

Klassifizierung von Schienendefekten aus Bild- und Beschleunigungsdaten mittels maschinellen Lernens

Masterarbeit

Alexander Lähns
382144

8. Juli 2023

Betreuer: Dr. Benjamin Baasch
Daniel Weber

Technische Universität Berlin
Fakultät IV – Elektrotechnik und Informatik
Institut für Energie- und Automatisierungstechnik
Fachgebiet Elektronische Mess- und Diagnosetechnik

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt gegenüber der Fakultät IV der Technischen Universität Berlin, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein. Ich versichere, dass diese Arbeit oder wesentliche Teile dieser Arbeit nicht bereits dem Leistungserwerb in einer anderen Lehrveranstaltung zugrunde lagen.

Datum, Ort

Unterschrift

Abstract

This research investigates the classification of railway defects with methods of machine learning. The goal is to develop an effective and reliable system that can support or replace the outdated traditional methods. To achieve this, the system has to be capable of classifying different types and degrees of wear of railway defects. Various approaches of supervised machine learning are evaluated and compared to find the best combinations. The data utilized in this study includes axle box acceleration measurements and video footage of the rails, both of which were acquired during a test drive. It is found that an initial binary classification to separate the damaged from the good samples is the best method to conquer the significant imbalanced classes. A second CNN is then used only to separate the damaged samples from each other. The results show that a system of multiple CNNs is capable of classifying railway defects. In conclusion, this paper represents an improvement upon previously established methods for wear detection in railway systems and marks new problems that should be addressed in the future.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aktuelle Methoden zur Untersuchung von Schienen	1
1.2	Aufbau des Messsystems	2
1.3	Bereitgestellter Datensatz	4
1.4	Verwandte Arbeiten	6
1.5	Ansatz	7
2	Theorie	8
2.1	Rad Schiene Kontakt	8
2.2	Arten von Schienendefekten	9
2.2.1	Rollkontaktermüdung	10
2.2.2	Verriffelung	11
2.3	Maschinelles Lernen	13
2.3.1	Metriken	13
2.3.2	Merkmalsextraktion	14
2.3.3	Entscheidungsbäume	17
2.3.4	Boosting	19
2.3.5	Neuronale Netze	21
3	Methoden - Durchführung	30
3.1	Datenaufbereitung	30
3.1.1	Erstellen der Datensätze	31
3.1.2	Merkmalsextraktion	32
3.2	Untersuchungen mit dem Random Forest Klassifikator	33
3.3	Untersuchungen mit dem AdaBoost Klassifikator	34
3.4	Untersuchungen mit CNNs	35
4	Auswertung	37
4.1	Auswertung Random Forest	37
4.1.1	Benchmarkversuch	37
4.1.2	Versuch mit verschobenen Labeln	38
4.1.3	Versuche mit dem dritten und vierten Datensatz	39
4.1.4	erste binäre Klassifikation	40
4.1.5	weiterführende binäre Klassifikationen	41
4.2	Auswertung AdaBoost	42
4.2.1	Vergleichsversuch zwischen AdaBoost und Random Forest	42
4.2.2	Binäre Klassifikation mit dem AdaBoost Klassifikator	43
4.3	Auswertung CNN	44
4.3.1	Probleme mit dem Datensatz	47
5	Fazit	50

6	Ausblick	51
	Literatur	52

1 Einleitung

Allein in Deutschland beläuft sich das Streckennetz auf 38 400 km. Dabei hat sich die Belastung der Strecken bzw. Schienen durch eine erhöhte Verkehrsleistung (50 % im Personen- und 90 % im Güterverkehr seit 1994) maßgeblich gesteigert [30]. Um trotz dieser hohen Belastung ein sicheres Befahren des Schienennetzes zu gewährleisten, ist es nötig, dieses in regelmäßigen Abständen zu kontrollieren, um mögliche Schäden frühzeitig erkennen zu können.

Da die aktuellen Methoden zur Schadenserkenkung bereits veraltet sind und dem steigendem Wartungsbedarf nicht mehr gerecht werden, soll nach neuen und besseren Methoden gesucht werden. Da dieses Problem nicht nur Deutschland betrifft, sondern weltweit auftritt, hat Strukton Rail (Niederlande) im Jahr 2019 das Projekt IN2SMART2 (intelligent Innovative Smart of Assets by integRated Technologies 2) gestartet. Im Zuge dieses Projektes sollen Lösungen gefunden werden, um dem steigenden Wartungsaufwand gerecht zu werden. Um dieses Ziel zu erreichen, arbeiten viele verschiedene Firmen und Institute, wie z.B. das DLR oder Siemens, mit Strukton Rail zusammen.

In dieser Arbeit soll nun nach einer Methode gesucht werden, die mittels künstlicher Intelligenz oder maschinellen Lernens zuverlässig beschädigte Schienen erkennen kann. Für die Untersuchung wird von Strukton Rail ein Datensatz, welcher Beschleunigungsdaten und Videodaten der zu untersuchenden Schienen enthält, bereitgestellt. Als Vorläufer dieser Arbeit dient die Doktorarbeit [1], in welcher die Möglichkeiten des unüberwachten Lernens untersucht werden.

1.1 Aktuelle Methoden zur Untersuchung von Schienen

Aktuell werden Schienen hauptsächlich mit drei verschiedenen Methoden untersucht. Die erste und älteste Methode stellt dabei die visuelle Untersuchung durch einen Mitarbeiter dar. Diese Methode ist sehr zeitaufwendig und daher auch sehr kostenintensiv. Außerdem ist das Urteil des Mitarbeiters subjektiv und daher anfällig für Fehleinschätzungen [23].

Eine Alternative stellt hier die Untersuchung mittels Ultraschall dar. Im Gegensatz zur visuellen Untersuchung kann mittels Ultraschall auch das Innere der Schiene untersucht werden. Diese Untersuchung wird mithilfe eines Messtrolleys (Abbildung 1.1) durchgeführt und kann so beide Gleise gleichzeitig untersuchen. Daraus ergibt sich bereits eine erhebliche Zeitersparnis im Vergleich zur visuellen Untersuchung. Die dritte Methode ist das sogenannte Wirbelstromverfahren. Dieses Verfahren wird mit einem Messzug durchgeführt und kann auch mit dem Ultraschallverfahren kombiniert werden. Um eine prüftechnische Anerkennung zu erhalten, verlangt z.B. die DB von Messzügen mit dem Wirbelstromverfahren, eine Strecke mit einer Geschwindigkeit von 60 km h^{-1} prüfen zu können. Ein Nachteil des Wirbelstromverfahrens ist, dass Messzüge in der Regel nur 4 Sonden gleichzeitig zur Verfügung stellen und so nur ein Teil der Schienenoberfläche gemessen

werden kann (Abbildung 1.2).



Bild 1.1: Darstellung des Ultraschallverfahrens
Quelle: [13]

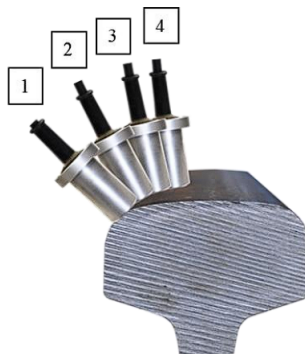


Bild 1.2: Darstellung des Wirbelstromverfahrens
Quelle: [5]

1.2 Aufbau des Messsystems

Im folgenden soll das Messsystem, welches zur Aufnahme der Daten verwendet wurde beschrieben werden. Das gesamte Messsystem befindet sich auf dem in Abbildung 1.6 gezeigten Waggon und umfasst verschiedene unabhängig voneinander arbeitende Messinstrumente. Dazu gehören z.B. diverse Kameras pro Schienenseite, ein GPS-Sensor und die sogenannten axle box acceleration Sensoren (ABA-Sensoren). Die ABA-Sensoren sind Beschleunigungssensoren, welche wie in Abbildung 1.3 dargestellt, an die Achse eines Rades angebracht werden. Mithilfe dieser Sensoren können Auslenkungen der Räder durch etwaige defekte Schienen gemessen werden. Gemessen werden pro Sensor Auslenkungen in den Richtungen x , y und z . Wie in Abbildung 1.3 dargestellt, spiegelt der Kanal z hier die vertikale Auslenkung wieder. Der Kanal in x -Richtung misst die Auslenkungen in

Fahrt- oder Gegenfahrtrichtung. Die seitlichen Auslenkungen vertikal zur Fahrtrichtung werden von den Sensoren des y-Kanals aufgezeichnet. Pro Schienenseite steht jeweils ein ABA-Sensor zur Verfügung, welcher mit einer Abtastfrequenz von 25.6 kHz arbeitet.

Die Kameras des Messsystems sind wie in Abbildung 1.5 dargestellt angeordnet und nehmen dauerhaft Bilder von den Schienen auf. In Abbildung 1.5 ist zu erkennen, dass es pro Schienenseite jeweils fünf verschiedene Kameras gibt. Diese sind in den Channeln 60 bis 65 und 70 bis 75 aufgeteilt. Da für die Bestimmung der Label nur die Daten aus Channel 61 und Channel 71 verwendet werden, sollen alle anderen Channel im folgenden nicht weiter betrachtet werden. Will man die ABA Daten und die Label aus den Videodaten übereinanderlegen, muss noch ein Abstand (wie in Abbildung 1.4 dargestellt) von 3.15 m zwischen dem Sensor und der Kamera berücksichtigt werden. Da der GPS-Sensor selbsterklärend ist und die anderen Messsysteme für diese Arbeit nicht relevant sind, soll an dieser Stelle nicht weiter auf diese eingegangen werden. Bereitgestellt werden die Messinstrumente alle von einem Waggon, welcher von einer Antriebslok bewegt wird. Der Waggon ist in Abbildung 1.6 dargestellt, mit einem roten Pfeil als Markierung für die ABA-Sensoren.

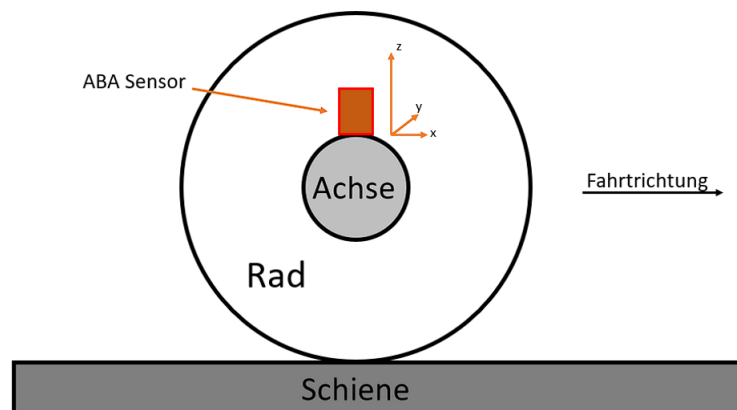


Bild 1.3: Schematische Darstellung ABA-Sensor

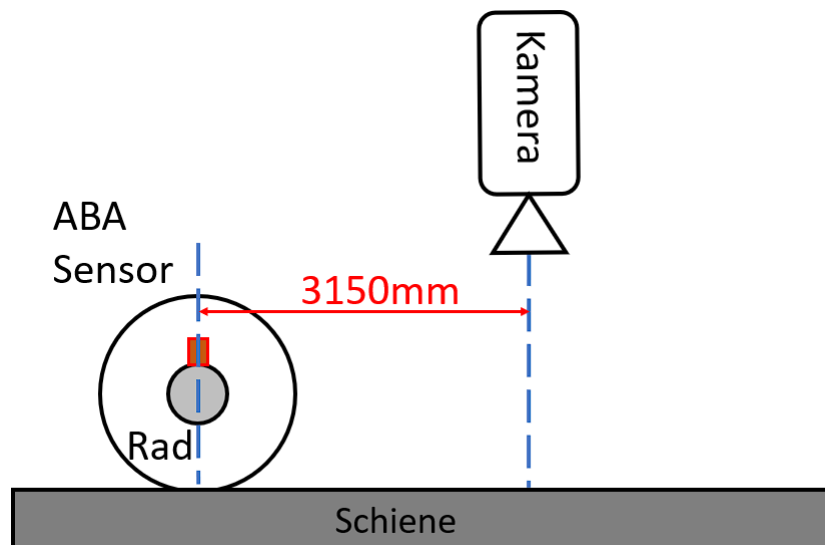


Bild 1.4: Anordnung ABA-Sensor und Kamera

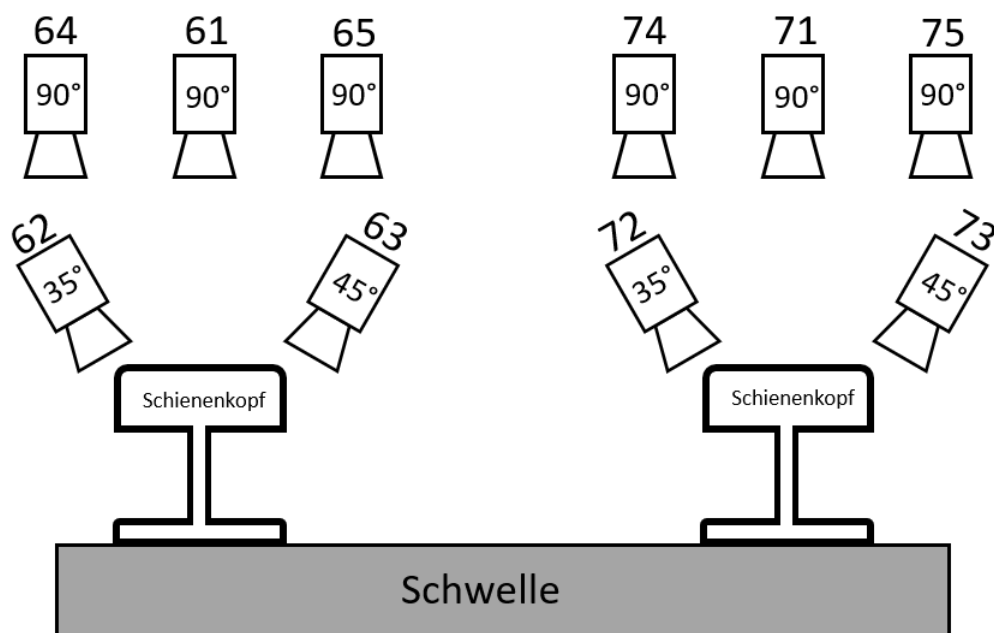


Bild 1.5: Schematischer Aufbau der Kameras

1.3 Bereitgestellter Datensatz

Da der Datensatz aus Daten verschiedener Messsysteme besteht und diese nicht direkt miteinander zusammenhängen, soll an dieser Stelle genauer auf die Daten eingegangen werden. Eine Besonderheit ist z.B., dass alle Daten beim Speichern zusätzlich mit einem Counter versehen werden. Dabei werden die ABA-Daten mit einem sogenannten internen Counter belegt, wobei ein interner Counter einen Abstand von 0.25 mm zum nächsten internen Counter angibt. Die Video- und GPS-Daten erhalten im Gegensatz dazu einen so-



Bild 1.6: Eurailscout Messwagen mit ABA-Sensoren

Quelle: [17]

genannten externen Counter, dieser beschreibt jeweils einen Abstand von 1 mm. Die im Datensatz enthaltenen Bilder bestehen jeweils aus 10.05 m zusammengefügt Videoaten und dienen als Grundlage für das Erstellen der Label der Schienenoberfläche. Dabei soll noch erwähnt werden, dass aufeinanderfolgende Bilder sich jeweils um 0.5 m überlappen.

Da die Label bereits vor dieser Arbeit aus den Bildern extrahiert wurden, soll darauf an dieser Stelle nicht weiter eingegangen werden. Es soll nur kurz gezeigt werden, in welcher Form die Label im Datensatz zur Verfügung stehen.

ID	ext. Counter	Pixel Start	Pixel Ende	Label Grad	Label Typ
61	27799264	14991	15071	3	1
71	28162164	12453	14075	3	4

Tabelle 1.1

Die Label sind nach dem Schema aus Tabelle 1.1 in einer .csv-Datei verfügbar. Dabei ist zu beachten, dass die ID bestimmt auf welche Schienenseite bzw. welchen Kamerachannel die Label bezogen sind. Der externe Counter gibt an, an welcher Stelle das 10.05 m lange Schienenstück beginnt. Des Weiteren ist die genaue Position des Labels auf dem Schienenstück in Pixeln angegeben. Wobei 2 Pixel einem externen Counter, also 1 mm, entsprechen. Der Grad und der Typ geben an, wie stark die Beschädigung ist und von welcher Art diese ist. Dabei gilt die folgende Aufschlüsselung (Tabelle 1.2 und 1.3):

Eine schwere Beschädigung vom Typ corrugation erhält nach dieser Aufschlüsselung also das Label 1_3.

Grad der Beschädigung	Label
Schwer	1
Mittel	2
Leicht	3
sonstige	4
Keine	0

Tabelle 1.2

Art der Beschädigung	Label
Squat	1
Headcheck	2
Corrugation	3
STUDs	4
Unbekannt	5

Tabelle 1.3

1.4 Verwandte Arbeiten

Als Vorläufer dieser Arbeit kann die Doktorarbeit [1] betrachtet werden. Dabei wurden die Daten mittels Anomaliedetektion in Form eines isolation forest Klassifikators untersucht. Anschließend wurden die Ergebnisse mithilfe der Label überprüft und in einer Tabelle (1.4) festgehalten. Die Tabelle zeigt den precision- und recall-Wert, dabei werden erst alle Merkmale einzeln betrachtet und in der letzten Zeile in Kombination miteinander. Im Bereich der precision konnten bereits Ergebnisse mit einem Score bis zu 83 % erzielt werden. Vor allem die Kurtosis und Skewness liefern an dieser Stelle hohe Werte. Im Bereich des recall-Score kann der beste Wert erreicht werden, wenn alle Merkmale gleichzeitig betrachtet werden. Mit 63 % ist der Wert trotzdem deutlich niedriger als im Bereich der precision. Insgesamt bestätigen die Ergebnisse aber das Potenzial des Datensatzes und versprechen bessere Ergebnisse in weiterführenden Untersuchungen.

features	precision	recall
RMS	0.66	0.33
kurtosis	0.82	0.47
skewness	0.83	0.45
peak-to-peak	0.73	0.46
crest factor	0.82	0.56
impulse factor	0.80	0.51
all features	0.74	0.63

Tabelle 1.4

In [26], [25] und [20] wird sich ausschließlich auf das erkennen von sogenannten Squats (Störungen) konzentriert. Dabei werden sowohl Simulationen als auch reale Daten verwendet, um das Erkennen solcher Squats möglichst zu optimieren.

1.5 Ansatz

Um möglichst viele Probleme zu adressieren und so optimale Ergebnisse zu erzielen, soll in dieser Arbeit ein inkrementeller Ansatz gewählt werden. Dabei werden zunächst einige Versuche durchgeführt, deren Ergebnisse und Erkenntnisse als Grundlage der weiterführenden Untersuchungen dienen sollen. Anschließend werden in weiteren Versuchen sowohl die Daten als auch die verwendeten Methoden angepasst. Dabei sollen zum einen die Erkenntnisse aus den Vorversuchen bestätigt und im Idealfall auch die Ergebnisse verbessert werden. Dieses Vorgehen wird solange wiederholt, bis das Endergebnis nicht mehr weiter verbessert werden kann oder den Anforderungen entspricht.

2 Theorie

In diesem Kapitel wird ein Überblick über die theoretischen Grundlagen gegeben, die für das Verständnis der Fragestellung und die Beantwortung der Forschungsfragen in dieser Arbeit von Bedeutung sind. Dabei werden grundlegende Konzepte, Theorien und Modelle, die in der Literatur diskutiert werden, zusammengefasst und betrachtet. Zunächst werden einige Grundlagen in Bezug auf den Kontakt zwischen Rad und Schiene diskutiert. Hierbei liegt der Fokus insbesondere auf den physikalischen Zusammenhängen zwischen Rad und Schiene, welche die Verwendung der ABA-Daten als valides Mittel zur Identifikation von defekten Schienen rechtfertigen. Anschließend werden klassische Schienendefekte betrachtet, wobei die Entstehung und Auswirkung allgemein erläutert wird. Da sowohl Klassifikationsalgorithmen in Form von Entscheidungsbäumen als auch neuronale Netze einen großen Teil dieser Arbeit einnehmen, werden deren Grundlagen genauer betrachtet. Hierbei wird insbesondere auf den Ansatz, den Aufbau und mögliche Probleme dieser Methoden eingegangen. Zusätzlich werden auch die Vorbereitungen und Möglichkeiten zur Bewertung der Klassifikation betrachtet.

2.1 Rad Schiene Kontakt

Um den physikalischen Zusammenhang zwischen Rad und Schiene besser zu erklären, wird in diesem Kapitel ein stark vereinfachtes Modell eines Zuges herangezogen. Darüber hinaus wird der Ansatz von [33] aus Kapitel 2 aufgegriffen, der den Zusammenhang zwischen Schienenfahrzeug und Schiene beschreibt.

Das vereinfachte Modell ist in Abbildung 2.1 dargestellt und zeigt einen Wagen bzw. Waggon und ein Rad als Punktmassen. Als vertikale Verbindung zwischen Rad und Waggon wird eine Feder verwendet. Auf den üblicherweise verwendeten Dämpfer wird hier verzichtet. Der Kontakt zwischen Rad und Schiene soll mit der Kontaktmechanik nach Hertz beschrieben werden und wird symbolisch mit einer Feder dargestellt. Die aus dem Kontakt resultierende Kraft F wirkt gleichermaßen auf das Rad und die Schiene. Obwohl dieser Ansatz sehr vereinfacht ist, ist er im Gebiet der Schienenfahrzeugtechnik weit verbreitet. Als Fahrtrichtung des Modells dient hier die x -Richtung, vertikale Auslenkungen werden in diesem Modell in z -Richtung dargestellt.

Mit der Bewegungsgleichung 2.1 für dieses System kann das Verhalten des Modells bei Auftreten einer externen Kraft $F(t)$ in die vertikale z -Richtung beschrieben werden. Die Matrizen M und K sind hier Diagonalmatrizen, welche die Massen M und Federkonstanten K des Waggons und des Rades beinhalten. Die externe Kraft $F(t)$ lässt sich wie in [33] mit der Hertzkonstante $k_{z,Hertz}$ und der Differenz zwischen den Auslenkungen des Rades z_{Rad} und der Schiene $z_{Schiene}$ bestimmen. Der Vektor $z(t)$ bzw. dessen zweite Ableitung $\ddot{z}(t)$ stellt die Auslenkung bzw. Beschleunigung der jeweiligen Komponenten zum Zeitpunkt t dar. An dieser Stelle wird deutlich, dass eine Auslenkung der Schiene ($z_{Schiene}$) maßgeblich

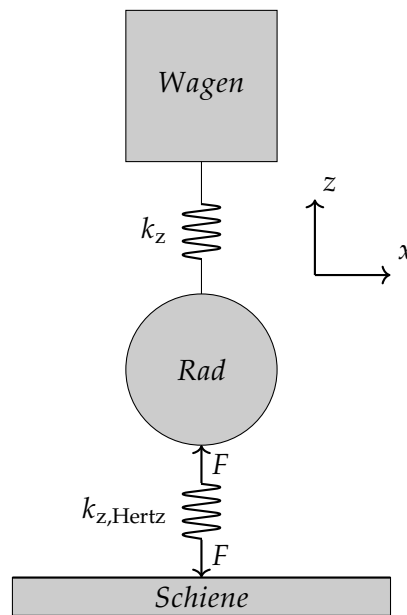


Bild 2.1: schematische Darstellung des Rad-Schiene Kontakts

die Kraft F beeinflusst und damit auch direkten Einfluss auf die Beschleunigung der Massen hat. Da die Räder Teil der Massematrix M sind, kann eine Messung ihrer Beschleunigung mittels ABA-Sensoren direkt mit dem Zustand der Schienen in Verbindung gebracht werden. Diese Behauptung gilt nicht nur für die Zusammenhänge in z -Richtung, sondern auch in x - und y -Richtung, da das Prinzip aus Bewegungsgleichungen und externer Kraft, wie in [10] beschrieben, für alle Richtungen angewendet werden kann.

$$[M]\{\ddot{z}(t)\} + [K]\{z(t)\} = \{F(t)\} \quad (2.1)$$

$$F(t) = k_{z,Hertz}(z_{\text{Rad}}(t) - z_{\text{Schiene}}(t)) \quad (2.2)$$

2.2 Arten von Schienendefekten

Im nächsten Teil dieses Kapitels sollen nun klassische Schienendefekte betrachtet werden. Hierbei handelt es sich um Schäden an der Schiene, die in der Regel durch äußere Einflüsse, wie z.B. Verschleiß oder Überlastung entstehen. An dieser Stelle soll noch erwähnt werden, dass es sich im Folgenden immer um Beschädigungen auf der Schienenoberfläche handelt. Da der Datensatz verschiedene Arten von Beschädigungen (1.2) bereitstellt, sollen diese im Folgenden genauer behandelt werden. Dabei liegt ein besonderer Fokus auf der potenziellen Gefahr, die von den Beschädigungen ausgeht, sowie dem Instandhaltungsprozess. Diese Aspekte sind von großer Bedeutung, um später eine Aussage über die Qualität der Klassifikation treffen zu können. Insbesondere ist es wichtig, potenziell gefährliche Beschädigungen zuverlässig zu klassifizieren. Auch der Grad der Beschädigung kann ein wichtiges Kriterium sein, da die jeweiligen Instandhaltungsmaßnahmen davon abhängen.

2.2.1 Rollkontaktermüdung

Eine der häufigsten Arten von Beschädigungen im Bahnbetrieb ist die Rollkontaktermüdung, die durch den Kontakt zwischen Rad und Schiene und die daraus resultierenden Beanspruchungen des Materials entsteht. Die bekanntesten Formen dieser Beschädigungen sind Headchecks und Squats.

Headchecks zeichnen sich typischerweise durch kleine Risse an der Fahrkante aus und treten häufig in langen Kurven auf. Diese Risse weisen oft einen Abstand von nur wenigen Millimetern auf und bilden sich auf der Schienenoberfläche meist im 45° Winkel aus. In Abbildung 2.2 ist ein Schienenstück mit den für Headchecks typischen Rissen dargestellt. Innerhalb des Materials weisen Headchecks üblicherweise einen Ausbreitungswinkel von 15° auf, was in dieser Form keine weitere Gefahr darstellt. Wenn die Risse jedoch sehr nahe beieinanderliegen, können sie sich verbinden und zu gefährlichen Ausbrüchen in den Schienen führen. Untersuchungen haben gezeigt, dass flache Risse sich bei fortlaufender Materialbeanspruchung sprunghaft in senkrechte Richtung ausbreiten können ([6]).

Im Gegensatz zu den Headchecks treten Squats initial nicht als Risse auf, sondern als kleine Einsenkungen am Schienenkopf. Squats sind vor allem, wie in Abbildung 2.3 dargestellt, an den schwarzen bzw. dunklen Punkten zu erkennen. Im fortschreitenden Prozess vergrößern sich diese Punkte und entwickeln sich schlussendlich zu halbkreis- oder v-förmigen Rissen weiter. Genau wie bei den Headchecks sind diese Risse zu Beginn eher flach. Daraus resultieren schlussendlich dieselben potenziellen Gefahren wie bei den Headchecks. Zusätzlich tritt bei den Squats noch das Problem auf, dass das Rad den Schienenkopf aufgrund der Einsenkung nicht mehr berührt ([34]). Plötzliche Umverteilungen der Kräfte in der Gleisanlage können durch Rollkontaktermüdung entstehen, was weitere Beschädigungen an anderen Stellen verursachen kann. Es ist daher wichtig, potenzielle Gefahrenquellen zu erkennen und Instandhaltungsprozesse sorgfältig durchzuführen. Insbesondere sollten Risse mit einer Tiefe von 0.2 bis 0.5 Millimetern im Auge behalten werden, da in diesem Bereich ein stabiles und kalkulierbares Risswachstum angenommen werden kann ([6]). Daher sollte auch die Instandhaltung in diesem Bereich durchgeführt werden, was betont, wie wichtig eine Früherkennung dieser Beschädigungen ist.

Eine weitere Art von Rollkontaktermüdung sind die sogenannten STUDs, auch als squar-tige Defekte bezeichnet, da sie optische Merkmale mit Squats teilen. Die Entstehung von STUDs ist noch nicht vollständig geklärt, aber es wird angenommen, dass sie keine Schienenbrüche verursachen ([11]). Damit wären STUDs deutlich harmloser als Squats oder Headchecks.

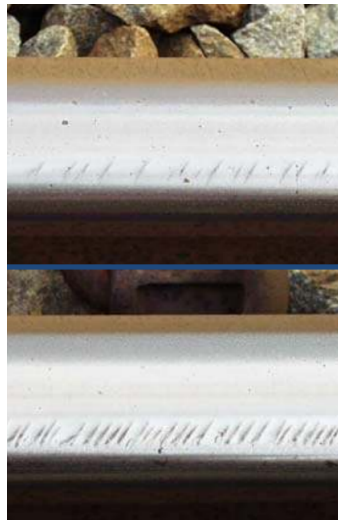


Bild 2.2: Darstellung von headchecks
Quelle: [31]



Bild 2.3: Darstellung von squats
Quelle: [31]

2.2.2 Verriffelung

Eine weitere Art von Beschädigungen wird häufig durch den Begriff Verwellung oder Verriffelung (englisch corrugation) zusammengefasst. Bei diesen Beschädigungen liegt keine Materialermüdung durch den Rollkontakt vor, sondern eine Materialverformung durch Schlupf. Schlupf kann beim Bahnbetrieb z.B. durch das Beschleunigen oder Bremsen der Schienenfahrzeuge entstehen, aber auch durch die starre Verbindung der Räder in Kombination mit den unterschiedlichen Kurvenradien der linken und rechten Schiene. Durch diesen Schlupf entstehen meist mehr oder weniger periodisch auftretende Unebenheiten auf der Schienenoberfläche [34] (Trackopedia und voestalpine Schienen, 2020). Diese Unebenheiten bilden im fortschreitenden Prozess verhärtete Wellen aus. Diese Wellen verursachen hohe dynamische Kräfte zwischen Rad und Schiene und führen so zu verstärkter Abnutzung der beteiligten Komponenten. Auch die Geräuschentwicklung kann bei der Überfahrt stark ansteigen und so zu einem Problem für den Bahnbetrieb in Städten oder Siedlungsgebieten werden. Obwohl auch diese Art von Beschädigung ernst zu nehmende Probleme und Folgeschäden mit sich bringt, ist sie für den gesamten Bahnbetrieb weniger gefährlich als Headchecks oder Squats.

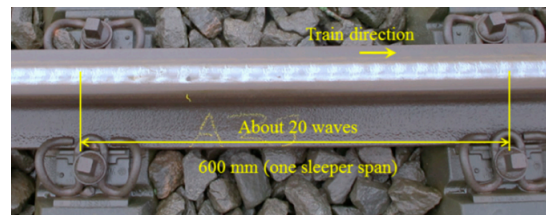


Bild 2.4: Darstellung von Schlupfwellen
Quelle: [19]

2.3 Maschinelles Lernen

Der Begriff des maschinellen Lernens bezieht sich auf eine Vielzahl von Methoden und Ansätzen, die automatisch Muster in einem Datensatz erkennen. Diese Muster können anschließend zur Vorhersage mit zukünftigen Daten verwendet werden. In dieser Arbeit können beispielsweise die ABA-Daten mithilfe des maschinellen Lernens in verschiedene Gruppen oder Muster unterteilt werden, die Informationen über den Zustand der Schiene geben. Dieser Zustand kann beispielsweise als intakt oder defekt definiert sein. Wenn diese Methode auf neue ABA-Daten angewendet wird, können sie in einen dieser Zustände klassifiziert werden. Da jeder gesehene Datenpunkt in eine Klasse sortiert wird, spricht man hier auch von Klassifikation.

Beim maschinellen Lernen werden im Allgemeinen zwei Ansätze unterschieden. Der erste Ansatz ist das überwachte Lernen. Das Ziel dieses Ansatzes ist es, ein Schema zu lernen, um die Eingangsdaten x auf den Ausgang y abzubilden. Die Variable x steht dabei für die Daten selbst und die Variable y beschreibt die Zielgröße oder auch das Label bzw. die Klasse der Daten. Ein gelabelter Datensatz mit N Daten hat in der Regel die Form $D = (x_i, y_i)_{i=1}^N$ und bildet seine Ausgänge y entweder kategorisch oder nominal ab. Kategorische Ausgänge können beispielsweise „intakte“ oder „defekte“ Schienen sein, während nominale Ausgänge jede beliebige Zahl annehmen können. Der zweite Ansatz beim maschinellen Lernen wird als unüberwachtes Lernen bezeichnet. Ein Beispieldatensatz für diesen Ansatz hätte folgende Form $D = (x_i)_{i=1}^N$. Der Unterschied zum überwachten Lernen besteht darin, dass keine Informationen über die Zielgrößen vorhanden sind. Da dieser Ansatz in dieser Arbeit nicht angewendet wird, soll darauf an dieser Stelle nicht weiter eingegangen werden.

Im Rahmen der Untersuchungen dieser Arbeit sollen verschiedene Klassifikationsalgorithmen getestet und evaluiert werden. Daher sollen die einzelnen Schritte wie z.B. das Extrahieren von Merkmalen oder das Optimieren des Modells im Folgenden genauer betrachtet werden. Des Weiteren werden die verwendeten Klassifikatoren in ihrer Funktionsweise beschrieben und verglichen, um eine spätere Auswahl zu begründen.

2.3.1 Metriken

Um die Güte der Klassifikation bewerten zu können, müssen zunächst einige Metriken festgelegt werden. Standardmäßig werden dabei die Sensitivität (recall) oder auch Trefferquote genannt und die Präzision (precision) verwendet. Auch die Begriffe falsch positiv und falsch negativ bzw. richtig positiv und richtig negativ werden für diesen Zusammenhang benötigt. Daher sollen diese im Folgenden kurz besprochen werden.

Auf das Beispiel der intakten und defekten Schienen bezogen, würde eine intakte Schiene als negativ und eine defekte Schiene als positiv bezeichnet werden. Die Adjektive richtig und falsch beschreiben an dieser Stelle die Richtigkeit der Klassifikation. Wird eine defekte Schiene vom Klassifikator als defekt erkannt, so spricht man von richtig positiv. Eine falsch positive Schiene wurde dagegen fälschlicherweise als intakt klassifiziert.

Die Sensitivität (Gleichung 2.3) gibt an, wie zuverlässig ein positives Objekt als positiv klassifiziert wird. Also mit welcher Wahrscheinlichkeit eine Schiene als defekt klassifiziert wird, in Bezug auf die Gesamtheit aller defekten Schienen. Daher sollte dieser Wert ge-

rade bei potenziell gefährlichen Beschädigungen eine besondere Rolle einnehmen, da ein Übersehen dieser Beschädigungen zu schweren Folgeschäden führen kann. Die Präzision (Gleichung 2.4) gibt an, mit welcher Wahrscheinlichkeit eine als defekt klassifizierte Schiene tatsächlich defekt ist. Dabei wird der Wert auf die Gesamtheit aller als defekt klassifizierten Objekte bezogen. Die Präzision sollte also einen möglichst hohen Wert aufweisen, um unnötigen Arbeitsaufwand bei falsch positiven Schienen zu vermeiden.

Um die allgemeine Leistung eines Klassifikators bewerten zu können, bieten sich noch die Metriken Genauigkeit (accuracy) und der f1-Score an. Dabei stellt die accuracy (Gleichung 2.5) dar, wie viele richtige Vorhersagen, also richtig positiv und richtig negativ in Bezug auf den gesamten Datensatz getroffen wurden. Der f1-Score (Gleichung 2.6) beschreibt ebenfalls die Genauigkeit, bezieht sich dabei aber auf die precision und den recall und sagt aus, wie gut positive und negative Beispiele klassifiziert werden konnten. Ein weiterer wichtiger Parameter im Bereich der Metriken ist der gewichtete Mittelwert (weighted average). Dieser wird sowohl für die Sensitivität als auch die Präzision gebildet und wichtet jede Klasse proportional zu ihrer Größe im Datensatz. Dieser Wert ist besonders nützlich, wenn die Klassen im Datensatz ungleichmäßig verteilt sind. In Gleichung 2.7 ist beispielhaft der gewichtete Durchschnitt für die precision dargestellt, die Variable n steht an dieser Stelle für die jeweilige Klasse. Der Ausdruck $n_{\text{precision}}$ stellt also die erzielte precision für die Klasse n dar. Um die precision anschließend wichten zu können, wird zusätzlich noch der Term n_{samples} eingeführt, dieser repräsentiert jeweils die Anzahl der Beispiele innerhalb der Klasse n .

$$\text{recall} = \frac{r_p}{r_p + f_n} \quad (2.3)$$

$$\text{precision} = \frac{r_p}{r_p + f_p} \quad (2.4)$$

$$\text{accuracy} = \frac{r_p + r_n}{r_p + f_p + r_n + f_n} \quad (2.5)$$

$$\text{f1-Score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (2.6)$$

$$\text{weighted avg} = \frac{\sum_{n=0}^N n_{\text{precision}} \cdot n_{\text{samples}}}{\sum_{n=0}^N n_{\text{samples}}} \quad (2.7)$$

2.3.2 Merkmalsextraktion

Wie in [14] und [27] beschrieben, stellt die Merkmalsextraktion einen Prozess zur Reduzierung der Dimension der Rohdaten dar. Um die Notwendigkeit dieses Prozesses zu veranschaulichen sind die Messdaten für ein Schienenstück in Abbildung 2.5 dargestellt. Die Abbildung zeigt die ABA Daten in z-Richtung vom Channel 61 und deckt ein komplettes Schienenstück (10.05 m) ab. An dieser Stelle ist zu beachten, dass die x-Achse hier in Pixeln angegeben ist. Dabei entsprechen 2 Pixel immer einem externen Counter, also 1 mm.

Da im Zuge der Merkmalsextraktion redundante Daten ohne Informationsverlust entfernt werden können, kann so ein schnellerer Lernprozess für den Klassifikator erzielt werden. Ein weiterer wichtiger Vorteil dieser Methode ist, dass so eine bessere Generalisierung in der Klassifikation erzielt werden kann. Dadurch wird das Modell robuster gegenüber neu-

en Daten. Auch für den Anwender kann eine Merkmalsextraktion Vorteile haben, da die Daten dadurch oft besser visualisiert werden können als z.B. in Abbildung 2.5. So lassen sich etwaige Probleme schon bei der Analyse der Daten erkennen, welche dann bei der Wahl des Klassifikationsalgorithmus berücksichtigt werden können.

In der Regel werden die Merkmale vom Anwender gewählt bzw. erstellt, aber auch automatische Merkmalsbildungen sind möglich. Werden die Merkmale händisch erstellt, wird oft auf Standardmerkmale wie z.B. Maximal- oder Minimalwerte, den RMS oder den CREST-Faktor zurückgegriffen. Aber auch sehr spezifische, auf das Problem angepasste Merkmale können verwendet werden, sofern das nötige Domänenwissen dafür vorhanden ist.

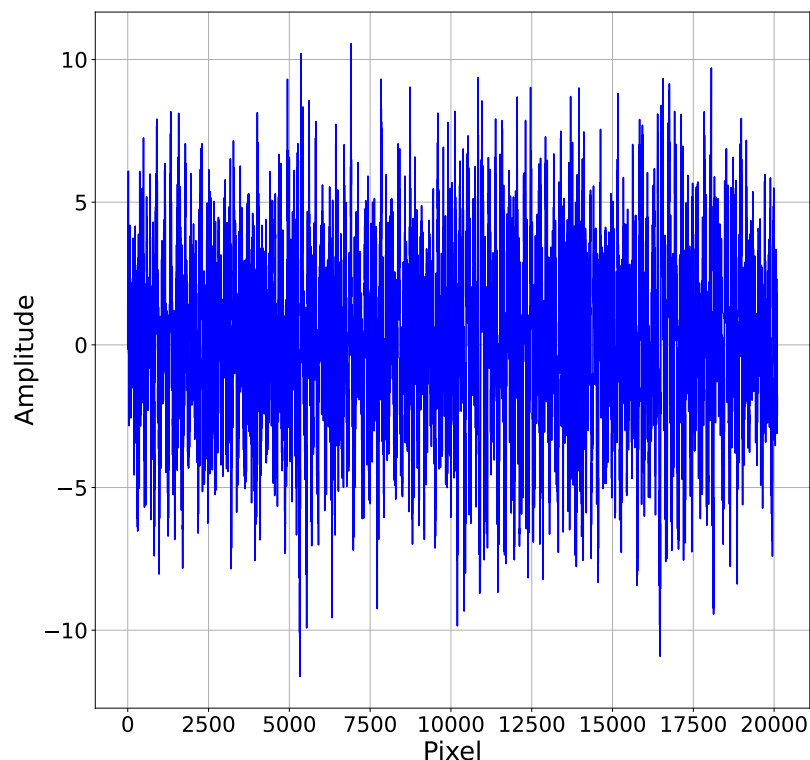


Bild 2.5: Rohdaten eines 10.05 m langen Schienenstücks mit leichter Störung

Die Auswahl der Merkmale sollte für jedes Problem individuell getroffen werden sowie den vorhandenen Datensatz und den Ansatz der Klassifikation berücksichtigen. Da die verwendeten Ansätze in dieser Arbeit alle auf Daten im Zeitbereich basieren, sollen auch nur Merkmale aus dem Zeitbereich betrachtet werden. Für Datensätze, die auf Beschleunigungsmessungen basieren, sind besonders Merkmale geeignet, die sich auf die Amplituden konzentrieren. So können z.B. besonders hohe oder niedrige Amplituden auf starke Beschleunigungen aufgrund einer deformierten Schiene hindeuten. Aus diesem Grund werden die Merkmale Maximal- und Minimalwert aus dem Datensatz extrahiert. Zusätz-

lich dazu werden noch der Mittelwert, der RMS-Wert und der Spitze-Spitze-Wert gebildet und extrahiert. Als letztes Amplitudenmerkmal wird der Crest-Faktor gebildet. Dieser stellt das Verhältnis zwischen den Spitzenwerten zum RMS-Wert dar. Damit lassen sich Daten hervorheben, die deutlich vom Effektivwert abweichen und damit im Idealfall eine beschädigte Schiene abbilden.

Um auch Informationen über die Häufigkeitsverteilung der Daten zu erhalten, werden noch die Merkmale der Kurtosis und der Skewness gebildet. Mithilfe der Skewness lässt sich darstellen, ob die Verteilung der Daten um den Mittelwert symmetrisch oder asymmetrisch ist. So können Beschleunigungsdaten, die an einer defekten Schiene aufgenommen wurden, zu einer asymmetrischen Verteilung führen und so von Daten einer intakten Schiene unterschieden werden. Ähnlich dazu funktioniert die Kurtosis. Diese gibt Informationen über die Steilheit der Verteilung der Daten um den Mittelwert an. Gerade die starken Beschädigungen können die Verteilung der Daten verzerren und so zu einer auffällig hohen oder niedrigen Kurtosis führen.

2.3.3 Entscheidungsbäume

Da in dieser Arbeit für die ersten Untersuchungen auf einen Entscheidungsbaum zurückgegriffen wird, sollen die Grundlagen dieses Klassifikators im Folgenden genauer betrachtet werden. Entscheidungsbäume sind einfach zu implementieren und ihre Ergebnisse leicht zu interpretieren. Damit sind diese Klassifikatoren ideal, um erste Modelle zu entwerfen und verschiedene Klassifikationsansätze zu testen und zu untersuchen. Kann mit einem signifikant komplexerem Modell kein besseres Ergebnis erzielt werden als mit einem rudimentären Entscheidungsbaum, kann dieses Modell schnell als ungeeignet für das Klassifikationsproblem betrachtet und verworfen werden. Ein Entscheidungsbaum arbeitet, wie in Abbildung 2.6 beschrieben, nach dem Prinzip eines Baumdiagramms. Dabei bestimmt der sogenannte Wurzelknoten oder auch Root genannt, den Anfang des Algorithmus. Der Wurzelknoten verzweigt sich über Äste weiter in Knoten (Nodes) und endet schließlich in den sogenannten Blätterknoten. In Abbildung 2.6 ist der Einfachheit halber nur eine Knotenebene dargestellt, für tiefere Baumstrukturen können aber beliebig viele Knotenebenen hinzugefügt werden. Die Blätterknoten stellen in der Regel die zu klassifizierenden Klassen dar, im Beispiel der ABA-Daten wären das intakte oder beschädigte (leicht, mittel und schwer) Schienen. Um einen Datenpunkt aus einem Trainingsdatensatz $D^{\text{train}} = (x_i^{\text{train}}, y_i^{\text{train}}) \big|_{i=1}^N$ in einer Klasse y einsortieren zu können, vergleicht der Entscheidungsbaum jeweils ein Merkmal x mit einem Schwellenwert S . So wird in jedem Knoten m der Merkmalsraum der Trainingsdaten in zwei einfache disjunkte Unterräume Q aufgeteilt.

Der Algorithmus hat nun die Aufgabe mit diesen zwei Entscheidungsmöglichkeiten, dass beste Merkmal und den besten Schwellenwert zu wählen, um die beste Trennung zu erhalten. Da solch ein Problem als NP-vollständig betrachtet wird ([15]) bietet sich an dieser Stelle ein sogenannter greedy Algorithmus zur Lösung an ([27]). An dieser Stelle zeigen sich bereits die ersten Vor- und Nachteile eines Entscheidungsbaum-Klassifikators, da greedy Algorithmen zwar sehr schnell sind, aber in der Regel nur ein lokales Optimum finden. Obwohl diese Art von Klassifikator nicht die optimalen Ergebnisse liefert, sind sie aufgrund ihres Aufbaus sehr robust gegenüber Ausreißern und leicht zu interpretieren. Die Qualität der Trennung am Knoten m wird mittels der „Unreinheit“ $Q_m(T)$ gemessen. Bestimmt wird $Q_m(T)$ in der Regel über das Verhältnis p_{mk} (Gleichung 2.8) der Klassen k aller Punkte N im Knoten m . Das Verhältnis p_{mk} wird anschließend in eine von verschiedenen Fehlerfunktionen $E()$ eingesetzt, um die „Unreinheit“ $Q_m(T)$ zu bestimmen.

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y = k) \quad (2.8)$$

$$\text{falschklassifikations Fehler : } E() = (1 - p_{mk}) \quad (2.9)$$

$$\text{Gini : } E() = \sum_k p_{mk}(1 - p_{mk}) \quad (2.10)$$

$$\text{Kreuzentropie : } E() = - \sum_k p_{mk} \log(p_{mk}) \quad (2.11)$$

Zu den bekanntesten Fehlerfunktionen gehören der Fehler der Falschklassifikationen (2.9), der Gini-Index (2.10) und die Kreuzentropie (2.11). Da der Gini-Index und die Funktion der Kreuzentropie differenzierbar sind und damit für numerische Optimierungen geeignet sind, werden diese bevorzugt verwendet. Ein weiterer Vorteil dieser beiden Funktionen gegenüber dem Fehler der Falschklassifikation ist die höhere Empfindlichkeit an den Knoten ([35]), was wahrscheinlicher zu einer „reinen“ Trennung führt.

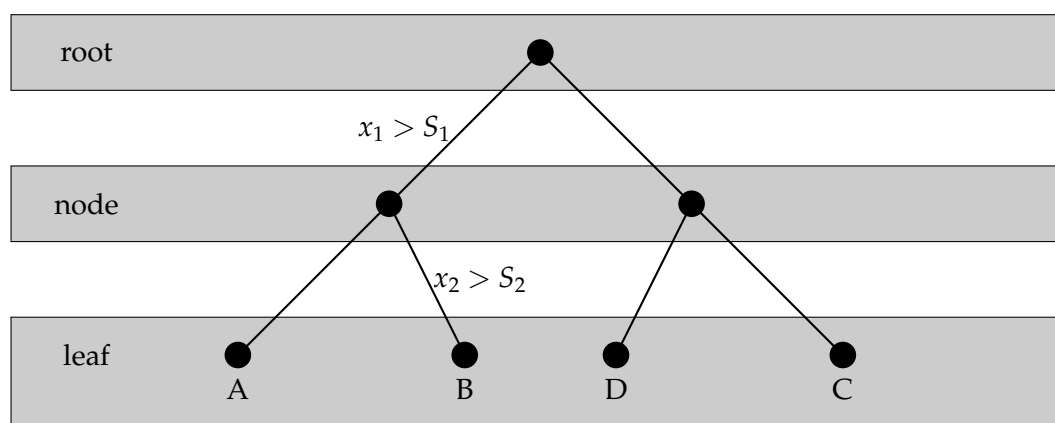


Bild 2.6: Schematische Darstellung eines Entscheidungsbaums

Abbildung 2.7 zeigt beispielhaft das Vorgehen eines Entscheidungsbaum-Klassifikators. Dabei werden auf die extrahierten Merkmale die Schwellwerte S_1 für das Merkmal Minimum und S_2 für das Merkmal Maximum angewendet. Betrachtet man die Trennung, welche sich aus den Schwellwerten ergibt, so lassen sich bereits mittels der ersten Trennung (S_1) einige vereinzelte Daten vom Rest trennen. Wird der übrig gebliebene Datensatz ein weiteres Mal getrennt (S_2), können die übrig gebliebenen potenziellen Störungen gefunden werden.

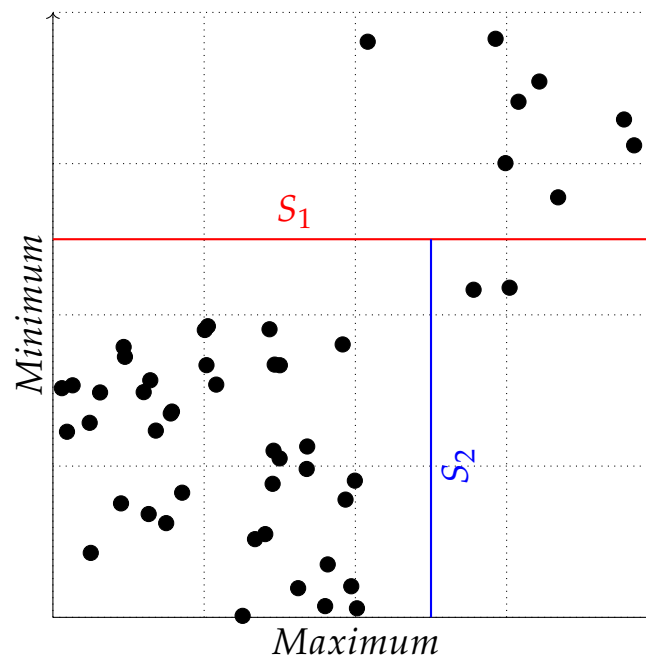


Bild 2.7: Merkmalstrennung durch einen Entscheidungsbaum-Klassifikator

2.3.4 Boosting

Das Boosting stellt eine Methode dar, welche aus einer Kombination verschiedener schwacher Modelle besteht. Diese verschiedenen Modelle werden auch als Ensemble bezeichnet und sollen eine höhere Vorhersagegenauigkeit erzielen als dieselben Modelle einzeln betrachtet. Boosting eignet sich für die Klassifikation und die Regression. Die Grundidee der Klassifikation mittels Boostings basiert auf der Hypothese des schwachen Klassifikators ([8]). Dabei wird davon ausgegangen, dass selbst ein schwacher Klassifikator bessere Vorhersagen treffen kann als das beim zufälligen Raten der Fall wäre. Als Klassifikator wird dabei häufig ein Entscheidungsbaum mit nur einer Verzweigung verwendet, prinzipiell wären aber auch andere Klassifikatoren möglich ([27]). Eine bewährte und häufig verwendete Boosting Methode ist das AdaBoost Algorithmus soll mithilfe der Abbildung 2.8 aus [28] veranschaulicht und erklärt werden (S. Raschka et al., 2021). Als Beispiel dient an dieser Stelle eine binäre Klassifikation mit Kreisen und Dreiecken als Repräsentanten der jeweiligen Klasse. Der Datensatz D hat die Form $D^{\text{train}} = (x_i^{\text{train}}, y_i^{\text{train}})_{i=1}^N$ und der Algorithmus wird mit $t = 1, \dots, T$ Wiederholungen angewendet. Die Gewichte $w_t(i)$ werden mit jeder Anwendung t für jeden Punkt x_i neu bestimmt. Mittels der Gewichte wird anschließend der Fehler ϵ_t bzw. die Güte dieses schwachen Klassifikators $h_t(x)$ bestimmt. Anschließend kann aus dem Fehler des schwachen Lernalters dessen Wichtigkeit α_t für die Gesamtklassifikation $H(x)$ bestimmt werden. Diese mathematischen Zusammenhänge wurden von Freund und Schapire ([7]) entwickelt und sind in den Gleichungen 2.12, 2.13 und 2.14 dargestellt.

Im ersten Schritt der Abbildung 2.8 ist eine erste einfache Trennung zu erkennen, dabei wird deutlich, dass zwei Punkte nicht richtig klassifiziert wurden. Im zweiten Abschnitt werden die Gewichte $w_t(i)$ entsprechend angepasst und vom folgenden schwachen Klassifikator leicht getrennt. Dieses Vorgehen wird in den folgenden Schritten solange wieder-

holt, bis die Summe aller schwachen Lerner $H(x)$ eine gute Klassifikation erreicht.

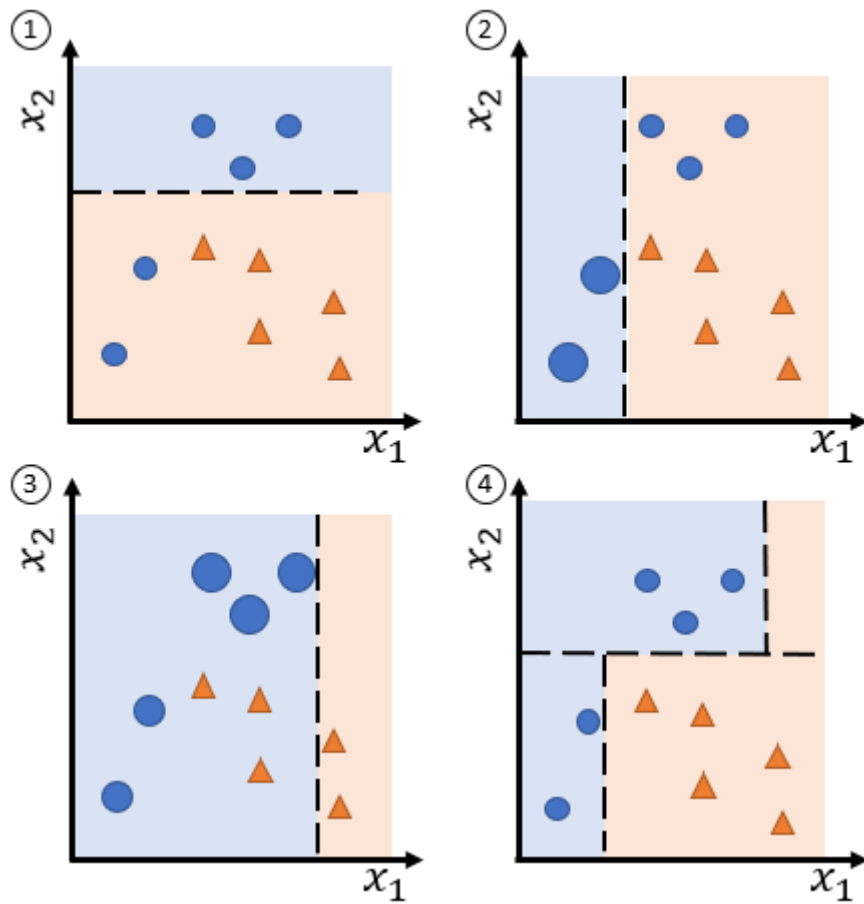


Bild 2.8: Schema des AdaBoostings
Quelle: [28]

$$\epsilon_t = \sum_{i=1}^I w_t(i) [h_t(x) \neq y_i] \quad (2.12)$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right) \quad (2.13)$$

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (2.14)$$

2.3.5 Neuronale Netze

Die ersten Untersuchungen im Bereich künstlicher neuronaler Netze (KNN) wurden im Jahr 1943 von McCulloch und Pitts ([24]) durchgeführt. Die erste große Weiterentwicklung in diesem Bereich konnte dann David E. Rumelhart ([29]) erzielen. In dieser Arbeit wurde der Backpropagationalgorithmus oder auch Fehlerrückführung genannt, als neuer Lernprozess für KNN beschrieben. Im Jahr 2009 konnten Alex Graves und Jürgen Schmidhuber ([12]) mit rekurrenten neuronalen Netzen mehrere Wettbewerbe im Bereich der Handschrifterkennung gewinnen.

Mit diesen Erfolgen und der stetig steigenden Rechenleistung moderner Computer haben neuronale Netze immer mehr an Bedeutung gewonnen und können mittlerweile in fast jedem Forschungsgebiet gefunden werden. So konnten in den letzten 10 Jahren nicht nur große Erfolge in der Bild- und Spracherkennung erzielt werden, sondern z.B. auch in der Systemidentifikation und Regelung ([32]) oder der Sensordatenanalyse ([9]). Aufgrund der erzielten Ergebnisse in den genannten Bereichen sollen neuronale Netze auch einen großen Teil dieser Arbeit ausmachen.

Im Gegensatz zu den bereits besprochenen Klassifikationsalgorithmen lernen KNN aus Beispielen. So werden zum Beispiel Sensordaten über Strukturen verarbeitet, die auf der Grundlage von vorherigen Erfahrungen erstellt wurden. Laut Zaknich ([37]) lassen sich KNN auch als parallel rechnende Architekturen beschreiben, die auf Methoden basieren, welche dem Gehirn ähneln.

Da jedes Problem, welches an ein KNN gestellt wird, andere Anforderungen und Ansätze erfordert, wurden auch die KNN stetig angepasst und weiterentwickelt. So gibt es z.B. den Ansatz der vorwärtsgerichteten KNN zu denen auch die ersten KNN Modelle von McCulloch und Pitts zählen. Aber auch die sogenannten neuronalen Faltungsnetze (CNN), welche besonders in der Bilderkennung gute Ergebnisse erzielen, gehören zu den vorwärtsgerichteten neuronalen Netzen. Diese vorwärtsgerichteten KNN (Feedforward Neural Networks) zeichnen sich dadurch aus, dass alle Verbindungen zwischen den einzelnen Schichten ohne Schleifen und nur in Vorwärtsrichtung verlaufen. Im Kontrast dazu gibt es noch die rekurrenten neuronalen Netze (RNN), wie sie z.B. von Alex Graves und Jürgen Schmidhuber zur Handschrifterkennung verwendet wurden. Diese Art von Netzen ist in ihrer Architektur nicht mehr auf die Vorwärtsrichtung beschränkt, sondern kann zwischen allen Schichten beliebig verbunden sein. Auf diese Weise lässt sich eine Art Gedächtnis erzielen, mit dessen Hilfe auch weit zurückliegende Informationen noch zur Entscheidungsfindung beitragen können. Solche RNN werden laut IBM in den bekannten Sprachsteuerungen Siri und Alexa verwendet ([16]).

Multi layer perceptron Netze

Im Folgenden sollen zunächst einige Grundlagen am Beispiel einfacher Architekturen behandelt werden. Zur Veranschaulichung ist in Abbildung 2.9 ein mehrschichtiges Perceptron dargestellt. Diese Architektur wird auch als "multi layer perceptron"(MLP) bezeichnet und stellt eine der grundlegendsten Formen der vorwärtsgerichteten KNN dar. MLP Netze bestehen, wie auch in der Abbildung dargestellt, immer aus einer Eingangsschicht

und einer Ausgangsschicht. Dazwischen können sich eine oder beliebig viele verdeckte Schichten befinden. Wie dargestellt, wird jedes Neuron einer Schicht in Vorwärtsrichtung mit jedem Neuron der nächsten Schicht verbunden. Diese Verbindungen werden jeweils einzeln mit Gewichten w versehen. Innerhalb eines Neurons wird aus allen Gewichten eine gewichtete Summe gebildet und anschließend an eine Aktivierungsfunktion $f(\cdot)$ weitergegeben. Der Ausgangswert des jeweiligen Neurons hängt dann von der gewichteten Summe der Eingänge und der Aktivierungsfunktion ab. Die beiden Gleichungen 2.15 und 2.16 stellen den mathematischen Zusammenhang zwischen den Schichten dar.

Es wird auch deutlich, dass der Ausgang y des KNN nur von den Gewichten und den verwendeten Aktivierungsfunktionen abhängt.

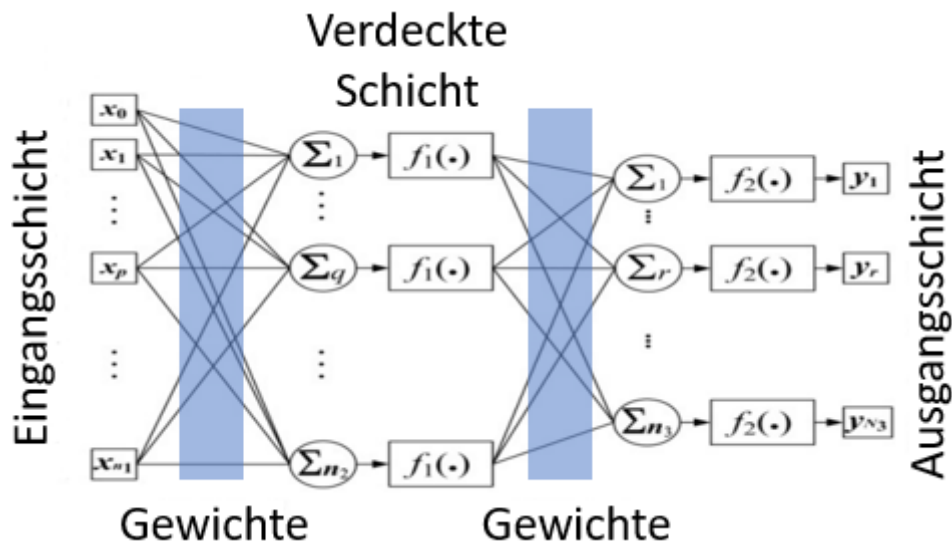


Bild 2.9: schematischer Aufbau eines MLP Netzes
Quelle: [22]

$$h_1 = f\left(\begin{bmatrix} w_0^1 & w_1^1 & \dots & w_n^1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}\right) \quad (2.15)$$

$$y_1 = f\left(\begin{bmatrix} w_0^2 & w_1^2 & \dots & w_n^2 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \end{bmatrix}\right) \quad (2.16)$$

Da die Aktivierungsfunktion maßgeblichen Anteil an der Ausgabe der jeweiligen Neuronen hat, sollte diese individuell für jedes Problem gewählt werden. Für binäre Klassifikationen bietet sich z.B. die Sigmoid-Funktion an (Gleichung 2.17). Wie in Abbildung 2.10 dargestellt, liefert die Sigmoid-Funktion Ausgabewerte in Form von Wahrscheinlichkeiten zwischen 0 und 1. Um zu entscheiden, bei welcher Wahrscheinlichkeit bzw. welchem

Ausgabewert sich das Modell für eine bestimmte Klasse entscheidet, wird ein Schwellwert festgelegt. Dieser kann z.B. auf 0.9 gesetzt werden, um nur Punkte in Klasse 1 zu sortieren, die mit einer sehr hohen Wahrscheinlichkeit auch zu dieser Klasse gehören. Mit diesem Schwellwert kann man großen Einfluss auf die Metriken Präzision und Sensitivität nehmen. So führt ein hoher Schwellwert eher zu guten Ergebnissen in der Präzision, kann aber dazu führen, dass die Sensitivität sinkt.

Handelt es sich dagegen um ein multinominales Klassifikationsproblem, so wird in der Regel auf die Softmax-Funktion (Abbildung 2.12) zurückgegriffen. Diese Funktion gibt eine Verteilung über mehrere Klassen aus, indem sie eine exponentielle Funktion auf die Ausgabe jedes Neurons anwendet und die Summe der Ergebnisse normiert (Gleichung 2.19). Wie bei der Sigmoid-Funktion wird auch hier wieder eine Wahrscheinlichkeit zwischen 0 und 1 ausgegeben. Dabei wird jeder Klasse eine Wahrscheinlichkeit zugewiesen, welche in ihrer Summe 1 ergeben. In der Regel wird dann die Klasse mit der höchsten Wahrscheinlichkeit als Vorhersage verwendet.

Eine weitere häufig verwendete Aktivierungsfunktion ist die ReLU-Funktion (Gleichung 2.18). Diese erzeugt wie in Abbildung 2.11 nur dann eine Ausgabe ($y \geq 0$), wenn der Eingangswert $x \leq 0$ ist. Im Gegensatz zur Sigmoid-Funktion und Softmax-Funktion ist die ReLU-Funktion nicht auf Ausgabewerte zwischen 0 und 1 begrenzt. Dadurch kann sie auch als Ausgabefunktion für Regressionsprobleme verwendet werden. In Klassifikationsproblemen wird die ReLU-Funktion dagegen nicht als Aktivierungsfunktion in den Ausgabeschichten verwendet, sondern in den tiefen Schichten. So führt die Sigmoid-Funktion als Aktivierungsfunktion in den tiefen Schichten oft zu dem Problem der verschwindenden Gradienten, was diese Funktion eher ungeeignet macht ([2]). Das Problem des verschwindenden Gradienten tritt auf, wenn der Gradient in tiefen Schichten eines neuronalen Netzes exponentiell kleiner wird und so dazu führt, dass die Gewichte in den frühen Schichten nicht effektiv trainiert werden können. Ein weiterer Grund für die Wahl der ReLU-Funktion in den tiefen Schichten ist ihre Einfachheit, da diese schnell berechnet werden kann und so deutlich effizienter ist als z.B. die Sigmoid-Funktion.

$$Sigmoid = \frac{1}{(1 + e^{-z})} \quad (2.17)$$

$$ReLU = \max(0, z) \quad (2.18)$$

$$Softmax = \frac{e^{-z_i}}{\sum_i e^{-z_i}} \quad (2.19)$$

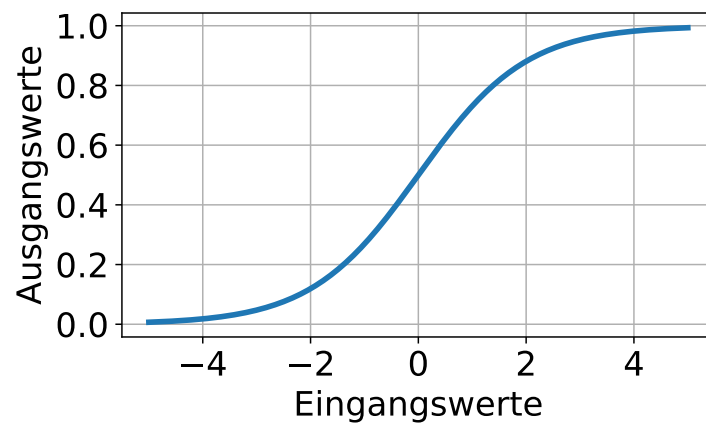


Bild 2.10: Sigmoid-Funktion

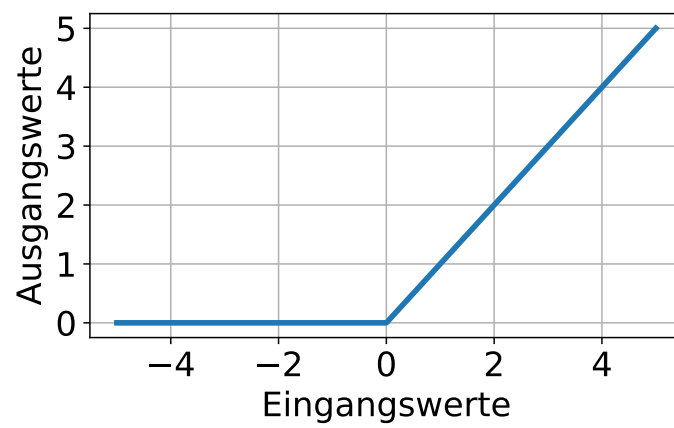


Bild 2.11: ReLU-Funktion

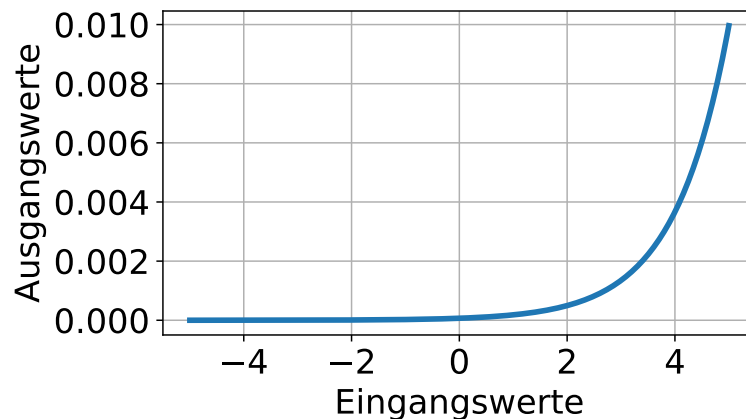


Bild 2.12: Softmax-Funktion

CNN

Da gerade die CNN in immer mehr Anwendungen vielversprechende Ergebnisse liefern, sollen diese auch in dieser Arbeit zum Einsatz kommen. Aus diesem Grund soll im Folgenden etwas genauer auf den Aufbau, die Funktionsweise und die Besonderheiten bzw. Unterschiede zu den MLP Netzen eingegangen werden.

Grundlegend lässt sich feststellen, dass ein CNN eher eine Erweiterung zu einem MLP darstellt als einen komplett neuen Ansatz. So besteht ein CNN wie in Abbildung 2.13 dargestellt aus der bereits bekannten MLP-Architektur und einer vorgeschalteten Faltungs- und Aggregations-Architektur. Die MLP-Architektur ist an dieser Stelle zwar häufig vertreten insgesamt aber kein notwendiger Baustein für ein CNN. Im folgenden soll aber von einem CNN mit MLP-Architektur ausgegangen werden.

Der erste Teil des CNN besteht aus Filter- und Aggregationsschichten (*convolutional* und *pooling*, engl.), welche sich je nach gewünschter Architektur beliebig oft wiederholen können. Die Filterschicht fenstert die Eingangsdaten mittels eines Kernels. Der Kernel selbst enthält festgelegte Gewichte, welche während des Trainings optimiert werden, und wird wie in Abbildung 2.14 dargestellt mit den Eingangswerten gefaltet. Ob und mit welcher Verstärkung die Filterschicht ihre Ergebnisse weitergibt, hängt erneut von einer Aktivierungsfunktion ab. Dabei werden an dieser Stelle in der Regel immer die bereits bekannten ReLU-Funktionen verwendet. Die so entstandenen Ergebnisse werden anschließend an das pooling Layer weitergegeben. Diese Schicht setzt erneut einen Filter mit festgelegter Größe auf die Eingangswerte an und gibt, wie in Abbildung 2.15 zu sehen, nur das jeweilige Maximum an die nächste Schicht weiter. Alternativ kann der Filter anstelle des Maximums auch den Mittelwert aller Werte innerhalb des Filters bilden und diesen weitergeben. Durch dieses Vorgehen erzielt das CNN eine deutliche Parameterreduktion, ohne dabei wichtige Merkmale zu verwerfen.

Im zweiten Teil des CNN befindet sich i.d.R. ein tiefes neuronales Netz nach dem MLP Schema. Da diese Architekturen wie bereits erwähnt nur Eingänge in Form von Vektoren verarbeiten können, wird an dieser Stelle noch eine sogenannte *flatten* Schicht vorgeschaltet. Diese Schicht schreibt alle Einträge der Ausgangsmatrix aus den Filter- und Aggregationsschichten hintereinander und erstellt so einen Vektor. Wird das CNN zur Klassifikati-

on benutzt, so wird in der letzten Schicht im Allgemeinen die bekannte Softmax-Funktion verwendet, bei Regressionsproblemen dagegen die lineare Aktivierungsfunktion.

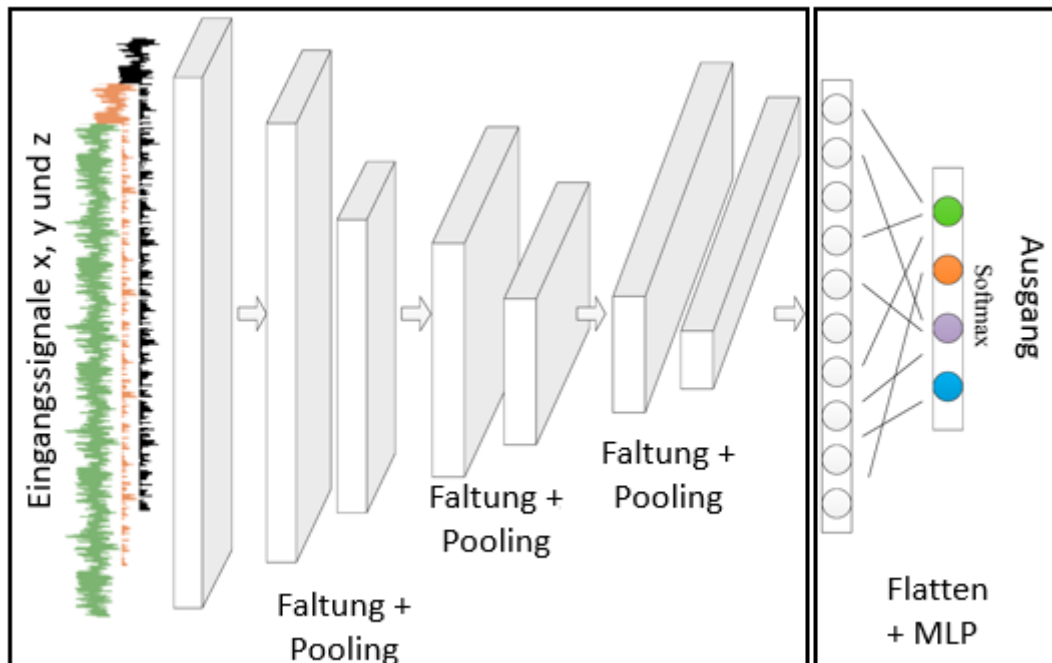


Bild 2.13: Aufbau CNN
Quelle: [3]

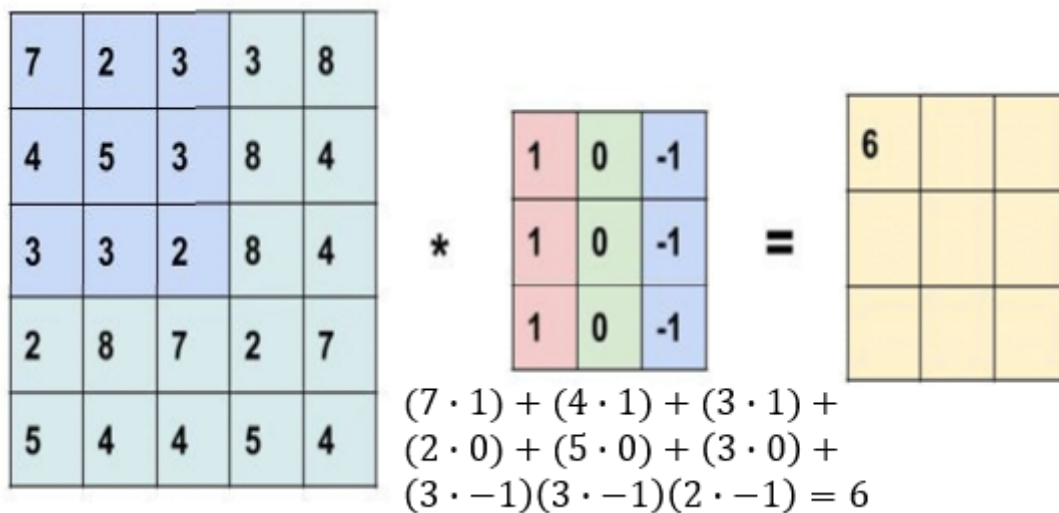


Bild 2.14: Funktionsprinzip eines Kernels
Quelle: [36]

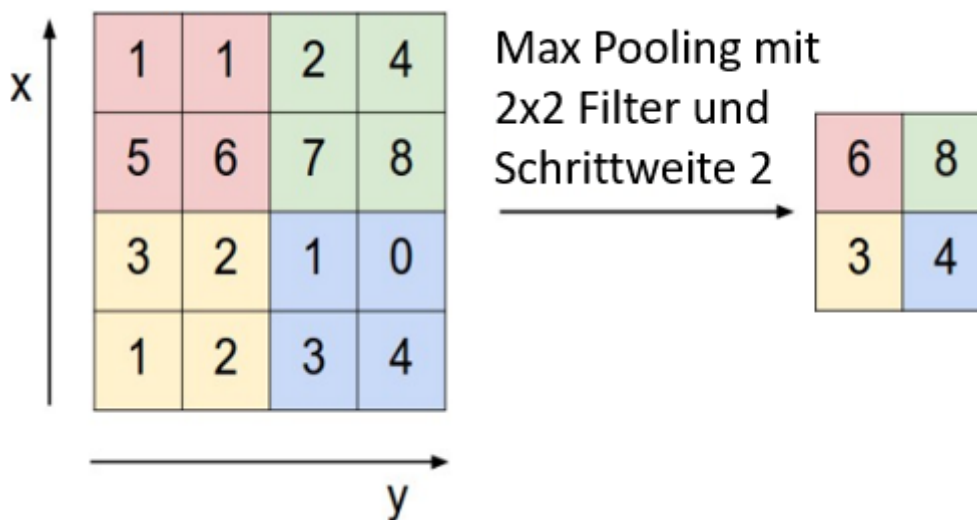


Bild 2.15: Funktionsweise einer Aggregationsschicht
Quelle: [4]

Training neuronaler Netze

Das Training von neuronalen Netzen ist ein wichtiger Bestandteil des maschinellen Lernens. Es umfasst die Anpassung der Gewichte in einem Netzwerk anhand von Trainingsdaten, um eine gewünschte Ausgabe zu erzielen. Aus diesem Grund soll das Training an dieser Stelle nochmal genauer betrachtet werden.

Das Ziel des Trainingsprozesses ist es die Gewichte des neuronalen Netzes so zu verändern, dass die Vorhersagen möglichst oft die Erwartungen erfüllen. Um die Gewichte des neuronalen Netzes anzupassen, wird der Backpropagation-Algorithmus verwendet. Backpropagation berechnet den Gradienten der Kostenfunktion des Netzes und passt die Gewichte entsprechend an, um die Abweichung zwischen den erwarteten und tatsächlichen Ausgaben des Netzes zu minimieren. Das Gradientenabstiegsverfahren (Abbildung 2.16) ist ein wichtiger Bestandteil des Backpropagation-Algorithmus und dient dazu, die Gewichte und in die Richtung zu ändern, die den Gradienten der Kostenfunktion maximiert. Die Abbildung 2.16 zeigt beispielhaft den iterativen Ablauf des Verfahrens. Von einem Startwert aus wird sich mit angepasster Schrittweite ϵ dem Minimum bzw. dem Endwert genähert. Dabei ist aber zu beachten, dass das Finden eines globalen Minimums theoretisch möglich ist, aber in der Regel nur lokale Minima gefunden werden. Eine gängige Variante ist das Stochastic Gradient Descent (SGD), das einen Mini-Batch von Trainingsdaten verwendet, um den Gradienten zu schätzen und die Gewichte entsprechend anzupassen. Eine andere Variante des Gradientenabstiegsverfahrens ist der Adam-Algorithmus, der eine dynamischere Anpassung der Gewichte ermöglicht und damit das Training beschleunigen und verbessern kann.

Die Wahl der Kostenfunktion ist entscheidend für den Erfolg des Trainingsprozesses. Für Klassifikationsprobleme wird häufig die Kreuzentropie-Kostenfunktion verwendet, während für Regressionsprobleme die mittlere quadratische Fehlerfunktion eingesetzt wird. Weist das Klassifikationsproblem ein starkes Ungleichgewicht in der Verteilung der Klas-

sen auf, bietet sich die Focal-loss-Funktion an, da hier die Minderheiten mehr Einfluss auf den Gradienten haben. Diese Funktion wurde 2018 von Tsung-Yi Lin et al. vorgestellt und kann auch in Keras standardmäßig verwendet werden ([21]).

Typischerweise werden die Trainingsdaten in Batches aufgeteilt, um die Effizienz des Trainingsprozesses zu erhöhen. Die Batchgröße oder auch Batchsize bezieht sich auf die Anzahl der Beispiele, die gleichzeitig durch das Netzwerk propagiert werden. Ein weiterer wichtiger Begriff beim Training von neuronalen Netzen ist die Epoche. Eine Epoche bezieht sich auf einen Durchlauf aller Trainingsdaten durch das Netzwerk. Wenn die Batchsize kleiner als die Anzahl der Trainingsdaten ist, bedeutet dies, dass das Netzwerk mehrere Batches benötigt, um eine Epoche abzuschließen. In diesem Fall wird das Netzwerk mehrmals aktualisiert, bevor es die gesamten Trainingsdaten einmal durchlaufen hat.

Ein großes Problem, das häufig beim Training von neuronalen Netzen auftritt, ist die sogenannte Überanpassung oder auch Overfitting genannt. Es tritt auf, wenn das Modell zu gut auf die Trainingsdaten passt und dadurch eine schlechte Generalisierung auf neue Daten aufweist. Man spricht an dieser Stelle auch davon, dass das Netz die Trainingsdaten nur auswendig gelernt hat, ohne dabei allgemeine Muster oder Merkmale zu erkennen. Daraus folgt, dass das Netzwerk zwar in der Lage ist, die Trainingsdaten gut zu klassifizieren oder zu modellieren, aber möglicherweise schlecht auf unbekannte Daten reagiert. Um dem Overfitting entgegenzuwirken, kann z.B. ein Validierungsdatensatz erstellt werden. Mit diesem Datensatz kann am Ende jeder Trainingsepoche getestet werden, wie der Klassifikator auf ungesehene Daten reagiert. Schneidet das Netzwerk auf dem Validierungsdatensatz schlecht ab, kann das ein Hinweis darauf sein, dass es zu Overfitting neigt und weitere Maßnahmen erforderlich sind. Eine weitere Möglichkeit besteht darin, ein Dropout zu verwenden, eine Technik, bei der zufällig ausgewählte Neuronen im Netzwerk während des Trainings deaktiviert werden. Damit lässt sich verhindern, dass das Netzwerk sich zu stark auf bestimmte Merkmale der Trainingsdaten verlässt. Insgesamt ist das Problem des Overfittings ein wichtiges Thema beim Training von neuronalen Netzen und sollte gerade bei kleinen Datensätzen genau beobachtet werden.

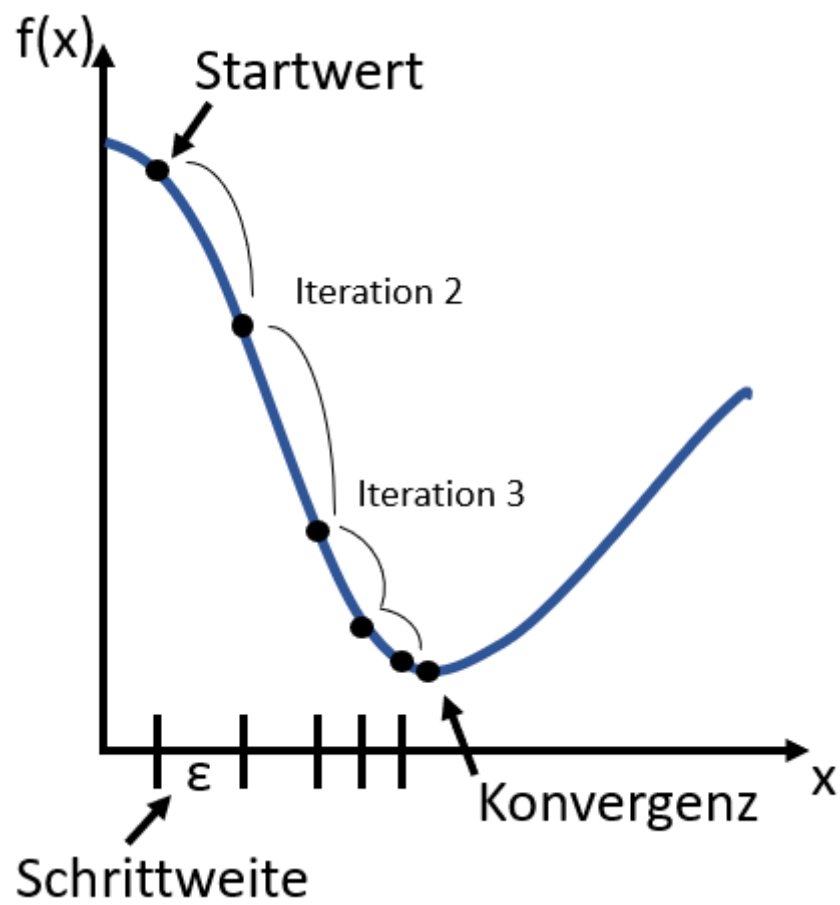


Bild 2.16: Darstellung des Gradientenabstiegsverfahren

3 Methoden - Durchführung

Dieses Kapitel beschreibt die Vorbereitungen und Umsetzungen der durchgeführten Versuche. Da im Vorfeld nicht klar ist, in welcher Form die Daten die besten Ergebnisse liefern, wird hier ein iterativer Ansatz verfolgt. Um bei diesem Ansatz möglichst viele Ergebnisse vergleichen zu können, werden nicht nur verschiedene Klassifikatoren verwendet, sondern auch unterschiedlich aufbereitete Daten bzw. Datensätze.

Die Grundlage stellt bei diesem Ansatz ein Random Forest Klassifikator dar. Dieser wird nacheinander auf alle aufbereiteten Datensätze angewendet. Die Datensätze mit den besten Ergebnissen werden anschließend noch an einen AdaBoost Klassifikator weitergegeben.

Im letzten Schritt werden die Datensätze mit den besten Ergebnissen an ein CNN weitergegeben. Die folgenden Versuche sollen sich darauf konzentrieren, die besten Methoden zur Unterscheidung des Grades der Beschädigung, also „intakt“, „schwer“, „mittel“ und „leicht“, zu ermitteln. Die Unterscheidung zwischen den Arten der Störungen wurde parallel zu dieser Arbeit bereits in [18] untersucht.

3.1 Datenaufbereitung

Da eine Klassifikation der ABA-Daten nur mit den entsprechenden Labeln möglich ist, werden zuerst alle Daten entfernt, welche kein Label aufweisen. Da die Label aus der .csv-Datei immer einem 10.05 m langen Schienenstück zugeordnet sind, werden zunächst diese Schienenstücke aus den Daten ausgeschnitten. Daraus resultiert dann eine Matrix der Form $[1, N, 3]$. Diese Matrix beinhaltet eine Zeitreihe aus N Werten für jeden der 3 Sensor-kanäle (x , y und z).

Im zweiten Schritt werden die ABA-Daten innerhalb dieser Matrix noch auf eine feste Geschwindigkeit von 60 km h^{-1} normiert. Das verhindert, dass schwere Beschädigungen bei langsamen Überfahrten ähnliche Amplituden aufweisen wie leichte Beschädigungen bei schnellen Überfahrten. Da die Länge der Strecke bekannt ist und die Fahrdauer über die konstante Abtastzeit von 25.6 kHz berechnet werden kann, wird für jedes 10.05 m Fenster die Durchschnittsgeschwindigkeit bestimmt. Anschließend werden, wie in Gleichung 3.1 dargestellt, alle Kanäle (x , y und z) der ABA-Daten mithilfe der Durchschnittsgeschwindigkeit angepasst.

$$x_{\text{neu}} = x \cdot \frac{v_{\text{Norm}}}{v} \quad (3.1)$$

Aus den so aufbereiteten Daten können anschließend diverse weitere Datensätze unterschiedlicher Form und Funktion gewonnen werden.

3.1.1 Erstellen der Datensätze

Der erste Datensatz, der erstellt wird, soll als Grundlage dienen und Vor- und Nachteile für die folgenden Ansätze sichtbar machen.

Um den ersten Datensatz zu erhalten, werden die im Vorfeld auf 10.05 m gefensterten Daten nochmals in kleinere Abschnitte von je $N=1000$ Samples unterteilt. Das gilt sowohl für die ABA-Daten aller 3 Kanäle als auch für die Label. Aus den Abschnitten der Label wird anschließend das Label mit der stärksten Wirkung bestimmt und als einziges Label für den gesamten Teilabschnitt verwendet. Da die linke und rechte Schiene (Kanäle 61 und 71) sowohl separate Daten als auch Label besitzen, wird dieses Verfahren auf beide Schienenseiten einzeln angewendet. Zur Veranschaulichung sind die daraus resultierenden Matrizen der Form $[M_{61}, 1000, 3]$ und $[M_{71}, 1000, 3]$ beider Schienenseiten jeweils für den Sensorkanal x in Abbildung 3.1 dargestellt. Verbindet man die beiden Matrizen der Kanäle 61 und 71 über die Dimension der Zeilen, so ergibt sich ein Gesamtdatensatz der Form $[M, 1000, 3]$. Die Zeilenanzahl M ergibt sich hier aus der Summe der Zeilenzahlen M_{61} und M_{71} .

Mithilfe eines zweiten Datensatzes soll untersucht werden, ob die verzögerte Reaktion der Räder auf defekte Schienen kompensiert werden kann. Dazu wird ein Datensatz nach dem selben Schema des ersten Datensatzes erstellt. Zusätzlich wird bei einem Defekt nicht mehr nur das betroffene Schienenstück als defekt markiert sondern auch das folgende. Damit soll erreicht werden, auch verzögerte Reaktionen in den Daten noch als defekte Schienen zu markieren. Der restliche Aufbau des Datensatzes bleibt wie in Abbildung 3.1 erhalten.

Mit dem dritten Datensatz soll untersucht werden, ob die Informationen beider Schienenseiten gleichzeitig bessere Ergebnisse liefern können als einzeln. Um diesen Datensatz zu erstellen, werden durch Fensterung erneut die Matrizen für Kanal 61 und Kanal 71 erstellt. Anschließend wird aus diesen beiden Matrizen eine einzige Matrix der Form $[M, 1000, 6]$ gebildet. Um die Label zu generieren, werden die Label beider Kanäle verglichen. Dabei wird für jedes Fenster immer das stärkste Label aus beiden Kanälen gewählt. Enthält das Fenster aus Kanal 61, z.B. eine leichte Störung und das Fenster aus Kanal 71 eine schwere Störung, so erhalten alle Daten das Label der schweren Störung. An dieser Stelle soll auch erwähnt werden, dass die Daten und Bilder bzw. Label von Kanal 71 eine Strecke von ca. 17 km abdecken. Für Kanal 61 sind dagegen nur Bilder und Label für die ersten 3.5 km und die letzten 450 m vorhanden. Daraus resultiert, dass der deutlich größere Mittelteil der Daten aus Kanal 61 kein Label erhalten hat. Da nicht ausgeschlossen werden kann, dass diese Daten auch Beschädigungen enthalten, können diese nicht zur Klassifikation verwendet werden. So ergibt sich ein Datensatz der ABA-Daten und Label beider Schienenkanäle über eine Strecke von ca. 4 km Länge enthält.

Der vierte Datensatz basiert stark auf dem dritten. An dieser Stelle wird aber die Methode zur Fensterung angepasst. Anstelle des vorherigen Rechteckfensters der Größe $N=1000$ und der Schrittweite $d=1000$ wird jetzt die Schrittweite auf $d=500$ reduziert. Dadurch soll unter anderem erreicht werden, dass dieselben Beschädigungen an unterschiedlichen Positionen des Fensters auftreten und so eine größere Vielfalt in den Datensatz bringen. Außerdem soll durch die Datenaugmentation die Größe des Datensatzes angehoben werden, da dieser aufgrund der kurzen gemeinsamen Strecke von ca. 4 km deutlich kleiner ausfällt als die ersten beiden. An dieser Stelle sollte beachtet werden, dass der Split in Trainings-, Test- und Validierungsdaten hier vor der Fensterung stattfinden muss. Sonst entsteht

die Gefahr, dass die Test- und Validierungsdaten Zeitreihen enthalten die um 500 Samples verschoben bereits im Trainingsdatensatz enthalten sind. Außerdem wird die alternative Fensterung nur auf den Trainingsdatensatz angewendet. Die Datensätze zur Validierung und zum Testen, werden mit einem Versatz von $d=1000$ erstellt.

Kanal 61 Sensor x	x1	x2	...	x999	x1000	Label
	2.3	1.1	...	0.1	0.9	0
	1.7	1.6	...	2	2.4	3

Kanal 71 Sensor x	x1	x2	...	x999	x1000	Label
	0.7	0.4	...	4.1	4	1
	2.1	1.9	...	0.1	0.3	0

Bild 3.1: Schematischer Aufbau des ersten Datensatzes

3.1.2 Merkmalsextraktion

Bevor die Datensätze an den Klassifikator weitergegeben werden können, müssen zunächst Merkmale bestimmt werden. Um für alle Versuche die gleichen Bedingungen zu schaffen, werden aus allen Datensätzen jeweils die gleichen Merkmale extrahiert. Da die Datensätze je nach Ansatz drei oder sechs Sensorkanäle gleichzeitig darstellen, werden die Merkmale einzeln für jeden Kanal bestimmt. Insgesamt werden acht verschiedene Merkmale pro Kanal bestimmt, diese sind:

- Mittelwert
- Maximum
- Minimum
- RMS-Wert
- CREST-Faktor
- Spitze-Spitze-Wert
- Skewness
- Kurtosis

Zur Veranschaulichung ist in Tabelle 3.1 einmal beispielhaft die Merkmalsextraktion für den RMS-Wert aus dem zweiten Datensatz dargestellt. Nach diesem Schema ergeben sich für den ersten Datensatz 24 Merkmale zur Klassifikation und für die übrigen Datensätze jeweils 48 Merkmale.

RMS_x^{71}	RMS_y^{71}	RMS_z^{71}	RMS_x^{61}	RMS_y^{61}	RMS_z^{61}
0.12	0.87	0.94	0.90	0.62	0.48
...

Tabelle 3.1: Beispiel für ein extrahiertes Merkmal (RMS) aus dem zweiten Datensatz

3.2 Untersuchungen mit dem Random Forest Klassifikator

Mithilfe des Random Forest Klassifikators werden insgesamt vier Versuche mit den vorher erstellten Datensätzen durchgeführt. Um für jeden Ansatz den besten Klassifikator zu bilden, wird mit der GridSearchCV() Funktion aus dem sklearn Paket von Python gearbeitet. Damit kann nicht nur eine Kreuzvalidierung durchgeführt werden, sondern es werden auch verschiedene Baumtiefen und Gütekriterien getestet. Außerdem soll an dieser Stelle erwähnt werden, dass in allen Versuchen der Parameter „class_weight“ an den Random Forest Klassifikator übergeben wird. Damit soll von Anfang an dem starken Ungleichgewicht in der Klassenverteilung entgegengewirkt werden. Dazu werden die Gewichte der Klassenverteilungen mithilfe von scikit-learn und Python so bestimmt, dass die Verteilung ausgeglichen wird.

Der erste Versuch soll an dieser Stelle die Grundlage bilden und einen Benchmark für alle weiteren Versuche darstellen. Dazu wird der erste Datensatz bzw. dessen Merkmale verwendet. Da dieser Versuch den einfachsten Ansatz darstellt, sollten die folgenden Versuche mindestens genauso gute Ergebnisse erzielen, um weiter in Betracht gezogen zu werden. Außerdem soll noch der Einfluss der Label der Klasse „sonstige“ auf das Klassifikationsergebnis untersucht werden. Da diese Label aus verschiedenen Gründen vergeben werden, bilden sich in den Daten auch verschiedene Muster, welche die Einteilung erschweren können. So werden z.B. Weichen, Schienenstöße, geschliffene Beschädigungen und weitere Besonderheiten in diesem Label zusammengefasst.

Der zweite Versuch soll zeigen, ob die Verzögerung der Reaktion der Räder ausgeglichen werden kann. Dazu wird ein Random Forest auf die Merkmale des zweiten Datensatzes angewendet. Der Aufbau des Klassifikators entspricht dabei, dem aus dem ersten Versuch. Es werden erneut die Daten mit dem Label „sonstige“ entfernt.

Im dritten Versuch soll untersucht werden, ob die Daten beider Schienenseiten gleichzeitig betrachtet mehr Informationen liefern. Dazu werden die Merkmale aus dem 3. und 4. Datensatz an den Random Forest weitergegeben. Da der 4. Datensatz aufgrund der Zusammensetzung nicht für einen Validierungssplit innerhalb der GridSearchCV() Funktion geeignet ist, werden an dieser Stelle diverse Parameter Konfigurationen mit einer for-Schleife getestet. Innerhalb der for-Schleife werden die Ergebnisse mit einem separaten Validierungsdatensatz validiert.

Bedingt durch das Messverfahren und den natürlichen Abnutzungsprozess im Bahnbetrieb sind beschädigte Schienen im Datensatz deutlich seltener als intakte Schienen. Um den Einfluss dieses Ungleichgewichts der Klassen zu untersuchen, soll ein vierter Versuch unternommen werden. In diesem Versuch werden die Datensätze so angepasst, dass die Label sich nur noch in beschädigt oder intakt unterscheiden. Es werden also alle Beschä-

digungen in einer Klasse zusammengefasst. Dadurch soll erreicht werden, dass sich das große Ungleichgewicht der Klassen verbessert. Zur Veranschaulichung sind in Tabelle 3.2 die Ursprünglichen und die binären Verhältnisse der Klassen dargestellt. Die Darstellung ist dabei so angepasst, dass die Klasse der intakten Schienen immer auf 1 skaliert wird.

Aufgrund der schlechten Leistungen des dritten und vierten Datensatzes sollen diese folgend nicht weiter betrachtet werden. Des Weiteren werden erneut die Beschädigungen der Klasse „sonstige“ entfernt.

Anschließend sollen noch die Möglichkeiten einer binären Trennungen innerhalb der beschädigten Klassen untersucht werden. Insgesamt soll ein Aufbau, wie in Abbildung 3.2 dargestellt implementiert werden. Nach dem die intakten Schienenstücke identifiziert und aussortiert sind, sollen in einem zweiten Random Forest die leichten Beschädigungen aussortiert werden. Dazu werden die schweren und mittleren Beschädigungen erneut in eine gemeinsame Klasse sortiert um die Klassenverteilung auszugleichen. In einem dritten Random Forest sollen abschließend die mittleren von schweren Beschädigungen getrennt werden.

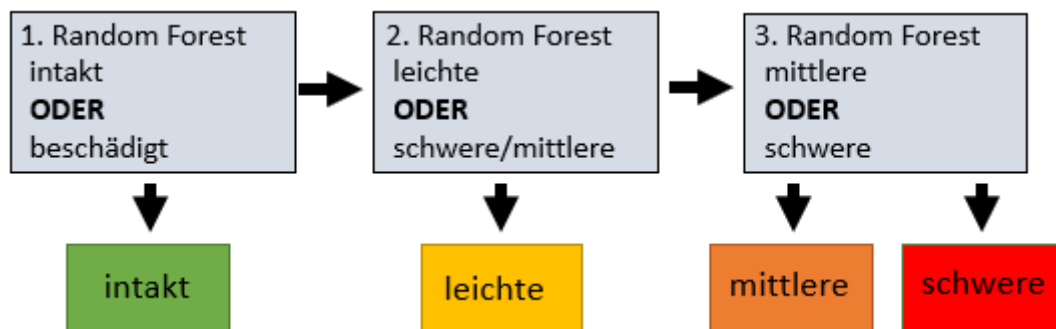


Bild 3.2: Schema zur binären Klassifikation aller Klassen

multinominale Verteilung				binäre Verteilung	
intakt	schwer	mittel	leicht	intakt	beschädigt
1	0.01	0.06	0.26	1	0.32

Tabelle 3.2: unterschiede der Klassenverteilungen

3.3 Untersuchungen mit dem AdaBoost Klassifikator

In diesem Kapitel soll untersucht werden, ob mit einem besseren Klassifikator auch bessere Ergebnisse erzielt werden können. Dazu sollen unter anderem die bereits gewonnenen Erkenntnisse aus den Versuchen mit dem Random Forest Klassifikator verwendet werden. So werden in sämtlichen Versuchen dieses Kapitels nur der Erste und der Zweite Datensatz aus den Random Forest Versuchen verwendet, da mit diesen bisher die besten Ergebnisse erzielt werden können.

Als Klassifikator wird an dieser Stelle der AdaBoost Klassifikator verwendet. Um eine bessere Vergleichbarkeit mit dem Random Forest Klassifikator zu ermöglichen, werden für die

schwachen Lerner des AdaBoost auch Random Forest Klassifikatoren angesetzt. Insgesamt ähnelt der ganze Versuchsaufbau stark dem Aufbau des Random Forest Klassifikators. So wird der AdaBoost Klassifikator ebenfalls aus dem scikit-learn Paket von Python verwendet und auch die GridSearchCV() Funktion kommt erneut zum Einsatz. Als Parameter werden in diesem Fall der Algorithmus „SAMME.R“ und ein Random Forest Klassifikator mit maximaler Tiefe von 1 übergeben. Außerdem wird die Anzahl der maximal möglichen schwachen Lerner auf einen Bereich von 30 bis 100 festgelegt.

Die Versuche des AdaBoost Klassifikators sollen identisch mit den ersten beiden Versuchen des Random Forest sein. Hier soll nur geprüft werden ob der AdaBoost prinzipiell bessere Ergebnisse liefert.

Anschließend wird in einem zweiten Versuch untersucht, wie der AdaBoost den binären Ansatz verarbeitet. Dazu werden erneut die ersten beiden Datensätzen so aufgeteilt, dass zunächst alle Störungen in eine gemeinsame Klasse sortiert werden. Zeigt der AdaBoost an dieser Stelle bessere Leistungen als der Random Forest Klassifikator werden auch die übrigen binären Klassifikationen durchgeführt um erneut alle Klassen sukzessiv voneinander zu trennen. Zeigt sich keine Verbesserung wird dieser Ansatz mit dem AdaBoost nicht weiter verfolgt.

3.4 Untersuchungen mit CNNs

Mithilfe der Erkenntnisse aus den Versuchen des Random Forest und des AdaBoost Klassifikators sollen im Folgenden die Versuche mit den CNNs durchgeführt werden. Da in allen anderen Versuchen die binäre Klassifikation die besten Ergebnisse geliefert hat, soll diese auch in den folgenden Versuchen eingesetzt werden. Außerdem wird auch in diesen Untersuchungen nur noch der erste Datensatz verwendet. Um alle Schienenstücke nach den Labels intakt, leicht, mittel und stark beschädigt trennen zu können, werden insgesamt 2 CNNs benötigt. Dazu wird als erstes ein CNN entworfen, welches intakte von beschädigten Schienen trennt. Ein zweites CNN soll dann anschließend die beschädigten Schienen in die Klassen „leicht“, „mittel“ und „schwer“ trennen.

Das Vorgehen zum Aufbau und Training der jeweiligen CNNs soll im Folgenden beschrieben werden. Zunächst soll noch erwähnt werden, dass der komplette Aufbau und Trainingsprozess mithilfe von Tensorflow bzw. Keras durchgeführt wird. Für das erste CNN soll eine Struktur ähnlich dem Beispiel aus Abbildung 2.13 verwendet werden. An dieser Stelle soll das CNN 3 convolutional-Schichten aufweisen, wobei jeweils 16 Filter und eine Kernelgröße von 8 angesetzt werden. Innerhalb jeder convolutional-Schicht wird noch eine Batchnormalisierung durchgeführt, außerdem wird für jede Schicht die ReLu-Funktion verwendet. Nur die Ausgänge der letzten Faltung werden durch ein Pooling über den globalen Durchschnitt an ein tiefes neuronales Netz weitergegeben. Das tiefe neuronale Netz besteht dabei aus zwei verdeckten Schichten, welche jeweils die ReLu Funktion verwenden und durch 12 respektive 6 Neuronen pro Schicht gebildet werden. Um einer Überanpassung vorzubeugen, kommt zusätzlich noch eine Drop-Schicht zum Einsatz. In der Ausgangsschicht kommt aufgrund der binären Klassifikation die Sigmoid-Funktion zum Einsatz. Aufgrund der schlechten Klassenverteilung wird für das erste CNN die Focal-loss-Funktion als Lossfunktion verwendet. Für den Trainingsprozess werden für beide CNNs folgende Parameter festgelegt. Die Epochenanzahl wird auf 300 begrenzt und es wird jeweils eine Batchgröße von 16 verwendet. Die Learning-Rate wird initial auf 0.0008

angesetzt und verringert sich nach 50 Epochen ohne Verbesserung immer um die Hälfte. Die Minimale Learning-Rate beträgt 0.0001.

Vor dem Training werden die Daten noch standardisiert und mit einem üblichen Split in Trainings- und Testdatebsatz aufgeteilt.

Nimmt man an, dass das erste CNN die Daten optimal trennt, erhält man für das zweite CNN einen Datensatz der eine deutlich ausgewogenere Klassenverteilung aufweist als der erste. Aus diesem Grund soll das zweite CNN eine multinominale Klassifikation durchführen und direkt die übrigen 3 Klassen („leicht“, „mittel“ und „schwer“) trennen. Das zweite CNN wird ähnlich zum ersten aufgebaut, erhält dabei aber ein convolutional Layer weniger, da der kleinere Datensatz eine noch größere Gefahr zur Überanpassung darstellt. Des Weiteren wird an dieser Stelle die Softmax-Funktion verwendet, da es sich hier um eine multinominale Klassifikation handelt. Als Lossfunktion wird die in Keras implementierte Kreuzentropie „sparse categorical crossentropy“ verwendet. Der Rest der Architektur und des Trainings bleiben identisch zum ersten CNN.

4 Auswertung

In diesem Kapitel sollen die vorher beschriebenen Versuche ausgewertet werden. Dazu werden unter anderem die Klassifikationsberichte in Form von Tabellen und Konfusionsmatrizen dargestellt. Um mögliche Überanpassungen zu bewerten, werden in einigen Fällen auch Trainings- und Validierungskurven abgebildet.

4.1 Auswertung Random Forest

Als Erstes sollen die Versuche des Random Forest Klassifikators ausgewertet werden. Damit sollen unter anderem die besten Ansätze herausgefiltert werden und erste Benchmark-Ergebnisse gesetzt werden.

4.1.1 Benchmarkversuch

Die Ergebnisse der ersten Untersuchungen (Benchmarkversuche) sind in den beiden Tabellen 4.1 und 4.2 dargestellt. Die Tabelle 4.1 zeigt die Ergebnisse mit dem 1. Datensatz und macht deutlich, dass dieser Datensatz kaum bessere Ergebnisse liefert als die Untersuchungen von Ahmad ([1]). Daten mit dem Label „schwer“ werden in diesem Versuch überhaupt nicht erkannt. Die Daten der Klasse „leicht“ liefern in diesem Versuch die besten Ergebnisse unter den defekten Klassen. Da diese Daten mit 333 Beispielen deutlich öfter auftreten als die Daten der Klassen „schwer“ und „mittel“, lässt sich annehmen, dass der Datensatz entweder nicht genug Beispiele der anderen Klassen beinhaltet oder die Verteilung der Klassen nicht ausgewogen genug ist. Weitere Hinweise auf das Problem der Klassenverteilung sind die gewichteten Durchschnitte. Diese liegen bei 74 % bzw. 70 % und damit deutlich über den Werten der unterrepräsentierten Klassen.

Die Klassifikationsergebnisse ohne die Daten mit dem Label „sonstige“ sind in Tabelle 4.2 dargestellt. Hier zeigt sich, dass das Entfernen der unspezifischen Störungen in fast allen Bereichen zu einer Verbesserung führt. So wird in diesem Versuch eine Genauigkeit von 78 % erreicht. Besonders in der Klasse der intakten und leicht beschädigten Schienen haben sich die Ergebnisse verbessert. Das lässt sich damit erklären, dass ein Großteil der Störungen des Labels „sonstige“ geschliffene Schweißnähte darstellt. Da diese auf das Profil des Schienenkopfes geschliffen werden, heben diese sich kaum noch von diesem ab und erzeugen so ein Verhalten in den ABA-Daten, welches einer intakten Schiene nahekommt. In den Klassen der mittleren und schweren Beschädigungen können dagegen keine Verbesserungen erzielt werden. Trotzdem soll das Entfernen der unspezifischen Störungen für zukünftige Versuche beibehalten werden.

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.85	0.77	0.81	1479
schwer	0	0	0	12
mittel	0.18	0.44	0.26	86
leicht	0.47	0.49	0.48	333
sonstige	0.46	0.48	0.47	91
gewichteter Durchschnitt	0.74	0.70	0.71	2001
Genauigkeit			0.70	2001

Tabelle 4.1: Ergebnisse Benchmark

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.87	0.83	0.85	1479
schwer	0.0	0.0	0.0	12
mittel	0.20	0.40	0.27	86
leicht	0.51	0.51	0.51	333
gewichteter Durchschnitt	0.77	0.75	0.76	1910
Genauigkeit			0.78	1910

Tabelle 4.2: Ergebnisse Benchmark ohne Label „sonstige“

4.1.2 Versuch mit verschobenen Labels

Tabelle 4.3 zeigt die Ergebnisse aus dem Versuch der verschobenen Label. Dabei fällt zunächst auf, dass die Genauigkeit im Vergleich zu den vorherigen Versuchen deutlich gesunken ist. Gerade die Resultate in den Klassen der intakten und leicht beschädigten Schienen sind in diesem Versuch schlechter als im Benchmarkversuch. Das gilt für die Präzision und die Sensitivität. An dieser Stelle sollte aber auch erwähnt werden, dass jetzt erstmals schwere Beschädigungen erkannt werden. Trotzdem sind die Ergebnisse mit 17 % Präzision und 14 % Sensitivität nach wie vor sehr schlecht. Allgemein lässt sich aber sagen, dass gerade die stärkeren Beschädigungen mehr von der Verschiebung des Labels profitieren. Die leichten Beschädigungen bzw. intakten Schienen sind von diesem Ansatz eher negativ beeinflusst.

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.84	0.66	0.74	1364
schwer	0.17	0.14	0.15	14
mittel	0.15	0.49	0.22	92
leicht	0.40	0.48	0.43	382
gewichteter Durchschnitt	0.71	0.61	0.65	1852
Genauigkeit			0.61	1852

Tabelle 4.3: Ergebnisse Versuch 2 mit verschobenen Labels

4.1.3 Versuche mit dem dritten und vierten Datensatz

An dieser Stelle werden die Versuche mit dem dritten und vierten Datensatz ausgewertet. Die Ergebnisse der beiden Versuche sind in den Tabellen 4.4 und 4.5 dargestellt. Bei der Klassifikation hat sich herausgestellt, dass in den Trainingsdaten nur ein einziges Beispiel der Klasse „schwer“ vertreten war. Aus diesem Grund werden die schweren und mittleren Beschädigungen hier in einer Klasse zusammengefasst. Speziell mit dem dritten Datensatz werden herausragende Ergebnisse in der Klassifikation der intakten Schienen erzielt. Die Ergebnisse in den Klassen der beschädigten Schienen sind dagegen ähnlich zu den vorherigen Versuchen. Ein Grund für diese schlechten Ergebnisse kann in der Anzahl der vorkommenden Vertreter dieser Klasse liegen. Was darauf schließen lässt, dass dieser Ansatz viel Potential hat, aber in dieser Arbeit nicht genug Daten vorliegen, um diesen weiter zu verfolgen.

Auch der vierte Datensatz zeigt in der Klasse der intakten Schienen vielversprechende Ergebnisse, bleibt bei allen anderen Klassen aber deutlich hinter dem Versuch mit dem dritten Datensatz zurück. Da von den Vertretern der beschädigten Klassen kein einziger erkannt wird, soll dieser Ansatz an dieser Stelle verworfen werden.

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.98	0.95	0.96	489
schwer/mittel	0.21	0.45	0.29	11
leicht	0.36	0.40	0.38	10
gewichteter Durchschnitt	0.95	0.93	0.94	510
Genauigkeit			0.61	510

Tabelle 4.4: Ergebnisse Versuch 3 mit Datensatz 3

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.83	0.99	0.91	327
schwer/mittel	0.0	0.0	0.0	14
leicht	0.0	0.0	0.0	92
gewichteter Durchschnitt	0.70	0.83	0.76	392
Genauigkeit			0.83	392

Tabelle 4.5: Ergebnisse Versuch 3 mit Datensatz 4

4.1.4 erste binäre Klassifikation

Wie in Abbildung 3.2 dargestellt bildet die erste binäre Klassifikation zwischen intakten und beschädigten Schienen die Grundlage für diesen Ansatz. Die Ergebnisse dieses Versuchs sind in den Tabellen 4.6 und 4.7 dargestellt. Gerade die Ergebnisse des zweiten Datensatzes zeigen eine deutliche Verbesserung. So wird an dieser Stelle eine Genauigkeit von 75 % anstelle von ursprünglichen 61 % erreicht. Auch die Sensitivität in der Klasse der intakten Schienen erzielt eine Verbesserung von 16 %. Der zweite Datensatz zeigt im Vergleich zum Ersten auch bessere Ergebnisse bei der Klassifikation der beschädigten Beispiele. Daher soll der zweite Datensatz, welcher den Ansatz der verschobenen Label repräsentiert, an dieser Stelle zum besser geeigneten Ansatz gewählt werden.

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.84	0.86	0.85	1479
beschädigt	0.49	0.46	0.47	431
gewichteter Durchschnitt	0.76	0.77	0.77	1910
Genauigkeit			0.77	1910

Tabelle 4.6: Ergebnisse der binären Klassifikation mit dem 1.Datensatz

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.83	0.82	0.83	1364
beschädigt	0.52	0.54	0.53	488
gewichteter Durchschnitt	0.75	0.75	0.75	1853
Genauigkeit			0.75	1853

Tabelle 4.7: Ergebnisse der binären Klassifikation mit dem 2.Datensatz

4.1.5 weiterführende binäre Klassifikationen

Wie bereits erwähnt, werden an dieser Stelle nur noch die Ergebnisse des zweiten Datensatzes gezeigt. In Tabelle 4.8 sind die Resultate der Trennung der leichten Beschädigungen dargestellt. Im Vergleich zur ersten Trennung können hier die Ergebnisse nochmals gesteigert werden. So wird in diesem Versuch eine Genauigkeit von 79 % erreicht. Auch die Daten der unterrepräsentierten Klasse konnten noch einen f1-Wert von 53 % erreichen und das trotz des geringen Anteils im Trainingsdatensatz.

Die letzte Trennung (Tabelle 4.9) zeigt dagegen deutlich schlechtere Resultate in der unterrepräsentierten Klasse. Die Ergebnisse in der Klasse „mittel“ zeigen, wie bereits alle anderen Majoritätsklassen davor, erneut Werte im Bereich um 85 %. Aufgrund der hohen Fehlerquote, die in jeder Klasse auftritt, welche mit den Daten aus der Klasse der schweren Beschädigungen zusammengefasst wird und der Tatsache, dass alle anderen Klassen in der Einzelbetrachtung signifikant besser abschneiden, lässt sich annehmen, dass die schweren Beschädigungen das größte Problem für diese Aufgabe darstellen. Das kann zum einen daran liegen, dass diese extrem selten vorkommen und daher kaum repräsentativ vom Random Forest gelernt werden können. Oder die Anpassung der Gewichte bei der Klassifikation zu schwierig für den Klassifikator ist, da das Verhältnis der Klassenverteilung zu groß ist. Insgesamt lässt sich zu diesem Versuch festhalten, dass die binäre Klassifikation ein interessanter Ansatz zur Regulierung der Klassenverteilung ist und daher weiterverfolgt werden sollte.

	Präzision	Sensitivität	f1-Wert	Anzahl
leicht	0.88	0.85	0.86	383
schwer/mittel	0.50	0.57	0.53	106
gewichteter Durchschnitt	0.80	0.79	0.79	1910
Genauigkeit			0.79	489

Tabelle 4.8: Ergebnisse der zweiten binären Klassifikation mit dem 2.Datensatz

	Präzision	Sensitivität	f1-Wert	Anzahl
mittel	0.87	0.85	0.86	92
schwer	0.12	0.14	0.13	14
gewichteter Durchschnitt	0.77	0.75	0.75	106
Genauigkeit			0.75	106

Tabelle 4.9: Ergebnisse der dritten binären Klassifikation mit dem 2.Datensatz

4.2 Auswertung AdaBoost

In diesem Teil der Auswertung werden die Versuche des AdaBoost Klassifikators ausgewertet. Dabei sollen diese vorrangig mit den Leistungen des Random Forest Klassifikators verglichen werden, um zu untersuchen, ob die vorherigen Erkenntnisse auch für den vermeintlich besseren Klassifikator gelten.

4.2.1 Vergleichsversuch zwischen AdaBoost und Random Forest

Die Ergebnisse für den ersten Versuch sind in den Tabellen 4.10 und 4.11 dargestellt. Beide Datensätze liefern bei der Klassifikation mittels AdaBoost ähnliche Ergebnisse wie bei der Klassifikation durch den Random Forest Klassifikator. Auch beim AdaBoost lässt sich wieder erkennen, dass der Ansatz der verschobenen Label in den Klassen „mittel“ und „schwer“ teilweise deutliche Steigerungen ermöglicht. Insgesamt lässt sich bei diesem Versuch sagen, dass der AdaBoost an einigen Stellen kleinere Verbesserungen erzielen kann. An anderen Stellen werden dagegen leichte Verschlechterungen im Vergleich zum Random Forest erzielt. Es lässt sich also kein klarer Trend feststellen, welcher den AdaBoost als den klar besseren oder schlechteren Klassifikator darstellt.

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.89	0.59	0.71	1479
schwer	0.02	0.08	0.04	12
mittel	0.10	0.55	0.17	86
leicht	0.40	0.47	0.43	333
gewichteter Durchschnitt	0.76	0.57	0.34	1910
Genauigkeit			0.69	1910

Tabelle 4.10: multinominale Klassifikation des AdaBoost mit Datensatz 1

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.85	0.59	0.70	1364
schwer	0.06	0.21	0.09	14
mittel	0.14	0.71	0.23	92
leicht	0.40	0.39	0.40	382
gewichteter Durchschnitt	0.72	0.55	0.61	1852
Genauigkeit			0.68	1852

Tabelle 4.11: multinominale Klassifikation des AdaBoost mit Datensatz 2

4.2.2 Binäre Klassifikation mit dem AdaBoost Klassifikator

Bei der binären Klassifikation mit dem AdaBoost zeigt sich ein ähnliches Bild wie bei der multinominalen Klassifikation. Im Vergleich mit den Ergebnissen aus dem Random Forest erzielt der AdaBoost minimal schlechtere Ergebnisse in der Gesamtbetrachtung. So werden bei der Genauigkeit nur Werte von 68 % bzw. 69 % (Tabelle 4.12 und 4.13) erreicht. Dafür konnte eine Steigerung im Bereich der Präzision für beide Datensätze erzielt werden. Abschließend lässt sich mit diesen Untersuchungen aber sagen, dass die schlechten Leistungen der Klassifikation nicht an den Klassifikatoren selbst liegen. Das bestätigt viel mehr die Vermutung, dass die Verteilung der Daten schlicht zu schlecht ist und schwer zu identifizierende Klassen zu selten auftreten.

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.87	0.71	0.78	1479
beschädigt	0.39	0.63	0.48	431
gewichteter Durchschnitt	0.76	0.69	0.72	1910
Genauigkeit				0.69

Tabelle 4.12: binäre Klassifikation des AdaBoost mit Datensatz 1

	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.84	0.70	0.76	1346
beschädigt	0.42	0.63	0.51	488
gewichteter Durchschnitt	0.73	0.68	0.69	1852
Genauigkeit				0.68

Tabelle 4.13: binäre Klassifikation des AdaBoost mit Datensatz 2

4.3 Auswertung CNN

Im letzten Abschnitt werden die Ergebnisse der CNNs ausgewertet. An dieser Stelle soll auch besonderer Wert auf die Kontrolle des Trainingsprozesses gelegt werden, da die Modelle aufgrund der Konstellation des Datensatzes zu Überanpassungen neigen könnten.

Die Klassifikationsergebnisse des 1.CNN sind in der Tabelle 4.14 dargestellt. Vergleicht man die Ergebnisse mit denen aus Tabelle 4.1 wird deutlich, dass das CNN in der Klasse „beschädigt“ signifikante Verbesserungen erzielen kann. So können die Präzision und die Sensitivität jeweils um 17 % gesteigert werden. Auch die Präzision und Sensitivität im Bereich der intakten Schienen kann mit dem CNN geringfügig verbessert werden. Da der Datensatz für das Training eines CNN sehr klein ist und damit eine Überanpassung des Modells begünstigt, ist in Abbildung 4.2 der Fehler über den Trainingsprozess dargestellt. Dabei wird ersichtlich, dass die Kurven des Trainings- und Validierungsfehlers über den gesamten Prozess beieinanderliegen. Dieses Verhalten spricht gegen eine Überanpassung des Modells. Zusammenfassend lässt sich zu diesem Versuch sagen, dass das CNN signifikant bessere Ergebnisse liefert als die bisher getesteten Klassifikatoren.

Label	Präzision	Sensitivität	f1-Wert	Anzahl
intakt	0.89	0.90	0.90	1479
beschädigt	0.66	0.63	0.64	431
gewichteter Durchschnitt	0.84	0.84	0.84	1910
Genauigkeit			0.84	1910

Tabelle 4.14: Ergebnisse des ersten CNN

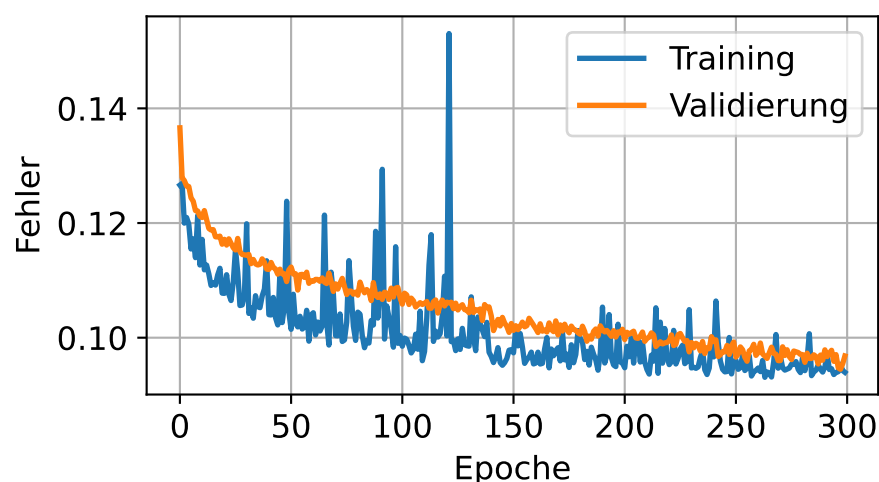


Bild 4.1: Trainingskurve des ersten CNN

In der Tabelle 4.15 sind die Ergebnisse der Klassifikationen des zweiten CNN dargestellt. Da an dieser Stelle eine multinominale Klassifikation durchgeführt wird, soll zusätzlich noch die Konfusionsmatrix (Abbildung 4.3) gezeigt werden. Die Ergebnisse zeigen, dass auch das CNN in Kombination mit dem ersten Datensatz massive Probleme mit den Daten der Klasse „schwer“ hat. Wie in vorherigen Versuchen auch, werden keine Beispiele dieser Klasse erkannt bzw. richtig zugeordnet. Für die Klassen „leicht“ und „mittel“ werden dagegen erneut gute Leistungen erzielt. Abschließend lässt sich zu diesem Versuch sagen, dass auch bei der letzten Trennung allgemein bessere Ergebnisse mit dem CNN erzielt werden können. Die großen Probleme mit der Klasse „schwer“ lassen sich an dieser Stelle ebenfalls auf die geringe Quantität der Beispiele dieser Klasse zurückführen. Betrachtet man die Konfusionsmatrix 4.3, lässt sich erkennen, dass 43 mittlere Beschädigungen als leichte Beschädigungen klassifiziert werden. Gleichzeitig werden auch noch 15 leichte Beschädigungen als mittlere Beschädigungen klassifiziert. Ein Grund dafür kann das Händische erstellen der Label sein. Die Beschädigung kann auf dem Bild stärker oder leichter aussehen, als sie tatsächlich ist und so bei der Klassifikation Probleme bereiten. Ein weiterer Grund wäre, dass die Normierung auf die Durchschnittsgeschwindigkeit nicht ausreicht, um die unterschiedlichen Fahrgeschwindigkeiten zu berücksichtigen. Es können also leichte Beschädigungen bei hohen Geschwindigkeiten wirken, als wären es mittlere Beschädigungen.

	Präzision	Sensitivität	f1-Wert	Anzahl
leicht	0.85	0.96	0.90	334
mittel	0.74	0.50	0.60	86
schwer	0.00	0.00	0.00	12
gewichteter Durchschnitt	0.81	0.84	0.82	432
Genauigkeit			0.84	432

Tabelle 4.15: Ergebnisse des 2. CNN

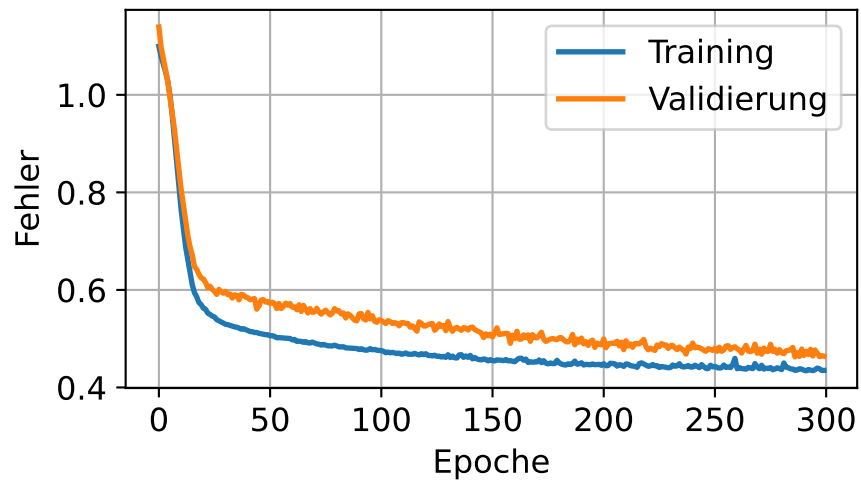


Bild 4.2: Trainingskurve des zweiten CNN

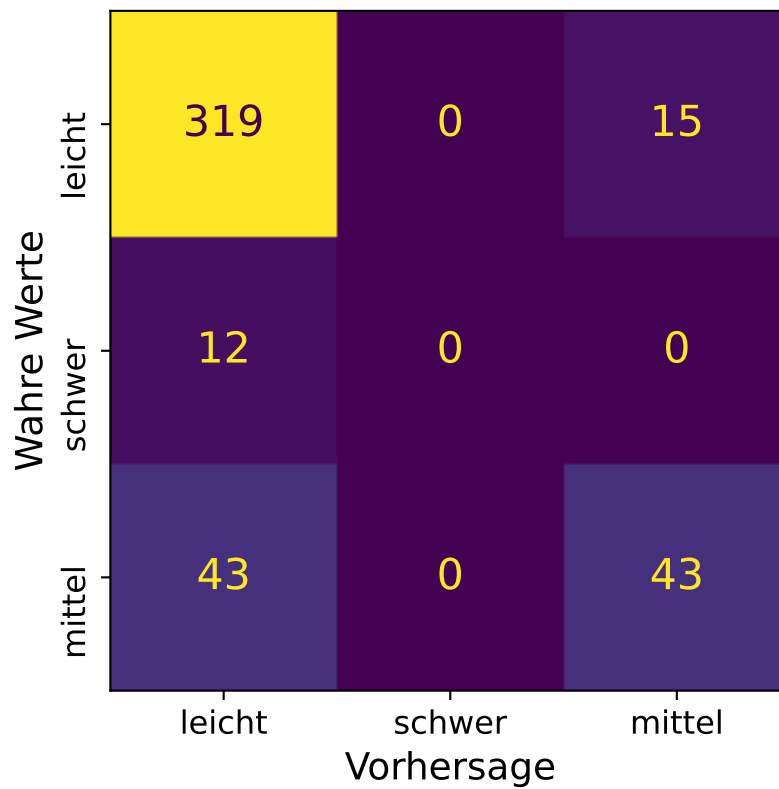


Bild 4.3: Konfusionsmatrix des zweiten CNN

4.3.1 Probleme mit dem Datensatz

In diesem Teil der Auswertung sollen abschließend noch einige Probleme bei der Klassifikation, die in Zusammenhang mit dem Datensatz auftreten, dargestellt werden.

Wie bereits erwähnt ist eins der größten Probleme der zur Klassifikation geeignete Teil des Datensatzes. So kann aufgrund der fehlenden Label für den Kanal 61 nur etwa ein Drittel der Gesamtstrecke für die Erstellung des dritten und vierten Datensatzes verwendet werden. Was besonders den dritten Datensatz schadet, da dieser trotz der geringen Größe interessante Ergebnisse liefert.

Ein weiteres bereits erwähntes Problem ist die schlechte Verteilung der Klassen, welche sich besonders auf das Erkennen der schweren Beschädigungen auswirkt. Diese Beschädigungen treten so selten auf, dass sie in den Trainingsdaten des dritten und vierten Datensatzes überhaupt nicht vertreten waren.

In den Abbildungen 4.4 und 4.5 ist ein weiteres Problem dargestellt. Hier sind jeweils die ABA-Daten in z-Richtung inklusive der zugehörigen Label eines Schienenstücks abgebildet. Besonders auf der Seite des Kanals 61 (Abbildung 4.4) wird deutlich, dass die ABA-Daten eine ausgeprägte Reaktion zeigen. Das gleiche Verhalten zeigt sich etwas schwächer in den Daten von Kanal 71. Eine Störung ist dagegen in keinem der beiden Kanäle erkannt hinterlegt. Dieses Phänomen kann in einem signifikanten Teil aller Schienenabschnitte gefunden werden.

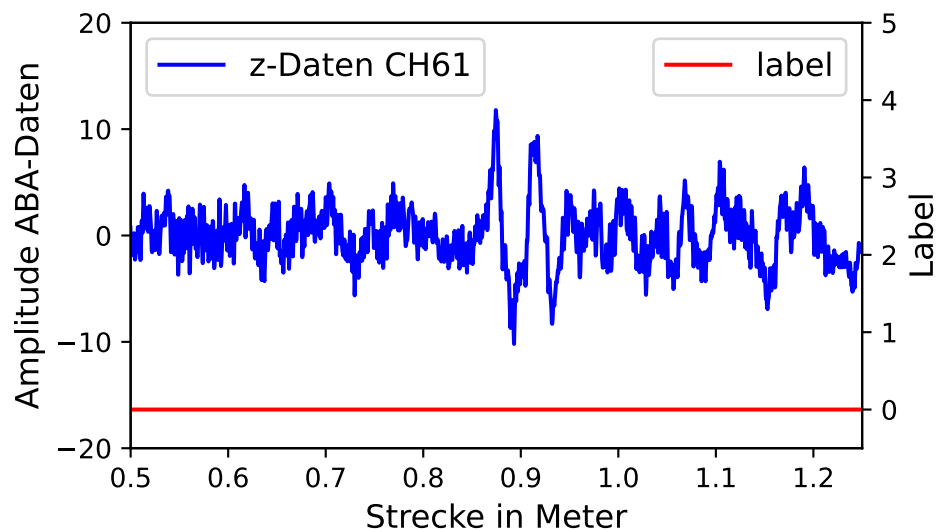


Bild 4.4: Reaktion in den ABA-Daten ohne Label

Mithilfe der Abbildungen 4.6 und 4.7 können weitere häufig auftretende Probleme in den Daten gezeigt werden. In Abbildung 4.6 lässt sich die starke Verzögerung zwischen dem Auftreten der Beschädigung und der eigentlichen Reaktion in den Daten erkennen. Aus diesem Grund wird der zweite Datensatz mit einer Verschiebung erstellt, um dieser Verzögerung entgegenzuwirken. Da dieses Verhalten aber nur sporadisch auftritt, werden bei

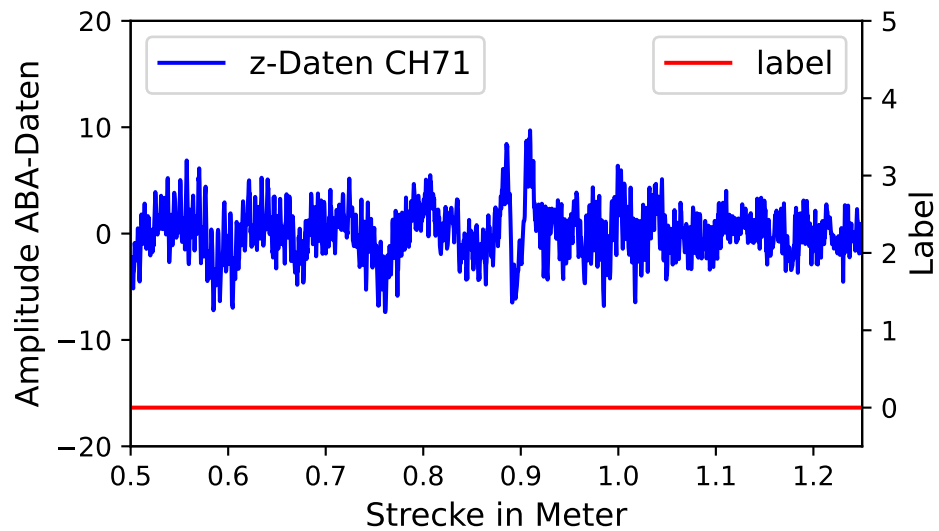


Bild 4.5: Reaktion in den ABA-Daten ohne Label

diesem Ansatz alle Daten, die schnell auf die Beschädigung reagieren, potentiell mit einem falschen Label versehen.

Das zweite Problem, was an dieser Stelle sichtbar wird, ist die Reaktion der Daten beider Kanäle auf Beschädigungen eines Kanals. Wie deutlich zu erkennen, stammt die Beschädigung hier aus der Seite des Kanals 61, eine Reaktion in den ABA-Daten ist hier aber in beiden Kanälen zu erkennen. Dabei tritt die stärkere Reaktion sogar im eigentlich falschen Kanal auf und erhält dabei das Label 0 bzw. intakt. Dieses Problem lässt sich theoretisch mit einem Ansatz umgehen, wie er für den dritten und vierten Datensatz verwendet wird. Aufgrund der fehlenden Label für Kanal 61 kann dieser Ansatz in dieser Arbeit aber nur bedingt verfolgt werden.

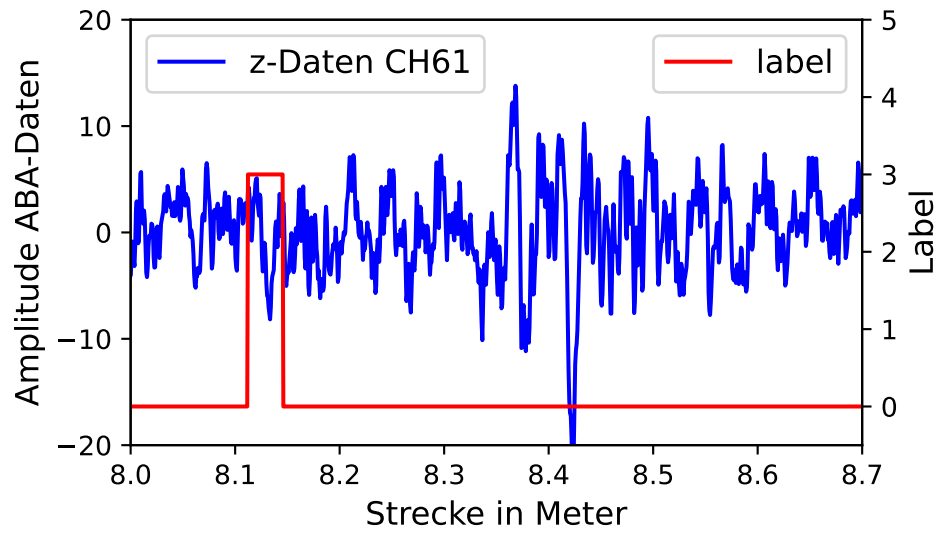


Bild 4.6: verzögerte Reaktion in den ABA-Daten

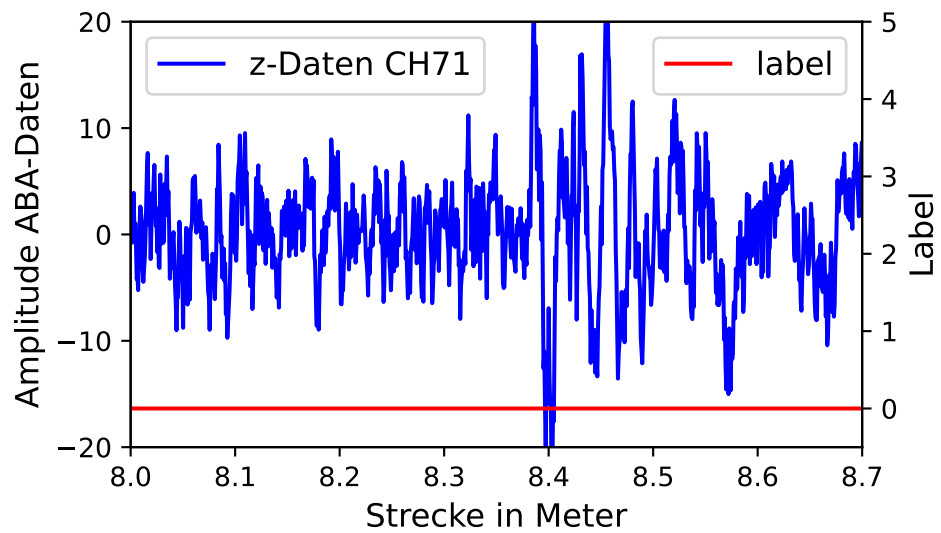


Bild 4.7: verzögerte Reaktion in den ABA-Daten

5 Fazit

Im Verlauf dieser Arbeit wurden verschiedene Versuche durchgeführt, um den Nutzen maschineller Lernverfahren zur Verschleißerkennung im Schienenverkehr zu untersuchen. Dabei wurde mit dem Random Forest Klassifikator zunächst ein rudimentärer bzw. elementarer Klassifikationsansatz verfolgt. Vergleicht man die Ergebnisse aus diesem Ansatz mit den Ergebnissen aus [1] lässt sich feststellen, dass kaum eine signifikante Steigerung erzielt werden konnte. Trotzdem konnten mit diesem Klassifikator diverse Erkenntnisse über die Daten gewonnen werden, welche in verschiedenen Ansätzen in Teilen erfolgreich umgesetzt werden konnten. So wurde z.B. deutlich, dass das enorme Ungleichgewicht innerhalb der Klassenverteilung eins der größten Probleme für die Klassifikation darstellt. Die abschließenden Untersuchungen wurden mit verschiedenen CNNs durchgeführt und konnten die in dieser Arbeit besten Ergebnisse erzielen. Dabei wurde ein zweistufiger Ansatz verfolgt, dieser sollte zunächst alle defekten und intakten Schienen voneinander trennen. Die als defekt klassifizierten Schienenstücke wurden dann in einem zweiten CNN noch auf den Grad ihrer Beschädigung klassifiziert. Durch die Verwendung der CNNs konnte in dieser Arbeit erstmals eine deutliche Steigerung der Ergebnisse im Vergleich zu denen von [1] erzielt werden. Es soll aber auch erwähnt werden, dass der Ansatz der zweistufigen CNNs deutlich mehr Potential aufweist, als hier aufgrund der geringen Menge an Daten ausgeschöpft werden konnte.

Aus den Ergebnissen der Versuche der CNNs lässt sich auch ableiten, dass ABA-Daten insgesamt viel Potential zur Klassifikation von Schienendefekten bilden. Und in Zukunft eine Alternative zu den aktuellen Methoden der Verschleißerkennung darstellen. Damit lässt sich abschließend sagen, dass diese Arbeit beweisen konnte, dass es Möglichkeiten gibt, den steigenden Wartungsaufwand im Bahnbetrieb zukünftig gerecht werden zu können.

6 Ausblick

An dieser Stelle soll noch auf einige Punkte eingegangen werden, welche in dieser Arbeit offengeblieben sind oder in Zukunft noch verbessert werden können.

So wurden die vorhandenen Bilddaten der Schienenstücke in dieser Arbeit überhaupt nicht berücksichtigt. Da nicht alle Bilddaten dieselbe Größe bzw. dasselbe Format hatten und auch die Bildqualität stark geschwankt hat, hätte an dieser Stelle zunächst sehr viel Zeit in die Aufbereitung der Bilder investiert werden müssen. Da das den zeitlichen Rahmen dieser Arbeit überstiegen hätte, wurde sich in einer weiteren Arbeit stärker auf die Klassifikation mittels der Bilddateien konzentriert ([18]). An dieser Stelle konnten besonders im Bereich der Erkennung der Art der Beschädigung gute Ergebnisse erzielt werden. Da einzig der zweite Datensatz gute Ergebnisse bei den Klassen „mittel“ und „schwer“ gezeigt hat, sollte dieser Ansatz zukünftig noch weiter verfolgt werden. So kann die Verschiebung der Label so angepasst werden, dass nur diese Beschädigungen davon beeinflusst werden und leichte Beschädigungen oder intakte Schienen von diesem Ansatz nicht berücksichtigt werden.

Auch der Ansatz, aus dem der dritte Datensatz hervorgeht, bietet viel Potential und wirkt effektiv dem Problem aus Abbildung 4.6 und 4.7 entgegen. Zusätzlich kann dieser Datensatz auch mit dem Ansatz der verschobenen Label für die Klassen der schweren und mittleren Beschädigungen kombiniert werden. Aus diesem Grund sollte dieser Ansatz definitiv in zukünftigen Arbeiten weiterverfolgt werden.

Für den Fall, dass erneut ähnliche Forschungen durchgeführt werden, sollte darauf geachtet werden, dass die Strecke nach Möglichkeit mehrmals abgefahren wird. Anschließend können die so gewonnenen Daten z.B. durch Mittelwertbildung verbunden werden. Das hätte den Effekt, dass zufällige Ausreißer in den ABA-Daten, die nicht auf einen Schienendefekt zurückzuführen sind, gedämpft werden.

Zusammenfassend lässt sich sagen, dass diese Arbeit einen wichtigen Schritt in Richtung moderner Verschleißerkennung im Bahnbetrieb darstellt. So konnte mit verschiedenen Klassifikatoren und Ansätzen nachgewiesen werden, dass Schienendefekte klassifiziert werden können. Damit unterstreicht diese Arbeit die Bedeutung von Methoden des maschinellen Lernens in der Bahnindustrie. Des Weiteren ergänzt diese Arbeit bisherige Forschungen.

Literatur

- [1] Wasim Ahmad. *Artificial Intelligence-based Condition Monitoring of Rail Infrastructure*. University of Twente, Enschede, The Netherlands, 2019.
- [2] Jason Brownlee. *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. 2019. URL: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>.
- [3] X. Chen u. a. *A Self-Adaptive 1D Convolutional Neural Network for Flight-State Identification*. 2019. URL: <https://doi.org/10.3390/s19020275> (besucht am 19.01.2023).
- [4] *Deep Learning for Computer Vision*. 2022. URL: <https://cs231n.github.io/convolutional-networks/#overview> (besucht am 23.01.2023).
- [5] Anika Dey, Hartmut Hintze und Jürgen Lützner. *Maschinelle Schienenbearbeitung nur noch mit integrierter Wirbelstromprüfung*.
- [6] Karl-Otto Edel, Grigori Budnitzki und Thomas Schnitzer. *Schienenfehler 2 Bruchmechanik rissartiger Schienenfehler*. Springer Vieweg, 2021. ISBN: ISBN 978-3-662-59093-5.
- [7] Yoav Freund und Robert E. Schapire. *A Short Introduction to Boosting*. 1999.
- [8] Yoav Freund und Robert E. Schapire. *Boosting - Foundations and Algorithms*. 2012.
- [9] Josef Gessler. *Sensor for food analysis applying impedance spectroscopy and artificial neural networks*. 2021.
- [10] Stuart Grassie und J. Kalousek. *Rail Corrugation: Characteristics, Causes and Treatments*. 1993.
- [11] Stuart Grassie und J. Kalousek. *Rail Corrugation: Characteristics, Causes and Treatments*. 1993.
- [12] Alex Graves und Jürgen Schmidhuber. *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks*. 2009.
- [13] Oke-ndt Group. *UDS2-73 rail flaw detector*. URL: <https://www.ndt.com.ua/en/products/ut/flaw-detectors/uds2-73> (besucht am 10.05.2023).
- [14] Isabelle Guyon u. a. *Feature Extraction Foundations and Applications*. Springer-Verlag, 2006. ISBN: 978-3-540-35487-1.
- [15] Laurent Hyafil und Ronald Rivest. *Constructing Optimal Binary Decision Trees is NP-complete*. 1976.
- [16] IBM. *What are recurrent neural networks?* URL: <https://www.ibm.com/topics/recurrent-neural-networks> (besucht am 18.01.2023).
- [17] Shift2Rail IN2SMART2. *D12.1 Detailed demonstrator definition, technical approach, and progress*. (Besucht am 2020).
- [18] Kanwal Jahan u. a. *Rail Surface Defect Detection and Severity Analysis using CNNs on Camera and Axle Box Acceleration Data*. 2023.

- [19] Shaoguang Li u. a. *New Insights into the Short Pitch Corrugation Enigma Based on 3D-FE Coupled Dynamic Vehicle-Track Modeling of Frictional Rolling Contact*. 2017. URL: <https://www.ndt.net/?id=15943> (besucht am 24. 01. 2023).
- [20] Zili Li, Maria Molodova und Alfredo Núñez. *Improvements in Axle Box Acceleration Measurements for the Detection of Light Squats in Railway Infrastructure*. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 62, NO. 7, 2015.
- [21] Tsung-Yi Lin u. a. *Focal Loss for Dense Object Detection*. 2018.
- [22] D.C. LO, Chih-Chiang Wei und En-Ping Tsai. *Parameter Automatic Calibration Approach for Neural-Network-Based Cyclonic Precipitation Forecast Models*. 2015.
- [23] Francescomaria Marino, Pier Luigi Mazzeo und Ettore Stella. *A Real-Time Visual Inspection System for Railway Maintenance: Automatic Hexagonal-Headed Bolts Detection*. IEEE Transactions on Systems Man, Cybernetics Part C (Applications und Reviews), 2007.
- [24] Warren McCulloch und Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. 1943.
- [25] Maria Molodova. *Detection of early squats by axle box acceleration*. 2012.
- [26] Maria Molodova, Zili Li und Alfredo Núñez. *Automatic Detection of Squats in Railway Infrastructure*. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 15, NO. 5, 2014.
- [27] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press Cambridge, 2012. ISBN: 978-0-262-01802-9.
- [28] Sebastian Raschka und Vahid Mirjalili. *Machine Learning mit Python und Keras, TensorFlow 2 und Scikit-learn: Das umfassende Praxis-Handbuch für Data Science, Deep Learning und Predictive Analytics*. Frechen: mitp-Verlag, 2021.
- [29] D. Rumelhart, Geoffrey E. Hinton und Ronald J. Williams. *Learning representations by back-propagating errors*. 1986.
- [30] Allianz pro Schiene. *Das Schienennetz in Deutschland*. 2021. URL: <https://www.allianz-pro-schiene.de/themen/infrastruktur/schienennetz/>.
- [31] Dr. Wolfgang Schöch. *Rollkontaktermüdung – Ein Überblick über ein weltweites Phänomen*. 2014. URL: <https://www.ndt.net/?id=15943> (besucht am 24. 01. 2023).
- [32] Nandini Sengupta, Md Sahidullah und Goutam Saha. *Lung sound classification using cepstral-based statistical features*. 2016.
- [33] X. Sheng, C.J.C Jones und D.J. Thompson. *A theoretical model for ground vibration from trains generated by vertical track irregularities*. Journal of Sound und Vibration, 2001.
- [34] Trackopedia und voestalpine Schienen. *Verschleiß und Schienenfehler*. 2020. URL: <https://www.trackopedia.com/lexikon/infrastruktur/oberbau/schienen/verschleiss-und-schienenfehler> (besucht am 26. 01. 2023).
- [35] Jerome Friedman Trevor Hastie Robert Tibshirani. *The Elements of Statistical Learning - Data Mining, Inference and Prediction*. 2. Aufl. Springer, 2017.
- [36] Rijul Vohra. *Convolutional Neural Networks*. 2019. URL: <https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e> (besucht am 23. 01. 2023).
- [37] Anthony Zaknich. *Neural Networks for Intelligent Signal Processing*. 2003.