

Anomalieerkennung in Kommunikationsdaten zur Datenselektion im Fahrzeug

Zur Erlangung des akademischen Grades eines
DOKTOR-INGENIEURS
von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

genehmigte
DISSERTATION
von

M. Sc. Julia Hofmockel
geb. in Ansbach

Tag der mündlichen Prüfung: 21. Mai 2019
Referent: Prof. Dr.-Ing. Eric Sax
Korreferent: Prof. Dr. Alexander Pretschner

Kurzfassung

Ein Fahrzeug generiert Daten, welche Zustand und Verhalten von Fahrer und Fahrzeug beschreiben. Das Sammeln der Informationen gibt dem Automobilhersteller die Möglichkeit, diese als Big Data zu verwenden und neuen Wert zu schöpfen. Beispielsweise schafft die Beobachtung während des gesamten Lebenszyklus die Grundlage für eine Produktoptimierung, oder aber das Marketing kann zielgerichtet auf die individuellen Wünsche des Kunden eingehen.

Herausforderungen hierbei liegen darin, einen Wert aus den Daten zu schöpfen und die Übertragung von Fahrzeug zu Backend reduzieren. Ein Oberklassefahrzeug generiert im Jahr 2017 2,1 MByte je Sekunde. Für die Übertragung der Datenmenge, die eine Flotte mit Millionen von Fahrzeugen erzeugt, ergeben sich daher Schwierigkeiten. Aufgrund der Übertragungskosten, der Datenschutzgrundverordnung und der Limitierung in der Datenübertragung ist bereits im Fahrzeug eine Selektion relevanter Informationen notwendig.

Der Ansatz der *one-class* Klassifizierung lernt anhand normaler Flottendaten, was normal ist und kann somit Abweichungen vom Normalzustand erkennen. Durch die Übertragung des trainierten Modells in die Fahrzeuge werden dort unmittelbar anormale Ereignisse erkannt, womit der Datentransfer zum Backend auf das Wesentliche, die außergewöhnlichen Vorkommnisse, reduziert wird. Durch eine stetige Aktualisierung des Modells entsteht eine dynamische Datensammlung, welche sicherstellt, dass im Backend noch fehlende Ereignisse übermittelt werden.

Der Vergleich unterschiedlicher Ansätze verdeutlicht, dass insbesondere das Replikator Neuronale Netz die Anforderungen erfüllt. Außerdem ist es in der Lage, einen Großteil real geschehener Anomalien zu erkennen, während kein normales Event fälschlicherweise als anormal eingestuft wird. Das Ziel der Datenreduktion um das Hundertfache führt zur tatsächlichen Einschränkung auf 0,715%. Die bedeutsamsten Ereignisse, wie Unfälle oder ABS-Eingriffe, sind klar erkennbar, doch eine durchschnittliche *Area under curve* von 0,8 legt dar, dass das Vorgehen kein Alarmsystem liefert. Es dient der Reduktion der Datenmenge, die ins Backend übertragen wird, wo weiteres Postprocessing erfolgen muss. Dieses besteht aus der Interpretation und der Einordnung von erkannten Ereignissen.

Abstract

A vehicle generates data describing its condition and the driver's behaviour. Collecting these information enables the producer to use it as big data and to add value. For example, the product can be optimized, because its health condition can be monitored during complete life time. Or marketing is responding to customers by knowing his or her individual wishes.

Challenge in this context is the generation of value out of the data and the transfer from vehicle to backend. A vehicle from the high-class segment alone produces 2.1 mbyte per second. A complete fleet with millions of cars therefore, produces an amount of data, which needs to be preselected in the vehicle. Due to the costs, data protection and limitation in the transfer, not all data can be transmitted.

The concept of one class-classification uses the fleet data to learn what is normal. With the learnt model deviations can be detected as anomalies. The model is sent into the vehicles where the transfer to the backend can directly be limited to the most relevant and unusual events. A continuous update of the model guarantees a dynamic collection of data. Only missing events are sent.

When comparing different approaches, the replikator neural network is the method, which meets the requirements. It is able to detect the majority of the real anomalies, while none of the normal events is erroneously predicted as abnormal. The aim of reducing the amount of data by a factor on hundred leads to a real restriction of 0.715%. The most meaningful events, like accidents or ABS interventions, are unambiguously detectable, but the average area under curve of 0.8 shows, that the approach can not be used as alarm

system in the vehicle. It is suitable to reduce the amount of data, transferred from vehicle to backend where further postprocessing is necessary. It must consist of the interpretation and the classification of the detected events.

Danksagung

Viele Leute, denen ich hiermit danken möchte, haben die Zeit meiner Doktorarbeit geprägt. Besonders danke ich meinem Doktorvater Herrn Prof. Dr.-Ing. Eric Sax für die gute Betreuung und Unterstützung. Seine wertvollen Anregungen haben nicht nur zum Gelingen dieser Arbeit beigetragen, sondern mich auch auf kommende Herausforderungen vorbereitet.

Herrn Prof. Dr. Alexander Pretschner danke ich für das Zweitgutachten und die engagierte Diskussion der Arbeitsinhalte. Zudem danke ich der Prüfungskommission, bestehend aus Herrn Prof. Dr.-Ing. Braun, Herrn Prof. Dr.-Ing. Powalla und Herrn Prof. Dr.-Ing. Leibfried. Roland Pfänder danke ich für die Gelegenheit zur Promotion in seiner Abteilung.

In den vergangenen Jahren in der AEV habe ich viel gelernt und mir hat es viel Freude bereitet, dort zu arbeiten. Gründe waren neben den erhellenden Runden mit Stefan und Florian auch die tolle Zusammenarbeit mit Ejaz und Dominik. Schön war auch der Austausch mit den Kollegen, insbesondere die anormalen Ausfahrten mit Daniel und die Überlegungen mit Lukas.

Ich möchte mich bei den ITIV- und FZI-Kollegen dafür bedanken, dass sie mich bei unseren Doc-Seminaren immer herzlich aufgenommen haben. Auch danke ich Johannes vom FAST für die gute Zusammenarbeit.

Am Wichtigsten, meine Familie und mein Freund Joschka. Danke für euer Verständnis und euren Rückhalt, ohne euch wäre die Promotion für mich nicht möglich gewesen.

Nürnberg, im Juni 2019

Julia Hofmockel

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung und Zielsetzung	8
1.1.1	Erkennen wertvoller Daten	9
1.1.2	Selektion wertvoller Daten	12
1.2	Wissenschaftlicher Beitrag	15
1.3	Struktur der Arbeit	18
2	Grundlagen und Definitionen	21
2.1	Datengrundlage	21
2.2	Erkennung von Anomalien mithilfe maschinellen Lernens	26
3	Konzept der Anomalieerkennung zur dynamischen Sammlung von Kommunikationsdaten	35
3.1	Ablauf Anomalieerkennung	35
3.2	Anomalieerkennung zur Datenselektion	37
3.3	Anforderungen	40
4	Umsetzung der Anomalieerkennung	45
4.1	Verfahren zur Anomalieerkennung	45
4.1.1	Statische Anomalieerkennung	47
4.1.2	Anomalieerkennung in multivariaten Zeitreihen	75
4.1.3	Anwendungsüberblick	80
4.2	Postprocessing	85
4.2.1	Entscheidung anhand des Anomalie Scores	85
4.2.2	Evaluationskriterien für Anomalien	89

4.2.3	Unterstützung bei der Interpretation einer Anomalie	93
5	Prototypische Anwendung	95
5.1	Datenüberblick	96
5.2	Evaluationsvorgehen	103
5.3	Experimente	109
5.3.1	Methodenvergleich	109
5.3.2	Bewertung der Anomalieerkennung	113
5.3.3	Synthetische Testfälle	120
5.3.4	Beispielhaftes Nachvollziehen erkannter Anomalien	131
6	Anwendungsfall: Erkennung von Getriebebeschäden	135
6.1	Modelloptimierung	138
6.1.1	Netzwerkarchitekturen und Topologien	140
6.1.2	Anpassung der Hyperparameter	141
6.2	Verwendung des optimalen Modells	144
7	Zusammenfassung und Ausblick	147
A	Prototypische Anwendung	155
A.1	Vergleich Trainings- und Kontrolldaten	155
A.2	Hyperparameter Isolation Forest	156
A.3	Synthetische Testfälle	156
A.4	Nachvollziehbarkeit erkannter Anomalien	156
B	Evaluierung Getriebebeschaden	161
	Abkürzungsverzeichnis	163
	Abbildungsverzeichnis	165
	Tabellenverzeichnis	169
	Literaturverzeichnis	172

1 Einleitung

It is a capital mistake to theorize before one has data. Insensibly, one begins to twist the facts to suit theories, instead of theories to suit facts.

– Sherlock Holmes –

Sherlock Holmes erkannte die Grundidee von *Big Data* bereits im Jahr 1891. Darunter versteht man große Datenmengen, aus denen Wissen extrahiert werden kann. Heutzutage findet Big Data größtenteils in den sozialen Medien statt. Soziale Netzwerke, *Online Shopping*, Kommentare und *Likes* auf Facebook oder Twitter sind Beispiele, in denen täglich Millionen von Datensätzen im Internet produziert werden [Gup15]. Der Einzelhandel benutzt Big Data bereits seit Anfang der 1990er zur Analyse von Korrelationen im Kaufverhalten der Kunden. Dadurch werden die Platzierungen einzelner Produktgruppen optimiert [Aut13].

Beschrieben werden kann Big Data durch 5 Vs: *Volume*, *Velocity*, *Variety*, *Veracity* und *Value* (vergleiche Bild 1.1). Das **V**olumen steht für die Verarbeitung enormer Datenmengen (Tera-/Peta-/Zeta Bytes an Daten). Die Tatsache, dass sich die Datenbasis im Laufe der Zeit vergrößert und ändert, wird durch *Velocity* abgebildet. Die *Variety* beschreibt die Mischung aus strukturierten und unstrukturierten Daten (z.B. Bilder, Audio, Videos), *Veracity* die Richtigkeit und Verfügbarkeit der Daten. *Value* bedeutet, dass aus den Daten nur Wert geschöpft werden kann, wenn sie vorverarbeitet, analysiert und visualisiert werden [IA15, BCD16].

Auch in der Automobilbranche spielt die Verarbeitung großer Datenmengen für die Entwicklung neuer Funktionen eine einnehmende Rolle. Seit 1980 steigen die Anzahl und Komplexität elektronischer Funktionen, da diese ins-

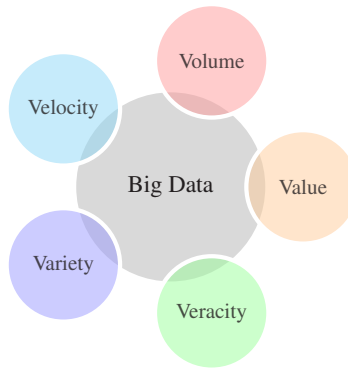


Bild 1.1: Big Data beschrieben durch 5 Vs

besondere den Marktwert des Fahrzeugs bestimmen [Gae18, EL12]. Beginnend mit Motorsteuergeräten oder Airbags haben sich die heutigen Funktionen hin zu den Themen Sicherheit und Infotainment entwickelt. Applikationen wie Parklenkassistent, Totwinkelüberwachung oder Spurhalteassistent wurden seit dem Jahr 2005 realisiert [Ver15]. Dieses Wachstum führt zur Zunahme miteinander kommunizierender Systeme und somit zu einer Steigerung der generierten Datenmenge [Ver17, EL12].

Fahrzeugintern tauschen Steuergeräte Informationen bezüglich Zustand und Verhalten von Fahrer und Fahrzeug über Bussysteme aus. Studien und Prognosen bezüglich der damit verbundenen Datenmenge zeigen allerdings Unterschiede. Im Jahr 2013 wird eine Zunahme der Daten, die ein Fahrzeug generiert, von 15 MByte pro Sekunde auf bis zu 350 MByte pro Sekunde im Jahr 2020 prophezeit [IBM14, tel13]. Hierbei beeinflussen Nutzungsverhalten und die Anzahl der Applikationen im Fahrzeug die Datenmenge [tel13]. Im Jahr 2015 werden für das Jahr 2020 215 MByte Fahrzeugdaten pro Stunde erwartet [Ant16]. Elmar Frickenstein, Bereichsleiter für vollautomatisiertes Fahren und Fahrerassistenz bei der BMW Group, spricht von

16 bzw. 40 Terabyte in acht Stunden, die hochautomatisierte bzw. vollautomatisierte Fahrzeuge generieren (vgl. Tabelle 1.1) [Spr18].

Bild 1.2 zeigt beispielhaft, wie das Datenvolumen von der Zeit bzw. Fahrzeugklasse abhängt. Je neuer ein Fahrzeug bzw. je höher das Segment, desto mehr Daten werden zwischen den Steuergeräten ausgetauscht. Damit verbunden ist der steigende Anteil in Software realisierter Funktionen. Die berechneten Datenmengen ergeben sich aus der Anzahl an Bussystemen und deren maximale Übertragungsraten.¹

In einem vernetzten Fahrzeug kommunizieren nicht nur die intern bestehenden Systeme miteinander. Vielmehr ist durch den Einzug des Internets ins Auto der Austausch mit anderen Fahrzeugen, der Infrastruktur und weiteren Verkehrsmitteln möglich [SMG⁺17]. Dadurch können Echtzeitanwendungen realisiert werden [BJH17]. Beispielsweise wird infolgedessen der Fahrer, noch bevor das Pannenfahrzeug oder das Glatteis zu erkennen ist, gewarnt. Städte oder Straßenmeistereien nutzen die Daten zur Lokalisierung

¹ berechnet aus CAN (Controller Area Network) mit maximaler Übertragungsrates von 1 MBit/s, LIN (Local Interconnect Network) mit maximaler Übertragungsrates von 20 KBit/s und FlexRay mit maximaler Übertragungsrates von 10 MBit/s [Dec13].

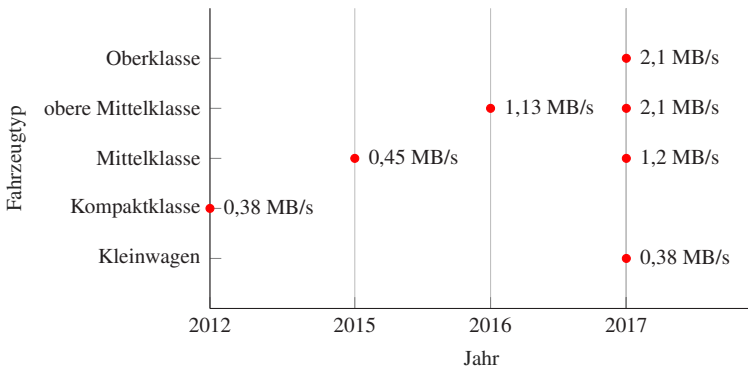


Bild 1.2: Datenaustausch zwischen den Steuergeräten in Megabyte (MB) pro Sekunde bei einer Busauslastung von 60%, Abhängigkeit von Fahrzeugtyp und Zeit

rung von zerborstenen Fahrbahnen oder Schlaglöchern [Gru16]. Mit dem für 2020 geplanten 5G-Netz, mit Übertragungsraten von mehreren Gigabytes pro Sekunde, wird dann auch automatisiertes Fahren möglich sein [NM17, Ver15]. Hierbei gilt es zwischen *Driver only*, *assistiertem*, *teilautomatisiertem*, *hochautomatisiertem*, *vollautomatisiertem* und *fahrerlosem* Fahren zu unterscheiden (vgl. Tabelle 1.1). Bei *Driver only*-Funktionen ist kein eingreifendes Fahrzeugsystem aktiv (z.B. Totwinkelüberwachung, Spurwarnassistent) [Ver15]. Anders ein *assistierendes* Fahrzeug, bei dem der Fahrer jedoch nicht davon befreit wird, die Situation ständig zu überwachen und gegebenenfalls einzugreifen (z.B. Abstandsregeltempomat im Jahr 2000). Fahrerassistenzfunktionen wie Stau- oder Parkmanöverassistent (Jahr 2015) lassen sich in die Kategorie *teilautomatisiert* einordnen. Beim automatisierten Fahren auf der Autobahn spricht man vom *hochautomatisierten* Fahren. Ein *vollautomatisiertes* Fahrzeug ist in der Lage, die Fahraufgabe in speziellen Anwendungsfällen eigenständig zu beherrschen und ist voraussichtlich erst ab 2030 zu erwarten [Ver15]. Die letzte Stufe, das *fahrerlose Fahren*, meint, dass von Start bis Ziel kein Fahrer erforderlich ist und das System somit die Fahraufgabe unter allen Umständen erfüllt [Ver15].

Grundvoraussetzung dafür ist, dass das Fahrzeug die aktuelle Umgebung richtig und genau erfasst, wofür Kamera-, Radar-, Sonar-, GPS- und Lidar-daten verwendet werden [Krz16]. Intel rechnet mit einer Datenmenge von 4000 Gigabyte, die zukünftig täglich durch autonome Fahrzeuge generiert werden. Allein die Kameras werden 20 bis 40 Megabit pro Sekunde erzeugen, Radar 10 bis 100 Kilobit pro Sekunde [Krz16].

Durch die Vernetzung stehen dem Automobilhersteller die Daten der Fahrzeugflotte zur Verfügung, was zum nächsten Begriff führt, der Schwarmintelligenz. Diese Bezeichnung lässt sich auf die Summierungsthese zurückführen, die besagt, dass die Entscheidung einer größeren Gruppe besser sein kann als die weniger Einzelner [Ari06]. Ein Beispiel ist die Ableitung eines Schwarmalgorithmus anhand des Verhaltens von Ameisenkolonien: Es geht

Tabelle 1.1: Grade der Automatisierung und ihre Definition nach [Ver15]

Nomenklatur	Fahraufgaben des Fahrers nach Automatisierungsgrad
Fahrerlos	Von Start bis Ziel ist kein Fahrer erforderlich. Das System übernimmt die Fahraufgabe vollumfänglich.
Vollautomatisiert	Das System übernimmt Quer- und Längsführung vollständig in einem definierten Anwendungsfall. Der Fahrer muss das System dabei nicht überwachen und wird vor dem Verlassen des Anwendungsfalles mit ausreichender Zeitreserve zur Übernahme der Fahraufgabe aufgefordert.
Hochautomatisiert	Das System übernimmt Quer- und Längsführung für einen gewissen Zeitraum in spezifischen Situationen. Der Fahrer muss das System dabei nicht überwachen und wird bei Bedarf zur Übernahme der Fahraufgabe mit ausreichender Zeitreserve aufgefordert.
Teilautomatisiert	Das System übernimmt Quer- und Längsführung (für einen gewissen Zeitraum oder/und in spezifischen Situationen). Der Fahrer muss das System dauerhaft überwachen und jederzeit zur vollständigen Übernahme der Fahraufgabe bereit sein.
Assistiert	Fahrer führt dauerhaft entweder die Quer- oder die Längsführung aus. Die jeweils andere Fahraufgabe wird in gewissen Grenzen vom System ausgeführt.
<i>Driver only</i>	Fahrer führt dauerhaft (während der gesamten Fahrt) die Längsführung (Beschleunigen/Verzögern) und die Querführung (lenken) aus.

um die Suche des kürzesten Weges bei der Futtersuche. Nachdem die ersten Ameisen die verschiedenen Wege gegangen sind, können sich die Darauffolgenden an der Pheromonspur der Vorgänger orientieren. Die Spur auf kürzeren Wegen ist stärker, da dieser Weg in gleicher Zeit von mehr Ameisen gegangen werden kann als ein Längerer. Es sind schneller mehr Pheromone auf dem Pfad [Bog13]. Daraus resultiert die *Ant-Colony-Optimization*, eine Optimierungsmethode, die z.B. für Routenplanungen verwendet wird [Bog13, DS04]. Übertragen auf die Automobilbranche bedeutet das: Viele Fahrzeuge wissen mehr als nur ein Fahrzeug. Allerdings führt erst eine intelligente Verwendung der Daten zu wirklicher Schwarmintelligenz. Mithilfe komplexer statistischer Methoden zur Datenanalyse und Mustererkennung können die Daten automatisiert ausgewertet werden, was neue Werte schafft

[Gad17]. Neben der ausschließlichen Analyse der Daten, können sie darüber hinaus zur Erstellung von selbstlernenden Systemen verwendet werden. Das erlernte System ist in der Lage, zukünftige Entscheidungen abhängig davon, worauf es trainiert wird, eigenständig zu treffen. Beispielsweise basiert autonomes Fahren auf Fahrerassistenzsystemen, die mithilfe künstlicher Intelligenz realisiert werden können [FL17].

Den Fahrzeugdaten kann bis 2030 ein Wert zwischen 450 und 750 US Dollar zugemessen werden [McK16]. Bis zu 76% der Kunden sind bereit, diese an den Fahrzeughersteller weiterzugeben [McK16]. Insbesondere für Services, die Zeit ersparen, wie die Hilfe bei der Parkplatzsuche, ist die Bereitschaft groß, Daten zu liefern und für die Leistung zu bezahlen [McK16]. [SWA14] gibt einen Überblick über die Möglichkeiten, Big Data für Innovationen entlang der automobilen Wertschöpfungskette zu nutzen:

Produktenwicklung: Das Produkt wird sowohl technisch als auch angebotseitig optimiert. Bauteile werden während ihres gesamten Lebenszyklus analysiert und dadurch verbessert. Außerdem werden fehlerhafte Teile frühzeitig erkannt und entsprechende Maßnahmen eingeleitet. Durch die Analyse des tatsächlichen Nutzungsverhaltens wird das Fahrzeug auf Kundenbedürfnisse angepasst, wodurch ein Wettbewerbsvorteil erlangt wird. Ein Beispiel hierfür ist die Analyse der Tastennutzung zur Verbesserung der HMI² Interaktionen.

Einkauf/Produktion: Durch die Analyse des Kundenverhaltens werden gefragte Modelle auf Lager produziert und selten genutzte Applikationen wegoptimiert [SWA14].

Marketing/Vertrieb: Die Erstellung von Kundenprofilen ermöglicht kundenspezifisches Marketing. So werden individuelle Angebote gemacht und Kundentrends an Forschung und Entwicklung weitergereicht. Ein konkre-

² *Human Machine Interface*, Schnittstelle zwischen Mensch und Maschine

tes Beispiel ist eine Information über Servicetermine oder Wartungsbedarf, angepasst an die individuellen Bedürfnisse des Kunden [SWA14].

After Sales: Die Datenanalyse ermöglicht es, Aussagen über Nutzung und Schadenhäufigkeit einzelner Bauteile im laufenden Betrieb zu treffen. Mängel im Fahrzeug werden erkannt und der Kunde wird darauf hingewiesen. Eine Wartung vor eigentlichem Ausfall ist möglich. Präventive Diagnose dient etwa zur Vorhersage der Batteriealterung auf Basis von realem Nutzungsverhalten. Ein anderes Beispiel ist die Kontaktaufnahme mit dem Fahrer, wenn er in Schneeregionen fährt, aber noch nicht mit Winterreifen ausgestattet ist [SWA14].

Pilotiertes Fahren: Das aus Daten der Fahrzeuge generierte Wissen dient der Weiterentwicklung des autonomen Fahrens. Beispielsweise wird durch die Fusionierung der Daten der Straßenzustand über das Fahrwerk erkannt und in eine hochauflösende Karte eingetragen.

Mobilität: Die Unterstützung bei der Parkplatzsuche oder die Navigation unter Berücksichtigung des Verkehrs-/ Straßenprofils sind Beispiele für realisierbare Digitale Services.

Die Verwendung und der Austausch der Daten bringen allerdings ebenso Risiken mit sich. Insbesondere die Datensicherheit gilt als das Gegenargument der Automatisierung [MBB17]. Fahrzeugfunktionen und elektronische Komponenten müssen gegen Cyberattacken und Manipulation geschützt sein. Sicherheitslücken ermöglichen es Hackern, in das Fahrzeug einzugreifen, was zu lebensbedrohlichen Szenarien führen kann. Ein Beispiel ist der Angriff auf einen Jeep im Jahr 2015, wodurch die 1000 km weit entfernten Computerspezialisten volle Kontrolle über das Fahrzeug erlangt haben [Har15].

Gleichermaßen spielt der Schutz der persönlichen Daten einzelner Personen eine Rolle [Ver17]. Kritisch ist zum Beispiel die Datenweitergabe vom Hersteller an Versicherungen, die dann Beiträge an das individuelle Fahrverhal-

ten anpassen [ADA17]. 70% der Deutschen beunruhigt dieses Thema, was auch zu Ängsten wie Identitätsdiebstahl führt [MBB17].

Eine gestörte Datenübertragung ist ein weiteres Risiko im Hinblick auf die Zuverlässigkeit der zunehmenden Vernetzung. Eine Echtzeitverarbeitung ist nur bei einer verzögerungsfreien Übertragung möglich [MBB17]. Zwar schreitet der Netzausbau immer weiter voran, doch gerade in ländlichen Gebieten gibt es noch Lücken [NM17].

Weitere Risiken, die dazu führen, dass trotz der Datenmenge kein Wert aus ihnen geschöpft werden kann, sind Fachkräftemangel und Interdisziplinarität. Laut Wolfgang Wahlster, dem Leiter des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI) fehlen *in Deutschland (...) unmittelbar 5 000 Leute mit KI-Expertise* [MBB17, Han17]. Auch besteht die Gefahr, dass Big Data am Zusammenspiel der unterschiedlichen Bereiche Technik, Betriebswirtschaft, Recht und Politik scheitert [MBB17].

1.1 Problemstellung und Zielsetzung

Zur beschriebenen Nutzung der Fahrzeugdaten für Produktentwicklung, Einkauf, Marketing, After Sales, pilotiertes Fahren und Mobilität müssen sie gespeichert und den Analysten zugänglich gemacht werden. Die Daten werden verschlüsselt und anonymisiert über das Mobilfunknetz an ein IT-Rechenzentrum (*Backend*) übertragen. Deren Speicherung, Verarbeitung und Analyse findet auf einer speziellen Backend-Software statt [AUD16a]. Datenquellen unterschiedlicher Kategorien stehen zur Verfügung. Die fahrzeuginternen Kommunikationsdaten, die zwischen den Steuergeräten über die Bussysteme ausgetauscht werden, liefern Informationen zu Fahrer und Fahrzeug. Des Weiteren stellen Kamera, Lidar, Radar, GPS und externe Provider Angaben bezüglich Lokalisierung (z.B. Straßenzustand, GPS-Koordinaten, Karteninformationen), Verkehr (z.B. Echtzeit-Verkehrsinformationen, Fußgänger) und Umgebung (z.B. Wetter- und Sicht-

verhältnisse) bereit. Die Dissertation konzentriert sich auf die erstgenannte Kategorie, die fahrzeugintern ausgetauschten Kommunikationsdaten. Zur Datensammlung sind zwei Vorgehensweisen denkbar. Zum einen können sämtliche Daten von den Fahrzeugen an das Backend versendet werden, um dort zur Analyse bereitzustehen. Zum anderen können die Daten im Fahrzeug vorselektiert werden, um die negativen Effekte des großen Datenvolumens und die damit verbundenen Herausforderungen der Datenübertragung zu meistern. In beiden Fällen ermöglicht eine zusätzliche Datenkompression weitere Verdichtung. Die Dissertation beschäftigt sich mit der Fragestellung, wie die Analysten mithilfe automatisierter Vorverarbeitung bei der Datenauswertung unterstützt werden können. Zudem wird betrachtet, wie sich die große Datenmenge auf die Übertragung von Fahrzeug zu Backend auswirkt. Bevor auf diese beiden Aspekte näher eingegangen wird, soll zunächst eine Beispielrechnung die Menge an Daten veranschaulichen. Ein typischer im Fahrzeug verbauter Datenlogger ermöglicht eine Übertragung von 12,6 KByte/s³. Diese technische Einschränkung reduziert die Datenmenge von 2,1 MByte, die beispielsweise ein Fahrzeug der Oberklasse pro Sekunde im Jahr 2017 maximal generiert, auf weniger als 0,6%. Doch selbst mit der Reduzierung werden noch 1,8 Terabyte Daten von 140 Fahrzeugen in neun Monaten generiert, wenn sie durchschnittlich eine Stunde pro Tag unterwegs sind. Im Serieneinsatz mit angenommen einer Million Fahrzeugen decken somit 16,8 Petabyte gerade einmal 0,6 % aller generierten Kommunikationsdaten ab (vgl. Bild 1.3).

1.1.1 Erkennen wertvoller Daten

Es stellt sich die Frage, wie aus der großen Menge an Daten der *Value* generiert werden kann. Eine Möglichkeit, Wert zu schöpfen ist eine anwendungsorientierte Verwendung der Daten. Funktionsentwickler oder Analysten nutzen die Daten, die für den speziellen Anwendungsfall relevant sind und

³ aktuell (2017) bei der AUDI AG verwendete Datenlogger

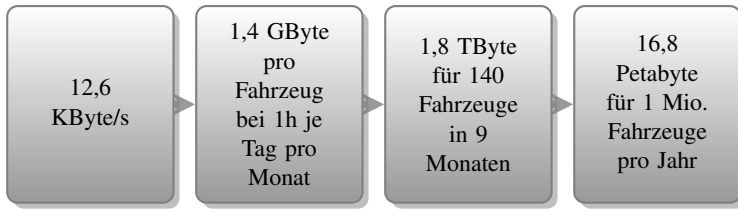


Bild 1.3: Beispielrechnung zur Veranschaulichung der Datenmenge auf dem Backend

erzeugen daraus Nutzen. Zum Beispiel sind für die Analyse des Langzeitverhaltens der Batterie Signale wie Ladezustand und Energiegehalt wichtig. Soll ein Kundenprofil erstellt werden, benötigt man hierfür die entsprechenden Informationen (z.B. Pendler oder Gelegenheitsfahrer, Langstrecken- oder Kurzstreckenfahrer, junger Fahrer oder Senior [SWA14]). Nachteil dieser anwendungsorientierten Analyse ist die starke Bindung an den konkreten Fall. Es muss vorher definiert werden, wonach gesucht wird. Eine Auswahl der Daten, die für den Sachverhalt von Interesse sind, ist notwendig, wodurch nicht das volle Potential der Daten ausgeschöpft wird.

Ein allgemeinerer Ansatz ist das Erkennen einer Anomalie, worunter man *eine Beobachtung, die so sehr von anderen Beobachtungen abweicht, dass sie den Verdacht hervorruft, durch einen anderen Mechanismus erzeugt worden zu sein*⁴, versteht [Haw80]. Eine ungewöhnliche Situation wird entdeckt, wodurch neues Wissen entsteht. Bei einer Anomalie kann es sich um unterschiedlichste Ereignisse handeln. Sie werden nicht im Preprocessing definiert, sondern im Postprocessing interpretiert:

- Es werden Schadensfälle, bzw. ungewöhnliches Verhalten, was zu Schadensfällen führt, als Anomalien erkannt (z.B. Getriebeschäden).
- Ein verändertes Kundenverhalten ist eine potentielle Anomalie.

⁴ (eigene Übersetzung) *An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.*, D. Hawkins [Haw80]

- Unerlaubte Systemeingriffe von außen, die möglicherweise den Straßenverkehr und infolgedessen Menschenleben gefährden, werden gefunden. In diesem Zusammenhang spricht man von *Intrusion* Detektion.
- Ortsabhängige Anomalien (z.B. Schlaglöcher) werden in eine Karte eingezeichnet.
- Bisher unvorhergesehene Fälle werden als Anomalie erkannt und führen zu entsprechenden Reaktionen.

Das Konzept ist unabhängig von konkreten Anwendungsfällen und demnach auch zukunftsfähig. Der Unterschied zwischen anwendungsorientierter Analyse und der Erkennung von Anomalien wird anhand von Bild 1.4 veranschaulicht. Ein Beispiel ist die Vorhersage von Getriebeausfällen: Schäden im Getriebe sollen erkannt werden, bevor es zum Ausfall kommt. Einerseits kann in Abstimmung mit den Entwicklern im Preprocessing genau spezifiziert werden, wie das normale Getriebeverhalten aussieht und in welchen Situationen ein Ausfall zu erwarten ist. Ein intelligentes System, das in der Lage ist, schadhafte Getriebe automatisiert vom Normalzustand zu unterscheiden, wird angelemt. Der allgemeinere Ansatz ist das Erkennen einer ungewöhnlichen Situation und die nachgelagerte Interpretation, in der die Erkenntnis, dass ein Problem im Getriebe vorliegt, erlangt wird. Vor-

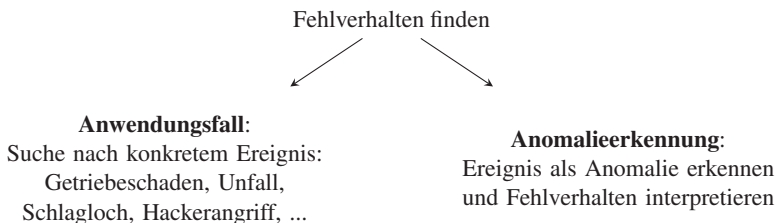


Bild 1.4: Unterscheidung zwischen Anwendungsfall und Anomalieerkennung

teil hierbei ist, dass die gegebene Datenmenge unvoreingenommen untersucht wird und Schäden erkannt werden können, die vorher nicht definiert und spezifiziert wurden. Das Beispiel lässt sich auf generelles Fehlverhalten verallgemeinern.

Der Automobilhersteller wird dabei unterstützt, neue Anwendungsfälle abzuleiten. Somit dient die Erkennung von Anomalien als Vorverarbeitungsschritt zur Filterung ereignisreicher Daten, wobei automatisiert definiert wird, was *ereignisreich* ist.

1.1.2 Selektion wertvoller Daten

Eine weitere Herausforderung stellt die Datenübertragung vom Fahrzeug an das Backend dar. Im Folgenden werden die damit verbundenen Probleme vorgestellt: Es gilt das **Prinzip der Sparsamkeit** und die entstehenden **Kosten** müssen reduziert werden. Außerdem zeigt die Limitierung der **Datenübertragung** über die Luftschnittstelle die physikalischen Grenzen auf. Das Sammeln aller generierten Daten ist nicht zulässig, da die gesetzlichen Anforderungen der **Datenschutzgrundverordnung** eingehalten werden müssen. Zuletzt wird der Ansatz des **Edge Computings** zusammengefasst und auf die Problematik des Datentransfers übertragen. Es wird dargelegt, dass die Selektion wertvoller Daten im Fahrzeug die Herausforderungen meistert.

Prinzip der Sparsamkeit: Ist eine Grundmenge an Daten auf dem Backend gegeben, kann daraus das benötigte Wissen abgeleitet werden. In diesem Fall liefern weitere Datensätze keine neuen Inhalte. Gemäß dem Sparsamkeitsprinzip nach William von Ockham sollte von mehreren Erklärungen eines Sachverhalts die Einfachste verwendet werden [Mit16]. Dieses Prinzip lässt sich auf den Datentransfer zwischen Fahrzeug und Backend übertragen. Daten, die keine neuen Informationen liefern, verkomplizieren die Datenbasis auf dem Backend unnötig.

Kosten: Der Automobilhersteller verfolgt grundsätzlich das Ziel der Gewinnerhöhung, welches durch eine Steigerung der Erlöse und durch eine Reduktion der Kosten erfolgt. Im Zuge der Elektrifizierung und Digitalisierung ist künftig ein noch stärkerer Kostendruck zu erwarten. Dies ist beispielsweise auf hohe Kosten der Batterie, des Ausbaus der Ladeinfrastruktur, auf die Entwicklung der Onlinedienste und auf den Betrieb dieser zurückzuführen. Daher sind Automobilhersteller stets bemüht, Einsparungspotentiale zu nutzen, um die nötige Innovationsleistung heute und zukünftig aufbringen zu können und einen rentablen Fortbestand des Unternehmens und seiner Beschäftigten zu sichern. Unter Beachtung der Übertragungskosten, der Betriebskosten und der Hardwarekosten im Fahrzeug ist es sinnvoll, Einsparungspotentiale zu identifizieren. Geht man von Übertragungskosten von etwa 6,90 Euro/GByte⁵ aus, führt das für ein Fahrzeug mit durchschnittlicher Fahrzeit von zwei Stunden am Tag zu monatlich 19 Euro. Hinzu kommen Speicherkosten von etwa 0,002 US Dollar pro GB⁶. Im Serieneinsatz mit einer Million Fahrzeuge, entstehen somit Aufwendungen von mehr als 228 Millionen Euro im Jahr. Eine Möglichkeit, diese laufenden Ausgaben zu senken, ist daher die Selektion relevanter Daten im Fahrzeug zur Reduktion von Datenvolumen. Die dafür entstehenden einmaligen Hardwarekosten belaufen sich selbst im ungünstigsten Fall, dem Einbau eines neuen Steuergeräts zur Realisierung des Algorithmus, auf weniger als 1% der jährlichen Übertragungskosten. Ferner entspannt sich die Lage mit dem im Jahr 2020 zumindest in Premiumfahrzeugen erwarteten Zentralrechner [Vol18]. Weitere Sensoren wie Kamera und Lidar verstärken den Effekt der Datenmenge noch drastischer (4000 Gigabyte täglich je Fahrzeug [Krz16]).

Datenübertragung: Die limitierte Bandbreite und der Datenverlust über die Luftschnittstelle (z.B. Funklöcher) spielen eine weitere Rolle. Mit der *Long Term Evolution (LTE)* als Mobilfunkstandard der vierten Generation

⁵ Vodafone, Stand 2018

⁶ Preis basiert auf Preisliste von Amazon Web Services, Januar 2018 <https://aws.amazon.com/de/s3/pricing/>

(4G) ist eine Datenübertragungsrate von bis zu 300 Mbit/s möglich. Der Nachfolger LTE-Advanced erreicht Raten von bis zu 1 Gbit/s [HTLN17]. Die Standards zur Übertragung decken daher die Datenmenge der im Fahrzeug generierten Kommunikationsdaten ab. Doch insbesondere in ländlichen Gebieten zeigt der Netzausbau noch Lücken [NM17]. Zudem müssen die zu übertragenden Daten im Fahrzeug vorverarbeitet werden, was Zeit bzw. Rechenpower voraussetzt.

Datenschutzgrundverordnung: Dazukommend ist der zentrale Ansatz, alle Daten an das Backend zu senden und dort auszuwerten, nicht mit der EU-Datenschutzgrundverordnung⁷ vereinbar. Die Grundverordnung besagt, dass *die Verarbeitung personenbezogener Daten (...) für festgelegte, eindeutige und rechtmäßige Zwecke erhoben und nicht in einer mit diesen Zwecken nicht zu vereinbarenden Weise verarbeitet werden*, was bedeutet, es ist nur erlaubt, die Daten zu sammeln und zu speichern, die für konkrete Anwendungsfälle bzw. Dienste verwendet werden.

Es muss daher geprüft werden, welche Daten zu übermitteln sind. Die Idee der Datenkompression schafft zwar die Reduzierung des Datentransfers, ist jedoch nicht konform mit der Datenschutzgrundverordnung. Zudem ermöglicht eine Kombination aus Datenselektion und -kompression eine stärkere Reduzierung als die alleinige Kompression aller Daten.

Edge Computing: Die Problematik der Datenübertragung von Fahrzeug zu Backend deckt sich mit dem Ansatz des *Edge Computings*. [Her17] zeigt auf, dass die Datenmenge, die vernetzte Geräte produzieren, die Leitungen in die Cloud verstopfen und die Telekomanbieter einen hohen Preis für den *Upload* verlangen. Daher ist die Idee des *Edge Computings*, die Daten nicht komplett in das Rechenzentrum zu laden, sondern direkt im Gerät selbst, einer Komponente am Rand der Infrastruktur, zu verarbeiten. Dies ermög-

⁷ Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG (Datenschutz-Grundverordnung)

licht neben der Kosteneinsparung auch die Reduzierung der Schwierigkeiten von Datenzentren, wie Energieverbrauch, Lastspitzen, Bandbreitennutzung und Datensicherheit und -schutz [Her17, SCZ⁺16]. Laut Kay Wintrich, dem Technical Director von Cisco Deutschland, (...), *ist das die einzige Möglichkeit, mit der großen Menge an anfallenden Daten umgehen zu können* [Her17], denn die enorme Datenmenge der Rohwerte, wird durch *Cloud Computing* nicht effizient verarbeitet werden können [SCZ⁺16]. Insbesondere wenn es um Echtzeitanwendungen geht, ist die konventionelle Vorgehensweise limitiert [VWB⁺16], weswegen eine Vorverarbeitung der Daten im Fahrzeug unumgänglich ist.

1.2 Wissenschaftlicher Beitrag

Durch die Selektion von Situationen, die für das Backend selten sind, kann die Datenmenge auf das Wesentliche, sprich auf anormale Informationen, reduziert werden und das Backend wird nicht mit unnötigen Angaben, wie beispielsweise zum wiederholten Mal die Sitzeinstellung desselben Fahrers, überflutet. Die Vorauswahl relevanter Daten im Fahrzeug erleichtert die Analyse auf dem Backend und die Probleme der Datenübertragung werden gemindert. Außerdem handelt es sich bei Anomalien um potentiell ungewöhnliche und für den Automobilhersteller interessante Ereignisse, wodurch Wert geschöpft wird. Der Fokus der Dissertation liegt darin, ungewöhnliches Fahr- und Fahrzeugverhalten zu identifizieren. Das automatisierte Erkennen unerwarteter Signalwerte soll den Experten dabei helfen, neue Potentiale zu entdecken.

Aktuell findet Anomalieerkennung in Fahrzeugdaten vorwiegend im Bereich Security statt, mit der Absicht, Angriffe zu identifizieren. Dazu wird fahrzeugintern analysiert, wann außergewöhnliche Vorkommnisse auftauchen [WKSZ18]. Bei den Anomalien handelt es sich um synthetische Datensätze, welche die in der Literatur gelisteten Attacken imitieren [TLJ16].

Die verwendeten Daten sind hierbei meist die rohen Bus-Paketdatenwörter und es betrifft den Datentransfer bzw. die Sequenz der Nachrichten [TLJ16, MS17, WZG⁺18]. Beispiele hierfür sind das Vertauschen von Message-Ids, um einen gefälschten Sender zu simulieren, oder das Einfügen harmloser Zufallspakete in den Live-Bus. Anomalien entstehen nur im Kontext mit benachbarten Paketen durch das Hinzufügen oder Nichterscheinen erwarteter Pakete [TLJ16]. Die Dissertation verwendet auch die über den Bus ausgetauschten Signale, richtet die Aufmerksamkeit allerdings weniger auf den Datentransfer der Pakete zwischen den Steuergeräten, sondern auf den semantischen Zusammenhang der Signale.

Diese Art der Anomalieerkennung zur Identifikation anormalen Fahr- und Fahrzeugverhaltens wurde bereits in [TD13, The14, NMJ16, NMJ15, ZCW⁺17] evaluiert. Sie kommen dabei zu Ergebnissen von bis zu 92,9% richtig erkannter Anomalien, bei sieben fälschlicherweise erkannten Fällen in einer Stunde, wenn sich die Analyse auf den Ruhezustand beschränkt. Für fahrende Fahrzeuge wird eine Vorhersagegenauigkeit von 100% (ausschließlich Autobahn) bzw. 93% erreicht [ZCW⁺17, The14]. Die Weiterentwicklung der Dissertation liegt in den folgenden Aspekten:

Methoden zur Anomalieerkennung: Die Unterschiede in der Verwendung der Methoden zur Anomalieerkennung sind

1. die Anzahl betrachteter Signale. Während in der Literatur nur ein bis fünf unterschiedliche Signale analysiert werden [TD13, The14, NMJ16, NMJ15, ZCW⁺17] (darunter Geschwindigkeit, Motordrehzahl, offene Tür, Fahrzeughaltung), hat die Arbeit das Ziel, zu prüfen, wie gut die Verfahren zur Anomalieerkennung mit höherdimensionalen Featureräumen von bis zu 100 Signalen umgehen können.
2. die Anwendung neuerer Verfahren. In der Literatur werden neu etablierte Verfahren (wie z.B. Isolation Forest) noch nicht zur Anomalieerkennung in Fahrzeugdaten verwendet.

3. die Verwendung realer und großer Datenmengen. Neben synthetischen Daten werden reale Fahrzeugdaten und Anomalien betrachtet; über 100 Stunden Fahrzeit stehen zur Verfügung.

Konzept der Anomalieerkennung: Weiterer Beitrag liegt in dem Konzept, Training und Anwendung räumlich zu trennen. Basierend auf den Flot- tendaten im Backend wird gelernt, was normal ist. Die Entscheidung im Fahrzeug erfolgt daher über ein anhand der Flotte trainiertes Referenzmo- dell, das stets aktualisiert wird und in den Fahrzeugen mitläuft. Ziel ist es, kontinuierlich dafür zu sorgen, dass immer die Ereignisse an das Backend gesendet werden, die noch fehlen, womit ein sich selbst regulierendes Sys- tem zur Datensammlung entsteht. Die Aufteilung folgt der Idee des *Edge Computings* und ermöglicht die Reduzierung des Datentransfers ohne ge- gen Datenschutzrichtlinien zu verstoßen.

Forschungsfragen: Ziel der Dissertation ist das Erkennen außergewöhn- licher Ereignisse zur Datenselektion im Fahrzeug. Die Bearbeitung der bei- den erstgenannten Punkte ermöglicht die Antwort auf die folgenden For- schungsfragen:

1. Mit welcher Methode können außergewöhnliche Fahrsituationen (An- omalien) in den fahrzeuginternen Kommunikationsdaten erkannt wer- den?
2. In welchem Maß kann die von Fahrzeug zu Backend gesendete Daten- menge durch die Erkennung von Anomalien reduziert werden?

Es werden die Forschungsfragen bearbeitet und in Summe kann festgehalten werden, dass

1. für die Fallbeispiele das Replikator Neuronale Netz die anderen Me- thoden zur Anomalieerkennung übertrifft;
2. die Anomalieerkennung im Fahrzeug nicht als Alarmsystem, sondern zur Reduzierung des Datentransfers geeignet ist. Die Aufteilung in

Trainings- und Kontrolldaten ergibt, dass das Ziel der Datenreduktion um das Hundertfache zur Einschränkung auf 0,715% der Kontrolldaten führt. Somit werden die 2,1 MByte, die ein Oberklassefahrzeug pro Sekunde generiert, auf 15 KByte/s gemindert;

3. eine durchschnittliche *Area Under Curve* von 0,8 bedeutet, dass das System der Anomalieerkennung Schwächen zeigt; nicht alle Anomalien lassen sich eindeutig von normalen Situationen trennen;
4. die bedeutsamsten Ereignisse, wie Unfälle oder ABS-Eingriffe, klar unterscheidbar sind und eine Verfeinerung der Kontexte die Ergebnisse verbessert;
5. zur optimierten Unterscheidung im Backend noch weiteres Postprocessing, bestehend aus Interpretation und Einordnung, erfolgen muss, da die Experten nicht alle erkannten Anomalien manuell bewerten können.

1.3 Struktur der Arbeit

Mit den Grundlagen der Fahrzeugdaten und des maschinellen Lernens wird in Kapitel 2 zunächst ein einheitliches Verständnis für die in den nachfolgenden Kapiteln verwendeten Begriffe vermittelt. Kapitel 3 beschreibt das Konzept und den Ablauf der Anomalieerkennung zur Reduzierung des Datentransfers. Zudem werden die Anforderungen an die gesuchte Methode vorgestellt. Kapitel 4 vermittelt den aktuellen Stand der Technik. Dazu werden Methoden zur Anomalieerkennung erläutert und im Hinblick auf die Konzeptumsetzung bewertet. Die Vorstellung ihrer Anwendungsfelder rundet die Einordnung ab. Außerdem werden die Schritte des Postprocessings diskutiert. Dieses beinhaltet die Entscheidung, ob es sich um eine Anomalie handelt, sowie die nachgelagerte Interpretation. Experimente und Ergebnisse befinden sich in Kapitel 5. Es wird beschrieben, welche Daten für die

Versuche zur Verfügung stehen und wie die Ergebnisse evaluiert werden. Es findet ein Methodenvergleich statt und anhand dem Verfahren mit bester Performance wird verdeutlicht, welche Anomalien erkannt werden können und was dies für den Datentransfer zwischen Fahrzeug und Backend bedeutet. Ein weiteres Beispiel ist in Kapitel 6 die Analyse von Getriebeschäden zur Vorhersage von Ausfällen. Es handelt sich hierbei um einen konkreten Anwendungsfall. Die Arbeit endet mit einer zusammenfassenden Diskussion der Ergebnisse und einem Ausblick in Kapitel 7.

2 Grundlagen und Definitionen

In einem Fahrzeug werden Daten generiert, die dessen Zustand und Lokalisierung beschreiben. Zudem kann das Verhalten des Fahrers durch Daten abgebildet werden. Eine Studie zeigt, dass bis zu 76% der Kunden bereit sind, diese Daten an die OEMs weiterzugeben, um die nächste Fahrzeuggeneration zu verbessern [McK16]. Insbesondere für Services, die Zeitersparnis gewährleisten (z.B. Hilfe bei der Parkplatzsuche), ist die Bereitschaft groß, Daten zu liefern und für die Leistung zu bezahlen. Dadurch lassen sich bis zum Jahr 2030 bis zu 750 Milliarden Dollar jährlich umsetzen [McK16]. Die Fahrzeuge können als mobile Datenquellen betrachtet werden, die untereinander und mit stationären Rechenzentren kommunizieren können [Köl16, Abt16]. Kapitel 2.1 gibt einen Überblick über die generierten Daten und deren Speicherung.

Erst durch die automatisierte Auswertung der Daten mit Hilfe statistischer Methoden oder selbst lernender Systeme kann Wert aus ihnen geschöpft werden [Gad17]. Kapitel 2.2 stellt die Grundlagen dieser Methoden mit Bezug auf die Erkennung außergewöhnlicher Ereignisse vor.

2.1 Datengrundlage

Derzeit sind in einem oberen Mittelklassenfahrzeug um die 131 Steuergeräte verbaut. Sie verarbeiten Sensorgrößen wie Motordrehzahl, Temperatur oder Druck zur Steuerung, Regelung und Überwachung verschiedener Funktionen [Rei15]. Beispiel sind die Motor- oder Getriebesteuerung. Die Steuergeräte sind durch Bussysteme miteinander verbunden. Hierbei sind die derzei-

tigen Standards CAN- (Controller Area Network), LIN- (Local Interconnect Network), FlexRay- und Most- (Media Oriented Systems Transport) Busse. Der CAN-Bus ist das dominierende Bussystem mit einer Datenübertragungsrate von bis zu 1Mbit/s [can15]. Noch mehr Daten kann der verbesserte CAN-FD übertragen, dessen Rate zwischen der des CAN-Busses (1Mbit/s) und der des FlexRays mit 10Mbit/s liegt [Dec13]. Übertragen wird ein sogenanntes Datenwort (Botschaft), das aus vier Teilen zusammengesetzt ist (vgl. Bild 2.1): Nach einer festen Kombination von Nullen und Einsen folgt die Adresse des Empfängers bzw. Absenders der Nachricht (abhängig von der Art des Bussystems). Der dritte Teil ist die Beschreibung, um welches Signal es sich handelt und welchen Wert dieses aktuell annimmt. Das Datenwort endet mit einer Kennzeichnung für das Ende der Nachricht [Win17].

Signal: Im Rahmen der Arbeit ist ein Signal die Beschreibung der Sensorgröße, welche zwischen den Steuergeräten transportiert wird (z.B. Lenkwinkel, Motordrehzahl).

Signalwert: Der Signalwert ist der zugehörige Wert, den das Signal zum aktuellen Zeitpunkt annimmt.

Kommunikationsdaten: Die Verknüpfungen aus Signal und Signalwert werden als Kommunikationsdaten bezeichnet.

Der LIN-Bus wird hauptsächlich für einen einfachen und kostengünstigen Austausch von unkritischen Daten wie Komfortinformationen verwendet. FlexRay und Most ersetzen den CAN wenn dieser an seine Grenzen bezüglich



Bild 2.1: Einfaches Datenwort nach [Win17]

lich Bandbreite und Echtzeitanforderungen stößt, wie es beispielsweise bei Multimediadaten im Fahrzeug der Fall ist [Win17, Sch12].

Bild 2.2 zeigt die Komplexität der internen Vernetzung und Architektur. Betrachtet man die dadurch entstehende Datenmenge, so fließen in einem Fahrzeug der oberen Mittelklasse zwischen 131 Steuergeräten etwa 2,1 Mbyte/s Kommunikationsdaten entlang der 48 Fahrzeugbusse (vgl. Bild 1.2).

Das zentrale Gateway (cGW) dient als Bindeglied zwischen Fahrzeug und Außenwelt. Das Steuergerät ist durch die Bussysteme mit den anderen Steuergeräten verbunden und bietet daher die Möglichkeit, als Datenlogger fahrzeugbezogene Kommunikationsdaten nach außen weiterzugeben [Rob18]. Durch Erweiterungen kann konfiguriert werden, welche Signale in welcher Rate übertragen werden sollen. Über eine Luftschnittstelle werden die im cGW geloggteten Daten an ein Backend übertragen, um dort durch den Fahrzeughersteller analysiert zu werden (vergleiche Bild 2.3). Wie in Kapitel 1 dargestellt, dienen die Daten zu Innovationen entlang der automobilen Wertschöpfungskette (z.B. Produktoptimierung durch den Zugriff auf das Verhalten zur Laufzeit oder verbessertes Marketing durch Kundenprofile).

Backend: Unter Backend versteht man den Datenspeicher, der die Daten von einer Vielzahl an Fahrzeugen aufnimmt und für Analysten zugänglich macht.

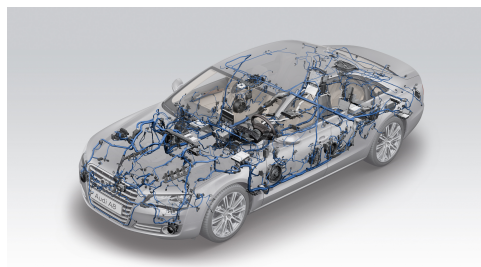


Bild 2.2: Interne Fahrzeugkommunikation

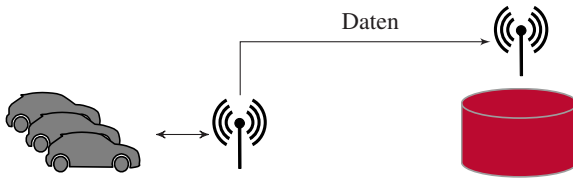


Bild 2.3: Datenübertragung von Fahrzeug zu Backend

Da von außen konfiguriert werden kann, welche Signale von den Fahrzeugen abgegriffen werden sollen, können mehrere Kampagnen gleichzeitig stattfinden, sodass verschiedene Fahrzeuge unterschiedliche Signalinformationen liefern.

Kampagne: Eine Kampagne ist eine von Experten definierte Signalliste für eine Baureihe. Es ist die Beschreibung welche Signale, in welcher Rate, vom Fahrzeug an das Backend übertragen werden.

Fahrzeugflotte: Die Fahrzeuge, die dieselbe Kampagne fahren, bilden eine Fahrzeugflotte.

Flottendaten: Die Daten, die eine Fahrzeugflotte generiert, werden als Flottendaten bezeichnet.

Schwarmdaten: Die Flottendaten mehrerer Kampagnen bilden in ihrer Gesamtheit eine Menge an Daten von unterschiedlichen Fahrzeugen. Sie werden als Schwarmdaten betrachtet.

Bild 2.4 verdeutlicht das Sammeln der Daten mittels Kampagnen. Zur Anonymisierung liegen die Daten sessionbasiert auf dem Backend.

Session: Eine Session beschreibt eine Fahrt von maximal einer Stunde; eine längere Fahrt wird durch mehrere Sessions abgebildet.

Tabelle 2.1 zeigt, wie eine Session auf dem Backend gespeichert wird. Um eine Verbindung zwischen den Signalen erkennen zu können, findet eine

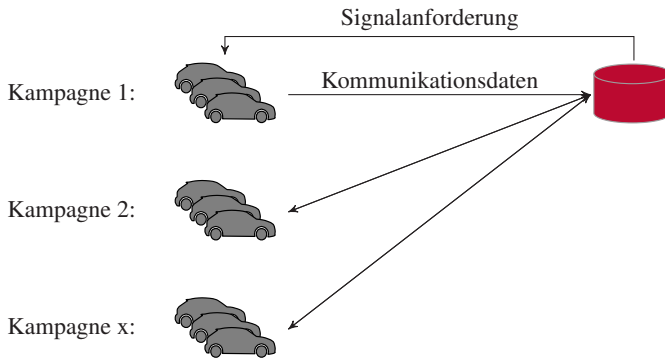


Bild 2.4: Definition von Kampagnen anhand unterschiedlicher Fahrzeugflotten

Weiterverarbeitung statt. Innerhalb einer Session werden pro Zeiteinheit die Werte aller Signale betrachtet, wodurch sich ein Signalvektor ergibt (vgl. Tabelle 2.2).

Signalvektor: Ein Signalvektor beschreibt den Zustand aller in einer Kampagne angeforderten Signale zu einem Zeitpunkt innerhalb einer Session. Wurde für den Zeitpunkt kein Wert übermittelt, wird der letzte gegebene Signalwert verwendet.

Ein Signalvektor kann aus qualitativen und quantitativen Signalen bestehen, welche anhand unterschiedlicher Skalenniveaus charakterisiert werden: [SL17, VK18]:

Tabelle 2.1: Beschreibung einer Session auf dem Backend

Session Id	Zeit	Signal	Signalwert
...

Tabelle 2.2: Beschreibung einer Session nach Weiterverarbeitung, d unterschiedliche Signale

Session Id	Zeit	Signal 1	...	Signal d
...	...	Signalwert 1	...	Signalwert d

Qualitative Signale: Qualitative Signale beschreiben Statuswerte wie zum Beispiel die Straßenkategorie (Autobahn, Landstraße, ...) oder die ausgewählte Fahrstufe bei einem Automatikgetriebe. Die Zustände der Signale sind numerisch codiert, ihre Abstände sind allerdings nicht messbar. Qualitative Signale können nominal oder ordinal sein:

Nominalskala: Nominale Signalwerte sind lediglich Merkmalausprägungen, die sich nicht vergleichen lassen (z.B. Statuswert Fahrstufe).

Ordinalskala: Ordinale Signalwerte lassen sich vergleichen und sortieren, allerdings kann nur eine natürliche Rangfolge angegeben werden, die Abstände sind nicht messbar (z.B. Kundenzufriedenheit *sehr gut, gut, ..., ungenügend*).

Quantitativen Signale: Quantitative Signale können gemessen oder gezählt werden. Die Abstände zwischen den Werten haben eine Bedeutung (z.B. Motordrehzahl oder Geschwindigkeit). Quantitative Signalwerte sind metrisch.

Metrische Skala: Metrische Signalwerte sind messbar.

2.2 Erkennung von Anomalien mithilfe maschinellen Lernens

Der Fokus der Arbeit liegt in der Erkennung von Anomalien als eine Möglichkeit mit der Menge an Daten umzugehen. In der Literatur wird eine Anomalie häufig als *eine Situation, die so sehr von anderen Beobachtungen abweicht, dass sie den Verdacht hervorruft, durch einen anderen Mechanis-*

mus erzeugt worden zu sein¹, definiert [Haw80]. [CBK09] beschreibt die Erkennung von Anomalien als das Problem unerwartete Muster zu finden. Ein anderer Begriff, der in diesem Zusammenhang auftaucht, ist die sogenannte *Novelty* Detektion. Unter *Novelty* versteht man zukünftige Datensätze, die sich von den Beobachtungen der Gegenwart unterscheiden und daher auch als Anomalien verstanden werden können [Mar03, PCCT14]. Ein *Ausreißer* wiederum ist eine Anomalie innerhalb gegenwärtiger Beobachtungen. Es unterscheidet sich von einer *Novelty* dahingehend, dass es sich bei dem Ausreißer um ein ungewöhnliches Ereignis innerhalb der gegebenen Beobachtungen handelt. Eine *Novelty* beschreibt ein ungewöhnliches Ereignis in zukünftigen Beobachtungen. Die Erkennung von Ausreißern ist ein wichtiger Preprocessing Schritt vor der Analyse der Daten, da sie unerwünschten Einfluss auf die Ergebnisse nehmen können [Mar03]. Beispiel hierfür wäre der Mittelwert der Datenmenge $X = \{2; 3; 5; 7; 3; 5; 6; 2; 7; 4; 6; 10000\}$, der durch den einen sehr großen Eintrag stark beeinflusst wird. Im Rahmen der Arbeit werden die Definitionen folgendermaßen übertragen:

Anomalie: Eine Anomalie wird durch einen bzw. eine aufeinanderfolgende Reihe von Signalvektoren beschrieben, die von den gegenwärtigen Flotendaten abweichen und daher eine *Novelty* darstellen. Hierbei wird in der Literatur häufig unterschieden zwischen [CBK09]:

Punktuelle Anomalie: Ein Datenpunkt wird als punktuelle Anomalie bezeichnet, wenn er anormal im Vergleich zu den anderen Daten ist.

Kontextuelle Anomalie: Ein Datenpunkt wird als kontextuelle Anomalie bezeichnet, wenn er nur in einem bestimmten Kontext anormal ist.

Kollektive Anomalie: Eine Sammlung von Datenpunkten wird als kollektive Anomalie bezeichnet, wenn diese in ihrer Gesamtheit anormal

¹ (eigene Übersetzung) *An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.*, D. Hawkins [Haw80]

im Vergleich zu den anderen Daten sind. Jeder Datenpunkt individuell fällt allerdings nicht auf.

Anomalie Score: Durch den Anomalie Score wird beschrieben, wie anomal ein Signalvektor ist. Je höher der Score, desto wahrscheinlicher ist es eine Anomalie.

In der Arbeit werden Anomalien mithilfe maschinellen Lernens erkannt. In diesem Zusammenhang spricht man von Features.

Maschinelles Lernen: Maschinelles Lernen ist die Untersuchung von Computeralgorithmen, die in der Lage sind, sich automatisch durch Erfahrung zu verbessern. [Ert16, Mit97].

Feature: Ein Feature ist die Beschreibung des Inhalts, welcher als Eingabe für das maschinelle Lernen verwendet wird.

Ein Signal kann als Feature verstanden werden. Jedoch ist auch eine höherwertige Zusammenfassung, wie z.B. der Mittelwert der Signalwerte, ein Feature.

Eine erforschte Methode des maschinellen Lernens ist das überwachte Lernen (engl. *supervised learning*), bei dem anhand von Eingabemerkmalen (z.B. Geschwindigkeit, Motordrehzahl und Temperatur) die Zielausgabe vorhergesagt wird. Bei der Klassifikation ist die Zielausgabe eine diskrete Klasse, wie zum Beispiel die Unterscheidung zwischen den Klassen *intaktes Getriebe* und *beschädigtes Getriebe*. Ist die Ausgabe eine kontinuierliche Variable (z.B. Lenkradwinkel), handelt es sich um eine Regression [Bis06].

Klassifikator: Ein Klassifikator sagt anhand von Features die entsprechende Klasse vorher. Zur Definition des Klassifikators sind Datensätze aller Klassen gegeben und die Abbildung von Merkmalausprägung zu Klasse wird gelernt [Alp10].

Unter anderem sind Support Vektor Maschinen und neuronale Netze Lösungsansätze für Klassifikationsprobleme.

Support Vektor Machine (SVM): Eine Support Vektor Machine definiert die Ebene, die Datensätze unterschiedlicher Klassen bestmöglich trennt [CV95].

Neuronales Netz: Mithilfe von neuronalen Netzen können Funktionen approximiert werden, die anhand der Eingabedaten die entsprechende Klasse vorhersagen können. Ein Netz besteht aus Neuronen, die durch Gewichtsvektoren miteinander verbunden sind. Die Neuronen sind angeordnet in Eingabe- und Ausgabeschicht, getrennt durch verborgene Schichten. Die Aktivierungsfunktion stellt dar, wie ein Neuron die Eingabe verarbeitet und weitergibt [RW18]. Eine genauere Beschreibung folgt in Kapitel 4.1.1.

Es wird unterschieden zwischen Training, Test, Validierung und Anwendung [Rip96]:

Training, Trainingsdaten: Mithilfe der Trainingsdaten wird der Klassifikator trainiert/gelernt.

Validierung, Validierungsdaten: Die Validierungsdaten, die nicht Teil der Trainingsdaten sind, werden verwendet, um den Klassifikator zu optimieren, indem die vom Benutzer zu definierenden Hyperparameter angepasst werden. Es wird das Modell ausgewählt, welches die besten Ergebnisse für die Validierungsdaten liefert.

Test, Testdaten: Nach Training und Validierung folgt der Test auf einem dritten Datensatz. Hier wird die Performance des Klassifikators durch den Vergleich zwischen tatsächlicher und vorhergesagter Klasse gemessen.

Anwendung, Anwendungsdaten: Erzielt der ausgewählte Klassifikator in Test und Validierung ausreichende Genauigkeit, wird er verwendet. Er wird auf neue Datensätze mit unbekannter Klasse angewandt und klassifiziert diese.

Grundwahrheit, Label: Für Training, Test und Validierung muss zur Bewertung die Grundwahrheit bzw. das sogenannte *Label* verfügbar sein, um sie mit der Vorhersage des Klassifikators vergleichen zu können.

Teil der Validierung ist die Optimierung des Referenzmodells durch die Anpassung der vom Benutzer zu definierenden Parameter (**Hyperparameter**). Zum Beispiel ist beim Einordnen der Daten in unterschiedliche Gruppen, die Anzahl an Gruppen ein Hyperparameter. Gängige Verfahren zur Optimierung der Hyperparameter sind die manuelle Suche des Optimums oder eine systematische Rastersuche innerhalb benutzerdefinierter Grenzen, speziell angepasst an den Anwendungsfall [BB12].

Durch die Aufteilung in die unterschiedlichen Phasen kann verhindert werden, dass eine Überanpassung an die Trainingsdaten übersehen wird.

Überanpassung, engl. Overfitting: Performt der Klassifikator für die Trainingsdaten sehr gut, liefert jedoch schlechte Ergebnisse in Test und Validierung, ist er überangepasst an die Trainingsdaten und dadurch für die zukünftige Anwendung unbrauchbar [Bis06].

Ein Beispiel für eine Überanpassung aus der Bildverarbeitung ist ein Klassifikator zur Erkennung von Panzern [Fra98]. Während die Trainingsdaten perfekt klassifiziert werden können, funktioniert er mit neuen Testdaten nicht mehr. Grund ist, dass sich die Trainingsphotos mit und ohne Panzer durch den Sonnenstand unterscheiden lassen. Diese Unterscheidung wurde gelernt und man spricht hier von Überanpassung, da dies nur für die Trainingsdaten richtig ist [Fra98]. Bild 2.5 veranschaulicht das Problem der Überanpassung anhand des Beispiels einer Funktionsapproximation [Bis06].

Um bei kleinen Testdatensätzen statistische Unsicherheiten zu verringern, wird die sogenannte *Kreuzvalidierung* verwendet [GBC16].

Kreuzvalidierung: Die Kreuzvalidierung evaluiert den Algorithmus mehrmals, indem der ursprünglich gegebene, gelabelte Datensatz jeweils zufällig

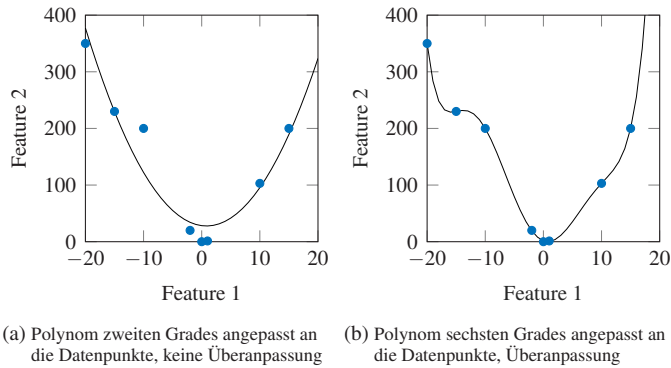


Bild 2.5: Beispiel der Überanpassung anhand von Polynomapproximation nach [Bis06]

in Trainings- und Testdaten aufgeteilt wird. Zur Bewertung dient dann die durchschnittliche Performance. Bei der k -fach Kreuzvalidierung wird die Datenmenge in k gleichgroße, disjunkte Teilmengen aufgeteilt und Training und Test k mal wiederholt. Hierbei wird jeweils eine Teilmenge als Testdatensatz verwendet, während die anderen $k - 1$ Teile die Trainingsdaten liefern [GBC16].

Eine Möglichkeit der Anomalieerkennung ist das sogenannte *one-class* Setup. Hierbei werden in der Trainingsphase nur Datenpunkte einer Klasse benutzt. Für die Erkennung von Getriebschäden bedeutet das, dass im Training keine Schäden zur Verfügung stehen. In der Anwendungsphase fallen die Ereignisse allerdings als anomal auf und werden durch weitere Interpretation erkannt. Bei einem traditionellen *two-class* Klassifikator wird bereits in der Trainingsphase gelernt, zwischen funktionsfähigen und defekten Getrieben zu unterscheiden, da Daten beider Ereignisse zum Erlernen verwendet werden.

One-class Anomaliedetektor: Der *one-class* Anomaliedetektor unterscheidet sich von einem traditionellen Klassifikator darin, dass im Training

nur Daten der Zielklasse (= normal) gegeben sind. Der *one-class* Anomalie-detektor wird als Beschreibung der Daten trainiert und davon abweichende Datenpunkte gelten als *anormal* [MT01, CBK09] (vgl. Bild 2.6). Der Grad der Abweichung wird durch den Anomalie Score beschrieben.

Referenzmodell, *one-class* Klassifikator: Der gelernte Klassifikator, der zwischen den Klassen *normal* und *anormal* unterscheidet, wird als Referenzmodell bzw. *one-class* Klassifikator bezeichnet.

Betrachtet man die Signalvektoren als voneinander unabhängig, werden sie statisch betrachtet. Wird jedoch ihr zeitlicher Verlauf als zusätzliche Information hinzugenommen, werden die Kommunikationsdaten zu Sequenzen. Da die Verläufe mehrerer Signale gleichzeitig betrachtet werden, beschreibt eine Session eine **multivariate Zeitreihe** (vgl. Tabelle 2.2). Bild 2.7 zeigt beispielhaft die Werte der Signale *angezeigter Verbrauch* und *Motordrehzahl* innerhalb einer Minute. Zusammen skizzieren sie eine multivariate Zeitreihe der Dimension $d = 2$, da pro Zeitpunkt beide Werte berücksichtigt werden.

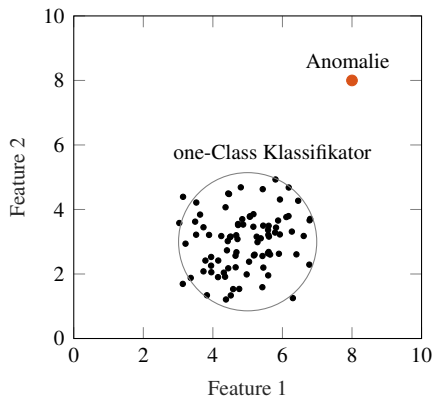


Bild 2.6: Idee der *one-class* Klassifikation

Ein in der Literatur viel zitiertes Problem ist der sogenannte *Fluch der Dimensionalität*, der auf die Problematik eines hoch dimensionalen Feature-Raums hinweist.

Fluch der Dimensionalität, engl. *curse of dimensionality*: Die Verwendung einer $L_p, p \geq 1$ Norm als Vergleichskriterium führt dazu, dass der Abstand zwischen zwei Datenpunkten für eine steigende Dimension gegen den Erwartungswert des Abstandes zwischen allen gegebenen Datenpunkten konvergiert. Dadurch verwischen die Unterschiede zwischen den Abständen unterschiedlicher Datenpunkte, was eine Vergleichbarkeit erschwert [Bis06, ZSK12].

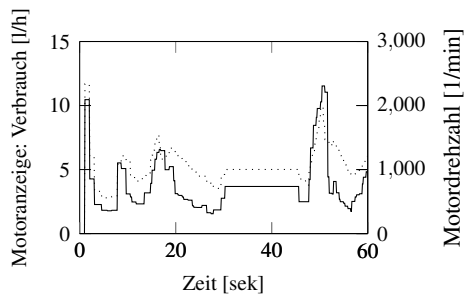


Bild 2.7: Geschwindigkeit (durchgezogen) und Motordrehzahl (gepunktet) als multivariate Zeitreihe

3 Konzept der Anomalieerkennung zur dynamischen Sammlung von Kommunikationsdaten

Die Anomalieerkennung soll anhand der auf dem Backend gegebenen Daten, die durch eine Fahrzeugflotte gesammelt werden, stattfinden. Die Idee der Dissertation ist es nicht, vorher eine Anomalie zu definieren, die es zu finden gilt. Stattdessen wird eine Beschreibung des normalen Fahrzeugverhaltens gelernt. Dabei wird angenommen, dass die Flottendaten normal sind. Natürlich können sich auch darunter Ausreißer befinden, diese stellen jedoch nur einen geringen Anteil dar [CBK09]. Sollten sie einen größeren Anteil in Anspruch nehmen, wären es gemäß der Definition keine Anomalien. Für zukünftige Datensätze werden im Fahrzeug Anomalien als Abweichungen vom gelernten, normalen Verhalten erkannt, welches durch das Referenzmodell beschrieben wird. In der Anwendungsphase wird dieses verwendet, um die Klassen neuer Datenpunkte vorherzusagen. Das Ziel ist es, neue Zustände und Abweichungen von den Flottendaten zu erkennen. Somit wird das normale Fahrzeugverhalten nicht durch den Entwickler, sondern anhand der real eingefahrenen Daten bestimmt.

3.1 Ablauf Anomalieerkennung

Trainings- und Anwendungsphase sind ein wesentlicher Teil zur Erkennung von Anomalien in den Kommunikationsdaten. Jedoch ist weiteres Postprocessing notwendig. Bild 3.1 zeigt den Ablauf, wie Anomalien erkannt und bewertet werden können.

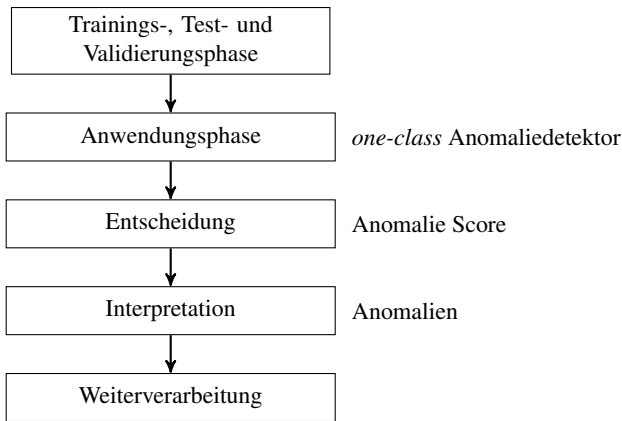


Bild 3.1: Ablauf Anomalieerkennung

Es wird ein *one-class* Anomaliendetektor trainiert, der die Trainingsdaten beschreibt und dabei annimmt, dass sie größtenteils normal sind. Liefert dieser in Test und Validierung ausreichende Genauigkeit, wird er verwendet, um in der Anwendungsphase für jedes Datum einen Anomalie Score zu berechnen. Die Scores beurteilen, wie anormal ein Datenpunkt ist. Soll jedoch eine klare Entscheidung getroffen werden, ob es sich um eine Anomalie handelt, reicht der Score alleine nicht aus. Eine Entscheidung ist beispielsweise nötig, wenn das Fahrzeug den Fahrer auf eine Anomalie hinweist oder wenn nur eine Anomalie an das Backend übertragen werden soll. Nachdem ein Datenpunkt als anormal klassifiziert wurde, folgt die Interpretation. Es muss beantwortet werden, warum die Anomalie auftritt und welche Signale sich außergewöhnlich verhalten. Dazu wird Expertenwissen benötigt, allerdings kann dieser durch die Vorverarbeitung der Anomalie auf das Wesentliche hingewiesen und dadurch unterstützt werden. Der letzte Schritt ist die Weiterverarbeitung des gewonnenen Wissens. Anomalien gleichen Typs werden zusammengefasst und in einen Kontext wie Zeit, Ort oder Fahrzeug gebracht. Beispielsweise sind erkannte Anomalien geeignet, um Fahrzeug-

besitzer frühzeitig vor möglichen Schadensfällen zu warnen. Wird die Verwendung eines Schalters als Anomalie erkannt, führt dies dazu, den Schalter in der nächsten Fahrzeuggeneration anders zu verbauen. Ein anderes Szenario ist eine ortsabhängige Anomalie, die z.B. Hinweise auf Straßenschäden gibt. Mithilfe der identifizierten Probleme können als weitere Maßnahme Anwendungsfälle abgeleitet werden, für die ein speziell optimierter Klassifikator gelernt wird. Dadurch wird ein Wissensapparat aufgebaut. Erkannte Anomalien, die schon bekannt sind, können in die entsprechende Klasse einordnet werden. Für neue, noch nicht identifizierte Anomalien, werden weitere Klassen geschaffen.

Die Anomalieerkennung dient demzufolge als Vorverarbeitungsschritt zur Einordnung der Daten. Anhand der erkannten Anomalien können gegebenenfalls Anwendungsfälle abgeleitet werden und neue, noch nicht bedachte Geschäftsideen entstehen. Auch werden Anomalien erkannt, die lediglich ein seltenes Ereignis darstellen. Diese sind nützlich, um den Datentransfer von Fahrzeug zu Backend sinnvoll zu reduzieren, da dieses dann nur mit fehlenden Daten angereichert wird.

3.2 Anomalieerkennung zur Datenselektion

Die Erkennung von anormalen Fahrzeugverhalten soll dazu dienen, bereits im Fahrzeug relevante Daten zu selektieren, die an das Backend übertragen werden. Die Datenmenge wird, nachdem das normale Verhalten gelernt wurde, auf die anormalen Ereignisse eingeschränkt. Bild 3.2 skizziert den Ansatz. Im ersten Schritt werden alle durch Experten definierten Kommunikationsdaten an das Backend gesendet, wo in der Trainingsphase das Referenzmodell gelernt wird (vgl. Bild 3.2a). Das trainierte Modell wird dann über die Luftschnittstelle in die Fahrzeuge übertragen, damit dort eine Abweichung als Anomalie erkannt und an das Backend übermittelt wird (vgl. Bild 3.2b).

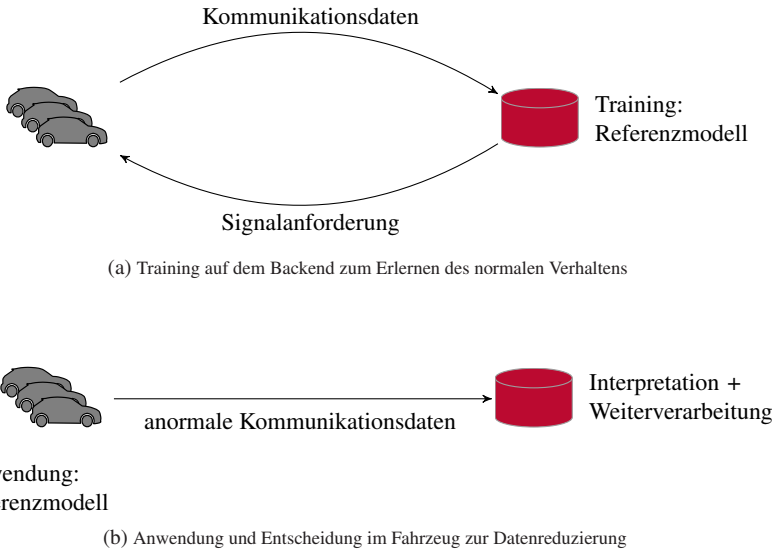


Bild 3.2: Anomalieerkennung zur Reduzierung des Datentransfers

Durch die Einschränkung auf Anomalien kann die von Fahrzeug an Backend übertragene Datenmenge reduziert werden. Es wird sichergestellt, dass es sich um außergewöhnliche Ereignisse handelt, die für das Backend neue Informationen liefern. Inwieweit die erkannten Anomalien zur Generierung von Wissen verwendet werden können, zeigt sich, wie in Kapitel 3.1 dargestellt, beim Postprocessing im Backend, bestehend aus Interpretation und der Weiterverarbeitung. Es wird bestimmt, ob es sich bei einer Anomalie nur um ein fehlendes Ereignis handelt, oder um eine Situation, die weitere Schritte zur Folge hat, wie z.B. die Kontaktierung des Fahrzeugbesitzers bei einem Schadensfall.

Die Übergänge zwischen Trainings- und Anwendungsphase sind zu definieren (vgl. Bild 3.3). Die Frage, wann das Referenzmodell fertig trainiert ist und vom Backend in die Fahrzeuge übertragen werden kann, hängt von der

Güte ab und wird in Kapitel 5.3.1 untersucht. Außerdem ergibt sich die Fragestellung, wie lange das Referenzmodell, welches in der Anwendungsphase verwendet wird, aktuell ist. Es muss festgestellt werden, wann eine neue Trainingsphase nötig ist, um dadurch die Daten dynamisch zu sammeln. Die Aktualität der Datenbasis auf dem Backend wird automatisiert sicherstellt, denn sobald der Referenzdatensatz veraltet ist, wird dieser im Rahmen einer neuen Trainingsphase aufgefrischt. Möglichkeiten hierfür werden als Teil des Ausblicks in Kapitel 7 behandelt.

Das Konzept der Anomalieerkennung wird in Kampagnen realisiert. Bild 3.4 verdeutlicht, dass sich die Schwarmdaten aus unterschiedlichen Baureihen zusammensetzen. Je Baureihe werden von Experten Signallisten definiert und die angeforderten Daten im Rahmen unterschiedlicher Kampa-

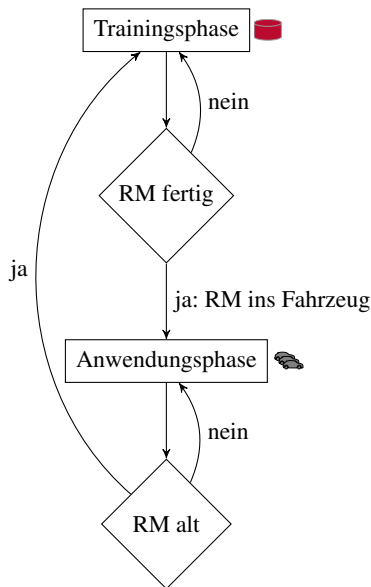


Bild 3.3: Übergänge zwischen Trainings- und Anwendungsphase zur dynamischen Datensammlung, RM: Referenzmodell

nen (K_x) gesammelt. Die innerhalb einer Kampagne aufgenommenen Daten werden als Flottendaten verwendet und dienen zum Training eines Referenzmodells.

Die Aufteilung in Trainings- und Kontrolldaten erfolgt zur Validierung und Anpassung der Hyperparameter, sodass das trainierte Modell die Mehrheit der Flottendaten als normal klassifiziert, denn nur dann kann es zur Reduzierung der Datenmenge im Fahrzeug verwendet werden. Da sowohl Trainings- als auch Kontrolldaten aus den Flottendaten gezogene Samples sind, wird angenommen, dass sie derselben Verteilung folgen. Wenn der Anteil erkannter Anomalien in etwa übereinstimmt, kann sichergestellt werden, dass das Referenzmodell weder *overfittet* (also nur für die Trainingsdaten funktioniert und alles andere als anormal erkennt), noch dass es zu stark generalisiert und alles als normal erkennt.

Das resultierende Referenzmodell wird in der Anwendungsphase im Fahrzeug verwendet. Alternativ stehen Testfälle zur Verfügung, die zur Evaluierung verwendet werden (siehe Kapitel 5). Ausgabe ist ein Anomalie Score, der den Grad der Anomalität darstellt. Anhand der Scores der Trainingsdaten wird ein Grenzwert festgelegt (vgl. Kapitel 4.2.1), sodass in der Anwendungsphase die Datenpunkte mit höherem Score an das Backend übertragen werden.

3.3 Anforderungen

Anomalien werden in verschiedenen Domänen und Forschungsbereichen erkannt [CBK09]. Eine Beurteilung der unterschiedlichen Verfahren zur Anomalieerkennung ist erforderlich. Sie müssen mit dem Ziel vereinbar sein, Anomalien im Fahrzeug basierend auf Flottendaten zu erkennen. Dabei müssen folgende Anforderungen erfüllt sein:

Übertragbarkeit ins Fahrzeug: Auf dem Backend wird eine Beschreibung des normalen Verhaltens anhand der Flottendaten trainiert. Zur Identi-

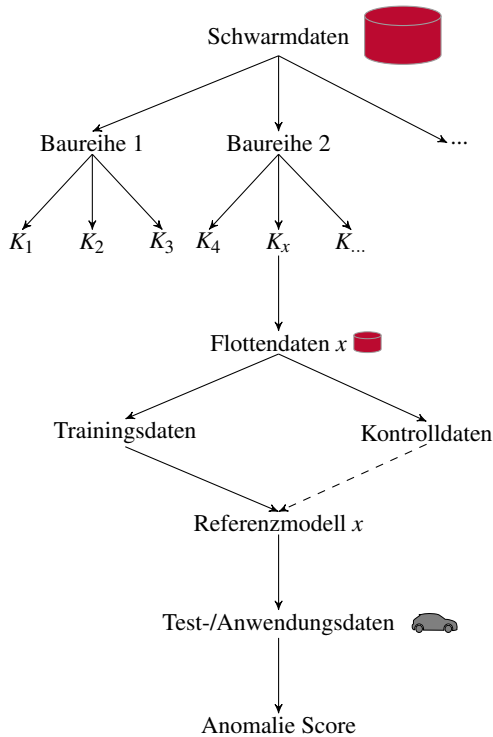


Bild 3.4: Realisierung verschiedener Referenzmodelle durch Kampagnen

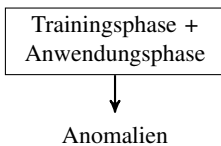
fikation von Anomalien soll diese sowohl im Backend als auch im Fahrzeug verwendet werden. Das bedeutet, es muss möglich sein, das im Backend trainierte Modell ins Fahrzeug zu übertragen.

In Bild 3.5 wird der Unterschied zwischen datenbasierter und modellbasierter Anomalieerkennung verdeutlicht. Im datenbasierten Ansatz findet keine Trennung zwischen Trainings- und Anwendungsphase statt; in der gegebenen Datenmenge wird nach Ausreißern gesucht. Wenn Anomalien im Fahrzeug erkannt werden sollen, sind die Anwendungsdaten allerdings klar von den Trainingsdaten abzugrenzen, da dort nicht auf die Flottendaten zugegrif-

fen werden kann. Der datenbasierte Ansatz erfüllt die Anforderung nicht. Der modellbasierte Ansatz als Alternative kann Anomalien auch im Fahrzeug detektieren, da eine Beschreibung der normal angenommenen Daten gelernt wird, die übertragbar ist.

Robustheit gegen Anomalien in Trainingsdaten: Die Annahme normaler Trainingsdaten ist sehr optimistisch und sieht in der Realität anders aus; auch auf dem Backend können Ausreißer liegen. Die Auffassung, dass die Menge an normalen Daten die wenigen Anomalien überdeckt, bedeutet, dass die unbekannt Anomalien in den Trainingsdaten nicht ins Gewicht fallen und daher wenig Einfluss auf das endgültige Referenzmodell haben. Trotzdem ist eine Methode vorzuziehen, die den Aspekt der *unreinen* Trainingsdaten einbeziehen kann. Somit gelten nicht alle Trainingsdaten als normal, sondern der Algorithmus definiert, welche Trainingsdaten zu außergewöhnlich sind und ignoriert diese zur Beschreibung der Normalität. Der in Bild 3.5 dargestellte datenbasierte Ansatz geht nicht von fehlerfreien Daten aus und ist daher robust. Er hat aber, wie im vorangegangenen Punkt gezeigt, Probleme mit der Übertragung ins Fahrzeug.

Datenbasierter Ansatz



Modellbasierter Ansatz

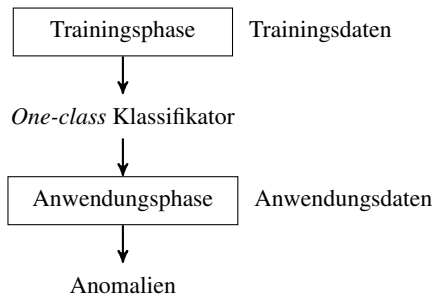


Bild 3.5: Aufteilung in Trainings- und Anwendungsphase

Unterstützung bei Interpretation: Eine erkannte Anomalie wird vom Fahrzeug auf das Backend übertragen. Dort muss sie auch interpretiert werden, um Aussagen treffen zu können, warum es sich um ein außergewöhnliches Ereignis handelt. Ein Verfahren, das bei der Interpretation unterstützen kann, ist vorteilhaft.

Anomalie Score als Ausgabe: Zur Einordnung eines Datenpunktes kann entweder ein Label, welches zwischen den Klassen *normal* und *anormal* unterscheidet, oder ein Anomalie Score gegeben sein. Ein Label als Ausgabe bedeutet, der trainierte *one-class* Anomaliedetektor kann direkt als Referenzmodell verwendet werden. Ein Score zeigt, wie wahrscheinlich der aktuelle Datenpunkt als Anomalie betrachtet werden kann. Er bietet daher höhere Flexibilität und ermöglicht eine Gewichtung der Anomalität. Allerdings muss zur Klassifikation eine Entscheidung getroffen werden, ab welcher Grenze zwischen *normal* und *anormal* unterschieden wird. Eine genauere Beschreibung zur Verwendung des Anomalie Scores wird in Kapitel 4.2.1 gezeigt.

4 Umsetzung der Anomalieerkennung

Anomalien werden in den unterschiedlichsten Bereichen erkannt. Beispiele sind die Erkennung von Netzwerk- oder Systemattacken (*Intrusion Detection*) oder die Entdeckung eines Betruges, insbesondere interessant für Banken und Versicherungen. Auch zur Filterung falscher Kreditkartentransaktionen spielt das Erkennen von Anomalien eine wichtige Rolle. In der Medizin können gefundene Anomalien bei der frühzeitigen Identifikation von Krankheiten und ungewöhnlichen Symptomen unterstützen. Aber auch in Bildern und Texten findet die Methode Verwendung [PY13].

Zur Erkennung von Anomalien werden Verfahren des maschinellen Lernens angewandt. Andere Gebiete, in der das maschinelle Lernen eingesetzt wird, sind Bildverarbeitung, Spracherkennung, die sogenannten *Recommender Systems* und auch Fahrerassistenzsysteme als Vorstufe des autonomen Fahrens [PAC15].

Der folgende Abschnitt stellt den aktuellen Forschungsstand der Anomalieerkennung vor. Außerdem findet die Übertragung auf anomales Fahrverhalten basierend auf Flottendaten statt. Die Umsetzung des in Kapitel 3 erarbeiteten Konzeptes wird dargelegt.

4.1 Verfahren zur Anomalieerkennung

Zur Erkennung von Anomalien in den Kommunikationsdaten werden die Signale pro Zeiteinheit zusammengefasst. Je Zeitpunkt ergibt sich daraus ein Signalvektor $\mathbf{x} \in \mathbb{R}^d$, der d unterschiedliche Signale beschreibt (vgl. Tabelle

2.2). Es besteht die Möglichkeit, die Vektoren unabhängig voneinander zu betrachten und damit Anomalien zu finden, die sich durch global gesehen, unerwartete Signalwerte (vgl. Bild 4.1a) oder gestörte Signalzusammenhänge (vgl. Bild 4.1b) ergeben.

Erst durch die Einbeziehung der zeitlichen Abhängigkeit zwischen den Vektoren werden die Fahrten als multivariate Zeitreihen verwendet (vgl. Bild 4.2). Dadurch beschränkt sich die Anomalieerkennung nicht nur auf statische Ereignisse, sondern es werden Ausreißer identifiziert, die aufgrund ihres Verlaufs über die Zeit auffallen. Ein weiteres Unterscheidungskriterium ist die Einordnung in punktuell (vgl. Bild 4.1) und kontextuell (vgl. Bild 4.2) [CBK09].

Der Unterschied zwischen statischer und dynamischer Anomalieerkennung in den Kommunikationsdaten wird durch folgende Beispiele verdeutlicht: Bild 4.3a zeigt ein Histogramm des Signals *Tankfüllstand*. Unabhängig von anderen Signalen oder dem Verlauf über die Zeit erkennt man, dass ein Füllstand unter 10% nicht vorkommt und diese Situation daher für zukünftige Beobachtungen als statische Anomalie gilt. Bilder 4.3b und 4.3c zeigen

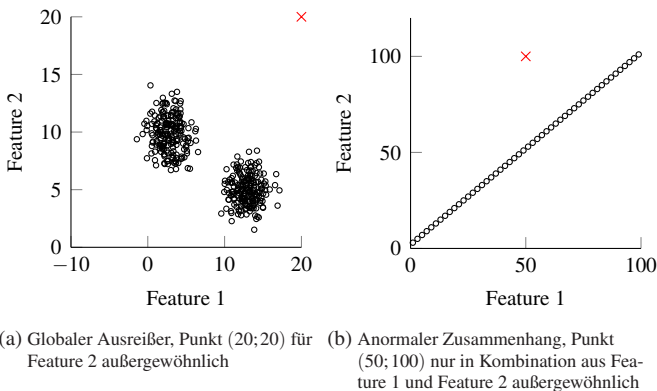


Bild 4.1: Statische Anomalieerkennung

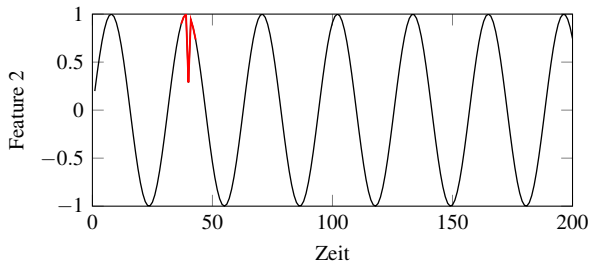


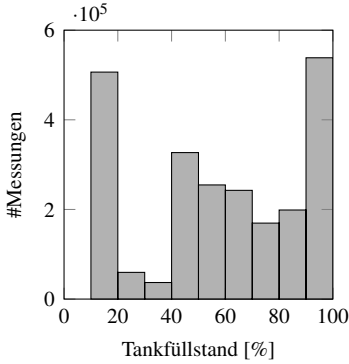
Bild 4.2: Anomales Verhalten im Zeitverlauf, unerwarteter Einbruch zum Zeitpunkt 40

den Zusammenhang zwischen den Signalen *Motoranzeige Verbrauch* und *Motordrehzahl*. Sie korrelieren und ein zu großer Abstand zu einem gewissen Zeitpunkt deutet auf eine statische Anomalie hin (vgl. Bild 4.3b). Ein Auseinanderdriften im Laufe der Zeit lässt auf eine dynamische Anomalie schließen (vgl. Bild 4.3c). Ein anderes Beispiel für einen anormalen Zeitverlauf ist ein unverhältnismäßig großer Sprung im Kilometerstand, wie in Bild 4.3d dargestellt.

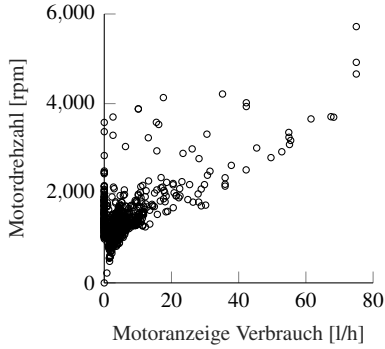
In Kapitel 4.1.1 wird zunächst ein Überblick über die statischen Methoden zur Erkennung von Anomalien gegeben. Da die Kommunikationsdaten jedoch als multivariate Zeitreihe behandelt werden können, wird anschließend in Kapitel 4.1.2 dargestellt, wie sich die Methoden auf zeitabhängige Daten übertragen lassen.

4.1.1 Statische Anomalieerkennung

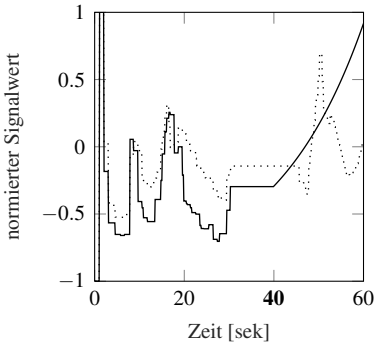
Im Folgenden werden die Formulierungen d Features und d Signale äquivalent verwendet. Ein Signalvektor $\mathbf{x} \in \mathbb{R}^d$ beschreibt den Zustand der d Signale zu einem gewissen Zeitpunkt. Da angenommen wird, dass die Vektoren unabhängig voneinander sind, geht der Zeitbezug verloren und man spricht von statischer Anomalieerkennung. Übertragen auf Tabelle 2.2 bedeutet das, die Zeilen unabhängig voneinander zu verstehen.



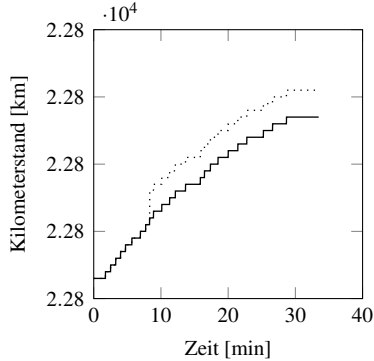
(a) Verteilung des Tankfüllstands



(b) Scatterplot Motoranzeige Verbrauch und Motordrehzahl, statische Betrachtung



(c) Motoranzeige Verbrauch (durchgezogen) und Motordrehzahl (gepunktet) in einer Minute (normiert), zeitabhängige Betrachtung



(d) Kilometerstand während etwa 30 Minuten Fahrt, durchgezogen: normaler Verlauf, gepunktet: unerwarteter Sprung

Bild 4.3: Vergleich von statischen Anomalien (a,b) und Anomalien im Zeitverlauf (c,d) am Beispiel der Kommunikationsdaten im Fahrzeug

Anhand der n Trainingsvektoren $\{\mathbf{x}_i\}_{i=1}^n$ wird die Unterscheidung der Klassen *normal* und *anormal* gelernt. Der zu überprüfende Datenpunkt wird als Testvektor $\mathbf{x}_v \in \mathbb{R}^d$ beschrieben. Zur Einordnung kann ein Klassifikator gelernt werden, der die Klasse entsprechend vorhersagt. Eine Alternative ist die Berechnung eines Anomalie Scores für den Punkt \mathbf{x}_v . Dieser zeigt, wie *anormal* der Datenpunkt ist und kann beim Vergleich mit anderen Punkten verwendet werden, um diese nach ihrer Normalität zu sortieren. Die Entscheidung, ob es sich um eine Anomalie handelt, wird dann nachgelagert durch die Bewertung des Scores getroffen.

Bild 4.4 zeigt, wie die Verfahren zur Anomalieerkennung unterschieden werden. Sie kann basierend auf einem statistischen Ansatz, einem nächsten Nachbar Ansatz, einem Clustering Ansatz, einem Klassifikationsmodell oder einem Isolation Forest erfolgen [CBK09, LTZ08].

Die Methode des nächsten Nachbarn bewertet einen Datenpunkt als *anormal*, falls die ihm nahestehenden Datenpunkte weit entfernt sind. In der Literatur wird an dieser Stelle häufig zwischen dichte- und distanzbasiert unterschieden [CBK09, Zha13]. Unter Clustering versteht man das Gruppieren von Datenpunkten, die sich ähnlich sind. Dabei kann die Ähnlichkeit zweier Punkte durch deren Abstand berechnet werden. Zur Kategorie der Klassifikationsmodelle gehören sowohl (*one-class*) Support Vektor Machines (SVMs) [AGA13] als auch Replikator Neuronale Netze (RepINN) [DCS14, HHWB02]. In beiden Ansätzen wird mithilfe der Trainingsdaten eine Funktion trainiert, die das normale Verhalten charakterisiert. Der Funktionswert an der Stelle des Testvektors \mathbf{x}_v als Anomalie Score entscheidet über dessen Einordnung in *normal* oder *anormal*. Eine eigene Kategorie bildet der Isolation Forest. Dieser beinhaltet eine Menge von Entscheidungsbäumen, für die angenommen wird, dass Anomalien leichter vom Rest der Daten zu kapseln sind und daher weit oben in den Bäumen stehen [LTZ08, LTZ12]. Eine genauere Darstellung der einzelnen Vorgehensweisen wird im folgenden Abschnitt gegeben.

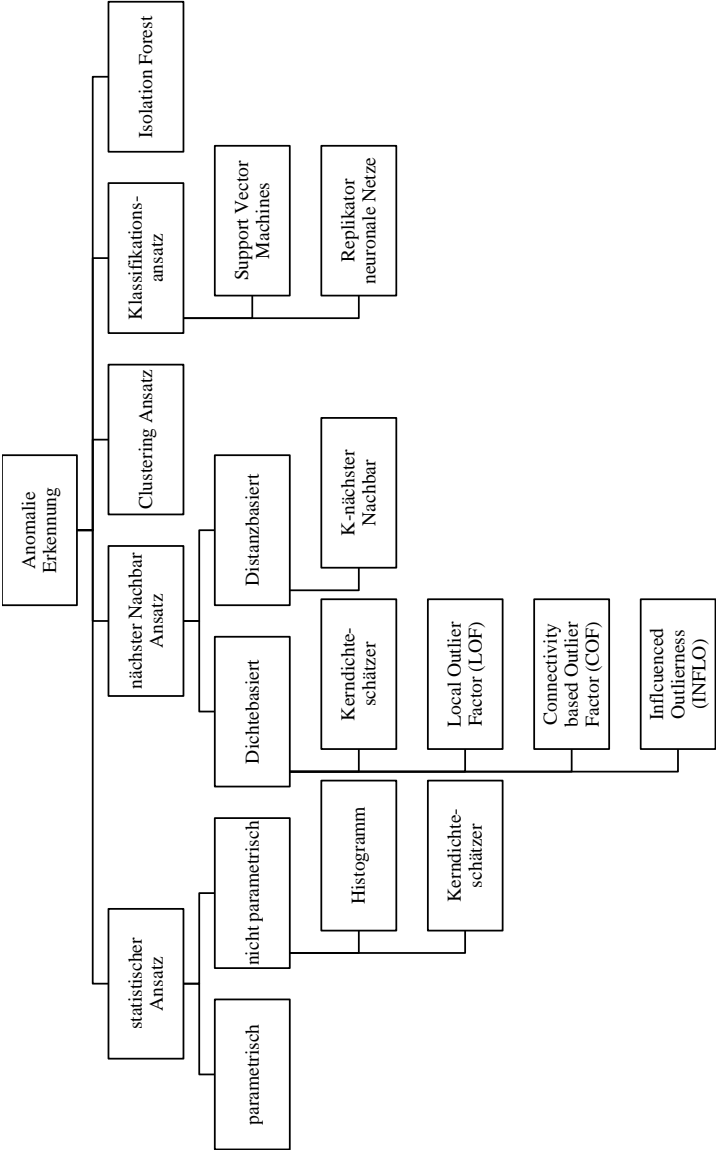


Bild 4.4: Methoden zur Anomalieerkennung

In allen Ansätzen wird zwischen Trainings- und Anwendungsphase unterschieden. Das bedeutet, es wird zunächst eine Beschreibung der Trainingsdaten gelernt (= Trainingsphase) und danach der Testvektor klassifiziert (= Anwendungsphase). Ausnahme ist der nächste Nachbar Ansatz. Hierbei ist der Testvektor \mathbf{x}_v Teil der Trainingsdaten. Es wird keine Beschreibung der Daten gelernt, stattdessen wird der Testvektor in Relation zum Rest der Daten gesetzt. Für alle anderen Techniken ist die Trennung zwischen Training und Anwendung möglich.

Die Verfahren, die eine Beschreibung der Daten lernen, werden als *one-class* Klassifikatoren verwendet. Der Klassifikator ist in der Lage, die Klasse neuer Datensätze vorherzusagen, während im Training nur die Daten einer Klasse hinzugezogen werden. Es handelt sich dabei um den in Bild 3.5 dargestellten **modellbasierten Ansatz**, da Anomalien als Abweichungen vom erlernten Referenzmodell erkannt werden [LLP07]. *One-class* Klassifikatoren sind gute Alternativen zu normalen Klassifikatoren, wenn eine Klasse stark unterrepräsentiert ist [DOSB10]. Dies gilt für Anomalien, da sie selten auftreten und wenige Fehlerdaten zur Verfügung stehen, die eindeutig als anomal gelabelt sind. Im Fall der Anomalieerkennung wäre es problematisch einen binären *two-class* Klassifikator zu trainieren, da unterschiedliche Typen von Anomalien nicht zu einer Klasse zusammengefasst werden können [MMHH⁺11]. Zum Beispiel handelt es sich sowohl bei einem Getriebebeschaden und als auch bei einem Unfall um eine Anomalie. Die Zusammenfassung zu einer Gruppe und das Trainieren eines *two-class* Klassifikators, welcher zwischen den Klassen *normal* und *Unfall/Getriebebeschaden* unterscheidet ist allerdings nicht zielführend. Die Signalverläufe der beiden Ereignisse weisen unterschiedliche Muster auf.

Die Vor- und Nachteile der Methoden werden in Tabelle 4.1 verglichen. Neben den Anforderungen aus Kapitel 3.3 (Übertragbarkeit in Fahrzeug, Robustheit gegen Anomalien in den Trainingsdaten, Unterstützung bei der Interpretation und Anomalie Score als Ausgabe) spielen auch andere Eigenschaften eine Rolle. Ein Anomalie Score bei dem es sich um eine Wahr-

scheinlichkeit handelt, erleichtert die Interpretation. Auch stellt sich die Frage, wie gut die Methode mit dem *Fluch der Dimensionalität* (vgl. Kapitel 2.2), also mit vielen unterschiedlichen, gleichzeitig betrachteten Signalen, umgehen kann. Die Verwendung von Dichte oder Abstand als Kriterium hat zur Folge, dass im höher dimensionalen Raum mit vielen irrelevanten Features, diese die relevanten Unterschiede verwischen. Dadurch lassen sich Anomalien nicht mehr von normalen Punkten unterscheiden [ZSK12, KM10]. Die hohe Dimensionalität bedeutet außerdem, dass die Verteilung der Daten nicht a-priori bestimmt werden kann. Da auch anormale Signalzusammenhänge erkannt werden sollen, sind Methoden, die voneinander abhängige Signale erwarten, zu bevorzugen. Ebenfalls werden Berechnungskomplexität und die Anzahl zu definierender Hyperparameter verglichen. Die Berechnungskomplexität gibt Aufschluss über die Trainingsdauer. Die Anzahl der Hyperparameter zeigt, wie viele zusätzliche Parameter optimiert werden müssen. Dies geschieht meist durch die manuelle Suche des Optimums innerhalb benutzerdefinierter Grenzen oder durch eine systematische Raster-suche, speziell angepasst an den Anwendungsfall [BB12]. Für Anwendungen, die einen hohen Automatisierungsgrad voraussetzen, ist die Einstellung der Hyperparameter eine besondere Herausforderung. Ein hoher Automatisierungsgrad erschwert die anwendungsorientierte Optimierung [CAH16]. Da im Rahmen der Arbeit Anomalien allgemein als Abweichung erkannt werden und eine Optimierung auf spezielle Anwendungsfälle erst zu einem darauffolgenden Zeitpunkt stattfindet, sind wenige Parameter vorteilhaft.

Statistischer Ansatz

Eine Möglichkeit zur Erkennung von Anomalien ist ein statistischer, parametrischer Ansatz. Dabei wird a-priori der Verteilungstyp bestimmt und die Parameter mithilfe der Daten gesetzt [CBK09, LLP07]. Anhand der Wahrscheinlichkeitsverteilung werden Punkte mit geringer Wahrscheinlichkeit als Anomalie identifiziert. Der Nachteil dieser Methode ist die Tatsache,

Tabelle 4.1: Methodenvergleich

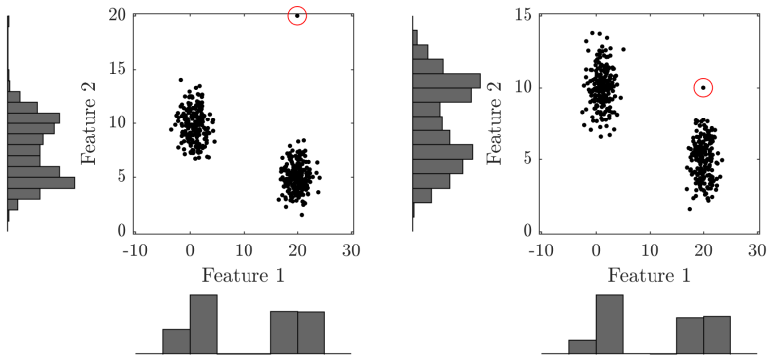
	Methoden					
	statistischer Ansatz	nächster Nachbar Ansatz	Clustering Ansatz	one-class SVM	Replikator Neural Net	Isolation Forest
Übertragbarkeit ins Fahrzeug	✓		✓	✓	✓	✓
Robustheit gegen Ausreißer in den Trainingsdaten	✓	✓	✓	✓		✓
Unterstützung bei der Interpretation	✓		✓		✓	
Anomalie Score als Ausgabe	✓	✓	✓	✓	✓	✓
Wahrscheinlichkeit als Anomalie Score	✓					
Besonders geeignet für hoch dimensionalen Feature Raum				✓	✓	✓
A-priori Bestimmung der Verteilung nicht nötig		✓	✓	✓	✓	✓
Signale werden nicht unabhängig voneinander betrachtet		✓	✓	✓	✓	✓
Berechnungskomplexität, Training	–	naiv: $O(n^2)$	kmeans: $O(k \cdot n \cdot i)$	$O(N^2)$, $O(N^3)$	–	$O(i \cdot \psi \log \psi)$
Wenige Hyperparameter		✓	✓			✓

dass die Wahrscheinlichkeitsverteilung a-priori bestimmt werden muss. Die Signale können als diskrete bzw. kontinuierliche Zufallszahlen unterschiedlicher Verteilungen betrachtet werden, für die eine gemeinsame kumulative Wahrscheinlichkeitsverteilung bestimmt wird. Die vereinfachte Annahme der Unabhängigkeit der Signale führt zur Identifikation einer Wahrscheinlichkeitsverteilung pro Signal. Beispiel ist die Voraussetzung der Normalverteilung [WVC⁺11].

Die a-priori Annahme der Verteilungsfunktion kann durch nicht parametrische Ansätze wie die Verwendung von Histogrammen [GD12] oder Kerndichteschätzer [LLP07] umgangen werden. Histogramme werden eingesetzt, um die Wahrscheinlichkeitsverteilung für jedes Signal a-posteriori mithilfe der Daten zu schätzen. Der in [GD12] vorgestellte Ansatz nimmt die Unabhängigkeit der Signale an und lernt jeweils ein Histogramm. Die approximierten Wahrscheinlichkeitsverteilungen werden daraufhin zur Berechnung des Anomalie Scores verwendet. Globale Ausreißer werden dadurch erkannt, allerdings werden anormale Zusammenhänge übersehen [GU16, GD12]. In den Beispielen in Bild 4.5 sind die Datenpunkte (20;20) und (20;10) globale Ausreißer. Betrachtet man die Features unabhängig voneinander, verhält sich der Ausreißer in Bild 4.5a für Feature 2 außergewöhnlich und wird als Anomalie erkannt. Die Tatsache, dass die Kombination aus Feature 1 und Feature 2 für den Datenpunkt (20;10) in Bild 4.5b ungewöhnlich ist, kann unter der Annahme voneinander unabhängiger Features allerdings nicht erkannt werden. Für keines der Features fällt der Datenpunkt individuell auf.

Bei Kerndichteschätzern wird die Wahrscheinlichkeitsdichte \tilde{p} des Datenpunktes \mathbf{x}_v mithilfe der n Trainingsdaten approximiert [LLP07, TS92, Par62]:

$$\tilde{p}(\mathbf{x}_v) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h(\mathbf{x}_i)^d} K\left(\frac{\mathbf{x}_v - \mathbf{x}_i}{h(\mathbf{x}_i)}\right) \quad (4.1)$$



(a) Globaler Ausreißer, Feature 2

(b) Globaler Ausreißer, Kombination Feature 1 und Feature 2

Bild 4.5: Histogrammbasierte Anomalieerkennung

Hierbei steht K für die Kernelfunktion und $h(x_i)$ für die Bandbreite von Datenpunkt \mathbf{x}_i . Im naiven Fall mit $h(\mathbf{x}_i) = h$ und

$$K(y) = \begin{cases} 0,5 & \text{falls } |y|_\infty \leq 1 \\ 0 & \text{sonst} \end{cases}, \quad (4.2)$$

wird somit ein d -dimensionaler Würfel um den zu bewertenden Punkt \mathbf{x}_v als Mittelpunkt gelegt und der Anteil der darin liegenden Datenpunkte berechnet (vgl. Bild 4.6). Je höher der Anteil, desto unwahrscheinlicher handelt es sich um eine Anomalie. Form und Breite des Hyperraums können durch die Kernelfunktion und dynamische Bandbreite adaptiert werden. Formel 4.1 lässt erkennen, dass die Wahrscheinlichkeit des Datenpunktes \mathbf{x}_v anhand der n Trainingsdaten berechnet wird.

Bewertung: Bei statistischen Methoden gestaltet sich die Übertragbarkeit ins Fahrzeug nur beim Kerndichteschätzer schwierig, da es sich um einen datenbasierten Ansatz handelt. Die restlichen Ansätze können sowohl

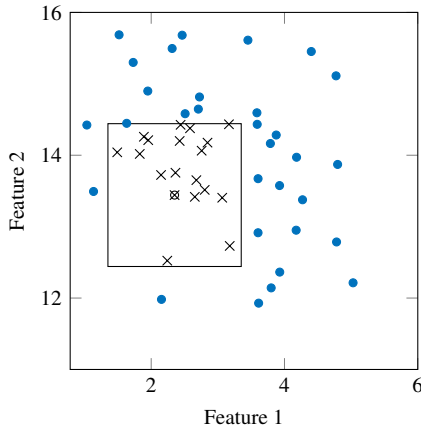


Bild 4.6: Kerndichteschätzer

daten- als auch modellbasiert verwendet werden, denn die Verteilungsfunktion wird anhand der Trainingsdaten berechnet und anormale Situationen werden durch kleine Wahrscheinlichkeiten repräsentiert. Die Robustheit gegen Anomalien in den Trainingsdaten ist gegeben, da nicht angenommen wird, dass sie ausschließlich normale Ereignisse repräsentieren. Die Wahrscheinlichkeit, dass der aktuelle Datenpunkt \mathbf{x}_v eine Anomalie ist, ergibt sich im histogrammbasierten Ansatz durch die Zusammensetzung der Wahrscheinlichkeiten der voneinander unabhängig angenommenen Signalwerte, was eine Interpretation erleichtert. Die Signale, die einen großen Beitrag zur Gesamtwahrscheinlichkeit leisten, werden erkannt. Statistische Ansätze liefern als Ausgabe wie wahrscheinlich es sich bei einem Datenpunkt um eine Anomalie handelt, was einem Score gleichzusetzen ist.

Somit sind die Anforderungen zur Anomalieerkennung in Kommunikationsdaten durch parametrische Ansätze und Histogramme erfüllt. Besonderer Vorteil ist die Tatsache, dass der Anomalie Score als Wahrscheinlichkeit leichter zu deuten ist. Ihre Nachteile liegen in der a-priori Bestimmung

der Verteilung der Daten bzw. in der Auffassung voneinander unabhängiger Signale. Die a-priori Bestimmung der Datenverteilung bedeutet nicht nur die Optimierung der Hyperparameter, sondern auch die Definition des zugrundeliegenden Modells, was insbesondere im hoch dimensionalen Raum schwierig ist.

Nächster Nachbar Ansatz

Der Ansatz des nächsten Nachbarn wird in der Literatur in dichte- und distanzbasierte Methoden unterteilt. Die Idee der Kerndichteschätzer kann auch als dichtebasiertes Verfahren eingestuft werden. Eine andere dichtebasierte Kenngröße ist der *Local Outlier Factor* (LOF). Dabei ergibt sich der Anomalie Score aus dem Verhältnis zwischen der durchschnittlichen lokalen Dichte der k nächsten Nachbarn des zu betrachtenden Datenpunktes \mathbf{x}_v und der eigenen Dichte. Ein großer Unterschied deutet auf eine Anomalie hin. Die lokale Dichte eines Datenpunktes ist der Kehrwert der durchschnittlichen Erreichbarkeitsdistanz zu seinen nächsten Nachbarn. Unter der Erreichbarkeitsdistanz von Punkt \mathbf{x}_A zu \mathbf{x}_B wiederum wird das Maximum des Abstandes zwischen den Punkten und des Abstandes von Datenpunkt \mathbf{x}_B zu dem k -nächsten Nachbarn verstanden [BKNS00].

Der LOF kann zwischen den beiden Fällen in Bild 4.7 unterscheiden. Der Abstand vom roten Punkt zu seinem nächsten Nachbarn ist in beiden Beispielen gleich, als anormal gilt er allerdings nur in Bild 4.7a.

Probleme hat das LOF-Verfahren, wenn die Dichte einer Anomalie ähnlich wie die seiner Nachbarn ist (vgl. Bild 4.8) oder in Situationen in denen Cluster mit unterschiedlicher Dichte nahe beieinander liegen (vgl. Bild 4.9). Varianten wie der *Connectivity based Outlier Factor* (COF) [TCFC02] oder der *Influenced Outlierness* (INFLO) [JTH⁺06] wurden entwickelt, um diese zu überwinden.

Wird die Distanz als Vergleichswert verwendet, wird der Abstand zu einer vorher definierten Anzahl von k Nachbarn betrachtet. Je näher die k nächsten

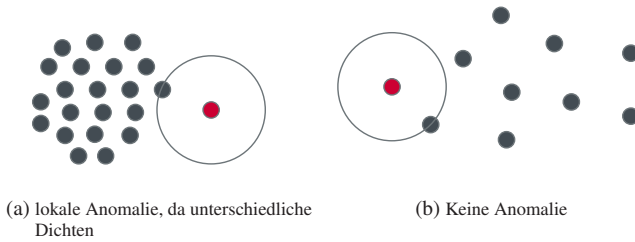


Bild 4.7: Local Outlier Factor



Bild 4.8: Nachteil LOF: Ähnliche Dichten nach [Zha13]

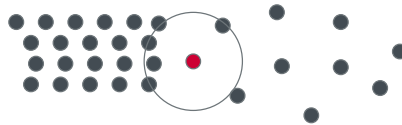


Bild 4.9: Nachteil LOF: Nähe unterschiedlicher Dichten nach [Zha13]

Nachbarn liegen, desto weniger wahrscheinlich stellt der Datenpunkt \mathbf{x}_v eine Anomalie dar [CBK09].

Bewertung: Die Vor- und Nachteile des Kerndichteschätzers lassen sich auf den nächsten Nachbar Ansatz übertragen. Es gibt keine Trennung zwischen Trainings- und Anwendungsphase (vgl. Bild 3.5), stattdessen wird der Testvektor \mathbf{x}_v direkt mit den Trainingsdaten verglichen und ein zu großer Abstand zum Rest der Daten deutet auf die Anomalie hin. Daraus ergibt sich auch die Problematik der Übertragbarkeit ins Fahrzeug. Der zu prüfende Testvektor kann im Fahrzeug definiert, dort jedoch nicht mit allen Flottendaten auf dem Backend verglichen und in Relation gesetzt werden, da diese nicht in den einzelnen Fahrzeugen zugänglich sind. Zur Anomalieer-

kennung in Kommunikationsdaten muss daher eine der anderen Methoden verwendet werden. Der nächste Nachbar Ansatz ist robust gegen Anomalien in den Trainingsdaten, da nicht von normalen Trainingsdaten ausgegangen wird. Für dichte-basierte Verfahren ist es schwierig nachzuvollziehen, welche Signale zur Anomalie führen, womit der Anwender nicht bei der Interpretation unterstützt wird. Die Ausgabe des Algorithmus ist eine Kennzahl, die anhand der Dichte bzw. der Distanz beschreibt, wie stark die Anomalität ist. Demzufolge handelt es sich um einen Anomalie Score. Ein weiterer Nachteil des nächsten Nachbar Ansatzes ist die hohe Rechenkomplexität von $O(n^2)$. Im einfachsten Fall muss jeder Datenpunkt mit jedem verglichen werden, was gerade für große Datenmengen nicht skaliert. Zwar verringern Lösungen wie das Bilden von Subsamples dieses Problem, dennoch sind im Bezug auf die Rechenkomplexität andere Verfahren geeigneter [CBK09]. Die Anzahl zu definierender Parameter wiederum ist klein. Für das LOF-Verfahren beispielsweise muss lediglich die Anzahl der Nachbarn optimiert werden. Probleme hat der Ansatz bei vielen unterschiedlichen Features. Mit steigender Dimension konvergiert der Abstand zum nächsten Nachbarn gegen den Abstand zum weit entferntesten Nachbarn, was dazu führt, dass der Kontrast zwischen den Abständen verschiedener Datenpunkte verblasst [BGRS99].

Clustering Ansatz

Das Clustering der Datenpunkte ist eine weitere Vorgehensweise zur Anomalieerkennung. Die Bewertung eines Datenpunktes kann von der Größe und Dichte des Clusters zu dem der Punkt gehört, als auch vom Abstand zum Clustermittelpunkt abhängen [CBK09]. Dabei wird unter der Größe die Anzahl der Datenpunkte im Cluster verstanden. Die Dichte beschreibt wie weit die Punkte eines Clusters voneinander entfernt sind. Eine Unterscheidung in Trainings- und Anwendungsphase ist möglich, indem im Training die Clus-

ter gelernt werden und der Datenpunkt \mathbf{x}_v dann in der Anwendungsphase durch eines der Cluster kategorisiert wird.

Ein gängiges Clusterverfahren ist die *k-means* Methode, bei dem die Trainingsdaten iterativ in eines der k Cluster eingeordnet werden. Die Clustermittelpunkte als Mittelwert der zugehörigen Datenpunkte werden entsprechend angepasst. Bild 4.10a zeigt beispielhaft wie Daten in zwei Cluster unterteilt werden.

Beim *EM*¹-Clustering handelt es sich um ein statistisches Verfahren, in dem die k Cluster unterschiedliche Normalverteilungen beschreiben, deren Parameter anhand der Daten gelernt werden [MB88]. Bild 4.10b zeigt an einem Beispiel das Erlernen von zwei Normalverteilungen.

Der in [EHPSX96] vorgestellte DBSCAN²-Ansatz ist ein dichtebasiertes Verfahren, welches zwei Datenpunkte clustert, wenn ein Punkt mindestens eine definierte Anzahl an Nachbarn innerhalb eines bestimmten Radius hat und der andere Punkt direkt oder indirekt (d.h. über weitere Punkte verbunden) in dieser Nachbarschaft liegt. Datenpunkte, die keinem Cluster zugewiesen werden können, werden als anomal bzw. Rauschen eingestuft (vgl. Bild 4.10c). Diese Clustering Methode kann jedoch auch als nächster Nachbar Ansatz interpretiert werden, da die entstehenden Cluster ausschließlich anhand der Datenpunkte beschreibbar sind. Es werden keine Eigenschaften wie Clustermittelpunkt oder -größe gelernt, stattdessen wird der Testvektor \mathbf{x}_v als Teil der Trainingsdaten untersucht. Eine Beschreibung der Daten als Ergebnis der Trainingsphase ist nicht gegeben.

Der Nachteil der Clusterverfahren ist, dass sie darauf ausgelegt sind, Cluster anstatt Anomalien zu finden [HXD03]. Als Lösung des Problems wird ein *Cluster Based Local Outlier Factor* (CBLOF) vorgestellt. Nach dem Gruppieren der Trainingsdaten wird für Datenpunkt \mathbf{x}_v der CBLOF berechnet. Dabei wird zwischen großen und kleinen Clustern unterschieden, indem die Cluster absteigend nach der Anzahl eingeschlossener Trainingsdaten sortiert

¹ EM - Expectation Maximation

² DBSCAN - Density-based spatial clustering of applications with noise

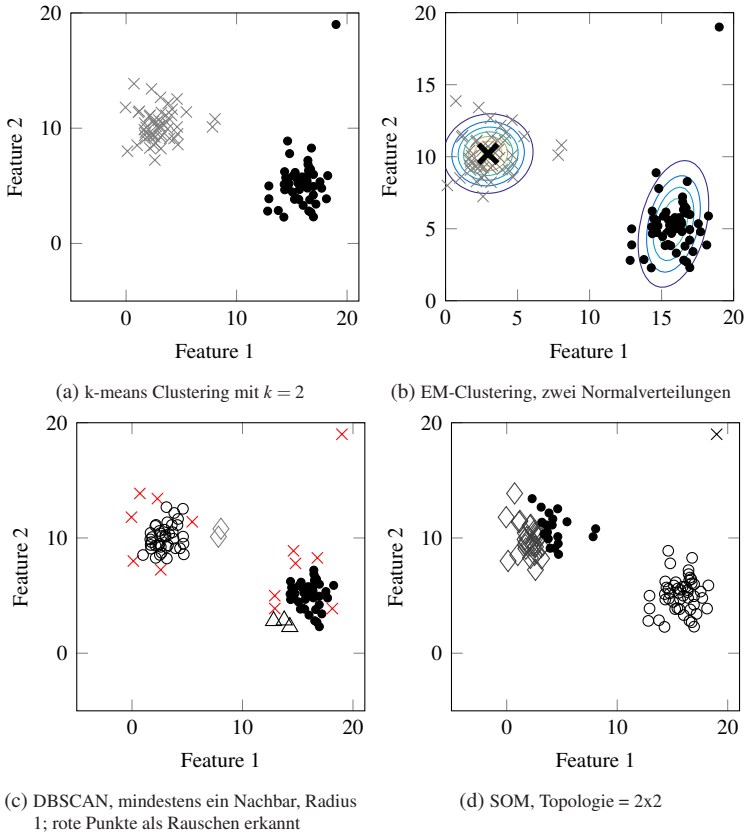


Bild 4.10: Vergleich verschiedener Clustermethoden

werden und die ersten b Cluster, die in Summe einen definierten Anteil aller Trainingsdaten einnehmen, als groß gelten. Der Faktor für Datenpunkt \mathbf{x}_v ergibt sich aus dem Produkt von Anzahl der Datenpunkte im selben Cluster und dem Abstand von \mathbf{x}_v zum Zentrum des nächsten großen Clusters [HXD03].

Das Clustering mithilfe von *Self Organizing Maps* (SOM) wird in [TAP14, BBCL14] und [Yan11] angewandt. Bild 4.11 verdeutlicht, dass es sich dabei um ein neuronales Netz handelt, das im Training den Abstand zwischen Datenpunkt \mathbf{x}_i und den Gewichtsvektoren $\mathbf{w}_j, j = 1, \dots, M$ der M Ausgabeneuronen verwendet. Beginnend mit zufälligen Werten für $\mathbf{w}_j^{(0)}$ wird iterativ das Neuron gefunden, dessen Gewichtsvektor \mathbf{w}_k minimalen Abstand zum aktuellen Trainingspunkt \mathbf{x}_i hat (*Best Map Unit* (BMU)). Die durch die Topologie definierten Nachbargewichte und die der BMU werden angepasst, sodass sie sich \mathbf{x}_i annähern [TAP14, Koh90, CFP98]:

$$\begin{aligned} \mathbf{w}_j^{(i+1)} &= \mathbf{w}_j^{(i)} - \alpha^{(i)} \Lambda^{(i,j,k)} (\mathbf{w}_j^{(i)} - \mathbf{x}_i), \\ & \quad i = 1, \dots, n \\ & \quad j = 1, \dots, M, \end{aligned} \tag{4.3}$$

mit n als Anzahl der Trainingsdatensätze, der Parameter $\alpha^{(i)} \in (0, 1)$ dient als Lernrate. Die Entfernungsgewichtungsfunktion $\Lambda^{(i,j,k)}$ zeigt, wie stark die Anpassung für Neuron j ist, welche vom Abstand zur BMU (Neuron k) abhängt. Bild 4.10d zeigt beispielhaft, wie die Datenpunkte geclustert werden, wenn eine SOM mit vier Ausgabeneuronen modelliert wird.

Nach dem Training werden zur Berechnung des Anomalie Scores für den Datenpunkt \mathbf{x}_v die m nächsten Neuronen (BMUs) selektiert, davon die mit wenigen Treffern im Training entfernt und anhand der verbleibenden Neuronen der durchschnittliche *Quantization Error* (QE) berechnet [TAP14]. Der

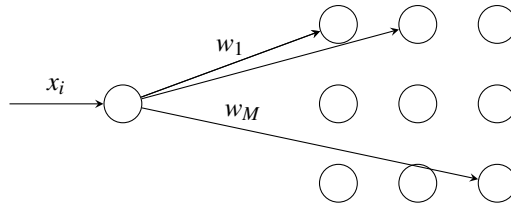


Bild 4.11: Self-organising Map, $w_j \in \mathbb{R}^d, j = 1, \dots, M, M = \text{Anzahl Ausgabeneuronen}$

Quantization Error ist der Abstand vom Testvektor \mathbf{x}_v zum Gewichtsvektor der k -ten BMU \mathbf{w}_k , für $k = 1, \dots, m$:

$$QE_k = \|\mathbf{x}_v - \mathbf{w}_k\|_2^2 \quad (4.4)$$

In [BBCL14] wird der *Quantization Error* mit den Abständen der normalen Trainingsdaten zu ihren BMUs verglichen, um den Datenpunkt als *normal* oder *anormal* zu klassifizieren. Zur Identifikation lokaler Ausreißer werden zusätzlich für jedes Cluster die Abstände zwischen den zugeordneten Trainingsdaten und der BMU als Clustermittelpunkt zur Bewertung verwendet. Das kann als Beschreibung der Dichte interpretiert werden. [Yan11] berechnet mithilfe des Medians der Gewichte aller Neuronen \mathbf{w}_{median} einen Anomalie Score. Es wird für den zu überprüfenden Datenpunkt \mathbf{x}_v die BMU selektiert und anhand dessen Abstand zu \mathbf{w}_{median} bestimmt, ob es sich um eine Anomalie handelt.

Bewertung: Die Methode des Clustering kann datenbasiert verwendet werden, was bedeutet, dass der zu testenden Datenpunkt \mathbf{x}_v als Teil der Trainingsdaten bewertet wird. Dadurch wiederum ergibt sich wie beim nächsten Nachbar Ansatz das Problem der Übertragbarkeit ins Fahrzeug. Hingegen kann eine trainierte abstrahierte Beschreibung der Daten, durch beispielsweise Clustermittelpunkt und -größe, übertragen werden. Damit sind beide in Bild 3.5 dargestellten Ansätze realisierbar. Obwohl nicht alle Trainings-

daten als *normal* angenommen werden, ist die Robustheit gegen Anomalien in Trainingsdaten nicht direkt gegeben, da Anomalien in den Trainingsdaten die Zusammensetzung der Cluster beeinflussen. Allerdings können Cluster mit wenigen Datenpunkten, als anormal gelten und entfernt werden, wodurch sich eine Robustheit ergibt. Der Anwender kann bei der Interpretation unterstützt werden, da es sich bei kleinen Clustern gegebenenfalls um dieselbe Art von Anomalie handelt. Außerdem kann geprüft werden, welche Features eines anormalen Datenpunktes am stärksten vom zugehörigen Clustermittelpunkt abweichen, was die Interpretation erleichtert. Auch ein Anomalie Score als Ausgabe ist gegeben. Er kann aus dem Abstand zwischen Datenpunkt \mathbf{x}_v und Clustermittelpunkt berechnet werden. Sind die Cluster erst einmal trainiert, ist die Anwendung schnell, da der Testdatenpunkt \mathbf{x}_v nur mit den Clustermittelpunkten verglichen werden muss. Die Dauer des Trainings und die Anzahl der zu optimierenden Hyperparameter hängt von der jeweiligen Methode ab. Die Komplexität des *k-means* Clustering beispielsweise ist linear in der Anzahl an Trainingsinstanzen, der Anzahl an Clustern und der Anzahl an Iterationen. Jeder Datenpunkt wird mit jedem Clustermittelpunkt verglichen und zugeordnet. Das wird wiederholt bis die Einordnung konvergiert. Im Hinblick auf die Anzahl der Hyperparameter muss beim *k-means* Clustering nur die Anzahl der Cluster vom Benutzer bestimmt werden. Das DBSCAN Verfahren benötigt neben der Mindestanzahl an Nachbarn noch den Radius der Nachbarschaft.

Wie beim nächsten Nachbar Ansatz wird auch beim Clustering der Abstand zwischen den Datenpunkten bzw. zwischen Datenpunkt und Clustermittelpunkt als Vergleichswert verwendet, wodurch eine hohe Dimensionalität des Feature-raumes negativen Einfluss auf die Ergebnisse nimmt.

Klassifikationsansatz

Bei Klassifikationsansätzen wird eine Funktion trainiert, die den Datenpunkt \mathbf{x}_v in der Anwendungsphase in *normal* oder *anormal* eingestuft. Hierbei wird zwischen *one-class* und *multi-class* Klassifikatoren unterschieden.

Multi-Class Klassifikatoren finden Anwendung, wenn konkret gelabelte Trainingsdaten zur Verfügung stehen. Ein Klassifikator wird gelernt, der bereits im Training zwischen verschiedenen Klassen unterscheidet. Methoden wie Entscheidungsbäume, SVMs oder Neuronale Netze können hierfür verwendet werden [CBK09].

Fehlen die Datenpunkte der Klasse *anormal* beim Training, findet die *one-class* Klassifikation [MT01] statt. Es wird ein Klassifikator trainiert, der in der Lage ist, die Klassen neuer Datensätze vorherzusagen, während im Training nur die Daten einer Klasse hinzugezogen werden. In der Literatur verwendete Methoden sind dabei Replikator Neuronale Netze [HHWB02, DCS14] oder *one-class* SVMs [AGA13].

Da im Rahmen der Arbeit von nicht gelabelten Anomalien ausgegangen wird, werden die *one-class* Klassifikatoren näher beschrieben.

Replikator Neuronales Netz: Bei einem Replikator Neuronales Netz handelt es sich um ein neuronales Netz. Es werden die Gewichte so trainiert, dass die Ausgabe der reproduzierten Eingabe entspricht. Bild 4.12 zeigt beispielhaft ein solches Netz mit Eingabeschicht, einer verborgenen Schicht und der Ausgabeschicht. Der Trainingsvektor $\mathbf{x}_i \in \mathbb{R}^d$ wird mit den Komponenten $\mathbf{x}_i = (x_1, x_2, \dots, x_d)$ als Netzeingabe verwendet.

Im Training werden die Gewichte so gesetzt, dass der Abstand zwischen Eingabevektor \mathbf{x}_i und Ausgabevektor $\tilde{\mathbf{x}}_i$, $\forall \mathbf{x}_i \in 1, \dots, n$ minimiert wird:

$$\min \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2. \quad (4.5)$$

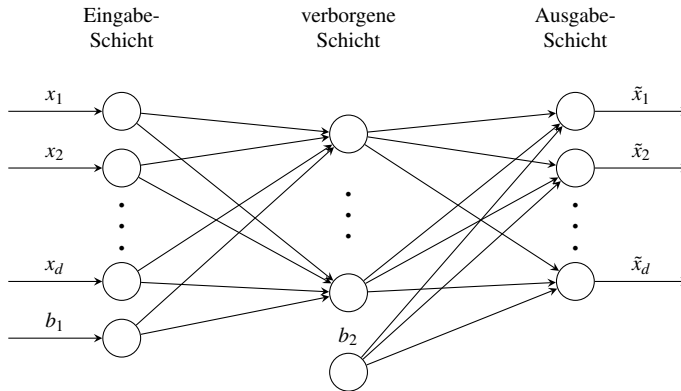


Bild 4.12: Replikator Neuronales Netz mit einer verborgenen Schicht

Formel 4.6 zeigt die Berechnung der Rekonstruktion $\tilde{\mathbf{x}}_i$ mit L als Anzahl der Schichten und $g_l(x)$ als Funktion, welche die Ausgabe der l -ten Schicht beschreibt:

$$\tilde{\mathbf{x}} = (g_L \circ g_{L-1} \circ \dots \circ g_1)(\mathbf{x}), \tag{4.6}$$

mit

$$g_k(x) = f_k(W_k \cdot x + b_k) \tag{4.7}$$

$$k = 1, \dots, L.$$

Hierbei ist W_l die Gewichtsmatrix zwischen den Neuronen der $(l - 1)$ -ten und l -ten Schicht, b_l der entsprechende Koeffizient. Die Aktivierungsfunktion f_k zeigt den Zusammenhang zwischen Neuroneneingabe und -ausgabe, welche in der darauffolgenden Schicht weiterverarbeitet wird.

Zum Erlernen der Gewichte $w = \{W_k, b_k\}_{k=1}^L$ wird Formel 4.5 als Funktion

$$Q(w) = \frac{1}{n} \sum_{i=1}^n Q_i(w) = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2 \tag{4.8}$$

mithilfe des Gradientenverfahrens minimiert. Dabei wird von zufälligen Startwerten $w^{(0)}$ ausgegangen, die in mehreren Iterationen angepasst werden. Es wird ein Schritt der Länge α in Richtung des steilsten Abstiegs adaptiert:

$$w := w - \alpha \frac{1}{n} \sum_{i=1}^n \Delta Q_i(w). \quad (4.9)$$

Anstatt alle Trainingsdaten $\{\mathbf{x}_i\}_{i=1}^n$ in einem Schritt zu verarbeiten, werden beim stochastischen Gradientenverfahren die Gewichte pro Trainingspunkt angepasst:

$$w := w - \alpha \Delta Q_i(w), \quad (4.10)$$

$i = 1, \dots, n$. Der Vorteil hierbei ist die Möglichkeit lokale Minima wieder verlassen zu können.

Beim Replikator Neuronales Netz besteht die Gefahr, die Identitätsfunktion zu lernen. Das bedeutet, der Eingabevektor \mathbf{x} wird schlicht als Ausgabe kopiert ($f(\mathbf{x}) = \mathbf{x}$) und jeder Testvektor \mathbf{x}_v wird perfekt reproduziert, unabhängig davon wie anormal er ist. Um das zu verhindern, wird ein zusätzlicher Sparsityterm ω_{sparsity} zum Optimierungsproblem 4.5 hinzugefügt [OF97]:

$$\min \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2 + \beta \cdot \omega_{\text{sparsity}} \quad (4.11)$$

mit β als Koeffizient. Die durchschnittliche Aktivität des Neurons j wird durch einen zusätzlichen Strafterm klein gehalten. Man spricht hierbei von der Regularisierung des Netzes, da der zusätzliche Sparsityterm auch zur Vermeidung von einer Überanpassung des Netzes beiträgt. Ein überangepasstes Netz kann die Trainingsdaten perfekt reproduzieren, ist jedoch für Testdaten, die nicht Teil der Trainingsdaten sind, unbrauchbar, da diese ausschließlich als anormal eingestuft werden. Andere Möglichkeiten das Netz zu regularisieren, sind das Dropoutverfahren, welches während des Trainings zufällige Neuronen ignoriert [SHK⁺14], und die L_1 - bzw. L_2 -

Regularisierung. Dabei werden die L_1 -Norm bzw. die quadrierte L_2 -Norm aller Gewichte zu einem zusätzlichen Strafterm im Optimierungsproblem [Tib94].

Das trainierte Netzwerk wird zur Berechnung des Rekonstruktionsfehlers von Datenpunkt \mathbf{x}_v verwendet, wobei es sich bei einem besser rekonstruierten Vektor weniger wahrscheinlich um eine Anomalie handelt. Das bedeutet, der Rekonstruktionsfehler entspricht dem Anomalie Score:

$$score_v = \|\mathbf{x}_v - \tilde{\mathbf{x}}_v\|_2^2. \quad (4.12)$$

Das Replikator Neuronale Netz wird in [DCS14] und [TG04] zur Erkennung von Anomalien angewandt. Dabei kommen beide Studien zu dem Ergebnis, dass nur eine verborgene Schicht ausreicht.

One-Class SVM: Die Idee der *one-class SVM* ist es, eine möglichst kleine sphärische Hülle um die als normal definierten Trainingsdaten $\{\mathbf{x}_i\}_{i=1}^n$ zu finden [TD04], auch *Support Vektor Data Description* (SVDD) genannt. Punkte außerhalb der Hülle werden als Anomalien erkannt:

$$\min F(R, \mathbf{a}) = R^2 \quad (4.13)$$

unter der Bedingung, dass

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2, \quad i = 1, \dots, n \quad (4.14)$$

mit R als Radius und \mathbf{a} als Mittelpunkt der Kugel. Durch Einführung einer Schlupfvariable ξ (engl. *slack variable*) können die Grenzen aufgeweicht werden, was dazu führt, dass Datenpunkte auch außerhalb der Hülle liegen können. Das ist vorteilhaft, wenn sich Ausreißer in den Trainingsdaten befinden:

$$\min F(R, \mathbf{a}) = R^2 + C \sum_{i=1}^n \xi_i, \quad (4.15)$$

unter der Bedingung, dass

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{a}\|^2 &\leq R^2 + \xi_i \\ \xi_i &\geq 0 \\ i &= 1, \dots, n. \end{aligned} \quad (4.16)$$

Mit dem Parameter C kann ein Trade-Off zwischen dem Radius und der Anzahl an Punkten außerhalb der Sphäre gesteuert werden. Eine Kernelfunktion transformiert die Datenpunkte in eine höhere Dimension, in der die Aufteilung in normale Daten innerhalb und anormale Daten außerhalb der Sphäre möglich ist. Die Lösung des Optimierungsproblems mithilfe der Lagrange Multiplikatoren führt zur Multiplikation zweier Trainingsvektoren \mathbf{x}_i und \mathbf{x}_j , $i, j = 1, \dots, n$, $i \neq j$. Mit dem sogenannten Kerneltrick werden die Vektoren direkt als Eingabe in die Kernelfunktion verwendet, ohne vorher in einen höherdimensionalen Raum abgebildet zu werden [HSS08, ABR64]. Der Testvektor \mathbf{x}_v gilt dann als Anomalie, wenn er außerhalb der Hülle liegt. Die Entfernung zum Mittelpunkt der Sphäre kann als Score verwendet werden.

Bild 4.13 zeigt beispielhaft die Kontur einer gelernten SVM mit der gaußschen radialen Basisfunktion (engl. *radial basis function*, RBF) als Kernel (vgl. Formel 4.17). Die Konturen verdeutlichen, dass der Anomalie Score steigt, je weiter die Datenpunkte vom Mittelpunkt der Sphäre entfernt sind.

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (4.17)$$

Eine andere Herangehensweise, mit demselben Ergebnis, ist das Lernen einer Ebene, welche die normalen Trainingsdaten bestmöglich vom Ursprung trennt [SWS⁺00]. Datenpunkte nahe dem Ursprung kennzeichnen Anomalien. Der Ansatz bietet Erweiterungen, die robuster sind, wenn sich Anomalien in den Trainingsdaten befinden. Bei der sogenannten robusten *one-class* SVM ist die Schlupfvariable ξ_i proportional zum Abstand zwischen dem

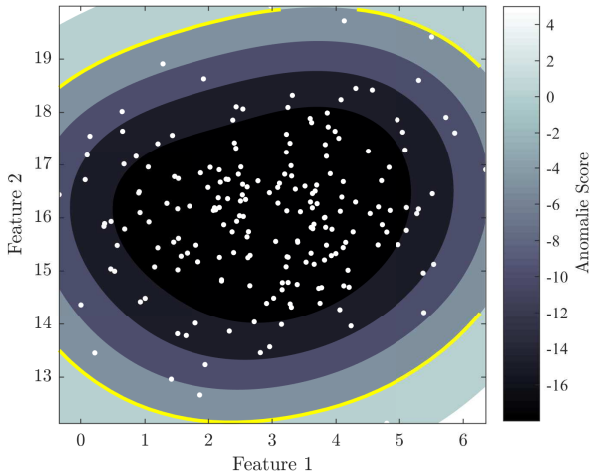


Bild 4.13: One-Class Support Vektor Machine mit RBF Kernel, gelb: gelernte Sphäre um die als normal angenommenen Datenpunkte

Trainingsvektor \mathbf{x}_i und dem Mittelpunkt aller Trainingsvektoren im Kernelraum, wodurch die Schlupfvariablen für weiter entfernte Punkte größer sind. Bei der *eta one-class SVM* wird ein zusätzlicher Parameter $v_i \in [0, 1]$ zum Optimierungsproblem hinzugefügt. Er definiert die Gewichtung von Trainingspunkt \mathbf{x}_i bei der Optimierung. Sprich Ausreißer mit $v_i = 0$ werden ignoriert. Beide Erweiterungen schneiden beim Vergleich mit anderen Methoden zur Anomalieerkennung gut ab [AGA13].

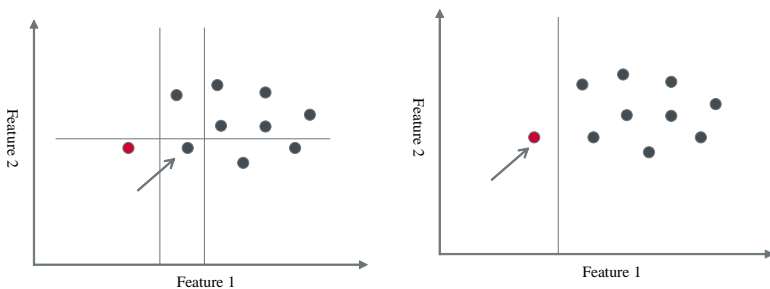
Bewertung: Bei einem Klassifikationsmodell handelt es sich um eine Funktion, die das normale Verhalten beschreibt, und daher ins Fahrzeug übertragen werden kann.

Die Robustheit gegen Anomalien in den Trainingsdaten wird insbesondere durch die *one-class SVM* erfüllt, wohingegen das Replikator Neuronale Netz von ausschließlich normalen Daten ausgeht. Der Vorteil am Replika-

tor Neuronales Netz ist die Unterstützung bei der Interpretation erkannter Anomalien. Mithilfe des Rekonstruktionsfehlers pro Signal kann bewertet werden, welche Signale ausschlaggebend am hohen Anomalie Score sind. Weniger einfach ist die Interpretation bei einer SVM, da die Eingabevektoren durch die Kernelfunktion in eine höhere Dimension gebracht werden und daher ihre ursprüngliche Gestalt schwierig nachvollziehbar ist. Beide Ansätze liefern einen Anomalie Score als Ausgabe. Beim Replikator Neuronales Netz entspricht dieser dem Rekonstruktionsfehler von Testvektor \mathbf{x}_v , bei der *one-class* SVM dem Abstand vom anomalen Datenpunkt \mathbf{x}_v zum Mittelpunkt der Sphäre, welche die normalen Daten umhüllt. Für Replikator Neuronale Netze hängt die Trainingskomplexität von den Eigenschaften des Netzes (z.B. Anzahl verborgener Schichten, Aktivierungsfunktion, Anzahl Neuronen) ab. Der Nachteil der *one-class* SVM ist die hohe Komplexität, die zwischen $O(n^2)$ und $O(n^3)$ liegt [BL07]. Für beide Klassifikationsansätze ist die Anzahl zu definierender Hyperparameter hoch. Für das Replikator Neuronale Netz müssen neben der Netztopologie auch die Parameter zur Regularisierung und für das Gradientenverfahren (z.B. Schrittweite α) definiert werden. Für die Support Vektor Maschine gilt es die Kernelfunktion, dessen Parameter und den Trade-Off zwischen Radius und Anzahl außenliegender Punkte (C) zu definieren. Bei den erweiterten SVMs, der *eta one-class* SVM und der robusten SVM, fallen zusätzliche Parameter an. Beide Verfahren können jedoch mit höher dimensionalen Featureräumen umgehen. Replikator Neuronale Netze haben ihren Ursprung in der Datenkompression und dem Erlernen einer niedriger dimensionalen Repräsentation des originalen Feature Raums. In diesem Zusammenhang spricht man dann von *Autoencoder* [HS06, Ng11, DCS14]. Für SVMs ist der Generalisierungsfehler unabhängig von der Anzahl an Features [PP14].

Isolation Forest

Der Isolation Forests ist eine neuartige Methode, die sich nicht in die vorgestellten Kategorien einordnen lässt. Anders als die oben beschriebenen Vorgehensweisen, wird hier kein Anomaliedetektor trainiert, der dahingehend optimiert wird, das normale Verhalten abzubilden, sondern vielmehr Anomalien zu isolieren [LTZ08, LTZ12]. Es werden die Trainingsdaten in Subgruppen unterteilt. Pro Subgruppe \mathbf{X}_{sub} wird ein sogenannter *Isolation Tree* gelernt. Dazu wird zufällig eines der d Signale selektiert. Für dieses Signal j wird ein zufälliger Trennwert p_j definiert, der innerhalb der möglichen Grenzwerte liegt. Durch die Unterscheidung der Datenpunkte mit $x_j < p_j$ und $x_j \geq p_j, \forall \mathbf{x} \in \mathbf{X}_{sub}$ ergeben sich erneut zwei Subgruppen, für die das beschriebene Verfahren rekursiv fortgeführt wird. Es entsteht ein Baum, der die Untergruppe \mathbf{X}_{sub} beschreibt. Unter der Annahme, dass sich Anomalien leichter vom Rest der Daten isolieren lassen (vergleiche Bild 4.14), werden die trainierten Bäume zur Berechnung des Scores von Datenpunkt \mathbf{x}_v verwendet. Dazu wird die durchschnittliche Pfadlänge in den Bäumen benutzt, welche beschreibt in welcher Ebene des Baumes sich \mathbf{x}_v befindet. Je weiter oben, desto wahrscheinlicher handelt es sich um eine Anomalie (vgl. Bild 4.15).



(a) Normaler Datenpunkt nach drei Schritten isoliert (b) Anormaler Datenpunkt nach einem Schritt isoliert

Bild 4.14: Isolation Forest Ansatz

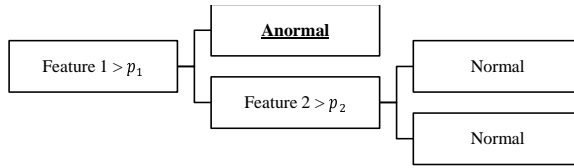


Bild 4.15: Bestimmung der Pfadlänge je Bäume, hier: Pfadlänge für Anomalie = 1

Bild 4.16 demonstriert anhand eines Beispiels, dass der Baum, mit dem Trennwert an oberster Stelle, der zwischen den Klassen *normal* und *anormal* unterscheidet, wahrscheinlicher ist. Das ist der Grund, warum die erwartete Pfadlänge eines anomalen Datenpunktes kürzer als die eines normalen ist. Für das Beispiel gilt [LTZ08, LTZ12]:

$$P(T_1) < P(T_2) \implies E(h_2) < E(h_0) < E(h_1), \quad (4.18)$$

mit $P(T_i)$ als die Wahrscheinlichkeit, den Baum T_i zu generieren und $E(h_i)$ als die erwartete Pfadlänge (h_i) von Datenpunkt x_i , $i \in \{0, 1, 2\}$:

$$E(h_i) = P(h(x_i) = 1) \cdot 1 + P(h(x_i) = 2) \cdot 2. \quad (4.19)$$

Die Parameter, die es zu definieren gilt, sind die Größe der Subgruppen (ψ) und die Anzahl der Bäume (t). Deren maximale Höhe wird anhand der Größe ψ gesetzt als $l = \text{ceil}(\log_2 \psi)^3$ [LTZ08, LTZ12].

Bewertung: Der Isolation Forest besteht aus einer Menge von Entscheidungsbäumen, welche die Trainingsdaten beschreiben. Da es sich um eine reduzierte Darstellung der Daten handelt, ist die Übertragbarkeit ins Fahrzeug gegeben. Allerdings führt eine steigende Anzahl an trainierten Bäumen (t) zu erhöhter Anwendungszeit, da der Testdatenpunkt \mathbf{x} , mit allen Bäumen abgeglichen werden muss. Die Methode ist robust gegen Anomalien in den

³ *ceil*: Auf nächsthöheren Integerwert runden.

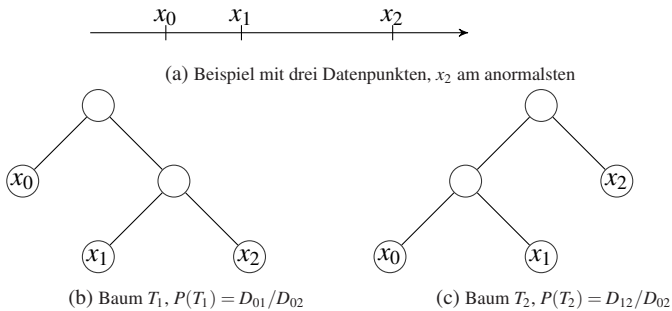


Bild 4.16: Beispiel zur Veranschaulichung des Isolation Forest, D_{ij} ist der Abstand von Punkt x_i zum Punkt x_j mit $D_{01} < D_{12}$, nach [LTZ12]

Trainingsdaten, da die Bäume sowohl mit als auch ohne Anomalien trainiert werden können. Referenz [LTZ08] zeigt, dass das Training mit ausschließlich normalen Datensätzen nur zu einer geringen Verschlechterung führt, was im Gegenzug bedeutet, dass Anomalien in den Trainingsdaten sogar gewünscht sind. Die große Anzahl unterschiedlicher, zufällig generierter Bäume erschwert die Interpretation. Durch die Zufälligkeit, kann auch ein Feature, das in mehreren Bäumen weit oben steht, nicht als ausschlaggebend interpretiert werden. Als Anomalie Score wird die durchschnittliche Baumtiefe des Datenpunktes, normiert auf die durchschnittliche Gesamtpfadlänge, verwendet. Anders als die anderen Ansätze verwendet die Isolation Forest Methode kein Dichte- oder Abstandsmaß zwischen den Datenpunkten, was die Trainingskomplexität mindert. Außerdem ist dadurch das Verfahren besonders gut geeignet für Datensätze mit vielen unterschiedlichen Features, leidet allerdings an zu vielen irrelevanten Eingangsgrößen. Der Isolation Forest gehört mit einer linearen Trainingskomplexität von $O(t\psi \log \psi)$ zu den schnellsten der vorgestellten Methoden, die zudem unabhängig von der Anzahl der Trainingsamples n ist. Anders als bei den Klassifikationsansatz sind die zu optimierenden Parameter leichter nachzuvollziehen, im

Vergleich zum nächsten Nachbar Ansatz oder dem Clustering müssen nur wenig mehr Hyperparameter gesetzt werden.

4.1.2 Anomalieerkennung in multivariaten Zeitreihen

Die im Kapitel 4.1.1 vorgestellten Methoden gehen davon aus, dass die Datenvektoren unabhängig voneinander sind. Da die Kombination der Signale jedoch als multivariate Zeitreihe behandelt werden kann, wird anschließend dargestellt, wie sich die Methoden auf zeitabhängige Daten anwenden lassen. Hierbei wird von z Zeitreihen unterschiedlicher Länge ausgegangen. Eine Zeitreihe beschreibt eine zusammengehörige Abfolge von Signalvektoren, wie es zum Beispiel eine Fahrt ist. Angewandt auf Tabelle 2.1 kann anhand der Zeitpunkte pro Fahrt eine Zeitreihe konstruiert werden. Zeitreihe Z setzt sich aus $m + 1$ Zeitpunkten zusammen:

$$Z = \{\mathbf{x}^{(t)}\}_{t=0}^m \quad (4.20)$$

mit $\mathbf{x}^{(t)} \in \mathbb{R}^d$.

Durch ein Zeitfenster der Länge p , das um h Zeiteinheiten verschoben wird, ergibt sich für Zeitreihe Z ein Feature Raum F . Für $h = 1$ gilt:

$$F = \{\mathbf{z}^{(t)}\}_{t=p}^m, \quad (4.21)$$

mit

$$\begin{aligned} \mathbf{z}^{(t)} &= (\mathbf{x}^{(t-p)}, \mathbf{x}^{(t-p+1)}, \dots, \mathbf{x}^{(t-1)}) \\ &= (x_1^{(t-p)}, \dots, x_d^{(t-p)}, \dots, x_d^{(t-1)}). \end{aligned} \quad (4.22)$$

Die Vergrößerung des Feature Raums von d zu $p \cdot d$ Features führt zum Problem, dass die Annahmen über die Datenstruktur, Abstandsmessungen und Algorithmen in hoch dimensionalen Räumen stark generalisieren [KM10]. Zudem müssen sowohl die Fensterlänge p als auch die Verschiebung h als Parameter definiert und an den Anwendungsfall angepasst werden.

Zur Anomalieerkennung in Zeitreihen kann zwischen drei Ansätzen unterschieden werden:

Feature Extraktion: Eine Möglichkeit statische Methoden auf Zeitreihen anzuwenden, ist das Extrahieren höherwertiger Features. Darunter versteht man Merkmale, welche die zeitliche Komponente beschreiben. Häufig verwendete Größen sind Mittelwert und Standardabweichung, Fourier- oder Wavelet-Transformationen, Autokorrelation oder Anzahl der Nulldurchgänge [FJ14, NAM01, Lia05, SJDLA11]. Die höherwertigen Features können dann als Eingabe für die statischen Methoden aus Kapitel 4.1.1 verwendet werden. Das Problem des vergrößerten Feature Raums kann durch die extrahierten Features reduziert werden.

Markov Kette: Markov Modelle werden vor allem zur Anomalieerkennung in univariaten Zeitreihen verwendet ($d = 1$). Es handelt sich um ein stochastisches Modell, welches die Zeitreihe als eine Abfolge von Zuständen betrachtet, deren Übergangswahrscheinlichkeiten gelernt werden. Eine Sequenz mit geringer Wahrscheinlichkeit gilt dann in der Anwendungsphase als anormal [YD03, Cha09, Ye00, CGLT15, NMJ15]. Schwierig ist die Verwendung von höherdimensionalen Räumen, da die Markov Kette für großes d beliebig komplex wird. Die Reduzierung zu einer univariaten Zeitreihe, indem der Datenvektor $\mathbf{x} \in \mathbb{R}^d$ auf Zustand $x_z \in \mathbb{R}$ abgebildet wird, ist allerdings ungenau für große d .

Zeitreihenvorhersage: Das in Kapitel 4.1.1 vorgestellte Replikator Neuronale Netz verwendet als einzige der Methoden den Unterschied zwischen der Vorhersage und tatsächlichem Wert als Anomalie Score:

$$\|\mathbf{x} - f(\mathbf{x})\|. \quad (4.23)$$

Diese Idee lässt sich auf Zeitreihen übertragen. Es wird eine Funktion gelernt, die anhand der Vergangenheit $\mathbf{z}^{(t)}$ die Gegenwart $\mathbf{x}^{(t)}$ vorhersagt. Der

Abstand zwischen Vorhersage und tatsächlicher Gegenwart wird als Anomalie Score verwendet:

$$\|\mathbf{x}^{(t)} - f_i(\mathbf{z}^{(t)})\| \quad (4.24)$$

Zur Zeitreihenvorhersage gibt es unterschiedliche Möglichkeiten, die Funktion f_i zu modellieren. Ähnlich wie bei der Aufteilung der statischen Methoden, dargestellt in Bild 4.4, wird hierbei zwischen stochastischen Verfahren, SVMs für Regression und neuronalen Netzen unterschieden [AA13]. Zu den stochastischen Methoden gehören unter anderem autoregressive Modelle (AR), welche die Gegenwart als Linearkombination der Vergangenheit abbilden:

$$\hat{x}^{(t)} = c + \sum_{i=1}^p w_i x^{(t-i)} + \varepsilon_t, \quad (4.25)$$

mit c als Konstante und ε_t als weißes Rauschen [Box70].

Das Replikator Neuronale Netz kann für die Zeitreihenvorhersage dahingehend angepasst werden, dass die Eingabeneuronen die Vergangenheit $\mathbf{z}^{(t)}$ und das Ausgabeneuron die Gegenwart $\mathbf{x}^{(t)}$ beschreiben (vgl. Bild 4.17). Hier wird davon ausgegangen, dass die Neuronenausgaben voll verbunden, aber nur in Verarbeitungsrichtung geleitet werden. Daher spricht man von einem sogenannten *feedforward* Netzwerk. Es existieren erweiterte Netztopologien, die zum Beispiel auch saisonale Abhängigkeiten abbilden können [AA13, ZPH98].

Weitere neuronale Netze, die für Zeitreihen verwendet werden, sind rekurrente Netze und convolutional Netze. Rekurrente Netze werden vor allem im Bereich der automatischen Spracherkennung angewandt. Die meisten in aktuellen Smartphones installierten Spracherkennungen basieren auf *Long-Short-Term-Memory* Netzwerken, eine spezielle Art rekurrenter Netze [HS97]. Bei der simpelsten Form einer rekurrenten Schicht sind alle verbundenen Neuronen miteinander verbunden, wodurch ein interner Speicher

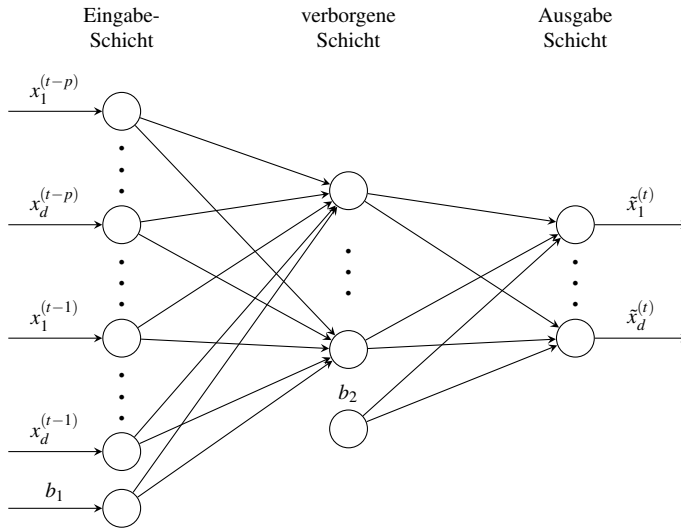


Bild 4.17: Zeitreihenvorhersage mit neuronalem Netz

modelliert wird [KG17]. Bild 4.18 verdeutlicht den Ansatz. Der Vorteil liegt darin, dass Muster in Sequenzen erkannt werden können [GFGS06].

Bei einem convolutional Netz werden die Gewichte zur Faltung der Zeitreihe gelernt [VF11]. Dazu müssen die Anzahl der Faltungen und deren Filterlängen definiert werden. Durch Einfügen einer weiteren Schicht, in der z.B. pro Faltung der maximale Wert verwendet wird (engl. *MaxPooling* [SMB10]), kann die Vorhersage von Zeitpunkt $\mathbf{x}^{(t)}$ erfolgen. Bild 4.19 zeigt den Ansatz beispielhaft anhand einer univariaten Zeitreihe ($d = 1$). In dem dargestellten Beispiel werden zwei Faltungen mit jeweils zwei Filtern gelernt. Der Vorteil liegt darin, dass weniger Gewichte zu optimieren sind, da pro Faltung die Filterlänge die Anzahl zu lernender Gewichte definiert. Convolutional Netzwerke finden hauptsächlich Anwendung in der Bildverarbeitung [KSH12, CMS12].

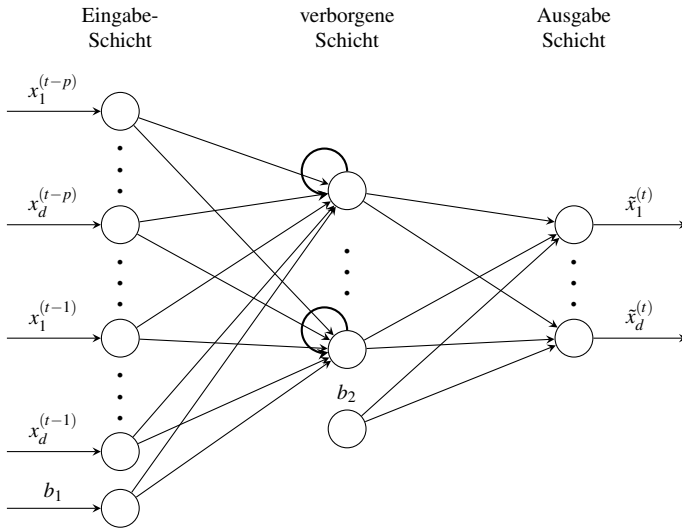


Bild 4.18: Zeitreihenvorhersage mit rekurrentem Netz

In Summe kann festgehalten werden, dass die Kommunikationsdaten mittels statischer als auch dynamischer Methoden verarbeitet werden können, um darin Anomalien zu finden. Zur Gewährleistung der Übertragbarkeit ins Fahrzeug muss auf die modellbasierten Verfahren (vgl. Bild 3.5) eingeschränkt werden, die einen *one-class* Klassifikator als Ausgabe liefern. Tabelle 4.1 verdeutlicht, dass insbesondere das Replikator Neuronale Netz, der Isolation Forest und das Clustering die Anforderungen erfüllen. Deren Vorhersagegenauigkeiten werden in Kapitel 5 anhand beispielhafter Anomalien geprüft. Die Vorteile der *one-class* SVM werden bei großen Datenmengen aufgrund der Rechenkomplexität relativiert.

Für Zeitreihen ist das Konzept der dynamischen Datensammlung durch das Extrahieren höherwertiger Features und die darauffolgende Anwendung statischer Methoden oder die Verwendung neuronaler Netze zur Zeitreihenvorhersage möglich.

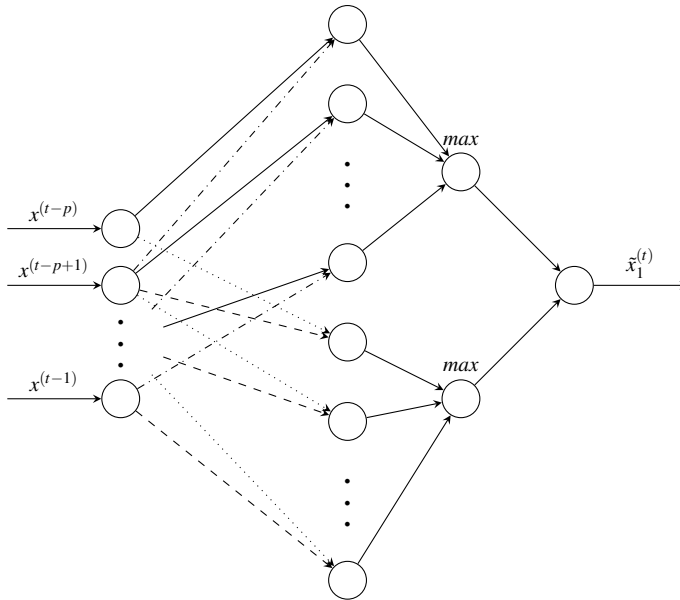


Bild 4.19: Zeitreihenvorhersage mit convolutional Netz, hier mit zwei Filtern

4.1.3 Anwendungsüberblick

Anomalien werden in den unterschiedlichsten Bereichen erkannt. In [PY13] wird ein Überblick über die verschiedenen Anwendungsgebiete gegeben. Neben der Erkennung von Netzwerk- oder Systemattaken (*Intrusion Detection*) spielt das Erkennen von Anomalien auch in der Medizin eine wichtige Rolle. Beispielsweise helfen sie bei der frühzeitigen Identifikation von Krankheiten und ungewöhnlichen Symptomen. Aber auch zur Erkennung von defekten bzw. bald defekten Sensoren in Sensornetzwerken findet die Methode Verwendung [PY13, PCCT14, GU16].

Der in der Literatur für Anwendungen am häufigsten verwendete Algorithmus ist die *one-class SVM* [DOSB10, MMHH⁺11, MYC⁺01, MSPMG15, CHN⁺13, The13]. Neben dem Vergleich mit anderen Verfahren wie dem

Replikator Neuronales Netz [MYC⁺01], wird auch zwischen der *one-* und *two-class* Klassifikation unterschieden:

Die Prognose von schwarzem Hautkrebs mithilfe einer *one-class* Klassifikation wird in [DOSB10] bearbeitet. Dazu wird eine *one-class* SVM gelernt, die Hautkrebspatienten ohne Metastasen als normal betrachtet. Abweichungen davon sollen Hinweise auf ein höheres Metastasenrisiko geben, mit dem Ziel, dieses so früh wie möglich zu erkennen. Die *one-class* SVM ordnet 75% der Patienten mit Metastasen und 59% der ohne Metastasen in die richtige Klasse ein. Zudem wird eine *two-class* SVM verwendet und gezeigt, dass die Genauigkeit mit sinkender Anzahl der Minderheitsklasse - Patient mit Metastase - abnimmt.

One-class und *two-class* SVMs werden auch in [MMHH⁺11] verglichen. Dazu wird die Hypothese, dass die Gehirnaktivität als Antwort auf traurige emotionale Stimuli bei Patienten mit Depressionen nicht normal ist, aufgestellt. Es werden zwei *one-class* SVMs trainiert, einmal mit gesunden Patienten als *normal* und einmal mit kranken Patienten als *normal*. Dabei wird eine Korrelation zwischen dem Anomalie Score und dem *Hamilton Rating Scale for Depression*, einem Score der als Stärke der Depression gelesen werden kann, dargelegt. Aufgrund der abnehmenden Genauigkeit beim Training mit den kranken Patienten als *normal* kann gezeigt werden, dass deren Heterogenität dazu führt, dass die Sphäre, die zwischen Normalität und Anomalie trennt, groß ist und dadurch auch gesunde Patienten als *normal*, sprich depressiv, eingestuft werden. Daher ist eine binäre Klassifikation, in der alle kranken Patienten gleich gelabelt werden, ungerecht.

Eine andere Anwendung, die eine *one-class* SVM verwendet, ist die Klassifikation von Dokumenten in [MYC⁺01]. Dazu wird der *Reuters* Datensatz, eine gelabelte Sammlung von Artikeln, auf Anomalien untersucht. Die *one-class* Klassifikation wird pro Artikelklasse durchgeführt und getestet. Es werden eine *one-class* SVM, eine *outlier* SVM⁴, ein Replikator Neuronales

⁴ Hier werden zunächst Anomalien basierend auf dem Abstand zum Ursprung gelabelt und danach eine *two-class* SVM trainiert.

Netz, und die Bayes Naive Methode⁵ verglichen. Als Ergebnis resultieren das Replikator Neuronale Netz und die *one-class* SVM als die Verfahren mit höchster Genauigkeit. Allerdings handelt es sich beim Replikator Neuronalen Netz um die robustere Methode, da die *one-class* SVM sehr sensitiv auf die Wahl der Kernelfunktion reagiert.

Referenz [The13] betrachtet die Kommunikationsdaten im Fahrzeug als multivariate Zeitserie, für die das normale Verhalten mithilfe einer *one-class* SVM gelernt wird. Dabei wird die Methode der *one-class* SVM nach [TD04] (SVDD vgl. Kapitel 4.1.1) favorisiert, da ein Trade-Off zwischen dem Volumen der Sphäre, die die normalen Daten einschließt und der Anzahl fälschlicherweise als Ausreißer markierten Trainingsdaten, gelernt wird. Es sind keine a-priori Informationen über erwartete Anomalien nötig [The13]. Beim Vergleich zwischen dem k -nächsten Nachbar Ansatz, LOF und SVDD, angewandt auf reale Datensätze, erzielt die SVDD die höchste Genauigkeit. Da es sich um ein technisches System handelt, kann von Rauschen ausgegangen werden, weswegen zur Bewertung pro Subsequenz die durchschnittliche Distanz der Datenvektoren zum Zentrum der durch die SVM definierte Sphäre berechnet wird [The14]. Zur Evaluierung werden Aufnahmen von Fahrzeugen im Ruhezustand analysiert und Anomalien erzeugt indem z.B. das Zündkabel einige Male entfernt wird oder mithilfe eines Potentiometers fehlerhafte Sensoren, ein fehlender Kontakt oder ein Kurzschluss simuliert werden. Die mit den Daten nicht manipulierter Fahrzeuge trainierte SVDD erkennt 92,9% der Anomalien richtig. In den 3958 Sekunden der Testdaten ohne Anomalien werden sieben Subse-

⁵ Die Wahrscheinlichkeit, dass es sich bei Dokument d um einen Datenpunkt der normalen Klasse E handelt ergibt sich aus dem Produkt der Wahrscheinlichkeiten aller k Schlüsselwörter $w_i, i = 1 \dots k$ im Dokument. Formel 4.26 zeigt die Berechnung mit n_{w_i} als Anzahl der Vorkommnisse von Schlüsselwort w_i im Dokument, mit m als Anzahl unterschiedlicher Schlüsselwörter und n als Anzahl aller Wörter im Text:

$$p(d|E) = \prod_{i=1}^k \frac{n_{w_i} + 1}{n + m} \quad (4.26)$$

quenzen fälschlicherweise als anomal eingestuft [The13]. Referenz [TD13] kommt außerdem zu dem Ergebnis, dass für das Training die Daten von mehreren Fahrer herangezogen werden müssen, um eine Überanpassung an einen konkreten Fahrer zu vermeiden.

Eine Anwendung aus der Industrie kann in [MSPMG15] gelesen werden. Darin wird versucht, Schäden in Turbomaschinen auf Ölplattformen so früh wie möglich zu detektieren. Es werden Sensordaten wie Motor- und Turbinendaten, als auch Informationen über den Ölfluss verwendet und auch hier führt der Mangel an gelabelten Anomalien zum *one-class* Klassifikationsansatz. Es werden die Daten des ersten Halbjahres 2012 von 64 Sensoren einer Turbomaschine in einer Frequenz von fünf Minuten abgelesen. Im Preprocessing wird eine Clustermethode zur Segmentierung der Zeitserien vorgestellt. Zur Vorhersage von Anomalien werden folgende Ansätze verglichen: Zum einen werden Konfidenzintervalle verwendet. Sie schließen die normalen Werte mit einer gewissen Signifikanz ein. Für jede Zeitserie wird das Intervall berechnet und Werte außerhalb werden als Ausreißer erkannt. Außerdem wird eine klassische *one-class* SVM und die Kombination aus Segmentierung und *one-class* SVM trainiert. Als Ergebnis lässt sich zusammenfassen, dass der Letzte der genannten Ansätze den anderen vorzuziehen ist.

Auch Schlaglöcher werden als Anomalien entdeckt. Es wird die Beschleunigung in y-Richtung (vertikale Richtung) mithilfe der Wavelet Transformation zerlegt und Features extrahiert. Die trainierte *one-class* SVM mit der gaußschen radialen Basisfunktion als Kernel (vgl. Formel 4.17) ist in der Lage, alle 21 Testschlaglöcher als Anomalie zu erkennen und nur sechs der 530 normalen Testabschnitte werden fälschlicherweise als Schlagloch detektiert [CHN⁺13].

Der **Isolation Forest Ansatz** wird in [SBM17] erfolgreich zur Identifikation von Anomalien im Prozessablauf des Plasmaätzens angewandt. Es geht um die Klassifikation von Fehlerursachen zur Maximierung der Geräteauslastung in der aktuellen Halbleiterfertigung [HLCM12]. Verwendet werden

hierfür die Daten der optische Emissionsspektrometrie, die während eines Wafer-Prozesses aufgezeichnet werden. Der Isolation Forest Ansatz erkennt um die 92% der Anomalien, während 83% der erkannten Anomalien wirklich anormal sind. Referenz [Nag17] betrachtet den Verbrauch in Gewerbegebäuden als komplexes System, wo mithilfe des Isolation Forests der Energieverbrauch optimiert und Energiedesign und -betrieb ausgerichtet werden kann. Allerdings wird hier gezeigt, dass der Ansatz Potentiale aufweist, aber noch verbessert werden muss.

Replikator Neuronale Netzwerke oder auch *Autoencoder* werden in [SHY⁺18] zur Erkennung von ungewöhnlichen Mustern auf CT-Aufnahmen des Gehirns verwendet. Der trainierte 3D-convolutional Autoencoder ist in der Lage, 68% der Anomalien zu erkennen, während 88% der normalen Bilder als solche erkannt werden. In [OY18] wird der Maschinensound eines *Surface-Mount Device* (SMD) aufgenommen, um anhand von Auffälligkeiten mögliche Ausfälle vorherzusagen. Auch hier wird der Ansatz des Replikator Neuronalen Netzes erfolgreich eingesetzt.

Anwendungen, für die ein **Markov Model** (vgl. Kapitel 4.1.2) verwendet wird, sind die Anomalieerkennung in einem Computer und Netzwerksystem [Ye00] und in Kommunikationsdaten im Fahrzeug [NMJ15]. In in [Ye00] beschränkt sich die Analyse auf eine Zeitreihe der Dimension $d = 1$. Die Signale unter Beobachtung sind Audit Events eines Unix-basierten Host-Rechners. [NMJ15] betrachtet die Daten als Zeitreihen und einzelne Signale, wie beispielsweise die Motordrehzahl, werden mithilfe von Markov Ketten analysiert. Der Hintergrund dabei ist, das Modell als plug'n play Software in sowohl neue als auch alte Fahrzeuge zu installieren, um die Fahrzeuge gegen Angriffe abzusichern. Unerwartete Zunahmen und Abnahmen von Geschwindigkeit und Motordrehzahl können, sowohl separat als auch in Kombination, erkannt werden. Normale Signalverläufe werden als normal identifiziert.

Referenz [CTM⁺15] verwendet einen **nächsten Nachbar Ansatz**, um anhand von Städten, die anormal auffallen, Rückschlüsse ziehen zu können,

welche Krankenhäuser mehr Operationen als erwartet verrechnen und daher möglicherweise betrügerische Absichten verfolgen. Basierend auf einem Datensatz des brasilianischen Gesundheitswesens werden kritische Krankenhäuser entdeckt.

Der Überblick verdeutlicht, dass die Idee der Anomalieerkennung mittels maschinellen Lernens in den unterschiedlichsten Bereichen Einsatz findet. In der Literatur überwiegend vertreten ist der Ansatz der *one-class SVM*. In keiner der aufgezeigten Anwendungen gelingt eine eindeutige Trennung der Zustände *normal* und *anormal*; unter den erkannten Anomalien befinden sich auch normale Datensätze. Zusammengefasst sind die Ergebnisse abhängig vom Anwendungsfall und dem ausgewählten Verfahren, weswegen im folgenden Kapitel 5 die geeigneten Methoden prototypisch verwendet werden. Geeignete Methoden sind die, welche die Anforderungen aus Abschnitt 3.3 erfüllen.

4.2 Postprocessing

Im Folgenden wird auf die Schritte des Postprocessings *Entscheidung* und *Interpretation* eingegangen (vgl. Bild 3.1). Es werden mögliche Herangehensweisen vorgestellt. Außerdem zeigt Abschnitt 4.2.2 welche Kennzahlen zur Bewertung der Anomalieerkennung geeignet sind.

4.2.1 Entscheidung anhand des Anomalie Scores

Wie in Kapitel 4.1.1 beschrieben, liefern alle vorgestellten Verfahren einen Anomalie Score ($score_v$) als Ausgabe für Datenpunkt \mathbf{x}_v . Ein gängiges Verfahren ist es, die Datenpunkte nach dem Score zu sortieren und die mit den Höchsten näher als Anomalien zu betrachten [GU16, KKSZ11]. Die Scores können auch für eine Klassifikation in *normal* oder *anormal* verwendet werden. Dazu wird ein Grenzwert τ definiert, sodass für

$$(score_v > \tau), \quad (4.27)$$

der Punkt x_v als Anomalie gilt. Die Festlegung der Grenze erfolgt im Post-processing. Zusätzliche Informationen, wie zum Beispiel die Tatsache, welche Signale verarbeitet werden und auch wie brisant diese sind, können den Wert der Grenze beeinflussen. Der Grenzwert kann auch an die Anforderungen angepasst werden, wie das Beispiel, dargestellt in Bild 4.20, verdeutlicht: Ist es das Ziel, alle Anomalien zu erkennen und fälschlicherweise als anormal erkannte Datenpunkte stellen kein Problem dar, muss $\tau < 1,5$ als Grenzwert verwendet werden, da dann der Score aller Anomalien a_1 , a_2 und a_3 den Grenzwert übersteigt. Zählt hingegen die Vorhersagegenauigkeit, wird für $2 \leq \tau < 2,5$ nur der anormale Datenpunkt a_1 falsch klassifiziert, für alle anderen Datenpunkte wird die richtige Klasse vorhergesagt. Je nach Gewichtung der Prioritäten ergeben sich andere Grenzen. In dem Beispiel wird von gelabelten Datenpunkten ausgegangen, wodurch eine gezielte Optimierung von τ möglich ist.

Ist diese Information nicht gegeben und eine anwendungsorientierte Optimierung der Grenze nicht möglich, muss die Festlegung rein datenbasiert erfolgen. So können beispielsweise die Scores der Trainingsdaten $S = \{score_i\}_{i=1}^n$ als Referenz verwendet. Darauf aufbauend kann der Grenzwert unterschiedlich bestimmt werden:

1. Eine Möglichkeit ist die Verwendung des x -Quantils (Q_x) der Trainingsscores als Grenzwert. Dadurch kann die in Kapitel 3.3 vorgestellte Robustheit gegen Anomalien in den Trainingsdaten nachgelagert erreicht werden, da die x Datenpunkte mit höchstem Score ausgeschlossen werden. Wählt man das 100%-Quantil, wird von aus-

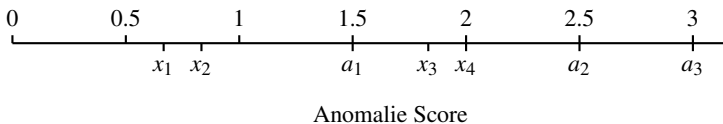


Bild 4.20: Beispielhafte Anomalie Scores für normale Datenpunkte (x) und anormale Datenpunkte (a)

schließlich normalen Trainingsdaten ausgegangen. Da Kommunikationsdaten Sensorgrößen sind, muss von Rauschen und Ungenauigkeiten ausgegangen werden [The14]. Die Annahme, dass alle Trainingsdaten normal sind, ist daher sehr optimistisch. Vorteil an dieser Art der Grenzfestlegung ist, dass reguliert werden kann, wie außergewöhnlich ein Zeitpunkt sein muss, um als anormal zu gelten:

$$\tau_{q_x} = Q_x(S) \quad (4.28)$$

2. Statt der a-priori Schätzung, wie viel Prozent der Trainingsdaten sich anormal verhalten, kann der Grenzwert ausschließlich anhand der Daten bestimmt werden. Es wird der Interquartilsabstand (*Interquartile range*, IQR) verwendet, um die Ausreißer der Trainingscores zu ignorieren [Tuk77]:

$$\tau_{IQR} = Q_{0,75}(S) + 1,5 \cdot IQR \quad (4.29)$$

3. Der Test nach Grubbs ist ein statistischer Test zur Identifikation von Ausreißern. Es wird von normalverteilten Trainingscores S ausgegangen und die Teststatistik

$$G = \frac{\max |S_i - \bar{S}|}{V_n^*(S)}, i = 1, \dots, n \quad (4.30)$$

berechnet, mit \bar{S} als Mittelwert und $V_n^*(S)$ als korrigierte Stichprobenvarianz der Trainingscores. Die Nullhypothese, dass sich in den Trainingscores keine nach oben gerichtete Ausreißer befinden, wird mit einem Signifikanzniveau von α abgelehnt, falls

$$G > \frac{n-1}{\sqrt{n}} \sqrt{\frac{t_{\frac{\alpha}{n}, n-2}^2}{n-2 + t_{\frac{\alpha}{n}, n-2}^2}}. \quad (4.31)$$

Hier ist $t_{\beta,m}$ das β -Quantil der t -Verteilung mit m Freiheitsgraden zum Niveau β .

Der erkannte Datenpunkt wird entfernt und der Test so lange wiederholt, bis alle verbleibenden Trainingscores als normal gelten. Deren maximaler Score wird dann als Grenzwert τ_{grubbs} verwendet [Gru69].

Die Grenzfestlegung im Postprocessing erzielt die in Kapitel 3.3 vorgestellte Robustheit. Durch den Grenzwert, definiert anhand der Trainingscores, werden außergewöhnliche Trainingsdaten vom Referenzmodell als Beschreibung der Normalität, ausgeschlossen. Die Definition, welche Trainingsdaten normal und somit Teil der Beschreibung des normalen Fahrzeugverhaltens sind, erfolgt für die Berechnungen (4.29) - (4.31) ausschließlich anhand der Daten, was einen höheren Automatisierungsgrad gewährleistet. Eine statistische Skalierung der Scores erleichtert durch die Umformung in Wahrscheinlichkeiten die Lesbarkeit und Interpretation [KKSZ11]. Dazu können die Trainingscores S verwendet werden, um eine Schätzung von Mittelwert und Standardabweichung zu berechnen [KKSZ11]. Anhand dessen kann dann eine Gamma- bzw. Normalverteilung modelliert werden. Die Verteilungsfunktion F wird standardisiert durch:

$$\frac{F - F(\mu_S)}{1 - F(\mu_S)}, \quad (4.32)$$

mit μ_S als Mittelwert der Scores. Die Entscheidung, welche Verteilung angepasst werden soll, wird anhand des Histogramms der Trainingscores getroffen. Bild 4.21 zeigt beispielhaft die Annäherung einer Gamma- bzw. Normalverteilung an die Trainingscores. Zur Bestimmung der Wahrscheinlichkeit von Datenpunkt \mathbf{x}_v wird die normierte Wahrscheinlichkeitsdichte verwendet.

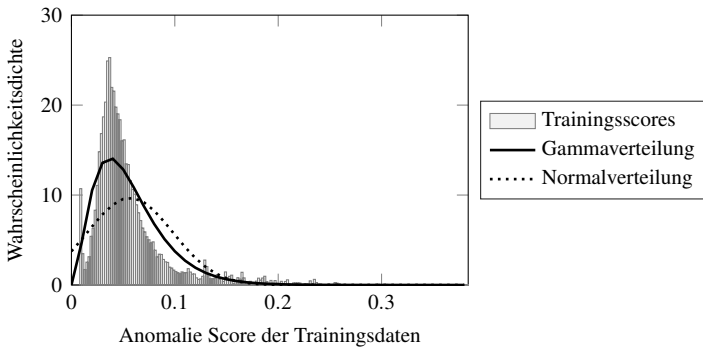


Bild 4.21: Anomalie Scores der Trainingsdaten und angepasste Verteilungsfunktionen

4.2.2 Evaluationskriterien für Anomalien

Die Scores definieren die Wahrscheinlichkeit, dass eine Beobachtung eine Anomalie ist. Die Grenze zur Unterscheidung zwischen *klassifiziert als anormal* oder *klassifiziert als normal* muss vom Benutzer gestellt werden. Wie in den Formeln (4.28) - (4.31) gezeigt, können dazu beispielsweise die Trainingscores als Basis verwendet werden.

Die Grenzfestlegung ist allerdings nicht trivial und hängt stark von den Methoden ab. Stehen gelabelte Testdaten zur Verfügung, kann man die *Receiver-Operating-Characteristic-Kurve* (ROC-Kurve) verwenden, um die Güte einer Methode für jeden möglichen Grenzwert zu visualisieren. Die ROC-Kurve ist ein Maß, das eine Bewertung der Methoden unabhängig von der Grenzfestlegung ermöglicht. Für jeden Grenzwert kann der Vergleich zwischen Grundwahrheit und vorhergesagtem Label mit Anomalie als *positives Ereignis* gezogen werden:

True Positives (TP): Anzahl der Anomalien, die als anormal erkannt werden.

False Positives (FP): Anzahl der normalen Datensätze, die als anormal erkannt werden.

True Negatives (TN): Anzahl der normalen Datensätze, die als normal erkannt werden.

False Negatives (FN): Anzahl der Anomalien, die als normal erkannt werden.

Es wird pro möglichen Grenzwert der Anteil der Anomalien, die als anormal klassifiziert werden (*True Positive Rate*, TPR) und der Anteil der normalen Datenpunkte, die fälschlicherweise als anormal erkannt werden (*False Positive Rate*, FPR), berechnet. Im Diagramm werden für die unterschiedlichen Grenzwerte die TPR als y-Wert gegen die FPR als x-Wert geplottet. Es wird von Testvektoren ausgegangen, die Datenpunkte der Klassen *normal* und *anormal* repräsentieren. Der größtmögliche Grenzwert τ führt dazu, dass alle Datenpunkte als normal eingestuft werden (TPR=0, FPR=0). Entsprechend folgt für das Minimum als Grenzwert, dass alle Anomalien erkannt werden. Jedoch werden auch alle normalen Datenpunkte als anormal klassifiziert (TPR=1, FPR=1). Unterschiedliche Methoden können anhand der Fläche, die die ROC-Kurve einnimmt (*Area under Curve*, AUC), verglichen werden, ohne einen Grenzwert zu definieren. Eine Fläche kleiner 0,5 bedeutet, dass selbst der Zufall besser als das Verfahren ist. Je größer die Fläche, desto besser ist das Verfahren zur Unterscheidung zwischen den Klassen geeignet.

Die ROC-Kurve für das Beispiel aus Bild 4.20 wird in Bild 4.22 gezeigt. Die entsprechenden Werte von TPR und FPR, resultierend aus den unterschiedlichen Grenzwerten, finden sich in Tabelle 4.2.

Ist ein Grenzwert definiert, sodass eine Klassifikation stattfindet, ist der F-Score gebräuchlich zur Bewertung von Datensätzen deren Labels ungleich verteilt sind, da besondere Rücksicht auf die Anomalien als positives Ereignis

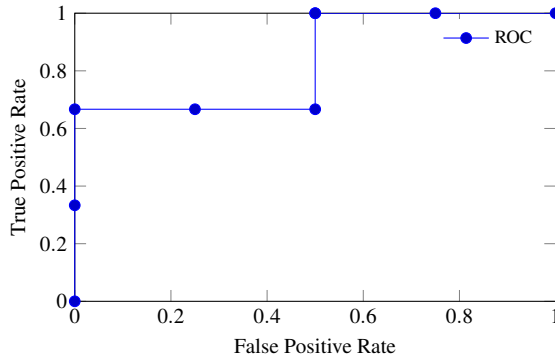


Bild 4.22: ROC-Kurve für das Beispiel aus Bild 4.20

Tabelle 4.2: Performance mit unterschiedlichen Grenzwerten für das Beispiel aus Bild 4.20

τ	3,0	2,67	2,5	2,0	1,53	1,0	0,67	0,0
TPR	0,0	0,33	0,67	0,67	0,67	1,0	1,0	1,0
FPR	0,0	0,0	0,0	0,25	0,5	0,5	0,75	1,0

nis genommen wird. Es handelt sich um das harmonische Mittel aus Genauigkeit (engl. *precision*) und Trefferquote (engl. *recall*) [BDA13]:

$$F\text{-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.33)$$

mit Precision als Anteil tatsächlicher Anomalien unter den gefundenen Anomalien,

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4.34)$$

und Recall als Anteil erkannter Anomalien unter den tatsächlichen Anomalien,

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (4.35)$$

Dies ermöglicht eine faire Bewertung, wenn wenige Datenpunkte der Klasse *anormal* im Testdatensatz vorliegen. Die Treffergenauigkeit (engl. *accuracy*) beispielsweise, die lediglich den Anteil richtig eingeordneter Klassen bewertet

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \quad (4.36)$$

würde bei anteilig wenigen Anomalien einen hohen Wert erzielen, wenn alle Datenpunkte als normal einstufung werden. Je höher der F-Score, desto besser ist die Methode zur Erkennung von Anomalien geeignet [BDA13, Faw06]. Da davon ausgegangen werden kann, dass die Anzahl an Anomalien deutlich kleiner ist als die normaler Datensätze und der Fokus auf ihnen liegt, ist der F-Score für dieses Problem ein passendes Kriterium. Zur Berechnung muss vorher ein eindeutiger Grenzwert festgelegt werden, um zwischen den Klassen *normal* und *anormal* unterscheiden zu können.

Um die zeitliche Abhängigkeit der Signale zu berücksichtigen, wird in Kapitel 4.1.2 zur Vorhersage von Testvektor \mathbf{x}_v die Vergangenheit verwendet. Doch auch für die Evaluierung und Entscheidung spielt der Zeitbezug eine Rolle. So kann statt eines einzelnen Testvektors \mathbf{x}_v eine Testzeitreihe Z_v der Länge $m + 1$ betrachtet werden (vgl. Formel 4.20). Diese kann in Zeitfenster der Länge p unterteilt werden (vgl. Formel 4.21). Es wird für jeden Datenpunkt $\mathbf{x}_v \in Z_v$ ein Anomalie Score berechnet. Jedoch werden für die weitere Betrachtung die Zeitfenster statt der einzelnen Punkte verwendet. Dies führt zur zusätzlichen Entscheidung, welcher Anteil eines Zeitfensters als *anormal* gelten muss, um es in der Gesamtheit als *anormal* einzustufen. Der Vorteil dabei ist, dass der Kontext in dem sich die Anomalie befindet, nicht verloren geht. Eine andere Vorgehensweise ist die Klassifikation der Punkte $\mathbf{x}_v \in Z_v$ und das bündeln zeitlich aufeinanderfolgender Anomalien zu einer. Durch eine Mindestanzahl zusammengehörender *anormal*er Datenpunkte können Fehlklassifikationen durch Rauschen und Mängel in den Daten reduziert werden [The14].

4.2.3 Unterstützung bei der Interpretation einer Anomalie

Im Hinblick auf die Nachvollziehbarkeit, warum eine Anomalie als anormal erkannt wird, sind Clusteransätze und Replikator Neuronale Netze besonders geeignet, da sich hier der Anomalie Score für Datenpunkt \mathbf{x}_v aus den einzelnen Scores pro Feature ergibt:

$$score = \frac{1}{d} \sum_{i=1}^d score_i. \quad (4.37)$$

Die Untersuchung, welche der d Features großen Einfluss auf den Anomalie Score nehmen, ermöglicht Rückschlüsse auf den Grund für das anormale Verhalten. Für die Methoden, deren Ausgaben weniger leicht zu interpretieren sind, kann folgendes Random Forest Verfahren als Alternative angewandt werden [KHP14, PK14, GPTM10]: Es werden die zu einer Anomalie zusammengefassten Datenpunkte als Repräsentant der Klasse *anormal* verwendet. Außerdem wird von den normal angenommen Trainingsdaten ein Sample gezogen, welches die Klasse *normal* vertritt. Damit wird ein binärer Entscheidungsbaum gelernt. Die Features, die zur Generierung des Baumes verwendet werden, sind wichtig, um die Anomalie vom Rest zu trennen. Sie können als die Features, entscheidend für die Detektion als Anomalie, interpretiert werden. Durch das Lernen mehrerer Bäume mit unterschiedlichen Untergruppen, findet ein Ranking der Features statt. Es kann gezählt werden, wie oft welches Feature zur Trennung in *normal* und *anormal* verwendet wird. Unterschied zu den Bäumen im Isolation Forest ist, dass die Features und die Trennwerte nicht zufällig gewählt werden. Sie werden so gesetzt, dass sich die Anomalie durch möglichst wenige Knoten vom Rest abgrenzen lässt, z.B. durch die Maximierung des Informationsgehalts [KHP14, PK14, GPTM10]. Mit diesem Verfahren kann die Entscheidung des Klassifikators nachvollzogen werden, für eine vollständige Interpretation der Anomalie ist jedoch weiteres Postprocessing erforderlich. Es benötigt einen Experten, der die Anomalie einordnet. Allerdings kann dieser durch

den Hinweis darauf, welche Signale ausschlaggebend und somit interessant für weitere Analysen sind, unterstützt werden.

Bedeutung des Postprocessings

Zusammengefasst können durch das Postprocessing, bestehend aus Entscheidung und Interpretation, die vorgestellten Anforderungen *Robustheit gegen Anomalien in den Trainingsdaten* und *Unterstützung bei der Interpretation* nachträglich erfüllt werden. Durch eine datenbasierte Bestimmung des Grenzwertes anhand der Trainingsscores können die außergewöhnlichsten Trainingspunkte ignoriert werden, was eine Robustheit gewährleistet. Der Fahrzeughersteller legt durch Formel 4.28 den Anteil normaler Trainingsdaten (x) als Parameter fest. Unabhängig von Fahrzeugflotte und Referenzmodell kann der Wert zur Regulierung der Datenübertragung verwendet werden. Je höher der Wert, desto weniger Daten werden übertragen und desto stärker muss ein Ereignis von den Trainingsdaten abweichen, um als Anomalie erkannt zu werden. Nur Datenpunkte die *anormaler*, sprich einen höheren Anomalie Score als $x\%$ der Trainingsdaten haben, werden vom Fahrzeug an das Backend übermittelt. Anhand von Beispielen wird in Kapitel 5 gezeigt, wie sich die Wahl des Grenzwertes auf die Anomalieerkennung auswirkt und welche Ereignisse für unterschiedliche x -Werte detektiert werden.

Für undurchsichtige Methoden liefert ein Random Forest Ansatz für Feature Selektion eine Möglichkeit erkannte Anomalien nachzuvollziehen. Ein Anomalie Score ist für alle Techniken gegeben. Daher reduziert sich das wesentliche Entscheidungskriterium auf die Übertragbarkeit ins Fahrzeug. Es gilt die Methode mit bester Performance zu finden. Möglichst viele Anomalien müssen erkannt werden ohne normale Ereignisse fälschlicherweise als anormal einzustufen.

5 Prototypische Anwendung

Zur Bewertung und zum Vergleich der in Kapitel 4.1.1 vorgestellten Methoden werden diese auf verschiedene Datensätze angewandt. Grundlage hierfür sind Fahrzeugdaten, gesammelt über 14 Monate in ganz Europa. Es wurden die persönlichen Dienstwagen von etwa siebzig AUDI-Managern mit einem Datenlogger ausgestattet und dadurch 630.000 km eingefahren. 400 Gigabyte an Datensätzen, bestehend aus 850 unterschiedlichen Signalen und zusammengesetzt aus sieben Fahrzeugtypen, stehen zur Auswertung bereit [AUD16b, Ste16].

Anhand der Fahrzeugdaten wird ein Referenzmodell als Beschreibung des normalen Fahrzeugverhaltens trainiert. Diese Daten sind nicht gelabelt, weswegen keine Gewissheit vorliegt, was darin Anormales passiert ist. Allerdings stehen den Benutzern immer die neuesten Fahrzeuge zur Verfügung und es ist bekannt, dass kein Fahrzeug aufgrund von Schäden, Mängeln oder Unfällen zurückgegeben wurde.

Zur Evaluierung liegen vollständig protokollierte Fahrten vor. Diese beinhalten normale Abschnitte, aber auch außergewöhnliche Fahrsituationen wie Unfälle, ABS- oder ESP-Eingriffe. Des Weiteren werden konkrete Fahrmanöver provoziert. Es wird beispielsweise eine besonders hohe Drehzahl erzeugt, mit offener Motorhaube gefahren oder eine Vollbremsung gemacht. Zusätzlich werden synthetische Testfälle generiert, die gestörte Signalzusammenhänge und kontextuelle Anomalien beschreiben.

Anschließend wird untersucht, in wie weit sich diese Ereignisse durch das Referenzmodell von den Flottendaten unterscheiden lassen. Es sind Beispiele, die zur Evaluierung des Konzepts dienen und die Anwendbarkeit des

Verfahrens zeigen. Eine genaue Beschreibung der Anomalien und der Datenbasis folgt in Abschnitt 5.1. Die Erläuterung, wie die Ergebnisse auch mit teilweise ungelabelten Daten evaluiert werden, befindet sich in Abschnitt 5.2, darauffolgend die Versuche und Ergebnisse in Abschnitt 5.3.

5.1 Datenüberblick

Es werden Ereignisse als *Anomalien* definiert und untersucht. Die Methoden werden evaluiert, indem überprüft wird, ob sie diese als anormal erkennen. Betrachtet werden Abschnitte aus den Flottendaten und Ereignisse, die mit einem Testfahrzeug selbst eingefahren und provoziert wurden. Zusätzlich werden synthetische Testfälle generiert:

Fahrevent, Flotte: Folgende Fahrevents, die auf schwierige Fahrverhalten schließen, sind in den Flottendaten geschehen:

1. Bei einem **Unfall** handelt es sich um eine sehr außergewöhnliche Fahr-situation, die vermieden werden muss.
2. Ein **ESP-Eingriff** korrigiert Fahrfehler, um ein Ausbrechen des Fahrzeugs zu verhindern.
3. Auch ein **ABS-Eingriff** kann als Anomalie verstanden werden, da sich eine Vollbremsung ableiten lässt.

Fahrevent, provoziert: Anomalien, die ein ungewöhnliches Fahrverhalten beschreiben, werden eingefahren. Um keine riskanten Manöver zu produzieren, beschränken sich die Testfälle auf folgende Ereignisse:

1. Beim Event **Überdrehzahl** liegt die Drehzahl in dem vom Hersteller angegebenen *roten Bereich*.
2. Während der Fahrt mit einer Geschwindigkeit von etwa 80 km/h wird eine **Vollbremsung** gemacht.
3. Während der Fahrt mit einer Geschwindigkeit von etwa 50 km/h wird die **Handbremse während der Fahrt** gezogen.

4. Zur außergewöhnlichen Beeinflussung der Dämpfung werden enge **Kreise in vor- und rückwärts Richtung** gefahren.
5. Die **Herausforderung des Spurhalterassistenten** ist eine Anomalie, die durch das gezielte Gegenlenken entsteht.

Status Fahrer/Fahrzeug, provoziert: Es werden die folgenden Anomalien provoziert, die einen ungewöhnlichen Zustand von Fahrzeug oder Fahrer beschreiben:

1. Die **Fahrt mit offener Tür** steht für Ereignisse, in denen der Fahrer Hinweise (Meldung, dass die Tür nicht geschlossen ist) nicht bekommt, ignoriert oder übersieht.
2. Da in der Flotte größtenteils nur eine Person im Fahrzeug sitzt, wird erwartet, dass das **Öffnen des Beifahrerfensters**, das **Öffnen des Fensters hinten rechts** und das **Öffnen aller Fenster** als Anomalien erkannt werden.
3. Das **Schließen des Fensters mit eingeklemmtem Buch** ist ein weiteres Ereignis.
4. Da es noch keine fahrerlosen Fahrzeuge gibt, ist die Situation, in der **nur der Beifahrer angeschnallt** ist, ungewöhnlich. Auch hier ignoriert oder übersieht der Fahrer Hinweise.
5. Bei den Fahrzeugen handelt es sich ausschließlich um neue Modelle, in denen der Heckdeckel elektrisch durch einen Knopf geschlossen werden kann. Das **Schließen des Heckdeckels per Hand** stellt daher eine Anomalie dar.
6. Auch ein **offener Tankdeckel** während der Fahrt tritt erwartungsgemäß selten auf.
7. Beim Abstellen des Fahrzeugs fährt der Spoiler mehrmals ein und wieder aus (**Spoiler spielt verrückt**), was als Anomalie aufgenommen wird.

8. Als weitere Anomalie wird das **Fernlicht am Tag** eingeschaltet.

Synthetische Testfälle: Zusätzlich werden synthetische Testfälle erzeugt, indem die Kontexte *konstant fahrendes Fahrzeug*, *anfahrendes Fahrzeug*, *stehendes Fahrzeug* und *abbremsendes Fahrzeug* aus den Flottendaten extrahiert und manipuliert werden. Es handelt sich um reale und normale Ereignisse, die verfälscht werden. Es werden unter anderem Anomalien generiert, die auf eine **Fahrzeugmanipulation** hindeuten, da der Fahrer durch seine Aktionen das Fahrzeug nicht beeinflussen kann. Eine detaillierte Beschreibung folgt in Abschnitt 5.3.3.

Bild 5.1 zeigt, wie sich die Datenbasis zusammensetzt. Das allgemeine Vorgehen aus Kapitel 3.2, Bild 3.4 bedeutet hier im konkreten Fall, dass die Schwarmdaten mittels sieben unterschiedlicher Kampagnen, die sich in der Baureihe unterscheiden, gesammelt wurden. Es liegen gelabelte Testfälle für zwei Baureihen vor: Die Fahrten mit anormalen Fahrevents innerhalb der Flotte (Unfall, ABS- und ESP-Eingriff; Baureihe 1) und die selbst durchgeführten Fahrten mit provozierten Ereignissen (Fahrevent und Status Fahrer/Fahrzeug; Baureihe 2). Für diese beiden Kampagnen werden Referenzmodelle trainiert und evaluiert. Zusätzlich stehen jeweils die synthetischen Testfälle zur Verfügung.

Tabelle 5.1 liefert eine zusammenfassende Beschreibung der Datenbasis. Für beide Kampagnen liegen über 100 Fahrstunden bereit. Diese dienen zum Training des Referenzmodells. Untersucht werden 36 bzw. 73 unterschiedliche Signale. Zur Evaluierung sind in Kampagne K_1 3,3 Stunden gelabelte Fahrzeit gegeben, worunter sich die Anomalien der Kategorie *Fahrevents* und *synthetische Testfälle* befinden. In Kampagne K_2 sind es 10 Stunden gelabelte Fahrzeit mit den Anomalien eingeordnet in *Fahrevents (provoziert)*, *Status Fahrer/Fahrzeug* und *synthetische Testfälle*. Die Testdaten repräsentieren hierbei die Daten der Anwendungsphase aus Bild 3.2b.

Es werden die im Folgenden aufgelisteten Signale untersucht. Doppelte Nennungen lassen sich erklären, da unterschiedliche Steuergeräte teilweise

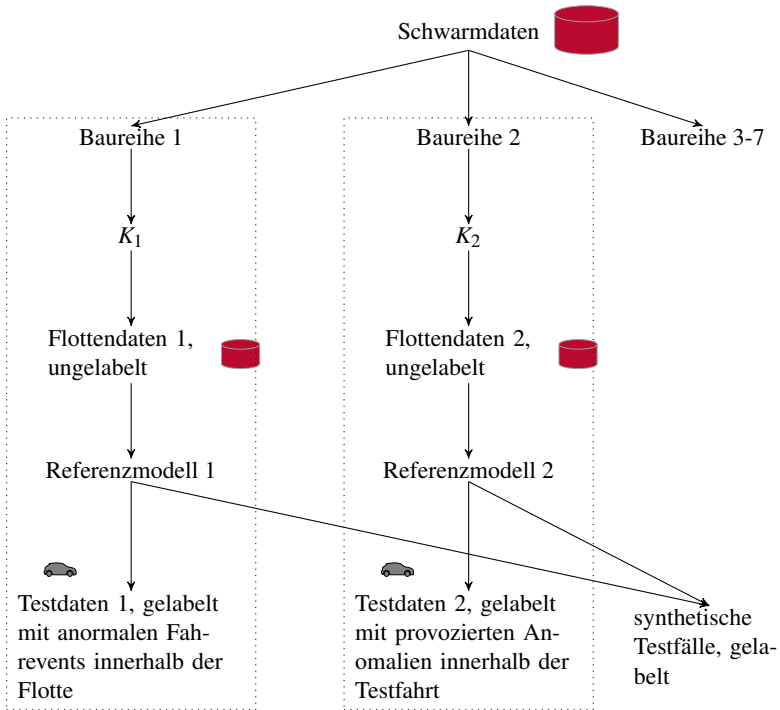




Bild 5.1: Generierung der Datenbasis

dieselbe Information liefern. Zur Erklärung der Signale, welche die Achsen des Fahrzeugs in x-, y- und z-Richtung verwenden, veranschaulicht Bild 5.2 diese.

Signalliste von Kampagne 1:

- Geschwindigkeit [km/h]
- Lenkmoment [Nm]
- Bremsdruck [bar]
- Längsbeschleunigung [m/s^2]
- Querbeschleunigung [g]
- Beschleunigung in x-Richtung [m/s^2]
- Beschleunigung in y-Richtung [m/s^2]
- Beschleunigung in z-Richtung [m/s^2]
- Winkelgeschwindigkeit (ω) um die z-Achse [$^\circ/s$]
- Lenkradwinkel [$^\circ$]
- Lenkradwinkelgeschwindigkeit [$^\circ/s$]
- Motordrehzahl [rpm]
- Fahrpedal-

Tabelle 5.1: Datenüberblick

	K_1	K_2
Baureihe	1	2
Flottendaten, ungelabelt 	103 Stunden Fahrt	102 Stunden Fahrt
Anzahl Signale (d)	36	73
Testdaten, gelabelt 	3,3 h Fahrt, davon 4,5% anormal	10 h Fahrt, davon 10% anormal
Anomalien	<ul style="list-style-type: none"> • Fahrevents innerhalb der Flotte • synthetische Testfälle 	<ul style="list-style-type: none"> • selbst provozierte Anomalien (Fahrevents, Status Fahrer/Fahrzeug) • synthetische Testfälle

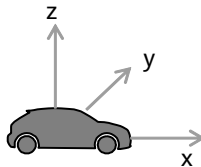


Bild 5.2: Achsenbeschriftung Fahrzeug

rohwert [%] • Verbrauchszähler [*microliter*] • Kühlmitteltemperatur [$^{\circ}\text{C}$] •
 Massenträgheit des Motors [kgm^2] • Öltemperatur [$^{\circ}\text{C}$] • Leerlaufsoldreh-
 zahl, die im Motor eingestellt ist [*rpm*] • Außentemperatur (ungef) [$^{\circ}\text{C}$]
 • Verlustmoment [-] • relative Luftfeuchte [%] • Anteil der Motorstand-
 zeit zwischen Zündung aus und nächstem Motorstart [*s*] • Kilometerstand
 [*km*] • Tankfüllstand [%] • Tankinhalt [*l*] • angezeigte Geschwindigkeit
 im Tachometer [*km/h*] • Außentemperatur [$^{\circ}\text{C}$] • Batteriespannung [*V*]
 • Luftgüte, Kohlenmonoxid-Anteil in der Luft (Luftgüte CO) • Luftgü-
 te, Stickoxide-Anteil in der Luft (Luftgüte NOX) • Steigung Berg [%] •
 Signatur der Zentralverriegelung • Motormoment vor Filter, nutzbar zur

Zug-/Schubererkennung beim Getriebe • Verlust Motormoment, beinhaltet Reibung vom Motor und Klimakompressor • Indiziertes Motormoment, d.h. idealisiertes berechnetes Moment ohne mechanische Verluste, aber mit externen Eingriffen • Heckdeckel Hallsensoren Positionserkennung (HD HS Pos)

Signalliste von Kampagne 2: Ähnliche Signalauswahl wie in Kampagne K_1 . Es fallen weg:

- Anteil der Motorstandzeit zwischen Zündung aus und nächstem Motorstart [s] • Tankfüllstand [%] • Tankinhalt [l] • Signatur der Zentralverriegelung

Hinzu kommen:

- Belegungserkennung Beifahrer vorne • Gurtwarnung Beifahrer vorne • Innenlicht vorne aktiviert • Leselicht vorne aktiviert • Blinkerhebel rechts • Richtungsblinken rechts • Blinkerhebel Fernlicht • Kombi-Takt Blinker links • Kombi-Takt Blinker rechts • Beifahrertür (BT) Fensterheber (FH) ist in Bewegung hoch • BT FH ist in Bewegung tief • BT FH befindet sich im mechanischen Block • BT FH, Fangbereich ist 4 Millimeter von oberer Lippendichtung • BT FH Öffnung [%] • BT FH ist in Bewegung hoch, automatisch • hinten rechts (HR) FH ist in Bewegung hoch, manuell • BT FH ist in Bewegung hoch, manuell • BT Spiegelheizung ein/aus • BT Tür geöffnet • BT Tür sicher geschlossen (Sperrklinke) • Fahrertür (FT) FH Öffnung [%] • FT Spiegelheizung ein/aus • absolute Heckklappenposition (HKP) • relative Heckklappenposition [%] • Fahrer brems • Fahreraktivität Start/Stop • Aktivitätsstatus Start/Stop-Koordinator (liegt Freigabe vor) • Automatikgetriebe, aktuelle Fahrstufe • Automatikgetriebe, momentane simulierte Gangstufe • y-Abstand zum Fahrzeug-Koordinatensystem [m] • Von Fahrassistenten angefordertes Lenkmoment [Nm] • eingestellter, aktiver Lenkeingriffszeitpunkt (früh/spät) • Zeit bis das Fahrzeug die Fahrbahnmarkierung überquert (*Time Lane Cross* (TLC)) [s] • Abstand vom Fahrzeug zur Fahrbahnmarkierung (*Distance Lane Cross* (DLC)) [m]

- Zuordnung auf welche Seite sich TLC und DLC beziehen, links/rechts
- Klemme 50: Startausführung
- Segelbetrieb freigegeben
- Bremslicht aktiv
- Innenbeleuchtung aktiv
- minimale Batteriespannung im letzten Start [V]
- Strombereich der Batterie [A]

Zur Bewertung, ob es sich bei den gegebenen Testfällen wirklich um seltene Ereignisse handelt, werden die Flottendaten näher untersucht:

Flottendaten der Kampagne 1: In Kampagne K_1 stehen zwei **Unfälle**, zwei **ABS-Eingriffe** mit gleichzeitigem **ESP-Eingriff** und vier **ABS-Eingriffe** als Anomalien zur Verfügung. Zusätzlich werden die synthetischen Testfälle geprüft. Zum Vergleich, in den Flottendaten befinden sich keine Unfälle und nur 0,0010% bzw. 0,1675% der Flottendaten beinhalten ABS- bzw. ESP-Eingriffe.

Flottendaten der Kampagne 2: In Kampagne K_2 liegen neben den synthetischen Testfällen auch die unter den Punkten *Fahrevent*, *proviziert* und *Status Fahrer/Fahrzeug* aufgelisteten Ereignisse als Anomalien vor. Das Ereignis **Überdrehzahl** ist mit unterschiedlichen Ausprägungen (von 4700 rpm bis über 6000 rpm) gegeben. Hier zeigen die Flottendaten, dass in 99,43% der Fahrzeit eine Drehzahl kleiner als 4700 rpm vorkommt. Das eingeklemmte Buch wird als noch nie geschehen vorausgesetzt. In 99,95% der Fälle sind sowohl Fahrer- als auch Beifahrerfenster um weniger als die Hälfte geöffnet, während hinten rechts in 0,0014% der Daten der Fensterheber nach oben gedrückt wird. Eine geöffnete Tür bei einer Fahrt schneller als 10 km/h liegt in 1,7% der Daten vor, bei Fahrzeugstillstand in etwa 7,4% der Daten. Die Öltemperatur liegt zu 99% zwischen 0 und 108°C und innerhalb der Flottendaten kann die Betätigung des Fernlichts nur einmal um ein Uhr nachts gefunden werden. Die maximale relative Heckklappenöffnung beträgt in den Flottendaten 91%, während beim Schließen per Hand die 100% erreicht werden können.

5.2 Evaluationsvorgehen

Die Flottendaten aus Bild 5.1 sind nicht gelabelt und es ist nicht bekannt, welche Ereignisse sich darunter befinden. Das in Kapitel 3.2 vorgestellte Konzept ist es, diese zum Training des Referenzmodells zu verwenden, um die Normalität zu definieren. Eine robuste Methode ist zeitgleich dazu fähig, außergewöhnliche Ereignisse zu ignorieren (vgl. Kapitel 3.3, *Anforderungen*). Es wird für die konkreten Beispiele untersucht, wie viel das Modell als normal abdeckt und in wie weit die Anomalien von den Flottendaten unterschieden werden können. Im Rahmen der prototypischen Anwendung werden folgende Verfahren verglichen:

- Replikator Neuronales Netz (ReplNN) mit einer verborgenen Schicht und d Neuronen (vgl. Kapitel 4.1.1).
- Isolation Forest (IForest) mit 100 Bäumen und je 400 Samples (vgl. Abschnitt 5.3.1).
- *Self Organizing Map* (SOM) mit einer rechteckigen Topologie.

Auf die *one-class* Support Vektor Machine wird aufgrund der hohen Rechenkomplexität von bis zu $O(n^3)$ verzichtet.

Bild 5.3 stellt das Evaluationsvorgehen für eine Kampagne vor. Hierbei liefern die unterschiedlichen Verfahren Referenzmodelle, die verglichen werden. Zur Beurteilung wird die Frage beantwortet, ob das Modell die Mehrheit der Flottendaten abdeckt. Außerdem wird untersucht, ob sich die Anomalien von den normal angenommenen Trainings- bzw. normal gelabelten Testdaten unterschieden lassen. Die dazu notwendigen Evaluationskriterien werden im Folgenden detailliert vorgestellt.

1. Deckt das Referenzmodell die Mehrheit der Flottendaten ab?

In Kapitel 3.2 wurde gezeigt, dass das Referenzmodell nur zur Reduzierung der Datenmenge im Fahrzeug verwendet werden kann, wenn es die Mehrheit der Flottendaten als normal klassifiziert. Um dies zu überprüfen, werden

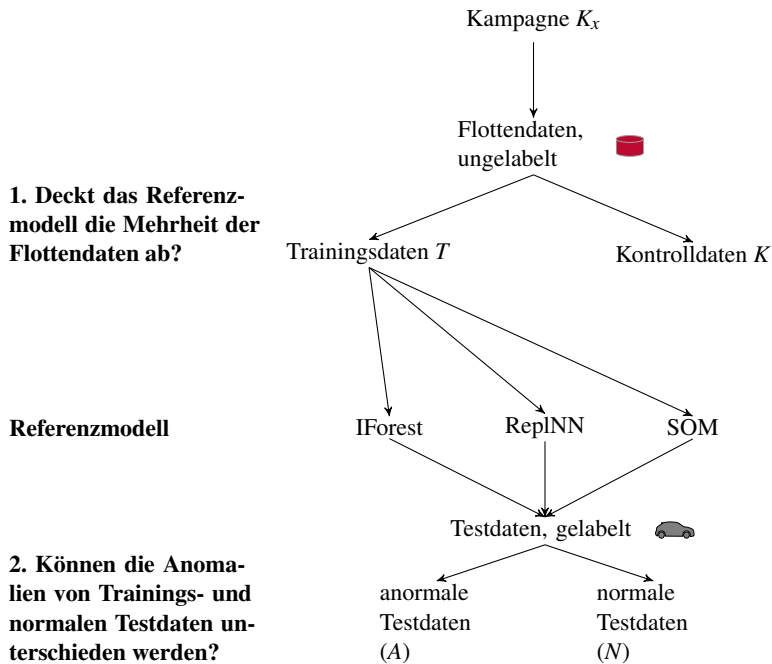


Bild 5.3: Evaluationsvorgehen je Kampagne

im ersten Schritt die ungelabelten Flottendaten aufgeteilt in Trainings (T)- und Kontrolldaten (K) (vgl. Bild 5.4).

Mithilfe der Trainingsdaten wird das Modell trainiert und deren Anomalie Score berechnet. Zur Spezifikation der *Mehrheit* werden dann gemäß Formel 4.28, vorgestellt in Abschnitt 4.2.1, aufsteigend 0% bis 100% der Trainingsdaten als normal angenommen und der daraus resultierende Grenzwert $\tau_{q_x}, x \in [0, 1]$ zur Klassifizierung der Kontrolldaten verwendet. Die Evaluierung überprüft, welcher Anteil der Kontrolldaten einen höheren Score hat und somit als Anomalie gilt. Der Anteil erkannter Anomalien in den Kon-

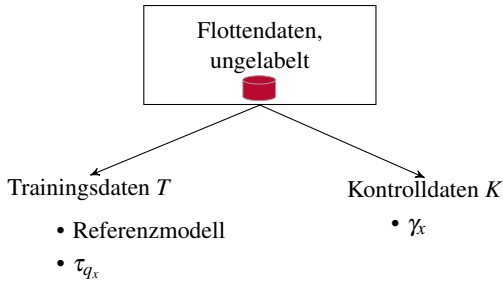


Bild 5.4: Aufteilung der ungelabelten Flottendaten in Trainings- und Kontrolldaten

trolldaten ist daher der Anteil mit einem höheren Score als $100 \cdot x \%$ der Trainingsdaten:

$$\gamma_x = \frac{|\text{score}(K) > \tau_{q_x}|}{|K|}. \quad (5.1)$$

Wie in Kapitel 3.2 dargestellt, sind sowohl Trainings- als auch Kontrolldaten aus den Flottendaten gezogene Samples, für die angenommen wird, dass sie derselben Verteilung folgen. Wenn der Anteil erkannter Anomalien in etwa übereinstimmt, kann sichergestellt werden, dass das Referenzmodell weder *overfittet* (also nur für die Trainingsdaten funktioniert und alles andere als anomal erkennt), noch dass es zu stark generalisiert und alles als normal erkennt.

2. Können die Anomalien von Trainings- und normalen Testdaten unterschieden werden? Die Fragestellung lässt sich in die beiden folgenden Teilfragen aufteilen:

Können die Anomalien von den Trainingsdaten unterschieden werden?

Zur Evaluierung wird geprüft, wie gut sich die Anomalien von den Trainingsdaten unterscheiden lassen (vgl. Bild 5.5). Dazu werden die Trainingsdaten als *normal* betrachtet und die in Kapitel 4.2.1 behandelte Fläche unter der ROC-Kurve berechnet (AUC_{tr}). Diese zeigt wie gut der Score geeignet ist, um die Ereignisse zu unterscheiden. Ein Wert nahe eins bedeutet per-

fekte Trennbarkeit, ein Wert kleiner 0,5 meint, dass selbst der Zufall besser wäre. Außerdem wird je Anomalie $a_j, j = 1, \dots, |A|$, mit A als die Menge der Anomalien, der Anteil der Trainingsdaten mit einem größeren Score als die Anomalie berechnet:

$$\xi_j = \frac{|\text{score}(T) > \text{score}(a_j)|}{|T|}, j = 1, \dots, |A|. \quad (5.2)$$

Der Wert kann als *False Positive Rate* interpretiert werden. Da die Trainingsdaten allerdings nur normal angenommen werden und keine Grundwahrheit vorliegt, entspricht das nicht ganz der Wahrheit. Es ergibt sich folgender Zusammenhang für Grenzwert τ_{q_x} :

$$\text{score}(a_j) > \tau_{q_x}, \text{ für } x = 1 - \xi_j. \quad (5.3)$$

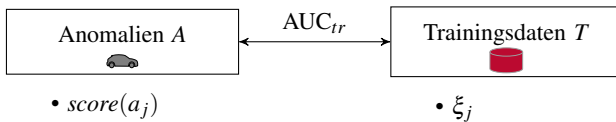


Bild 5.5: Verwendung der Trainingsdaten zur Bewertung der Anomalieerkennung

Können die Anomalien von normalen Fahrsituationen unterschieden werden?

Da die gelabelten Testdaten sowohl Anomalien als auch normale Fahrtabschnitte verfügen, werden diese verwendet, um zu evaluieren, wie gut sich beispielhaften Anomalien von den normalen Datenpunkten unterscheiden lassen (vgl. Bild 5.6). Wieder wird die Fläche unter der ROC-Kurve (AUC_{te}) als Kriterium verwendet. Außerdem wird berechnet, was die Wahl des Grenzwertes τ_{q_x} für die fälschlicherweise als anormal erkannten, nor-

malen Testdaten bedeutet. Da hier Labels vorliegen, handelt es sich bei dem Wert um die *False Positive Rate*:

$$FPR_x = \frac{|\text{score}(N) > \tau_{q_x}|}{|N|}, x \in [0, 1]. \quad (5.4)$$

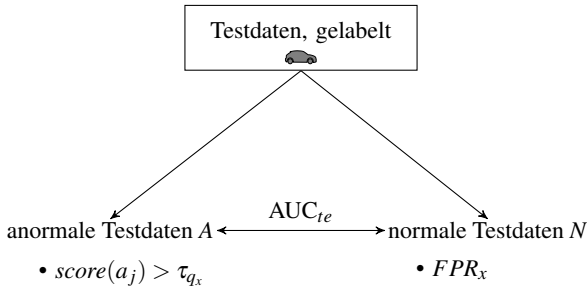


Bild 5.6: Verwendung der gelabelten Testdaten zur Bewertung der Anomalieerkennung

Für die Testfälle ist die Grundwahrheit nicht pro Zeiteinheit, sondern für eine Folge von Zeitpunkten (*Zeitabschnitt*) gegeben. Der durch das Referenzmodell vorhergesagte Anomalie Score allerdings berechnet sich pro Zeiteinheit (vgl. Bild 5.7). Dadurch kann der vorhergesagte Anomalie Score nicht direkt mit der Grundwahrheit verglichen werden. Eine Herangehensweise ist deshalb die Berechnung des durchschnittlichen Scores pro Zeitabschnitt als Gesamtscore S (vgl. Abschnitt 4.2.2). Anstatt alle Zeitpunkte einzubeziehen, die möglicherweise größtenteils normale Momente beinhalten, werden nur die t größten Scores zur Berechnung verwendet. T beschreibt die Gesamtanzahl an Datenpunkten pro Zeitabschnitt, s_i den Anomalie Score des i -ten Datenpunktes:

$$S = \frac{1}{t} \sum_{i=1}^t s_i, \quad (5.5)$$

$$s_1 \geq s_2 \geq \dots \geq s_T.$$

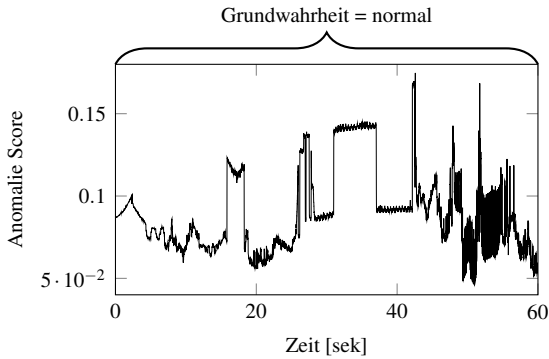


Bild 5.7: Diskrepanz zwischen Anomalie Score und Grundwahrheit: Anomalie Score für jede hundertste Millisekunde, Grundwahrheit pro Minute

Eine andere Herangehensweise ist die Definition von zwei Grenzwerten: Ab wann ist ein Zeitpunkt anormal (τ) und wie viele Zeitpunkte müssen pro Zeitabschnitt anormal sein, damit dieser als anormal gilt und mit der Grundwahrheit verglichen werden kann (ζ) (vgl. Kapitel 4.2.1):

$$\text{Minute} = \begin{cases} \text{anormal}, & \text{wenn } |M| > \zeta, \\ \text{normal}, & \text{sonst} \end{cases} \quad (5.6)$$

$$M = \{(s_1, s_2, \dots, s_T) | s_i > \tau, i = 1, \dots, T\}.$$

Die Zusammenfassung zu Zeitabschnitten hat den Vorteil, dass Rauschen, das durch Übertragungsprobleme oder Messfehler entsteht, ausgeschlossen wird [The13]. Außerdem wird durch die Übertragung des gesamten anormal klassifizierten Zeitabschnittes sichergestellt, dass der Kontext der Anomalie erhalten bleibt.

5.3 Experimente

Zum Vergleich der unterschiedlichen Methoden werden die im vorigen Abschnitt 5.2 beschriebenen Kriterien berechnet.

In Abschnitt 5.3.1 werden zunächst die vorgestellten Methoden (IForest, RepINN, SOM) dahingehend bewertet, ob sie die Mehrheit der Flottendaten abdecken und generell geeignet sind, die Anomalien zu unterscheiden. Dazu werden auch die zu definierenden Hyperparameter des Isolation Forest Ansatzes optimiert. Zusätzlich wird betrachtet, ab wie vielen Trainingsdaten die Verfahren konvergieren und der in Kapitel 3.2, Bild 3.3 konzeptuell dargestellte Übergang von Trainings- zu Anwendungsphase durchgeführt werden kann. Abschnitt 5.3.2 veranschaulicht anhand der Anomalie Scores, welche realen Ereignisse erkannt werden. Die Evaluierung der synthetischen Testfälle folgt in Abschnitt 5.3.3.

5.3.1 Methodenvergleich

Zunächst soll die Frage beantwortet werden, ob das Referenzmodell die Mehrheit der Flottendaten abdeckt. Dadurch kann abgeleitet werden, in wie weit der Datentransfer von Fahrzeug zu Backend grundsätzlich reduziert werden kann. Bild 5.8 zeigt das angepasste Quantil-Quantil-Diagramm der unterschiedlichen Algorithmen für Kampagne K_1 . Die x-Achse des Diagramms beschreibt, welcher Anteil der Trainingsdaten als normal angenommen wird, womit der Grenzwert τ_{q_x} , für $x = 0, \dots, 1$ berechnet wird. Die y-Achse zeigt den entsprechenden Anteil normal erkannter Kontrolldaten ($1 - \gamma_x$). Die durchgezogene Diagonale dient als Referenzlinie, die ein perfektes Übereinstimmen aufzeigt. Werden zu viele Kontrolldaten als anormal klassifiziert, deutet das auf *Overfitting* hin, während ein zu kleiner Anteil möglicherweise auf eine zu starke Generalisierung zurückzuführen ist. Die Diagramme für Kampagne K_2 befinden sich im Anhang 5.8.

Die Bilder zeigen, dass in beiden Kampagnen der Isolation Forest und das Replikator Neuronale Netz ein Referenzmodell liefern, welches auf die

Kontrolldaten übertragbar ist. Tabelle 5.2 zeigt beispielhaft, dass die Wahl des Grenzwertes τ_{q_x} mit $x = 0,95$ eine Reduzierung des Datentransfers gewährleistet. Bei 95% normal angenommener Trainingsdaten, werden bei der Anwendung des Isolation Forests im Fahrzeug 3,45% (Kampagne K_1) bzw. 4,1% (Kampagne K_2) der Daten an das Backend übertragen; für das Replikator Neuronales Netz sind es 4,28% bzw. 5,64%. Zusätzlich zeigt die Tabelle 5.2 die Reduzierung von γ_x bei einer Erhöhung auf $x = 0,99$. Abschnitt 5.2 beschreibt, dass die Anomalie Scores je *Zeitraum* nach Formel 5.5 berechnet werden. Verwendet man anstatt des Mittelwertes die

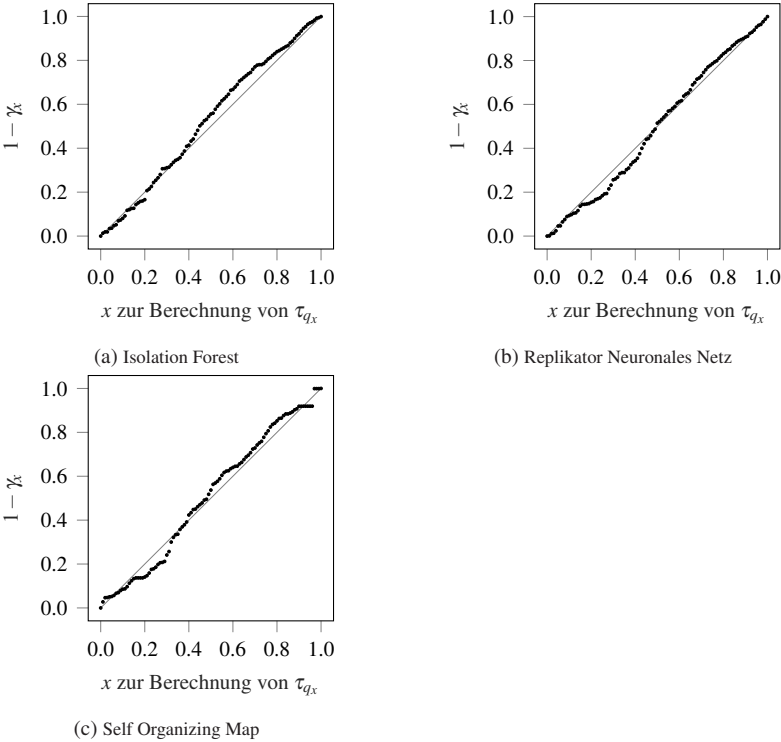


Bild 5.8: Vergleich Trainings- und Kontrolldaten, Kampagne anhand Anomalie Score nach 5.5, Kampagne K_1

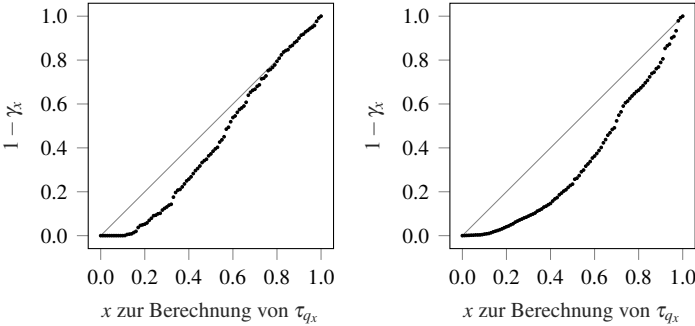
Tabelle 5.2: Anteil erkannter Anomalien in den Kontrolldaten für Grenzwert τ_{q_x} mit $x = 0,95$ und $x = 0,99$

	K_1		K_2	
	$\gamma_{(0,95)}$	$\gamma_{(0,99)}$	$\gamma_{(0,95)}$	$\gamma_{(0,99)}$
IForest	0,0346	0,0072	0,0410	0,0047
ReplNN	0,0428	0,0064	0,0564	0,0079
SOM	0,0808	0,0008	0,0278	0,0196

Rohwerte je Zeitpunkt, ändern sich die Ergebnisse des Replikator Neuronale Netzes, wie in Bild 5.9 dargestellt. Der Anteil anormal erkannter Kontrolldaten γ_x steigt in Kampagne K_2 . Der Datentransfer wird dann bei angenommenen 95% normaler Trainingsdaten auf nur 9,92% statt auf 5,64% reduziert. Die Berechnung der Durchschnittswerte nach Formel 5.5 mindert daher diesen Effekt.

Da alle Verfahren den Datentransfer grundsätzlich reduzieren, stellt sich die Frage, welcher der Methoden die Anomalien am zuverlässigsten erkennt. Um die Entscheidung unabhängig von einem Grenzwert zu treffen, sind in Tabelle 5.3 die Werte AUC_{tr} und AUC_{te} aufgeführt. Sie zeigt, dass das Replikator Neuronale Netz in beiden Kampagnen am besten zwischen den Anomalien und den Trainingsdaten (AUC_{tr}) bzw. den normalen Testdaten (AUC_{te}) unterscheiden kann.

Eine durchschnittliche AUC von 0,8 demonstriert, dass das System nur bedingt dazu geeignet ist, Anomalien von normalen Ereignissen zu trennen. Abschnitt 5.3.2 führt auf, welche Anomalien erkannt werden können und verdeutlicht die Problematik der Rate fälschlicherweise anormal erkannter Datenpunkte.



(a) Replikator Neuronales Netz, Kampagne K_2 (b) Replikator Neuronales Netz, Kampagne K_2 , Anomalie Score je Zeitpunkt

Bild 5.9: Vergleich Trainings- und Kontrolldaten anhand Anomalie Score je Zeitpunkt

Tabelle 5.3: Methodenvergleich anhand der AUC

Methode	K_1		K_2	
	AUC_{te}	AUC_{tr}	AUC_{te}	AUC_{tr}
IForest	0,5750	0,5379	0,6083	0,7945
ReplNN	0,7845	0,7857	0,7605	0,8570
SOM	0,3128	0,4079	0,7720	0,5493

Isolation Forest: Einfluss der Hyperparameter

Zum Trainieren des Isolation Forests müssen zwei Hyperparameter vom Benutzer definiert werden, die Anzahl der Bäume t und die Subsample-Größe ψ . Die Wahl der Parameter beeinflusst die Ergebnisse. Tabelle 5.4 vergleicht unterschiedliche Setups anhand der Mittelwerte von AUC_{tr} und AUC_{te} aus den Kampagnen K_1 und K_2 . Beginnend bei den in [LTZ12] vorgeschlagenen Standardwerten ($t = 100$, $\psi = 256$) werden die Parameter iterativ erhöht und es zeigt sich, dass der Schritt von $t = 100$ auf $t = 150$ bzw. $t = 200$ keine Veränderung bringt, womit zusätzliche Bäume nicht nötig sind. Eine Zunahme von Subsamples verbessert die Werte bis zu $\psi = 400$. Zusammenfassend

sind die verwendeten Parameter $\psi = 400$ und $t = 100$ für den Anwendungsfall ausreichend. Eine detaillierte Auflistung befindet sich im Anhang A.2.

Bedeutung der Datenmenge

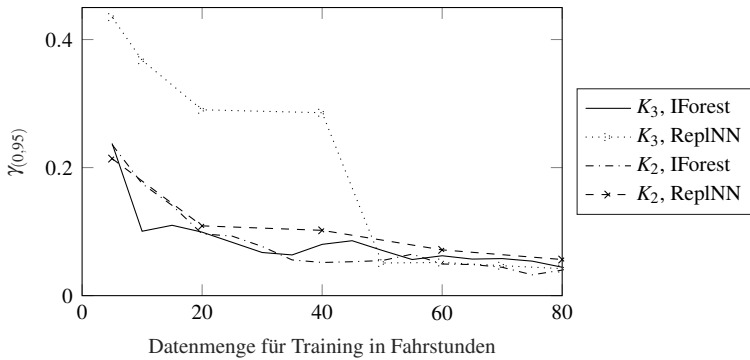
Die Bedeutung der Datenmenge wird in Bild 5.10 dargestellt. Es wird die Frage beantwortet, wie viele Daten zum Training nötig sind und wann das Referenzmodell in die Fahrzeuge übertragen werden kann. Bild 5.10 zeigt den Anteil erkannter Anomalien in den Kontrolldaten $\gamma_{(0,95)}$. Die Datenmenge wird in der Anzahl der aufgenommenen Fahrstunden dargestellt. Aufgrund der hohen Rechenzeit werden für das Replikator Neuronale Netz weniger Zusammensetzungen evaluiert. Dennoch erkennt man, dass der Anteil anormal erkannter Kontrolldaten $\gamma_{(0,95)}$ in beiden Kampagnen, unabhängig von der Methode, ab etwa 50 Stunden gegen den erwarteten Wert $\gamma_{(0,95)} = 0,05$ konvergiert; zusätzliche Daten ändern die Ergebnisse kaum. Die Untersuchung demonstriert, dass der Übergang von Trainings- zu Anwendungsphase aus Bild 3.3 festgelegt werden kann.

5.3.2 Bewertung der Anomalieerkennung

Im nächsten Schritt wird untersucht, welche der Anomalien, wie gut erkannt werden. Die folgenden Bilder 5.11 und 5.12 vergleichen die Anomalie Sco-

Tabelle 5.4: Vergleich der Hyperparameter für Isolation Forest, Mittelwert aus K_1 und K_2 und AUC_{tr} und AUC_{te} , detaillierte Auflistung im Anhang A.2

s	t		
	100	150	200
256	0,5992	0,6053	0,6252
400	0,6712	0,6343	0,6364
500	0,6205	0,6475	0,6208
700	0,6295	0,6349	0,6492

Bild 5.10: Einfluss der Trainingsdatenmenge auf $\gamma_{(0,95)}$

res, berechnet durch den Isolation Forest Ansatz bzw. das Replikator Neuronale Netz. Anhand τ_{q_x} mit $x = 1 - \xi_i$ wird dargestellt, wie gut sich die Anomalien von den normalen Trainingsdaten unterscheiden lassen (vgl. Formel 5.3), je größer der Anteil Trainingsdaten mit einem kleineren Anomalie Score als die Anomalie (x), desto besser die Trennbarkeit. Beispielhaft wird der Grenzwert τ_{q_x} mit $x = 0,99$ eingezeichnet. Die hervorgehobenen Anomalien lassen sich perfekt von den Trainings- und normalen Testdaten abgrenzen ($\xi_i = 0,0$). Die Bewertung der *False Positive Rate* erfolgt im darauffolgenden Schritt anhand von Bild 5.13.

Bild 5.11 zeigt, dass das Replikator Neuronale Netz in Kampagne K_1 alle Ereignisse, mit Ausnahme der ABS-Bremsungen, die kürzer als eine Sekunde dauern, von 99,8% der Trainingsdaten trennt. Der Isolation Forest hat mit einer der ABS-Eingriffe Probleme und insgesamt resultieren mehr Trainingsdaten mit höherem Score als die Anomalien. Dennoch sind beide Methoden fähig, die meisten dieser sehr außergewöhnlichen Fahrsituationen anhand des Anomalie Scores vom Großteil der Trainingsdaten zu trennen. Gerade die Unfälle können vom Replikator Neuronale Netz perfekt ($\xi_i = 0,0$) erkannt werden. Der folgende Abschnitt 5.3.4 zeigt auf, dass der

Grund hierfür ist, dass ein Signalwert mehr als das Doppelte von den Werten der Trainingsdaten abweicht.

Die Anomalien der zweiten Kampagne K_2 sind in Bild 5.12 aufgelistet. Auch hier erkennt man die bessere Performance des neuronalen Netzes. Achtzehn der achtundzwanzig Anomalien werden mit dem Grenzwert τ_{q_x} mit $x = 0,99$ erkannt. Lediglich die Ereignisse *Überdrehzahl*, *Handbremse während der Fahrt ziehen*, *Kreise fahren* und *nur Beifahrer angeschnallt* sind weniger eindeutig trennbar. Die *Überdrehzahl* verdeutlicht anhand des sinkenden Scores bei sinkender Anzahl Umdrehungen, je näher das Ereignis an den Flottendaten liegt, desto geringer ist der vom neuronalen Netz berechnete Anomalie Score. Der Isolation Forest hat generell Schwierigkeiten und ist für diese Testfälle nicht brauchbar.

Zur Bewertung der Anomalieerkennung wird in Bild 5.13 zusätzlich der Anteil fälschlicherweise anormal erkannter Testdaten in Abhängigkeit des Grenzwertes τ_{q_x} dargestellt; konkrete Werte listet Tabelle 5.5. Es wird deutlich, dass das Replikator Neuronale Netz in beiden Kampagnen eine geringere *False Positive Rate* als der Isolation Forest erzielt. In allen Fällen wird bei ausschließlich normal angenommenen Trainingsdaten ($x = 1,0$) kein normaler Datenpunkt als anormal eingestuft. Allerdings beschränken sich dann die gefunden Anomalien auf die Ereignisse *Unfall* und *Fensterheber mit eingeklemmtem Buch*; in den Bildern 5.11 und 5.12 durch die Hervorhebung gekennzeichnet. Eine Auflockerung der Grenze auf $x = 0,99$ erhöht die Anzahl erkannter Anomalien wesentlich, während die *False Positive Rate* für das Replikator Neuronale Netz nicht steigt. Zur Erfassung weiterer Anomalien (*Vollbremsung*, *Überdrehzahl*, *nur Beifahrer angeschnallt*) zeigt Tabelle 5.5, dass die Lockerung des Grenzwertes τ_{q_x} auf $x = 0,9$ zu mehr als 7% fälschlicherweise anormal eingestuftener Ereignisse führt (für ReplNN).

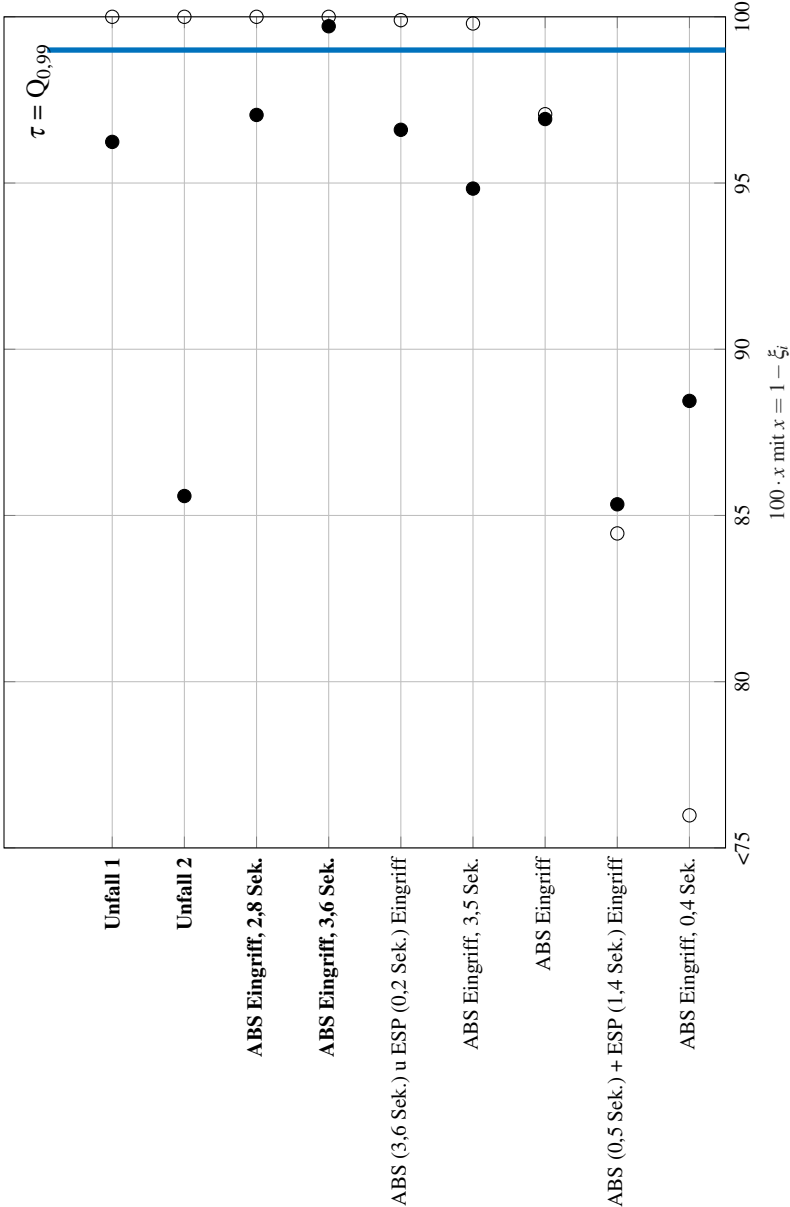
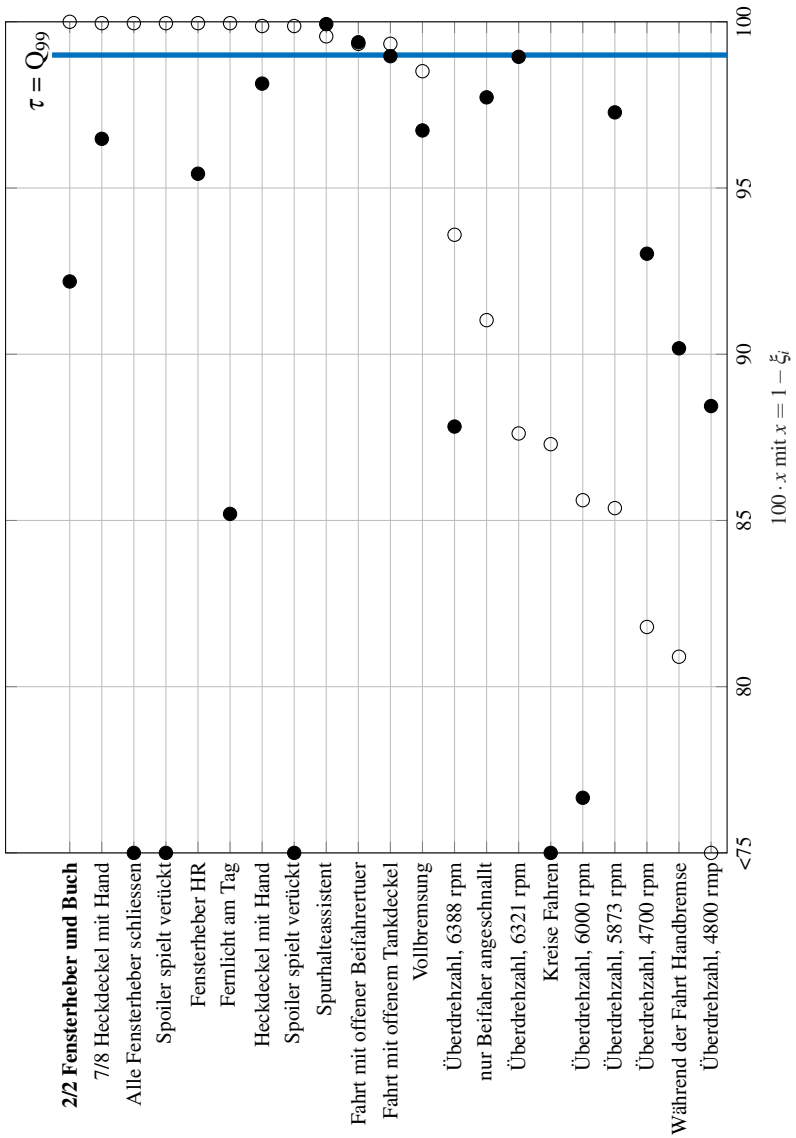


Bild 5.11: Vergleich ReplNN (○) und IForest (●), K_1

Bild 5.12: Vergleich RepINN (○) und IForest (●), K_2

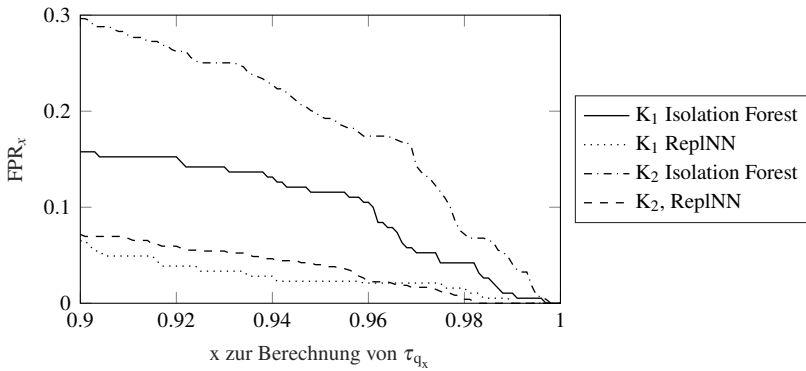


Bild 5.13: FPR_x für die verschiedenen Methoden und Kampagnen in Abhängigkeit von τ_{q_x}

Tabelle 5.5: FPR_x in Abhängigkeit von Grenzwert τ_{q_x}

x	IForest		ReplNN	
	K_1	K_2	K_1	K_2
0,0	0,0	0,0	0,0	0,0
0,01	0,0105	0,0408	0,0	0,0
0,05	0,1156	0,1945	0,0229	0,0402
0,1	0,1577	0,2965	0,0650	0,0716

Die Ergebnisse lassen sich für den gegebenen Datensatz wie folgt zusammenfassen.

- Alle getesteten Verfahren sind zur Reduzierung des Datentransfers von Fahrzeug zu Backend geeignet. Die Aufteilung in Trainings- und Kontrolldaten führt zu folgendem Ergebnis: Bei angenommen 1% anormaler Trainingsdaten ($x = 0,01$ für τ_{q_x} : entspricht dem Ziel einer Datenreduktion um Faktor 100), ist eine durchschnittliche Reduktion der Kontrolldaten auf 0,595% (Isolation Forest) bzw. 0,715% (Replikator Neuronales Netz) möglich (vgl. Tabelle 5.2: Mittelwert aus K_1 und K_2). Das bedeutet für die 2,1 MByte, die ein Oberklassefahrzeug

pro Sekunde generiert, eine geminderte Übertragungsrate von 12,5 bzw. 15 KByte/s.

- Das Replikator Neuronale Netz erkennt bei 1% anormal angenommener Trainingsdaten (= Datenreduktion um Faktor 100) die Mehrheit der real eingefahrenen Anomalien während kein Ereignis fälschlicherweise als anormal eingestuft wird. Die Ereignisse *ABS Eingriff unter einer Sekunde, Überdrehzahl, Handbremse während der Fahrt, Kreise fahren* und *nur Beifahrer angeschnallt* werden nicht gefunden.
- Die Zusammenfassung der Anomalie Scores zu Zeitabschnitten (Minuten) nach Formel 5.5 hat den Vorteil, dass Rauschen in den Daten ignoriert wird und somit der Datentransfer weiter reduziert werden kann. Außerdem wird sichergestellt, dass der Kontext indem die Anomalie auftritt nicht verloren geht.
- Die Versuche zeigen, dass das Replikator Neuronale Netz dem Isolation Forest Ansatz und dem Clustering mit SOMs vorzuziehen ist. Die real eingefahren Anomalien können besser von den Trainingsdaten unterschieden werden und es erzielt eine geringere *False Positive Rate*.
- Sicherheitskritische Situationen (Unfälle, ABS- und ESP-Eingriffe) können sowohl vom Isolation Forest als auch vom Replikator Neuronalen Netz erkannt werden.
- Die außergewöhnlichsten und definitiv nicht in den Flottendaten vorkommenden Ereignisse *Unfall* und *Fensterheber mit eingeklemmtem Buch* sind mithilfe des Replikator Neuronales Netzes eindeutig von den Trainingsdaten unterscheidbar. Bei dem Ziel der Datenreduzierung auf 0%, sind dies die Ereignisse, die übertragen werden.
- Sollen weitere Anomalien erkannt werden (z.B. nur Beifahrer angeschnallt), führt die Herabstufung des Grenzwertes auf $x = 0,9$ für τ_{q_x}

zu einer *False Positive Rate* von 6,8%. Für ein Oberklassefahrzeug (2,1 MByte/s) ergibt sich hierbei ein Volumen von 142,8 KByte/s, die die Experten unnötigerweise (da normal) bewerten müssten. Ein solches Vorgehen ist nicht realisierbar, weswegen weitere Verarbeitungsschritte (z.B. Optimierung, Zusammenfassung gleicher Anomalien) nötig sind.

5.3.3 Synthetische Testfälle

Im vorherigen Abschnitt 5.3.2 werden hauptsächlich reale Anomalien erkannt, bei denen es sich um punktuelle Ausreißer handelt, da ein bzw. mehrere Signalwerte weit genug vom Rest der Daten entfernt sind. Außerdem vertreten die Testfälle nur einzelne wenige Zeitpunkte. Um die Methoden anhand einer größeren Datenbasis zu bewerten, werden daher synthetische Anomalien generiert. Durch das zufällige, mehrfache Extrahieren und Manipulieren normaler Fahrsituationen aus den Flottendaten (vgl. Bild 5.14) entstehen je Anomalietyp hundert verschiedene Ausprägungen, weswegen als Vergleichskriterien AUC_{te} und AUC_{tr} aus Abschnitt 5.2 verwendet werden. Neben kontextuellen Anomalien legen die folgenden Abschnitte weitere globale Ausreißer und gestörte Signalzusammenhänge dar. Die Testfälle können nicht als real angenommen bzw. übernommen werden, denn zur Generierung sind nicht alle Zusammenhänge des Gesamtsystems bekannt. Es handelt sich um synthetische Testfälle, die dazu dienen, die Anwendbarkeit, aber auch die Grenzen der Methodik aufzuzeigen.

Gestörte Signalzusammenhänge

Durch die Manipulation korrelierender Signale wird zunächst analysiert, wie das Verfahren mit gestörten Signalzusammenhängen umgeht. Die folgenden Anomalien werden generiert:

1. Eine Beziehung wird geschwächt, dadurch dass die Geschwindigkeitsanzeige ein dreifaches bzw. ein Drittel der tatsächlichen Ge-

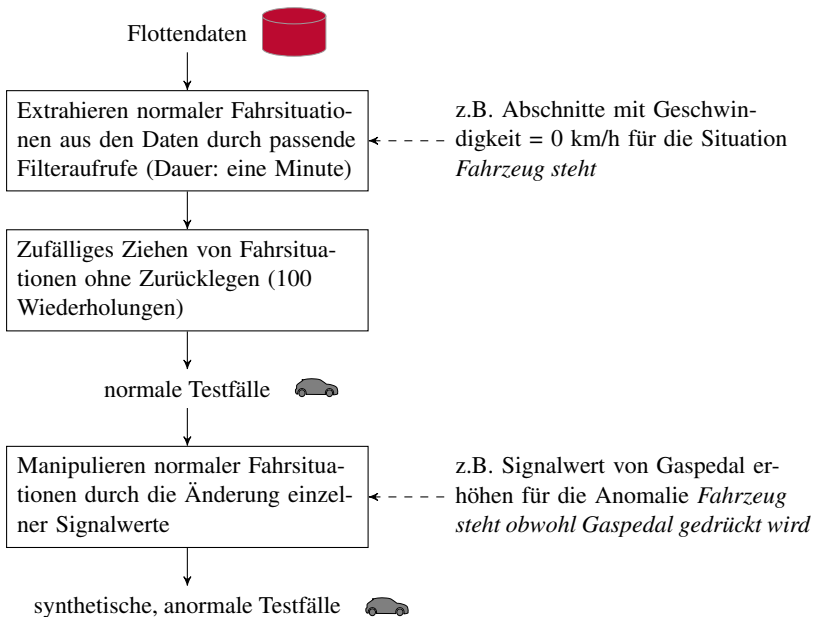


Bild 5.14: Synthetische Anomalien durch Extrahieren und Manipulieren normaler Fahrsituationen

schwindigkeit, oder aber null während das Fahrzeug fährt, anzeigt (**defekte Geschwindigkeitsanzeige**). Die Testfälle werden, wie in Bild 5.14 darstellt, durch hundertfaches Extrahieren verschiedener Abschnitte konstanter Fahrt (mindestens eine Minute mit maximaler Geschwindigkeitsdifferenz von 3 km/h) und das Manipulieren des Signalwertes der Geschwindigkeitsanzeige hervorgebracht.

2. In gleicher Weise wird ein anomales Verhältnis zwischen Öl- und Kühlmitteltemperatur (vgl. Bild 5.15) als weiterer Testfall erzeugt. Abschnitte mit einer Kühlmitteltemperatur größer als 100°C werden

- extrahiert und die Öltemperatur wird durch das 10% Quantil der Flottendaten ersetzt (**Öltemperatur herabgesetzt**).
3. Zusätzlich werden Abschnitte mit einer Kühlmitteltemperatur geringer als 49°C extrahiert und die Öltemperatur manipuliert, indem es durch das 90% Quantil der Flottendaten ersetzt wird (**Öltemperatur erhöht**).
 4. Weiteres Beispiel ist die **aktive Spiegelheizung bei einer Außentemperatur von über 30° C**.
 5. Es wird die Anomalie **aktive Lichter bei inaktiver Batterie** generiert. Dazu werden aus den Flottendaten Abschnitte mit inaktiver Batterie gezogen und die Lichter auf *ein* gesetzt. Hierbei ist zwischen zwei Testfällen zu unterscheiden: Einmal werden alle Lichter aktiviert, während das zweite Beispiel das Fernlicht auslöst.
 6. Konstante Fahrten schneller als 160 km/h werden manipuliert, indem Lenkradwinkel und Lenkradgeschwindigkeit außergewöhnlich hohe Werte (110° und 200°/s) erhalten (**anormales Lenkrad bei schneller Fahrt**).
 7. Zusätzlich wird mithilfe der realen Testfälle die Anomalie **Fahrt mit offener Tür** kreiert. Dazu werden die real eingefahrenen Anomalien extrahiert, hundertmal kopiert und die Signale, die stärker mit der Geschwindigkeit als mit der Beifahrertür korrelieren, geändert. Deren Signalwerte werden durch die normaler Fahrten ersetzt. Entsprechend wird der Normalzustand **Stand mit offenerer Beifahrertür** erzeugt.
 8. In gleicher Weise entsteht das anormale Verhalten eines offenen Heckdeckels bei konstanter Fahrt (**Fahrt mit geöffnetem Heckdeckel**). Der normale Gegenspieler ist hierbei der **offene Heckdeckel während das Fahrzeug steht**.

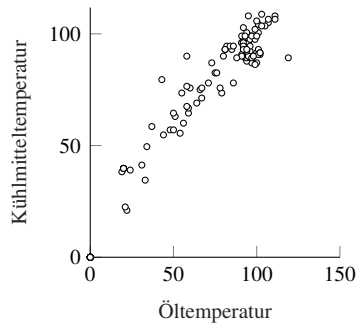


Bild 5.15: Zusammenhang Öl- und Kühlmitteltemperatur

Tabelle 5.6 zeigt die Ergebnisse und vergleicht anhand der Fläche unter der ROC-Kurve (AUC), wie gut sich die anomalen Fälle und deren korrespondierenden Normalzustände von Trainings- und Testdaten unterscheiden lassen. Zusätzlich wird für die in Tabelle 5.5 gelisteten Grenzwerte der Anteil erkannter Anomalien berechnet. Eine detaillierte Aufzählung für das Replikatoren Neuronale Netz und den Isolation Forest findet sich im Anhang in Tabelle A.3. Dort wird bestätigt, dass auch in diesem Fall das Replikatoren Neuronale Netz mit einer durchschnittlichen $AUC = 0,93$ dem Isolation Forest ($AUC = 0,75$) vorzuziehen ist.

Außerdem veranschaulicht die Tabelle 5.6, dass die manipulierten Zusammenhänge einen höheren Anomalie Score als die entsprechende normale Situation haben, doch lassen sie sich nicht eindeutig von Trainings- und Testdaten unterscheiden. Eine AUC_{lr} von 0,9351 für die herabgesetzte Öltemperatur würde beispielsweise bedeuten, dass bei einem Grenzwert τ_{q_x} mit $x = 0,99$ nur 7,5% der Fälle erkannt werden. Ein Grenzwert τ_{q_x} mit $x = 0,9$ wiederum macht die Detektion von 67,5% der Ereignisse möglich, was zu einer *False Positive Rate* von durchschnittlich 6,8% führt (vgl. Tabelle 5.5). Besser erkannt werden können die manipulierten Geschwindigkeitsanzeigen und die Fahrten mit geöffnetem Heckdeckel bzw. geöffneter

Tabelle 5.6: Vergleich synthetischer Anomalien (grau markiert) mit normalen Gegenspielern anhand der AUC

	AUC		Anteil erkannter Anomalien mit Grenzwert				
	AUC_{tr}	AUC_{re}	$\tau_{q_{0,90}}$	$\tau_{q_{0,95}}$	$\tau_{q_{0,99}}$	$\tau_{q_{1,0}}$	
1.	Fahrt schneller 250 km/h, normal	0,4970	0,3992	0,0000	0,0000	0,0000	0,0000
	angezeigte Geschwindigkeit*3	0,9950	0,9938	0,9800	0,9550	0,9550	0,8700
	angezeigte Geschwindigkeit/3	0,9717	0,9637	0,9550	0,9550	0,9550	0,8700
	angezeigte Geschwindigkeit = 0	0,9797	0,9746	0,9550	0,9550	0,9550	0,8700
2.	Kühlmitteltemperatur >100 °C, normal	0,3532	0,2809	0,0000	0,0000	0,0000	0,0000
	Öltemperatur <10% Quantil, normal	0,6731	0,6079	0,0450	0,0000	0,0000	0,0000
	Öltemperatur herabgesetzt, manipuliert	0,9351	0,9473	0,6750	0,5000	0,0750	0,0000
3.	Kühlmittel <49°C, normal	0,6574	0,5913	0,0450	0,0000	0,0000	0,0000
	Öltemperatur >90% Quantil, normal	0,3491	0,2902	0,0000	0,0000	0,0000	0,0000
	Öltemperatur erhöht, manipuliert	0,9200	0,9261	0,5745	0,5226	0,0250	0,0000
4.	Spiegelheizung bei 30 °C	0,4155	0,2319	0,0000	0,0000	0,0000	0,0000
5.	inaktive Batterie, normal	0,2294	0,1145	0,0000	0,0000	0,0000	0,0000
	aktive Lichter bei inaktiver Batterie	0,9996	0,9935	1,0000	1,0000	1,0000	1,0000
	aktive Lichter bei inaktiver Batterie, ohne Fernlicht	0,5489	0,3233	0,0000	0,0000	0,0000	0,0000
6.	Fahrt schneller 160 km/h, normal	0,4452	0,3589	0,0000	0,0000	0,0000	0,0000
	anormales Lenkrad bei Fahrt schneller 160 km/h	0,9585	0,9668	1,0000	0,5150	0,0300	0,0300
7.	Stand mit geöffneter Tür, normal	0,7734	0,6864	0,1800	0,1500	0,0500	0,0000
	Fahrt mit geschlossener Tür, normal	0,5558	0,4440	0,1500	0,0800	0,0100	0,0000
	Fahrt mit offener Tür, manipuliert	0,9656	0,9737	1,0000	1,0000	1,0000	0,0000
8.	Stand mit geöffnetem HD, normal	0,9854	0,9935	1,0000	1,0000	1,0000	0,0000
	Fahrt mit geöffnetem HD, manipuliert	0,9816	0,9915	1,0000	1,0000	1,0000	0,0000

Beifahrertür. Hier werden bei 99% normal angenommener Trainingsdaten durchschnittlich 97,3% der anormalen Ereignisse bestimmt und die *False Positive Rate* in den normalen Testdaten liegt bei 0% (vgl. Tabelle 5.5).

Tabelle 5.6 verdeutlicht, dass der Anomalie Score zwischen einer geöffneten Tür bei stehendem und fahrendem Fahrzeug unterscheidet. In den Trainingsdaten ist die Tür in 7,4% der Standzeit und in 1,7% der Fahrzeit (> 10 km/h) geöffnet, was ausreicht, um zwischen diesen Fällen differenzieren zu können. Allerdings wird von den normal definierten Ereignissen der *geöffnete Heckdeckel bei Stand* als anormal erkannt, was bedeutet, ein geöffneter Heckdeckel gilt prinzipiell als außergewöhnlich. Laut Definition ist das richtig, da das Ereignis in Kampagne K_2 in nur 0,006% der Flottendaten eintritt, einen wirklichen Mehrwert für den Automobilhersteller stellt es jedoch nicht direkt dar. Somit werden seltene, aber nicht als ungewöhnlich angenommene Vorkommnisse als anormal erkannt, wodurch Nutzen für das Unternehmen erst mit längerfristigen Betrachtungen (z.B. sind die Heckdeckel bestimmter Fahrzeuge häufig nicht richtig geschlossen) entsteht. Ähnlich verhält es sich mit dem aktiven Fernlicht. Es taucht in den Trainingsdaten nur einmal auf, was der Grund ist, warum die Anomalie *aktive Lichter bei inaktiver Batterie* erkannt wird.

Diese Beschaffenheit des neuronalen Netzes wird in Bild 5.16 sichtbar. Sie zeigt in Abhängigkeit der Kurtosis eines jeden Signals an, wie gut es sich von den Trainingsdaten unterscheiden lässt (AUC_{tr}), wenn es auf den möglichen Maximalwert gesetzt wird. Eine hohe Kurtosis beschreibt eine Verteilung mit starken Peaks und hoher Anfälligkeit für Ausreißer [Str13], wie es für den Heckdeckel oder das Fernlicht der Fall ist. Es wird deutlich, dass Signale mit hoher Kurtosis allein durch den für sie außergewöhnlich hohen Wert zur Detektion als Anomalien führen und somit das Referenzmodell beeinflussen.

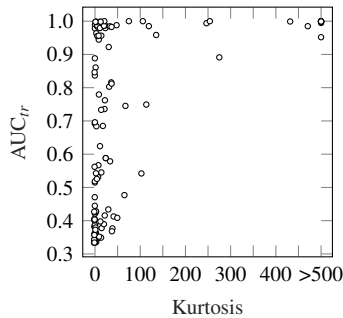


Bild 5.16: Trennbarkeit vom Maximum je Signal von den Trainingsdaten anhand AUC_{tr} in Abhängigkeit der Kurtosis

Kontextuelle Anomalien

Im Folgenden verlagert sich der Fokus auf kontextuelle Anomalien, Ereignisse, die nur innerhalb einer bestimmten Gegebenheit als anormal gelten (vgl. Kapitel 2.2). Hierzu werden die folgenden Kontexte aus den Flottendaten extrahiert:

- Konstant fahrendes Fahrzeug (*konst. Fahrt*); das Fahrzeug fährt für mindestens eine Minute konstant mit einer maximalen Geschwindigkeitsdifferenz von 3 km/h.
- Anfahrendes Fahrzeug (*Anfahrt*); das Fahrzeug beschleunigt mindestens zwei Sekunden, sodass sich die Geschwindigkeit erhöht.
- Stehendes Fahrzeug (*Stand*); das Fahrzeug steht mindestens eine Minute bei einer Geschwindigkeit von 0 km/h.
- Abbremsendes Fahrzeug (*Abbremsen*); das Fahrzeug bremst mindestens zwei Sekunden ab, sodass sich die Geschwindigkeit verringert.

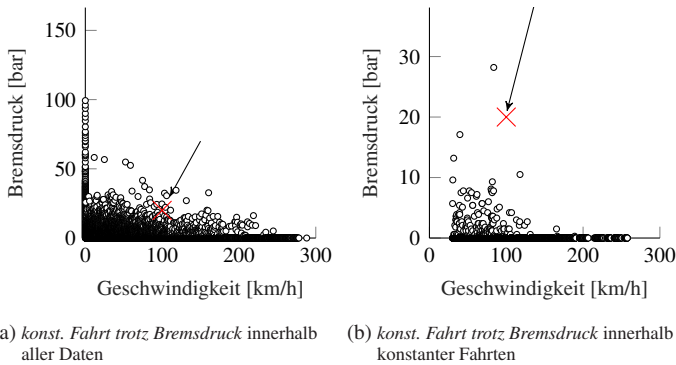
Es werden Anomalien generiert, die auf eine Manipulation hindeuten, da der Fahrer durch seine Aktionen das Fahrzeug nicht beeinflussen kann, und

die nur innerhalb des entsprechenden Kontextes anormal sind. Durch wiederholtes Extrahieren und Manipulieren (vgl. Bild 5.14) entstehen jeweils hundert Testfälle:

1. **konst. Fahrt trotz Bremsdruck:** Das Fahrzeug fährt konstant, obwohl der Bremsdruck steigt und das Gaspedal auf 0% steht. Der Bremsdruck steigt innerhalb einer Minute linear von null bar auf das 95% Quantil der Flottendaten. Die Wahl der oberen Grenze fällt auf das Quantil, um zu verhindern, dass die Anomalie aufgrund eines besonders hohen Bremsdruckes entsteht.
2. **Anfahrt trotz Bremsdruck:** Das Fahrzeug fährt an, obwohl der Bremsdruck steigt und das Gaspedal auf 0% steht. Der Bremsdruck steigt innerhalb der Anfahrt linear von null bar auf das 95% Quantil der Flottendaten.
3. **Stand trotz Gaspedal:** Das Fahrzeug bleibt stehen, obwohl das Gaspedal gedrückt wird. Das Pedal steigt innerhalb einer Minute linear von null Prozent auf das 95% Quantil der Trainingsdaten.
4. **Abbremsen trotz Gaspedal:** Das Fahrzeug bremst ab, obwohl das Gaspedal gedrückt wird. Das Pedal steigt innerhalb einer Minute linear von null Prozent auf das 95% Quantil der Trainingsdaten.

Bild 5.17a zeigt beispielhaft, dass ein Bremsdruck von 20 bar bei einer Geschwindigkeit von 100 km/h keine Seltenheit ist. Beschränkt man die Daten allerdings nur auf Abschnitte, in denen sich die Geschwindigkeit um weniger als 3 km/h ändert, zeigt Bild 5.17b, dass der Bremsdruck in dieser Situation anormal ist.

Die Wahl der vier Kontexte *konst. Fahrt*, *Anfahrt*, *Stand* und *Abbremsen* ist eine Variante, die Fahrsituationen zu unterscheiden [HVBL17]. Werden sie als Klassen definiert und ein Klassifikator zur Vorhersage dieser trainiert, liefert die zehnfach Kreuzvalidierung (vgl. Kapitel 2.2) eines Random Fo-

Bild 5.17: Anomalie *konstante Fahrt trotz Bremsdruck* in unterschiedlichen Kontexten

rests mit zehn Bäumen eine Vorhersagegenauigkeit von 98,33% in Kampagne 1 bzw. 97,61% in Kampagne 2. Die hohen Werte führen zu dem Schluss, dass die Kontexte prinzipiell anhand der Signalwerte unterschieden werden können und somit die Chance besteht, dass auch das Verfahren zur Anomalieerkennung diese Unterscheidung treffen kann.

Tabelle 5.7 listet die Ergebnisse und zeigt anhand der Fläche unter der ROC-Kurve, wie sie sich verändern, wenn a) die Trainings- und Testdaten so gefiltert werden, dass sie lediglich den entsprechenden Kontext beschreiben und b) eine Signalauswahl erfolgt. Hierfür wird auf die für Fahrmanöver relevanten Signale *Geschwindigkeit*, *Bremsdruck*, *Fahrpedalstellung*, *Motordrehzahl*, *Längs- und Querschleunigung* und *Lenkradwinkel* reduziert [FMS⁺17, HVBL17]. Die Tabelle 5.7 zeigt anhand der Beispiele, dass das Replikator Neuronale Netz sowohl durch die Signal- als auch durch die Datenauswahl verbessert werden kann. Ohne diese Schritte ist die AUC_{tr} (vgl. Bild 5.5) von maximal 0,5312 kaum besser als der Zufall und daher nicht ausreichend, um die Anomalien von den Trainingsdaten zu unterscheiden. Bild 5.18 visualisiert die ROC-Kurve und bestätigt, dass mehr als 20% der Trainingsdaten als anomal eingestuft werden, um in Kampagne K_1 mindestens eine der Anomalien vom Typ **konst. Fahrt trotz Bremsdruck** zu

erkennen. Da es in dieser Untersuchung um die Unterscheidung der Anomalien von den *normalen* Trainingsdaten geht, entspricht der Anteil anomaler Trainingsdaten der FPR (vgl. Abschnitt 5.2).

Bewertung der synthetischen Testfälle

Zusammengefasst sind die synthetischen Testfälle Beispiele, die einen höheren Anomalie Score als ihre normalen Gegenspieler aufweisen. Sie verdeutlichen aber auch die Grenzen der Methode: Es werden Anomalien erkannt, die per Definition selten sind, dem Automobilhersteller aber keinen direkten Mehrwert liefern. Prinzipiell ist ein geöffneter Heckdeckel während das Fahrzeug steht nichts Außergewöhnliches. Erst wiederholtes Auftreten und längerfristige Beobachtungen können verwendet werden. Ein geöffneter Heckdeckel bedeutet dann beispielsweise, dass dieser für bestimmte Fahrzeuge nicht voll funktionsfähig ist, oder dass einem Kleinwagenbesitzer gegebenenfalls ein größeres Fahrzeug angeboten werden kann. Daher ist die Anomalieerkennung nur mit der Weiterverarbeitung im Backend hilfreich, wo sie einsortiert und womöglich entkräftet werden. Zusätzlich

Tabelle 5.7: Trennbarkeit der Anomalie von Trainings- und Testdaten in Abhängigkeit von Daten- und Signalauswahl, dazu Mittelwert aus K_1 und K_2

		alle Signale		Signalauswahl	
		AUC_{te}	AUC_{tr}	AUC_{te}	AUC_{tr}
1. konst. Fahrt trotz Bremsdruck	alle Daten	0,7815	0,5312	0,9250	0,6971
	nur Kontext	1,0000	0,9984	1,0000	1,0000
2. Anfahrt trotz Bremsdruck	alle Daten	0,5515	0,4359	0,6692	0,6062
	nur Kontext	0,8834	0,8586	0,8168	0,8181
3. Stand trotz Gaspedal	alle Daten	0,6851	0,5290	0,9840	0,7287
	nur Kontext	0,9760	0,9460	0,9931	0,9851
4. Abbremsen trotz Gaspedal	alle Daten	0,5016	0,4357	0,7291	0,6116
	nur Kontext	0,8961	0,8847	0,8210	0,8131

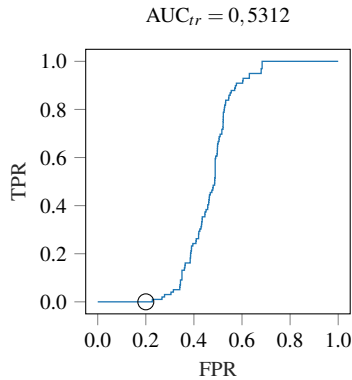


Bild 5.18: ROC-Kurve zu Testfällen *konst. Fahrt trotz Bremsdruck, Kampagne K₃*

besteht die Option, mehr Trainingsdaten zu sammeln, damit die seltenen aber normalen Ereignisse häufiger auftauchen. Zudem zeigen die Testfälle, dass die Anomalien sich besser von den Trainingsdaten unterscheiden lassen als ihre normalen Gegenspieler, eine perfekte Trennung ist allerdings nicht gegeben. Sollen diese erkannt werden, führt eine Herabsetzung des Schwellwertes demnach zu einer erhöhten Rate fälschlicherweise erkannter Ereignisse.

Ohne erweiterte Vorverarbeitungsschritte scheitert das Verfahren bei der Erkennung von kontextuellen Anomalien. Das Trainieren mehrerer Referenzmodelle für unterschiedliche Kontexte verbessert die Ergebnisse. Zur Realisierung ordnet ein Klassifikator hierzu die Datenpunkte als vorbereitender Schritt in die entsprechende Kategorie ein, woraufhin das passende Referenzmodell verwendet werden kann. Eine solche Vorverarbeitung verfeinert die Methode, wodurch auch die Schwierigkeiten mit seltenen Ereignissen eingegrenzt werden können (z.B. Anteil geöffneter Heckdeckel während Stand: 0,02%, Anteil offener Heckdeckel während Fahrt: 0%).

5.3.4 Beispielhaftes Nachvollziehen erkannter Anomalien

Kapitel 3.1 zeigt, dass das Erkennen einer Anomalie nicht ausreicht. Im Postprocessing muss auch der Grund, warum die Situation als außergewöhnlich gilt, herausgefunden werden. Die Interpretation erfolgt auf dem Backend und bedarf manueller Beurteilungen. Der erste Schritt ist die Identifikation der anormal auffallenden Signale. Dazu wird das in Kapitel 4.2.3 vorgestellte Random Forest Verfahren für die Daten aus Tabelle 5.1 angewandt. Die erkannte Anomalie repräsentiert hierbei die Klasse *anormal*, während die Flottendaten die Klasse *normal* beschreiben. Daraus werden 250 Entscheidungsbäume trainiert. Anhand der generierten Bäume kann die Relevanz jedes Signals berechnet werden. Die Signale mit besonders hoher Wichtigkeit sind entscheidend zur Trennung zwischen den Klassen. Tabellen A.4 und A.5 im Anhang zeigen je Anomalie, welche durch das Random Forest Verfahren als wichtig erkannt werden. In diesem Kapitel soll der Vorgang beispielhaft für die Anomalien **Unfall** und **Schließen des Fensters mit eingeklemmtem Buch** gezeigt werden. Es handelt sich um Testfälle, weswegen bekannt ist, was genau passiert ist. Es wird überprüft, ob die erkannten Anomalien mit den gegebenen Ereignissen in Verbindung gebracht werden können.

Zwar ist die Identifikation von Unfällen keine große Herausforderung, Fahrerassistenzsysteme wie die Notfallbremse oder das sogenannte *PreSave*-System dienen bereits zur Vermeidung von Unfällen bzw. zur Verringerung ihrer Schwere [HKBL11, Mer18], allerdings soll der Abschnitt zur Verdeutlichung des Weges von *erkannter Anomalie* zu *interpretierter Situation* dienen. Wie in Kapitel 4.2.3 geschildert, kann ein Experte bei der Interpretation unterstützt werden, indem er auf die auffälligen Signale hingewiesen wird. Da ein Unfall eine für jedermann nachvollziehbare Situation ist, wird dieser als Beispiel verwendet. Die Analyse des Experten kann nachempfunden werden. Auch das *Schließen des Fensters mit eingeklemmtem Buch* ist ein Ereignis, das den Schritt der Interpretation greifbar macht.

Die Minute des Unfalls wird direkt erkannt, aber die Schlussfolgerung, dass es sich um einen Unfall handelt, kann nicht ohne Weiteres gemacht werden. In Kampagne K_1 sind 36 unterschiedliche Signale gegeben. Das heißt, die Komponenten, die ausschlaggebend sind, dass der Datenpunkt als Anomalie klassifiziert wird, müssen automatisiert identifiziert werden.

Die fünf Signale, die das Random Forest Verfahren als am wichtigsten einstuft, sind in Tabelle 5.8 dargestellt. Die berechnete *Feature Relevanz* ist normiert auf das Maximum.

Die Tabelle zeigt die festgestellten Signale: *Winkelgeschwindigkeit um die z-Achse*, *Bremsdruck*, *Längsbeschleunigung*, *Beschleunigung in x-Richtung* und die *Querb beschleunigung*. Deren Werte werden in Bild 5.19 gezeigt. Die Information, dass zu einem Zeitpunkt die Winkelgeschwindigkeit um die z-Achse des Fahrzeugs (vgl. Bild 5.2) bei 180 Grad/s liegt, ermöglicht die Interpretation, dass sich das Fahrzeug gedreht hat. Eine Drehung lässt auf einen Unfall schließen, womit dieser nicht nur als anormal erkannt, sondern auch als solcher interpretiert werden kann.

Als weiteres Beispiel wird das **Schließen des Fensters mit eingeklemmtem Buch** in Tabelle 5.9 verdeutlicht. Das Verfahren schlussfolgert, dass der Grund der Anomalie an der Fensteröffnung des Beifahrers liegen muss, da es das Signal mit höchster Relevanz ist. Der Verlauf von Signal *Beifahrer Fensterheber-Schalter automatisch hoch* wird in Bild 5.20 gezeigt. Innerhalb von nur dreißig Sekunden ändert sich der Status auf *Fenster hoch*

Tabelle 5.8: Interpretation der Anomalie *Unfall 2* durch Random Forest

	Feature Relevanz
Winkelgeschwindigkeit um die z-Achse	1,000
Bremsdruck	0,615
Längsbeschleunigung	0,442
Beschleunigung in x-Richtung	0,398
Querb beschleunigung	0,336

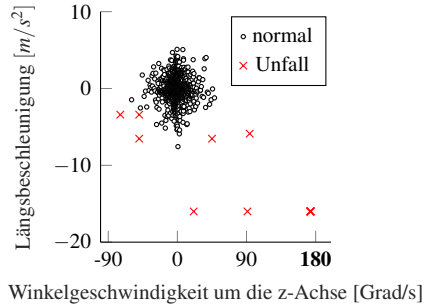


Bild 5.19: Scatterplot der Signale mit höchster Wichtigkeit, Unfall

= 1 elf mal. Die Tatsache, dass ein Buch eingeklemmt ist, kann daraus zu- gegeben nicht gelesen werden, aber ein ungewöhnlich hoch frequentierter Wechsel der Schaltung wird erkannt.

Das Random Forest Verfahren zur Nachvollziehbarkeit erkannter Anomali- en wird in diesem Kapitel exemplarisch anhand von zwei Ereignissen durch- geführt. Die Signale, die maßgebend zur Trennung der Anomalie vom Rest der Daten sind, werden herausgearbeitet. Allerdings kann durch zufällige Zusammenhänge das Ergebnis verfälscht werden. Zudem liefern die wich- tigsten Signale lediglich ein Indiz dafür was passiert sein könnte und ein Interpretationsspielraum ist gegeben. Es handelt sich somit nur um einen ersten Schritt in Richtung *Interpretation*.

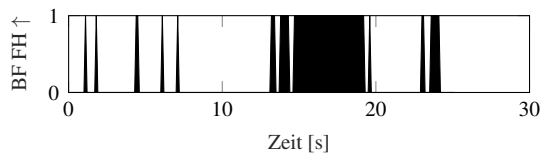
Bild 5.20: Verlauf von Signal *Status des Beifahrer Fensterhebers (automatisch hoch)*

Tabelle 5.9: Interpretation der Anomalie *Fenster schließen mit eingeklemmtem Buch*

	Feature Relevanz
BT FH automatisch hoch	1,000
BT FH Öffnung	0,416
Kilometerstand	0,189
BT FH manuell hoch	0,181
Verlustmoment	0,159

Insgesamt zeigen die Versuche, dass das Replikator Neuronales Netz die realen Testfälle am besten erkennt, ohne normale Ereignisse fälschlicherweise als anomal zu klassifizieren. Insbesondere sicherheitsrelevante Situationen (z.B. Unfall) lassen sich perfekt vom Rest der Daten trennen. Die synthetischen Testfälle verdeutlichen, dass der Fokus des neuronalen Netzes auf globalen Ausreißern liegt und somit kontextuelle Anomalien übersehen werden. Außerdem bedeutet das auch, dass Ereignisse erkannt werden, die lediglich selten auf dem Backend sind, aber nicht anomal im Sinne des Entwicklers. Erst durch das Nachvollziehen der ausschlaggebenden Signale auf dem Backend wird die Anomalie bewertet.

6 Anwendungsfall: Erkennung von Getriebeschäden

Ein konkretes Szenario, in dem die Kommunikationsdaten als Zeitreihen betrachtet werden, ist die Erkennung von Getriebeschäden. Ziel ist es, ungewöhnliches Verhalten vorzeitig zu erkennen, sodass ein möglicher Ausfall vorhergesagt und vermieden werden kann. Hierbei beschränkt sich die Analyse auf lediglich ein Signal und die Grundwahrheit ist je Anfahrtsvorgang gegeben. Der Anwendungsfall ist gelöst von den Untersuchungen in Kapitel 5, vielmehr sollen die Methoden zur Anomalieerkennung in Zeitreihen aus Kapitel 4.1.2 analysiert werden. Das anormale Verhalten ist bekannt, was bedeutet, das Problem ist a-priori definiert. Trotzdem dient der Ansatz zur Evaluierung der *one-class* Klassifizierung, da geprüft wird, ob ein Getriebeschaden als Abweichung vom gelernten, normalen Verhalten identifiziert werden kann.

Innerhalb der Anwendung wird zwischen *Anfahrt* und *Tour* unterschieden. Eine *Anfahrt* meint die Situationen, in denen ein Fahrzeug anfährt, wie beispielsweise im Stau oder an der Ampel. Eine komplette Fahrt von Einsteigen und Motor starten bis Motor ausschalten und Aussteigen wird als *Tour* bezeichnet. Schäden im Getriebe, die zu einem möglichen Ausfall führen können, werden durch die *Drehzahl der Getriebewelle* während einer *Anfahrt* beschrieben. Es handelt sich um hochfrequente Amplitudensprünge, die entstehen, da die Welle nicht greift. Bild 6.1 vergleicht die erste Sekunde einer *Anfahrt* mit und ohne Probleme im Getriebe. Das Verhalten ist den Experten bekannt und die Betrachtung des Signalverlaufs ermöglicht die manuelle Einordnung in *normal* oder *anormal*.

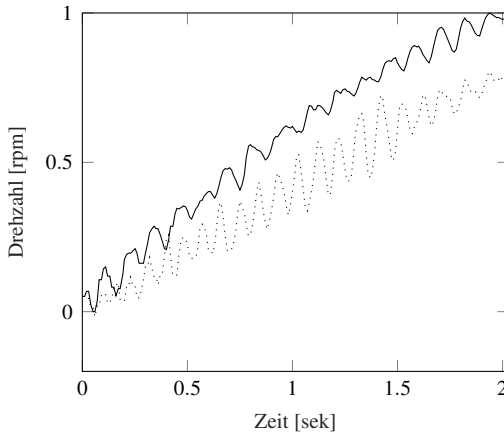


Bild 6.1: Unterschied zwischen normaler Anfahrt (durchgezogen) und anormaler Anfahrt (gepunktet)

Es wurden Daten mit zwei unterschiedlichen Versuchsträgern eingefahren, wovon ein Fahrzeug ein beschädigtes Getriebe hat. Für die Analyse ist nur der Verlauf der Getriebewellendrehzahl mit einer Abtastfrequenz von 100 Hz verfügbar. Folgende Daten wurden aufgenommen und ein Experte hat die Anfahrten betrachtet und manuell klassifiziert:

- 152 Touren mit dem Fahrzeug ohne Getriebebeschaden; *normale Touren* (etwa 70 Stunden Fahrt)
 - mit 280 normalen Anfahrten (etwa 7,6 Minuten)
- 8 Touren mit dem Fahrzeug mit Getriebebeschaden; *anormale Touren* (etwa 2,38 Stunden Fahrt)
 - mit 46 normalen Anfahrten (etwa 1,37 Minuten)
 - mit 37 anormalen Anfahrten (etwa 1,77 Minuten)

Es ist bekannt, dass der fehlerhafte Verlauf der Drehzahl nur während einer Anfahrt erkennbar ist, weswegen sich die Analyse auf diese Situationen

beschränkt. Bild 6.2 veranschaulicht die Verwendung der Daten zur Evaluierung. Es wird untersucht, wie gut die anormalen Anfahrten erkannt werden können. Zum Test werden die Anfahrten der anormalen Touren verwendet; die normalen Anfahrten der normalen Touren dienen zum Training des Referenzmodells. Aufgrund der kleinen Datenbasis werden die Ergebnisse nicht wie in Kapitel 5 einmalig berechnet. Stattdessen werden Trainings- und Testphase mehrmals wiederholt und Mittelwert und Standardabweichung der Fläche unter der ROC-Kurve (AUC, vgl. Kapitel 4.2.1) als Evaluationskriterien berechnet. Pro Iteration wird nur ein zufälliger Ausschnitt von 80% der normalen Anfahrten zum Training verwendet. Dieses Vorgehen stellt bei wenigen Testdaten sicher, dass eine statistische Sicherheit des Evaluationskriteriums gegeben ist [GBC16].

Bei dem Signal *Drehzahl der Getriebewelle* handelt es sich um eine Zeitreihe und das Fehlverhalten ist nur über den Verlauf erkennbar. Daher wird statt einzelner Zeitpunkte ein Zeitfenster betrachtet. Im Folgenden beschreibt die Fensterlänge p wie viele Zeitpunkte verwendet werden (vgl. Formel 4.22). Zur Suche nach dem optimalen Modell, unabhängig von der Grenzfestlegung, wird die Fläche unter der ROC-Kurve (AUC) verwendet. Neben dem Isolation Forest Ansatz und einer SVM werden unterschiedliche neuronale Netze zur Zeitreihenvorhersage trainiert. Für das Modell mit größter Fläche werden dann die optimalen Grenzwerte bestimmt: Ab wann gilt ein Zeit-

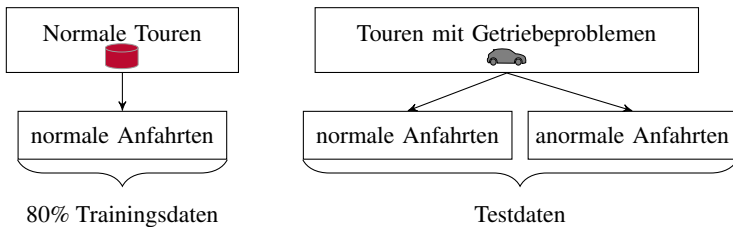


Bild 6.2: Datenaufteilung zur Evaluierung der Getriebeschäden pro Iteration

punkt als anormal und wie viele Zeitpunkte müssen innerhalb einer Anfahrt anormal sein, damit die komplette Anfahrt erkannt wird (vgl. Formel 5.6). In Kapitel 6.2 wird das optimale Modell ausgewertet. Dazu zeigen *Recall* und *Precision* den Anteil anormal erkannter Schäden und den Anteil tatsächlicher Schäden unter den erkannten Anomalien (vgl. Formeln 4.33 - 4.35).

6.1 Modelloptimierung

Im ersten Schritt wird anhand der AUC nach dem optimalen Modell gesucht. Die Ergebnisse basieren auf 25 Iterationen. Neben dem Isolation Forest Ansatz und der *one-class* SVM werden unterschiedliche Netzwerkarchitekturen zur Zeitreihenvorhersage untersucht. Die Variation von Netzwerktopologie, Fensterlänge und Hyperparameter führt sukzessiv zum Optimum.

Methodenvergleich

Da lediglich das Signal *Drehzahl der Getriebewelle* betrachtet wird und der Schaden nur im Zeitverlauf erkannt werden kann, wird der Featureraum um die kleinste Einheit von $x^{(t)}$ zu $(x^{(t-1)}, x^{(t-2)})$ vergrößert (vgl. Kapitel 4.1.2). Die Fensterlänge $p = 2$ ist die Anzahl der vergangenen Datenpunkte, die zur Vorhersage von Zeitpunkt $x^{(t)}$ verwendet werden. Folgende Methoden werden getestet:

- *One-class* SVM mit RBF-Kernel,
- Isolation Forest mit 256 Bäumen,
- Zeitreihenvorhersage, $x^{(t)} = f(x^{(t-1)}, x^{(t-2)})$. Hierfür wird ein *feed-forward* neuronales Netz mit einer verborgenen Schicht bestehend aus zehn Neuronen konstruiert. Die Parameter sind zufällig gesetzt und werden iterativ optimiert.

Im Folgenden werden die Netze durch die Schreibweise

$$[a, b_1, b_2, \dots, b_L, c] \quad (6.1)$$

dargestellt, wobei a für die Anzahl der Eingabeneuronen, b_l für die Anzahl der Neuronen in der l -ten Schicht ($l = 1, \dots, L$; L : Gesamtanzahl verborgener Schichten) und c für die Anzahl der Ausgabeneuronen steht. Bei der Darstellung $[2, 10, 1]$ handelt es sich somit um ein Netz, in dem sich die Eingabeschicht aus zwei Neuronen für die Zeitpunkte $x^{(t-1)}$ und $x^{(t-2)}$ zusammensetzt. Es gibt eine verborgene Schicht mit zehn Neuronen und das Neuron der Ausgabeschicht liefert die Vorhersage des aktuellen Zeitpunktes $x^{(t)}$ (vgl. Bild 6.3).

Die niedrige AUC in Tabelle 6.1 zeigt, dass zur Erkennung des Getriebeschadens die *one-class* SVM und der Isolation Forest nicht geeignet sind. Das Neuronale Netz liefert eine durchschnittliche AUC von etwa 0,8855.

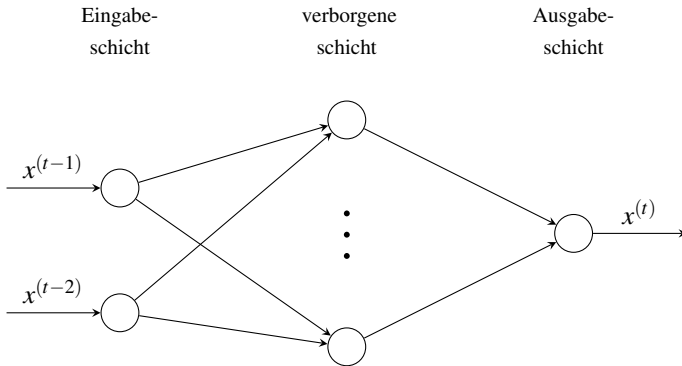


Bild 6.3: Netz zur Vorhersage der Getriebewellendrehzahl anhand der zwei letzten vergangenen Werte

Tabelle 6.1: Methodenvergleich zur Erkennung von Getriebebeschäden anhand der mittleren AUC aus 25 Iterationen und Fensterlänge $p = 2$

	AUC
Neuronales Netz, $[2, 10, 1]$	0,8855 \pm 0,0107
<i>one-class</i> SVM	0,7406 \pm 2,22* 10^{-16}
Isolation Forest	0,7640 \pm 0,0033

6.1.1 Vergleich Netzwerkarchitekturen und Topologien

Da das Neuronale Netz zur Zeitreihenvorhersage die erfolgversprechendste Methode ist, wird der Einfluss von Netzwerkarchitektur und -topologie auf die Ergebnisse untersucht. Daher findet ein Vergleich zwischen den in Kapitel 4.1.2 vorgestellten Methoden statt:

- *Feedforward* Netzwerk,
- Rekurrentes Netzwerk,
- Convolutional Netzwerk: Zum Erlernen eines convolutional Neuronalen Netzes müssen als weitere Parameter die Filterlänge (*FiL*) und die Anzahl an Faltungen definiert werden. Als Filterlängen werden die beiden Extremwerte analysiert: $FiL = 2$ und $FiL = p - 1$. Die Anzahl verborgener Schichten entspricht in diesem Fall der Anzahl an Faltungen.

Zusätzlich werden unterschiedliche Fensterlängen von $p = 0$ bis $p = 15$ geprüft; wie viel der Vergangenheit soll zur Vorhersage der Gegenwart $x^{(t)}$ verwendet werden. Der Einfluss der Netzwerkkomplexität wird anhand eines weiteren Netzwerks betrachtet. Eine erhöhte Komplexität ergibt sich aus mehr Schichten und Neuronen (Topologie: $[p, 50, 10, 50, 1]$).

Tabelle 6.2 zeigt die maximale durchschnittliche AUC für das *feedforward* Netz mit Fensterlänge $p = 2$ und Topologie $[2, 10, 1]$. Die erhöhte Komplexität zu $[p, 50, 10, 50, 1]$ ermöglicht für keine der Verfahren eine Verbesserung

der Ergebnisse und auch durch die Vergrößerung der Fensterlänge von $p = 2$ zu $p = 15$ entsteht kein Vorteil.

Des Weiteren wird in Tabelle 6.3 der Einfluss der Neuronenanzahl untersucht. Zusätzliche Gewichte verbessern die Ergebnisse allerdings nicht. Eine genauere Auflistung findet sich in Tabelle B.1 in Anhang B.

6.1.2 Anpassung der Hyperparameter

Nach der Auswahl von Methode, Netzwerkarchitektur und -topologie folgt die Anpassung der Hyperparameter. Diese setzen sich zusammen aus Fensterlänge und Regularisierungsparameter.

Tabelle 6.2: Vergleich verschiedener Netzwerkarchitekturen anhand der mittleren AUC aus 25 Iterationen

Netzwerkarchitektur	Netzwerktopologie	
	$[p, 10, 1]$	$[p, 50, 10, 50, 1]$
	Fensterlänge $p = 0$	
<i>Feedforward</i> Netz	$0,7256 \pm 0,0363$	$0,7179 \pm 0,0433$
	Fensterlänge $p = 2$	
<i>Feedforward</i> Netz	$0,8854 \pm 0,0107$	$0,7483 \pm 0,0177$
Convolutional Netz, $FiL = 2$	$0,8679 \pm 0,0069$	$0,8650 \pm 0,0038$
Convolutional Netz, $FiL = 1$	$0,8709 \pm 0,0057$	$0,8658 \pm 0,0044$
Rekurrentes Netz	$0,8726 \pm 0,0283$	$0,7746 \pm 0,0500$
	Fensterlänge $p = 15$	
<i>Feedforward</i> Netz	$0,8477 \pm 0,0265$	$0,7737 \pm 0,0501$
Convolutional Netz, $FiL = 2$	$0,8465 \pm 0,0132$	$0,8196 \pm 0,0087$
Convolutional Netz, $FiL = 14$	$0,8450 \pm 0,0112$	$0,8417 \pm 0,0107$
Rekurrentes Netz	$0,8160 \pm 0,0643$	$0,7775 \pm 0,0522$

Tabelle 6.3: Vergleich unterschiedlicher Neuronenanzahl für *feedforward* Netz mit Fensterlänge $p = 2$ anhand der mittleren AUC aus 25 Iterationen

	AUC
<i>Feedforward</i> Netz $[2, 3, 1]$	$0,8798 \pm 0,0152$
<i>Feedforward</i> Netz $[2, 5, 1]$	$0,8761 \pm 0,0216$
<i>Feedforward</i> Netz $[2, 10, 1]$	$0,8854 \pm 0,0107$
<i>Feedforward</i> Netz $[2, 50, 1]$	$0,8715 \pm 0,0238$
<i>Feedforward</i> Netz $[2, 100, 1]$	$0,8755 \pm 0,0147$

Vergrößerung der Fensterlänge:

Es findet die schrittweise Ausdehnung der Fensterlänge p statt. Hierfür wird das *feedforward* Netz mit einer verborgenen Schicht aus zehn Neuronen verwendet ($[p, 10, 1]$). Die AUC erhöht sich bei einer Fensterlänge $p > 2$ nicht signifikant (vgl. Tabelle 6.4). Das bedeutet, eine Vergrößerung der Fensterlänge ist nicht nötig.

Tabelle 6.4: Vergleich unterschiedlicher Fensterlängen p für Netztopologie $[p, 10, 1]$ anhand der mittleren AUC aus 25 Iterationen

	AUC
$p = 1$	$0,8727 \pm 0,0179$
$p = 2$	$0,8854 \pm 0,0107$
$p = 3$	$0,8809 \pm 0,0223$
$p = 4$	$0,8837 \pm 0,0173$
$p = 5$	$0,8894 \pm 0,0198$
$p = 6$	$0,8872 \pm 0,0259$
$p = 7$	$0,8865 \pm 0,0344$

Vergleich unterschiedlicher Regularisierung:

In Kapitel 4.1.1 werden der Sparsityterm und die L_1 - bzw. L_2 -Regularisierung als weitere Parameter eines neuronalen Netzes zur Reduzierung von Überanpassung vorgestellt. Die bisher höchste AUC von 0,8854 resultiert aus einem Netzwerk mit einer Sparsity gleich 0,01 (d.h. die durchschnittliche Aktivität/Ausgabe eines Neurons liegt nahe 0,01) und einer L_2 -Regularisierung von 0,01 (d.h. die L_2 -Norm der Gewichte muss kleiner als 0,01 sein). Der Einfluss dieser Parameter auf die Ergebnisse wird betrachtet indem für die beiden *feedforward* Netzwerke $[2, 10, 1]$ und $[2, 50, 10, 50, 1]$ exemplarisch andere Werte gewählt werden:

- *Feedforward* 1: Sparsity Faktor = 0,01, L_2 -Regulierung $< 0,01$;
- *Feedforward* 2: Sparsity Faktor = 0,001, L_2 -Regulierung $< 0,05$;
- *Feedforward* 3: Sparsity Faktor = 0,001, L_2 -Regulierung $< 0,001$.

Die Ergebnisse in Tabelle 6.5 zeigen, dass für das einfache Netz mit nur einer verborgenen Schicht und zehn Neuronen die Ergebnisse durch eine Vergrößerung der Faktoren, sprich einer weniger starken Regularisierung, nur gering verbessert werden können. Die durchschnittliche AUC steigt von 0,8718 auf 0,8855. Im Vergleich, für das Netz $[2, 50, 10, 50]$ steigen die Ergebnisse mit stärkerer Regularisierung. Grund dafür könnten die vielen zu lernenden Gewichte sein, die zu einer Überanpassung an die Trainingsdaten führen. Durch einen strengeren Sparsity Faktor kann dieser Einfluss reduziert werden.

Zusammengefasst zeigen die Untersuchungen, dass eine Erhöhung der Komplexität durch Netzwerkarchitektur, -topologie oder durch Vergrößerung der Fensterlänge keine Verbesserung der Ergebnisse bringt. Die Vorhersage der Gegenwart $x^{(t)}$ anhand der zwei letzten vorhergegangenen Zeitpunkte liefert die besten Werte. Das *feedforward* Netz mit Topologie $[2, 10, 1]$, einem Sparsity Anteil von 0,01 und einer L_2 -Regularisierung kleiner 0,01 (*feedforward* Netz 1) erreicht die maximale durchschnittliche AUC von etwa 0,8855.

Tabelle 6.5: Vergleich unterschiedlicher Sparsity Faktoren für das *feedforward* Netz mit $p = 2$ anhand der mittleren AUC aus 25 Iterationen

	$[2, 10, 1]$	$[2, 50, 10, 50, 1]$
<i>Feedforward</i> Netz 1	0,8854 \pm 0,0107	0,7483 \pm 0,0176
<i>Feedforward</i> Netz 2	0,8793 \pm 0,0106	0,8489 \pm 0,0250
<i>Feedforward</i> Netz 3	0,8718 \pm 0,0069	0,8697 \pm 0,0079

Im letzten Schritt wird untersucht, welche Grenzen geeignet sind, um eine Anfahrt anhand des Scores in *normal* oder *anormal* einzuordnen. Nach Formel 5.6 müssen zwei Schwellwerte bestimmt werden: Ab wann gilt ein Zeitpunkt als *anormal* (τ), und wie viele Zeitpunkte müssen innerhalb einer Anfahrt *anormal* sein, damit die gesamte Anfahrt erkannt wird (ζ). Das verwendete Evaluationskriterium ist hierbei der F-Score nach Formel 4.33. Mithilfe einer Rastersuche und dem Mittelwert aus 25 Wiederholungen wird der höchste F-Score von 0,8460 für τ_{q_x} mit $x = 0,97$ und $\zeta = 0,05$ gefunden.

6.2 Verwendung des optimalen Modells

Die bisherigen Vergleiche evaluieren die Einordnung in normale bzw. *anormale* Anfahrten während der Touren mit dem defekten Fahrzeug. Dieses Vorgehen stimmt mit dem aus Kapitel 5 überein, indem zwischen *Anomalie* und *keine Anomalie* innerhalb der gefahrenen Testfahrten unterschieden wird. In der Anwendung allerdings ist es wichtiger, die Fahrzeuge voneinander zu unterscheiden. Daher wird im nächsten Schritt überprüft, wie sich die Ergebnisse mit einem erweiterten Testdatensatz verändern. Es werden zum Training 90% der Anfahrten der normalen Touren verwendet. Die restlichen 10% werden als Vertreter der Klasse *normal* zum Testdatensatz hinzugefügt. Bild 6.4 verdeutlicht die Aufteilung. Auch hier werden in jeder der 25 Iterationen die Trainingsanfahrten zufällig gezogen.

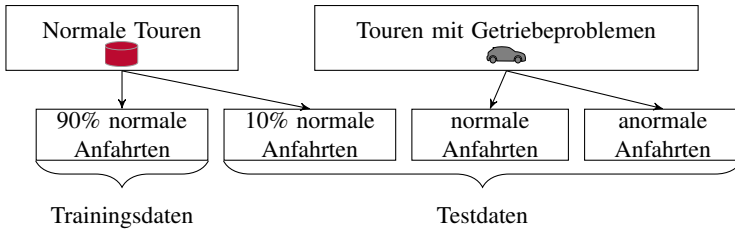


Bild 6.4: Datenaufteilung zur Evaluierung der Getriebebeschäden

Tabelle 6.6 lässt eine Verringerung des F-Scores für das *feedforward* Netz $[2, 10, 1]$ erkennen. Precision und Recall zeigen, dass die Verschlechterung darauf zurückzuführen ist, dass die zusätzlichen Anfahrten fälschlicherweise als anormal eingestuft werden, die Precision sinkt. Es werden 77,3% der fehlerhaften Anfahrten erkannt und 80,45% der erkannten Anomalien sind tatsächlich anormal.

Zusammengefasst wird ein Klassifikator verwendet, der innerhalb des Trainings nur normale Anfahrten gesehen hat und somit die Normalität beschreibt. Durch die *one-class* Anomaliedetektion können die Fehler erkannt werden, ohne zusätzliches Wissen oder fehlerhafte Anfahrten ins Training zu stecken. Allerdings ist ein F-Score von 0,7860 kein zufriedenstellendes Ergebnis. Das Modell kann nicht im Fahrzeug verwendet werden, um

Tabelle 6.6: Vergleich unterschiedlicher Testdatensets für das *feedforward* Netz $[2, 10, 1]$ mit Fensterlänge $p = 2$ anhand des mittleren F-Scores aus 25 Iterationen

	Nur Anfahrten aus den Touren mit Getriebeproblemen (vgl. Bild 6.2)	Zusätzliche Anfahrten aus den normalen Touren (vgl. Bild 6.4)
F-Score	$0,8460 \pm 0,0452$	$0,7860 \pm 0,0479$
Precision	$0,9400 \pm 0,0230$	$0,8045 \pm 0,0496$
Recall	$0,7730 \pm 0,0738$	$0,7730 \pm 0,0738$

den Fahrer beispielsweise zu warnen, da die *Precision* von 80,45% bedeutet, dass jeder fünfte Fahrer fälschlicherweise einen Hinweis bekommt. Ein *two-class* Klassifikator, der bereits im Training Datenpunkte beider Klassen verwendet und auf die Unterscheidung hin optimiert wird, erzielt bessere Ergebnisse [RHM17]. Somit muss für einen konkreten Anwendungsfall auch ein spezifizierter (*two-class*) Klassifikator trainiert werden. Die Idee der Anomalieerkennung anhand einer *one-class* Klassifizierung dient lediglich als Anhaltspunkt, welche Daten von Interesse sein könnten und näher analysiert werden müssen.

7 Zusammenfassung und Ausblick

Da *Big Data* auch in der Automobilbranche angekommen ist, liegen für den Hersteller die Herausforderungen darin, einen Wert aus den Daten zu schöpfen und die Übertragung von Fahrzeug zu Backend zu reduzieren. Das Ziel der Arbeit war es daher, außergewöhnliche Ereignisse in den fahrzeugin-ternen Kommunikationsdaten zu erkennen, woraufhin Daten vorselektiert werden können. Eine Vorauswahl im Fahrzeug erfüllt das Prinzip der Sparsamkeit und ermöglicht neben der Kostensenkung auch die Einhaltung der Datenschutzgrundverordnung.

Es wurde die Idee der *one-class* Anomalieerkennung vorgestellt, welche dem zu Beginn zitierten Ansatz von Sherlock Holmes folgt: *It is a capital mistake to theorize before one has data. Insensibly, one begins to twist the facts to suit theories, instead of theories to suit facts.* Denn nicht der Entwickler, sondern die Flottendaten beschreiben was normal ist. Die Herangehensweise bietet neue Potentiale zur Anomaliedetektion ohne konkrete a-priori Definition. Die Einschränkung der Datenübertragung bedeutet, dass das Backend mit fehlenden bzw. seltenen Situationen angereichert wird, wohingegen reichlich vertretene Ereignisse eingespart werden. Aufgrund der Annahme, dass die Mehrheit der Fahrzeuge intakt ist, werden somit Anomalien identifiziert, die den Automobilhersteller auf Fehlerquellen und Verbesserungspotentiale hinweisen.

Da Anomalien in Fahrzeugdaten erkannt werden sollen, erläuterte Kapitel 2 zunächst genauer, um welche Daten es sich handelt. Außerdem wurde eine Einführung in die Begriffe des maschinellen Lernens gegeben, da nur eine automatisierte Auswertung der Daten den Analysten zur Wissensgene-

rierung und Wertschöpfung dient. Das Konzept der Anomalieerkennung in den Fahrzeugdaten wurde in Kapitel 3 entwickelt. Dies umfasst das Sammeln der Flottendaten, das Training des Referenzmodells im Backend und die Anwendungsphase in den Fahrzeugen. Daraus ergaben sich die Anforderungen an die Methodik. Das Modell wird im Backend trainiert, muss aber in die Fahrzeuge übertragbar sein. Zudem wird angenommen, dass die Daten im Backend größtenteils normal sind, Ausreißer sollten allerdings erkannt und ignoriert werden. Des Weiteren sind Methoden zu bevorzugen, die den Anwender bei der Interpretation aufgedeckter Anomalien unterstützen und einen Score anstatt einer Klasse als Ausgabe liefern. Kapitel 4 gab einen Überblick über den aktuellen Stand der Technik und erörterte, wie das Konzept aus Kapitel 3 umgesetzt werden kann. Sowohl statische als auch dynamische Methoden zur Anomalieerkennung wurden eingeführt und ein Anwendungsüberblick zeigte, in welchen Bereichen Anomalien erfolgreich gefunden werden. Außerdem behandelte das Kapitel die Schritte des Post-processings, bestehend aus der Grenzfestlegung und Interpretation. Nach der Berechnung des Anomalie Scores muss anhand dessen eine Einordnung in *normal* oder *anormal* erfolgen. Es wurden die Scores der Trainingsdaten zur Definition eines Grenzwertes verwendet. Zusätzlich stellte das Kapitel ein Random Forest Verfahren zur Erklärung festgestellter Anomalien anhand der auffälligen Signale vor. Unabhängig von der Methode zur Anomalieerkennung werden im Postprocessing anormale Trainingsdaten ignoriert und erkannte Anomalien nachvollzogen.

Als beispielhafte Anwendung verwendete Kapitel 5 real eingefahrene Flottendaten und provozierte Anomalien. Neben Ereignissen wie Unfällen oder ABS-Eingriffen wurden auch weniger gravierende Situationen evaluiert (z.B. das Schließen des Heckdeckels per Hand oder ein eingeklemmtes Buch im Fenster). Die Anwendung zeigte, dass für diese Beispiele das Replikator Neuronale Netz am besten geeignet ist. Das Ziel der Datenreduktion um das Hundertfache führte in der Anwendungsphase im Fahrzeug dazu, dass durchschnittlich 0,715% der Daten einen höheren Score als der

Grenzwert aufwiesen und somit als Anomalie eingeordnet und an das Backend übertragen wurden. Dabei erkannte das neuronale Netz den Großteil der real eingefahrenen Anomalien, während keine der normalen Datenpunkte fälschlicherweise als anormal eingestuft wurden. Weder das Clusteringverfahren, noch die Isolation Forest Methode lieferten für die konkreten Beispiele annähernd gute Ergebnisse. Zudem wurde gezeigt, dass die Ereignisse, die definitiv nicht im Training vorliegen (Unfall und eingeklemmtes Buch im Fenster), durch ihren hohen Anomalie Score eindeutig von den Flottendaten unterscheidbar sind und insbesondere sicherheitsrelevante Ereignisse (Unfall und ABS-/ESP-Eingriff) entdeckt werden können. Eine durchschnittliche AUC von 0,8 demonstrierte, dass sich die ungewöhnlichsten Ereignisse unterscheiden ließen, sollten allerdings weitere Anomalien erkannt werden, bewirkte die Herabstufung des Grenzwertes eine *False Positive Rate* von 6,8%. Für die Datenmenge von 2,1 MByte/s, die ein Oberklassefahrzeug generiert, ergibt sich in diesem Fall ein Volumen von 142,8 KByte/s, die die Experten unnötigerweise (da normal) bewerten müssten, was das System für den Serieneinsatz verbesserungsbedürftig macht.

Die wenigen realen Anomalien deckten nur einzelne, hervorstechende Signalwerte ab. Daher wurden in Abschnitt 5.3.3 synthetische Fälle überprüft, welche die Schwachstellen der Methode verdeutlichten. Zum einen konnten nur punktuelle Anomalien erkannt werden. Ausreißer innerhalb spezieller Kontexte waren nicht von den Trainingsdaten unterscheidbar. Erst eine geeignete Aufteilung der Daten ergab verbesserte Ergebnisse. Das Training unterschiedlicher Referenzmodelle für verschiedene Fahrsituationen war notwendig, um eine Verfeinerung dieser zu erreichen. Außerdem zeigte sich, dass die synthetischen Beispiele einen deutlich höheren Anomalie Score haben als ihre normalen Gegenspieler, doch eine perfekte Trennbarkeit war nicht gegeben. Für die Verwendung im Fahrzeug ergibt sich daraus eine Herabstufung des Schwellwertes, ab wann ein Ereignis als anormal gilt. Dies wiederum impliziert, dass die Anzahl übertragener, normaler Datensätze steigt. Es handelt sich zum Beispiel bei einem *offenen Heckdeckel* nach

Definition um eine Anomalie, da es ein Ereignis ist, welches nur 0,006% der Trainingsdaten abdeckt, einen *Business Value* stellt es hingegen nicht direkt dar. Erst längerfristige Analysen, die beispielsweise Hinweise darauf geben, dass die Heckdeckel bestimmter Fahrzeuge häufig nicht richtig geschlossen sind, liefern Unternehmensnutzen.

Die Anomalieerkennung im Fahrzeug ist daher nicht als Alarmsystem, sondern zur Reduzierung des Datentransfers geeignet. Nur auf dem Backend fehlende Ereignisse werden übertragen. Mehrwert für den Automobilhersteller entsteht frühestens mit weiterer Verarbeitung im Backend, da dann bewertet werden kann, ob es sich bei der Anomalie um ein tatsächlich interessantes Ereignis handelt. Der erste Schritt, das Nachvollziehen der ausschlaggebenden Signale mithilfe von Entscheidungsbäumen, veranschaulichten die Kapitel 4.2.3 und 5.3.4.

Kapitel 6 beschäftigte sich mit der Vorhersage eines Getriebebeschadens als Anwendungsfall. Unterschied zu den restlichen Analysen war hierbei die Verwendung der Zeitreihe und die Einschränkung auf nur ein Signal. Die Topologie und Parameter eines neuronalen Netzes zur Zeitreihenvorhersage wurden sukzessive optimiert. Es stellte sich heraus, dass bei konkretem Vorwissen ein *two-class* Klassifikator, der zwischen defekten und intakten Getrieben unterscheidet, der Anomalieerkennung vorzuziehen ist. Der Ansatz der Anomalieerkennung war in der Lage, 77,3% der Schäden zu erkennen, während nur 80,45% der erkannten Anomalien tatsächlich anomal waren. Diese Ergebnisse würden beim Einsatz im Fahrzeug bedeuten, dass jeder fünfte Fahrer fälschlicherweise eine Warnung erhält, was nicht praktikabel ist. Ein optimierter *two-class* Klassifikator erzielt in diesem Fall mit einem F-Score von fast 0,95 bessere Ergebnisse [RHM17].

Zusammengefasst bedeutet das für den Automobilhersteller, dass eine Reduzierung des Datentransfers gewährleistet und reguliert werden kann, doch werden auch Datenpunkte übertragen, bei denen es sich lediglich um seltene Ereignisse, nicht aber um tatsächlich anormales Fahrer- oder Fahrzeugverhalten handelt. Eine Herabsetzung des Schwellwertes, sodass mehr Anoma-

lien übertragen werden, verstärkt den Effekt zusätzlich. Außerdem ist das Referenzmodell nur in der Lage, punktuelle Anomalien zu erkennen, während kontextuelle Ausreißer übersehen werden. Jedoch ermöglicht die Realisierung des Konzepts innerhalb einer Prototypenflotte eine iterative Verbesserung des Referenzmodells. Werden im Backend die Schwachstellen erkannt, kann es entsprechend angepasst werden. Wie in Kapitel 5.3.3 beispielhaft gezeigt, kann die Optimierung durch das Sammeln zusätzlicher Daten oder durch die Vorauswahl von Signalen und Kontexten erfolgen.

Bewirkt eine solche Verfeinerung die Verringerung der *False Positive Rate*, dann wird das Konzept der Anomalieerkennung im Fahrzeug in Zukunft einen wichtigen Platz in der Automobilbranche einnehmen. Weitere Bedingung dafür ist, dass das Postprocessing auf dem Backend erfolgreich durchgeführt wird, sodass den Experten eine überschaubare und repräsentative Vorauswahl an Anomalien zur manuellen Betrachtung vorgelegt werden kann. Gelingt dies, dann dient die Erkennung von Anomalien – trotz bestehenden Verbesserungsbedarfes – der Qualitätssicherung, da defekte Bauteile identifiziert werden können. Der Hersteller bekommt ein Gespür für potentielle Fehlerquellen und in Abstimmung mit den Spezialisten können entsprechend angepasste Klassifikatoren zur Vorhersage konkreter Schadensfälle trainiert und im Fahrzeug als Dienst integriert werden. Das hat zur Folge, dass Gewährleistungskosten sinken und Kundenzufriedenheit und Loyalität steigen [Ale15].

Die Anomalieerkennung kann auch als Regressionstest verstanden werden. Modifikationen ausgiebig geprüfter Modelle bzw. Software durch Derivatentwicklung, Modellerneuerung oder Softwareupdates können getestet werden. Es wird das bereits trainierte Referenzmodell verwendet und die beobachteten Anomalien stellen eine Abweichung vom Sollzustand dar. Die Analyse dieser Ereignisse ermöglicht die Identifikation von Mängeln. Die Verwendung in Erprobungsflotten bewirkt ein gezieltes Triggern der aufzeichnenden Daten. Wurde eine Anomalie erkannt, wird die gesamte Buskommunikation unmittelbar vor, während und nach dem Event aufgezeich-

net. Daher existiert neben dem Entwickler ein weiteres System zur Aktivierung der Datenaufzeichnung. Auch besteht die Möglichkeit, das Referenzmodell bereits in der virtuellen Erprobung zu verwenden, um frühzeitig auf Diskrepanzen mit dem Vorgängermodell zu reagieren.

Das Konzept, dass nicht der Entwickler, sondern die Flottendaten definieren, was *normal* ist, wird in Zukunft durch die steigende Komplexität der Funktionen von großer Wichtigkeit sein. Es können nicht alle Testfälle a-priori abgedeckt werden, weswegen die Anomalieerkennung a-posteriori Fehlerfälle meldet.

Um diese Ziele zu erreichen, muss als nächste Maßnahme ein Konzept für das Postprocessing im Backend aufgebaut werden. Die Arbeit zeigte Verfahren auf, um die ausschlaggebenden Signale nachzuvollziehen. Für eine vollständige Interpretation bedarf es weiterer Bausteine wie Visualisierung, Interpretation und Evaluation (Teil des Prozesses *Knowledge Discovery in Databases* (KDD) [FPsS96]). Zur Aufbereitung werden Anomalien selben Typs gebündelt und in einen Zusammenhang wie Fahrer, Zeit, Ort oder Fahrzeugkomponente gebracht. Zukünftiges Ziel muss ein Wissensapparat sein, der fortlaufend aufgebaut wird. Dieser ordnet die erkannten Anomalien in Klassen ein und ermöglicht eine passende Maßnahme. Hinzukommend ist eine Verbesserung des Referenzmodells notwendig. Die Arbeit zeigte die exemplarische Aufteilung in vier unterschiedliche Fahrsituationen. Im weiteren Verlauf müssen diese verfeinert und konkretisiert werden.

Damit der Automobilhersteller verlässliche Aussagen erhält, sind zusätzliche Testfälle erforderlich. Die prototypische Anwendung in Kapitel 5 diente lediglich als Beispiel und die Ergebnisse gelten nur für den konkreten Datensatz.

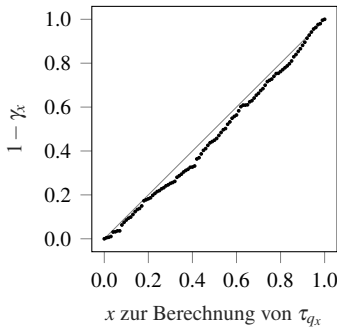
Ergänzend muss zur Realisierung des Gesamtsystems der in Bild 3.3 dargestellte Übergang von Anwendungs- zu Trainingsphase automatisiert erkannt werden. Sobald das Referenzmodell veraltet ist, befindet sich auch die Datenbasis auf dem Backend nicht auf dem neuesten Stand und wird somit im Rahmen einer neuen Trainingsphase aktualisiert. Anhand eines

ungewöhnlichen Sprungs in der Ankunftsrate der Anomalien kann ein sogenannter *Change Point* erkannt werden. Unterschiedliche Vorgehensweisen, um einen solchen *Change Point* automatisiert zu erkennen, sollten evaluiert werden.

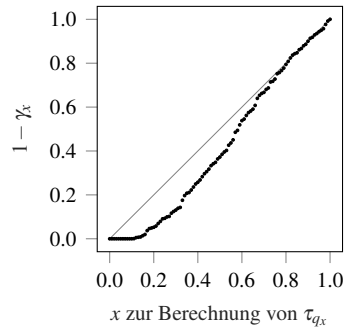
Zur Verwirklichung des Gesamtsystems der Anomalieerkennung zur Datenreduktion wird das Referenzmodell im Fahrzeug verwendet. Die Untersuchungen ignorieren die Grenzen der Vorverarbeitung im Fahrzeug. Die Implementierung und Validierung auf dem Steuergerät ist einer der nächsten Schritte.

A Prototypische Anwendung

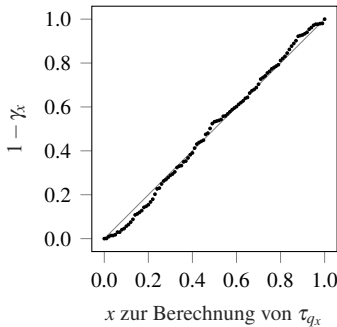
A.1 Vergleich Trainings- und Kontrolldaten



(a) IForest



(b) ReplNN



(c) SOM

Bild A.1: Vergleich Trainings- und Kontrolldaten anhand Anomalie Score nach 5.5, Kampagne K_2

A.2 Hyperparameter Isolation Forest

Tabelle A.1: Vergleich der Hyperparameter für Kampagne K_1

s	t					
	100		150		200	
	AUC _{te}	AUC _{tr}	AUC _{te}	AUC _{tr}	AUC _{te}	AUC _{tr}
256	0,5479	0,7293	0,5225	0,7063	0,5677	0,7763
400	0,6083	0,7945	0,5683	0,7806	0,5350	0,7643
500	0,5477	0,7392	0,5329	0,7837	0,5293	0,7420
700	0,5474	0,7430	0,5372	0,7361	0,5406	0,7607

Tabelle A.2: Vergleich der Hyperparameter für Kampagne K_2

s	t					
	100		150		200	
	AUC _{te}	AUC _{tr}	AUC _{te}	AUC _{tr}	AUC _{te}	AUC _{tr}
256	0,5731	0,5465	0,6279	0,5644	0,5963	0,5605
400	0,6616	0,6203	0,6100	0,5781	0,6632	0,5831
500	0,6056	0,5896	0,6586	0,6150	0,6024	0,6094
700	0,6214	0,6064	0,6417	0,6247	0,6674	0,6281

A.3 Synthetische Testfälle

A.4 Nachvollziehbarkeit erkannter Anomalien

Tabelle A.3: Evaluierung synthetischer Testfälle anhand der Fläche unter der ROC-Kurve

	K1				K2				
	Iforest AUC _{ir}	AUC _{ie}	ReplNN AUC _{ir}	AUC _{ie}	Iforest AUC _{ir}	AUC _{ie}	ReplNN AUC _{ir}	AUC _{ie}	
1.	Fahrt > 250 km/h, normal	0,9156	0,8585	0,5436	0,5233	0,6352	0,3855	0,4504	0,2752
	angezeigte Geschwindigkeit*3	0,8548	0,8507	0,9944	0,9974	0,9259	0,8693	0,9956	0,9903
	angezeigte Geschwindigkeit/3	0,7954	0,8017	0,9613	0,9604	0,9182	0,8630	0,9820	0,9670
	angezeigte Geschwindigkeit = 0	0,7980	0,8043	0,9720	0,9726	0,9186	0,8632	0,9874	0,9765
2.	Kühlmitteltemperatur >100 °C, normal	0,5908	0,6503	0,4994	0,4872	0,2343	0,0762	0,2071	0,0745
	Öltemperatur <10% Quantil, normal	0,9621	0,9491	0,7835	0,8299	0,5773	0,3147	0,5627	0,3860
	Öltemperatur herabgesetzt, manipuliert	0,6967	0,7264	0,9838	0,9916	0,3162	0,1176	0,8863	0,9030
3.	Kühlmittel <49°C, normal	0,9621	0,9491	0,7819	0,8190	0,6310	0,3863	0,5329	0,3635
	Öltemperatur >90% Quantil, normal	0,5786	0,6464	0,4060	0,4192	0,2531	0,1032	0,2922	0,1612
	Öltemperatur erhöht manipuliert	0,9450	0,9252	0,9761	0,9875	0,5645	0,3355	0,8640	0,8647
4.	Spiegelheizung bei 30 °C	-	-	-	-	0,2909	0,0983	0,4155	0,2319
	Stand mit geöffneter Tür, normal	-	-	-	-	0,8569	0,7109	0,7734	0,6864
7.	Fahrt mit geschlossener Tür, normal	-	-	-	-	0,5634	0,3471	0,5558	0,4440
	Fahrt mit offener Tür, manipuliert	-	-	-	-	0,9648	0,8516	0,9656	0,9737
8.	Stand mit geöffnetem HD, normal	-	-	-	-	0,9868	0,9208	0,9854	0,9935
	Fahrt mit geöffnetem HD, manipuliert	-	-	-	-	0,9973	0,9767	0,9816	0,9915
	Mittelwerte für Anomalien	0,8180	0,8217	0,9775	0,9819	0,7370	0,6219	0,8847	0,8623

Tabelle A.4: Ausschlaggebende Signale ausgewählter Anomalien, erkannt durch Random Forest Verfahren, Teil 1

Anomalie	Signal 1	Signal 2	Signal 3
Unfall 1	Beschleunigung in x-Ri. Lenkwinkelgeschw.	ω um die z-Achse Geschwindigkeit	Längsbeschleunigung angezeigte Geschwindigkeit
ABS + ESP Eingriff 1	ω um die z-Achse	Bremsdruck	Längsbeschleunigung
Unfall 2	Kilometerstand	Lenkwinkelgeschw.	Luftgüte Stickoxide
ABS Eingriff	Bremsdruck	relative Luftfeuchte	Längsbeschleunigung
ABS Eingriff	Beschleunigung in y-Ri.	ω um die z-Achse.	Querbeschleunigung
ABS Eingriff	Bremsdruck	Beschleunigung in x-Ri.	Längsbeschleunigung
ABS Eingriff	Bremsdruck	Außentemperatur	Beschleunigung in x-Ri.
ABS + ESP Eingriff 2	Bremsdruck	Beschleunigung in x-Ri.	Längsbeschleunigung
Spoiler spielt verrückt	BT FH in Bewegung hoch	Öltemperatur	Innenbeleuchtung aktiv
Fahrt mit offenem Tank	BT FH in Bewegung hoch	Öltemperatur	Innenbeleuchtung aktiv
Fernlicht am Tag	Blinkerhebel Fernlicht	Erkennung BF vorne	Außentemperatur
Fahrt mit offener BT	BT FH in Bewegung hoch	angezeigte Geschwindigkeit	Geschwindigkeit
Überdrehzahl 4700 rpm	aktuelle Fahrstufe	Motordrehzahl	angezeigte Geschwindigkeit

Tabelle A.5: Ausschlaggebende Signale ausgewählter Anomalien, erkannt durch Random Forest Verfahren, Teil 2

Anomalie	Signal 1	Signal 2	Signal 3
Überdrehzahl, 5873 rpm	Blinkerhebel rechts	Kilometerstand	Leerlaufsolldrehzahl
Fensterheber HR	HR FT manuell hoch	Kilometerstand	HD HS Pos
FH mit Buch	BT FH automatisch hoch	BT FH Öffnung	Kilometerstand
nur BF angeschnallt	HD HS Pos	Erkennung BF vorne	relative Luftfeuchte
Spoiler spielt verrückt	absolute HKP	relative HKP	BT Sperrklinke
HD mit Hand schließen	absolute HKP	relative HKP	BT FH in Bewegung hoch
Alle FH schließen	BT FH Öffnung	BT FH in Bewegung tief	HD HS Pos
Überdrehzahl, 6231 rpm	Kilometerstand	Motordrehzahl	Strombereich der Batterie
Spurhalteassistent	y-Abstand zum Fahrzeug-Koordinatensystem	Kilometerstand	Außentemperatur
Handbremse und Fahrt	Kilometerstand	Öltemperatur	Aktivitätsstatus Start/Stopp
FH mit Buch	BT FH automatisch hoch	BT FH Öffnung	Strombereich der Batterie
HD mit Hand schließen	relative HKP	BT FH Bewegung hoch	Gurtwarnung BF vorne
Vollbremsung	Bremsdruck	Beschleunigung in x-Ri.	Längsbeschleunigung
Kreise fahren	ω um die z-Achse	Kilometerstand	Beschleunigung in y-Ri.

B Evaluierung Getriebebeschaden

Tabelle B.1: Vergleich unterschiedlicher Neuronenanzahl für das *feedforward* Netz mit Fensterlänge $p = 2$

	AUC
<i>Feedforward</i> Netz [2, 3, 1]	$0,8796 \pm 0,0133$
<i>Feedforward</i> Netz [2, 4, 1]	$0,8801 \pm 0,0184$
<i>Feedforward</i> Netz [2, 5, 1]	$0,8795 \pm 0,0156$
<i>Feedforward</i> Netz [2, 6, 1]	$0,8790 \pm 0,0163$
<i>Feedforward</i> Netz [2, 7, 1]	$0,8773 \pm 0,0181$
<i>Feedforward</i> Netz [2, 8, 1]	$0,8798 \pm 0,0149$
<i>Feedforward</i> Netz [2, 9, 1]	$0,8764 \pm 0,0207$
<i>Feedforward</i> Netz [2, 10, 1]	$0,8817 \pm 0,0120$
<i>Feedforward</i> Netz [2, 20, 1]	$0,8790 \pm 0,0178$
<i>Feedforward</i> Netz [2, 30, 1]	$0,8826 \pm 0,0134$
<i>Feedforward</i> Netz [2, 40, 1]	$0,8781 \pm 0,0153$
<i>Feedforward</i> Netz [2, 50, 1]	$0,8788 \pm 0,0189$
<i>Feedforward</i> Netz [2, 100, 1]	$0,8841 \pm 0,0131$

Abkürzungs- und Symbolverzeichnis

Abkürzungen

ABS Antiblockiersystem

AUC Area Under ROC-Curve

BASt Bundesanstalt für Straßenwesen

BMU Best Map Unit

BF Beifahrer

BT Beifahrertür

CAN Controller Area Network

cGW zentrale Gateway

CBLOF Cluster Based Local Outlier Factor

COF Connectivity Based Outlier Factor

DFKI Deutsches Forschungszentrum für Künstliche Intelligenz

DLC *Distance Lane Cross*

ESP Elektronisches Stabilitätsprogramm

FH Fensterheber

FN False Negatives

- FP** False Positives
- FT** Fahrertür
- GByte** Gigabyte
- HD HS Pos** Heckdeckel Hallsensor Position
- HKP** Heckklappenposition
- HR** hinten, rechts
- INFLO** Influenced Outlierness
- IQR** Interquartile range
- KDD** Knowledge Discovery in Databases
- LIN** Local Interconnect Network
- LOF** Local Outlier Factor
- LTE** Long Term Evolution
- MByte,MB** Megabyte
- MOST** Media Oriented Systems Transport
- QR** Quantization Error
- RBF** Radial Basis Function
- ReplNN** Replikator Neuronales Netz
- Ri.** Richtung
- ROC** Receiver-Operating-Characteristic
- SOM** Self Organizing Maps

SVDD Support Vektor Data Description

SVM Support Vektor Machine

TLC *Time Lane Cross*

TN True Negative

TP True Positives

Symbole

A Menge der Anomalien, Amper

C Celsius

K Menge der Kontrolldaten

m Meter

N Menge der normalen Testdaten

Nm Newtonmeter

rpm *Rounds per Minute*, Umdrehungen pro Minute

s Sekunde

T Menge der Trainingsdaten

V Volt

° Grad

Abbildungsverzeichnis

1.1	Big Data beschrieben durch 5 Vs	2
1.2	Datenaustausch zwischen den Steuergeräten	3
1.3	Beispielrechnung zur Veranschaulichung der Datenmenge auf dem Backend	10
1.4	Unterscheidung zwischen Anwendungsfall und Anomalieerkennung	11
2.1	Einfaches Datenwort nach [Win17]	22
2.2	Interne Fahrzeugkommunikation	23
2.3	Datenübertragung von Fahrzeug zu Backend	24
2.4	Definition von Kampagnen anhand unterschiedlicher Fahrzeugflotten	25
2.5	Beispiel der Überanpassung anhand von Polynomapproximation nach [Bis06]	31
2.6	Idee der <i>one-class</i> Klassifikation	32
2.7	Geschwindigkeit und Motordrehzahl als multivariate Zeitreihe	33
3.1	Ablauf Anomalieerkennung	36
3.2	Anomalieerkennung zur Reduzierung des Datentransfers	38
3.3	Übergänge zwischen Trainings- und Anwendungsphase	39
3.4	Realisierung verschiedener Referenzmodelle durch Kampagnen	41
3.5	Aufteilung in Trainings- und Anwendungsphase	42
4.1	Statische Anomalieerkennung	46
4.2	Anomales Verhalten im Zeitverlauf	47

4.3	Vergleich von statischen und dynamischen Anomalien	48
4.4	Methoden zur Anomalieerkennung	50
4.5	Histogrammbasierte Anomalieerkennung	55
4.6	Kerndichteschätzer	56
4.7	Local Outlier Factor	58
4.8	Nachteil LOF: Ähnliche Dichten nach [Zha13]	58
4.9	Nachteil LOF: Nähe unterschiedlicher Dichten nach [Zha13] . .	58
4.10	Vergleich verschiedener Clustermethoden	61
4.11	Self-organising Map	63
4.12	Replikator Neuronales Netz mit einer verborgenen Schicht . . .	66
4.13	One-Class Support Vektor Machine	70
4.14	Isolation Forest Ansatz	72
4.15	Bestimmung der Pfadlänge je Baum	73
4.16	Beispiel zur Veranschaulichung des Isolation Forest	74
4.17	Zeitreihenvorhersage mit neuronalem Netz	78
4.18	Zeitreihenvorhersage mit rekurrentem Netz	79
4.19	Zeitreihenvorhersage mit convolutional Netz	80
4.20	Beispielhafte Anomalie Scores	86
4.21	Anomalie Scores der Trainingsdaten und angepasste Verteilungsfunktionen	89
4.22	ROC-Kurve für das Beispiel aus Bild 4.20	91
5.1	Generierung der Datenbasis	99
5.2	Achsenbeschriftung Fahrzeug	100
5.3	Evaluationsvorgehen je Kampagne	104
5.4	Aufteilung der ungelabelten Flottendaten in Trainings- und Kontrolldaten	105
5.5	Verwendung der Trainingsdaten zur Bewertung der Anomalieerkennung	106
5.6	Verwendung der gelabelten Testdaten zur Bewertung der Anomalieerkennung	107

5.7	Diskrepanz zwischen Anomalie Score und Grundwahrheit	108
5.8	Vergleich Trainings- und Kontrolldaten, K_1	110
5.9	Vergleich Trainings- und Kontrolldaten anhand Anomalie Score je Zeitpunkt	112
5.10	Einfluss der Trainingsdatenmenge auf $\gamma_{(0,95)}$	114
5.11	Vergleich ReplNN und IForest, K_1	116
5.12	Vergleich ReplNN und IForest, K_2	117
5.13	FPR_x für die verschiedenen Methoden und Kampagnen in Ab- hängigkeit von τ_{q_x}	118
5.14	Synthetische Anomalien durch Extrahieren und Manipulieren normaler Fahrsituationen	121
5.15	Zusammenhang Öl- und Kühlmitteltemperatur	123
5.16	Trennbarkeit vom Maximum je Signal von den Trainingsdaten in Abhängigkeit der Kurtosis	126
5.17	Anomalie <i>konstante Fahrt trotz Bremsdruck</i> in unterschiedli- chen Kontexten	128
5.18	ROC-Kurve zu Testfällen <i>konst. Fahrt trotz Bremsdruck</i> , Kam- pagne K_3	130
5.19	Scatterplot der Signale mit höchster Wichtigkeit, Unfall	133
5.20	Verlauf von Signal <i>Status des Beifahrer Fensterhebers (automa- tisch hoch)</i>	133
6.1	Unterschied zwischen normaler und anormaler Anfahrt	136
6.2	Datenaufteilung zur Evaluierung der Getriebeschäden	137
6.3	Netz zur Vorhersage der Getriebewellendrehzahl	139
6.4	Datenaufteilung zur Evaluierung der Getriebeschäden	145
A.1	Vergleich Trainings- und Kontrolldaten, K_2	155

Tabellenverzeichnis

1.1	Grade der Automatisierung und ihre Definition nach [Ver15] . . .	5
2.1	Beschreibung einer Session auf dem Backend	25
2.2	Beschreibung einer Session nach Weiterverarbeitung	26
4.1	Methodenvergleich	53
4.2	Performance mit unterschiedlichen Grenzwerten	91
5.1	Datenüberblick	100
5.2	Anteil erkannter Anomalien in den Kontrolldaten	111
5.3	Methodenvergleich anhand der AUC	112
5.4	Vergleich der Hyperparameter für Isolation Forest	113
5.5	FPR_x in Abhängigkeit von Grenzwert τ_{q_x}	118
5.6	Vergleich synthetischer Anomalien mit normalen Gegenspielern	124
5.7	Trennbarkeit der Anomalie von Trainings- und Testdaten in Ab- hängigkeit von Daten- und Signalauswahl	129
5.8	Interpretation der Anomalie <i>Unfall 2</i> durch Random Forest . . .	132
5.9	Interpretation der Anomalie <i>Fenster schließen mit eingeklemm-</i> <i>tem Buch</i>	134
6.1	Methodenvergleich zur Erkennung von Getriebebeschäden an- hand der mittleren AUC	140
6.2	Vergleich unterschiedlicher Netzwerkarchitekturen	141
6.3	Vergleich unterschiedlicher Neuronenanzahl	142
6.4	Vergleich unterschiedlicher Fensterlängen	142
6.5	Vergleich unterschiedlicher Sparsity Faktoren	144

6.6	Vergleich unterschiedlicher Testdatensets	145
A.1	Vergleich der Hyperparameter für Kampagne K_1	156
A.2	Vergleich der Hyperparameter für Kampagne K_2	156
A.3	Evaluierung synthetischer Testfälle anhand der Fläche unter der ROC-Kurve	157
A.4	Ausschlaggebende Signale ausgewählter Anomalien, erkannt durch Random Forest Verfahren, Teil 1	158
A.5	Ausschlaggebende Signale ausgewählter Anomalien, erkannt durch Random Forest Verfahren, Teil 2	159
B.1	Vergleich unterschiedlicher Neuronenanzahl	161

Literaturverzeichnis

- [AA13] ADHIKARI, R. ; AGRAWAL, R. K.: An Introductory Study on Time Series Modeling and Forecasting. In: *Computing Research Repository* abs/1302.6613 (2013), S. 1–67
- [ABR64] AIZERMAN, M.A. ; BRAVERMAN, E.A. ; ROZONOER, L.: Theoretical foundations of the potential function method in pattern recognition learning. In: *Automation and Remote Control* 25 (1964), S. 821–837
- [Abt16] ABTHOFF, T.: Big-Data-Technologien in der Fahrzeugentwicklung. In: *ATZelektronik* 11 (2016), Nr. 5, S. 44–51
- [ADA17] ADAC: *ADAC Untersuchung: Datenkrake Pkw*. https://www.adac.de/infotestrat/adac-im-einsatz/motorwelt/datenkrake_auto.aspx, 2017. – aufgerufen am 16.01.2017
- [AGA13] AMER, M. ; GOLDSTEIN, M. ; ABDENNADHER, S.: Enhancing One-class Support Vector Machines for Unsupervised Anomaly Detection. In: *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. New York, NY, USA : ACM, 2013 (ODD '13), S. 8–15
- [Ale15] ALEXANDER THAMM GMBH: *Predictive Maintenance in der Automobilbranche*. Business Application Research Center - BARC GmbH, <http://barc.de/uploads/search-file/bc596e0d5bce8804c73f4256a3d67f4fc7590525.pdf>, 2015. – aufgerufen am 27.03.2018

- [Alp10] ALPAYDIN, E.: *Introduction to Machine Learning*. 2. Cambridge, MA, USA : The MIT Press, 2010
- [Ant16] *Antrag der Fraktionen der CDU/CSU und SPD Intelligente Mobilität fördern –Die Chancen der Digitalisierung fuer den Verkehrssektor nutzen*. 2016
- [Ari06] ARISTOTELES: *Politik*. München : dtv Sachbuch, 2006. – übersetzt von Olof Gigon
- [AUD16a] AUDI AG: *Schwarmintelligenz/Car-to-X*. <https://www.audi-mediacyber.com/de/techday-connectivity-6597/schwarmintelligenzcar-to-x-6602>, 2016. – aufgerufen am 13.11.2017
- [AUD16b] AUDI AG: *Wie der Audi Daten sammelt*. <https://blog.audi.de/audi-pilotprojekt/>, 2016. – aufgerufen am 13.11.2017
- [Aut13] AUTOMOTIVEIT: *Big Data automotive- Eine Sonderedition von automtiveIT*. Media-Manufaktur GmbH, http://www.automotiveit.eu/wp-content/uploads/2013/09/Sonderheft_BigData.pdf, 2013. – aufgerufen am 10.11.2017
- [BB12] BERGSTRA, J. ; BENGIO, Y.: Random Search for Hyperparameter Optimization. In: *Journal of Machine Learning Research* 13 (2012), S. 281–305
- [BBCL14] BELLAS, A. ; BOUVEYRON, C. ; COTTRELL, M. ; LA-CAILLE, J.: Anomaly Detection Based on Confidence Intervals Using SOM with an Application to Health Monitoring. In: *Advances in Self-Organizing Maps and Learning Vector Quantization*. Cham : Springer International Publishing, 2014, S. 145–155

- [BCD16] BUYYA, R. ; CALHEIROS, R. N. ; DASTJERDI, A.V.: *Big Data: Principles and Paradigms*. 1. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2016
- [BDA13] BEKKAR, M. ; DJEMAA, H.K. ; ALITOUICHE, T.A.: Evaluation measures for models assessment over imbalanced data sets. In: *Journal Of Information Engineering and Applications* 3 (2013), Nr. 10
- [BGRS99] BEYER, K. S. ; GOLDSTEIN, J. ; RAMAKRISHNAN, R. ; SHAFT, U.: When Is "Nearest Neighbor" Meaningful? In: BEERI, C. (Hrsg.) ; BUNEMAN, P. (Hrsg.): *Proceedings of the 7th International Conference on Database Theory*. London, UK : Springer-Verlag, 1999 (ICDT '99), S. 217–235
- [Bis06] BISHOP, C. M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006
- [BJH17] BOSLER, M. ; JUD, C. ; HERZWURM, G.: Connected-Car-Services: eine Klassifikation der Plattformen für das vernetzte Automobil. In: *HMD Praxis der Wirtschaftsinformatik* 54 (2017), Nr. 6, S. 1005–1020
- [BKNS00] BREUNIG, M. M. ; KRIEGEL, H.-P. ; NG, R. T. ; SANDER, J.: LOF: Identifying Density-based Local Outliers. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA : ACM, 2000 (SIGMOD '00), S. 93–104
- [BL07] BOTTOU, L. ; LIN, C.-J.: Support Vector Machine Solvers. In: BOTTOU, L. (Hrsg.) ; CHAPELLE, O. (Hrsg.) ; DECOSTE, D. (Hrsg.) ; WESTON, J. (Hrsg.): *Large Scale Kernel Machines*. Cambridge, MA, USA : MIT Press, 2007, S. 301–320

- [Bog13] BOGON, T.: *Agentenbasierte Schwarmintelligenz*. Wiesbaden : Springer Fachmedien Wiesbaden, 2013
- [Box70] BOX, G. E. P. ; JENKINS, G. M. (Hrsg.): *Time series analysis forecasting and control*. San Francisco, CA, USA : Holden-Day, 1970
- [CAH16] CUI, Y. ; AHMAD, S. ; HAWKINS, J.: Continuous Online Sequence Learning with an Unsupervised Neural Network Model. In: *Neural Computation* 28 (2016), Nr. 11, S. 2474–2504
- [can15] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling. 2015 (ISO 11898-1:2015(en)). – Forschungsbericht
- [CBK09] CHANDOLA, V. ; BANERJEE, A. ; KUMAR, V.: Anomaly Detection: A Survey. In: *ACM Computing Surveys* 41 (2009), Nr. 3, S. 15:1–15:58
- [CFP98] COTTRELL, M. ; FORT, J.C. ; PAGÈS, G.: Theoretical aspects of the SOM algorithm. In: *Neurocomputing* 21 (1998), Nr. 1, S. 119 – 138
- [CGLT15] CARPINONE, A. ; GIORGIO, M. ; LANGELLA, R. ; TESTA, A.: Markov chain modeling for very-short-term wind power forecasting. In: *Electric Power Systems Research* 122 (2015), S. 152 – 158
- [Cha09] CHANDOLA, V.: *Anomaly detection for symbolic sequences and time series data*, University of Minnesota, Diss., 2009
- [CHN⁺13] CONG, F. ; HAUTAKANGAS, H. ; NIEMINEN, J. ; MAZHELIS, O. ; PERTTUNEN, M. ; RIEKKI, J. ; RISTANIEMI, T.: Applying Wavelet Packet Decomposition and One-Class

- Support Vector Machine on Vehicle Acceleration Traces for Road Anomaly Detection. In: GUO, C. (Hrsg.); HOU, Z.-G. (Hrsg.); ZENG, Z. (Hrsg.): *Advances in Neural Networks – ISNN 2013*. Berlin, Heidelberg : Springer-Verlag, 2013, S. 291–299
- [CMS12] CIREGAN, D. ; MEIER, U. ; SCHMIDHUBER, J.: Multi-column deep neural networks for image classification. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, S. 3642–3649
- [CTM⁺15] CARVALHO, L. F. M. ; TEIXEIRA, C. H. C. ; MEIRA, W. ; ESTER, M. ; CARVALHO, O. ; BRANDAO, M. H.: A simple and effective method for anomaly detection in healthcare. In: *4th Workshop on Data Mining for Medicine and Healthcare, 2015 SIAM International Conference on Data Mining, Vancouver*, 2015
- [CV95] CORTES, C. ; VAPNIK, V.: Support-Vector Networks. In: *Machine Learning* 20 (1995), Nr. 3, S. 273–297
- [DCS14] DAU, H. A. ; CIESIELSKI, V. ; SONG, A.: Anomaly Detection Using Replicator Neural Networks Trained on Examples of One Class. In: DICK, G. (Hrsg.) u. a.: *Proceedings of the 10th International Conference on Simulated Evolution and Learning* Bd. 8886. Cham : Springer International Publishing, 2014 (SEAL 2014), S. 311–322
- [Dec13] DECKER, P.: Wege vom klassischen CAN zum verbesserten CAN FD. In: *Elektronik automotive* 4 (2013), S. 38–41
- [DOSB10] DREISEITL, S. ; OSL, M. ; SCHEIBBÖCK, C. ; BINDER, M.: Outlier Detection with One-Class SVMs: An Application to Melanoma Prognosis. In: *AMIA Annual Symposium 2010*, 2010, S. 172–176

- [DS04] DORIGO, M. ; STÜTZLE, T.: *Ant Colony Optimization*. Scituate, MA, USA : Bradford Company, 2004
- [EHPSX96] ESTER, M. ; H.-P.KRIEGEL ; SANDER, J. ; XU, X.: A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996 (KDD'96), S. 226–231
- [EL12] EBERT, C. ; LEDERER, D.: Evolution und Trends in der E/E-Der Weg zum schnellen, wirksamen und trotzdem flexiblen Agieren. In: *Automobil-Elektronik* (2012). http://www.automobil-elektronik.de/wp-content/uploads/sites/7/2012/12/AEL_06_2012_gesamt.pdf. – aufgerufen am 15.01.2018
- [Ert16] ERTEL, W.: *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*. 4. Wiesbaden : Springer Vieweg, 2016
- [Faw06] FAWCETT, T.: An Introduction to ROC Analysis. In: *Pattern Recogn. Lett.* 27 (2006), Nr. 8, S. 861–874
- [FJ14] FULCHER, B. D. ; JONES, N. S.: Highly Comparative Feature-Based Time-Series Classification. In: *IEEE Transactions on Knowledge and Data Engineering* 26 (2014), Nr. 12, S. 3026–3037
- [FL17] FALCINI, F. ; LAMI, G.: Deep Learning in Automotive: Challenges and Opportunities. In: MAS, A. (Hrsg.) u. a.: *Software Process Improvement and Capability Determination*. Cham : Springer International Publishing, 2017, S. 279–288

- [FMS⁺17] FUGIGLANDO, U. ; MASSARO, E. ; SANTI, P. ; MILARDO, S. ; ABIDA, K. ; STAHLMANN, R. ; NETTER, F. ; RATTI, C.: Driving Behavior Analysis through CAN Bus Data in an Uncontrolled Environment. In: *Computing Research Repository* abs/1710.04133 (2017)
- [FPsS96] FAYYAD, U. ; PIATETSKY-SHAPIO, G. ; SMYTH, P.: From Data Mining to Knowledge Discovery in Databases. In: *AI Magazine* 17 (1996), S. 37–54
- [Fra98] FRASER, N.: *Neural Network Follies*. <https://neil.fraser.name/writing/tank/>, 1998. – aufgerufen am 28.05.2017
- [Gad17] GADATSCH, A.: Big Data – Datenanalyse als Eintrittskarte in die Zukunft. In: *Big Data für Entscheider: Entwicklung und Umsetzung datengetriebener Geschäftsmodelle*. Wiesbaden : Springer Fachmedien Wiesbaden, 2017, S. 1–10
- [Gae18] GAERTNER, C.: Der Fall der Automobilindustrie. In: GÄRTNER, C. (Hrsg.) ; HEINRICH, C. (Hrsg.): *Fallstudien zur Digitalen Transformation*. Wiesbaden : Gabler Verlag, 2018, S. 1–35
- [GBC16] GOODFELLOW, I. ; BENGIO, Y. ; COURVILLE, A.: *Deep Learning*. Cambridge, MA, USA : MIT Press, 2016. – <http://www.deeplearningbook.org>
- [GD12] GOLDSTEIN, M. ; DENGEL, A.: Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm. In: WÖLFL, Stefan (Hrsg.): *35th German Conference on Artificial Intelligence*. Berlin, Heidelberg : Springer-Verlag, 9 2012, S. 59–63
- [GFGS06] GRAVES, A. ; FERNÁNDEZ, S. ; GOMEZ, F. ; SCHMIDHUBER, J.: Connectionist Temporal Classification: Labelling

- Unsegmented Sequence Data with Recurrent Neural Networks. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA : ACM, 2006 (ICML '06), S. 369–376
- [GPTM10] GENUER, R. ; POGGI, J.-M. ; TULEAU-MALOT, C.: Variable Selection Using Random Forests. In: *Pattern Recognition Letters* 31 (2010), Nr. 14, S. 2225–2236
- [Gru69] GRUBBS, F. E.: Procedures for Detecting Outlying Observations in Samples. In: *Technometrics* 11 (1969), Nr. 1, S. 1–21
- [Gru16] GRUNDHOFF, S.: *Audi Auto der Zukunft: Für den Kunden von morgen*. Automobil Produktion, <https://www.automobil-produktion.de/hersteller/neue-modelle/audi-testet-das-vernetzte-auto-der-zukunft-fuer-den-kunden-von-morgen-118.html>, 2016. – aufgerufen am 20.03.2018
- [GU16] GOLDSTEIN, M. ; UCHIDA, S.: A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. In: *PLoS ONE* 11 (2016), 04, Nr. 4, S. 1–31
- [Gup15] GUPTA, A.: Big Data analysis using Computational Intelligence and Hadoop: A study. In: *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015, S. 1397–1401
- [Han17] HANDELSBLATT: *Spezialisten gesucht*. 3 2017. – aufgerufen am 18.04.2018
- [Har15] HARLOFF, T.: *Auto aus der Ferne gehackt: Der Fahrer ist machtlos*. Süddeutsche Zeitung, <http://www.sueddeutsche.de/auto/auto-aus-der-ferne-gehackt-der->

- fahrer-ist-machtlos-1.2577174, 2015. – aufgerufen am 26.01.2018
- [Haw80] HAWKINS, D.: *Identification of Outliers*. Dordrecht : Springer Netherlands, 1980
- [Her17] HERRMANN, W.: *Läutet Fog Computing das Ende der Cloud ein?* Computerwoche, <https://www.computerwoche.de/a/laeutet-fog-computing-das-ende-der-cloud-ein,3330809>, 2017. – aufgerufen am 10.02.2019
- [HHWB02] HAWKINS, S. ; HE, H. ; WILLIAMS, G. ; BAXTER, R.: Outlier Detection Using Replicator Neural Networks. In: KAMBAYASHI, Y. (Hrsg.) ; WINIWARTER, W. (Hrsg.) ; ARIKAWA, M. (Hrsg.): *Data Warehousing and Knowledge Discovery*. Berlin, Heidelberg : Springer-Verlag, 2002, S. 170–180
- [HKBL11] HUMMEL, T. ; KÜHN, M. ; BENDE, J. ; LANG, A.: Fahrerassistenzsysteme Ermittlung des Sicherheitspotenzials auf Basis des Schadensgeschehens der Deutschen Versicherer / Gesamtverband der Deutschen Versicherungswirtschaft e. V., Unfallforschung der Versicherer. 2011. – Forschungsbericht
- [HLCM12] HONG, S. J. ; LIM, W. Y. ; CHEONG, T. ; MAY, G. S.: Fault Detection and Classification in Plasma Etch Equipment for Semiconductor Manufacturing-Diagnostics. In: *IEEE Transactions on Semiconductor Manufacturing* 25 (2012), Nr. 1, S. 83–93
- [HS97] HOCHREITER, S. ; SCHMIDHUBER, J.: Long Short-Term Memory. In: *Neural Computation* 9 (1997), Nr. 8, S. 1735–1780
- [HS06] HINTON, G. E. ; SALAKHUTDINOV, R. R.: Reducing the dimensionality of data with neural networks. In: *Science* 313 (2006), Nr. 5786, S. 504–507

- [HSS08] HOFMANN, T. ; SCHÖLKOPF, B. ; SMOLA, A. J.: Kernel methods in machine learning. In: *The Annals of Statistics* 36 (2008), Nr. 3, S. 1171–1220
- [HTLN17] HO, Q.D. ; TWEED, D. ; LE-NGOC, T.: LTE-Advanced: An Overview. In: *Long Term Evolution in Unlicensed Bands*. Cham : Springer International Publishing, 2017 (Springer-Briefs in Electrical and Computer Engineering), S. 21–30
- [HVBL17] HALLAC, D. ; VARE, S. ; BOYD, S. P. ; LESKOVEC, J.: Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data. In: *Computing Research Repository* abs/1706.03161 (2017)
- [HXD03] HE, Z. ; XU, X. ; DENG, S.: Discovering Cluster-based Local Outliers. In: *Pattern Recognition Letters* 24 (2003), Nr. 9-10, S. 1641–1650
- [IA15] ISHWARAPPA ; ANURADHA, J.: A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology. In: *Procedia Computer Science* 48 (2015), S. 319 – 324. – International Conference on Computer, Communication and Convergence (ICCC 2015)
- [IBM14] IBM CORPORATION, IBM SOFTWARE: *Telematics, big data and analytics: Driving the automotive industry forward*. 2014
- [JTH⁺06] JIN, W. ; TUNG, A. K. H. ; HAN, J. ; ; WANG, W.: Ranking Outliers Using Symmetric Neighborhood Relationship. In: NG, W.-K. (Hrsg.) ; KITSUREGAWA, M. (Hrsg.) ; LI, J. (Hrsg.) ; CHANG, K. (Hrsg.): *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg : Springer-Verlag, 2006, S. 577–593

- [KG17] KENDALL, A. ; GAL, Y.: What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In: *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017
- [KHP14] KOPP, M ; HOLENA, M. ; PEVNY, T.: Interpreting and Clustering Outliers with Sapling Random Forests. In: *ITAT 2014. European Conference on Information Technologies - Applications and Theory 14*, 2014
- [KKSZ11] KRIEGEL, H.P. ; KROGER, P. ; SCHUBERT, E. ; ZIMEK, A.: Interpreting and Unifying Outlier Scores. In: *Proceedings of the 2011 SIAM International Conference on Data Mining*, 2011, S. 13–24
- [Köl16] KÖLLNER, C.: *Das Geschäft mit den Fahrzeugdaten.* Springer Fachmedien Wiesbaden, <https://www.springerprofessional.de/car-to-x/sicherheit-fahrbezogener-daten/das-geschaeft-mit-den-fahrzeugdaten/10868002>, 2016. – aufgerufen am 12.02.2019
- [KM10] KEOGH, E. ; MUEEN, A.: Curse of Dimensionality. In: SAMMUT, C. (Hrsg.) ; WEBB, G. I. (Hrsg.): *Encyclopedia of Machine Learning*. Boston, MA, USA : Springer US, 2010, S. 257–258
- [Koh90] KOHONEN, T.: The self-organizing map. In: *Proceedings of the IEEE* 78 (1990), Nr. 9, S. 1464–1480
- [Krz16] KRZANICH, B.: *Data is the New Oil in the Future of Automated Driving.* Intel, <https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/>, 2016. – aufgerufen am 14.11.2017
- [KSH12] KRIZHEVSKY, A. ; SUTSKEVER, I. ; HINTON, G. E.: ImageNet Classification with Deep Convolutional Neural Net-

- works. In: PEREIRA, F. (Hrsg.) ; BURGESS, C. J. C. (Hrsg.) ; BOTTOU, L. (Hrsg.) ; WEINBERGER, K. Q. (Hrsg.): *Advances in Neural Information Processing Systems 25*. Red Hook, NY, USA : Curran Associates, Inc., 2012, S. 1097–1105
- [Lia05] LIAO, T. W.: Clustering of time series data - a survey. In: *Pattern Recognition* 38 (2005), Nr. 11, S. 1857 – 1874
- [LLP07] LATECKI, L.J. ; LAZAREVIC, A. ; POKRAJAC, D.: Outlier Detection with Kernel Density Functions. In: PERNER, P. (Hrsg.): *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition*. Berlin, Heidelberg : Springer-Verlag, 2007 (MLDM '07), S. 61–75
- [LTZ08] LIU, F.T. ; TING, K.M. ; ZHOU, Z.-H.: Isolation Forest. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. Washington, DC, USA : IEEE Computer Society, 2008 (ICDM '08), S. 413–422
- [LTZ12] LIU, F.T. ; TING, K.M. ; ZHOU, Z.-H.: Isolation-Based Anomaly Detection. In: *ACM Transactions on Knowledge Discovery from Data* 6 (2012), Nr. 1, S. 3:1–3:39
- [Mar03] MARSLAND, S.: Novelty Detection in Learning Systems. In: *Neural Computing Surveys* 3 (2003), S. 157–195
- [MB88] MCLACHLAN., G. J. ; BASFORD., K. E.: *Mixture models: Inference and applications to clustering*. New York, NJ, USA : Marcel Dekker, 1988
- [MBB17] MERTENS, P. ; BARBIAN, D. ; BAIER, S.: Risiken. In: *Digitalisierung und Industrie 4.0 - eine Relativierung*. Wiesbaden : Springer Fachmedien Wiesbaden, 2017, S. 145–158

- [McK16] MCKINSEY & COMPANY: *Monetizing car data. New service business opportunities to create new customer benefits.* <http://www.mckinsey.com/industries/automotive-and-assembly/our-insights/monetizing-car-data>, 2016. – aufgerufen am 31.08.2017
- [Mer18] MERCEDES-BENZ: *Schutz in Unfallsituationen: Mercedes-Benz PRE-SAFE.* <https://www.mercedes-benz.com/de/mercedes-benz/innovation/schutz-in-unfallsituationen-mercedes-benz-pre-safe/>, 2018
- [Mit97] MITCHELL, T.M.: *Machine Learning.* 1. New York, NY, USA : McGraw-Hill, Inc., 1997
- [Mit16] MITTELSTRASS, J.: *Enzyklopädie Philosophie und Wissenschaftstheorie, Bd. 6: O - Ra.* Berlin Heidelberg : Springer-Verlag, 2016
- [MMHH⁺11] MOURÃO-MIRANDA, J. ; HARDOON, D. R. ; HAHN, T. ; MARQUAND, F. F. ; WILLIAMS, S. C. ; SHAWE-TAYLOR, J. ; BRAMMER, M.: Patient classification as an outlier detection problem: An application of the One-Class Support Vector Machine. In: *NeuroImage* 58 (2011), Nr. 3, S. 793 – 804
- [MS17] MARCHETTI, M. ; STABILI, D.: Anomaly detection of CAN bus messages through analysis of ID sequences. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, S. 1577–1583
- [MSPMG15] MARTÍ, L. ; SANCHEZ-PI, N. ; MOLINA, J. M. ; GARCIA, A. C. B.: Anomaly Detection Based on Sensor Data in Petroleum Industry Applications. In: *Sensors* 15 (2015), Nr. 2, S. 2774–2797

- [MT01] MARTINUS, D. ; TAX, J.: *One-class classification – Concept-learning in the absence of counter-examples*, Technische Universiteit Delft, dissertation, 2001
- [MYC⁺01] MANEVITZ, L. M. ; YOUSEF, M. ; CRISTIANINI, N. ; SHAWE-TAYLOR, J. ; WILLIAMSON, B.: One-Class SVMs for Document Classification. In: *Journal of Machine Learning Research* 2 (2001), S. 139–154
- [Nag17] NAGANATHAN, H.: *Energy Analytics for Infrastructure: An Application to Institutional Buildings*, Arizona State University, Diss., 2017
- [NAM01] NANOPOULOS, A. ; ALCOCK, R. ; MANOLOPOULOS, Y.: Feature-based Classification of Time-series Data. In: MASTORAKIS, N. (Hrsg.) ; NIKOLOPOULOS, S. D. (Hrsg.): *Information Processing and Technology*. Commack, NY, USA : Nova Science Publishers, Inc., 2001, S. 49–61
- [Ng11] NG, A.: Sparse Autoencoders / Stanford University. 2011. – Forschungsbericht
- [NM17] NGUYEN, K. ; METZ, E.: *Warum es mitten In Deutschland immer noch kein schnelles Netz gibt*. T-Online, http://www.t-online.de/digital/handy/id_82058324/darum-sind-in-dem-lte-netz-immer-noch-luecken.html, 2017. – aufgerufen am 22.12.2017
- [NMJ15] NARAYANAN, S.N. ; MITTAL, S. ; JOSHI, A.: Using Data Analytics to Detect Anomalous States in Vehicles. 2015. – Forschungsbericht
- [NMJ16] NARAYANAN, S.N. ; MITTAL, S. ; JOSHI, A.: OBD SecureAlert: An Anomaly Detection System for Vehicles. In: *IEEE Workshop on Smart Service Systems (SmartSys 2016)*, 2016

- [OF97] OLSHAUSEN, B. A. ; FIELD, D. J.: Sparse coding with an overcomplete basis set: A strategy employed by V1? In: *Vision Research* 37 (1997), Nr. 23, S. 3311 – 3325
- [OY18] OH, D. Y. ; YUN, I. D.: Residual Error Based Anomaly Detection Using Auto-Encoder in SMD Machine Sound. In: *Sensors* 18 (2018), Nr. 5
- [PAC15] PORTUGAL, I. ; ALENCAR, P. S. C. ; COWAN, D. D.: The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. In: *Computing Research Repository* abs/1511.05263 (2015)
- [Par62] PARZEN, E.: On Estimation of a Probability Density Function and Mode. In: *The Annals of Mathematical Statistics* 33 (1962), Nr. 3, S. 1065–1076
- [PCCT14] PIMENTEL, M. A. F. ; CLIFTON, D. A. ; CLIFTON, L. ; TARASSENKO, L.: Review: A Review of Novelty Detection. In: *Signal Processing* 99 (2014), S. 215–249
- [PK14] PEVNY, T. ; KOPP, M.: Explaining Anomalies with Sampling Random Forests. In: *ITAT 2014. European Conference on Information Technologies - Applications and Theory 14*, 2014
- [PP14] PAPPU, V. ; PARDALOS, P. M.: High-Dimensional Data Classification. In: ALESKEROV, F. (Hrsg.) ; GOLDENGORIN, B. (Hrsg.) ; PARDALOS, P. M. (Hrsg.): *Clusters, Orders, and Trees: Methods and Applications*. New York, NY, USA : Springer New York, 2014, S. 119–150
- [PY13] PAHUJA, D. ; YADAV, R.: Outlier Detection for Different Applications:Review. In: *International Journal of Engineering Research & Technology* 2 (2013), Nr. 3

- [Rei15] REIF, K.: Steuergerät. In: *Ottomotor-Management in Überblick*. Wiesbaden : Springer Fachmedien Wiesbaden, 2015, S. 198–213
- [RHM17] RICHTER, F. ; HARTKOPP, O. ; MATTFELD, D. C.: Automatic Defect Detection by Classifying Aggregated Vehicular Behavior. In: KRYSZKIEWICZ, M. (Hrsg.) u. a.: *Foundations of Intelligent Systems*. Cham : Springer International Publishing, 2017, S. 205–214
- [Rip96] RIPLEY, B. D.: *Pattern Recognition and Neural Networks*. Cambridge : Cambridge University Press, 1996
- [Rob18] ROBERT BOSCH GMBH: *Zentrales Gateway*. <https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/connectivity-solutions/central-gateway/>, 2018. – aufgerufen am 18.04.2018
- [RW18] REY, G.D. ; WENDER, K.F.: *Neuronale Netze - Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. Bd. 3. Bern : Hogrefe Verlag, 2018. – <http://www.neuronalesnetz.de/>, aufgerufen am 14.09.2017
- [SBM17] SUSTO, G. A. ; BEGHI, A. ; MCLOONE, S.: Anomaly detection through on-line isolation Forest: An application to plasma etching. In: *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2017, S. 89–94
- [Sch12] SCHAAL, H.-W.: Ethernet und IP im Kraftfahrzeug. In: *Elektronik automotive* 4 (2012), S. 38–41
- [SCZ⁺16] SHI, W. ; CAO, J. ; ZHANG, Q. ; LI, Y. ; XU, L.: Edge Computing: Vision and Challenges. In: *IEEE Internet of Things Journal* 3 (2016), Nr. 5, S. 637–646

- [SHK⁺14] SRIVASTAVA, N. ; HINTON, G. ; KRIZHEVSKY, A. ; SUTSKEVER, I. ; SALAKHUTDINOV, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: *Journal of Machine Learning Research* 15 (2014), Nr. 1, S. 1929–1958
- [SHY⁺18] SATO, D. ; HANAOKA, S. ; Y, Nomura ; TAKENAGA, T. ; MIKI, S. ; YOSHIKAWA, T. ; HAYASHI, N. ; ABE, O.: *A primitive study on unsupervised anomaly detection with an autoencoder in emergency head CT volumes*. 2018
- [SJDLA11] SPIEGEL, S. ; JAIN, B. J. ; DE LUCA, E.W ; ALBAYRAK, S.: Pattern Recognition in Multivariate Time Series: Dissertation Proposal. In: *Proceedings of the 4th Workshop on Workshop for Ph.D. Students in Information & Knowledge Management*. New York, NY, USA : ACM, 2011 (PIKM '11), S. 27–34
- [SL17] SCHUSTER, T. ; LIESEN, A.: *Statistik für Wirtschaftswissenschaftler: Ein Lehr- und Übungsbuch für das Bachelor-Studium*. Berlin Heidelberg : Springer, 2017 (Springer-Lehrbuch)
- [SMB10] SCHERER, D. ; MÜLLER, A. ; BEHNKE, S.: Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In: *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III*. Berlin, Heidelberg : Springer-Verlag, 2010 (ICANN'10), S. 92–101
- [SMG⁺17] STASS, S. ; MATHONY, H.-J. ; GAVANESCU, C. ; PASSMANN, C. ; HÖTZER, D.: Connected car – driver assistance sensors get connected. In: ISERMANN, R. (Hrsg.): *Fahrerassistenzsysteme 2017*. Wiesbaden : Springer Fachmedien Wiesbaden, 2017, S. 173–185

- [Spr18] SPRINGER FACHMEDIEN WIESBADEN: Unsere Automobile werden täglich schlauer. In: *ATZextra* (2018). – Ausgabe Sonderheft 4/2018
- [Ste16] STERN: *Für den Kunden von morgen*. <https://www.stern.de/auto/fahrberichte/audi-testet-das-vernetzte-auto-fuer-den-kunden-von-morgen-7034182.html>, 2016. – aufgerufen am 18.04.2018
- [Str13] STROHE, H.G.: *Lexikon Statistik*. Wiesbaden : Gabler Verlag, 2013
- [SWA14] STRICKER, K. ; WEGENER, R. ; ANDING, M.: *Big Data revolutioniert die Automobilindustrie*. Bain & Company Germany, http://www.bain.de/Images/Bain-Studie_Big%20Data%20revolutioniert%20die%20Automobilindustrie_FINAL_ES.pdf, 2014. – aufgerufen am 10.11.2017
- [SWS+00] SCHÖLKOPF, B. ; WILLIAMSON, R. C. ; SMOLA, A. J. ; SHAW-TAYLOR, J. ; PLATT, J. C.: Support Vector Method for Novelty Detection. In: SOLLA, S. A. (Hrsg.) ; LEEN, T. K. (Hrsg.) ; MÜLLER, K. (Hrsg.): *Advances in Neural Information Processing Systems 12*. Cambridge, MA, USA : MIT Press, 2000, S. 582–588
- [TAP14] TIAN, J. ; AZARIAN, M. H. ; PECHT, M.: Anomaly detection using self-organizing maps-based K-nearest neighbour Algorithm. In: *Proceedings of the European Conference of the Prognostics and Health Management Society*, 2014
- [TCFC02] TANG, J. ; CHEN, Z. ; FU, A. ; CHEUNG, D.: Enhancing Effectiveness of Outlier Detections for Low Density Patterns. In: *Proceedings of the 6th Pacific-Asia Conference on Ad-*

- vances in Knowledge Discovery and Data Mining*. London, UK : Springer-Verlag, 2002 (PAKDD '02), S. 535–548
- [TD04] TAX, D. M. J. ; DUIN, R. P. W.: Support Vector Data Description. In: *Machine Learning* 54 (2004), Nr. 1, S. 45–66
- [TD13] THEISSLER, A. ; DEAR, I.: An Anomaly Detection Approach to Detect Unexpected Faults in Recordings from Test Drives. In: *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 7 (2013), Nr. 7, S. 100 – 107
- [tel13] T-SYSTEMS INTERNATIONAL GMBH, FOCUS INDUSTRIES MARKETING: White Paper Big Data in Automotive: Chancen und Herausforderungen. 2013. – Forschungsbericht
- [TG04] TÓTH, L. ; GOSZTOLYA, G.: Replicator Neural Networks for Outlier Modeling in Segmental Speech Recognition. In: YIN, F.-L. (Hrsg.) ; WANG, J. (Hrsg.) ; GUO, C. (Hrsg.): *Advances in Neural Networks – ISNN 2004*. Berlin, Heidelberg : Springer-Verlag, 2004, S. 996–1001
- [The13] THEISSLER, A.: *Detecting Anomalies in Multivariate Time Series from Automotive Systems*, Brunel University, Diss., 2013
- [The14] THEISSLER, A.: Anomaly detection in recordings from in-vehicle networks. In: *Proceedings of Big Data Applications And Principles*, 2014
- [Tib94] TIBSHIRANI, R.: Regression Shrinkage and Selection Via the Lasso. In: *Journal of the Royal Statistical Society, Series B* 58 (1994), S. 267–288
- [TLJ16] TAYLOR, A. ; LEBLANC, S. ; JAPKOWICZ, N.: Anomaly Detection in Automobile Control Network Data with Long

- Short-Term Memory Networks. In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016, S. 130–139
- [TS92] TERRELL, G. R. ; SCOTT, D. W.: Variable Kernel Density Estimation. In: *The Annals of Statistics* 20 (1992), Nr. 3, S. 1236–1265
- [Tuk77] TUKEY, J. W. ; TUKEY, J. W. (Hrsg.): *Exploratory Data Analysis*. Reading, Mass., USA : Addison-Wesley, 1977 (Behavioral Science: Quantitative Methods)
- [Ver15] VERBAND DER AUTOMOBILINDUSTRIE E.V.: Automatisierung - Von Fahrerassistenzsystemen zum automatisierten Fahren / Verband der Automobilindustrie e.V. 2015. – Forschungsbericht. – <https://www.vda.de/dam/vda/publications/2015/automatisierung.pdf>, aufgerufen am 18.04.2018
- [Ver17] VERBAND DER AUTOMOBILINDUSTRIE E.V.: Zugang zum Fahrzeug und zu im Fahrzeug generierten Daten / Verband der Automobilindustrie e.V. 2017. – Forschungsbericht. – <https://www.vda.de/de/services/Publikationen/zugang-zum-fahrzeug-und-zu-im-fahrzeug-generierten-daten.html>, aufgerufen am 15.01.2018
- [VF11] VAN FLEET, P.: *Discrete Wavelet Transformations: An Elementary Approach with Applications*. Hoboken, NJ, USA : John Wiley and Sons, Inc., 2011
- [VK18] VÖLKL, K. ; KORB, C.: *Deskriptive Statistik: Eine Einführung für Politikwissenschaftlerinnen und Politikwissenschaftler*. Wiesbaden : Springer Fachmedien Wiesbaden, 2018 (Springer-Lehrbuch)

- [Vol18] VOLKSWAGEN AG: *Volkswagen entwickelt größtes digitales Ökosystem der Autoindustrie*. <https://www.volkswagen.com/de/news/stories/2018/08/volkswagen-develops-the-largest-digital-ecosystem-in-the-automot.html>, 2018. – aufgerufen am 27.09.2018
- [VWB⁺16] VARGHESE, B. ; WANG, N. ; BARBHUIYA, S. ; KILPATRICK, P. ; NIKOLOPOULOS, D. S.: Challenges and Opportunities in Edge Computing. In: *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, 2016, S. 20–26
- [Win17] WINZKER, M.: Bussysteme in der Automobiltechnik. In: *Elektronik für Entscheider: Grundwissen für Wirtschaft und Technik*. Wiesbaden : Springer Fachmedien Wiesbaden, 2017, S. 175–179
- [WKSZ18] WEBER, M. ; KLUG, S. ; SAX, E. ; ZIMMER, B.: Embedded Hybrid Anomaly Detection for Automotive CAN Communication. In: *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*, 2018
- [WVC⁺11] WANG, C. ; VISWANATHAN, K. ; CHOUDUR, L. ; TALWAR, V. ; SATTERFIELD, W. ; SCHWAN, K.: Statistical techniques for online anomaly detection in data centers. In: *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, 2011, S. 385–392
- [WZG⁺18] WANG, C. ; ZHAO, Z. ; GONG, L. ; ZHU, L. ; LIU, Z. ; CHENG, X.: A Distributed Anomaly Detection System for In-Vehicle Network Using HTM. In: *IEEE Access* 6 (2018), S. 9091–9098
- [Yan11] YAN, X.: Multivariate outlier detection based on self-organizing map and adaptive nonlinear map and its applica-

- tion. In: *Chemometrics and Intelligent Laboratory Systems* 107 (2011), Nr. 2, S. 251–257
- [YD03] YEUNG, D.Y. ; DING, Y.: Host-based intrusion detection using dynamic and static behavioral models. In: *Pattern Recognition* 36 (2003), Nr. 1, S. 229 – 243
- [Ye00] YE, N.: A markov chain model of temporal behavior for anomaly detection. In: *Proceedings of the Workshop on Information Assurance and Security*. West Point, NY, USA : United States Military Academy, 2000, S. 171–174
- [ZCW⁺17] ZHANG, M. ; CHEN, C. ; WO, T. ; XIE, T. ; BHUIYAN, M. Z. A. ; LIN, X.: SafeDrive: Online Driving Anomaly Detection From Large-Scale Vehicle Data. In: *IEEE Transactions on Industrial Informatics* 13 (2017), Nr. 4, S. 2087–2096
- [Zha13] ZHANG, J.: Advancements of Outlier Detection: A Survey. In: *ICST Transactions on Scalable Information Systems* 13 (2013), Nr. 1, S. 1–26
- [ZPH98] ZHANG, G. ; PATUWO, B. E. ; HU, M. Y.: Forecasting with artificial neural networks: The state of the art. In: *International Journal of Forecasting* 14 (1998), Nr. 1, S. 35 – 62
- [ZSK12] ZIMEK, A. ; SCHUBERT, E. ; KRIEGEL, H.-P.: A survey on unsupervised outlier detection in high-dimensional numerical data. In: *Statistical Analysis and Data Mining* 5 (2012), Nr. 5, S. 363–387

Publikationen:

HOFMOCKEL, J. ; RICHTER, F., SAX, E.: Automatic Defect Detection by One-Class Classification on Raw Vehicle Sensor Data. In: *Foundations of Intelligent Systems* 23, 2017, S. 378 –384

HOFMOCKEL, J., SAX, E.: Isolation Forest for Anomaly Detection in Raw Vehicle Sensor Data. In: *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems* 1, 2018, S. 411 – 416

HOFMOCKEL, J., MASINO J., THUMM, J., SAX, E., GAUTERIN, F.: Multiple vehicle fusion for a robust road condition estimation based on vehicle sensors and data mining. In: *Cogent Engineering* 5 (2018)

DÜSTERHÖFT, F. , HEGER, T., HOFMOCKEL, J., KLEE, P.-A., KLÖPFER, C., LAUBIS, K., SCHMIDT-SAUTTER, M., MASINO, J.: Fahrzeugsensoren als Echtzeit-Informationsquelle für die Qualität der Straßenverkehrsinfrastruktur. In: *Straße und Autobahn* 4 (2018)

Betreute Studienarbeiten:

PÖSCHL M.: Analyse und Interpretation von Fahrzeugdaten mit Unterstützung durch Maschinelles Lernen. Bachelorarbeit. Ostbayerische Technische Hochschule Regensburg, 2016

EJAZUDDIN, M.: Environmental Entropy for Automated Driving. Masterarbeit. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2018

SCHREIBER D.: Innovative Softwaretechnologien für biologisch inspirierte Maschinenintelligenz. Praktikum. RWTH Aachen University, 2018