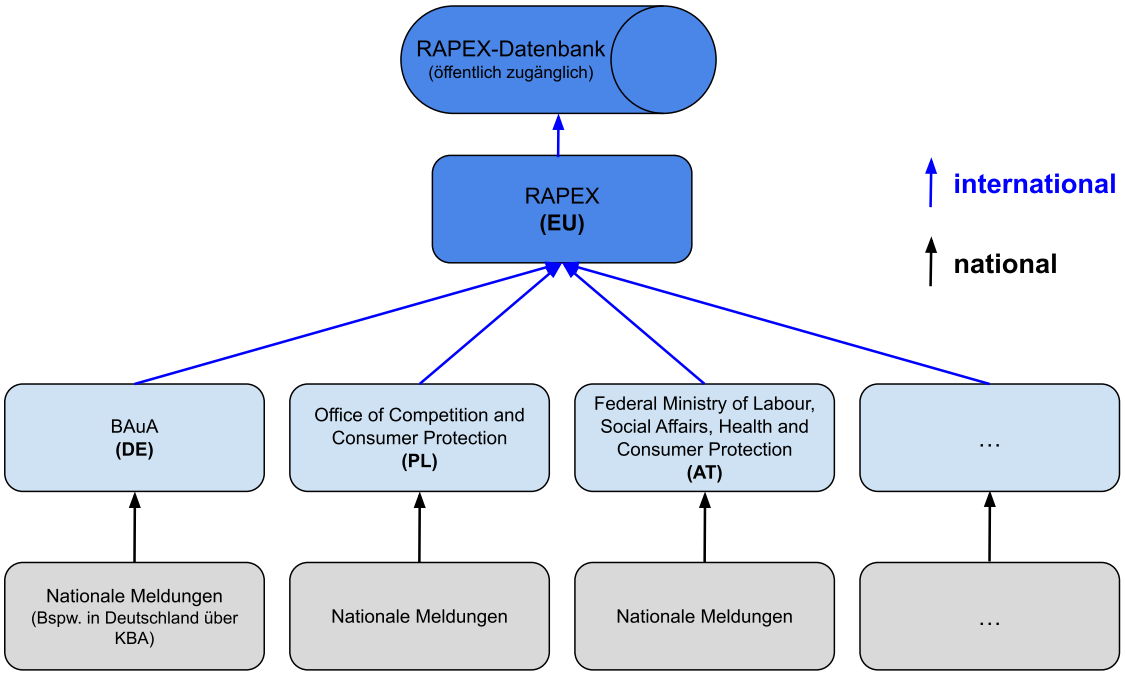

Abbildung 2: Graphische Darstellung Meldeweg RAPEX

Nach einer erfolgreichen Meldung und Prüfung eines Rückrufes wird dieser wöchentlich in der RAPEX-Datenbank veröffentlicht (siehe Abbildung 2). Explizit bezogen auf Kraftfahrzeuge finden sich hier etwa 5000 Fahrzeugrückrufe (Stand 04/2023) seit dem Jahr 2005.

2.1.3. NHTSA

Das amerikanische Gegenstück in dem Bereich der Fahrzeugrückrufe zum KBA heißt National Highway Traffic Safety Administration (NHTSA) und ist die dortige Bundesbehörde für Straßen- und Fahrzeugsicherheit.

Der Meldeweg von NHTSA kann bei dem jeweiligen Wirtschaftsakteur als auch bei dem Endnutzer beginnen. Mögliche Quellen für Informationen über fehlerhafte Produkte können hierbei Hersteller, Endnutzer, Strafverfolgungsbehörden, Medien und Online-Quellen sein. Auch auf Informationen von Fahrzeugforschungs- und Testzentren sowie auf spezielle Unfalluntersuchungen wird durch die NHTSA zurückgegriffen. Auf der Seite der NHTSA findet sich hierfür auch ein Formular, um als Endnutzer sicherheitsrelevante Mängel an die Behörde zu melden.

Anhand der genannten Informationsquellen wird durch die NHTSA kontinuierlich geprüft, wie häufig die Mängel auftreten und wie problematisch der potenzielle Schweregrad der Mängel ist. Wird hierbei festgestellt, dass ein Mangel sicherheitsrelevant ist, besteht die Möglichkeit der Einleitung einer ausführlichen Untersuchung, um festzustellen, ob ein Rückruf durch den Hersteller gerechtfertigt wäre [11]. Auch zu einem Rückruf kann führen, wenn festgestellt wird, dass das Fahrzeug nicht dem Federal Motor Vehicle Safety Standard (FMVSS) entspricht. Dies ist eine Vorschrift, die die Mindestvoraussetzungen für sicherheitsrelevante Bauteile innerhalb der USA festlegt [12, S. 2].

Treten sicherheitsrelevante Mängel auf oder wird eine Nonkonformität zur oben genannten Vorschrift festgestellt, die zu einem Rückruf des Fahrzeugs führen, werden diese Rückrufe veröffentlicht und in die Datenbank der NHTSA eingetragen.

Zur Vereinfachung ist der Ablauf des Meldeweges in Abbildung 3 graphisch dargestellt.

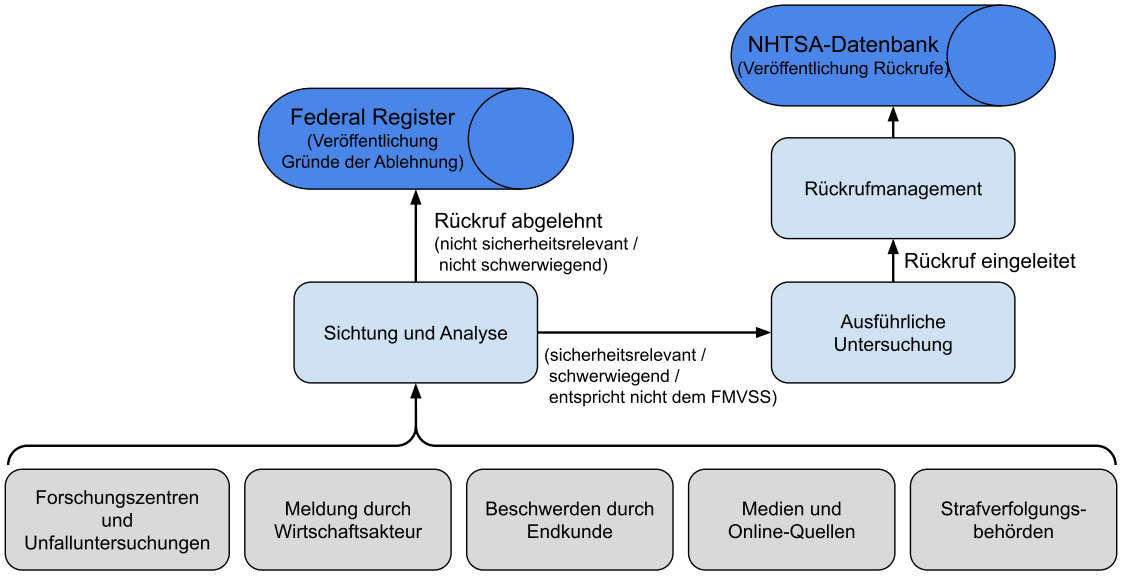

Abbildung 3: Graphische Darstellung Meldeweg NHTSA

Die NHTSA-Datenbank beinhaltet etwa 240.000 Einträge zu Fahrzeugrückrufen seit dem Jahr 1966. Somit sind die Daten für den weiteren Verlauf – insbesondere für historische Betrachtungen – ideal verwendbar.

2.2. Machine Learning, Deep Learning, Artificial Intelligence

„Künstliche Intelligenz ist die Fähigkeit eines Computers oder computergesteuerten Roboters, Aufgaben zu lösen, die normalerweise von intelligenten Wesen erledigt werden“ [13]. Mit diesem Zitat versucht B.J. Copeland die sogenannte Künstliche Intelligenz (KI) in einer Definition zusammenzufassen. Hierbei ist zu beachten, dass Künstliche Intelligenz ein weit gefasster Begriff ist, der sehr oft und teilweise vorschnell Anwendung findet. Außerdem ist auch Künstliche Intelligenz nicht gleich Künstliche Intelligenz. Der aktuelle Stand ist, dass es mehrere Schichten der KI gibt und hauptsächlich zwischen den drei Kategorien Artificial Intelligence (AI), Machine Learning (ML) und Deep Learning (DL) unterschieden wird [14].

Artificial Intelligence beschreibt die Oberkategorie der drei Begriffe. Hierunter fällt jeder künstliche Prozess, der Entscheidungen trifft, indem er menschliches Verhalt nachahmt.

Unter Machine Learning sind Methoden zu verstehen, bei denen Maschinen anhand von Trainingsdaten lernen, ihnen unbekannte Daten einzuordnen. Als Beispiele fallen hierunter die Klassifikation, Regression als auch das Deep Learning.

Deep Learning ist eine weitere Unterkategorie und Teilgebiet des ML. Es wird angewandt, um mithilfe von mehrschichtigen Systemen aus großen Datenmengen zu lernen.

Eine graphische Darstellung der KI-Kategorien findet sich in Abbildung 4.

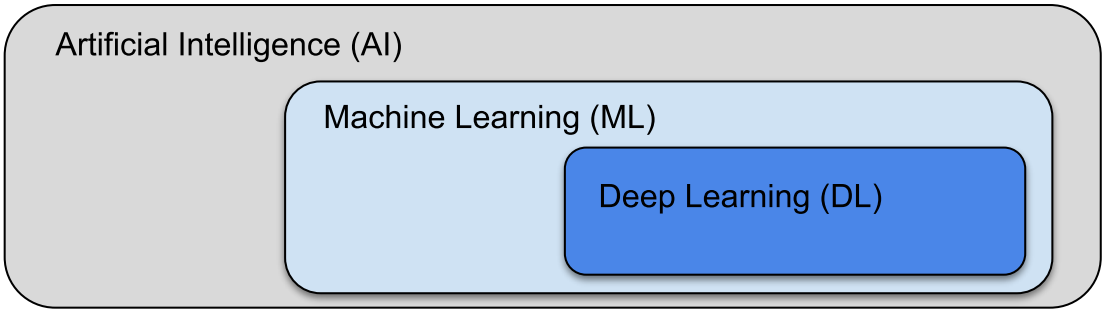

Abbildung 4: Schematische Abbildung Unterkategorien AI

Für den folgenden Verlauf der Arbeit werden Verfahren des Machine Learning und des Deep Learning angewandt, um Rückrufdaten zu kategorisieren.

Allgemein wird bei ML-Verfahren zwischen unterschiedlichen Ansätzen differenziert: Unsupervised Learning, Supervised Learning und Reinforced Learning. Der Ansatz der neuronalen Netze ist Bestandteil des Deep Learning.

Prinzipiell basieren die Verfahren auf einem sogenannten Software-Agenten (siehe Abbildung 5), der einem einfachen Eingabe-Verarbeitung-Ausgabe-Vorgehen ähnelt.

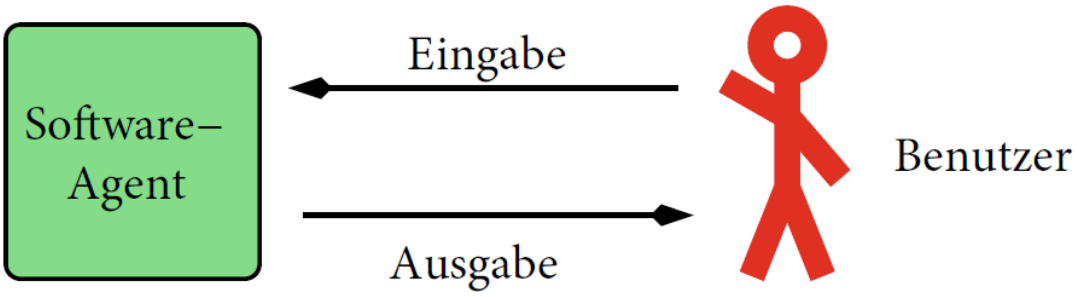

Abbildung 5: "Ein Software-Agent mit Benutzerinteraktion" [15, S. 18]

Es wird ein Eingabesignal in das Modell eingegeben. Das Modell agiert wie eine Blackbox und gibt anhand von zuvor gelernten Vorgehensweisen ein Ergebnissignal wieder aus.

2.2.1. Unsupervised Learning

Unsupervised Learning – auch unüberwachtes Lernen genannt – beschreibt die Methode, mit der ein Computer ohne manuelle Vorgaben Muster in Daten erkennt und damit „lernt“.
Vereinfacht gesagt, werden Daten in die Maschine eingespeist. Es wird aber keine Lösung vorgegeben und die Maschine kategorisiert die Daten nach eigenem Ermessen oder zeigt erkannte Zusammenhänge an.
Die mathematische Betrachtung des Unsupervised Learning ist die Erstellung einer mathematischen Funktion, um von einer Eingabe X auf eine Ausgabe Y zu schließen.
Hierfür können auch Modelle anhand von Datensätzen (vor-)trainiert werden. Mit einem trainierten Modell oder steigenden Datenzahlen werden die ML-Verfahren immer genauer. Da die Verfahren aber mit unmarkierten Daten arbeiten und keine

Lösung vorgegeben wird, kann auch kein Fehler berechnet oder das Modell nicht belohnt werden [16, S. 24].

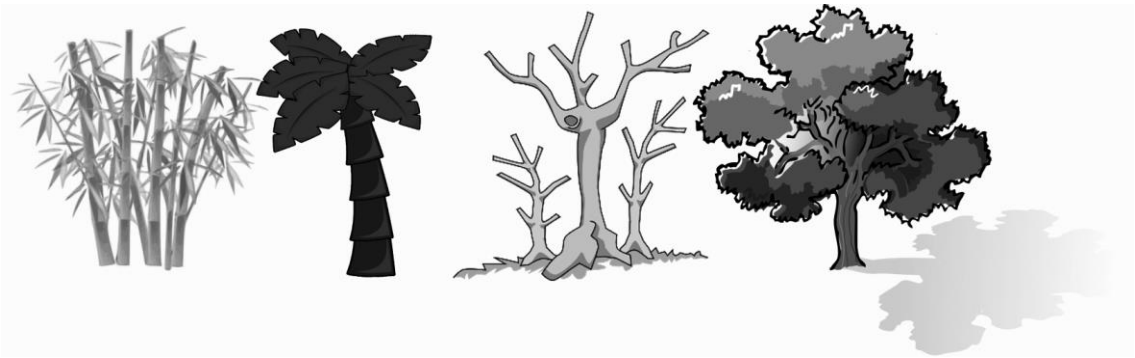

Abbildung 6: „Zeichnungen verschiedener Pflanzen“ [16, S. 25]

Ein passendes Beispiel von Jörg Frochte [16] für die Variabilität, die aus unüberwachten Lernmethoden entstehen kann, findet sich in der Abbildung 6. Hier sind vier unterschiedliche Bäume dargestellt und je nachdem, welche Personen diese Bilder betrachten, würden wahrscheinlich unterschiedliche Zusammenhänge erkannt werden.

Eine Gruppe würde den dritten Baum aussortieren, da dieser im Gegensatz zu den anderen keine Blätter trägt und abgestorben aussieht. Die nächste Gruppe hingegen würde den ersten Baum aussortieren, weil es kein Gehölz ist [16, S. 25]. Eine dritte Gruppe würde vielleicht wegen der weit oben angebrachten Blätter das zweite und vierte Bild gruppieren.

Theoretisch könnte es so unendlich lange weitergehen, da jeder anhand unterschiedlicher Merkmale, an denen er sich orientiert, auch anders gruppieren könnte. So unterschiedlich diese Lösungen alle sind, sind sie alle trotzdem nicht falsch. Schließlich geht es um eine reine Gruppierung anhand ähnlicher oder eben eine Differenzierung sehr divergierender Attribute.

Dieses Vorgehen symbolisiert das Prinzip der sogenannten Clustering-Analyse des Unsupervised Learning. Bei diesem Verfahren ist es das Ziel, sogenannte Cluster zu finden, die sich durch geringe Distanz der Clustermitglieder zueinander darstellen (Intragruppen-Homogenität) [17, S. 180].

Gleichzeitig wird versucht, dass zwischen den Clustern eine möglichst große Distanz vorliegt (Intergruppen-Heterogenität) [17, S. 180]. Besonders eignet sich diese

Methode, um noch unbekannte Strukturen und Zusammenhänge oder Ausreißer in Datensätzen zu entdecken.
Ein weiteres Verfahren des Unsupervised Learning ist die Assoziationsanalyse. Hierbei handelt es sich um einen Algorithmus, der beispielsweise dafür verantwortlich ist, nach einem Kauf bei einem beliebigen Online-Portal ähnliche oder dazu passende Produkte vorzuschlagen. Dies basiert auf einer Analyse der Verhaltensweisen der Kunden und der Assoziation ähnlicher Verhaltensweisen. Kaufen zum Beispiel viele Kunden bei Nudeln auch Pesto dazu, wird zukünftigen Kunden bei Abschluss eines Kaufes von Nudeln eine Packung Pesto vorgeschlagen.
Es gibt noch viele weitere Verfahren des Unsupervised Learning. Die beiden genannten sind aber die am häufigsten verwendeten Verfahren. Insbesondere die Cluster-Analyse stellt sich für den Einsatzzweck in dieser Arbeit sehr sinnvoll dar.
Allgemein kann das Unsupervised Learning sehr gut eingesetzt werden, um Muster in Daten zu erkennen, die vielleicht vorher noch nicht entdeckt wurden. Zudem ist ein enormer Vorteil, dass keine vorab manuell gelabelten Daten benötigt werden. Von gelabelten Daten spricht man im Sinne der ML-Verfahren dann, wenn die Rohdaten schon kategorisiert wurden [18]. Ungelabelt bedeutet in dem Falle dann das Fehlen der Angabe zur Kategorie. Im hiesigen Anwendungsfall entsprechen die Label der Zuordnung der einzelnen Rückrufe zur Kategorie „softwarebedingt“ und „nicht softwarebedingt“.
Andererseits haben die Methoden auch schwerwiegende Nachteile. Unüberwachte Methoden sind zumeist deutlich ungenauer als beispielsweise Überwachte Lernmethoden. Hinzu kommt, dass es oftmals sehr schwer ist, Modellhyperparameter, passend einzustellen. Denn meist ist es nicht klar, wie das Modell arbeitet und nach welchen Mustern es kategorisiert. Die Hyperparameter sind die Parameter, mit denen die ML-Verfahren im Training gesteuert werden können, z. B. die Lernrate oder Parameter der Optimierungsfunktion [19].

2.2.2. Supervised Learning

Supervised Learning – auch überwachtes Lernen genannt – ist das Vorgehen einer Maschine anhand von gekennzeichneten Daten beizubringen, wie sie Entscheidungen treffen soll. Hierfür bedarf es beispielsweise bei Datensätzen immer eines manuell gelabelten Datensatzteils, an dem das ML-Modell trainiert werden kann.

Dadurch werden Muster erkannt, auf deren Basis Entscheidungen für die Ausgabe anhand einer Eingabe erfolgen soll. Beispiele, in denen der Einsatz der Methode besonders geeignet ist, sind Klassifikationen als auch Regressionen [16, S. 21].
Vorteil des Supervised Learning ist die hohe Präzision der Ausgabe und eine einfache Prüfbarkeit der Genauigkeit anhand von gelabelten Prüfdatensätzen.
Allerdings ist zu beachten, dass die Genauigkeit sehr stark mit der Quantität als auch der Qualität der gelabelten Traininingsdatensätzen zusammenhängt. Des Weiteren kann es zur Herausforderung werden, genügend gelabelte Trainingsdaten zu sammeln, um das ML-Modell ausreichend zu trainieren. Zudem bedarf es beim Supervised Learning vorab eines manuellen Labeln, das für den Nutzer zeitintensiv ist. Im Gegensatz zum Unsupervised Learning können hier auch keine neuen Muster erkannt werden, die nicht schon in den Trainingsdaten vorhanden sind.
Eine für den Einsatzzweck dieser Arbeit geeignete Methode des Unsupervised Learning ist die Decision-Tree-Klassifikation. Hierbei handelt es sich um ein effizientes Verfahren, das eine klare Einteilung in Kategorien ermöglicht.

2.2.3. Reinforcement Learning

Das Reinforcement Learning bezeichnet eine maschinelle Lernmethode, die in einer dynamischen Umgebung agiert. Das System muss ein bestimmtes Ziel erreichen, ohne zu wissen, was das Ziel ist oder ob es dem Ziel nahegekommen ist. Zu gewissen Zeitpunkten erhält das System ein positives oder negatives Feedback. Dadurch versucht es, die beste Strategie für das aufgeführte Problem zu finden.
Die zwei wichtigsten Attribute des Reinforcement Learning sind also das Trial-and-Error-Prinzip sowie das verzögerte Feedback (virtuelle Belohnung/Bestrafung) [20, S. 1-2].
Aufgrund der recht allgemeinen und weit reichenden Anwendbarkeit ist ein großer Vorteil des Modells die Einsetzbarkeit in verschiedensten Zwecken. Zudem ist es auch in komplexen und dynamischen Umgebungen einsetzbar, in denen es kompliziert wäre, eine vorab festgelegte Strategie zu finden. Ein weiterer Vorteil ist das lebenslange Lernen des Modells. Hierbei kann es sich durch Übung kontinuierlich anpassen und verbessern.
Als Problem muss der Blackbox-Effekt betrachtet werden. Modelle des Reinforcement Learning sind meist sehr kompliziert und basieren oftmals auf Neuronalen

Netzen. Daher ist es für einen Menschen häufig recht kompliziert oder gar unmöglich, genau nachzuvollziehen, wie das Modell Entscheidungen trifft. Aufgrund der hohen Komplexität erfordert die Methode auch eine hohe Rechenleistung und wird bei großer Skalierung sehr langsam und somit teuer.
Die Methoden des Reinforcement Learning sind für die statische Umgebung der Datenkategorisierung tendenziell ungeeignet und werden im Folgenden nicht weiter thematisiert.

2.2.4. Neuronale Netze

Als weiterer Ansatz finden sich die Neuronalen Netze wieder. Der Grundgedanke stammt aus der Natur und orientiert sich an Nervenzellen und Gehirnen. Das menschliche Gehirn besteht aus vielen Milliarden Neuronen und jedes der Neuronen übernimmt einen sehr kleinen Teil der Verarbeitung von Informationen. Erst durch die Verschaltung der Nervenzellen erreicht der Mensch seine Intelligenz und Fähigkeit auch mit komplexen Informationen umzugehen.
Auf diesem Konzept stützt sich die Grundidee der neuronalen Netze. Jedoch geschehen die Aktivierung und Verschaltung der Neuronen nicht auf Grundlage von elektrischen Potentialen, sondern auf mathematischer Basis.
Bei einem neuronalen Netz erhält ein einzelnes Neuron eine Anzahl n von Eingabeparametern x_n. Im Gegensatz zum menschlichen Gehirn geschieht das Sammeln der elektrischen Potentiale hier nun über die Summierung der eingehenden Signale [15, S. 268]. Die summierten Gewichte werden dann mithilfe einer sogenannten Aktivierungsfunktion f zu einer Ausgabe des Neurons berechnet. Diese Ausgabe wird dann als neu gewichtete Summe an die Neuronen der nächsten Schicht weitergegeben. Ein Modell eines solchen Neurons ist in Abbildung 7 dargestellt.

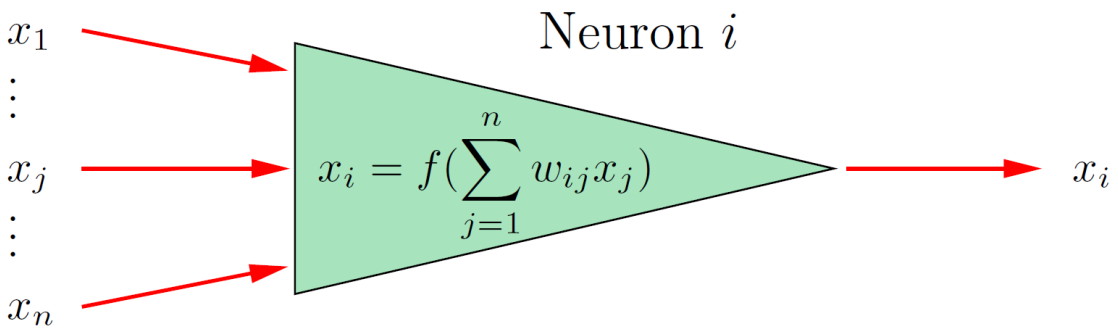

Abbildung 7: Aufbau eines formalen Neurons [15, S. 269]

Unter den Neuronalen Netzen wird zwischen drei Hauptkategorien unterschieden: Feedforward- (FFNN), Recurrent- (RNN) und Convolutional Neural Network (CNN).

Das FFNN ist ein Netzwerk aus Neuronen, bei denen die Informationen von einer Eingangsschicht aus Neuronen über versteckte Neuronen-Schichten zu einer Ausgangsschicht durchgeleitet werden. Dadurch wird ein Ausgangswert erzeugt. Es werden keine Feedback-Iterationen gemacht und die Neuronen leiten die Werte einfach nur weiter. Ein einfaches Neuron eines FFNN-Netzwerkes findet sich in Abbildung 8.

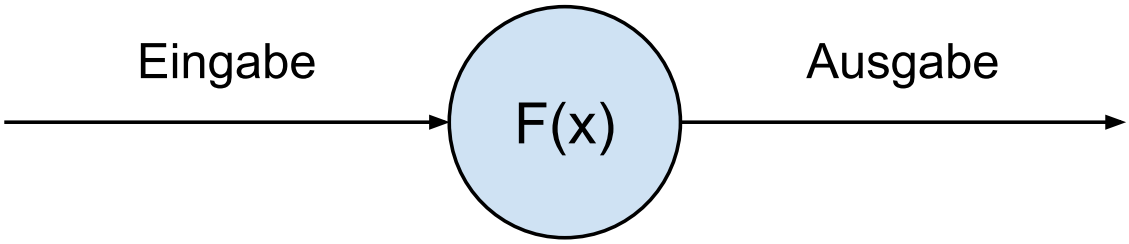

Abbildung 8: Neuron eines FFNN

Werden bei einem FFNN Feedback-Verbindungen eingebaut, so wird dieses Netzwerk zu einem RNN [21, S. 164]. Ein einfaches neuronales Netz mit nur einem Neuron mit Rückkopplung ist in Abbildung 9 dargestellt.

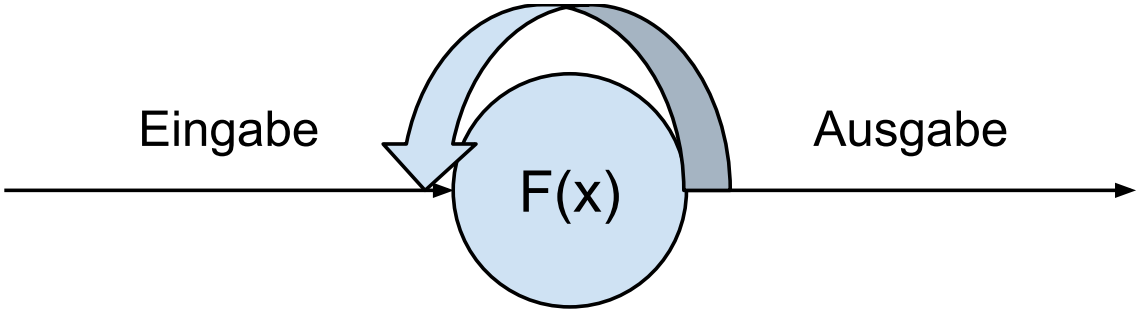

Abbildung 9: Neuron eines RNN

Das CNN wird bei komplexen Problemen verwendet. Hierbei werden spezielle mathematische Operationen eingesetzt, um die Komplexität des Netzwerks zu reduzieren. Die Netzwerke werden oft für Schriftzeichen- oder Bilderkennung eingesetzt, da diese viel Rechenleistung benötigen. Schließlich können die Netzwerke aufgrund von nicht vollständiger Vernetzung und sogenannter Faltungs- und Poolingsschichten [15, S. 303] den Rechenaufwand minimieren. CNN lassen sich auch auf tokenisierte Texte anwenden und zur Textklassifikation einsetzen. Es wird hier

insbesondere von der Mustererkennung der CNN profitiert, durch die hier Muster in Form von Wortsequenzen erfasst werden können [22].

Sowohl FFNN als auch RNN und CNN eignen sich für den Einsatz zur Textklassifikation. Alle drei werden im Laufe dieser Arbeit zur Kategorisierung eingesetzt und die Ergebnisse und Effizienz miteinander verglichen.

3. Methodik und Auswahl geeigneter Verfahren

3.1. Literatur- und Forschungsrecherche zur Leitfrage

Um einen Überblick über den aktuellen Stand der Forschung zu ermöglichen, wird eine Literatur- und Forschungsrecherche durchgeführt.
Im Folgenden wird der aktuelle Forschungsstand in den Bereichen Fahrzeugrückrufe und der Kategorisierung mithilfe von Machine Learning untersucht. Dafür werden deutsch- und englischsprachige Literatur und Publikationen näher betrachtet.

Das Thema Fahrzeugrückrufe hat einen erheblichen finanziellen Einfluss auf die Automobilbranche. Dementsprechend ist das Interesse daran sehr hoch. Dies schlägt sich auch in der hohen Anzahl der Publikationen nieder. Allerdings existieren kaum Ausarbeitungen, die explizit auf die Forschungsfrage des Einflusses von Software-Fehlern auf Fahrzeugrückrufe eingehen.
Im Allgemeinen werden von unterschiedlichen Instituten regelmäßig Studien zur Zuverlässigkeit von Fahrzeugen publiziert.
Der Allgemeine Deutsche Automobil-Club (ADAC) publiziert jährlich eine eigens erstellte Pannenstatistik [23]. Diese basiert auf der Auswertung der Straßenwachteinsätze [23, S. 8] des jeweiligen Jahres und stellt aktuelle Pannenkennziffern unterschiedlicher Fahrzeuge nach Fahrzeugklasse bereit. Die Zuverlässigkeit ist hier thematisch nicht an Fahrzeugrückrufe gebunden, sondern basiert auf eigenen Pannenstatistiken und ist somit nicht relevant für diese Arbeit.

Im amerikanischen Bereich stellt beispielsweise J.D. Power ähnliche Studien bereit [24] und thematisiert hierbei die Zuverlässigkeitsstatistiken auf Basis der Fahrzeugmarke. Dies geschieht allerdings anhand von Umfragen von etwa 30.000 Fahrzeugbesitzern [24, S. 3]. Somit hat die Studie ebenso keinen Zusammenhang zu Fahrzeugrückrufen und ist im weiteren Verlauf der Arbeit thematisch nicht relevant.

In einer Studie der European Alliance for Innovation (EAI) [25] werden die Einflüsse von Software-Fehlern auf Fahrzeugrückrufe thematisch angeschnitten. Jedoch werden nur einige wenige Fahrzeugrückrufe aufgezählt und oberflächlich erklärt. Eine genauere Untersuchung des Einflusses wird nicht durchgeführt, da der Fokus der

Ausarbeitung auf der Beschreibung von Normen zu Qualitätsmanagement und -sicherung der Software liegt [25, S. 3-5].

Auch auf Fahrzeugrückrufe geht eine Studie des China National Institute of Standardization [26] ein. Im Zuge der Studie werden aktuelle Trends der Rückrufe der NHTSA näher betrachtet und bewertet. Die Studie orientiert sich aber an Gesamtzahlen der Rückrufe und ist eine allgemeine historische Betrachtung. Der Fokus liegt hier auf Trends und Charakteristiken der Gesamtzahlen von Fahrzeugrückrufen und hat keinen Bezug zu Software-Fehlern.

Eine Studie, die eine sinnvolle Klassifikation unterschiedlicher Fahrzeugrückrufe untersucht, ist die des Beijing Institute of Technology [27] aus dem Jahr 2010. Hierbei wird auf NHTSA-Rückrufdaten mit dem Stand Oktober 2010 [27, S. 4307] zurückgegriffen. Die Kategorisierung der Rückrufe erfolgt mithilfe eines Cluster-Algorithmus. Es wird in der Studie auf die hierarchische Clusteranalyse mit einem Algorithmus in Matlab zurückgegriffen [27, S. 4308-4309]. Zusammenfassend werden in der Studie 32 Klassifikationskategorien für den Datensatz erarbeitet [27, S. 4310]. Die Kategorien beinhalten jedoch keine Software-Fehler, sondern gliedern sich in übergeordnete Kategorien wie beispielsweise „Electric equipment“, „Steering system“, „Front axle“, etc. Diese Kategorisierung ähnelt stark der „Component Description“ (COMPNAME), die bereits in der FLAT_RCL-Datei der NHTSA-Datenbank [28] enthalten ist. Für die in dieser Bachelorarbeit geforderte, wesentlich präzisere Kategorisierung ist die Studie daher nur von peripherer Relevanz.

Rückrufe, die mit Software zusammenhängen, sind im Zeitalter der Digitalisierung nicht nur in der Fahrzeugindustrie ein Problem. Abseits der Fahrzeugtechnik findet sich im Bereich der Medizintechnik die Untersuchung von Jay G. Ronquillo und Diana M. Zuckerman [29], die sich mit Rückrufen der medizinischen Informationstechnologie aufgrund von Software-Fehlern befasst. Anhand der Datenbank der US-Behörde für Lebensmittel und Arzneimittel, der Food and Drug Administration (FDA), werden die Rückrufe der letzten fünf Jahre, bei denen medizinische Geräte hauptsächlich aufgrund von Software-Fehlern zurückgerufen wurden, untersucht [29, S. 540-541]. Daraus resultierend werden die Ergebnisse diskutiert

[29, S. 546- 549] und im gleichen Zuge Handlungsempfehlungen für die FDA-Regularien ausgesprochen.

Abseits der genannten Veröffentlichungen hat Klaus Rüdiger Bährle im Rahmen einer Masterarbeit am Karlsruher Institut für Technologie eine „Marktanalyse zu sicherheitskritischer Software in der Automobilbranche“ [30] veröffentlicht. Unter anderem wird hier auch der Anteil der softwarebedingten Fahrzeugrückrufe anhand der RAPEX und NHTSA-Datenbank untersucht. Die genannte Arbeit stammt aus dem Jahr 2016 und basiert auf RAPEX- und NHTSA-Daten aus dem Jahr 2015 [30, S. 110] [30, S. 125].
Die Ausarbeitung behandelt ausführlich die Auswahl der Datenbanken und stellt daraus abgeleitete Statistiken dar. Die genaue Methodik der Untersuchung und Kategorisierung der Rückrufursachen wird allerdings nicht nachvollziehbar dargestellt. Auch beinhaltet sie keine ersichtlichen Hilfsmittel des maschinellen Lernens.
Des Weiteren erfolgt eine qualitative Untersuchung der Software-Fehler innerhalb des EU-Raumes. Aufgrund der sehr geringen Anzahl von betroffenen Rückrufen („98 softwarebedingte Rückrufe“ [30, S. 111]) wird dies mithilfe einer manuellen Betrachtung [30, S. 117] durchgeführt. Hierbei wird auf RAPEX als auch auf die Datenbank des ADAC zurückgegriffen [30, S. 111].
Abschließend wird in der Ausarbeitung ein Ergebnis für den prozentualen Anteil der Software-Fehler auf Basis der zurückgerufenen Fahrzeuge errechnet und für unterschiedliche Zeiträume genannt. Für den Zeitraum 2006-2015 ergeben sich in den USA hierbei 10,3 % bis 10,9 % [30, S. 121].
Im EU-Raum ergibt die Abschätzung durch softwarebedingte und nicht eindeutig zuordenbare Rückrufe 10,4 % [30, S. 110].
Aufgrund der veralteten Datengrundlage der Arbeit (Stand 2015) ist die Datenauswertung obsolet und eignet sich nur zur groben Plausibilisierung der Ergebnisse und einem historischen Abgleich dieser. Insbesondere durch die starke Zunahme des Software-Anteils im Fahrzeug in den letzten Jahren werden aktuelle Daten benötigt. Außerdem wird nicht zwischen der Anzahl der Rückrufe und der Anzahl der betroffenen Fahrzeuge unterschieden. Diese Unterscheidung ist im weiteren Verlauf der Arbeit für abschließende Aussagen noch von großer Bedeutung.

3.2. Auswahl geeigneter Machine Learning-Verfahren

Wie zuvor in der Arbeit erwähnt, existieren unzählige Arten des Machine Learning. Um im weiteren Verlauf plausible Ergebnisse effizient produzieren zu können, müssen die vielversprechendsten Methoden anhand von klar definierten Entscheidungskriterien gewählt werden.

Ein für die Entscheidungsfindung bekanntes Hilfsmittel ist hierfür eine gewichtete Entscheidungsmatrix [31, S. 100-101]. Zu Beginn werden Anforderungskriterien aufgestellt und mit einer Paarvergleichsmatrix gegeneinander gewichtet. Mithilfe der gewählten Kriterien und deren Gewichtung kann dann die Entscheidungsmatrix erstellt werden.

In der Paarvergleichsmatrix werden in der jeweiligen Zelle immer die Zeile mit der Spalte verglichen. Ist die Zeile wichtiger als die Spalte, wird eine 2 eingetragen. Ist die Spalte wichtiger als die Zeile, wird eine 0 eingetragen. Sollten beide gleich relevant sein, wird eine 1 eingetragen. Zum Schluss werden die Beträge der Zeilen summiert und vermerkt. Die Summe jeder Zeile ist auch die Gewichtung des Anforderungskriteriums.

Bei der hier verwendeten modifizierten Entscheidungsmatrix wird eine Bewertungsskala der einzelnen Methoden anhand der Anforderungskriterien von 0 bis 4 gewählt, wobei 0 das schlechteste Ergebnis und 4 das Idealergebnis ist. Die Bewertungen in den jeweiligen Zeilen werden dann mit den zuvor getroffenen Gewichtungen multipliziert und nach der spaltenweisen Multiplikation summiert. Diese Summe stellt die Produktsumme (PS) und somit die Grundlage für den Technischen Wert dar. Das Verhältnis aus der Division in der Gleichung (1) liefert dann für jeden der Lösungsansätze einen technischen Wert TW im Bereich 0 bis 1, gerundet auf zwei Nachkommastellen.

$$TW = \frac{TW_n}{TW_{ideal}} \tag{1}$$

Hierbei entspricht der Wert 1 der idealen und 0 der schlechtesten Lösung. Anhand des technischen Wertes erhalten die Lösungen einen Rang und können mithilfe dessen miteinander verglichen werden.

In dem Auswahlverfahren werden unterschiedliche Methoden des Machine Learning diskutiert. Ziel ist es, am Schluss etwa drei Methoden zur Auswahl zu haben. Mit diesem Verfahren können dann Ergebnisse produziert und auch miteinander verglichen werden.

Folgende Auswahlkriterien für die Paarvergleichsmatrix in Tabelle 1 werden gewählt:

A: Präzision des Verfahrens (im gewählten Problem)
B: Interpretierbarkeit (Blackbox-Effekt)
C: Aufwand Vorarbeit (Labeln der Daten)
D: Ressourcenaufwand (Dauer der Klassifikation)
E: Benutzerfreundlichkeit (Programmierkomplexität)

-	A	B	C	D	E	**Summe**
A	-	2	2	2	2	8
B	0	-	0	1	1	2
C	0	2	-	1	1	4
D	0	1	1	-	1	3
E	0	1	1	1	-	3

Tabelle 1: Paarvergleichsmatrix Anforderungskriterien ML-Verfahren

Mit den gewichteten Kriterien, wird nun in Tabelle 2 die Entscheidungsmatrix aufgestellt.

Die folgenden sechs ML-Verfahren werden miteinander verglichen und bewertet:

1. Clustering-Analyse
2. Assoziationsanalyse
3. Klassifikationsverfahren

4. Regressionsverfahren
5. Reinforcement Learning
6. Neuronale Netze

Methode	A	B	C	D	E	*PS*	*TW*	**Rang**
Gewichtung	8	2	4	3	3	-	-	-
1. Clustering-Analyse	1	2	4	4	4	52	0,65	**2.**
2. Assoziationsanalyse	0	2	4	4	2	38	0,48	**4.**
3. Klassifikationsverfahren	4	2	1	3	3	58	0,73	**1.**
4. Regressionsverfahren	1	3	1	3	3	36	0,45	**5.**
5. Reinforcement Learning	2	1	1	1	1	28	0,35	**6.**
6. Neuronale Netze	3	1	1	2	1	39	0,49	**3.**
Ideal	4	4	4	4	4	80	1,0	-

Tabelle 2: Entscheidungsmatrix ML-Verfahren

Die Entscheidungsmatrix liefert die folgenden drei Verfahren als am besten geeignet für das zu bearbeitende Problem:

1. Klassifikationsverfahren (überwachtes Lernen)
2. Clustering-Analyse (unüberwachtes Lernen)
3. Neuronale Netze (überwachtes Lernen)

Diese Verfahren werden ausgewählt und im weiteren Verlauf verwendet, um Kategorisierungen der Fahrzeugrückrufe vorzunehmen.
Aufgrund dessen, dass auch überwachte Lernmethoden ausgewählt werden und hierfür gelabelte Daten benötigt werden, bedarf es bei der Datenaufbereitung eines zusätzlichen Schritts, um einen Teil der Daten zu labeln.

4. Deskriptive Datenanalyse und -aufbereitung

4.1. Datendeskription

Die gebündelten Datensätze der Fahrzeugrückrufe sind direkt auf den Internetseiten von RAPEX [32] und NHTSA [28] beziehbar. Für den weiteren Verlauf werden die Daten vom Stand 08.05.2023 (RAPEX) und 02.06.2023 (NHTSA) bezogen und verarbeitet.

4.1.1. Deskription RAPEX

RAPEX ermöglicht es, nach Kategorien, Ländern, Risikoarten etc. zu filtern und die Ergebnisse gebündelt als Excel-Datei (xlsx) oder als XML-Datei ausgeben zu lassen [32]. Gesammelt entsteht somit eine Datei mit 5064 Einträgen – gefüllt mit Fahrzeugrückrufen. Unter anderem werden folgende, für diese Arbeit relevante Informationen zu den jeweiligen Rückrufen vermerkt:

- Alert number: Eindeutig identifizierende Zahlenkombination für jede Rückrufkampagne
- Alert submitted by: Das meldende Land
- Country of origin: Produktionsland
- Product; Name: Produktname
- Description: Beschreibung des Produkts
- Brand: Hersteller
- Risk legal provision/Risk: Beschreibung des Risikos und der Ursache
- etc…

Allerdings werden – trotz Filterung der Daten nach der Kategorie Motor vehicles – auch Rückrufe von Helmen, Minibikes, Anhängern, Schneemobilen und Fahrrädern vermerkt. Diese sind bei der Analyse nicht erwünscht und werden automatisiert herausgefiltert. Nach einer anfänglichen Filterung dieser Stichwörter bleiben nur noch 4966 Einträge übrig.
Da die RAPEX-Daten erst seit dem Jahr 2005 gesammelt werden, muss hier keine Filterung der Rückrufjahre vorgenommen werden.
In der RAPEX-Datenbank sind alle betroffenen Fahrzeugmodelle eines Rückrufes schon zusammengefasst. Darin sind keine Doppelungen enthalten.

Für die Bearbeitung der Daten mithilfe der ML-Algorithmen wird das XML-Format gewählt, da dieses von Python im Import besser verarbeitet werden kann. Des Weiteren besteht der Vorteil, dass es für den Download keine Beschränkung auf 5000 Rückrufe gibt. Denn diese Beschränkung besteht für den Download der xlsx-Datei.

4.1.2. Deskription NHTSA

Die durch NHTSA bereitgestellten Daten sind direkt gebündelt und ohne Einschränkung der Anzahl von Rückrufen beziehbar. Jedoch handelt es sich hierbei um eine Tabelle, die als Text-Datei (FLAT_RCL.txt) gespeichert wurde. Die Text-Datei (txt) wird mithilfe von Excel in eine Excel-Datei (xlsx) zurückgewandelt, um sie lesbarer zu machen. Bei der Verarbeitung in Python wird die Text-Datei (txt) direkt mithilfe der Pandas-Bibliothek als dataframe importiert [33].
Die Beschreibung der einzelnen Datenbankspalten findet sich nicht direkt in der Datenbank, sondern ist einzeln in einer separaten Text-Datei (txt) zu finden (RCL.txt). Aufgrund der Spezialisierung der NHTSA-Datenbank auf Fahrzeuge, finden sich hier Informationen, die sich genauer auf Fahrzeugrückrufe beziehen. In der Datenbank werden folgende relevante Informationen zu den jeweiligen Rückrufen vermerkt:

- CAMPNO: Eindeutig identifizierende Buchstaben- und Zahlenkombination für jede Rückrufkampagne. Die ersten zwei Stellen stellen das Jahr da (beispielsweise 14 für 2014.)
- MAKETXT: Fahrzeughersteller
- MODELTXT: Fahrzeugmodell
- YEARTXT: Modelljahr des Fahrzeuges (Das Modelljahr entspricht in der Bedeutung der Fahrzeugversion und muss nicht gleich dem Fahrzeugbaujahr sein.)
- COMPNAME: Kategorie des fehlerhaften Bauteils
- RCLTYPECD: Grobe Einteilung des Rückrufs (V = Vehicle, I = Improvement, E = Equipment, T = Tire, C = Children (child seats))
- POTAFF: Anzahl der betroffenen Fahrzeuge
- DESC_DEFECT: Zusammenfassung der Fehlerursache
- CONSEQUENCE_DEFECT: Zusammenfassung der Fehlerfolge
- CORRECTIVE_ACTION: Zusammenfassung der durchzuführenden Maßnahme
- etc…

Im Gegensatz zur RAPEX-Datenbank wird bei den NHTSA-Daten für jedes betroffene Fahrzeugmodell und auch Modelljahr einer Rückrufkampagne ein neuer Eintrag gesetzt. Aufgrund dessen treten innerhalb einer Rückrufkampagne häufig Mehrfacheinträge in den Rückrufen auf. Beispielsweise der Rückruf 08V397000 [34] besitzt acht Einträge, da er die Modelljahre 1996 bis 2003 des Audi A8 betrifft und für jedes Modelljahr ein neuer Eintrag gesetzt wird.

Aufgrund solcher Mehrfacheintragungen von einer Rückrufkampagne beläuft sich die Anzahl der Einträge in den unbereinigten Daten der NHTSA auf 241.945.
Durch eine Entfernung der mehrfach vermerkten Einträge der Datenbank verbleiben noch 27.025 Einträge.
Die Spalte RCLTYPECD teilt die Rückrufe direkt auf unterschiedliche grobe Kategorien auf. Da für die Auswertung beispielsweise Kindersitze oder Reifen irrelevant sind, werden nur Rückrufe mit den Buchstaben V, E oder I beibehalten. Nach dieser Filterung bleiben noch 26.032 Rückrufe bestehen.
Nach einer Filterung von Rückrufen nach dem Jahr 2000 verbleiben nur noch 15.268 Rückrufe in den Daten. Diese Filterung ist nötig, da vor dem Modelljahr 2000 Fahrzeuge kaum relevante Software besaßen. Demzufolge würden sie die Gesamtstatistik lediglich verfälschen.
Unter die 15.268 Rückrufe fallen aber auch noch Rückrufe von Helmen, Schneemobilen, Anhängern, Fahrrädern und Minibikes. Da diese für die Auswertung irrelevant sind, werden sie entfernt. Nach dieser Bereinigung der Daten bleiben noch 13.415 Einträge übrig.
Für die Bearbeitung der Daten mithilfe der ML-Algorithmen wird das bereitgestellte Text-Format gewählt, da dieses von den genutzten Python-Tools gut verarbeitet werden kann. Es wird auch als einziges durch NHTSA zur Verfügung gestellt.

4.2. Datenaufbereitung

Um von ML-Algorithmen sinnvoll und effizient verarbeitet werden zu können, müssen die Daten aus den Datenbanken vorab aufbereitet werden.
Hierfür werden vier Schritte vorgenommen, um die Textdaten vorzubereiten: Rauschentfernung/Datenbereinigung, Tokenisierung, Normalisierung und Vektorisierung [35]. Es ist zu beachten, dass bei komplexeren Verfahren wie CNN weitere Aufbereitungsschritte wie zum Beispiel Padding (4.2.6) erforderlich sind (vgl. Abbildung 21).

4.2.1. Rauschentfernung/Datenbereinigung

Die Rauschentfernung wird zumeist als erster Schritt vorgenommen. Im Zuge dessen werden Daten wie Überschriften, Fußnoten, HTML/XML-Dateireste entfernt [35]. Dadurch wird unnötiges Rauschen beseitigt, sodass nur noch ein reiner Text verbleibt. Hierfür wird die Python Bibliothek Pandas [33] und die darin enthaltene Möglichkeit für den Import verwendet.

Ein zur Datenbereinigung relevanter Vorgang ist die Entfernung von falschen, doppelt aufgeführten oder irrelevanten Rückrufen.
In der Datenbank von RAPEX werden alle für die weitere Analyse irrelevanten Spalten (Informationskategorien) entfernt. Als möglicherweise relevant werden die in 4.1.1 genannten Kategorien festgelegt und der Rest mithilfe eines Python Skriptes bei dem Import der Daten direkt entfernt.
Zudem werden – wie in 4.1.1 beschrieben – auch alle Rückrufe entfernt, die nur indirekt Fahrzeuge betreffen. Dies geschieht mithilfe eines Python-Skriptes, das die Zeilen jeweils vollständig nach den Stichwörtern durchsucht. Wenn eines der genannten Wörter auftaucht, wird die Zeile entfernt.

Die Daten der NHTSA werden identisch aufbereitet. Die in 4.1.2 genannten Informationskategorien werden gewählt und die restlichen Informationen entfernt. Rückrufe, die irrelevante Themengebiete beinhalten, werden ebenso entfernt.
Als zusätzlicher Schritt im Gegensatz zur RAPEX-Datenbank müssen bei den NHTSA-Daten auch doppelte Einträge entfernt werden. Diese können anhand der

NHTSA Campaign Number einfach identifiziert werden. Da die Mehrfacheintragungen bis auf Marke, Modell und Modelljahr vollständig identisch sind, können diese jeweils immer zu einer Zeile reduziert werden. Dies wird so umgesetzt, dass bei Bestehen von mehreren Einträgen zu einer Nummer, diese einfach entfernt werden und nur der erste Eintrag bestehen bleibt.
Zudem ist es – wie in 4.1.2 beschrieben – bei den NHTSA-Daten sinnvoll, das Erscheinungsdatum der Rückrufe zu filtern. Infolgedessen werden nur Rückrufe berücksichtigt, die seit dem Jahr 2000 veröffentlicht wurden.

4.2.2. Tokenisierung

Die Tokenisierung wird durchgeführt, um lange Texte in kleinere Teile umzuwandeln. Hierbei können lange Texte zu einzelnen Sätzen umgewandelt werden oder auch Sätze in einzelne Worte [35]. Im angewendeten Fall werden die Sätze mithilfe der Open Source Software „Natural Language Toolkit“ [36] innerhalb der Python-Umgebung tokenisiert.
Im angewendeten Beispiel ist die zweite Variante, also Sätze zu Worten umwandeln, die sinnvollere Auswahl. Somit würde aus
[‘Fahrzeugrückrufe sind ein hochrelevantes Thema für Fahrzeughersteller.‘]
[‘Fahrzeugrückrufe‘, ‘sind‘, ‘ein‘, ‘hochrelevantes‘, ‘Thema‘, ‘für‘, ‘Fahrzeughersteller‘, ‚.‘] werden.
Die Tokenisierung stellt sich als sinnvoll dar, um beispielsweise spezifische Muster innerhalb von Texten zu erkennen oder Worte zu zählen.

4.2.3. Normalisierung

Durch die Normalisierung wird eine Homogenität der Struktur der unterschiedlichen Wörter erreicht. Dazu gehören die Umwandlung aller Buchstaben in Klein- oder Großbuchstaben, Entfernung von Satzzeichen, Umwandlung von Zahlen in Worte, etc. [35]. Auch die Entfernung von Füllwörtern, die für den eigentlichen Inhalt nicht von Relevanz sind, ist Teil der Normalisierung.
Nach dem Schritt der Normalisierung würde der oben genannte Satz normalisiert nun nur noch aus einzelnen Worten, also Tokens, bestehen und wie folgt aussehen:
[‘fahrzeugrückrufe‘, ‘hochrelevantes‘, ‘thema‘, ‘fahrzeughersteller‘].

4.2.4. Vektorisierung – Variante für Clustering

Als letzter Schritt der Datenaufbereitung für die Machine Learning-Modelle wird eine Vektorisierung der Texte vorgenommen. Hierbei wird aus den Tokens ein Vektor erstellt, der für jeden Rückruf das Vorkommen und optional auch die Häufigkeit eines Wortes darstellt. Insbesondere für die Anwendung von textbasierten Machine Learning-Modellen wird häufig die Textvektorisierung verwendet. Am eingeführten Beispiel würde nun aus dem normalisierten Satz der in Tabelle 3 abgebildete Vektor entstehen.

	fahrzeugrückrufe	**hochrelevantes**	**thema**	**fahrzeughersteller**
1	1	1	1	1

Tabelle 3: Vektorisierter Beispielssatz

Wenn nun ein zweiter Satz hinzukommt, wird bei der Vektorisierung geprüft, ob die darin enthaltenen Wörter schon in der erstellten Matrix enthalten sind. Ist dies der Fall, wird in der zweiten Zeile nun in der entsprechenden Spalte eine 1 eingetragen. Ist das Wort noch nicht enthalten, wird eine neue Spalte hinzugefügt. Beispielhaft ist dies in Tabelle 4 aufgeführt. Als zweiter, schon bereinigter Beispielsatz wird ***[‘software‘, ‘bedeutendes‘, ‘thema‘, ‘fahrzeughersteller‘]*** verwendet.

	fahrzeug-rückrufe	**hochrele-vantes**	**thema**	**fahrzeug-hersteller**	**software**	**bedeuten-des**
1	1	1	1	1	0	0
2	0	0	1	1	1	1

Tabelle 4: Vektorisierte Beispielssätze

Ein bedeutender Grund für die Anwendung der Vektorisierung besteht darin, dass bestimmte Modelle wie Neuronale Netzwerke eine einheitliche Eingabeform erfordern, bei der alle Daten die gleiche Länge und Anzahl von Dimensionen haben müssen. Dies ist eine grundlegende Voraussetzung für die Verwendung solcher Modelle. Die Vektorisierung ermöglicht dies, indem sie die Daten automatisch in

Vektoren umwandelt, wobei jeder Vektor die Anzahl der individuellen Wörter im gesamten Datensatz als Spalten enthält.

4.2.5. Aufteilung der Daten

Bei dem zweiten Teil der gewählten Verfahren handelt es sich um überwachte Lernmethoden. Insbesondere liegt hier der große Unterschied in der Vorarbeit, dem Labeln der Daten. Dies ist ein manueller Schritt, der im Voraus bei einer zufällig ausgewählten Anzahl von Daten geschehen muss. Vorab müssen die Daten in drei Teile aufgeteilt werden [37]:

1. Trainingsdaten, um das Modell für den vorgesehenen Einsatzzweck zu trainieren. Diese Daten müssen vorab gelabelt werden.
2. Validierungsdaten, um festzustellen, mit welcher Genauigkeit das trainierte Modell arbeitet und eventuell auch Feinabstimmungen des Modells vornehmen zu können. Auch diese Daten müssen vorab gelabelt werden.
3. Testdaten, um mit dem trainierten Modell arbeiten zu können und erproben zu können, ob es sinnvolle Kategorisierungen trifft. Dabei handelt es um dem Modell unbekannte Daten, die keine Label besitzen.

Die Aufteilung und das genaue Verhältnis der Daten variiert je nach Anzahl der Daten und verwendetem Modell [37]. Im Folgenden werden die Daten in einem Verhältnis von etwa 60 % Trainingsdaten, 10 % Validierungsdaten und 30 % Testdaten aufgeteilt. Diese Aufteilung ergibt sich aus einer durchgeführten empirischen Auswertung der Genauigkeit.
Folglich werden 70 % der Daten vorab gelabelt und mit diesen die Modelle der überwachten Lernverfahren trainiert.

4.2.6. Padding

Die Anwendung von Padding ist insbesondere bei der Verwendung von Modellen wie Convolutional Neural Networks von großer Bedeutung. Wie auch bei der Vektorisierung besteht eine der Hauptanwendungen von Padding darin, sicherzustellen, dass Daten mit variabler Länge auf eine einheitliche Länge und Anzahl von

Dimensionen gebracht werden, um sie effektiv in den Modellen zu verarbeiten [38]. Somit ist Padding als eine Art Alternative zur Vektorisierung zu sehen.

Das Padding bedarf einer angepassten Tokenisierung. Hierbei wird auf den Tokenizer der TensorFlow Keras Bibliothek [39] zurückgegriffen. Dieser wird im Unterschied zu der Tokenisierung in 4.2.2 anhand aller vorhandenen Wörter im Trainingsset initialisiert und erstellt mit diesen Wörtern ein Vokabular.

Um das Vorgehen an den vorangegangenen, schon normalisierten Beispielsätzen ***[‘fahrzeugrückrufe‘, ‘hochrelevantes‘, ‘thema‘, ‘fahrzeughersteller‘]*** und ***[‘software‘, ‘thema‘, ‘fahrzeughersteller‘]*** zu erklären, ist in Tabelle 5 das erstellte Vokabular dargestellt. Um das Padding an sich besser darzustellen, wurde der zweite Satz im Unterschied zum vorherigen Beispiel um das Wort „bedeutendes“ gekürzt.

1.	**thema**
2.	**fahrzeughersteller**
3.	**fahrzeugrückrufe**
4.	**hochrelevantes**
5.	**software**

Tabelle 5: Vokabular Beispielsätze

Jedem Wort wird eine Ganzzahl zugewiesen und diese nach der Häufigkeit des Vorkommens des Wortes sortiert. Da in den zwei Beispielssätzen „thema“ und „fahrzeughersteller“ zweifach vorkamen, liegen diese auf den ersten zwei Einträgen. Der Rest der Wörter kam nur einmalig vor und belegt somit die restlichen drei Einträge.

Im nächsten Schritt wird jeder Satz anhand des erstellten Vokabulars in eine Sequenz von Ganzzahlen gewandelt. Fertiggestellt sind die Ganzzahlsequenzen in Tabelle 6 dargestellt.

1.	**[3, 4, 1, 2]**
2.	**[5, 1, 2]**

Tabelle 6: Tokenisierte Beispielsätze (Padding)

Da für CNN alle eingespeisten Sequenzen die gleiche Länge haben müssen, werden die Sequenzen noch gepaddet. Hierbei werden alle Sequenzen mit Nullen aufgefüllt, sodass die Sequenzen alle die Länge der längsten Sequenz des Datensatzes besitzen. Die Auffüllung mit Nullen kann mithilfe eines Parameters entweder von vorne (Pre-Padding) oder von hinten (Post-Padding) geschehen [40]. Da für CNN Pre-Padding besser geeignet ist [41], wird dieses auch im Folgenden verwendet.

Im angeführten Beispiel wäre der erste Satz mit vier Ganzzahlen der längste Satz. Dementsprechend wird der zweite Satz nun in Tabelle 7 mithilfe einer 0 aufgefüllt, sodass beide Ganzzahlsequenzen gleich lang sind.

1.	**[3, 4, 1, 2]**
2.	**[0, 5, 1, 2]**

Tabelle 7: Gepaddete Beispielsätze

5. Machine Learning-Prozess und Kategorisierung

5.1. Clustering-Analyse

Als erstes der drei anzuwendenden Verfahren ist es sinnvoll, das unüberwachte Clustering zu wählen. Für das Clustering bedarf es sehr wenig Vorarbeit und es können eventuell unentdeckte Strukturen in den Datensätzen entdeckt werden.

Um das Clustering durchzuführen, wird auf die bekannteste Methode, das k-means-Clustering [42], zurückgegriffen. Dabei handelt es sich rechnerisch um einen einfachen Algorithmus, der die euklidische Distanz zwischen den Datenpunkten berechnet. Hierbei wird versucht, innerhalb von Clustern eine kleinstmögliche Distanz zwischen den Datenpunkten (Intragruppen-Homogenität) und gleichzeitig zwischen den Datenpunkten unterschiedlicher Cluster eine größtmögliche Distanz (Intergruppen-Heterogenität) herzustellen [43, S. 490].
Vorab müssen die einzelnen Parameter für das Clustering festgelegt werden. Darunter fallen zum Beispiel die Anzahl der Cluster k, die Clustering-Methode, die maximale Anzahl an Iterationen und die Anzahl der zufälligen Initialisierungen des Algorithmus.
K-means ist ein einfacher und somit schneller Clustering-Algorithmus: Durch Bilden von zufälligen Cluster-Zentroiden, deren Anzahl aber fest vorgegeben sein muss, und Messen der euklidischen Distanz der Datenpunkte zu den Clustern weist er die Datenpunkte den Clustern zu.
Der erste Schritt des Prozesses ist in einer zweidimensionalen Darstellung in Abbildung 10 dargestellt. Hierbei werden bei der ersten Iteration die Zentroiden zufällig in den Raum gelegt.

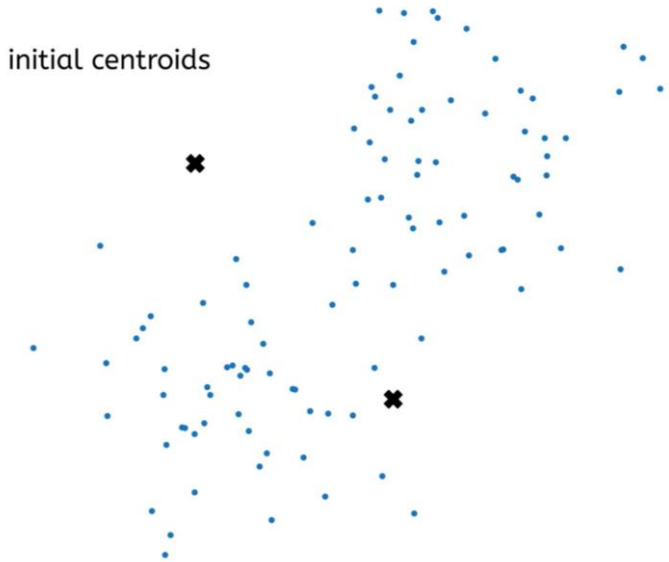

Abbildung 10: Erste Initialisierung k-means [44]

Im zweiten Schritt werden die Abstände von allen Datenpunkten zu den jeweiligen Zentroiden errechnet. Jeder der Datenpunkte wird dann auf Basis der Abstände zu den Zentroiden einem von den beiden zugeordnet. Grafisch ist diese Aufteilung in Abbildung 11 in rot und grün dargestellt. Auf der Grafik lässt sich zur Vereinfachung auch eine sogenannte Entscheidungsgrenze ziehen, um die optische Zuordnung zu vereinfachen.

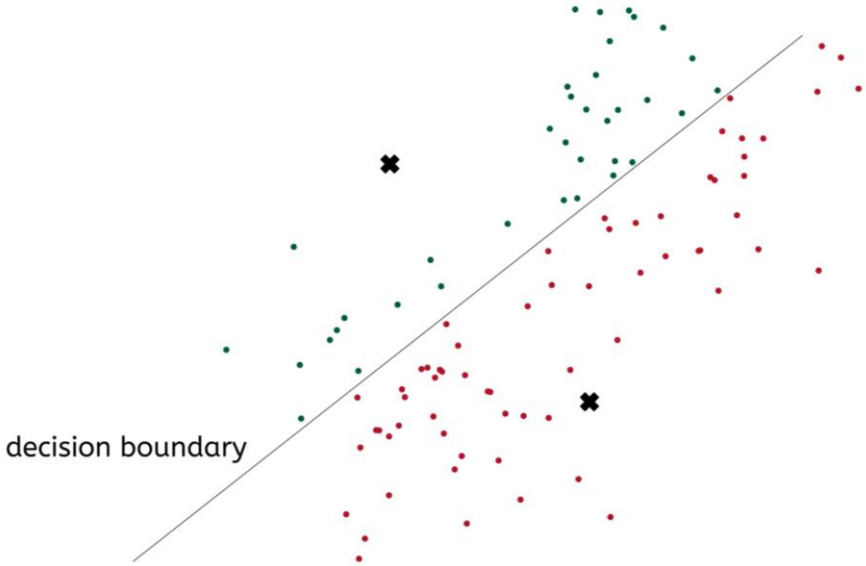

Abbildung 11: Bilden der Entscheidungsgrenze k-means [44]

Der nächste Schritt ist nun eine mathematische Bestimmung des echten Mittelpunktes der kategorisierten Cluster. An diesem neu bestimmten Mittelpunkt in Abbildung 12 wird nun das jeweilige Cluster-Zentroid verschoben.

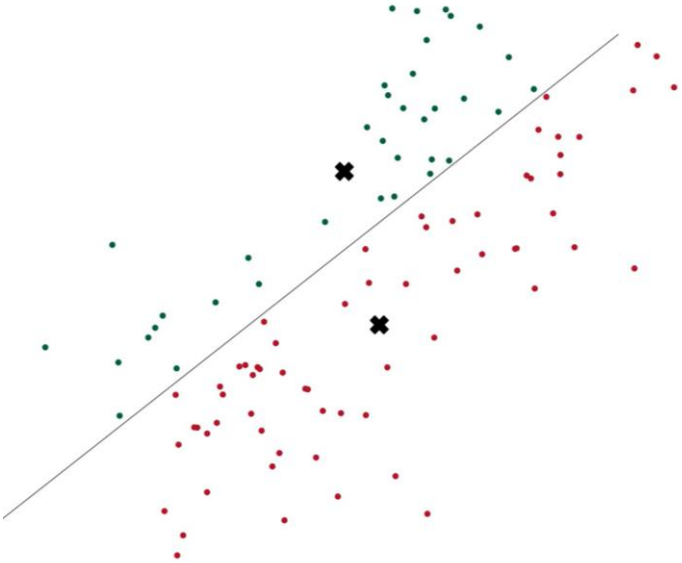

Abbildung 12: Verschieben der Zentroiden zum Mittelpunkt k-means [44]

Der zweite und dritte Schritt werden je nach Parametrierung unterschiedlich oft iteriert, wobei beim nächsten Durchlauf der Mittelwert des jeweiligen Clusters als neues Cluster-Zentroid gewählt wird [45]. Im Idealfall sollte so lange iteriert werden, bis sich eine Konvergenz einstellt und die Ergebnisse sich trotz weiterer Iterationen nicht verbessern. In Abbildung 13 ist ein weiterer Zwischenschritt und letztendlich das Endergebnis dargestellt.

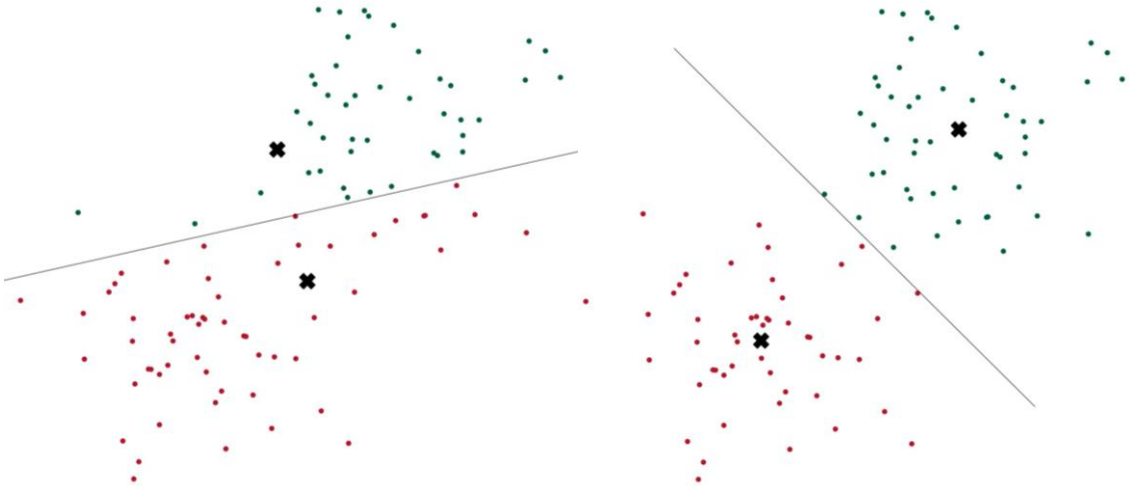

Abbildung 13: Zwischenschritt Iterationen und Endergebnis k-means [44]

Der k-means++ Algorithmus ist vom allgemeinen Prinzip identisch zum k-means. Einziger Unterschied ist, dass dieser bei der ersten Initialisierung nur den ersten Zentroid zufällig im Raum wählt. Die restlichen Zentroiden werden auf Basis der Abstände der Datenpunkte zum ersten Zentroid gewählt [46]. Somit werden die Zentroiden direkt am Anfang breit im Raum verteilt. Dadurch wird der Algorithmus effizienter, da er schneller die gewünschte Konvergenz erreicht.

Die zugewiesenen Cluster werden in einem Ergebnis-dataframe abgespeichert und können im Anschluss grafisch dargestellt werden. Auch ist es möglich, eine Anzahl von relevanten Featuremarks, also beispielhaft gewählte Stichworte, aus den jeweiligen Clustern anzuzeigen.

Sind die Ergebnisse nicht zufriedenstellend, können über die Hyperparameter Feinanpassungen vorgenommen werden.

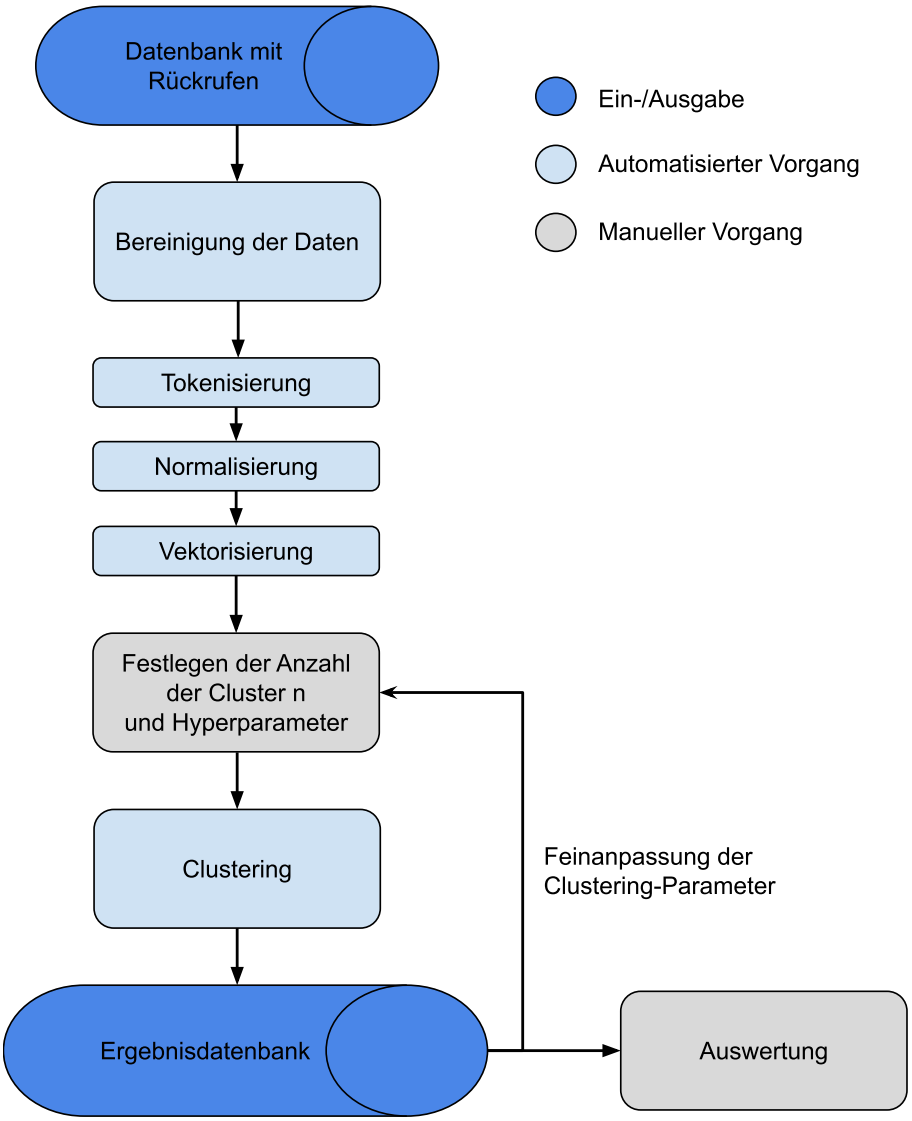

Abbildung 14: Flussdiagramm Clustering-Analyse

Der beschriebene Ablauf ist der Übersichtlichkeit wegen in Abbildung 14 dargestellt. Zusätzlich zur internen Abspeicherung innerhalb der Python-Umgebung wird das Ergebnis-dataframe auch als Excel-Datei abgespeichert, um dort die Clustering Ergebnisse in einer übersichtlichen Form darstellen und analysieren zu können.

Nach einer praktischen Umsetzung des Clustering-Verfahrens mit k-means werden allerdings ohne Weiteres keine für die Arbeit sinnvollen Ergebnisse erreicht. Dies liegt daran, dass zwar die Rückrufe in unterschiedliche Kategorien eingeteilt werden, aber keine Kategorie explizit mit Software-Fehlern zusammenhängt. Denn die Kategorisierung erfolgt rein auf der euklidischen Distanz der vektorisierten Texte.

Eine verbesserte Methode, um die Bedeutung von Software-Fehlern durch das Machine Learning-Verfahren besser deuten zu können, ist eine Gewichtung von software-relevanten Stichworten.
Die Länge der Vektoren der Rückrufe entspricht der Summe aller individuellen Wörter, die im Datensatz vorkommen. Im Anschluss wird jeder Dateneintrag Wort für Wort auf dessen Vorkommen geprüft. Ist das dem jeweiligen Index zugeordnete Word im Rückruftext enthalten, wird im Vektor eine 1 gesetzt. Kommt es nicht darin vor, bleibt eine 0 bestehen. Dies wird zeilenweise für jeden Rückruf durchgeführt und für jede der Zeilen ein solcher Vektor gebildet. Im Anschluss werden alle Rückrufe – bestehend aus den vektorisierten Texten – als Matrix gespeichert, um sie dann zeilenweise clustern zu können.
Das Ziel der Arbeit ist es nicht unterschiedliche Rückrufe zu kategorisieren, sondern lediglich zwei Gruppen zu bilden: softwarebedingte Rückrufe und nicht softwarebedingte beziehungsweise hardwarebedingte Rückrufe.
Mit der Annahme, dass beispielsweise Begriffe wie „software“, „update“, „memory“, „algorithm“, „version“ oder „calibration“ auf jeden Fall in Rückrufen vorkommen, die softwarebedingt sind, wird die aus den einzelnen Vektoren bestehende Matrix neu gewichtet. Es können auch andere, mehr oder weniger Begriffe verwendet werden. Nach praktischer Umsetzung und stichprobenartiger manueller Sichtung von Rückrufen ergaben sich die genannten Begriffe allerdings als sinnvolle Auswahl.

Die Begriffe finden sich nun in der Rückruf-Matrix als Spalten wieder. Ist in der Spalte eine 1 vermerkt, taucht der Begriff in der jeweiligen Zeile und somit im Rückruf auf. Dementsprechend ist die Wahrscheinlichkeit sehr hoch, dass es sich um ein Software-Problem handelt.
Die Grundidee ist, dass diese Zeile stärker gewichtet werden soll, um somit in einer vieldimensionalen Darstellung der Matrix einen stärkeren Ausschlag des jeweiligen Vektors. Folglich wird eine größere euklidische Distanz zwischen softwarebedingten und nicht-softwarebedingten Rückrufen erreicht. Dadurch soll dem Algorithmus ein sinnvolles Kategorisieren ermöglicht werden.
Die geänderte Gewichtung wird durch Spaltenmultiplikationen innerhalb der Matrix erreicht. Diese betrifft die Spalten, deren zugewiesenes Vokabular auch dem innerhalb des Wörterbuches entspricht. Wichtig ist es, die Anzahl der Cluster K auf 2 zu reduzieren, um die genau zwei gewünschten Kategorien zu erreichen. Iterativ wurde festgestellt, dass die beste Genauigkeit mit einer Gewichtung von 10 erreicht wird.
Zur Veranschaulichung werden die zwei vektorisierten Sätze aus Tabelle 4 mit den Stichworten neu gewichtet. Die in Tabelle 8 dargestellte Matrix stellt die nun gewichteten Ergebnisse dar.

	fahrzeug-rückrufe	**hochrele-vantes**	**thema**	**fahrzeug-hersteller**	**software**	**bedeuten-des**
1	**1**	**1**	**1**	**1**	**0**	**0**
2	**0**	**0**	**1**	**1**	**10**	**1**

Tabelle 8: Gewichtete vektorisierte Beispielssätze

Um den Unterschied in der vieldimensionalen Darstellung zu visualisieren, wird die Matrix für die bessere Darstellbarkeit in Tabelle 9 und Tabelle 10 auf die letzten drei Spalten der Tabelle 4 und Tabelle 8 reduziert. Zudem wird der erste Satz rot und der zweite Satz blau eingefärbt. Dieses Farbschema findet sich in den jeweils darauffolgenden Abbildungen wieder.
Abbildung 15 stellt die ungewichtete Variante grafisch dar. Wie hier erkennbar ist, ist die euklidische Distanz nach Gleichung (2) zwischen den zwei Datenpunkten der zwei Sätze recht gering und liegt in dem Fall bei $\sim 1{,}4$.

$$d(qi,\ pi) = \sqrt{\sum_{i=1}^{n} (q_i - pi)^2} \tag{2}$$

	fahrzeughersteller	software	bedeutendes
1	1	0	0
2	1	1	1

Tabelle 9: Reduzierte Darstellung ungewichtete und vektorisierte Beispielssätze

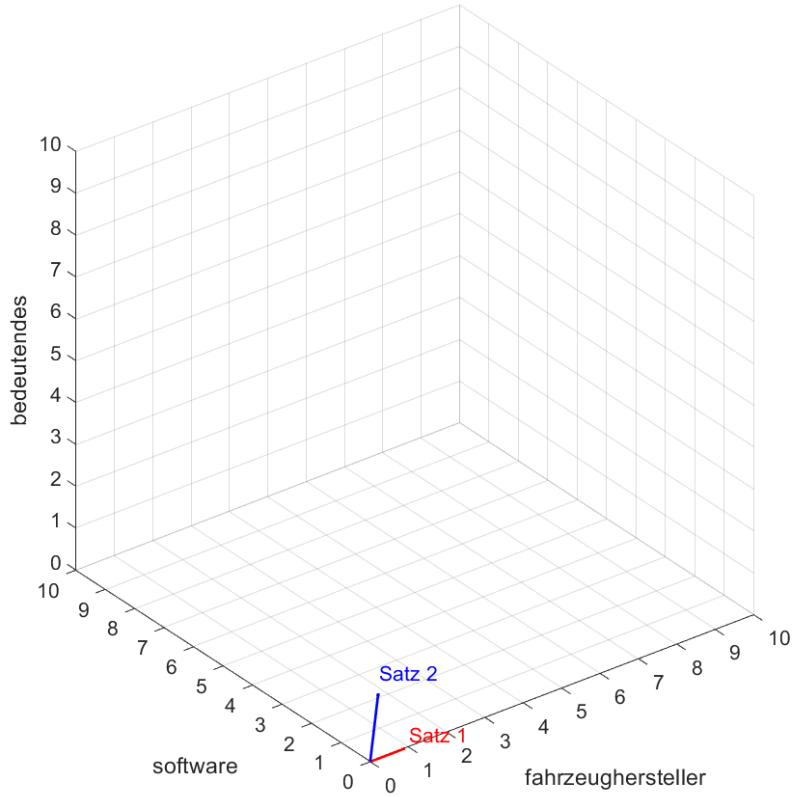

Abbildung 15: Dreidimensionale Darstellung 2 Beispielssätze ungewichtet

Abbildung 16 stellt nun die gewichtete Variante grafisch dar. Wie hier erkennbar ist, ist die euklidische Distanz nach Gleichung (2) zwischen den zwei Datenpunkten der Sätze nun wesentlich höher und liegt in dem Fall bei $\sim 10{,}1$. Somit ist eine Einteilung in eindeutige Cluster wesentlich einfacher.

	fahrzeughersteller	software	bedeutendes
1	1	0	0
2	1	10	1

Tabelle 10: Reduzierte Darstellung gewichtete und vektorisierte Beispielssätze

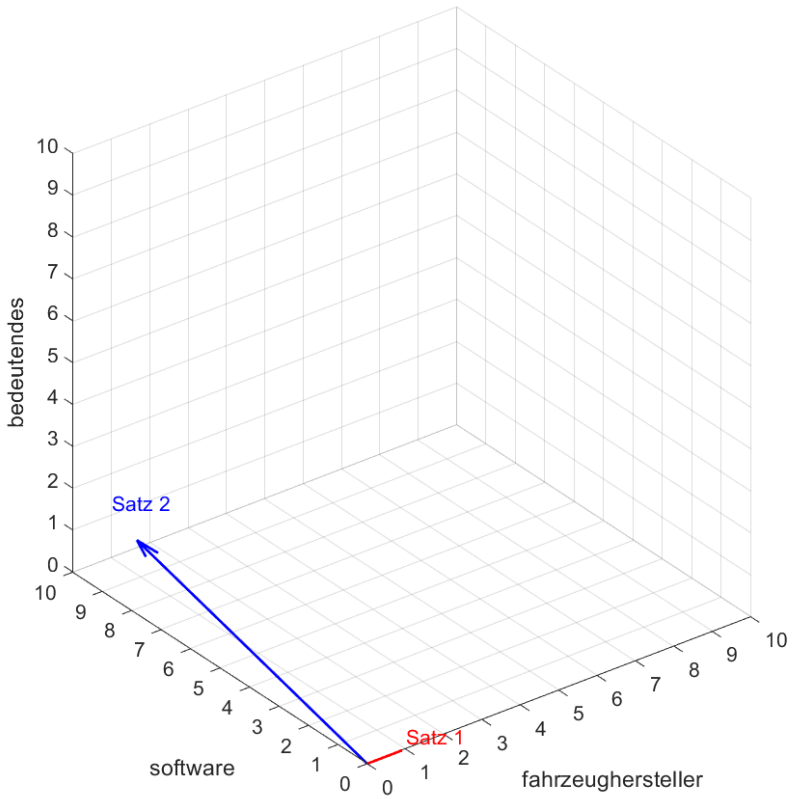

Abbildung 16: Dreidimensionale Darstellung 2 Beispielssätze gewichtet

Noch deutlicher wird der Unterschied, sobald – wie in Abbildung 17 – mehrere Sätze hinzukommen, da hier eine eindeutige Zuordnung zu Clustern ohne Gewichtung schwierig ist. Die sechs Beispielssätze liegen nun alle eng beieinander und es ist keine eindeutige Struktur erkennbar.

	fahrzeughersteller	software	bedeutendes
1	1	0	0
2	1	1	1
3	0	0	1
4	1	0	1
5	1	1	0
6	0	1	1

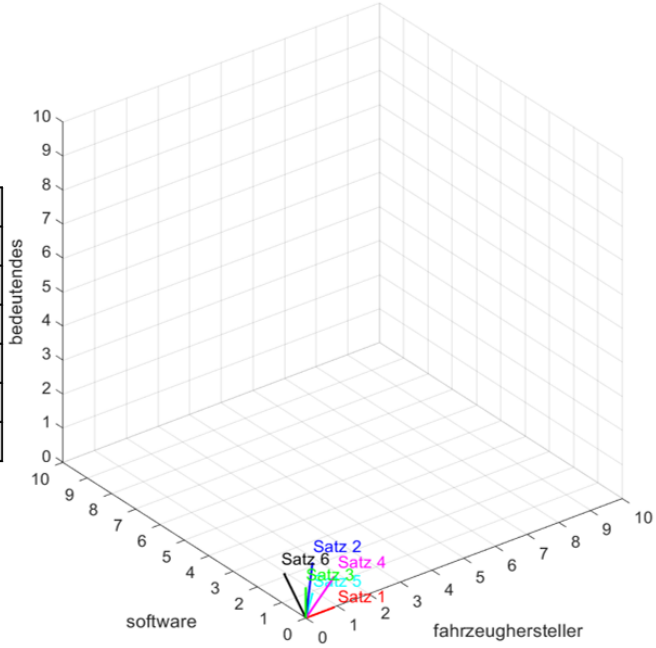

Abbildung 17: Dreidimensionale Darstellung 6 Beispielssätze ungewichtet

In Abbildung 18 sind hingegen die gleichen Beispielssätze wie in Abbildung 17 enthalten, nur sind diese hier gewichtet. In der grafischen Darstellung sind durch die wesentlich erhöhte euklidische Distanz nun eindeutige Cluster – hier in rot und grün dargestellt – erkennbar und zuweisbar.

	fahrzeughersteller	software	bedeutendes
1	1	0	0
2	1	10	1
3	0	0	1
4	1	0	1
5	1	10	0
6	0	10	1

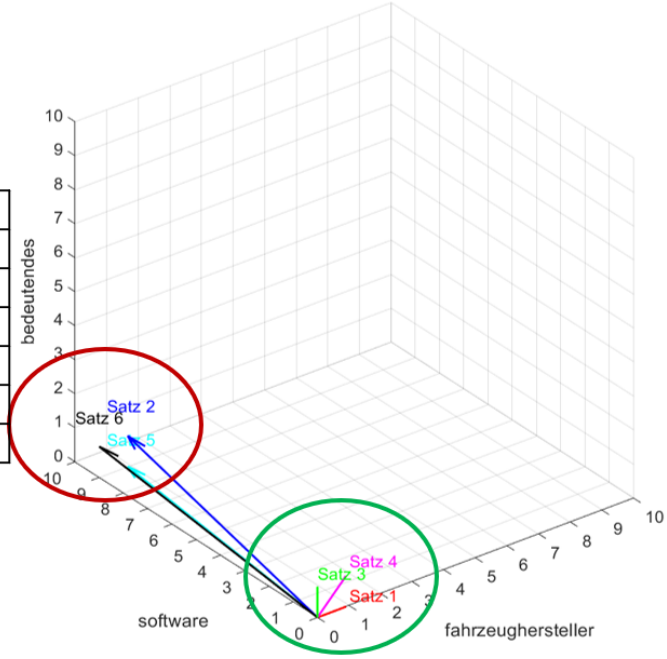

Abbildung 18: Dreidimensionale Darstellung 6 Beispielssätze gewichtet

Mit der gewichteten Methode des k-means-Clustering können auch mit dem unüberwachten Vorgehen gute und präzise Kategorisierungen von Software-Fehlern erreicht werden.
Mithilfe einer manuellen Prüfung einer Stichprobe von 1000 zufällig gewählten, kategorisierten Rückrufen ergibt sich die Konfusionsmatrix in Tabelle 11.

		True Class	
		1	**0**
Predicted Class	**1**	40 (TP)	8 (FP)
	0	0 (FN)	952 (TN)

Tabelle 11: Konfusionsmatrix 1000 Stichproben gewichteter k-means

Auffällig ist, dass bei der verwendeten Methode deutlich mehr False-Positives (Software-Fehler wird angezeigt, obwohl keiner vorliegt) auftreten. In dieser Stichprobe liegt der Anteil der falsch zugeordneten Rückrufe bei 0,8 % in Relation zu der Gesamtanzahl der Stichprobe von 1000. Die Quote von False-Positives in Relation zu den Gesamt-Positives liegt bei etwa 16,67 %.
Der Grund der False-Positives ist, dass beispielsweise der Einbau eines Hardwareteils einer neueren „version“ als Korrekturmaßnahme aufgrund des Vorkommens des Wortes „version“ als Software-Fehler vermerkt wird, obwohl dies nichts mit Software zu tun hat.
False-Negatives (Software-Fehler wird nicht markiert, obwohl ein solcher vorliegt) treten dagegen nicht auf, da bei einem Fehlen der Schlagwörter der Rückruf mit sehr hoher Wahrscheinlichkeit nichts mit Software zu tun hat.

5.2. Klassifikationsverfahren – Decision Tree

Als Algorithmen werden für die überwachten Lernverfahren die zwei geläufigsten Methoden für überwachte Textklassifikation gewählt: Decision Tree Classifier [47] und Random Forest Classifier [48].

Beide entstammen der Scikit-Learn-Bibliothek und basieren auf Entscheidungsbäumen. Die Modelle haben einen leicht abgeänderten Ablauf im Vergleich zum Clustering. Dieser findet sich in Abbildung 19 wieder.

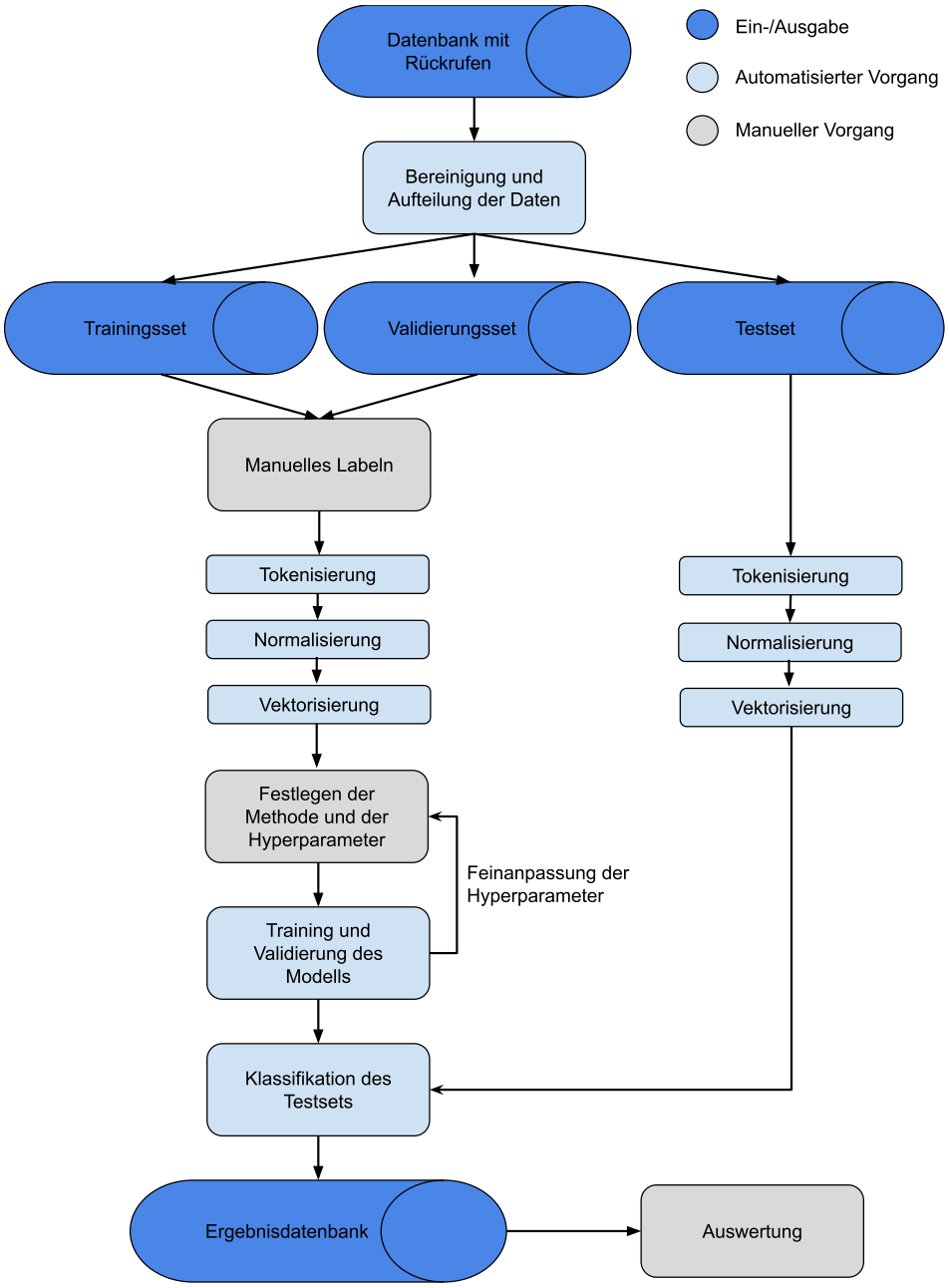

Abbildung 19: Flussdiagramm überwachtes Lernverfahren

Da das Modell schon aufgeteilte (vgl. 4.2.5) und gelabelte Daten benötigt, wird aufgrund der vorab erzielten, guten Ergebnisse das Clustering-Verfahren verwendet um das Trainings- und Validierungsset zu labeln. Weil bei der Clustering-Methode hauptsächlich False-Positives auftreten (vgl. 5.1), reicht es aus, nur die Positives (Software-Fehler erkannt) manuell zu prüfen und zu kategorisieren. Aufgrund der Datenmenge spart diese Methode viel Arbeitsaufwand.
Mit den nun aufgeteilten und gelabelten Daten können die folgenden überwachten Lernverfahren verwendet werden.

Der Random Forest Classifier ist im Prinzip eine Zusammensetzung aus mehreren unterschiedlichen und unabhängigen Decision Tree Classifier [49].
Aufgrund der empirisch ermittelten, deutlich schlechteren Ergebnisqualität des Random Forest Classifier (vgl. 6.1) wird dieser im Folgenden nicht näher betrachtet.
Der Decision Tree ist eine hierarchische Struktur und besteht aus Wurzelknoten, internen Knoten und Blattknoten [50].

Beispielhaft ist in Abbildung 20 der fertig trainierte Entscheidungsbaum aus der Textklassifikation der Rückrufe dargestellt. Hierbei ist der oberste Knoten mit dem Begriff „software“ der Wurzelknoten. In diesem Knoten wird geprüft, ob in dem jeweiligen Rückruf das Wort „software“ vorkommt. Wenn es nicht darin vorkommt und die Bedingung $software \leq 0{,}5$ erfüllt ist, verläuft der Pfad nach links und endet dann schon im dortigen Blattknoten. Dieser ist daran erkennbar, dass er kein Wort mehr beinhaltet und die Ergebnisse eindeutig zugeordnet werden konnten ($value = [7439{,}0]$). Somit wurden 7439 Rückrufe bereits zutreffend der nicht softwarebedingten Kategorie zugeordnet.
Allerdings sagt der initiale Wurzelknoten aus, dass 7550 Rückrufe existieren, die dem Cluster 0 – also als nicht softwarebedingt – zugeordnet werden ($value = [7550{,}450]$).
Somit befinden sich nun auf dem rechten Pfad noch 111 Rückrufe, die nicht softwarebedingt waren, aber trotzdem das Wort „software“ beinhaltet haben. Zudem werden alle 450 Rückrufe mitgenommen, die im Trainingsset als softwarebedingt eingestuft sind. Erst in dem folgenden internen Knoten mit dem Wort „inspect“ als Entscheidungsgrundlage, werden diese weiter aufgeteilt.

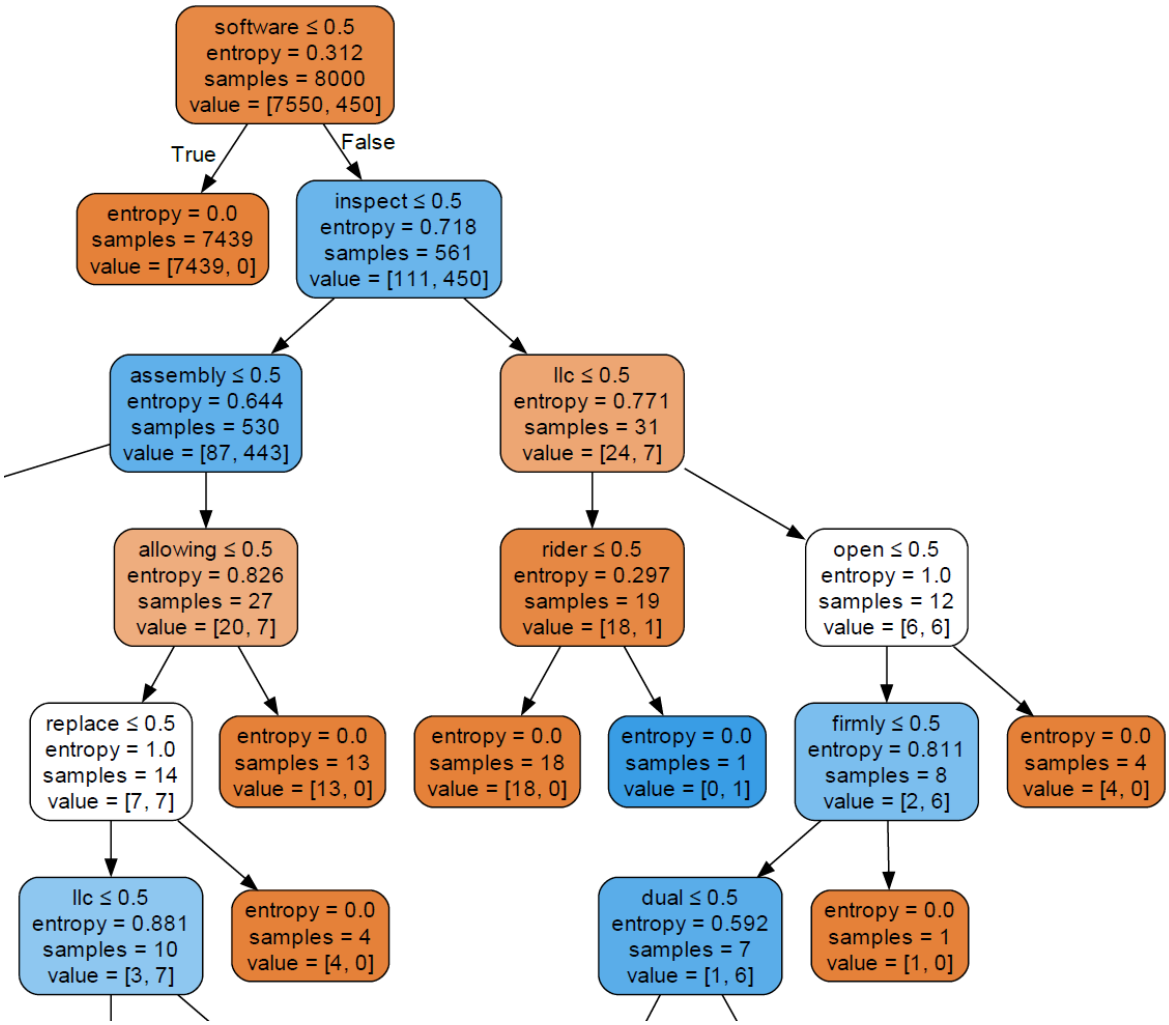

Abbildung 20: Trainierter Entscheidungsbaum

Dieses Vorgehen wird nun so lange wiederholt und der Entscheidungsbaum weiter aufgebaut, bis alle Rückrufe mithilfe von Wahr und Falsch Entscheidungskriterien eingeordnet wurden und das Modell fertig trainiert ist.

Wird das Modell nun auf einen neuen und unbekannten Datensatz angewendet, wird es die Rückrufe anhand der Entscheidungskriterien bewerten und einer Kategorie zuordnen.

Mit dem Verfahren des Entscheidungsbaumes werden keine Reihenfolgen von Wörtern oder Muster in Texten erkannt und bewertet. Es basiert lediglich auf dem Vorkommen von einzelnen Stichwörtern. Folglich können sich teilweise unsinnige Entscheidungskriterien ergeben. Beispielsweise finden sich in dem fertig trainierten Modell auch Marken- und Modellnamen wie „cherokee“, „evoque“ oder „nova“.

Dies ist eigentlich nicht Sinn der Entscheidungskriterien und basiert nun eher auf einem Zufall und dem Mangel von genug Trainingsdaten.
Aufgrund dessen werden im Folgenden Neuronale Netzwerke betrachtet, da diese auch Wortreihenfolgen und insbesondere Muster in den Wortstrukturen erkennen können.

5.3. Neuronale Netzwerke

Das Vorgehen für FFNN und RNN ist hierbei identisch zu dem der Klassifikationsverfahren in 5.2. Es wird lediglich die verwendete Methode durch FFNN oder RNN ersetzt. Aufgrund der vergleichsweise schlechten Qualität der Ergebnisse von FFNN und RNN (vgl. 6.1) wird im Folgenden nur auf CNN eingegangen. Hierbei wird die Bibliothek TensorFlow Keras [51] verwendet.

CNN benötigen eine etwas andere Vorverarbeitung der Daten mithilfe einer abweichenden Tokenisierung und zusätzlichem Padding. Dieses abweichende Vorgehen ist in 4.2.6 näher beschrieben. Durch eine empirische Ermittlung wird festgestellt, dass CNN wesentlich besser – Senkung der durchschnittlichen Fehlerquote von 0,56 % auf 0,32 % – mit nicht normalisierten Daten arbeiten. Demzufolge wird das CNN-Verfahren ohne eine vorherige Normalisierung verwendet.

Der Aufbau des hierbei verwendeten CNN besteht aus fünf Schichten:

1. Einer Einbettungsschicht, um einen dichten Vektorraum zu erzeugen, in dem die Eingaben – also die tokenisierten und gepaddeten Daten – dargestellt werden können. Die eingespeisten Ganzzahlsequenzen in zweidimensionaler Form werden in einer Rechenoperation zu Gleitkommazahlen gewandelt und vieldimensional dargestellt. Der Sinn hiervon ist es, Speicher und Rechenleistung zu sparen.
   Dem Modell wird vorab mitgeteilt, wie groß das ausgelesene Vokabular und welche Länge die Sequenzen haben. Zudem wird die Dimensionalität der Vektoren bestimmt. Bei der Dimensionalität wird der Wert von 100 gewählt, der ein guter Kompromiss aus Genauigkeit und Effizienz darstellt. Bei einer zu hohen Anzahl wird zudem sogenanntes Overfitting und daraus folgend schlechtere Ergebnisse riskiert.
   Demnach entsteht in dem Schritt der Einbettung beispielhaft aus einer 32x10-Matrix nun eine 32x10x100-Matrix [52].

2. Einer Faltungsschicht, um mithilfe von Filtern lokale Muster und Merkmale in den Daten zu erkennen. Hierbei wird eine Anzahl von 32 Filtern gewählt, da dies die Modellkomplexität reduziert, ohne zu viele Informationen zu verlieren. In der Faltungsschicht wird die Kernel-Größe von 5 gewählt, um einen guten Kompromiss zwischen breitem Kontextverständnis und der Beachtung spezifischer Merkmale zu haben. Als Aktivierungsfunktion wird hier auf Rectified Linear Unit (ReLU) zurückgegriffen, da dies eine gängige Methode für CNN darstellt.
3. Einer Poolingschicht, in der die Dimensionalität der Daten reduziert werden, um nur die relevantesten Informationen beizubehalten. Eine empirische Auswertung ergibt, dass das *GlobalMaxPooling1D* [51] die besten Ergebnisse für den Anwendungsfall liefert.
4. Einer normalen Dense-Schicht mit 32 Neuronen der Aktivierungsfunktion ReLU, in der die eigentliche Gewichtung und Klassifikation geschieht.
5. In der letzten Schicht wird eine Dense-Schicht mit nur einem Neuron mit der Aktivierungsfunktion *Sigmoid* verwendet, um eine eindeutige binäre Klassifikation zu erreichen.

Der Ablauf der Klassifikation mit CNN ist in Abbildung 21 dargestellt. Er unterscheidet sich lediglich in der Datenvorverarbeitung für das CNN zum Verfahrensablauf in 5.2.
Die Vektorisierung entfällt, da diese durch eine interne Funktion des Tokenizers ersetzt wird. Durch den Entfall der Normalisierung werden bessere Ergebnisse erreicht, somit entfällt diese vollständig.
Zuletzt wird das Padding (4.2.6) durchgeführt und die Daten im Anschluss in das Modell eingespeist.

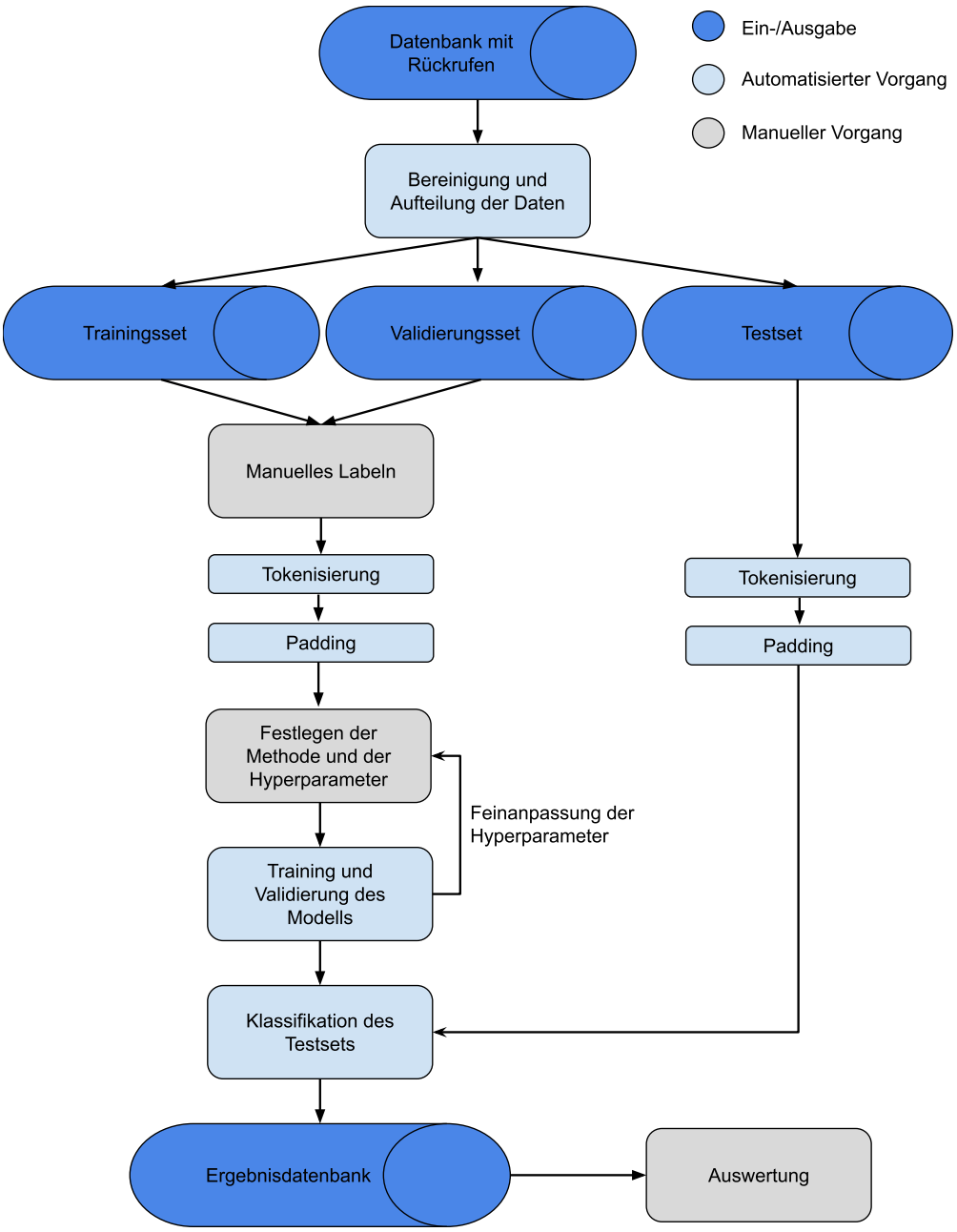

Abbildung 21: Flussdiagramm CNN

6. Ergebnisse und Diskussion

6.1. Vergleich angewandter Verfahren

Um die unterschiedlichen Lernverfahren sinnvoll miteinander vergleichen zu können, werden diese empirisch erprobt. Dies geschieht anhand einer für alle Versuche identischen Stichprobe von 1000 zufällig gewählten Rückrufen, die zuvor manuell gelabelt wurden.
Da aufgrund von zufälligen Initialisierungen der Modelle oder auf Stochastik basierenden Bestandteilen der Optimierungsfunktionen bei jedem Durchlauf leicht abweichende Ergebnisse erreicht werden, werden pro Verfahren fünf Versuche durchgeführt.
Die ermittelten Fehlerquoten der fünf Ergebnisse werden arithmetisch gemittelt und daraus eine durchschnittliche Gesamtfehlerquote pro Verfahren errechnet.
Anhand dieser Fehlerquote können Ränge vergeben und die einzelnen Verfahren gegeneinander bewertet werden. Die Ergebnisse der Versuche finden sich in Tabelle 12.

Verfahren	**V1**	**V2**	**V3**	**V4**	**V5**	**Fehlerquote ⌀**	**Rang**
Clustering	0,9 %	0,9 %	0,9 %	0,9 %	0,9 %	0,90 %	**3.**
Decision Tree	0,7 %	0,6 %	0,4 %	0,5 %	0,6 %	0,56 %	**2.**
Random Forest	1,8 %	2,3 %	1,9 %	2,2 %	1,8 %	2,00 %	**6.**
FFNN	1,1 %	1,1 %	1,2 %	1,5 %	1,2 %	1,22 %	**5.**
RNN	1,0 %	1,1 %	1,3 %	1,0 %	1,2 %	1,12 %	**4.**
CNN	0,3 %	0,2 %	0,4 %	0,4 %	0,3 %	0,32 %	**1.**

Tabelle 12: Stichprobenauswertung ML-Verfahren

Aus den Ergebnissen schlussfolgernd lässt sich die höchste Präzision bei dem CNN feststellen. Dies ergibt sich daraus, dass durch Filter- und Poolingschichten die spezifischen Merkmale am besten aus den Textdaten herausgearbeitet werden können und die angehängten Neuronenschichten die Rückrufe sinnvoll einordnen können. Der einzige Nachteil bei den CNN ergibt sich lediglich durch die längere Laufzeit aufgrund des komplexen Aufbau des Netzes. Dieser Nachteil ist allerdings in dem

Anwendungsfall der Rückrufkategorisierung vernachlässigbar, da es sich hierbei trotzdem noch von Laufzeiten knapp unter einer Minute handelt.
Bei größeren Dateimengen könnte hier nun auf einen Decision Tree zurückgegriffen werden, da dieser absolut zwar eine marginal höhere Fehlerquote hat, die Laufzeit aber nur bei wenigen Sekunden liegt.

Da im vorliegenden Anwendungsfall die geringe Trainingszeit nicht sonderlich von Relevanz ist, wird in der Auswertung der NHTSA-Daten das CNN für die Kategorisierung und Auswertung verwendet (siehe 6.3).

6.2. Kategorisierung RAPEX-Datenbank

Bei der Kategorisierung der RAPEX-Daten ist es aufgrund von mangelnder Datenqualität der Rückrufe häufig nicht möglich, eindeutig zu bestimmen, was die genaue Ursache eines Rückrufes ist. Somit können auch bei einigen Rückrufen keine Rückschlüsse gezogen werden, ob der Rückruf aufgrund eines Software-Fehlers geschehen ist oder nicht.

Als Beispiel ist hier ein Rückruf mit der Alert Number 1502/10 aus dem Jahre 2010 aufzuführen. Hierbei ist die einzige Information im Rückruf: *„Under certain circumstances, there is a possibility that the driver's airbag will not deploy in the event of an accident."* [53]. Daraus lässt sich nur der Grund des Rückrufes – ein mögliches Nichtauslösen des Airbags – ableiten, aber nicht die Ursache. Insbesondere könnte die Ursache bei einem Airbag-Fehler durchaus softwarebedingt sein und wird durch das ML-Verfahren aber nicht als Software-Fehler markiert.

Daher resultieren – insbesondere im Vergleich zu den NHTSA-Daten – niedrige Verhältnisse der Software-Fehler. In absoluten Zahlen beläuft sich die im Clustering-Verfahren ermittelte Anzahl der softwarebedingten Rückrufe auf nur etwa 135 von insgesamt 4966 Einträgen (vgl. Abbildung 22).

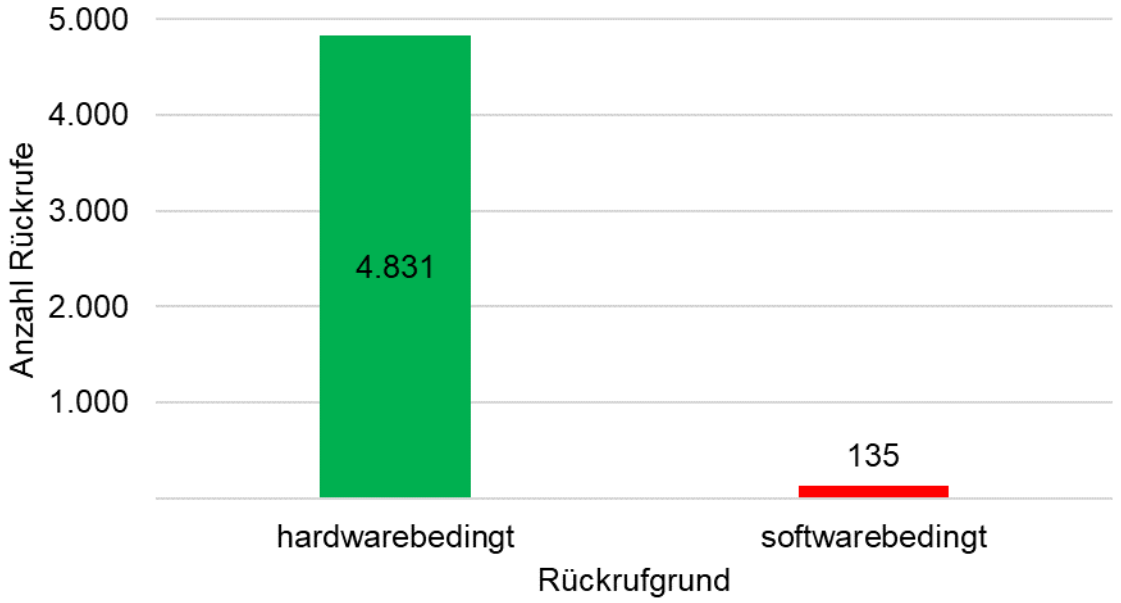

Abbildung 22: Balkendiagramm absolute Zahlen Clustering Rückrufe RAPEX

In relativen Zahlen liegt somit der Anteil der softwarebedingten Rückrufe bei etwa 2,72 % (vgl. Abbildung 23).

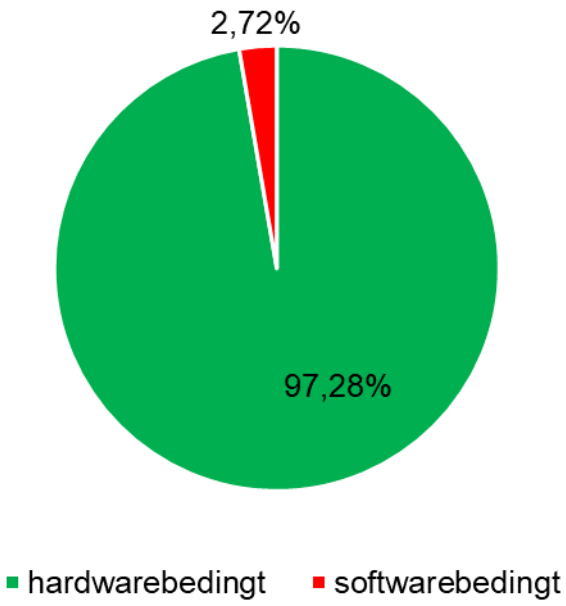

Abbildung 23: Kreisdiagramm relative Zahlen Clustering Rückrufe RAPEX

Komplexere und überwachte Verfahren werden aufgrund der mangelnden Datenqualität nicht angewandt und die Daten der RAPEX-Datenbank im Folgenden in der Auswertung vernachlässigt.

6.3. Kategorisierung NHTSA-Datenbank

Da die NHTSA-Datenbank nur von einer Behörde zentralisiert geführt wird und im Gegensatz zu RAPEX nicht aus lediglich zusammengetragenen Informationen unterschiedlicher Länder besteht, ist die Datenqualität sehr gut. In den Rückrufen ist sowohl Rückrufgrund als auch die Rückrufursache meist deutlich erkennbar. Dies bietet eine ideale Grundlage für eine genaue ML-gestützte Erkennung von softwareabhängigen Rückrufen.

Aufgrund der besseren Datenlage der NHTSA-Datenbank und somit einer genaueren Einordnung der Rückrufe ergibt sich auch ein höherer Anteil von softwarebedingten Rückrufen. Zudem werden die Rückrufe mit dem Verfahren der CNN genauer kategorisiert als die RAPEX-Daten.
In absoluten Zahlen sind 736 Rückrufe bei einer Gesamtanzahl von 13.415 Rückrufen softwarebedingt (vgl. Abbildung 24).

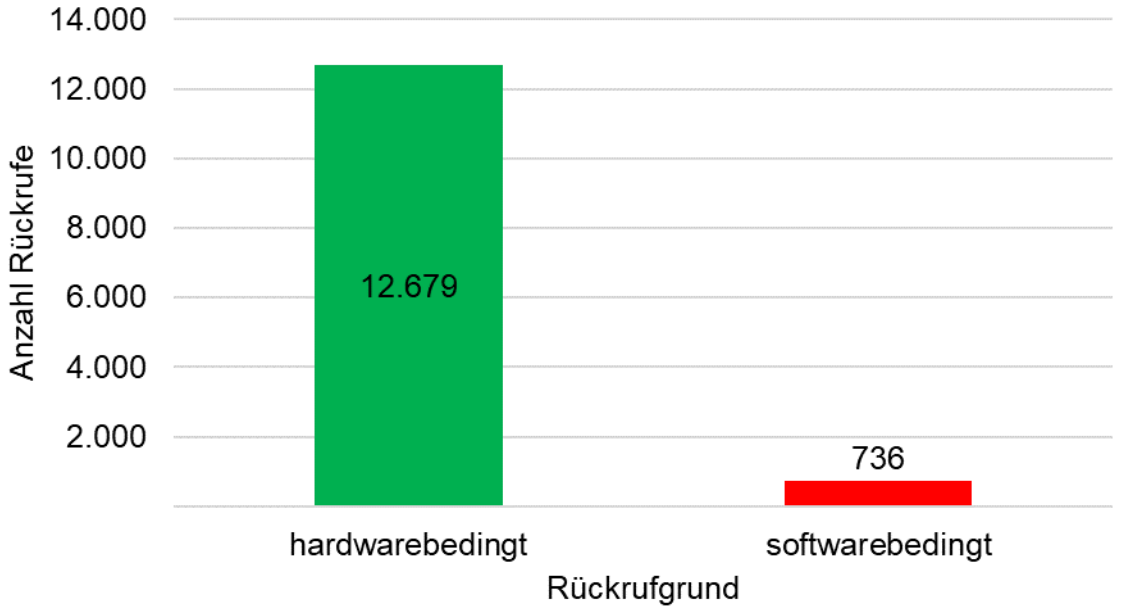

Abbildung 24: Balkendiagramm absolut Rückrufe NHTSA

In relativen Zahlen liegt somit der Anteil der softwarebedingten Rückrufe bei NHTSA etwa bei 5,49 % (vgl. Abbildung 25).

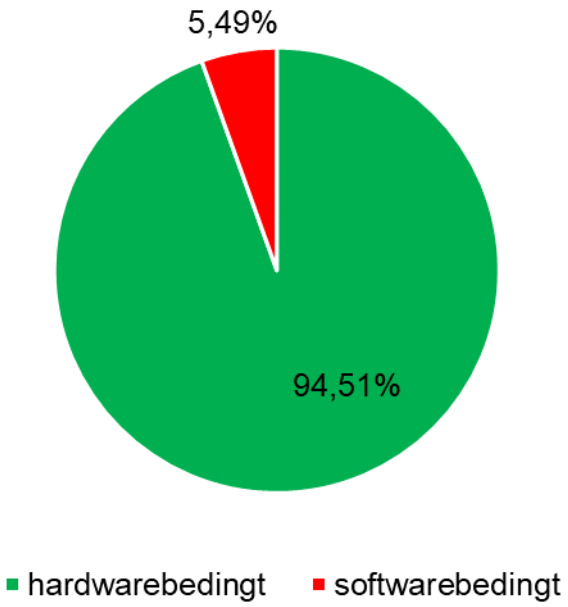

Abbildung 25: Kreisdiagramm relativ Rückrufe NHTSA

Zudem ist in den Daten der NHTSA auch die Anzahl der potenziell betroffenen Fahrzeuge angegeben (POTAFF). Mit dieser Angabe lassen sich hier auch absolute und relative Verhältnisse in Bezug auf die Fahrzeuganzahl schließen. Absolut sind etwa 42.700.000 von knapp 508.000.000 insgesamt potenziell betroffenen Fahrzeugen aufgrund von Software-Fehlern zurückgerufen worden (vgl. Abbildung 26).

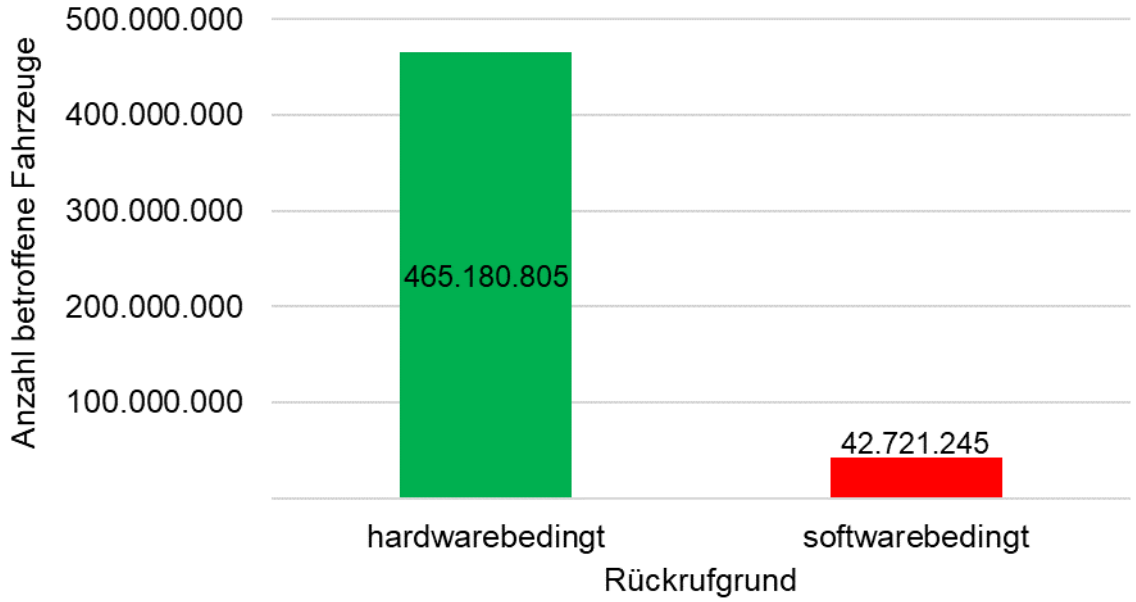

Abbildung 26: Balkendiagramm absolut betroffene Fahrzeuge NHTSA

Relativ betrachtet ergeben sich hierbei etwa 8,41 % der Fahrzeuge, die aufgrund von Software-Fehlern zurückgerufen wurden (vgl.Abbildung 27).

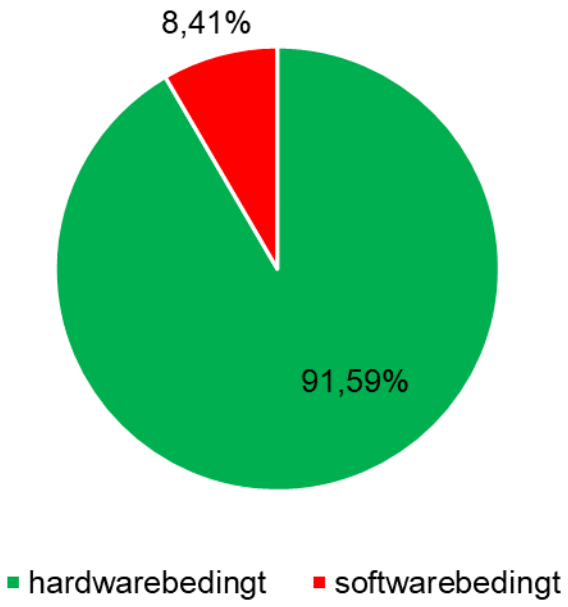

Abbildung 27: Kreisdiagramm relativ betroffene Fahrzeuge NHTSA

6.3.1. Historische Entwicklung

Aus den kategorisierten Daten lassen sich nun auch historische Darstellungen erzeugen. Interessant ist an dieser Stelle eine Betrachtung der Modelljahre 2000 bis 2022 und ein Vergleich der absoluten Anzahl der softwarebedingten Rückrufe pro Modelljahr in Abbildung 28 und dem relativen Anteil der SW-bedingten Rückrufe an allen Rückrufen pro Modelljahr in Abbildung 29.

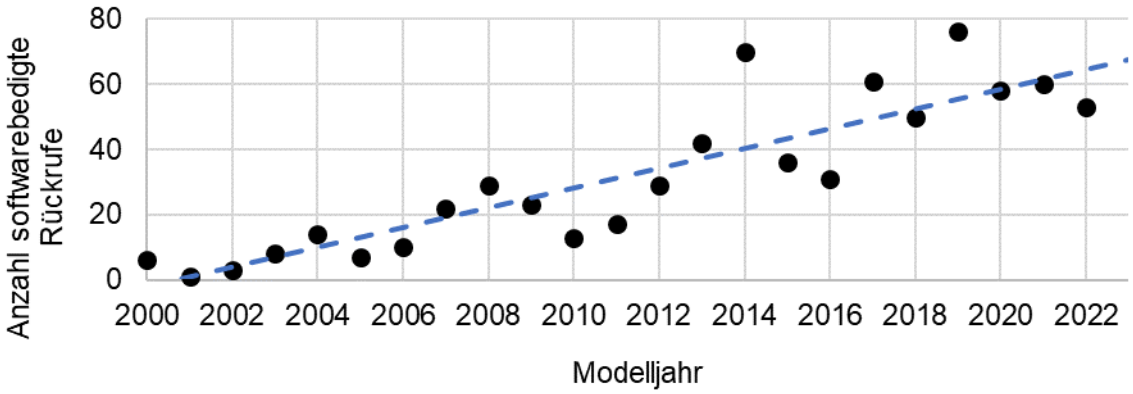

Abbildung 28: Absolute Anzahl softwarebedingte Rückrufe pro Modelljahr

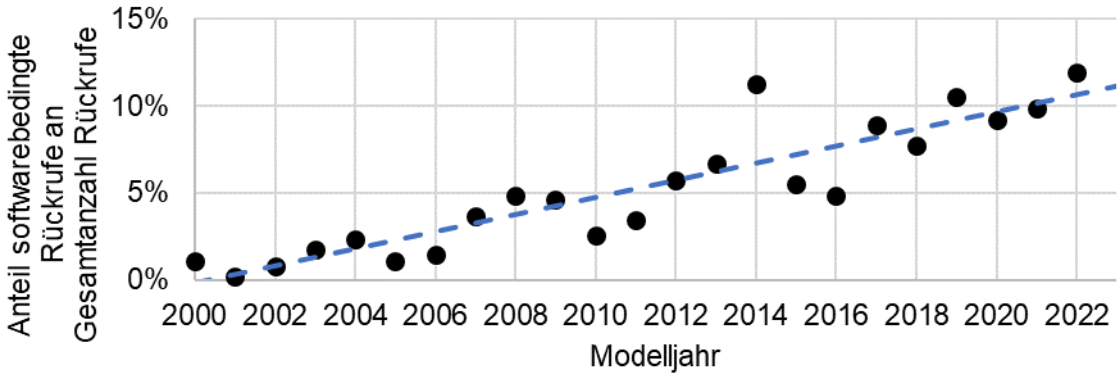

Abbildung 29: Relativer Anteil softwarebedingte Rückrufe pro Modelljahr

Grundsätzlich zeigen die beiden Diagramme einen stetigen Anstieg der softwarebedingten Rückrufe in den letzten 22 Jahren. Verdeutlicht wird dies durch die abgebildete Trendlinie.

Zudem ist erkennbar, dass bis auf kleinere Ausreißer der Anstieg der absoluten Anzahl ungefähr proportional zum relativen Anteil verläuft.

Dies zeigt, dass die Anzahl der softwarebedingten Rückrufe aufgrund von immer komplexerer Software gestiegen ist.

Interessant in der Betrachtung der Abbildung 28 und Abbildung 29 ist auch eine einfache Extrapolation der linearen Trendlinie in der Form $y = m * x + b$. Die gerundeten Ergebnisse für diese Extrapolation für die Modelljahre bis 2050 finden sich in Tabelle 13.

	SW-Rückrufe absolut $y = 3{,}0237 * x - 6049{,}4$	**SW-Rückrufe relativ** $y = 0{,}004907 * x - 9{,}816652$
2025	74	12,0 %
2030	89	14,5 %
2035	104	16,9 %
2040	119	19,4 %
2050	149	24,3 %

Tabelle 13: Lineare Extrapolation Historie Software-Rückrufe

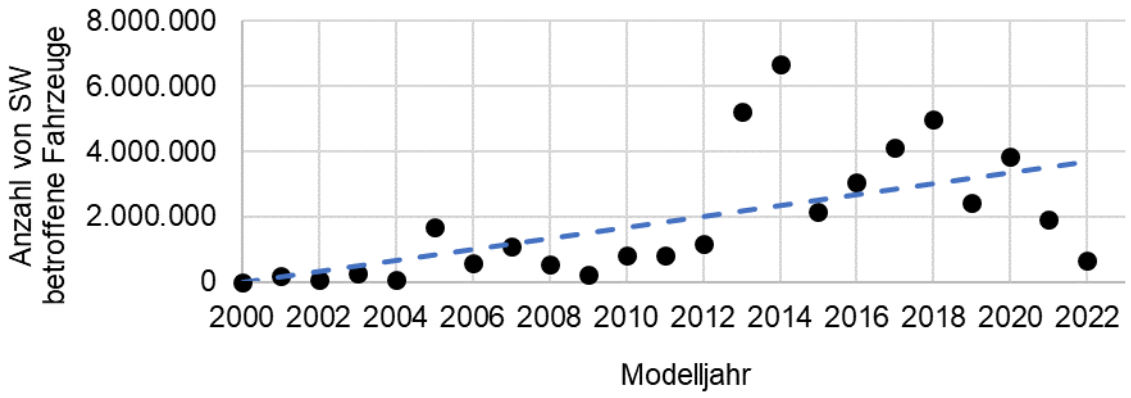

Abbildung 30: Absolute Anzahl softwarebedingte Rückrufe in Fahrzeugen pro Modelljahr

In Abbildung 30 ist die Anzahl der Fahrzeuge, die aufgrund von Software-Fehlern zurückgerufen wurden, in absoluten Zahlen dargestellt. Auffällig ist bei der Betrachtung das Modelljahr 2013 und 2014 mit starken Ausreißern im Gesamtverlauf. Das Modelljahr 2014 stellt mit etwa 6,7 Millionen Rückrufen im zeitlichen Verlauf sogar das am stärksten betroffene Modelljahr dar.

Im Abgleich mit den einzelnen Rückrufen lässt sich der Ausreißer im Jahr 2014 mit dem Rückruf 16V651000 [54] des Konzerns General Motors LLC erklären. Hier wurden aufgrund eines Problems mit dem Airbag-System Fahrzeuge der Marken Chevrolet, Cadillac, GMC und Buick zurückgerufen.
Der Rückrufgrund war, dass gewisse Bedingungen während der Fahrt die Airbag-Modul-Software in einen Diagnosemodus versetzten. Während dieser Modus aktiv ist, löst der Airbag im Falle eines Unfalles nicht aus.
Von diesem Fehler waren etwa 3,6 Millionen Fahrzeuge aus dem Modelljahr 2014 betroffen und wurden zurückgerufen. Allein dieser Rückruf verursacht über 50 % der zurückgerufenen Fahrzeuge aus dem Jahr 2014. Aufgrund dessen besteht auch der enorme Ausschlag.
Die hohe Anzahl im Modelljahr 2013 von etwa 5,2 Millionen Fahrzeugen lässt sich nicht mit nur einem Rückruf erklären. Hierbei kamen mehrere große Rückrufe zusammen. Unter anderem war der Konzern FCA US LLC bei etwa 1,4 Millionen Fahrzeugen der Marken Chrysler, Dodge, Jeep und RAM von einem Sicherheitsproblem der Radio-Software betroffen (15V461000 [55]).

Auch der Rückruf 17V302000 [56] verursachte aufgrund eines falsch gesetzten Parameters innerhalb des Airbag-Systems den Rückruf von etwa einer Million Fahrzeuge der Marke RAM.
Überdies ausschlaggebend ist der Rückruf 14V138000 [57], in dem Fahrzeuge der Marken Nissan und Infiniti aufgrund einer falschen Software-Kalibrierung im Airbag-System zurückgerufen wurden. Insgesamt waren auch hier knapp eine Million Fahrzeuge betroffen.
Interessant wird der Vergleich des relativen Anteils der softwarebedingten Rückrufe aus Abbildung 29 in der Gegenüberstellung zum relativen Anteil der Fahrzeuge, die aufgrund von softwarebedingten Problemen zurückgerufen wurden in Abbildung 31.

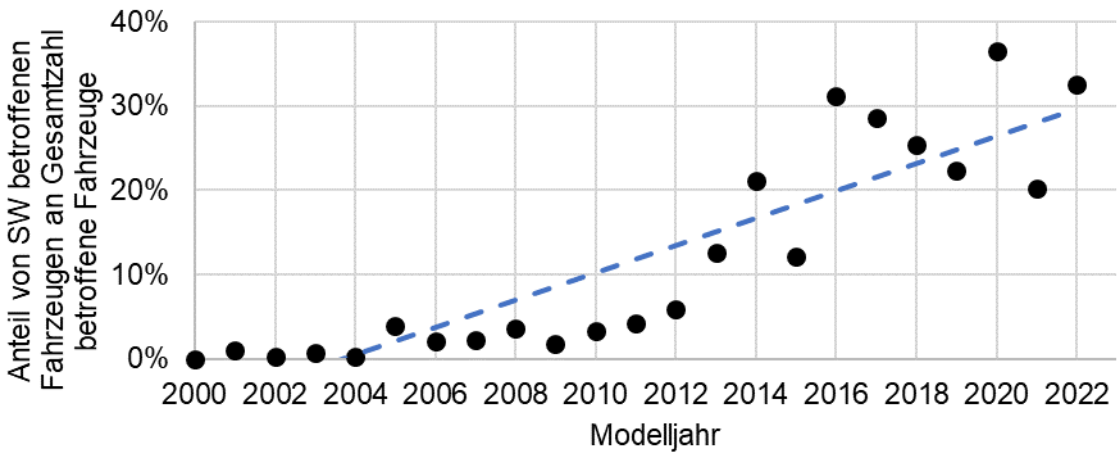

Abbildung 31: Relativer Anteil softwarebedingte Rückrufe in Fahrzeugen pro Modelljahr

Auffällig ist in der Betrachtung die enorme Differenz zwischen den Rückruf- und Fahrzeugzahlen in den Jahren nach 2012. Waren beispielsweise im Jahr 2016 4,8 % der Rückrufe softwarebedingt, waren es trotzdem 31,3 % der zurückgerufenen Fahrzeuge, die aufgrund von Software-Fehlern zurückgerufen wurden.

Dieser starke Anstieg der Differenz zwischen Anzahl der Rückrufe und Anzahl der Fahrzeuge ist dadurch zu begründen, dass bei Software-Fehlern oftmals Fahrzeuge in großer Zahl betroffen sind. Da die Software der Automobilhersteller meist in sehr vielen unterschiedlichen Fahrzeugen verwendet wird, sind bei einem Rückruf dann auch vielzählige Fahrzeuge schlagartig betroffen.

Außerdem wird eine Vereinheitlichung von Softwarestrukturen angestrebt. Die Software der Fahrzeuge soll möglichst identisch sein, um Entwicklungs- und Wartungskosten gering zu halten. Nachteil hiervon ist aber die hohe Anzahl von betroffenen Fahrzeugen, falls doch Probleme mit der Software auftreten.
Außerdem spielt auch die sinkende Modellvielfalt eine entscheidende Rolle. Durch die sinkende Anzahl von Fahrzeugderivaten werden neue Fahrzeuge weiter vereinheitlicht. Tritt dann ein Software-Fehler spezifisch bei einem Modell auf, sind bei einem Rückruf mehr Fahrzeuge auf einmal betroffen.

Im arithmetischen Mittel der Modelljahre seit 2000 handelt es sich um 58.045 Fahrzeuge pro Software-Rückruf und nur um 36.829 pro Hardware-Rückruf (vgl. Tabelle 14). Allerdings ist die Standardabweichung σ bei beiden Rückrufkategorien sehr hoch, was auf eine starke Streuung der Daten hindeutet.
Aus diesem Grund wird in Tabelle 14 zusätzlich der Median dargestellt, der eine genauere Schätzung der zentralen Tendenz liefert und zudem robuster gegenüber Ausreißern ist [58]. Aufgrund dessen stellt dieser hier die bessere Wahl dar. Der Median zeigt hier deutlich, dass mit 2499 wesentlich mehr Fahrzeuge bei Software-Rückrufen betroffen sind als dies mit 463 Fahrzeugen bei Hardware-Rückrufen der Fall ist.

	softwarebedingt	**hardwarebedingt**
Ø Anzahl Fahrzeuge pro Rückruf	58.045	36.829
Standardabweichung σ	210.293	190.662
Median Fahrzeuge pro Rückruf	2.499	463

Tabelle 14: Durchschnitt, Standardabweichung und Median Anzahl Fahrzeuge pro Rückruf HW/SW

In einer linearen Extrapolation für die Anzahl der Fahrzeuge, die aufgrund von Software-Fehlern zurückgerufen werden, stellen sich die Ergebnisse gerundet in Tabelle 15 dar.

	SW-Fahrzeuge absolut $y = 168249{,}68x - 336492712{,}7$	**SW-Fahrzeuge relativ** $y = 0{,}016168 * x - 32{,}395516$
2025	4.207.000	34,5 %
2030	5.048.000	42,6 %
2035	5.889.000	50,6 %
2040	6.731.000	58,7 %
2050	8.413.000	74,9 %

Tabelle 15: Lineare Extrapolation Historie Software-Fahrzeuge

Laut dieser einfachen Extrapolation stellt sich schon im Jahr 2035 eine Quote von 50 % aller zurückgerufenen Fahrzeuge, die aufgrund von Software-Fehlern zurückgerufen würden, dar. Im Jahr 2050 liegt die Voraussage schon bei 75 % und somit sehr hoch.

6.3.2. Auswertung der Hersteller

Relevant ist auch die Betrachtung der Herstellerabhängigkeit von softwarebedingten Rückrufen. Zuerst sind in Abbildung 32 die 15 Hersteller mit den meisten softwarebedingten Rückrufen sortiert nach Anzahl der softwarebedingten Rückrufe dargestellt. Die prozentuale Beschriftung in dieser Darstellung ergibt sich aus der Gesamtanzahl der Software-Rückrufe der 15 abgebildeten Hersteller. Allein an der Anzahl der Rückrufe gemessen, machen insbesondere Mercedes-Benz, Ford und BMW knapp ein Drittel der Gesamtanzahl aus.

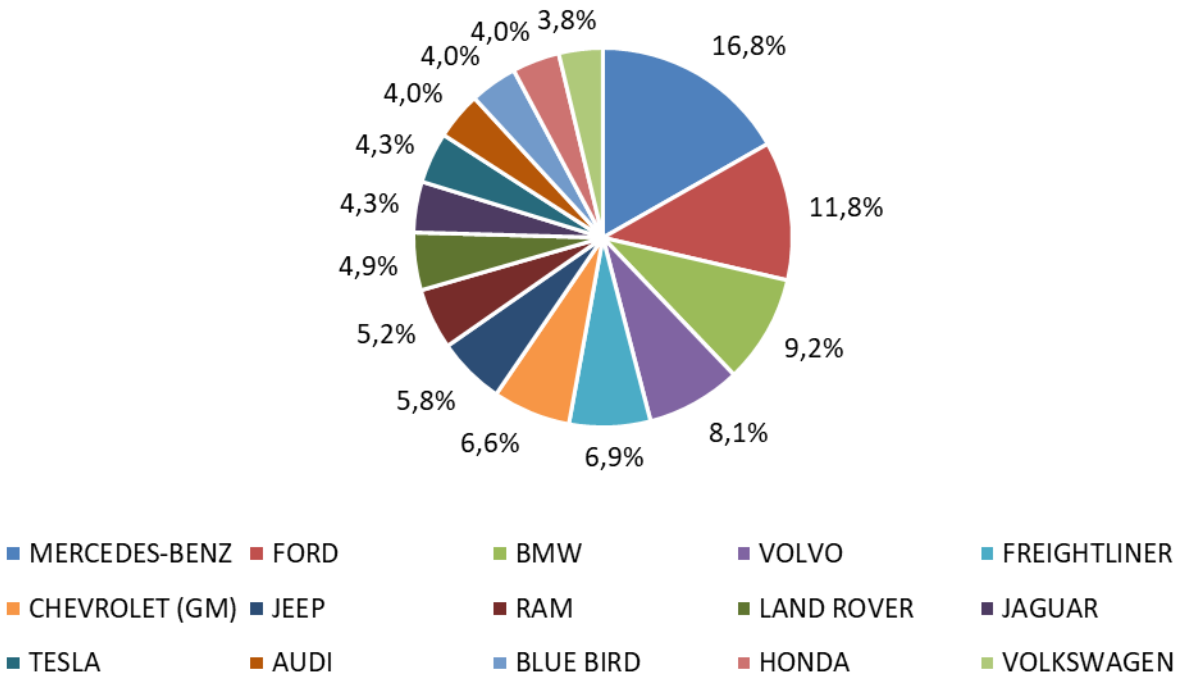

Abbildung 32: Relativer Anteil Software-Fehler-Rückrufe pro Marke

In Abbildung 33 sind die gleichen Hersteller in identischer Reihenfolge dargestellt, jedoch sind hier die Anzahl der von den Rückrufen betroffenen Fahrzeuge und der prozentuale Anteil an der Gesamtzahl der von Software-Rückrufen betroffenen Fahrzeuge der dargestellten Hersteller angegeben.

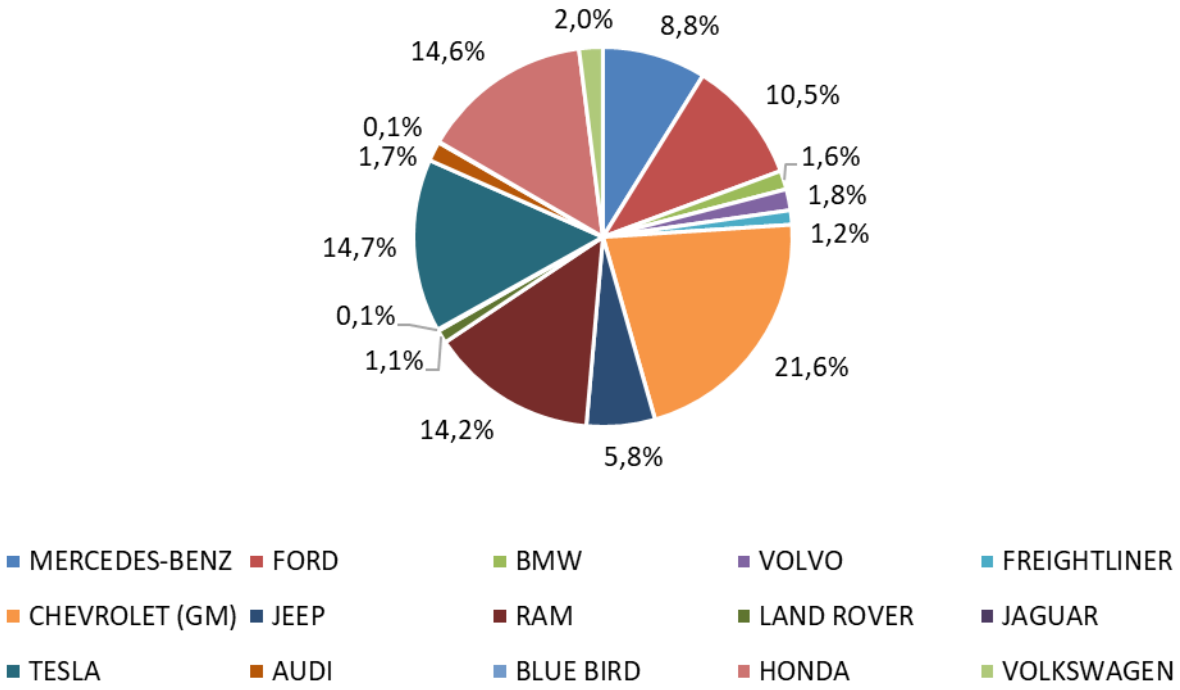

Abbildung 33: Relativer Anteil Software-Fehler-Fahrzeuge pro Marke

Auffällig sind im Vergleich nun die starke Umverteilung und der Unterschied der Fahrzeuganzahl zur Anzahl der Rückrufe. Auf Basis der Fahrzeuganzahl machen nun Chevrolet, Tesla und Honda etwa ein Drittel aller Fahrzeuge aus.

Bei Tesla beispielsweise liegt dies insbesondere an der kleinen Modellpalette von vier aktuellen Modellen zum Stand 2023 [59]. Dies bedeutet für den Hersteller, dass bei einem Rückruf eines Modells sehr viele Fahrzeuge auf einmal betroffen sind. Zudem verwendet der Hersteller Tesla modellübergreifend eine zum großen Teil identische Software. Demnach können aufgrund eines Software-Fehlers dann direkt auch mehrere Modellreihen betroffen sein.

Chevrolet und Honda sind dagegen Fahrzeugmarken, die zu den vier meistverkauften Fahrzeugmarken in den USA im Jahr 2022 gehören [60]. Folglich betreffen die Rückrufe auch sehr viele Fahrzeuge in den USA. Allerdings liegt Chevrolet mit knapp über 6 Millionen zurückgerufenen Fahrzeugen, die aufgrund von Software-Fehlern zurückgerufen wurden, auf dem ersten Platz aller Marken in den USA. Begründen lässt sich dies unter anderem auch damit, dass beim Eintrag von Rückrufen von Chevrolet sich auch oftmals andere Hersteller – unter anderem Buick, Cadillac, Corvette, GMC – des Konzerns GM LLC verbergen. Dies resultiert dann in einer sehr hohen Anzahl von betroffenen Fahrzeugen und die in dieser Auswertung unter den Rückruf der Marke Chevrolet fallen.

Abbildung 34 stellt den Anteil der Fahrzeuge, die von Software-Rückrufen betroffen sind, an der Gesamtanzahl der zurückgerufenen Fahrzeuge von den 15 jeweiligen Herstellern dar. Im Anteil der Fahrzeuge, die aufgrund von Software-Fehlern zurückgerufen wurden, liegt Tesla mit etwa 81% (vgl. Abbildung 34) unter der Liste der 15 Hersteller mit Abstand auf dem ersten Platz.

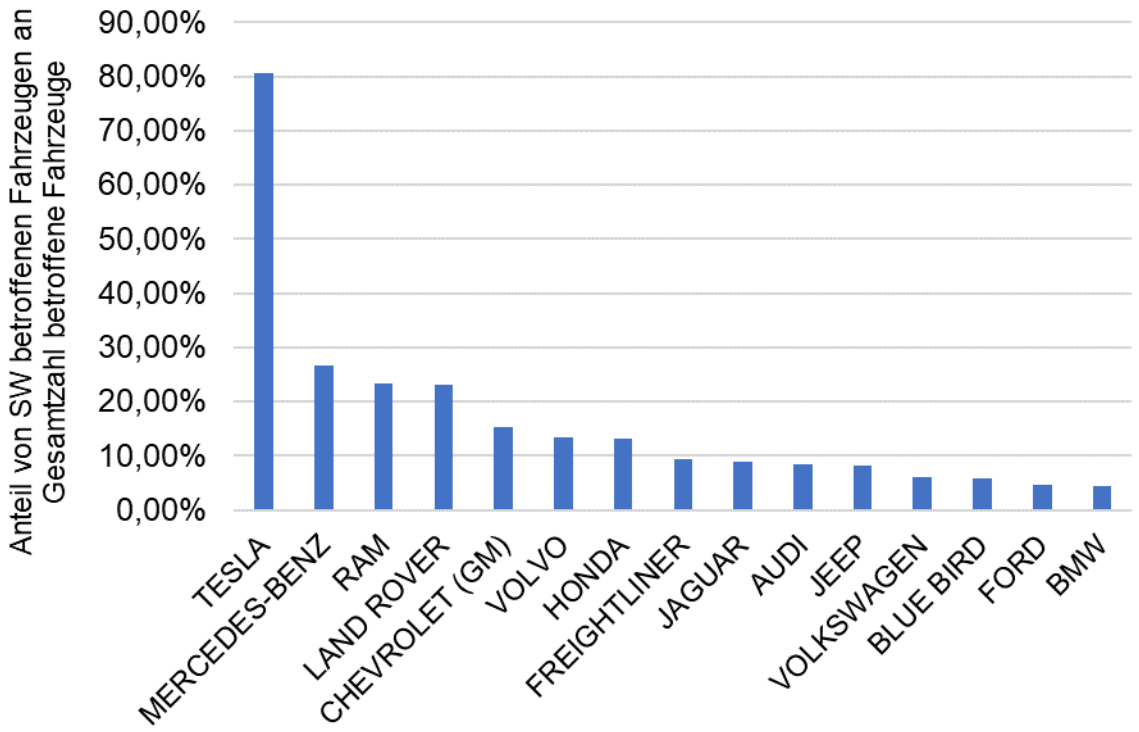

Abbildung 34: Anteil softwarebedingte Rückrufe an Gesamtrückrufen in Fahrzeugen pro Hersteller

Bei Tesla liegt dieser auffällig hohe Anteil an Software-Fehlern hauptsächlich an der hohen Integration von komplexer Software, dem Großteil der Steuerung und Berechnung über Software und einer zentralisierten Fahrzeugstruktur, bei der der Hauptteil der Steuerung in einem zentralen Computer geschieht. Solch eine zentralisierte Software-Struktur erhöht die Wahrscheinlichkeit für komplexe Software-Strukturen und somit auch für Software-Fehler.

Vorteilhaft an der Software-Struktur von Tesla ist aber die Behebung von Fehlern. Die Fahrzeuge müssen bei Rückrufen zumeist nicht in die Werkstatt, sondern die Fehler werden über Over-the-Air-Updates (OTA) zentral behoben. Somit entstehen nur sehr geringe Kosten für den Hersteller und die Kundenzufriedenheit leidet nicht unter ungeplanten Werkstattaufenthalten. Beispielhaft kann hier der Rückruf 22V818000 [61] herangezogen werden, in dem der Hersteller Tesla laut eigener Angabe innerhalb von weniger als zwei Wochen über 97 % aller Fahrzeuge das benötigte Software-Update installiert hatten [62]. Demnach waren sowohl der Fehler als auch das Sicherheitsrisiko behoben.

6.3.3. Differenz Rückrufjahr - Modelljahr

Durch die Informationen zu Modelljahr (YEARTXT) und Rückrufjahr (CAMPNO) in den Daten der NHTSA lassen sich diese auch auswerten.
Das Ziel der Auswertung ist es festzustellen, wie viel Zeit vergeht, bis ein Fehler entdeckt und durch einen Rückruf behoben wird. Interessant hierbei ist der Vergleich zwischen Soft- und Hardware-Rückrufen.

Um die zwei Rückrufarten miteinander vergleichen zu können, wird das Rückrufjahr von dem Modelljahr subtrahiert. Dadurch entsteht für jeden Rückruf eine Differenz an Jahren Δt, die benötigt wurden, bis das jeweilige Fehlersymptom auffällig und ein Rückruf eingeleitet wurde.
Nun werden pro Differenzzeitraum – in einem Abstand von jeweils einem Jahr – die Anzahl der Rückrufe in Abbildung 35 aufgetragen. Zudem werden die Daten auch auf Soft- und Hardware-Rückrufe aufgeteilt und die Anzahl der Rückrufe über zwei unterschiedliche, vertikale Achsen aufgetragen, um einen Vergleich des Verlaufes zu ermöglichen.
In der grafischen Darstellung in Abbildung 35 ähneln sich die Verläufe optisch recht stark. Die einzige Auffälligkeit ist lediglich der Abbruch der Linie der Software-Rückrufe nach 15 Jahren. Im Zeitraum von 2000 bis 2023 existieren keine softwarebedingten Rückrufe, die erst nach über 15 Jahren entdeckt wurden.
Für Hardware-Rückrufe gilt dies nicht: Bei diesen wurden die Fehler sogar bis zu 21 Jahre nach dem ersten Modelljahr entdeckt und zurückgerufen.

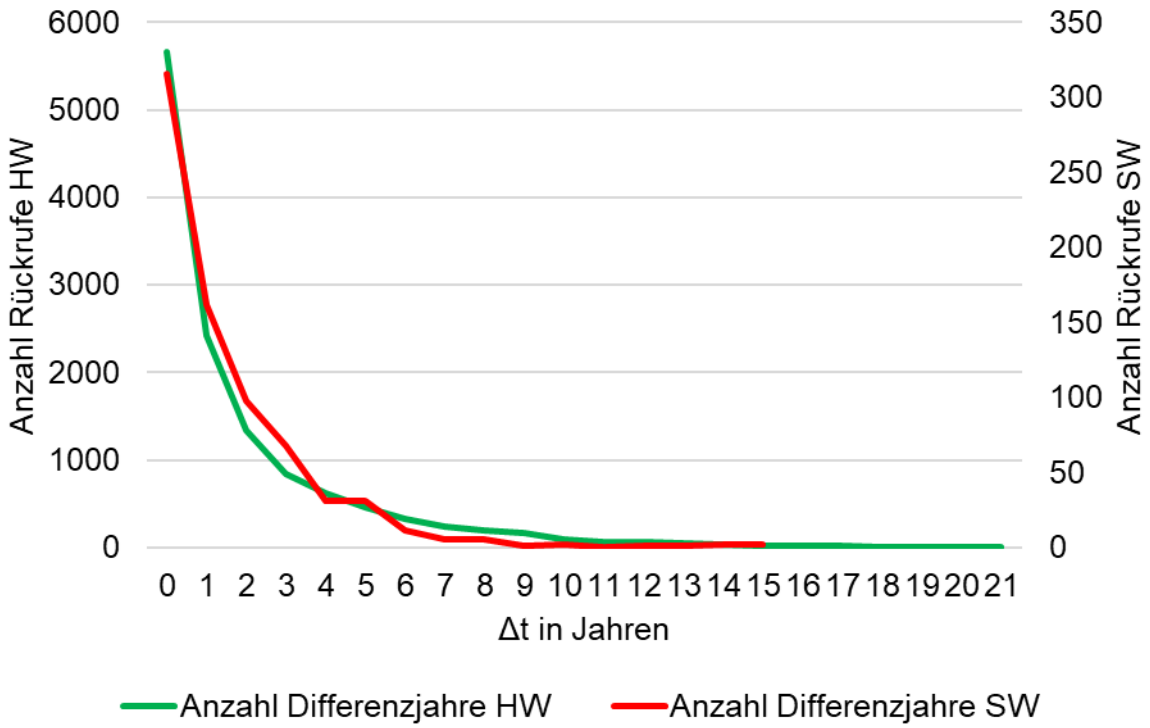

Abbildung 35: Differenz Rückrufjahr – Modelljahr

Ein genauerer Vergleich lässt sich mithilfe einer normierten Datenbasis treffen. Hierzu wird anhand des jeweils höchsten Wertes der Anzahl der Soft- und Hardware-Rückrufe eine Skala von 0 % bis 100 % festgelegt und für jedes Differenzjahr wird ein relativer Wert errechnet. Diese Darstellung findet sich in logarithmischer Darstellung in Abbildung 36.

In dieser Darstellung finden sich nun zwei Auffälligkeiten. Zunächst ist auffällig, dass in den ersten drei Jahren die Software-Kurve über der Hardware-Kurve liegt. Begründen lässt sich dies damit, dass sich Software-Fehler meist recht kurzfristig nach der Produktion des jeweiligen Fahrzeuges durch Fehlfunktionen bemerkbar machen.

Eine weitere, starke Auffälligkeit ist, dass die Software-Fehler nach etwa fünf Jahren einen Schnittpunkt mit der Hardware-Kurve haben und ab dort bis auf eine Ausnahme im 15. Jahr dauerhaft unter den Hardware-Fehlern liegen. Nach dieser Ausnahme existieren aber keine Software-Fehler, die mehr als 15 Jahre unauffällig blieben und somit fällt die Software-Kurve auf null.

Die Hardware-Kurve hingegen sinkt kontinuierlich ab, bleibt aber über null. Dies liegt daran, dass Hardware-Fehler zwar auch mit der Zeit immer unwahrscheinlich werden, aber insbesondere durch Korrosionsprobleme auch nach langer Zeit noch

sicherheitsrelevante Einschränkungen mitbringen können, die erst nach vielen Jahren auftauchen.

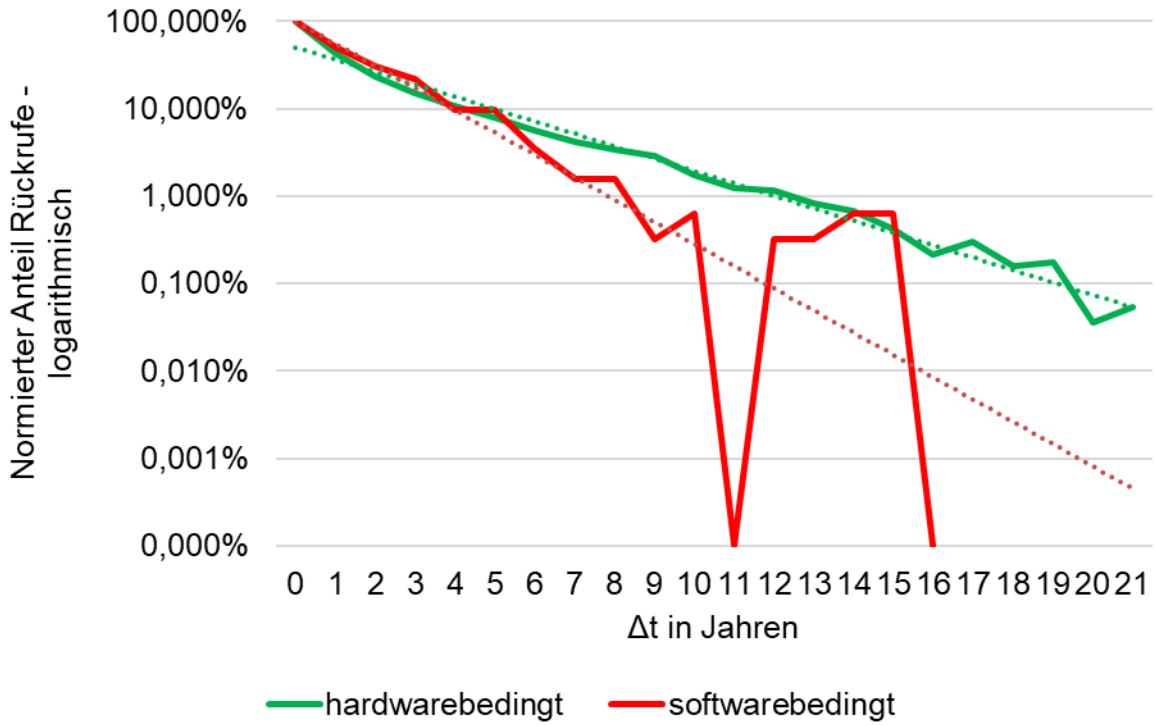

Abbildung 36: Normierte Differenz logarithmisch mit Trendlinie - Rückrufe

Noch wesentlich deutlicher wird der Unterschied in einer Betrachtung der Anzahl der betroffenen Fahrzeuge der jeweiligen Rückrufe. Die Darstellung in Abbildung 37 ist identisch zu Abbildung 36 – bis auf die Betrachtung der Fahrzeuganzahl anstatt der Rückrufanzahl. Auch hier wurden die Klassen Soft- und Hardware nach der höchsten absoluten Anzahl auf 100 % normiert.

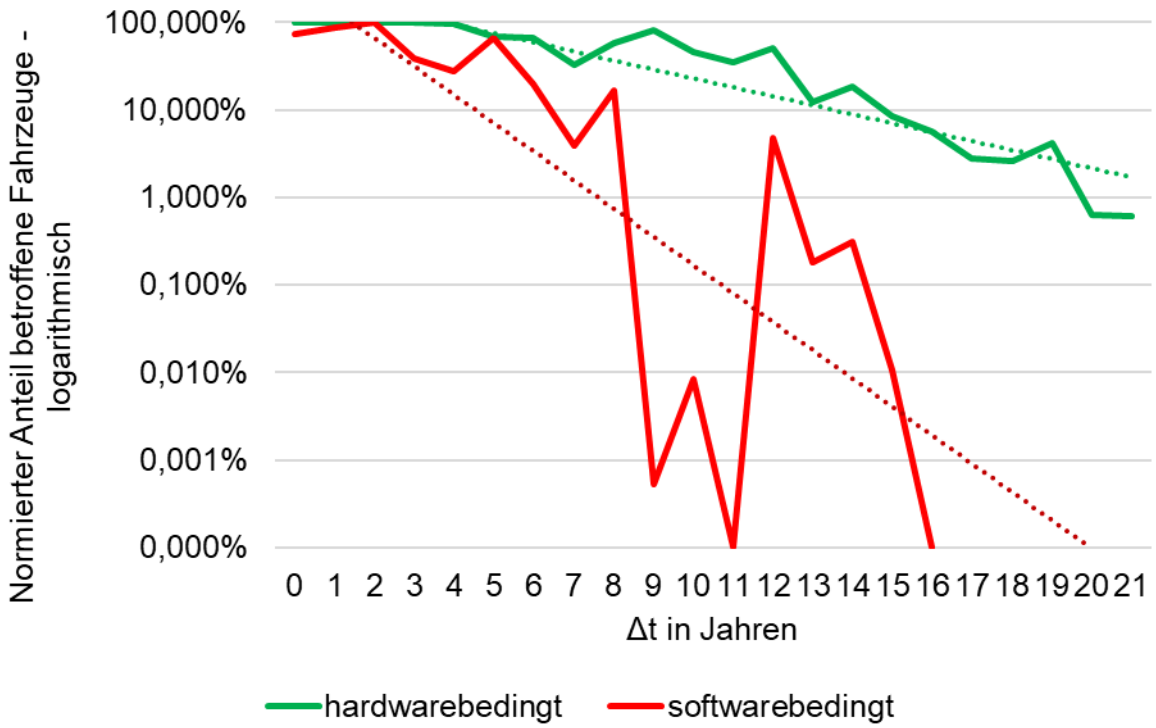

Abbildung 37: Normierte Differenz logarithmisch mit Trendlinie – Fahrzeuge

Ersichtlich ist nun ein sehr starker Steigungsunterschied zwischen den zwei Verläufen. Hier wird deutlich, dass Fahrzeuge aufgrund von Software-Fehlern wesentlich wahrscheinlicher in den ersten Jahren zurückgerufen werden, wohingegen Fahrzeuge mit Hardware-Fehlern auch nach längerer Zeit noch zurückgerufen werden. Dies bekräftigt erneut die These, dass bei Hardware-Rückrufen durch beispielsweise Korrosion auch nach vielen Jahren noch Rückrufe auftauchen, während bei Software die Fehler zumeist schnell auffällig werden und somit entdeckt werden können.

6.3.4. Diskussion der Auswertungen

Die Auswertungen der Rückrufe nach Hersteller in Abbildung 32, Abbildung 33 und Abbildung 34 sind kritisch zu betrachten. Durch den Datenbereinigungsprozess in 4.2.1 werden alle Einträge einer Rückrufkampagne zusammengefasst. Die Einträge sind bis auf teilweise Marke, Modell und Modelljahr vollständig identisch. Hierbei geht es dann um Rückrufe, die von einem Konzern durchgeführt wurden wie beispielsweise in dem in 6.3.1 beschriebenen Rückruf 16V651000 [54]. Hier rief der Konzern General Motors LLC alle Fahrzeuge mit der fehlerhaften Software auf dem

Airbag-Steuergerät zurück. Davon sind somit einige Marken des Konzerns betroffen. Allerdings wird es in der bereinigten Datenbank nur als Rückruf der Marke Chevrolet aufgeführt, da dieses Modell der Marke als erstes in der ursprünglichen Datenbank gelistet ist. Somit werden etwa 3,6 Millionen zurückgerufene Fahrzeuge der Marke Chevrolet zugeordnet, obwohl diese auf mehrere Marken des Konzerns GM aufgeteilt waren.
Dies würde sich allerdings auch mit einer anderen Art der Datenbereinigung nicht verhindern lassen, da die Anzahl der Fahrzeuge in den ursprünglichen Daten nicht nach Hersteller angegeben ist, sondern nur die Gesamtzahl pro Rückruf.

Aufgrund der Löschung der Duplikate beschreiben auch die historischen Daten nicht alle betroffenen Modelljahre, sondern es taucht pro Rückruf nur jeweils das erste betroffene Modelljahr auf. Da sich die Modelljahre eines Rückrufs aber meist nur im Rahmen von ein bis zwei Jahren bewegen, ist diese Schwankung eher vernachlässigbar.
Für beide Arten der Auswertung können die Daten aber als Tendenz zum Verlauf oder einer Korrelation mit dem vertreibenden Konzern genutzt werden.

Auch zu beachten ist es, dass die relativen Angaben in den Grafiken der Auswertung nach Hersteller sich nicht auf die Grundgesamtheit aller Fahrzeuge berufen, sondern lediglich auf die genannten 15 Hersteller.

7. Zusammenfassung und Ausblick

7.1. Fazit

Die Eingangsfrage der Arbeit, ob ein Einfluss von Software-Fehlern auf Fahrzeugrückrufe besteht, muss eindeutig bejaht werden. Allein aufgrund der Softwaremenge als auch deren Komplexität steigt die Wahrscheinlichkeit für Software-Fehler an. Da diese Software auch in sicherheitsrelevanten Baugruppen verwendet wird, werden damit einhergehend auch Rückrufe aufgrund von Software häufiger.

Die genaue Quantifizierung der softwarebedingten Rückrufe stellt sich in relativen Zahlen von etwa 1 % im Jahr 2000 bis zu etwa 12 % aller Rückrufe im Jahr 2022 dar.
Dieser Anstieg der Anzahl der Rückrufe spiegelt etwa die Erwartungshaltung wider, wenn der Anstieg der Softwaremenge und -komplexität betrachtet wird.
Allerdings ist die Anzahl der Rückrufe eher zweitrangig von Relevanz. Die Anzahl der Fahrzeuge, die aufgrund von Software-Fehlern zurückgerufen wurde, ist in dieser Betrachtung wesentlich relevanter. Eine genaue Quantifizierung beträgt etwa 0,02 % der Fahrzeuge des Modelljahres 2000 bis hin zu etwa 33 % aller Fahrzeuge im Modelljahr 2022 dar. Dieser enorme Anteil der Software-Fehler ist bedingt durch den Fakt, dass bei Software-Fehlern meistens alle Modelle oder sogar alle Steuergeräte mit der Software betroffen sind. Dies beläuft sich teilweise auf mehrere Millionen Fahrzeuge, wohingegen bei Hardware-Fehlern eher nur kleine Chargen oder sogar nur Fahrzeuge von einem Produktionstag betroffen sind.
Ein Drittel aller Fahrzeuge stellt eine enorme Menge und somit auch hohe Kosten für Hersteller dar.
In der Betrachtung der Differenz von Modell- zu Rückrufjahr fällt insbesondere auf, dass es in den ersten Jahren zwar anteilig mehr Fahrzeuge gibt, die aufgrund von Software zurückgerufen wurden. Allerdings fällt dies nach wenigen Jahren stark ab und sinkt auf einen Anteil von 0 %, wohingegen die Anzahl von Fahrzeugen mit Hardware-Problemen auch nach vielen Jahren noch hoch bleibt.
Der hohe und stetig steigende Anteil von softwarebedingten Rückrufen spiegelt die hohe Relevanz einer guten Software-Struktur wider. Zudem stellt es auch die Relevanz von Over-The-Air-Updates dar. Musste beispielsweise bei einem Rückruf

eines Bremsbauteils das Fahrzeug unbedingt in die Werkstatt gebracht werden, um das betroffene Bauteil auszutauschen, ist dies bei Software theoretisch nicht mehr nötig.
In diesem Bereich stellt sich Tesla als Vorreiter dar, der seine Software-Updates größtenteils über die Internetverbindung der Fahrzeuge ausrollt und den Kunden Werkstattbesuche erspart. Demzufolge ermöglichen sie trotz hoher Anzahl an zurückgerufenen Fahrzeugen eine hohe Kundenzufriedenheit.

Die methodische Anwendbarkeit von Machine Learning für die Kategorisierung von Fahrzeugrückrufen ist ebenfalls klar gegeben. Zu beachten ist jedoch, dass die Daten in einer verarbeitbaren Form vorliegen müssen und zudem die Datenqualität ausreichend gut sein muss, um wie in diesem Beispiel in bestimmte Kategorien (softwarebedingt/nicht softwarebedingt) einzuteilen.
Aufgrund der fehlenden Daten des KBA und der unzureichenden Qualität der Daten aus RAPEX werden nur die Daten der NHTSA ausführlich bearbeitet und ausgewertet.
Wie in dieser Arbeit gezeigt wird, sind je nach Datenlage und Art der Kategorisierung bestimmte Machine Learning-Verfahren besser geeignet als andere.
Im gegebenen Anwendungsfall der NHTSA-Daten eignet sich vor allem das gewichtete Clustering-Verfahren für eine erste Gruppierung von ungelabelten Daten.
Der Entscheidungsbaum liefert als überwachtes Verfahren mithilfe gelabelter Daten in kurzer Zeit eine effiziente Kategorisierung.
Das CNN stellt sich als das komplexeste Verfahren dar, das auch qualitativ die besten Ergebnisse liefert. Allerdings ist das CNN aufgrund der Komplexität ein sehr rechenaufwändiger Algorithmus und somit auch der langsamste. Dies sollte insbesondere bei der Arbeit mit noch größeren Datensätzen berücksichtigt werden. In der hier gegebenen Anwendung spielt die Rechenzeit jedoch eine untergeordnete Rolle.

7.2. Ausblick

Software ist zum jetzigen Zeitpunkt schon ein unverzichtbarer Bestandteil moderner Mobilität. Durch stetig steigende Komplexität der erforderlichen Funktionen und der Ansprüche von Kunden und Herstellern wird voraussichtlich auch zukünftig noch die Menge und insbesondere die Komplexität der Software in Fahrzeugen zunehmen. Insbesondere die geplante Entwicklung von Fahrzeugen zu Software Defined Vehicles – also einem durch Softwarefunktionalitäten definierten System – wird diesen Anstieg zusätzlich bestärken.
Hinzu kommt die auch weiterhin sinkende Anzahl der angebotenen Fahrzeugderivate der Hersteller und eine Vereinheitlichung der Software-Systeme in Fahrzeugen. Beispielsweise mit MB.OS [63] oder VW.OS [64] möchten große Automobilhersteller einheitliche Software-Plattformen für ihre Fahrzeugflotten einführen. Hierdurch werden mehr Fahrzeuge durch potenzielle Fehler in den Software-Strukturen gefährdet. Folglich ist es die Aufgabe der Industrie, die durch die Vereinheitlichung gesparten Kapazitäten nun in ein stabiles und ausgiebig erprobtes System zu investieren.
Weiterhin nicht zu vernachlässigen sind auch Qualitätssicherung und -management von aktueller Software-Entwicklung mithilfe von Normen wie Automotive SPICE [65], ISO 26262 [66] oder auch ISO/IEC 25010 [67].

Um die Auswirkung von Rückrufen auf Image und Kosten gering zu halten, ist es unverzichtbar, durch die bessere Vernetzung der Fahrzeuge eine einfache Wartbarkeit mithilfe von OTA-Updates zu ermöglichen. Dadurch fallen Rückrufe sowohl kostentechnisch beim Hersteller als auch imagemäßig beim Endkunden wesentlich weniger ins Gewicht. Auch die Schnelligkeit des Erreichens der Erfüllungsrate und demzufolge des sicheren Zustands der Fahrzeuge kann mithilfe solcher Updates wesentlich beschleunigt werden. Vor allem der Hersteller Tesla nutzt diese Technologie sehr ausgiebig und erreicht dementsprechend fast vollständige Erfüllungsraten innerhalb kürzester Zeit (beispielsweise 97 % in unter zwei Wochen [62]).

So viele Vorteile die Technologie der OTA-Updates mit sich bringt, bergen sich leider auch Gefahren darin. Durch die Einfachheit solcher Updates besteht die Gefahr,

aufgrund hoher Ausrollfrequenzen der Updates sicherheitsrelevante Mängel zu übersehen. Auch eine Kontrolle durch die Behörden fällt hierbei immer schwerer, da diese die Rückrufe und deren Erfüllungsrate prüfen müssen, bei einer immer steigenden Frequenz aber davon überrollt werden können [68]. Somit ist es die Aufgabe der Politik den Weg freizumachen, um die prüfenden Behörden auf den Anstieg der Ausrollfrequenzen von Software-Updates vorzubereiten.

Als zusammenfassende und abschließende Handlungsempfehlung lässt sich nun für Automobilhersteller die Empfehlung für strenge Qualitätssicherungsmaßnahmen und einen Ausbau zuverlässiger OTA-Update-Strukturen aussprechen.
Die zuständigen Behörden und die darüberstehende Politik sollte sich auf den aktuell bestehenden und in Zukunft noch mehr verstärkten Anstieg von softwarebedingten Fahrzeugrückrufen einstellen, um diese auch weiterhin sinnvoll überwachen zu können, sodass die Sicherheit im Straßenverkehr gewährleistet werden kann.

Für weitere Ausarbeitungen dieser Arbeit ergeben sich unterschiedliche Möglichkeiten für Folgearbeiten.
Zum einen wäre es interessant in einigen Jahren mithilfe der trainierten CNN-Modelle die aktuellen Daten der NHTSA zu beziehen und auszuwerten. Im Anschluss könnten die Vorhersagen aus Tabelle 13 und Tabelle 15 validiert oder widerlegt und angepasst werden. In diesem Zusammenhang wäre eine Auswertung interessant, die zeigt, wie gut sich eine einfache lineare Extrapolation der Daten aus dieser Arbeit in der Realität bewährt.
Darüber hinaus könnten Methoden des maschinellen Lernens zur weiteren qualitativen Untersuchung der softwarebedingten Rückrufe eingesetzt werden. Hier könnten die genauen Ursachen der Software-Fehler (Konfigurationsmanagement, falsche Parametrierung, Kalibrierungsproblem, etc.) anhand der Daten der NHTSA durchgeführt werden. Zusätzlich könnten die Part 573-Berichte der NHTSA, die den Rückrufen beigefügt sind, mit einbezogen werden.
Aus den Part 573-Berichten ist zumeist auch der Zulieferer des betroffenen Bauteils oder eben der Software ersichtlich. Wenn die Auswertung anhand dieser Berichte erfolgt, wäre hier auch die Auswertung der Zulieferer sehr interessant.

Ein weiterer Aspekt wäre die Betrachtung des deutschen Marktes mithilfe der Daten des KBA. Allerdings bedarf es hierzu einer Bereitstellung der Daten des KBA, welche in naher Zukunft nicht absehbar ist.
Alternativ könnten auch Daten des ADAC bezogen werden, falls hier eine Einigung mit dem ADAC bezüglich der Lizenzen möglich ist.

Literatur

[1] Mercedes-Benz Group. „Mercedes-Benz Drive Pilot.” https://group.mercedes-benz.com/innovation/case/autonomous/drive-pilot.html (Zugriff am: 6. August 2023).

[2] S. Kochanthara, Y. Dajsuren, L. Cleophas und M. van den Brand, „Painting the landscape of automotive software in GitHub,“ in *Proceedings of the 19th International Conference on Mining Software Repositories*, Pittsburgh Pennsylvania, D. Lo, S. McIntosh und N. Novielli, Hg., 2022, S. 215–226, doi: 10.1145/3524842.3528460.

[3] McKinsey, „Umsatzprognose der weltweiten Automobilindustrie im Bereich Software sowie elektrische und elektronische Bauteile nach Segmenten in den Jahren 2020 bis 2030 (in Milliarden US-Dollar),“ *Statista*, 2019. [Online]. Verfügbar unter: https://de.statista.com/statistik/daten/studie/1047098/umfrage/umsatzprognose-im-bereich-automotive-software-und-elektronik/

[4] Kraftfahrt-Bundesamt, „Marktüberwachungsbericht 2021,“ 2022. [Online]. Verfügbar unter: https://www.kba.de/DE/Themen/Marktueberwachung/Allgemein/Marktueberwachungsbericht/marktueberwachungsbericht_2021.pdf?__blob=publicationFile&v=7

[5] T. Bodewig, *Der Rückruf fehlerhafter Produkte: Eine Untersuchung der Rückrufpflichten und Rückrufansprüche nach dem Recht Deutschlands, der Europäischen Union und der USA* (Jus Privatum Bd. 36). Tübingen: Mohr Siebeck, cop, 1999. [Online]. Verfügbar unter: https://books.google.de/books?id=9j_UwTZT8iQC

[6] Kraftfahrt-Bundesamt, „Kodex zur Durchführung von Rückrufaktionen,“ 2021. [Online]. Verfügbar unter: https://www.kba.de/DE/Themen/Marktueberwachung/Rueckrufe/Kodex/kodex_pdf.pdf?__blob=publicationFile&v=2

[7] Amtsblatt Europäische Union, „Amtsblatt der EU 2004 L 381,“ 2004. [Online]. Verfügbar unter: https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=OJ:L:2004:381:FULL&from=SK

[8] Kraftfahrt-Bundesamt. „Marktüberwachung KBA.” https://www.kba.de/DE/Themen/Marktueberwachung/Allgemein/marktueberwachung_allgemein_node.html;jsessionid=0C8C92C0800EC14815743C1B63EE3931.live21323 (Zugriff am: 11. April 2023).

[9] BVL. „Das Europäische Schnellwarnsystem für Verbraucherprodukte (RAPEX).” https://www.bvl.bund.de/DE/Arbeitsbereiche/01_Lebensmittel/01_Aufgaben/04_Warn_und_Informationssysteme/00_Warn_und_Informationssysteme/03_RAPEX/RAPEX_node.html (Zugriff am: 5. April 2023).

[10] European Commission, „Business Gateway contacts,“ 2023. [Online]. Verfügbar unter: https://ec.europa.eu/safety/consumers/consumers_safety_gate/menu/documents/Business_Gateway_contacts.pdf

[11] NHTSA. „Resources Related to Investigations and Recalls | NHTSA.” https://www.nhtsa.gov/resources-investigations-recalls#recalls (Zugriff am: 5. April 2023).

[12] NHTSA, „Motor Vehicle Safety Defects and Recalls: What Every Vehicle Owner Should Know,“ 2017. [Online]. Verfügbar unter: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/14218-mvsdefectsandrecalls_041619-v2-tag.pdf

[13] B.J. Copeland. „Artificial intelligence (AI) | Definition, Examples, Types, Applications, Companies, & Facts.” https://www.britannica.com/technology/artificial-intelligence (Zugriff am: 6. April 2023).
[14] M. Copeland. „The Difference Between AI, Machine Learning, and Deep Learning?” https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/ (Zugriff am: 4. August 2023).
[15] W. Ertel, *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung* (Computational Intelligence). Wiesbaden: Springer Vieweg, 2016.
[16] Jörg Frochte, *Maschinelles Lernen Grundlagen und Algorithmen in Python*, 2019.
[17] U. Bankhofer und J. Vogel, *Datenanalyse und Statistik: Eine Einführung für Ökonomen im Bachelor* (Lehrbuch). Wiesbaden: Betriebswirtschaftlicher Verlag Dr. Th. Gabler / GWV Fachverlage GmbH, Wiesbaden, 2008.
[18] IBM. „What is Data Labeling?” https://www.ibm.com/topics/data-labeling (Zugriff am: 9. August 2023).
[19] IBM. „Hyperparameter optimization and tuning.” https://www.ibm.com/docs/en/wmla/2.2.0?topic=training-hyperparameter-optimization (Zugriff am: 9. August 2023).
[20] R. S. Sutton und A. Barto, *Reinforcement learning: An introduction* (Adaptive computation and machine learning). Cambridge, Massachusetts, London, England: The MIT Press, 2018.
[21] I. Goodfellow, Y. Bengio und A. Courville, *Deep learning* (Adaptive computation and machine learning). Cambridge, Massachusetts: The MIT Press, 2016. [Online]. Verfügbar unter: https://www.deeplearningbook.org/
[22] V. Choubey, „Text classification using CNN - Voice Tech Podcast - Medium,“ *Voice Tech Podcast*, 08. Juli 2020. https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9 (Zugriff am: 14. Juli 2023).
[23] ADAC, „ADAC Pannenstatistik 2021,“ 2021. [Online]. Verfügbar unter: https://assets.adac.de/image/upload/v1651050882/ADAC-eV/KOR/Text/PDF/Pannenstatistik2021_iy7knx.pdf
[24] J.D. Power, „U.S. Vehicle Dependability Study,“ Feb. 2023. [Online]. Verfügbar unter: https://www.jdpower.com/sites/default/files/file/2023-02/2023011%20U.S.%20VDS.pdf
[25] J. Pancik, A. Vemola, R. Kledus, M. Semela und A. Bradac, „Auto recalls and software quality in the automotive sector,“ *ICST Transactions on Scalable Information Systems*, Jg. 5, Nr. 17, S. 1–8, 2018. doi: 10.4108/eai.29-5-2018.154808. [Online]. Verfügbar unter: https://eudl.eu/doi/10.4108/eai.29-5-2018.154808
[26] Yongqin, F. Xiaorui, Z. Le Liu und W. Li, „Analysis on the Trends and Characteristics of Vehicle Recalls in the United States,“ in *Proceedings of the 2021 International Conference on Culture, Design and Social Development (CDSD 2021)*, Harbin, China, 2022, doi: 10.2991/assehr.k.220109.001.
[27] L. Lanxiang, G. Li, C. Zhihui und H. Chunsong, „ResHDrch of classification for defective components of automotive recall based on clustering algorithm,“ in *2011 Chinese Control and Decision Conference (CCDC)*, Mianyang, China, 2011, S. 4307–4310, doi: 10.1109/CCDC.2011.5968983.
[28] NHTSA. „NHTSA Datasets and APIs | NHTSA.” https://www.nhtsa.gov/nhtsa-datasets-and-apis (Zugriff am: 25. April 2023).
[29] J. G. Ronquillo und D. M. Zuckerman, „Software-Related Recalls of Health Information Technology and Other Medical Devices: Implications for FDA

Regulation of Digital Health," *The Milbank quarterly*, Jg. 95, Nr. 3, S. 535–553, 2017. doi: 10.1111/1468-0009.12278. [Online]. Verfügbar unter: https://onlinelibrary.wiley.com/doi/10.1111/1468-0009.12278

[30] Klaus Rüdiger Bährle, „Marktanalyse zu sicherheitskritischer Software in der Automobilbranche," Lehrstuhl für Entrepreneurship, Technologie-Management und Innovation (EnTechnon), Karlsruher Institut für Technologie, Karlsruhe, 2016. [Online]. Verfügbar unter: https://baldur.iti.kit.edu/qpr/marktanalyse-baehrle.pdf

[31] J. H. Becker, H. Ebert und S. Pastoors, *Praxishandbuch berufliche Schlüsselkompetenzen: 50 Handlungskompetenzen für Ausbildung, Studium und Beruf* (SpringerLink Bücher). Berlin: Springer, 2018.

[32] European Commission. „Safety Gate: the EU rapid alert system for dangerous non-food products: Free text search." https://ec.europa.eu/safety-gate-alerts/screen/search?resetSearch=true (Zugriff am: 2. Mai 2023).

[33] The pandas development team, *Pandas*. Zenodo, 2023. [Online]. Verfügbar unter: https://zenodo.org/record/7857418#.ZGzZiXZBwQ8

[34] NHTSA. „Recall 08V397000." https://www.nhtsa.gov/recalls?nhtsaId=08V397000 (Zugriff am: 9. August 2023).

[35] Matthew Mayo. „Text Data Preprocessing: A Walkthrough in Python - KDnuggets." https://www.kdnuggets.com/2018/03/text-data-preprocessing-walkthrough-python.html (Zugriff am: 5. Mai 2023).

[36] Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python: NLTK : Natural Language Toolkit*. O'Reilly Media Inc., 2009. Zugriff am: 23. Mai 2023. [Online]. Verfügbar unter: https://www.nltk.org/

[37] T. Shah, „About Train, Validation and Test Sets in Machine Learning," *Towards Data Science*, 06. Dezember 2017. https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7 (Zugriff am: 7. Juni 2023).

[38] Caner Kilinc, „Padding for NLP," 02. April 2020. https://medium.com/@caner-kilinc/padding-for-nlp-7dd8598c916a (Zugriff am: 14. Juli 2023).

[39] TensorFlow Keras. „TensorFlow Tokenizer v2.13.0." https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer (Zugriff am: 17. Juli 2023).

[40] TensorFlow, „Understanding masking & padding," https://www.tensorflow.org/guide/keras/understanding_masking_and_padding (Zugriff am: 9. August 2023).

[41] M. Dwarampudi und N. V. S. Reddy, „Effects of padding on LSTMs and CNNs," Mrz. 2019. Zugriff am: 8. August 2023. [Online]. Verfügbar unter: https://arxiv.org/pdf/1903.07288.pdf

[42] S. Lloyd, „Least squares quantization in PCM: KMeans Reference," *IEEE Trans. Inform. Theory*, Jg. 28, Nr. 2, S. 129–137, 1982, doi: 10.1109/TIT.1982.1056489.

[43] K. Backhaus, B. Erichson, S. Gensler, R. Weiber und T. Weiber, „Clusteranalyse," in *Multivariate Analysemethoden*, K. Backhaus, B. Erichson, S. Gensler, R. Weiber und T. Weiber, Hg., Springer Gabler, Wiesbaden, 2021, S. 489–575. [Online]. Verfügbar unter: https://link.springer.com/chapter/10.1007/978-3-658-32425-4_8

[44] Hannes Hinrichs. „Der K-Means Algorithmus - Einfache und intuitive Erklärung (Deutsch/German)." https://www.youtube.com/watch?v=Gn6fPYD1oIU (Zugriff am: 5. Juli 2023).

[45] J. Chouinard, „What is KMeans Clustering Algorithm (with Example) – Python,“ *Jean-Christophe Chouinard*, 05. Mai 2022. https://www.jcchouinard.com/kmeans/ (Zugriff am: 7. Juni 2023).
[46] S. Kumar, „Understanding K-Means, K-Means++ and, K-Medoids Clustering Algorithms,“ *Towards Data Science*, 11. Juni 2020. https://towardsdatascience.com/understanding-k-means-k-means-and-k-medoids-clustering-algorithms-ad9c9fbf47ca (Zugriff am: 6. Juli 2023).
[47] scikit-learn. „Decision Trees.” https://scikit-learn.org/stable/modules/tree.html (Zugriff am: 19. Juli 2023).
[48] scikit-learn. „RandomForestClassifier.” https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (Zugriff am: 19. Juli 2023).
[49] A. Sharma, „Random Forest vs Decision Tree | Which Is Right for You?,“ *Analytics Vidhya*, 11. Mai 2020. https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/ (Zugriff am: 19. Juli 2023).
[50] IBM. „Was ist ein Entscheidungsbaum.” https://www.ibm.com/de-de/topics/decision-trees (Zugriff am: 27. Juli 2023).
[51] Keras Team. „Keras documentation: GlobalMaxPooling1D layer.” https://keras.io/api/layers/pooling_layers/global_max_pooling1d/ (Zugriff am: 18. Juli 2023).
[52] TensorFlow. „TensorFlow Embedding v2.13.0.” https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding (Zugriff am: 19. Juli 2023).
[53] European Commission. „Safety Gate: the EU rapid alert system for dangerous non-food products Alert number: 1502/10.” https://ec.europa.eu/safety-gate-alerts/screen/webReport/alertDetail/-17772 (Zugriff am: 13. Juni 2023).
[54] NHTSA. „Recall 16V651000.” https://www.nhtsa.gov/recalls?nhtsaId=16V651000 (Zugriff am: 27. Juli 2023).
[55] NHTSA. „Recall 15V461000.” https://www.nhtsa.gov/recalls?nhtsaId=15V461000 (Zugriff am: 27. Juli 2023).
[56] NHTSA. „Recall 17V302000.” https://www.nhtsa.gov/recalls?nhtsaId=17V302000 (Zugriff am: 27. Juli 2023).
[57] NHTSA. „Recall 14V138000.” https://www.nhtsa.gov/recalls?nhtsaId=14V138000 (Zugriff am: 27. Juli 2023).
[58] Writing Science. „Mittelwert, Median und deskriptive Statistik – Writing Science.” https://writing-science.com/ghostwriter-statistik/mittelwert-median-statistische-lagermasse/ (Zugriff am: 10. August 2023).
[59] ADAC. „Tesla: Alle Modelle, Infos und News.” https://www.adac.de/rund-ums-fahrzeug/autokatalog/marken-modelle/tesla/?filter=ONLY_RECENT&sort=SORTING_DESC (Zugriff am: 27. Juli 2023).
[60] Statista. „U.S.: best-selling car brand 2022.” https://www.statista.com/statistics/264362/leading-car-brands-in-the-us-based-on-vehicle-sales/ (Zugriff am: 27. Juli 2023).
[61] NHTSA. „Recall 22V818000.” https://www.nhtsa.gov/recalls?nhtsaId=22V818000 (Zugriff am: 3. August 2023).
[62] Ansley, Alexander (NHTSA), „Recall 22V818000 Part 573 Report,” [Online]. Verfügbar unter: https://static.nhtsa.gov/odi/rcl/2022/RCLRPT-22V818-6624.PDF
[63] Mercedes-Benz Group. „Mercedes-Benz Operating System (MB.OS) | Mercedes-Benz Group.” https://group.mercedes-benz.com/karriere/ueber-uns/mercedes-benz-operating-system/ (Zugriff am: 21. Juli 2023).

[64] CARIAD. „VW.os: The heart of our tech stack.” https://cariad.technology/de/en/news/stories/vw-os-software-platform-explained.html (Zugriff am: 21. Juli 2023).
[65] VDA. „Automotive SPICE® | Home.” https://www.automotivespice.com/ (Zugriff am: 3. August 2023).
[66] ISO. „ISO 26262-1:2018.” https://www.iso.org/standard/68383.html (Zugriff am: 3. August 2023).
[67] ISO. „ISO/IEC 25010:2011.” https://www.iso.org/standard/35733.html (Zugriff am: 3. August 2023).
[68] J. Rest, „Angst im Autopilot-Team: Wie Elon Musks Mär vom autonomen Fahren zusammenbricht,“ *manager magazin*, 20. Juli 2023. https://www.manager-magazin.de/unternehmen/autoindustrie/tesla-elon-musks-autopilot-schwindel-bricht-zusammen-a-717bf9bf-295c-473d-a3b4-9c5561d75468 (Zugriff am: 3. August 2023).