

Digitalisierung der Herstellungskette von C/C-SiC durch den Einsatz künstlicher Intelligenz

Dissertation

zur Erlangung des akademischen Grades

Dr.-Ing.

eingereicht an der
Mathematisch-Naturwissenschaftlich-Technischen Fakultät
der Universität Augsburg

von

Tobias Marc Lehnert

Stuttgart, März 2023



Erstgutachter: Prof. Dr. Dietmar Koch

Zweitgutachter: Prof. Dr. Michael Kupke

Tag der mündlichen Prüfung: 23.10.2023

*Wer seine Ziele nicht an den Sternen festmacht,
kommt nicht mal auf den Kirchturm.*

Patrick Swayze

Inhaltsverzeichnis

Danksagung	III
Nomenklatur	IV
Kurzfassung	V
Abstract	VI
1 Einleitung	1
1.1 Ceramic Matrix Composites	1
1.2 Künstliche Intelligenz in der Materialwissenschaft	4
1.3 Zielsetzung der Arbeit.....	5
2 Grundlagen	10
2.1 Physikalische und datenbasierte Modelle	10
2.2 Begriffsdefinitionen KI und ML.....	11
2.3 KI-Algorithmen.....	13
2.4 Bestimmung der Modell-Genauigkeit.....	23
2.5 Preprocessing	25
2.6 Machine-Learning Grundlagen	40
2.7 Mikrostruktur-Simulation	48
3 Entwicklung eines Web-Interfaces	52
3.1 Notwendigkeit und Ziele des Programms	52
3.2 Überblick und Funktionen	52
3.3 Bilderkennung.....	61
4 Entwicklung des GUI „DataTracker“	75
4.1 Notwendigkeit und Ziele des Programms	75
4.2 Interaktiver Laufzettel.....	76
4.3 Erstellen von PDF-Zusammenfassungen	77
4.4 Interaktive Diagramme	78
4.5 KI-Auswertung	80

5	Physikalische Modellbildung und begleitende Tests	86
5.1	Mikrostruktursimulation	86
5.2	Praktische Versuche	94
6	Ergebnisse und Diskussion	98
6.1	Methodische Erkenntnisse	98
6.2	Materialwissenschaftliche Erkenntnisse	110
7	Zusammenfassung und Ausblick	123
7.1	Zusammenfassung.....	123
7.2	Ausblick	126
8	Literatur	128
Anhang		137
A)	Beispielhafte Anwendung des Gradientenabstiegsverfahrens auf ein einfaches Perzeptron	137
B)	Durchführung eines Vorwärts- und Rückwärtsdurchgangs durch ein mehrschichtiges Neuronales Netzwerk.....	141

Danksagung

Im Vorfeld möchte ich einigen Personen danken, die mich im Verlauf dieser Arbeit unterstützt haben.

Ganz besonderer Dank gilt natürlich meinem Doktorvater Prof. Dr. Dietmar Koch, der trotz seines vollen Terminkalenders immer die Zeit und Muße aufgebracht hat, in zahllosen Meetings meinen Ausführungen zuzuhören, um hier und da wichtige Weichen zu stellen. Diese ausführliche Betreuung hat mich nicht nur thematisch unterstützt, sondern mir auch das Gefühl gegeben, an etwas Bedeutendem zu arbeiten. Mein Dank gilt außerdem Prof. Dr. Michael Kupke für das Anfertigen des Zweitgutachtens.

Weiterhin möchte ich einigen Kolleginnen und Kollegen für die ein oder andere Stunde Zusatzarbeit danken, die meine Promotion ihnen beschert hat. Das betrifft zum einen Fiona Kessel und Lion Friedrich, für die Anfertigung von REM-Aufnahmen, und zum anderen Felix Vogel, Matthias Scheiffele, Marco Smolej und Stefan Frick, für die Herstellung, Bearbeitung und Präparation von Proben. An dieser Stelle auch ein herzliches Dankeschön für all die guten Ideen, die in den zahlreichen Gesprächen entstanden sind.

Nicht zuletzt möchte ich mich auch bei meiner Freundin für ihre Unterstützung, sowie die Geduld und das Verständnis für meine manchmal knapp bemessene Zeit bedanken. Ebenso bedanke ich mich bei meiner Familie für das beständige Interesse an meiner Arbeit und dafür, dass ihr mir wie immer den Rücken freigehalten habt.

Nomenklatur

Abkürzung	Bedeutung
AI	Artificial Intelligence
ANN	Artificial Neural Network
C	Kohlenstoff
C/C-SiC	Kohlenstofffaserverstärktes Kohlenstoff-Siliziumkarbid
CCR	Carbon Conversion Ratio
CFK	Kohlenstofffaserverstärkter Kunststoff
DT	Decision Tree
EFS	Einzelfasersilizierung
FEM	Finite Elemente Methode
KI	Künstliche Intelligenz
KNN	Künstliches Neuronales Netzwerk
LR	Lasso Regression
LSI	Liquid Silicon Infiltration
MAR	Missing At Random
MCAR	Missing Completely At Random
ML	Machine Learning
MNAR	Missing Not At Random
MSE	Mean Squared Error
NN	Neuronales Netzwerk
REM	Raster Elektronen Mikroskop
RF	Random Forest
RMSE	Root Mean Squared Error
RTM	Resin Transfer Molding
RVE	Repräsentatives Volumenelement
Si	Silizium
SiC	Siliziumkarbid
SQL	Structured Query Language

Kurzfassung

Die vorliegende Arbeit untersucht die Auswirkungen von Herstellungsparametern auf die Mikrostruktur des Werkstoffs C/C-SiC. Dabei wurde erstmalig eine Auswertung basierend auf künstlicher Intelligenz (KI) anhand eines vergleichsweise großen Datensatzes von 163 C/C-SiC Proben getätigt. Insgesamt konnten auf diese Weise sieben Herstellungsparameter identifiziert werden, die signifikant mit bestimmten Mikrostrukturmerkmalen korrelieren.

Um die Untersuchung durch künstliche Intelligenz zu ermöglichen, wurde ein digitaler Zwilling der Prozesskette erstellt, welcher die Möglichkeit bietet, Daten automatisch zu erfassen, zu verarbeiten und auszuwerten. Alle notwendigen Programme und Funktionen entstanden dabei durch eigenständige Entwicklung unter Anwendung verschiedener Programmiersprachen und sind auch auf andere Datensätze anwendbar. In diesem Zuge wurden verschiedene KI-Algorithmen und Preprocessing-Methoden miteinander verglichen und ihre Auswirkung auf die Ergebnisgenauigkeit diskutiert. Unter den Algorithmen erwies sich RandomForest als am geeignetsten, wobei ein maximales Bestimmtheitsmaß von $R^2 = 0,67$ erreicht wurde. Als wichtigste Herstellungsparameter für die Prognose erwiesen sich dabei die Dichte, Massenänderung und Porosität im silizierten Probenzustand. Unter Ausklammerung der Silizierung wurde ein Bestimmtheitsmaß von $R^2 = 0,51$ erreicht, wobei hier die Porositäten nach den Prozessschritten Polymerisation, Temperung und Pyrolyse, sowie das verwendete Fasermaterial als wichtigste Parameter identifiziert wurden. Die gefundenen Korrelationen der Machine-Learning Modelle konnten zudem durch die Ergebnisse aus Mikrostruktursimulationen unterstützt werden, welche auf der FEM (Finite Elemente Methode) basierten. Während bisherige Modelle dabei nur sehr kleine Mikrostrukturbereiche behandelten, konnten in dieser Arbeit erstmalig relevante Größenordnungen für die Rissbildung während der Pyrolyse unter Einbeziehung mehrerer Lagen eines Laminataufbaus untersucht werden. So konnten die bereits durch die KI-Modelle gefundenen Korrelationen zwischen Porosität, verwendetem Fasermaterial und Mikrostruktureigenschaften durch die Simulationen erneut bestätigt werden. Die Quantifizierung der simulierten Mikrostrukturen erfolgte dabei durch Bildauswertung über eigens entwickelte Python-Skripte.

Zusammenfassend lässt sich feststellen, dass durch die entwickelten Methoden eine Untersuchung der Auswirkungen von Herstellungsparametern auf die Mikrostruktur von C/C-SiC Proben im großen Maßstab ermöglicht wurde, wodurch zukünftig eine Optimierung der Herstellungsprozesse erfolgen kann.

Abstract

This work examines the effects of manufacturing parameters on the microstructure of C/C-SiC. For the first time, an evaluation based on artificial intelligence (AI) was conducted using a relatively large dataset of 163 C/C-SiC samples. In total, seven manufacturing parameters were identified that are significantly correlated with certain microstructural features.

To enable the investigation through artificial intelligence, a digital twin of the process chain was created, which allowed for data to be automatically collected, processed, and evaluated. All necessary programs and functions were created through independent development using various programming languages and are also applicable to other data sets. Several AI algorithms and preprocessing methods were compared, and their impact on the accuracy of the results was discussed. Among the algorithms, RandomForest proved to be the most suitable, achieving a maximum determination coefficient of $R^2 = 0.67$. The most important manufacturing parameters for the prediction were found to be density, mass change, and porosity in the siliconized sample state. Excluding siliconization, a determination coefficient of $R^2 = 0.51$ was achieved, with porosities after the process steps of polymerization, tempering, and pyrolysis, as well as the fiber material used, identified as the most important parameters. The correlations found by the machine learning models were also supported by the results of microstructure simulations based on the finite element method (FEM). While previous models only dealt with very small microstructure areas, this work was able to investigate relevant orders of magnitude for crack formation during pyrolysis, considering multiple layers of a laminate structure. Thus, the correlations between porosity, fiber material used, and microstructure properties, found by the AI models, were confirmed again by the simulations. The quantification of the simulated microstructures was done through image analysis using custom-developed Python scripts.

In summary, the developed methods enable a large-scale investigation of the effects of manufacturing parameters on the microstructure of C/C-SiC samples, allowing for future optimization of the manufacturing processes.

1 Einleitung

Dieses Kapitel bietet eine Einführung in das Thema aus der Vogelperspektive, sodass der Leser einen groben Überblick über den Umfang der Arbeiten sowie den aktuellen Stand der Technik erhält.

1.1 Ceramic Matrix Composites

CMCs (von *engl.*: Ceramic Matrix Composites) sind eine relativ junge und technologisch hochkomplexe Werkstoffklasse, welche sich besonders gut für den Einsatz in thermisch hochbeanspruchten Anwendungsbereichen eignet. Einsatztemperaturen oberhalb 1100 °C stellen für heutige metallische Superlegierungen (Nickel-Basis- oder Cobalt-Legierungen) ohne aktive Kühlung eine kritische Grenze dar. Insbesondere im Bereich der Flugzeugtriebwerke, bei rückkehrfähigen Raumfahrzeugen und bei Hochleistungs-Bremsen für Automobile treten jedoch oft Spitzentemperaturen über 1500 °C auf [1]. Diese Anforderungen werden sich in zukünftigen Gasturbinen voraussichtlich sogar noch bis auf Temperaturen von etwa 1800 °C steigern, da sich dadurch Wirkungsgrad und Leistungsdichte erhöhen lassen [2]. Aufgrund dieser Tatsachen ist zukünftig ein vermehrter Einsatz von keramischen Werkstoffen wahrscheinlich. Eine Übersicht zu den Einsatzgebieten verschiedener Materialien je nach Temperatur ist in Abbildung 1 gegeben.

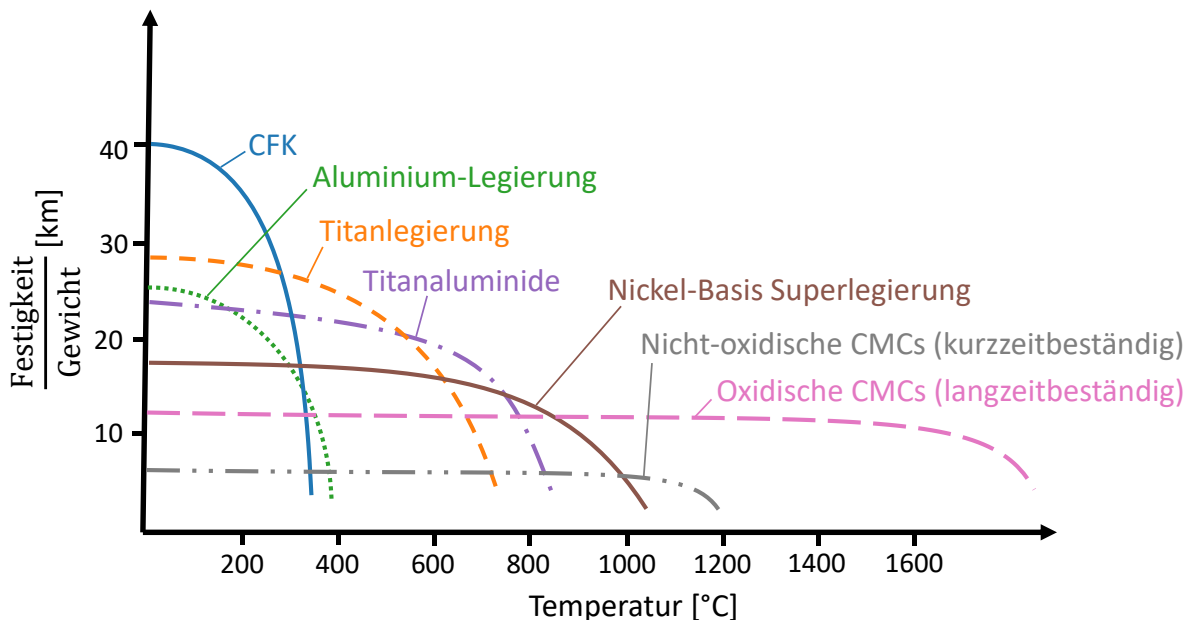


Abbildung 1: Einsatzgebiete unterschiedlicher Materialien für verschiedene Betriebstemperaturbereiche nach [3].

Im Vergleich zu Metallen zeichnen sich CMCs durch eine höhere Temperaturstabilität aus, wobei sie nur etwa ein Drittel der Dichte besitzen. Gleichzeitig zeigen CMCs ein

schadenstolerantes, nicht-sprödes Bruchverhalten, was sie von ihren monolithischen Pendanten abhebt. Aufgrund dieser Vorteile haben SiC/SiC (Siliziumkarbidfaserverstärktes Siliziumkarbid) und Al_2O_3/Al_2O_3 (Aluminiumoxidfaserverstärktes Aluminiumoxid) bereits Einzug in die Produkte kommerzieller Turbinenhersteller wie General Electric Aviation, Pratt & Whitney und Safran erhalten [4, 5]. Das verringerte Gewicht, sowie das Potenzial des gesteigerten Wirkungsgrads aufgrund höherer Einsatztemperaturen, führen zu einem hohen Kraftstoffeinsparungspotenzial. So konnte beispielsweise in Corman, Luthra et al. [6] gezeigt werden, dass durch den Einsatz von CMC-Brennkammerauskleidungen und Ummantelungen der ersten Stufe einer MS7001FA Flugzeugturbine jährlich Kosten von 830 k€ pro Maschine eingespart werden können – und das bei gleichzeitiger Senkung der NO_x, CO und UHC Emissionen. Letzteres macht diese Werkstoffe nicht nur ökonomisch, sondern auch ökologisch interessant. Im Rahmen des Klimaschutzgesetzes der Bundesregierung sollen die Treibhausgas-Emissionen bis 2030 gegenüber 1990 um 65 % gesenkt und bis 2045 vollständige Klimaneutralität erreicht werden soll [7]. CMCs könnten dabei einen wertvollen Beitrag zur Erreichung der Ziele leisten. Um dies zu gewährleisten, müsste jedoch ihr Marktanteil erhöht werden. Die globale Marktgröße von CMCs betrug 2018 etwa 8,1 Billionen USD und 2020 etwa 9,4 Billionen USD und wird, je nach Quelle, für das Jahr 2029 auf etwa 23,3 Billionen USD geschätzt, was einer durchschnittlichen jährlichen Wachstumsrate (CAGR) von etwa 8 – 12 % entspricht. Dabei machte die Luft- und Raumfahrt im Jahr 2018 mit 36 % den größten Anteil aus, gefolgt von Rüstung mit etwa 25% [8, 9].

Der Hauptgrund, welcher CMCs bisher daran gehindert hat, in größerem Maßstab kommerziell Verwendung zu finden, ist in den hohen Herstellungskosten zu sehen. Diese rangieren je nach Herstellmethode zwischen einigen hundert und einigen tausend Euro pro Kilo und damit mehr als eine Größenordnung über Metallen. Neben den hohen Rohmaterialpreisen haben einen großen Anteil daran v.a. die hohen Energiekosten durch Hochtemperaturprozesse, welche einige Tage bis Monate andauern können [10]. Insbesondere bei den Hochtemperaturprozessen besteht ein enormes Einsparungspotenzial, wodurch sowohl Kosten als auch Treibhausgasemissionen gesenkt werden, und die Attraktivität von CMCs für kommerzielle Anwender deutlich gesteigert werden können.

In dieser Arbeit wird der Werkstoff C/C-SiC behandelt, welcher ein CMC beschreibt, das auf in Kohlenstoff-Siliziumkarbidmatrix eingebetteten Kohlenstofffasern basiert. Der Herstellungsprozess von C/C-SiC gliedert sich dabei hauptsächlich in drei Schritte auf:

- Infiltration der Fasern mit Harz und thermische Aushärtung zu einem CFK-Werkstoff (Prozesstemperaturen von ca. 200°C über mehrere Stunden)

- Pyrolyse des CFK-Bauteils zu einem C/C Vorformling (Prozesstemperaturen von ca. 1600°C über mehrere Tage)
- Infiltration des C/C Vorformlings mit flüssigem Silizium (Silizierung) zu einem C/C-SiC Werkstoff (Prozesstemperaturen von ca. 1600°C über mehrere Tage)

Die Herstellungsrouten für C/C-SiC und eine beispielhafte Auswahl anfallender Daten sind in Abbildung 2 verdeutlicht.

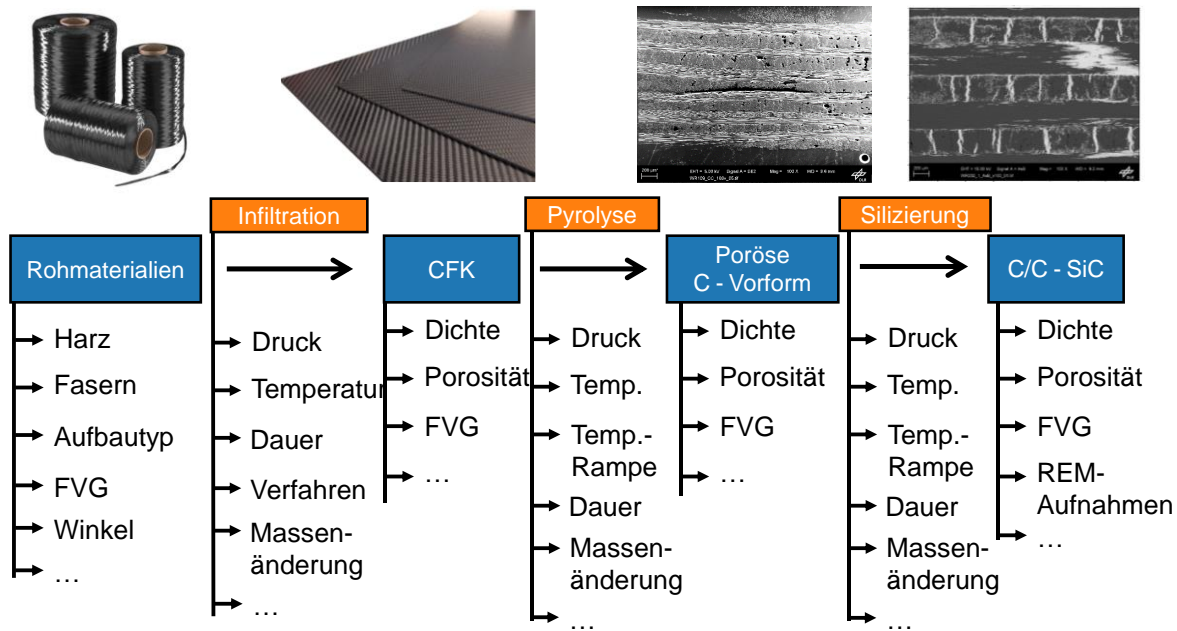


Abbildung 2: Skizze der anfallenden Daten in der Prozesskette der C/C-SiC Herstellung [11, 12].

Die Wahl der Herstellungsparameter sowie Ausgangsrohstoffe hat einen erheblichen Einfluss auf die erzielten mechanischen und thermischen Eigenschaften des Materials. Diese haben wiederum erheblichen Einfluss auf die potenziellen Einsatzgebiete des Endprodukts. Innerhalb der letzten Jahrzehnte wurde deshalb großflächige Forschung betrieben, um das Verständnis für die Zusammenhänge zwischen Herstellung und resultierenden Eigenschaften zu fördern [13–15]. Einen essenziellen Baustein dafür lieferte die Untersuchung von Mikrostrukturen. Durch Auswertungen von REM-Aufnahmen (Rasterelektronenmikroskop-Aufnahmen) konnte gezeigt werden, dass bestimmte Materialeigenschaften mit bestimmten strukturellen Formationen zusammenhängen. Das schadenstolerante Bruchverhalten von CMCs im Vergleich zu monolithischen Keramiken wird beispielsweise ganz erheblich vom Ausmaß der Einzelfasersilizierung beeinflusst [16]. Als Einzelfasersilizierung (EFS) wird das Umschließen einzelner Fasern durch flüssiges Silizium während der Silizierung bezeichnet. Aufgrund der resultierenden Reaktion der Fasern mit Silizium, ist dieses Phänomen meist unerwünscht, da es zur Versprödung des Werkstoffs führt. Liegt eine sehr starke EFS vor, wird die Mikrostruktur als XD-Struktur bezeichnet, ist die EFS nur sehr schwach vorhanden, wird die Struktur als XB-

Struktur bezeichnet [17]. Abbildung 3 zeigt ein Beispiel zweier Mikrostrukturen mit unterschiedlich starkem Anteil an EFS.

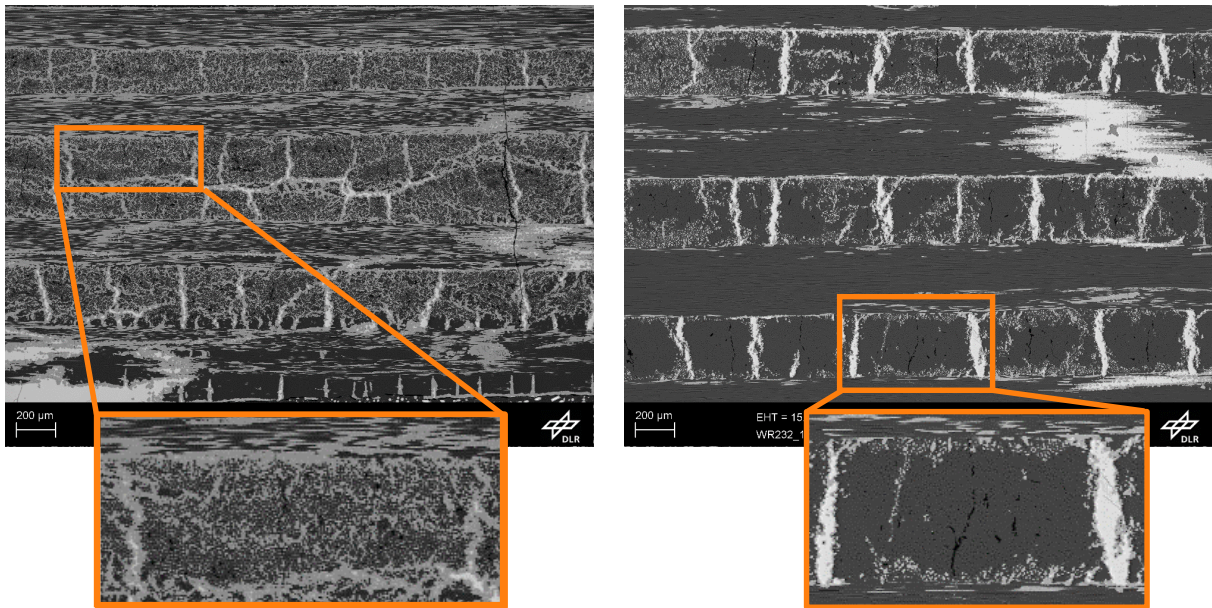


Abbildung 3: Vergleich zweier Mikrostrukturen mit starker EFS (links) und schwacher EFS (rechts); weiß: Silizium, hellgrau: Siliziumkarbid, dunkelgrau: Kohlenstoff, schwarz: Risse/Poren.

1.2 Künstliche Intelligenz in der Materialwissenschaft

Künstliche Intelligenz ist aktuell in allen Industrieländern und Branchen auf dem Vormarsch. Ihrem Einsatz wird das Potenzial zugeschrieben, mittels „Smart Manufacturing“ eine neue Ära der industriellen Produktion einzuleiten [18, 19]. So sieht die KI-Strategie der Bundesregierung beispielsweise vor, Deutschland zu einem globalen Führer auf dem Themengebiet KI-Technologien zu machen, wofür in den Jahren 2019 bis 2022 bereits ein Budget von 2,5 Milliarden Euro bereitgestellt wurde, um die globale Wettbewerbsfähigkeit des Landes zu sichern [20].

Im Bereich der CMCs kann künstliche Intelligenz durch Überwachung und intelligentes Eingreifen in den Produktionsprozess dabei helfen, die hohen Herstellungskosten zu verringern und die Qualität des Endprodukts zu erhöhen. Folglich wurde die Digitalisierung auch in der CMC-Produktion in den letzten Jahren stark vorangetrieben. Die vormals hauptsächlich auf Erfahrung beruhenden Vorgehensweisen wurden damit nach und nach durch statistische Erfassungen und Auswertungen ergänzt. Neben der gesteigerten Effizienz lassen sich dadurch auch Verbesserungen in den Bereichen Materialeigenschaften, Reproduzierbarkeit und Qualitätssicherung erreichen. Durch die enormen Fortschritte im Bereich künstlicher Intelligenz innerhalb des letzten Jahrzehnts wurde außerdem ein komplett neuer Strauß an Möglichkeiten der Datenverarbeitung und -auswertung verfügbar. Sachverhalte, welche sich aufgrund ihrer Größe und Komplexität der menschlichen Auswertung entzogen, oder diese

zumindest extrem erschweren, können nun auf sehr effiziente Weise maschinell ausgewertet werden. Auch Aufgaben, die für Menschen sehr zeitintensiv sind, können durch intelligente Automatisierung bis zu mehreren Größenordnungen schneller erledigt werden [18, 21].

So wurde beispielsweise in Aggour, Gupta et al. [22] ein tiefes Neuronales Netzwerk dazu verwendet, um Mikrostrukturaufnahmen von CMC Bauteilen zu charakterisieren. Dadurch konnten anhand vorliegender Bilder die IST-Zustände der Fasern, Matrix und ggf. Beschichtung, sowie der wahre Faservolumengehalt im Bauteil oder die Poren- und Defektdichte sehr effizient und schnell quantifiziert werden. Die Quantifizierung erlaubt weiterhin eine objektive Vergleichbarkeit zwischen Mikrostrukturen verschiedener Proben, was für eine Optimierung des Herstellprozesses unerlässlich ist, und außerdem deutlich genauer, als eine rein qualitative Aussage durch einen Menschen. In Ghayour et al. [23] wurde eine KI für die Vorhersage der Vickershärte keramischer Proben auf Grundlage erhobener Prozessdaten wie Sintertemperaturen, -zeit und -druck, sowie dem Zusatz von Additiven verwendet. Solche Modelle bieten generell das Potenzial, zeit- und kostenintensive Laborversuchsreihen zu minimieren und dadurch Projektkosten einzusparen. Ein Modell mit ähnlicher Zielsetzung wurde in Xiang, Guanghui et al. [24] dazu verwendet, um den Zusammenhang zwischen Zugfestigkeit von CMC-Proben und diversen Mikrostrukturparametern zu ermitteln.

Trotz der steigenden Anzahl wissenschaftlicher Arbeiten, die sich mit dem Einsatz künstlicher Intelligenz in der Materialwissenschaft befassen, ist die Verbreitung im Bereich der CMCs noch sehr begrenzt. Die vorhandene Literatur bezieht sich größtenteils auf metallische Werkstoffe und nicht-keramische Verbundmaterialien. Weiterhin sind die vorhandenen Datensätze im Bereich der CMCs aufgrund der hohen Materialkosten oft sehr klein, was eine Auswertung durch KI-Methoden erschwert. So wurden die der KI-Auswertung zugrundeliegende Datenbasis in Xiang, Guanghui et al. [24] beispielsweise aus mehreren bereits veröffentlichten wissenschaftlichen Arbeiten zusammengetragen, was allerdings einen hohen Grad an unvollständig ausgefüllten Daten zur Folge hatte.

1.3 Zielsetzung der Arbeit

Das Hauptziel der vorliegenden Arbeit ist die vollständige Digitalisierung des Herstellungsprozesses von C/C-SiC, welches durch das Flüssigsilizierverfahren (*engl.*: Liquid Silicon Infiltration, oder auch LSI-Verfahren) hergestellt wird. Dadurch soll die wissenschaftliche Fragestellung beantwortet werden, welche die wichtigsten Einflussparameter entlang der Produktionskette sind, die zu bestimmten Mikrostruktur-Ausprägungen führen.

Dabei war zu Beginn der Arbeit noch keinerlei digitale Infrastruktur vorhanden, welche zur Bearbeitung der Fragestellung notwendig ist. Das neu zu entwickelnde System musste daher in der Lage sein, Prozessdaten zu sammeln, zu speichern und zu verarbeiten, wobei die Auswertung der Daten auf dem Einsatz künstlicher Intelligenz basieren sollte. Die so ermittelten Korrelationen wurden außerdem anhand von Mikrostruktur-Simulationen überprüft, um auch ein Verständnis für die Kausalitäten zu erhalten. Um die letztendliche Zielgröße „Mikrostruktur“ überhaupt quantifizieren zu können, musste durch das System automatisch eine charakteristische Kennzahl aus den verfügbaren REM-Aufnahmen ermittelt werden. Eine graphische Repräsentation des Umfangs dieser Arbeit ist in Abbildung 4 gegeben [25].

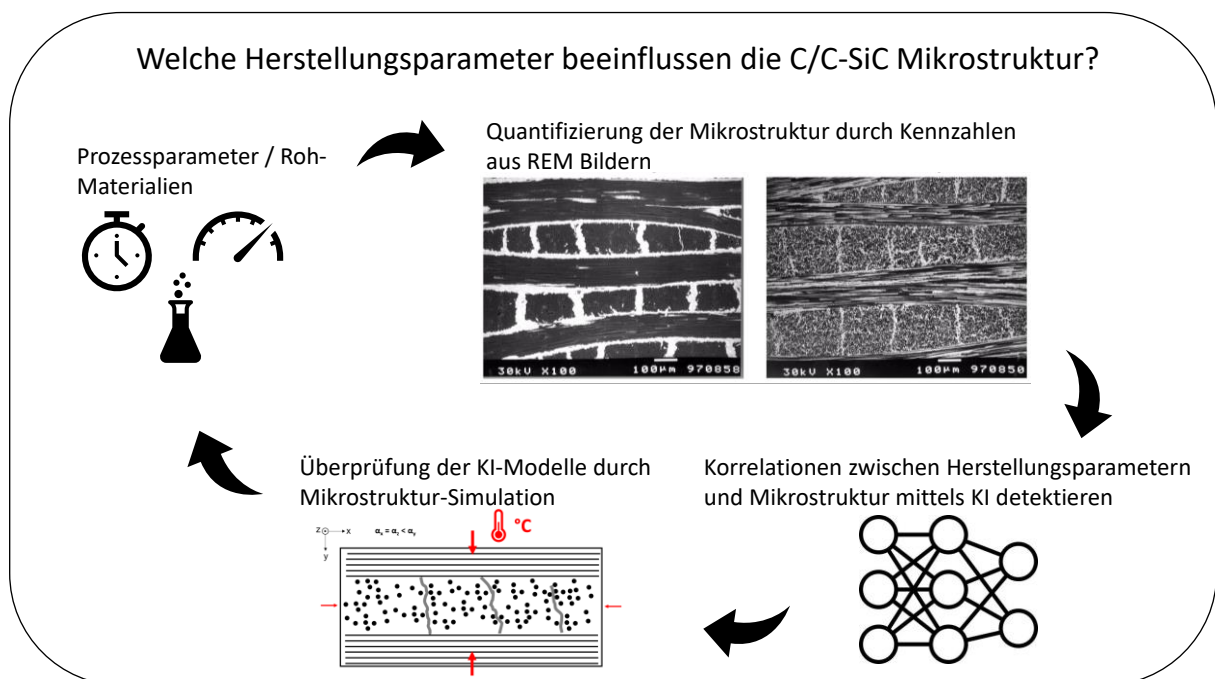


Abbildung 4: Haupt-Fragestellung und Übersichtsplan der vorliegenden Arbeit.

In einer vorangegangenen Arbeit am Deutschen Zentrum für Luft- und Raumfahrt (DLR) wurde durch Jain [5] bereits der Zusammenhang zwischen (simulierter) Mikrostruktur und (simulierten) mechanischen Eigenschaften durch KI-Methoden untersucht (siehe gestrichelte Umrandung in Abbildung 5). Um den Kreis zu schließen, wurden in der vorliegenden Arbeit nun die Zusammenhänge zwischen Herstellungsparametern und Mikrostruktur ermittelt (siehe grau eingefärbter Bereich in Abbildung 5). Sie liegt also im Produktionsverlauf vor der Arbeit von Jain und endet ohne Überschneidung dort, wo letztere beginnt. Ein weiteres Unterscheidungsmerkmal ist, dass bei Jain noch, mangels realer Daten, auf Simulationen zurückgegriffen wurde, um die Datenbasis für das Anlernen der KI-Modelle zu bieten. Im Gegensatz dazu wurden in dieser Arbeit dafür Herstellungs- und Prozessdaten verwendet.

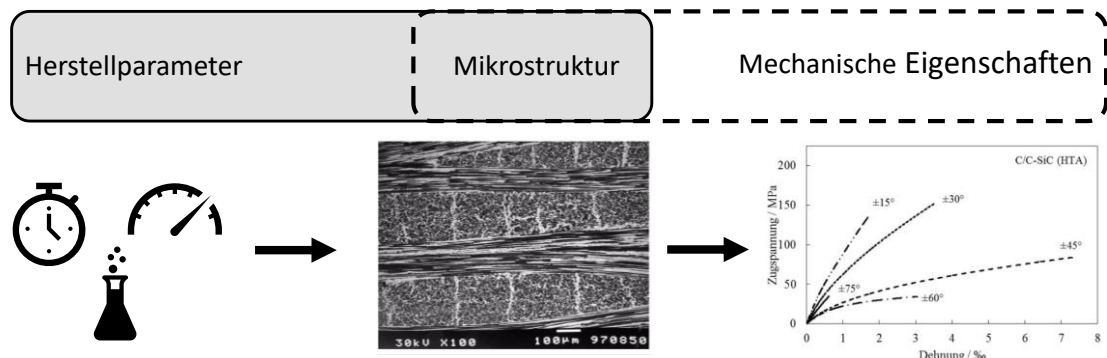


Abbildung 5: Dreiklang zwischen Herstellungsparametern, Mikrostruktur und mechanischen Eigenschaften inklusive Abgrenzung des in dieser Arbeit untersuchten Bereichs (grau hinterlegt) von dem in Jain [5] untersuchten Bereich (gestrichelte graue Linie).

Vor Beginn der Arbeit wurde die Dokumentation der wissenschaftlichen Tätigkeiten in der Abteilung Keramische Verbundstrukturen (KVS) des DLR Stuttgart hauptsächlich durch Excel-Listen und andere lokale Dateiformate geführt. Um eine vollständige Digitalisierung der Herstellungsrouten zu ermöglichen, müssen daher folgende Punkte erfüllt werden:

1. Erstellung eines Systems, welches Daten sammeln, verarbeiten, speichern und darstellen kann. Die Bedienung soll über ein einfaches Web-Interface erfolgen, um dessen Benutzung auch für Mitarbeiter ohne Programmier-Hintergrund zu ermöglichen. Hierfür wird ein lokaler Server erstellt, auf den ein Zugriff per Internetbrowser möglich ist. Die Datenspeicherung wird über eine PSQL-Datenbank gelöst. Die Datenverarbeitung erfolgt automatisch im Hintergrund durch Python Skripte über ein Django-Backend.
2. Auf dieser Grundlage kann eine datenbasierte Modellbildung erfolgen, welche Zusammenhänge zwischen Eingangs- und Ausgabeparametern knüpfen und Prognosen tätigen kann. Dies wird mithilfe von KI-Methoden verwirklicht, welche über ein in Python programmiertes Graphical User Interface (GUI) zugänglich gemacht werden. Neben der Auswertung ist dieses später als „DataTracker“ bezeichnete Tool auch für die Visualisierung von Ergebnissen aus der Datenbank zuständig.
3. Parallel erfolgt eine physikalische Modellierung der Pyrolyse in Multimech, um ein Verständnis für die internen Vorgänge im Material zu erschaffen. Die Ergebnisse werden anschließend mit den datenbasierten Modellen abgeglichen.

Diese Vorgehensweise ist graphisch in Abbildung 6 dargestellt.

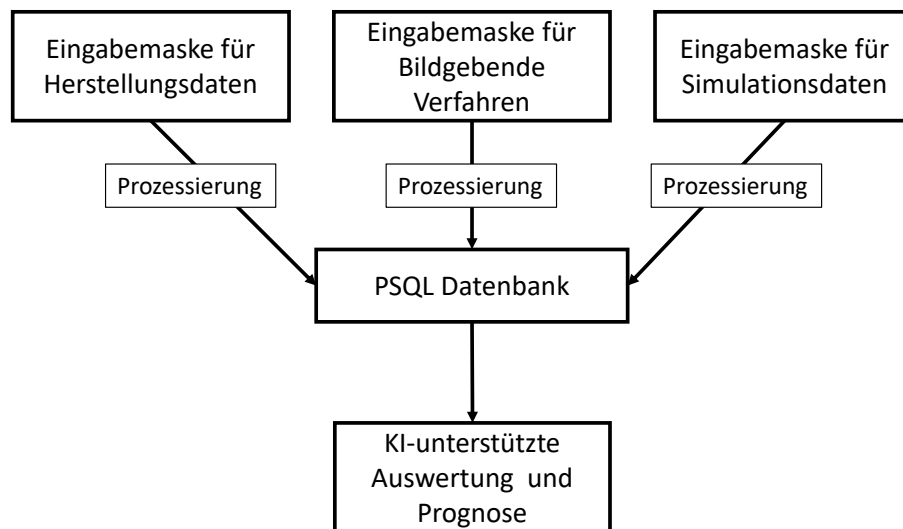


Abbildung 6: Übersicht über die notwendigen Schritte zur Digitalisierung der Prozesskette von C/C-SiC.

Eine Zusammenfassung der in dieser Arbeit verwendeten Programmiersprachen, Auszeichnungssprachen und Frameworks ist in der folgenden Auflistung gegeben:

- Django
- Python
- JavaScript
- HTML
- CSS
- PSQL

Zusätzlich wurde für die Mikrostruktursimulation folgende kommerzielle Software verwendet:

- Simcenter Multimech 2022 (Siemens)

Als Datengrundlage für die KI-Modelle wurde ein Datensatz von 163 C/C-SiC Proben verwendet, welcher innerhalb der letzten Jahre am DLR in Stuttgart hergestellt und dokumentiert wurde. Einige wenige Proben wurden auch im Zuge dieser Promotionsarbeit hergestellt und ausgewertet. Dem für die CMC-Welt vergleichsweise großen Datensatz stand die ebenfalls große Anzahl von 45 Herstellungsparametern gegenüber, welche zu unterschiedlichen Zeitpunkten der Produktion aufgenommen, und je nach Probe unterschiedlich gut dokumentiert wurden. Einige Beispiele aufgenommener Herstellungsparameter wurden bereits in Abbildung 2 gezeigt.

Insgesamt ist diese Arbeit in sieben Kapitel untergliedert. Kapitel 1 ist mit dieser Einführung in das Thema beendet. In Kapitel 2 werden die notwendigen theoretischen Grundlagen für die verwendeten mathematischen Methoden, Modelle und Systematiken umfassend erläutert.

Kapitel 3 und 4 befassen sich mit den beiden in dieser Arbeit entwickelten Programmen, welche für die Datenerfassung, -verarbeitung und -speicherung verwendet wurden. Dabei handelt es sich bei Kapitel 3 um ein Web-Interface, welches über einen Internet-Browser erreichbar ist, und über welches Prozessdaten einfach und übersichtlich eingegeben werden können. Eine Verarbeitung und Speicherung der Daten in eine PSQL-Datenbank erfolgt dabei automatisch durch ein Django-Backend. Kapitel 4 befasst sich mit dem Programm „DataTracker“, welches sich einfach über eine Installationsdatei auf jedem DLR-Rechner installieren lässt. DataTracker ist in der Lage, die in der Datenbank gespeicherten Daten zu visualisieren, sowie KI-Modelle zu trainieren und anzuwenden. Einen gleichermaßen wichtigen wie komplexen Prozessschritt innerhalb der C/C-SiC Herstellung stellt die Pyrolyse dar. Deswegen wird in Kapitel 5 die physikalische Modellbildung des Pyrolyseprozesses anhand eines FEM-Modells mit der kommerziellen Software Simcenter Multimech 2022 behandelt. Hier wird anhand von Mikrostruktursimulationen untersucht, welche Faktoren einen besonders großen Einfluss auf das entstehende Rissmuster der Probe haben. Weiterhin wird in diesem Kapitel auf die Materialtests zur Ermittlung der mechanischen Eigenschaften für die Simulationen eingegangen und der Pyrolyseprozess praktisch untersucht. Die Ergebnisse werden anschließend mit den datenbasierten Modellen abgeglichen. In Kapitel 6 erfolgt eine ausführliche Diskussion der gewonnenen Ergebnisse. Kapitel 7 rundet die Arbeit schließlich mit einer Zusammenfassung und einem Ausblick auf zukünftig mögliche Erweiterungen ab.

2 Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen erläutert, welche für die Anwendung der ausgewählten Verfahren und Modelle notwendig sind. Es eignet sich besonders zum Nachschlagen später verwendeter Fachtermini, falls diese dem Leser nicht geläufig sind.

2.1 Physikalische und datenbasierte Modelle

Im Verlauf der Arbeit wird häufig von physikalischen und datenbasierten Modellen gesprochen, weswegen an dieser Stelle eine Definition der beiden Arten aufgestellt wird.

Ein Modell wird als „physikalisch“ bezeichnet, wenn es aufgrund von vordefinierten Zusammenhängen (Formeln) eine physikalische Berechnung durchführt, wie es beispielsweise in der Finite Elemente Methode (FEM) gemacht wird. Ergebnisse physikalischer Modelle sind für den Fachmann transparent und vollständig nachvollziehbar.

Auf der anderen Seite wird ein Modell als „datenbasiert“ bezeichnet, wenn es statistische Zusammenhänge zwischen mehreren Variablen selbst erlernt, ohne dass diese vorgegeben wurden, und daraufhin durch einen Wissenstransfer zu Prognosen über bisher unbekannte Situationen fähig ist. Das Paradebeispiel dafür ist die künstliche Intelligenz. Datenbasierte Modelle erfordern keine vom Menschen vorgegebenen Rechenvorschriften, sondern können diese selbst erstellen. Allerdings ist am Ende nicht immer ersichtlich, wie ein solches Modell zu seinen Erkenntnissen gelangt ist. Die Kern-Merkmale von physikalischer und datenbasierter Modellierung sind in Tabelle 1 festgehalten.

Tabelle 1: Unterschiede zwischen physikalischer und datenbasierter Modellierung.

	Physikalisches Modell (FEM)	Datenbasiertes Modell (KI)
Kurzbeschreibung	Trifft Vorhersagen aufgrund bekannter Formeln und Gesetzmäßigkeiten, Ergebnis genau erklärbar	Trifft Vorhersagen aufgrund erlernter Zusammenhänge, oft jedoch Black Box
Grundlage für Vorhersage	Vorgegebene Zusammenhänge	Selbst erlernte Zusammenhänge
Rechenzeit	Oft lang, abhängig von der Größe und Komplexität der Zusammenhänge	Eher kurz, sobald angelernt, Training aber möglicherweise langwierig
Genauigkeit abhängig von...	Komplexität & Größe des Problems, bei numerischen	Anzahl & Qualität der Trainingsdaten

2.2 Begriffsdefinitionen KI und ML

Der Begriff der künstlichen Intelligenz (KI) umfasst ein weites Feld von Anwendungsgebieten. Marvin Minsky, einer der frühesten Forscher auf dem Gebiet der künstlichen Intelligenz, definierte diesen Überbegriff 1956 als „Die Wissenschaft davon, Maschinen dazu zu bringen, Dinge zu tun, deren Ausführung vom Menschen Intelligenz erfordert“ [26]. Machine Learning (ML) ist wiederum ein Teilbereich der künstlichen Intelligenz, und bezeichnet die Fähigkeit, Wissen aus Erfahrung zu generieren. Oft werden die beiden Begriffe jedoch synonym verwendet. Im maschinellen Lernen existiert wiederum eine Vielzahl von Algorithmen, welche jeweils über spezielle Vor- und Nachteile verfügen. Eine Übersicht über die Begriffsabgrenzungen ist in Abbildung 7 dargelegt [27].

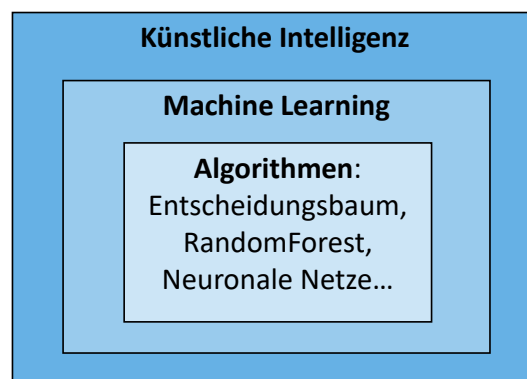


Abbildung 7: Begriffszwiebel für künstliche Intelligenz [27].

In dieser Arbeit wurden vier verschiedene Algorithmen verwendet und miteinander verglichen, auf die an späterer Stelle detaillierter eingegangen wird. Wird eine Instanz eines Algorithmus anhand eines spezifischen Trainings-Datensatzes trainiert, so wird diese als „Modell“ bezeichnet. Unabhängig davon kann maschinelles Lernen in drei verschiedene Arten untergliedert werden:

1. Überwachtes Lernen
2. Nicht-überwachtes Lernen
3. Bestärkendes Lernen

Das überwachte Lernen entspricht im biologischen Vergleich dem Lernen mit einem Lehrer. In diesem Fall liegt die richtige Antwort für eine Frage also vor, sodass die vorhergesagte Größe direkt mit dem wahren Wert verglichen werden, und damit die Größe des Fehlers festgestellt

werden kann. Anhand dieser Information kann eine Anpassung des Algorithmus erfolgen, die zu einer Verbesserung führt [28, 29].

Im Gegensatz dazu stehen beim Nicht-überwachten Lernen keine Lösungen zu den Trainingsfragen zur Verfügung. Hier muss das System selbst explorativ nach vorher unbekanntem Mustern und Zusammenhängen in den nicht-kategorisierten Daten suchen [28]. Nicht-überwachtes Lernen wird beispielsweise für Clustering und Segmentierungen eingesetzt, lässt sich allerdings nicht für die Beantwortung der hier vorliegenden Fragestellung verwenden, weshalb es für diese Arbeit nicht relevant ist.

Bei der dritten Variante, dem bestärkenden Lernen, wird der Algorithmus durch Belohnung und Bestrafung trainiert, wodurch, wie beim überwachten Lernen, ein Feedback von außen vorliegt. Allerdings ist dieses Feedback nur qualitativ und gibt daher keinen Aufschluss über die Größe des Fehlers. In einer bestimmten Situation wird lediglich festgestellt, ob eine bestimmte Aktion positive oder negative Auswirkungen hat [30]. Die Parameteranpassung geschieht für gewöhnlich so lange, bis sich ein Gleichgewichtszustand einstellt [29]. Bestärkendes Lernen kann daher als Mischform von überwachtem und nicht-überwachtem Lernen verstanden werden.

Aufgrund der Natur des vorliegenden Problems, wird in dieser Arbeit das überwachte Lernen verwendet. Hat man die Art des Problems und die damit verbundene Lernart identifiziert, gibt es eine Vielzahl an unterschiedlichen Algorithmen zur Auswahl. In der einschlägigen Literatur finden sich Erfahrungswerte dafür, welche Algorithmen sich für welche Problemstellungen besonders gut eignen. Beispielsweise erreichen Gefaltete Neuronale Netzwerke (*engl.*: Convolutional Neural Networks) besonders hohe Genauigkeiten bei Bilderkennungsaufgaben, was allerdings nur für den Fall gilt, dass auch die nötige Mindestanzahl an Trainingsdaten vorliegt [27]. Trifft dies nicht zu, können traditionelle Machine-Learning Ansätze wie RandomForests oder Entscheidungsbäume bessere Ergebnisse liefern. In manchen Fällen ist es auch vorteilhaft, mehrere Algorithmen miteinander zu vergleichen, um das beste Resultat zu erhalten. Im Zuge dieser Arbeit wurden folgende KI-Algorithmen untersucht, um die Korrelationen zwischen Herstellungsparametern und Mikrostruktur von C/C-SiC zu untersuchen:

- Entscheidungsbaum (DT)
- RandomForest (RF)
- Künstliches Neuronales Netzwerk (NN)
- Lasso Regression (LR)

Doch selbst nach der Auswahl eines bestimmten Algorithmus, gibt es noch viel Spielraum, um dessen Vorhersagegenauigkeit zu beeinflussen. Denn die Parameter des Algorithmus selbst, die sog. „Hyperparameter“, beeinflussen die Vorhersage in großem Maße und werden daher im späteren Verlauf der Arbeit für jedes Modell bestimmt (siehe Kapitel 6.1.6). So verfügt ein RandomForestRegressor aus dem Modul ScikitLearn über 17 einstellbare Hyperparameter, beispielsweise die Anzahl der Bäume oder die Minimalanzahl von Proben, welche vorliegen muss, um ein Blatt zu bilden. Der Vorgang, die optimalen Parameter für den Algorithmus zu finden, wird dabei als „Hyperparametertuning“ bezeichnet. Meistens wird dabei für jeden Parameter ein Bereich festgelegt, innerhalb dessen er variiert wird. Aus den Variationsbereichen aller Parameter wird dann eine Matrix gebildet. Anschließend wird eine zufällige Anzahl an Kombinationen aus dieser Matrix getestet und das beste Modell beibehalten [31].

2.3 KI-Algorithmen

Im Laufe der Zeit haben sich im Bereich der künstlichen Intelligenz unterschiedliche Verfahren etabliert, von denen alle über die Grundfähigkeiten und -eigenschaften des maschinellen Lernens verfügen, die sich aber in Funktionsweise, Methodik und Aufbau stark unterscheiden können. Eine Auswahl dieser Algorithmen wird nun in den folgenden Unterkapiteln näher erläutert.

2.3.1 Entscheidungsbäume

Entscheidungsbäume werden aufgrund ihrer guten Interpretierbarkeit häufig in der Geo- und Materialwissenschaft verwendet [32]. Sie stellen hierarchische Verkettungen von Abfragen dar, anhand derer eine Prognose für eine Zielgröße abgegeben werden kann. Welche Abfragen in welcher Reihenfolge gestellt werden, wird dabei durch den Algorithmus selbst bestimmt [33]. Dabei können unterschiedliche zugrundeliegende Algorithmen bei der Erstellung der Bäume Anwendung finden. Entscheidungsbäume können gleichermaßen für numerische als auch kategorische Daten verwendet werden. Die Funktionsweise von Entscheidungsbäumen ist in Abbildung 8 an einem einfachen Beispiel skizziert. Die Zielgröße in dieser Arbeit ist die Carbon Conversion Ratio (CCR), welche durch die automatische Auswertung von REM-Bildern für jede Probe ermittelt wird, wie später in Kapitel 3.3.1 näher erläutert wird. Es soll nun eine Prognose für die CCR einer unbekannt Probe abgegeben werden, zu der keine REM Bilder vorliegen, deren Prozessparameter jedoch bekannt sind. In der realen Anwendung sind Entscheidungsbäume häufig deutlich komplexer als in Abbildung 8 skizziert. Die Positionen

innerhalb eines Entscheidungsbaums, an denen die Daten durch eine Abfrage in zwei verschiedene Pfade aufgeteilt werden, werden als „Knoten“ bezeichnet. Als „Blätter“ bezeichnet man hingegen diejenigen Positionen, welche sich am Ende eines Pfades befinden und die Prognose für die Zielgröße beinhalten.

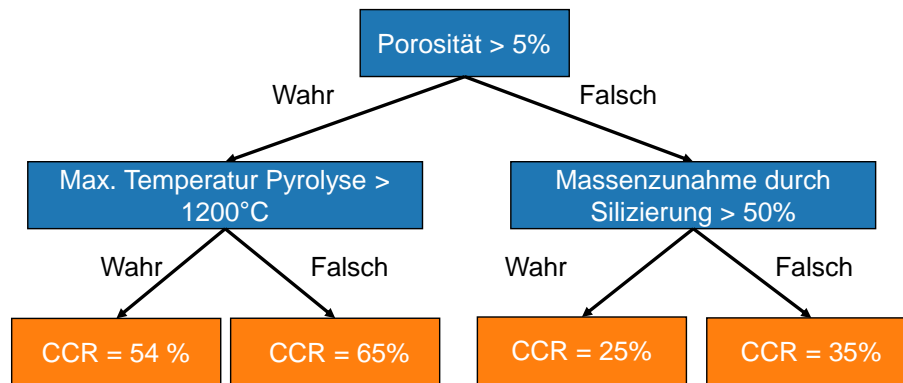


Abbildung 8: Vereinfachte Darstellung eines Entscheidungsbaums zur Bestimmung der Carbon Conversion Ratio (CCR) anhand verschiedener Prozessparameter. Blau: Knoten, Orange: Blätter.

Um zu entscheiden, welche Fragen an welchen Knoten des Baums verwendet werden sollen, muss die Güte der Unterteilungen quantifiziert werden. Hierfür wird für Klassifizierungsaufgaben der Information Gain (IG) oder die Gini Impurity verwendet [34]. Bei Regressionsaufgaben wird oft die mittlere quadratische Abweichung (MSE von *engl.*: Mean Squared Error) herangezogen, welche nach Gleichung (1) berechnet wird. Durch den MSE kann für jede Unterteilungsmöglichkeit quantifiziert werden, um wie viel der Modellfehler durch sie zu- oder abnimmt.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (1)$$

Dabei bezeichnet Y_i den wahren Wert der Zielgröße i , während \hat{Y}_i den vom Modell berechneten Wert für i bezeichnet, wobei insgesamt eine Anzahl von n Messpunkten vorliegt. Die MSE wird für alle verfügbaren Parameter und innerhalb eines Parameters für alle denkbaren Unterteilungen der Messpunkte berechnet. Das so gefundene Minimum wird anschließend als Aufteilungs-Kriterium verwendet, und so ein neuer Knoten gebildet. Das Vorgehen wird für alle darauffolgenden Knoten wiederholt, bis ein Abbruchkriterium erreicht wird. Abbildung 9 zeigt beispielhaft eine solche Unterteilung von Messpunkten in zwei Kategorien, was einer Abfrage an einem einzelnen Knoten im Entscheidungsbaum entspricht. Je nachdem, wie die Grenze (orangefarbene Linie in Abbildung 9) gesetzt wird, ergeben sich andere Fehlerwerte (MSE). Die Unterteilung mit dem geringsten MSE wird ein Kandidat für einen Knoten. Nun kann für jeden Parameter die Unterteilung mit dem minimalen MSE

gefunden werden. Anschließend werden unter all diesen Kandidaten dann der Parameter und die Unterteilung mit dem insgesamt geringste MSE ausgewählt, um einen Knoten zu bilden.

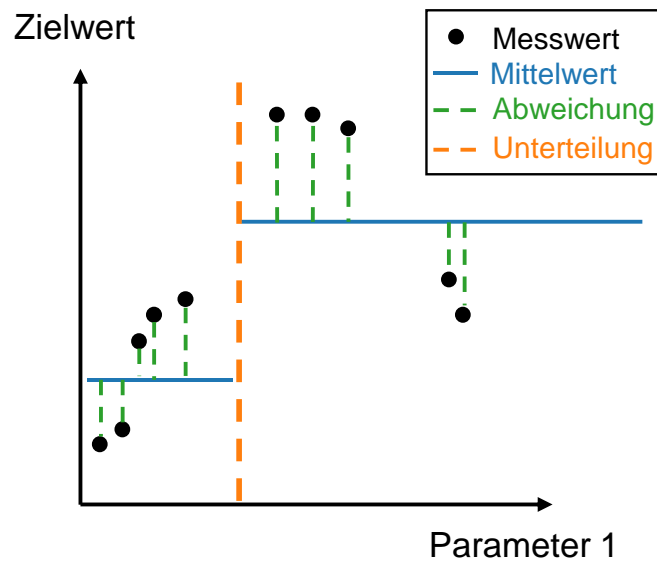


Abbildung 9: Beispielhafte Darstellung einer einzelnen Unterteilung von Messpunkten eines beliebigen Parameters in zwei Kategorien mit ihren jeweiligen Mittelwerten. Die Wahl der Grenze (orange) hat Einfluss auf die Mittelwerte (blau) und Abweichungen (grün).

Das gesamte Vorgehen zur Bildung eines Entscheidungsbaums für Regressionsaufgaben ist in Schema 1 deutlich gemacht.

Schema 1: Vorgehen bei der Bildung eines Entscheidungsbaums für Regressionsprobleme

Schritt 1: Für jeden Parameter P_i des Baums...

Schritt 1.1: Für alle Unterteilungsmöglichkeiten U_n der Datenpunkte...

Schritt 1.1.1: Berechne die Varianz-Reduktion des Modells durch die Verwendung von U_n als Knotenfrage für P_i für beide neu entstehenden Knoten

Schritt 1.1.2: Gewichte die Varianz-Reduktion für beide Knoten anhand der Anzahl der betroffenen Proben und summiere zur Gesamt-Varianz

Schritt 1.2: Speichere die Unterteilung U_n mit der höchsten Varianz-Reduktion als Kandidat für die tatsächlich verwendete Unterteilung

Schritt 2: Wähle aus allen Kandidaten die Unterteilung mit der größten Varianz-Reduktion und teile die Daten, sodass zwei neue Knoten entstehen

Schritt 3: Wiederhole die Schritte 1-2 für alle neu entstandenen Knoten solange, bis die Varianz einen vorgegebenen Grenzwert unterschreitet

Aufgrund ihrer Simplität sind Entscheidungsbäume sehr intuitiv und einfach zu verstehen, wodurch auch eine klare Ableitung von Regeln erfolgen kann [33]. Weiterhin vorteilhaft ist die Fähigkeit, sowohl kategorische als auch numerische Werte verarbeiten zu können. Nachteilig ist die bei großen Probenanzahlen steigende Berechnungskomplexität, die Neigung zur Überanpassung an die Daten, sowie die oft geringere Genauigkeit im Vergleich zu komplexeren

ML-Methoden [33]. Auf die Überanpassung an die Daten, auch „Overfitting“ genannt, wird in Kapitel 2.6.2 detaillierter eingegangen.

2.3.2 RandomForest Algorithmus

Der RandomForest Algorithmus ist ein Ensemble-Lernalgorithmus, welcher auf einer Vielzahl an unterschiedlichen Entscheidungsbäumen basiert [35]. Empirische Studien konnten zeigen, dass Ensemble-Lernalgorithmen oft eine höhere Vorhersagegenauigkeit, sowie eine geringere Neigung zum Overfitting aufweisen, als ihre Basis-Estimatoren [36]. Der RandomForest Algorithmus ist außerdem besonders geeignet für Probleme, bei denen die Anzahl der Variablen deutlich größer ist, als die Anzahl der Beobachtungen, und kann, wie Entscheidungsbäume, auch nicht-lineare Zusammenhänge verarbeiten [35]. Zudem wird automatisch die Wichtigkeit der einzelnen Variablen berechnet und erlaubt so ein Ranking. Aus dem Gesamt-Datensatz wird durch Bootstrapping (siehe Kapitel 2.6.4) ein Stichproben-Datensatz für jeden Baum des RandomForest gebildet. Mit jedem dieser Datensätze wird ein einzigartiger Entscheidungsbaum trainiert, sodass sich die Bäume untereinander aufgrund der unterschiedlichen zugrundeliegenden Daten unterscheiden. Die Genauigkeit eines einzelnen Baums kann durch die out-of-bag Datenpunkte (OOB) bestimmt werden, also durch diejenigen Datenpunkte, welche durch Bootstrapping nicht in dem Datensatz gelandet sind, der zum Trainieren des Baums herangezogen wurde. Dieses Vorgehen ist auch in Abbildung 10 schematisch dargestellt. Soll nun eine Prognose für eine bestimmte Problemstellung abgegeben werden, wird jeder einzelne Entscheidungsbaum des Ensembles befragt und durch Aggregation das endgültige Ergebnis ausgegeben.

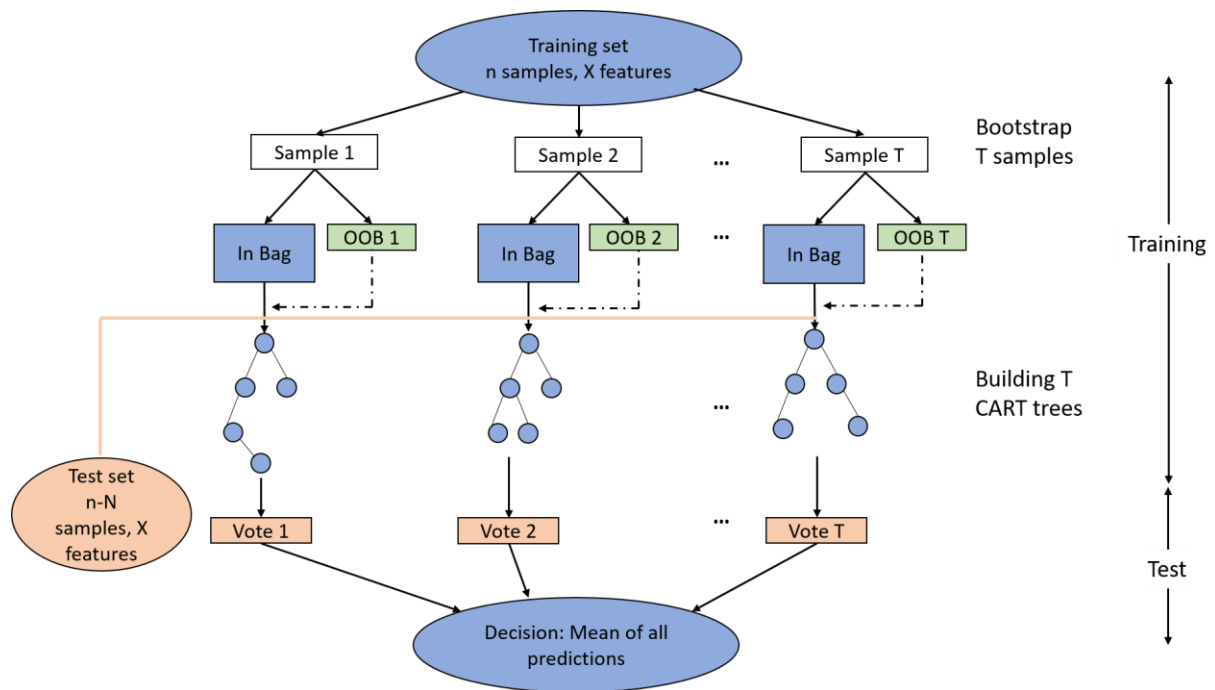


Abbildung 10: Vorgehen beim Erstellen der Estimatoren eines RandomForest [37, 25].

Wie bereits beschrieben, weist der RandomForest Algorithmus aufgrund seiner geringeren Neigung zum Overfitting oft Vorteile hinsichtlich Robustheit und Vorhersagegenauigkeit gegenüber dem Entscheidungsbaum auf. Nachteilig ist neben dem höheren Rechenaufwand vor allem die schlechtere Nachvollziehbarkeit, was die abgegebene Vorhersage betrifft.

2.3.3 Neuronale Netzwerke

Neuronale Netzwerke sind von der Biologie inspirierte Informations-Verarbeitungs-Paradigmen, welche in ihrem Aufbau dem menschlichen Gehirn ähneln [29]. Die Verbreitung neuronaler Netzwerke hat v.a. in den letzten Jahren durch die schnelle Entwicklung von leistungsfähigeren Rechnern stark zugenommen [27]. Neuronale Netze lassen sich in die folgenden Architekturen unterteilen:

- Vollständig verbundene neuronale Netze
- Gefaltete neuronale Netze
- Rekurrente neuronale Netze

Gefaltete neuronale Netze werden v.a. in der Bilderkennung verwendet. Sie bedienen sich einiger Schichten, welche Bilddateien im Kodierungsblock vor-verarbeiten, bevor sie im Prädiktionsblock in einem vollständig verbundenen neuronalen Netzwerk münden [27]. Rekurrente neuronale Netze werden v.a. bei sequentiellen Daten verwendet, also dort wo die Reihenfolge eine Rolle spielt, beispielsweise bei Aktienkursen.

In dieser Arbeit werden ausschließlich vollständig verbundene neuronale Netze verwendet (ANN oder KNN). Diese setzen sich aus mehreren hintereinandergeschalteten Schichten (*engl.*: Layers) aus Neuronen zusammen, welche Informationen durch Verknüpfungen aneinander weitergeben können. In der ersten Schicht, der sog. Input-Schicht, werden die Merkmale des Problems eingegeben. Dies entspricht in dieser Arbeit beispielsweise den Probenkennwerten (Fasermaterial, Probendicke, Porosität...) und Herstellungsparametern (Pyrolysetemperatur, Dauer der Aushärtung...). In der letzten Schicht wird dann eine Prognose passend zum jeweiligen Problem abgegeben, wobei diese Schicht aus einem oder mehreren Neuronen bestehen kann. In dieser Arbeit besteht sie aus einem einzelnen Neuron, welches einen Wert für die Carbon Conversion Ratio ausgibt. Dazwischen gibt es eine oder mehrere Schichten von Neuronen, die sog. „Versteckten Schichten“ (*engl.*: Hidden Layers), welche jeweils eine beliebige Anzahl von Neuronen enthalten können. In einem vollständig verbundenen Neuronalen Netzwerk sind alle Neuronen einer Schicht mit den Neuronen der Folgeschicht durch Gewichte verbunden. Abbildung 11 beschreibt ein einfaches vollständig verbundenes Neuronales Netz mit einer Input-Schicht, zwei versteckten Schichten und einer Output-Schicht.

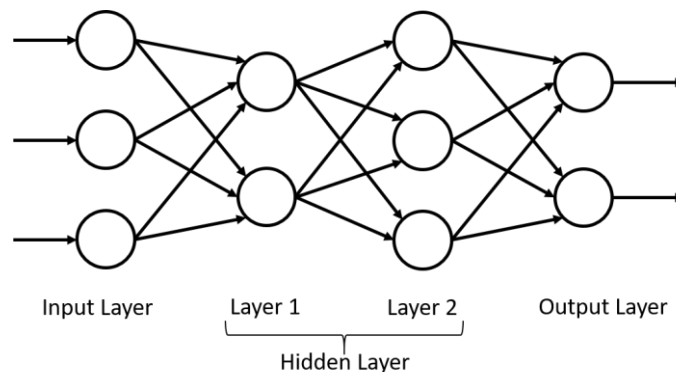


Abbildung 11: Modell eines vollständig verknüpften neuronalen Netzwerks mit einer Input-Schicht, zwei versteckten Schichten und einer Output-Schicht [29].

Um zu verstehen, wie Neuronale Netze funktionieren, muss zunächst das Neuron erklärt werden. Abbildung 12 zeigt den Aufbau eines künstlichen Neurons mit mehreren Eingängen X_p und einem Ausgang v_k . Die Eingänge entsprechen dabei Zahlenwerten, welche mit den Gewichten w_{kp} multipliziert und anschließend aufsummiert werden. Diese Summe wird jedoch nicht einfach ausgegeben, sondern erst durch eine Aktivierungsfunktion auf den Bereich zwischen 0 und 1 projiziert. Wird ein Schwellenwert überschritten, gibt das Neuron ein Signal weiter, welcher letztlich den Output eines Neurons darstellt.

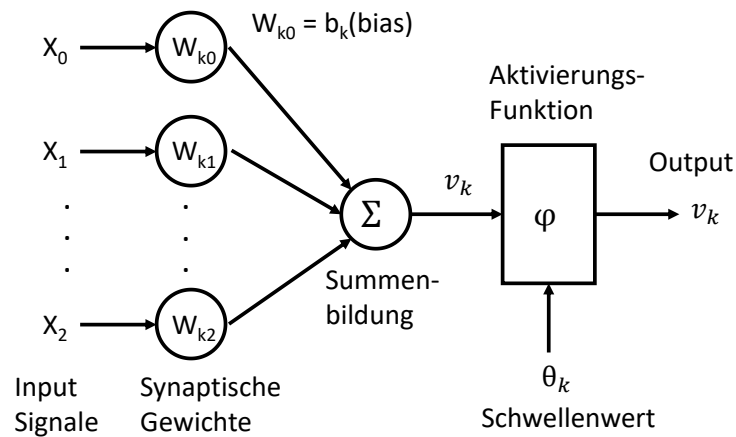


Abbildung 12: Modell eines künstlichen Neurons mit mehreren gewichteten Eingängen, Summenbildung und einer Aktivierungsfunktion, welche eine Ausgabe zwischen 0 und 1 erzeugt [29].

Für die Aktivierungsfunktion gibt es viele in der Literatur erwähnte Beispiele, von denen eine häufig verwendete Auswahl in Tabelle 2 aufgeführt ist [27, 38–40]:

Tabelle 2: Häufig verwendete Aktivierungsfunktionen.

Aktivierungs-funktion	Formel	Bemerkung
Sigmoide	$f(x_i) = \frac{1}{1+e^{x_i}}$	Sehr langsame Gewichts-anpassung in Bereichen geringer Steigung („Vanishing-Gradient“ Problem)
Rectified Linear Unit (ReLU)	$f(x_i) = \max(0, x_i)$	Kein „Vanishing-Gradient“-Problem, schnellere Berechnung als Sigmoide und Tangens Hyperbolicus, „Dead-Neuron“ Problem kann auftreten
Exponential Linear Unit (ELU)	$f(x_i) = \begin{cases} x & \text{für } x > 0 \\ \alpha(e^x - 1) & \text{für } x \leq 0 \end{cases}$	Kein „Vanishing-Gradient“ Problem, kein „Dead-Neuron“ Problem
Softmax	$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$	Wird hauptsächlich in der letzten Schicht eines Neuralen Netzwerks verwendet, da bei Klassifizierung direkt Wahrscheinlichkeiten der Klassen ausgegeben werden. Allerdings besteht das „Vanishing-Gradient“ Problem
Tangens Hyperbolicus	$f(x_i) = \frac{\sinh(x_i)}{\cosh(x_i)}$	Konvergiert schneller als Sigmoide, da symmetrisch um den Ursprung, aber „Vanishing-Gradient“ Problem

Auf die Phänomene „Vanishing-Gradient“ und „Dead Neuron“ wird im Folgenden bei der Erläuterung der Aktivierungsfunktionen näher eingegangen.

Aktivierungsfunktion

Eine Aktivierungsfunktion in einem KNN definiert, wie hoch der Output eines Neurons bei einem gegebenen Satz von Inputs ist [40]. Inspiration für dieses Vorgehen ist erneut das menschliche Gehirn, in welchem unterschiedliche Neuronen durch unterschiedlich starke Reize aktiviert werden [27]. Ein Beispiel hierfür ist die Sigmoide-Funktion, welche in Abbildung 13 dargestellt ist. Die Sigmoide-Funktion $g(x_i) = \frac{1}{1+e^{x_i}}$ skaliert die gewichtete Summe aus Inputs eines Neurons (siehe Abbildung 12) auf den Bereich 0 bis 1. Ihre Steigung berechnet sich zu $g'(x_i) = \frac{1}{1+e^{x_i}} \left(1 - \frac{1}{1+e^{x_i}}\right)$.

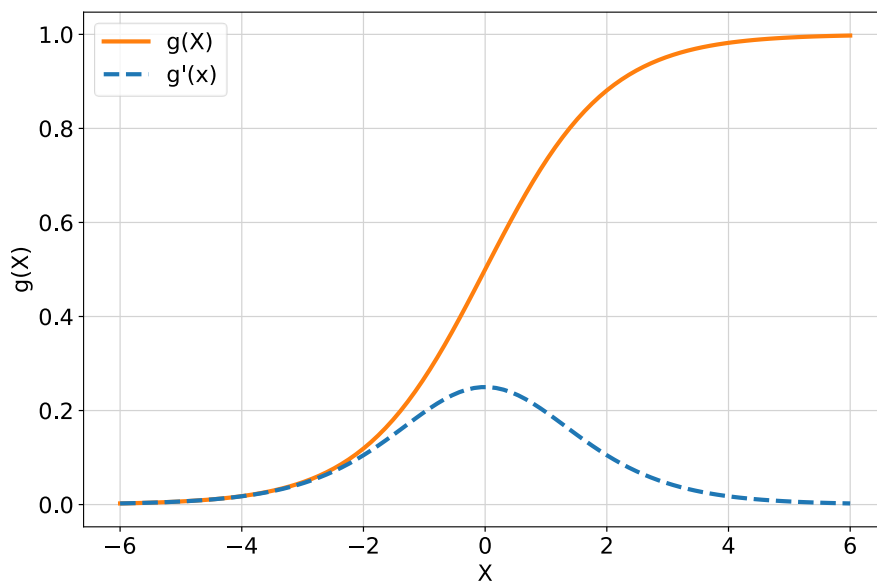


Abbildung 13: Graphische Darstellung der Sigmoide-Funktion $g(x)$ und ihrer Ableitung $g'(x)$ [40].

Obwohl die Sigmoide früher häufig verwendet wurde, besitzt sie das Problem, dass ihre Steigung für sehr große und sehr kleine Werte schnell sehr klein wird. Dabei wird der Gradient von $g'(x) = 0$ erst bei einem Input von $\pm\infty$ erreicht, was als „weich-sättigend“ bezeichnet, und mathematisch wie folgt dargestellt werden kann [40]:

$$\lim_{x \rightarrow \pm\infty} g'(x) = 0 \quad (2)$$

Dies führt zu einer nur sehr geringen Gewichts Anpassung (da Δw gegen 0 wandert) und einer damit verbundenen langen Rechenzeit gegenüber Aktivierungsfunktionen, die diese Eigenschaft nicht besitzen. Dieses Verhalten ist in der Literatur als „Vanishing-Gradient“ Problem bekannt [40]. Eine Aktivierungsfunktion, die dieses Verhalten nicht aufweist und heutzutage daher häufiger in tiefen neuronalen Netzwerken zu finden ist, ist beispielsweise die ReLu- Funktion (Rectified Linear Unit), welche in Abbildung 14 dargestellt ist.

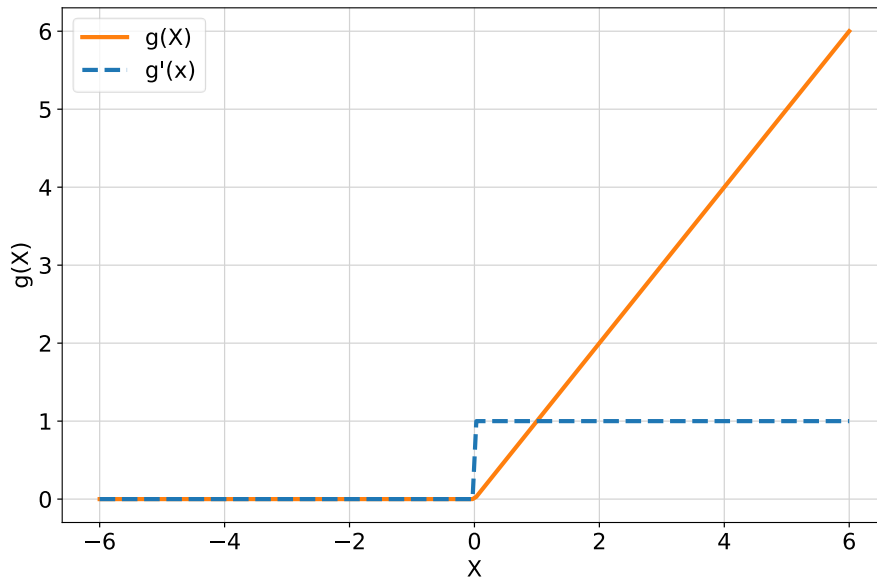


Abbildung 14: Graphische Darstellung der ReLu-Funktion $g(x)$ und ihrer Ableitung $g'(x)$ [40].

Die ReLu Funktion gibt für negative Inputwerte ($X \leq 0$) stets den Wert 0 zurück, während positive Inputwerte ($X > 0$) unverändert zurückgegeben werden. Die Steigung ist damit eine Stufenfunktion mit den Werten 0 und 1, wodurch das Netzwerk rechnerisch sehr effizient wird und eine Gewichts Anpassung auch bei sehr großen Inputwerten möglich ist, da die Steigung nicht gegen 0 geht. Allerdings tritt bei negativen Werten ungeachtet des Inputs keine Änderung der Gewichte auf, was bedeutet, dass die entsprechenden Neuronen ihren Output beim Training nicht anpassen können. Dieses Phänomen wird als „Dead-Neuron“-Problem bezeichnet. Während einzelne inaktive Neuronen noch keine signifikante Auswirkung auf das Netzwerk haben, führt ein ausgeprägtes Vorhandensein zu dem Problem, dass das Netzwerk seine Fähigkeit zu lernen verliert und dadurch unbrauchbar wird [41, 42].

Das Lernen innerhalb eines Neuronalen Netzwerks erfolgt allgemein durch eine iterative Anpassung der Gewichte, welche die Neuronen verknüpfen, um den Gesamtfehler E zu minimieren [27]. Neuronen, deren Output zu einer besseren Prognose für ein Trainingsbeispiel führen, werden verstärkt, Neuronen, die das Gegenteil bewirken, werden abgeschwächt. Das dabei verwendete Lernverfahren wird als Backpropagation bezeichnet, welches auf dem Gradientenabstiegsverfahren beruht. Diese beiden Verfahren sind von elementarer Bedeutung für Neuronale Netzwerke und werden daher im Folgenden etwas genauer beleuchtet.

Gradientenabstieg

Der Gradientenabstieg ist eine iterative Methode, eine zuvor definierte Kostenfunktion zu minimieren und stellt damit einen der essenziellsten Bausteine eines Neuronalen Netzwerks dar. Voraussetzung ist, dass die partielle Ableitung der Funktion gebildet werden kann, welche als Gradient bezeichnet wird [43]. Wurde der Gradient bzw. die Steigung bestimmt, kann ein

kleiner Schritt in Richtung des Minimums der Kostenfunktion getätigt werden, wodurch sich der Fehler verkleinert. Ein ausführliches Beispiel für eine händische Gradientenabstiegsberechnung anhand eines einzelnen Neurons findet sich in Anhang A).

Lernen im Mehrschichtigen Neuronalen Netz

In Neuronalen Netzen kommt jedoch eine Vielzahl von Neuronen vor, welche über eine noch größere Zahl an Gewichten miteinander verknüpft sind. Die Vorgehensweise bei mehrschichtigen Neuronalen Netzen ist jedoch ähnlich zu der in Anhang A) beschriebenen, mit zwei kleinen Unterschieden [27]:

- Der Gesamtfehler $E(w)$ ist nun abhängig von mehreren Gewichten
- Es wird ein Anteil δ_j eingefügt, welcher für jedes Gewicht w_{ij} berechnet werden muss

Der Gesamtfehler $E(w)$ ergibt sich, wie in Gleichung (3) dargestellt, zu [27]:

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

Wobei y_i den realen Wert der Zielgröße und \hat{y}_i den vom Neuronalen Netz berechneten Wert für die Zielgröße beschreibt. Dabei ergibt sich der berechnete Zielwert bei Input x_i , Gewicht w_i und Aktivierungsfunktion A zu:

$$\hat{y}_i = A(w_i \cdot x_i) \quad (4)$$

Die Änderung jedes Gewichts ergibt sich dabei zu [27]:

$$\Delta w_i = -\eta \cdot \delta_j \cdot o_i \quad (5)$$

Mit Lernrate η , Gewichtsanteil δ_j und Output des Neurons o_i . Dabei berechnet sich δ_j (unter Annahme einer Sigmoiden Aktivierungsfunktion) zu [27]:

$$\delta_j = \begin{cases} o_j \cdot (1 - o_j) \cdot (y_i - \hat{y}_i) & \text{falls } j \text{ Output - Neuron} \\ o_j \cdot (1 - o_j) \cdot \sum_k \delta_j \cdot w_{jk} & \text{falls } j \text{ Hidden - Neuron} \end{cases} \quad (6)$$

Ein ausführliches händisches Berechnungsbeispiel für einen vollständigen Vorwärts- und Rückwärtsdurchgang (Backpropagation) durch ein Neuronales Netzwerk, findet sich in Anhang B). Dabei findet im Rückwärtsdurchgang durch die iterative Anpassung der Gewichte der eigentliche Lernprozess statt.

2.3.4 LASSO-Regression

Die LASSO-Regression (von *engl.*: Least absolute shrinkage and selection operator regression) ist ein lineares Modell, das aufgrund seiner Simplizität schnell und einfach zu verwenden ist. Dabei handelt es sich um eine Variante von linearer Regression mit einem

zusätzlichen Term, welcher eine künstliche Verzerrung einführt, um robuster gegen Overfitting zu sein [44].

Bei der gewöhnlichen multiplen linearen Regression wird ein Satz von Prädiktorvariablen x_i und ihre Gewichte β_i verwendet, um eine Zielvariable Y abzuschätzen, wie in Gleichung (7) dargestellt [45]. Dabei stellt ε den Fehlerterm dar.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots + \beta_n x_n + \varepsilon \quad (7)$$

Die Werte für β_i werden dabei nach der Methode der kleinsten Quadrate ausgewählt, sodass der Fehlerterm $\sum (Y_i - \hat{Y}_i)^2$ minimiert wird, wobei \hat{Y}_i den vom Modell vorhergesagten Wert für die Zielvariable entspricht. Im Gegensatz dazu wird bei der LASSO-Regression versucht, die Summe der quadratischen Abweichungen zuzüglich eines Zusatzterms (auch: L1-Term) zu minimieren, wie in Gleichung (8) dargestellt [46].

$$\sum_{i=1}^n \left(y_i - \sum_j x_{ij} \beta_i \right)^2 + \lambda \sum_{j=1}^p |\beta_i| \quad (8)$$

Der Einfluss des Zusatzterms, und damit die Stärke der künstlichen Verzerrung, kann über den Parameter λ eingestellt werden, wodurch dieser einen Hyperparameter des Algorithmus darstellt. Der Vorteil gegenüber linearer Regression liegt in der Robustheit gegenüber Varianz innerhalb der Daten und der daraus folgenden besseren Eignung für kleine, verrauschte Datensätze. Nachteilig ist die Tatsache, dass ein zusätzlicher Parameter in das Modell eingefügt wird, welcher erst auf das jeweilige Problem angepasst werden muss [47, 48].

2.4 Bestimmung der Modell-Genauigkeit

Zur Bestimmung der Genauigkeit von Machine-Learning Modellen gibt es viele verschiedene Kennzahlen. Bei Regressionsproblemen werden meist der mittlere absolute Fehler *MAE* (*engl.*: Mean Absolute Error), der mittlere quadratische Fehler *MSE* (*engl.*: Mean Squared Error) oder das Bestimmtheitsmaß R^2 herangezogen [49]. Letzteres wird in dieser Arbeit verwendet, unabhängig des ausgewählten Modell-Algorithmus.

Das Bestimmtheitsmaß kann als Anteil der erklärten Streuung S_e an der gesamten Streuung S_{ges} verstanden werden. Dabei bezeichnet die erklärte Streuung diejenige Streuung, welche durch das Modell vorhergesagt werden kann. Die Residuenstreuung S_R , auch nicht-erklärte Streuung genannt, beschreibt hingegen die Streuung, welche auch durch das Modell nicht erklärt werden kann. Die Gesamtstreuung ist die Summe aus erklärter und Residuen-Streuung.

$$S_{ges} = S_e + S_R$$

$$\sum_{i=1}^m (\bar{y} - y_i)^2 = \sum_{i=1}^m (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^m (\hat{y}_i - y_i)^2 \quad (9)$$

Dabei beschreibt y_i den Messwert des i -ten Datenpunkts, \hat{y}_i den durch das Modell geschätzten Wert für den i -ten Datenpunkt und \bar{y} den Mittelwert aus allen Messwerten. Das Bestimmtheitsmaß berechnet sich nach folgender Formel [49]:

$$R^2 = \frac{S_e^2}{S_{ges}^2} = 1 - \frac{S_R^2}{S_{ges}^2} = 1 - \frac{\sum_{i=1}^m (\hat{y}_i - y_i)^2}{\sum_{i=1}^m (\bar{y} - y_i)^2} \quad (10)$$

Zum besseren Verständnis werden erklärte Streuung, Residuenstreuung und Gesamtstreuung in Abbildung 15 anhand eines einzelnen Datenpunkts (X_i/Y_i) graphisch dargestellt.

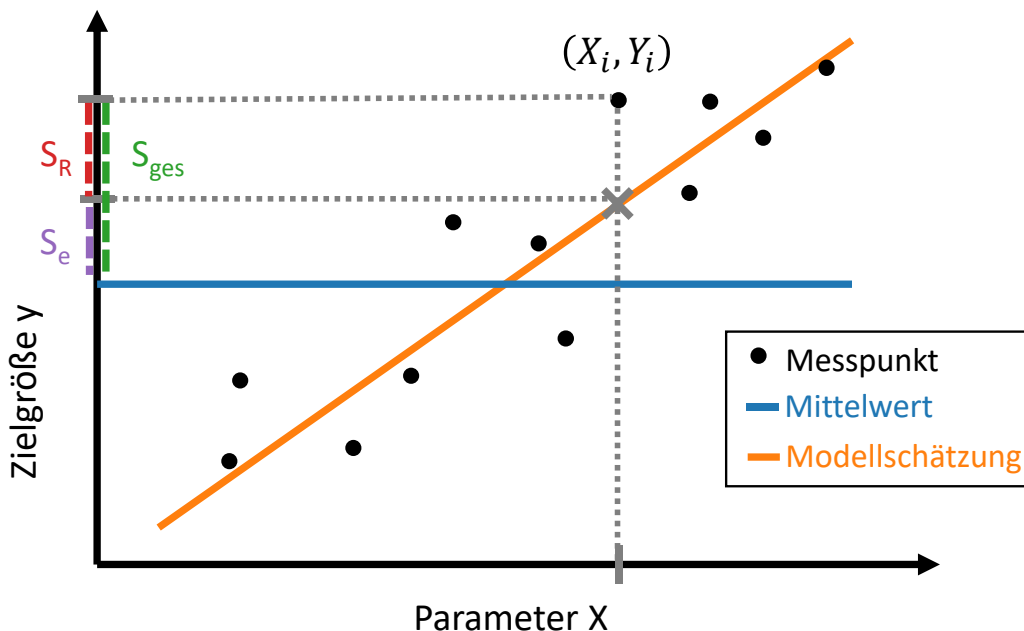


Abbildung 15: Schematische Darstellung von Gesamtstreuung S_{ges} , erklärter Streuung S_e und nicht-erklärter Streuung S_R anhand eines einzelnen Datenpunkts. Berechnet wird die Streuung immer durch Aufsummieren der quadratischen Abweichungen aller Datenpunkte.

Das Bestimmtheitsmaß bewegt sich dadurch im Wertebereich zwischen $-\infty$ und 1, wobei die Extremwerte in der Praxis kaum erreicht werden. Ein Wert von $R^2 = 0$ würde bedeuten, dass das Modell genauso gut ist wie ein Modell, das immer den Mittelwert vorhersagt (Gerade mit Steigung 0), ein Bestimmtheitsmaß von $R^2 = 1$ würde bedeuten, dass alle Messpunkte auf der Regressionsgeraden liegen (und eine Residuenstreuung von $S_R = 0$ vorliegt). Werte $R^2 < 0$ treten auf, wenn das Modell schlechtere Vorhersagen als das Mittelwerts-Modell macht. Tabelle 3 bis Tabelle 5 zeigen Beispiele für verschiedene R^2 Werte.

Tabelle 3: Beispielhafte Berechnung für $R^2 = 1$.

Wahrer Wert y_i	Vorhersage \hat{y}_i	$y_i - \hat{y}_i$	S_R^2 $= (y_i - \hat{y}_i)^2$	$y_i - \bar{y}$	S_{ges}^2 $= (y_i - \bar{y})^2$
10	10	0	0	-10	100
20	20	0	0	0	0
30	30	0	0	10	100
Mittelwert = 20			$\Sigma = 0$		$\Sigma = 200$

$$R^2 = 1 - \frac{S_R^2}{S_{ges}^2} = 1 - \frac{0}{200} = 1$$

Tabelle 4: Beispielhafte Berechnung für $R^2 = 0$.

Wahrer Wert y_i	Vorhersage \hat{y}_i	$y_i - \hat{y}_i$	S_R^2 $= (y_i - \hat{y}_i)^2$	$y_i - \bar{y}$	S_{ges}^2 $= (y_i - \bar{y})^2$
10	20	-10	100	-10	100
20	20	0	0	0	0
30	20	10	100	10	100
Mittelwert = 20			$\Sigma = 200$		$\Sigma = 200$

$$R^2 = 1 - \frac{S_R^2}{S_{ges}^2} = 1 - \frac{200}{200} = 0$$

Tabelle 5: Beispielhafte Berechnung für $R^2 < 0$.

Wahrer Wert y_i	Vorhersage \hat{y}_i	$y_i - \hat{y}_i$	S_R^2 $= (y_i - \hat{y}_i)^2$	$y_i - \bar{y}$	S_{ges}^2 $= (y_i - \bar{y})^2$
10	30	-20	400	-10	100
20	30	-10	100	0	0
30	10	20	400	10	100
Mittelwert = 20			$\Sigma = 900$		$\Sigma = 200$

$$R^2 = 1 - \frac{S_R^2}{S_{ges}^2} = 1 - \frac{900}{200} = -3,5$$

2.5 Preprocessing

Bisher wurde stillschweigend vorausgesetzt, dass für das Training von KI Modellen ein ausreichend vorbereiteter Datensatz vorliegt. Da dies in der Praxis zu Beginn beinahe nie der Fall ist, müssen meist mehrere Vorverarbeitungsschritte erfolgen, um einen Datensatz für ein Machine-Learning Modell zugänglich zu machen. Dies wird in der Literatur mit dem englischen Begriff „Preprocessing“ betitelt. Preprocessing umfasst beispielsweise das Umwandeln von Schriftsprache in Zahlen, das Normieren von Wertebereichen, mathematische Transformationen, den Umgang mit fehlenden Werten und viele weitere Operationen, die dem eigentlichen Trainingsprozess vorausgehen. Preprocessing hat daher einen sehr großen Einfluss

auf das Ergebnis eines Machine-Learning Modells [50]. Die für diese Arbeit wichtigsten Preprocessing-Methoden werden nun in den folgenden Unterkapiteln genauer erläutert.

2.5.1 Normalverteilung

Für viele statistische Verfahren, wie etwa Hypothesentests und Gauß-Tests, wird eine Normalverteilung der Daten vorausgesetzt [51]. Auch um die Güte mancher Machine-Learning Modelle, beispielsweise lineare oder logistische Regression, bewerten zu können, müssen deren Fehlerterme normalverteilt sein [52]. Dafür ist es notwendig zu überprüfen, ob diese Annahme für die eigenen Daten zutrifft [53]. Dies kann mithilfe analytischer oder graphischer Methoden geschehen. Zu den häufig verwendeten analytischen Methoden zählen beispielsweise [54, 55]:

- Kolmogorov-Smirnov Test
- Shapiro-Wilk Test
- Anderson-Darling Test

In dieser Arbeit wird der Shapiro-Wilk Test verwendet, da dieser nach Razali et al. [51] der mächtigste der drei Tests ist. Allen oben genannten Tests liegt dieselbe Nullhypothese zugrunde, dass die Daten normalverteilt sind. Die Gegenhypothese ist damit, dass die Daten nicht normalverteilt sind. Wie in der Wissenschaft üblich, wird ein Vertrauensniveau von mindestens 95%, bzw. eine Irrtumswahrscheinlichkeit von höchstens 5% angesetzt. Damit ergeben sich zwei mögliche Test-Ergebnisse für einen so definierten Normalverteilungstest:

1. Errechnet der Test einen p-Wert von weniger als 0,05, so ist die Nullhypothese zu verwerfen und es wird gefolgert, dass die untersuchten Daten nicht normalverteilt sind.
2. Errechnet der Test einen p-Wert größer als 0,05, wird die Nullhypothese angenommen und davon ausgegangen, dass die Daten normalverteilt sind.

Der zweite Schluss ist streng genommen nicht vollständig korrekt, da ein Wert $p > 0,05$ nur aussagt, dass die Nullhypothese nicht ausgeschlossen werden kann, was nicht zwangsläufig bedeutet, dass sie nachgewiesen wurde. In der Praxis wird dieser Schluss jedoch häufig trotzdem angewendet und auch in dieser Arbeit wird bei Werten von $p > 0,05$ von einer Normalverteilung ausgegangen. Dem Shapiro-Wilk Test liegt dabei folgende Formel zugrunde [51]:

$$p = \frac{(\sum_{i=1}^n a_i y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \text{ mit } a_i = (a_1, \dots, a_n) = \frac{m^T V^{-1}}{\sqrt{m^T V^{-1} V^{-1} m}} \quad (11)$$

Wobei y_i die geordneten Werte der Stichprobe, \bar{y} den Mittelwert der Stichprobe und a_i einen nachzuschlagenden Tabellenwert darstellen [56].

Die analytischen Methoden haben allerdings einen entscheidenden Nachteil: sie sind abhängig von der Stichprobengröße und tendieren dazu, größere p-Werte für kleinere Stichprobenumfänge zu berechnen. Daher gibt es neben den analytischen Methoden auch graphische Methoden zur Bestimmung der Normalverteilung, von denen zwei häufig verwendete aufgeführt sind [57]:

- Histogramm
- Quantil-Quantil Graph

In dieser Arbeit wird auf das Histogramm zur graphischen Bestimmung zurückgegriffen, um den analytischen Shapiro-Wilk Test zu ergänzen. Kann die geforderte Voraussetzung nach Normalverteilung nicht erfüllt werden, bieten nicht-lineare Transformationen, wie die Box-Cox-Transformation, eine Möglichkeit, diese künstlich zu erzeugen [52]. Auf lineare und nicht-lineare Transformationen wird näher in Kapitel 2.5.5 eingegangen.

2.5.2 Multikollinearität

Viele ML-Modelle setzen voraus, dass alle Eingangsparameter unabhängig voneinander sind, also keine starken Korrelationen untereinander aufweisen. Das Gegenteil, nämlich das Vorhandensein von starken Korrelationen innerhalb der Eingangsparameter, wird als Multikollinearität bezeichnet. Je nach verwendetem Algorithmus kann Multikollinearität zu unterschiedlich starken Problemen führen. Allgemein lässt sich sagen, dass bei starken Korrelationen zwischen zwei oder mehreren Eingangsparametern redundante Informationen im Datensatz existieren, was eine Isolation der Beziehung zwischen Eingangs- und Zielgröße verhindert. Letzteres stellt jedoch oft eine Voraussetzung dar, beispielsweise bei der linearen Regression. Der daraus resultierende Nachteil ist eine hohe Varianz bei der Schätzung der Koeffizienten, was ein weniger robustes Modell zur Folge hat. Kleine Änderungen im Datensatz können dann zu drastisch unterschiedlicher Einschätzung der Wichtigkeit der Eingangsparameter führen. Das Schätzen der Wichtigkeiten ist jedoch ein essenzieller Bestandteil des Preprocessing, wie später in Kapitel 2.6.5 genauer beschrieben wird. Eine wirkungsvolle Gegenmaßnahme besteht hier in der Entfernung redundanter Parameter aus dem Datensatz. Um redundante Parameter detektieren zu können, existieren unterschiedliche statistische Methoden, von denen in dieser Arbeit mit dem Pearson-Korrelationskoeffizient und dem Variance Inflation Factor (VIF) die beiden häufigsten erläutert werden [58].

Der Pearson-Korrelationskoeffizient r beschreibt einen Wert von -1 bis 1, wobei die Extremwerte für starke indirekt proportionale, bzw. proportionale Korrelation sprechen. Ein Wert von 0 entspricht hingegen dem Ausbleiben von Korrelation. Anzumerken ist weiterhin,

dass die Pearson-Korrelation grundsätzlich nur lineare Zusammenhänge erkennen kann. Berechnet werden die Pearson-Koeffizienten nach Gleichung (12).

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} \quad (12)$$

Wobei \bar{X} und \bar{Y} die Mittelwerte der Variablen X und Y darstellen und X_i und Y_i die Werte der einzelnen Datenpunkte. Im Zähler werden die Werte damit zunächst durch Abzug der Mittelwerte zentriert und die Summe der Kreuzprodukte gebildet, während der Nenner die Größenordnung auf den Wertebereich -1 bis 1 skaliert. Demnach kann der Korrelationskoeffizient als die standardisierte, zentrierte Summe der Kreuzprodukte zweier Variablen verstanden werden. Wurden die Pearson-Korrelationskoeffizienten für alle Eingangsparameter untereinander berechnet, kann zur Visualisierung eine sog. „Heatmap“ erzeugt werden, welche jeden Parameter über allen anderen Parametern aufträgt und kleine bzw. große Werte unterschiedlich einfärbt (siehe Abbildung 16). Anschließend muss eine Grenze definiert werden, ab der Parameter als redundant gelten und aus dem Datensatz ausgeschlossen werden. In der Praxis wird hier meist ein Pearson-Korrelationskoeffizient von $|r| > 0,8$ oder $|r| > 0,9$ herangezogen [59–61].

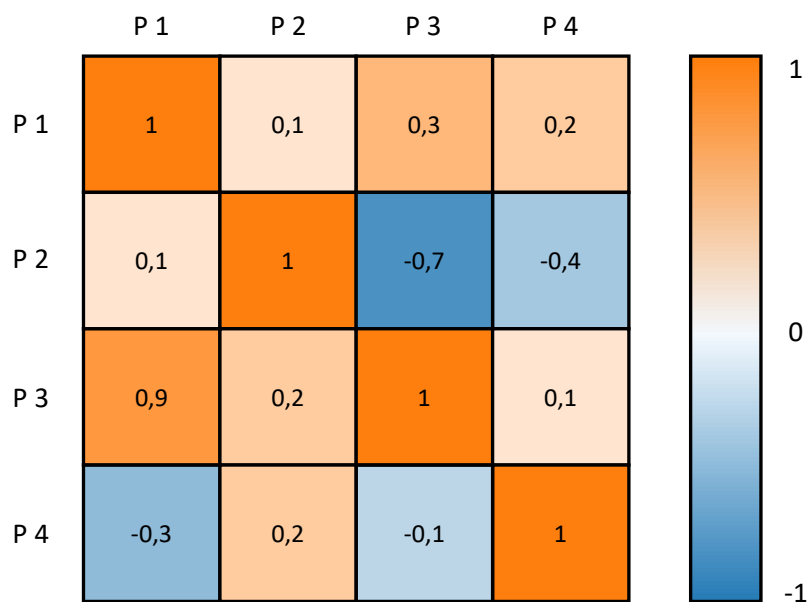


Abbildung 16: Beispielskizze einer Heatmap mit Pearson-Korrelationskoeffizienten von vier Herstellungsparametern P1-P4 inklusive Farblegende. Werte nahe 1 oder -1 stellen starke direkte bzw. indirekte Korrelationen dar (tiefblaue oder -orangefarbene Farbe); die Hauptdiagonale hat immer den Wert 1.

Eine weitere Möglichkeit, die Eingangsparameter auf Multikollinearität zu überprüfen, ist die Berechnung des Variance Inflation Factors. Dieser misst, wie stark Multikollinearität die Varianz einer Regressionsanalyse erhöht, wobei höhere Werte als problematischer zu betrachten sind. Für ein lineares Regressionsmodell wie in Gleichung (13) mit den Parametern

$(X_1, X_2, X_3, \dots, X_k)$, kann der VIF_k über das Bestimmtheitsmaß des k -ten Parameters R_k^2 nach Gleichung (14) berechnet werden.

$$Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \dots + \alpha_k X_k + e \quad (13)$$

$$VIF_k = \frac{1}{1 - R_k^2} \quad (14)$$

Als Grenzwert für redundante Parameter wird in der Praxis meist ein Wert von $VIF = 10$ verwendet, oberhalb dessen ein Parameter aus dem Datensatz entfernt wird [62].

2.5.3 Umgang mit fehlenden Daten

Ein besonders häufig auftretender Fall während des Preprocessing, ist das Vorhandensein von fehlenden Werten im Datensatz. Generell lässt sich sagen, dass Machine-Learning Modelle nicht mit fehlenden Werten rechnen können, sodass diese entweder durch einen Schätzwert ersetzt, oder die entsprechenden Zeilen komplett aus dem Datensatz entfernt werden müssen. Allerdings können auch Muster und Systematiken in den fehlenden Werten wichtige Informationen enthalten, weswegen es wichtig ist, die Ursache des Fehlens zu identifizieren. Beispielsweise könnte es sein, dass bestimmte Messungen besonders aufwendig oder teuer sind, und daher nicht bei jedem Datenpunkt durchgeführt werden können. Auch Fehlfunktionen am Messgerät selbst können zu unbrauchbaren Daten führen, die später entfernt werden müssen. Und auch der Mensch ist eine zuverlässige Fehlerquelle. In der Literatur wird daher zwischen drei verschiedenen Fehl-Mechanismen unterschieden [63–65]:

- Missing completely at random (MCAR)
- Missing at random (MAR)
- Missing not at random (MNAR)

Wie der Name schon andeutet, beschreibt MCAR den Fall, dass keine Systematik hinter den fehlenden Werten steckt, sodass ein fehlender Wert für jeden (beobachteten und nicht beobachteten) Datenpunkt gleichermaßen wahrscheinlich ist. Ein Beispiel hierfür ist eine tägliche Gewichtsmessung mit einer Waage, bei der sich nach einer unbekanntem Zeit die Batterien entladen. Hier fehlt der Messwert eines Tages schlicht aufgrund von Zufall, der nichts mit der Messung zu tun hat. Allerdings trifft dieser Fall in der Praxis oft nicht zu.

Häufiger tritt der Fall MAR auf: hier ist die Fehlstellenwahrscheinlichkeit nur innerhalb der beobachteten Datenpunkte gleich. Ein Beispiel dafür wäre eine Waage, die auf einer weichen Oberfläche platziert wird, und daher mehr fehlende Werte produziert, als eine Waage die auf einer harten Oberfläche platziert wird. Auch hier ist die Wahrscheinlichkeit für ein Fehlen für jeden Messwert gleich groß, allerdings hat die Art der Messung hier Einfluss auf die

Fehlerwahrscheinlichkeit. Daher ist MAR in der Praxis deutlich häufiger als MCAR und wird oft als Grundlage für Modelle verwendet. MAR liegt auch in dieser Arbeit vor, da manche Messungen aufgrund ihres hohen Aufwands generell weniger häufig vorgenommen wurden, als andere.

Der letzte Fall (MNAR) liegt vor, wenn die Fehlstellenwahrscheinlichkeit für jeden Datenpunkt unterschiedlich ist und die Gründe dafür unbekannt sind. Als Beispiel kann wieder die bereits genannte Waage herangezogen werden. In diesem Fall könnte sich aber der Mechanismus abnutzen, je länger die Waage in Betrieb ist, ohne dass dies bemerkt würde. Wenn dann auch noch die schwereren Objekte zu einem späteren Zeitpunkt gemessen würden, so würde die Verteilung der Messwerte verzerrt werden. Ein anderes Beispiel für MNAR wäre eine öffentliche Umfrage, bei der diejenigen Mitbürger mit schwächerer Meinung weniger oft teilnehmen und daher unterrepräsentiert sind.

Je nach Fehlstellen-Mechanismus (MCAR, MAR, MNAR) können unterschiedliche Methoden für den Umgang mit fehlenden Werten erlaubt sein oder nicht. Beispielsweise kann die weiter unten beschriebene Entfernung von Datenzeilen mit fehlenden Werten nur verwendet werden, falls MCAR vorliegt. Ist dies nicht der Fall, kann dies zu signifikant verzerrten Korrelationen führen [64].

Um lückenhafte Datensätze für Machine-Learning Modelle zugänglich zu machen, gibt es verschiedene Möglichkeiten:

- Entfernen aller Datenzeilen, die fehlende Daten enthalten
- Schätzen der fehlenden Daten
- Bei fehlenden kategorischen Daten: Einführen einer „Unbekannt“-Kategorie

Da die fehlenden Werte in dieser Arbeit durch verschiedene Schätz-Methoden ermittelt werden, wird auf diesen Punkt im folgenden Unterkapitel genauer eingegangen.

2.5.4 Schätzen fehlender Werte

Das Schätzen fehlender Werte kann entweder auf Grundlage eines einzelnen Parameters („univariat“) oder mehrerer Parameter („multivariat“) geschehen.

Einfache univariate Schätzung

Die einfachste und schnellste Methode, um fehlende Werte abzuschätzen, ist die Bildung eines Mittelwerts pro Spalte und dessen Verwendung als Schätzwert für die unbekannte Größe. Durch die Mittelwert-Schätzung wird allen fehlenden Werten eines Parameters der Mittelwert eben jenes Parameters zugewiesen. Der Nachteil dieser Methode ist, dass mögliche Korrelationen zwischen den verschiedenen Parametern nicht beachtet werden, um die

fehlenden Werte abzuschätzen. Ein fiktives Beispiel hierfür ist in Tabelle 6 gegeben, wo Werte für Massenverluste während der Pyrolyse und während der Silizierung bei der jeweiligen Prozesstemperatur festgehalten wurden. Zwar können die fehlenden Werte der Spalten „ Δm Pyrolyse [%]“ und „ Δm Silizierung [%]“ durch Zahlenwerte ersetzt werden, jedoch wird dabei ignoriert, dass höhere Temperaturen auch tendenziell höhere Massenverluste zur Folge haben.

Tabelle 6: Fiktives Beispiel einer einfachen univariaten Schätzung fehlender Werte.

Parameter 1	Parameter 2	Parameter 3
Temperatur Pyrolyse [°C]	Δm Pyrolyse [%]	Δm Silizierung [%]
1000	17	26
900	15	31
1600	43	89
1650	45	fehlt → Schätzung: 60,25
1450	fehlt → Schätzung: 30,0	95

Multivariate iterative Schätzung

Eine anspruchsvollere Methode für das Schätzen fehlender Werte ist durch iterative Schätzer gegeben. Diese beschreiben jeden Parameter, welcher fehlende Werte enthält, als eine Funktion anderer Parameter und verwenden letztere dann für eine Schätzung [66]. Meist wird dazu ein eigenes KI-Modell angelernt, welches ausschließlich dazu da ist, die fehlenden Werte auf Grundlage der vorhandenen Werte zu schätzen. Dadurch werden Ähnlichkeiten zwischen den Proben besser berücksichtigt, als bei der Mittelwert-Schätzung. Dafür wird wieder das fiktive Beispiel von Tabelle 6 herangezogen. Erkennbar ist, dass mit höherer Pyrolysetemperatur auch eine größere Massenänderung während der Pyrolyse und Silizierung vorherrscht. Diese Information wird beim iterativen Schätzen verwendet.

Für die Startwerte wird, wie beim einfachen Schätzen, der Mittelwert jeder Spalte für die fehlenden Werte herangezogen (siehe Tabelle 6). Anschließend wird allerdings nicht abgebrochen, sondern die fehlenden Werte iterativ aktualisiert. Dazu wird ein statistisches Modell zugrunde gelegt, in diesem Beispiel wird aus Gründen der Einfachheit eine multivariate lineare Regression verwendet, in der Praxis kann aber auch jedes andere Modell zum Einsatz kommen (beispielsweise: Entscheidungsbaum, RandomForest, KNeighbors, BayesianRidge...). Zunächst wird willkürlich Parameter 2 als Zielgröße festgelegt, Parameter 1 und 3 werden also für das Anlernen des Modells verwendet. Dazu wird die letzte Zeile von Tabelle 6 für das Modellfitting weggelassen, da diese den gesuchten Wert enthält. Für alle anderen fehlenden Werte wird der Mittelwert der Spalte angenommen. Daraus ergibt sich folgende Regressionsformel:

$$P2(P1, P3) = -22,36 + 0,040 \cdot P1 + 0,023 \cdot P3 \quad (15)$$

Hierdurch lässt sich ein Schätzwert für den fehlenden Wert aus Parameter 2 berechnen, wie das Einsetzen in Gleichung (15) zeigt:

$$P2(1450, 95) = -22,36 + 0,040 \cdot 1450 + 0,023 \cdot 95 = 37,4 \quad (16)$$

Dieser wird nun in die Tabelle eingesetzt, wodurch sich Tabelle 7 ergibt.

Tabelle 7: Berechnung des fehlenden Werts von Parameter 2 innerhalb der 1. Iteration.

Parameter 1	Parameter 2	Parameter 3
Temperatur Pyrolyse [°C]	Δm Pyrolyse [%]	Δm Silizierung [%]
1000	17	26
900	15	31
1600	43	89
1650	45	60,25
1450	30,0 37,4	95

Anschließend wird analog der fehlende Wert von Parameter 3 berechnet, wozu wieder ein lineares Modell gefittet wird, indem diesmal die vorletzte Zeile aus Tabelle 7 weggelassen wird. Die lineare Regressionsgleichung lautet in diesem Fall:

$$P3(P1, P2) = 152,4 - 0,288 \cdot P1 + 9,40 \cdot P2 = 100,3 \quad (17)$$

Dies resultiert in einem Wert von 100,3. Dieser wird wieder in die Tabelle eingesetzt, wodurch sich Tabelle 8 ergibt:

Tabelle 8: Berechnung des fehlenden Werts von Parameter 3 innerhalb der 1. Iteration; die vorletzte Zeile wird dabei nicht zur Berechnung der Regressionsgleichung herangezogen.

Parameter 1	Parameter 2	Parameter 3
Temperatur Pyrolyse [°C]	Δm Pyrolyse [%]	Δm Silizierung [%]
1000	17	26
900	15	31
1600	43	89
1650	45	60,25 100,3
1450	30,0 37,4	95

Damit ist die erste Iteration abgeschlossen. Nun wird anhand der neuen Werte wieder eine Regressionsgleichung für den fehlenden Wert in Merkmal 2 erstellt, die sich von Gleichung (15) unterscheidet, da ihr andere Wertepaare zugrunde liegen.

Die Veränderung der Werte im Beispiel wurde in Tabelle 9 für die ersten vier Iterationen aufgeführt. Abbildung 17 zeigt denselben Verlauf als Grafik. Hier wird deutlich, dass die Werte sich anfangs noch stark ändern, mit zunehmender Iterationszahl jedoch gegen einen bestimmten

Wert konvergieren. In der Praxis muss deshalb ein Abbruchkriterium gefunden werden, ab dem der aktuelle Schätzwert als endgültiges Ergebnis beibehalten wird.

Tabelle 9: Berechnete Werte der ersten vier Iterationen bei Verwendung von iterativer multivariater Schätzung im oben beschriebenen Beispiel.

Iteration	Fehlender Wert P2	Fehlender Wert P3
1	30,0	60,3
2	37,4 (+24,7%)	100,3 (+66,3%)
3	38,8 (+3,7%)	94,8 (-5,5%)
4	39,6 (+2,1%)	93,3 (-1,6%)

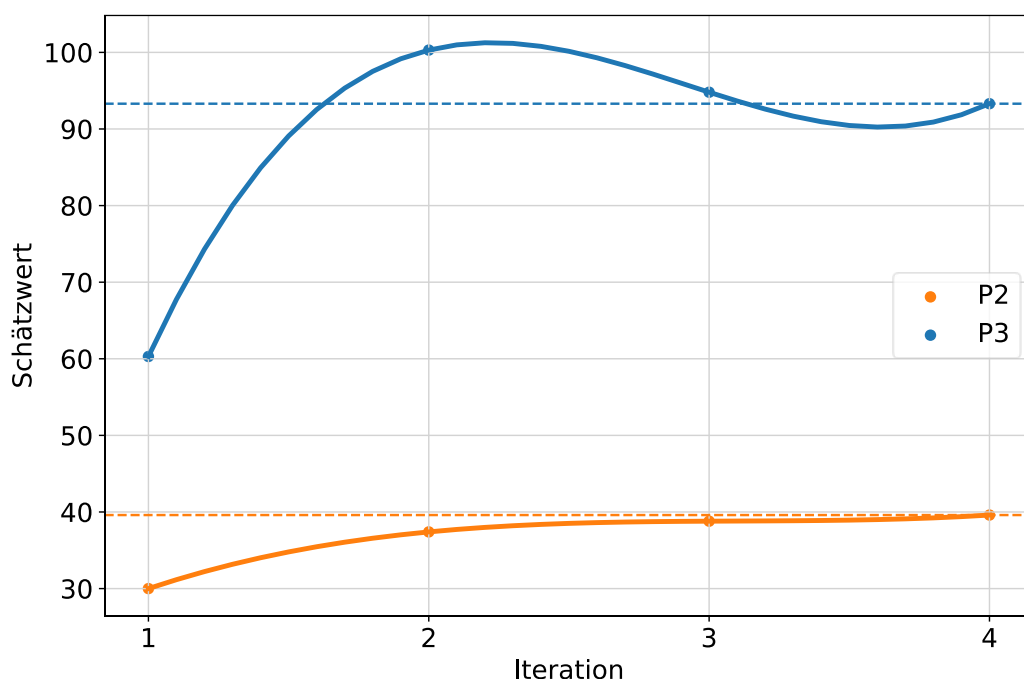


Abbildung 17: Änderung der Schätzwerte des obengenannten Beispiels beim iterativen multivariaten Schätzen; orange: Schätzung für fehlenden Wert in Parameter 2, blau: Schätzung für fehlenden Wert in Parameter 3. Nach einer gewissen Einschwingphase konvergieren die Schätzungen zu einem bestimmten Grenzwert hin (gestrichelte farbige Linien).

Oft wird zusätzlich eine Obergrenze für die Anzahl der zu durchlaufenden Iterationen festgelegt, sodass bei ausbleibender Konvergenz keine Endlosschleife entsteht. Eine weitere Stellgröße, welche einen Einfluss auf die Abschätzung hat, ist die Anzahl der Parameter, welche zur Berechnung der fehlenden Werte herangezogen wird. So könnte es im oberen Beispiel sein, dass der zu schätzende Parameter P2 nur mit Parameter P1 korreliert, nicht aber mit Parameter P3, welcher deshalb nicht in die lineare Regression aufgenommen werden sollte. Weiterhin ist zu beachten, dass auch die Reihenfolge der Spalten, welche für das Schätzen herangezogen wird, einen Einfluss auf das Ergebnis hat. Im obigen Beispiel wurde zunächst Parameter 2 geschätzt und dann Parameter 3, wobei eine vertauschte Reihenfolge jedoch zu einem leicht veränderten Ergebnis führen würde. Hier stehen wieder eine Vielzahl möglicher Strategien zur

Auswahl: beispielsweise könnte von links nach rechts oder rechts nach links vorgegangen werden, oder aber es wird zunächst die Spalte mit den wenigsten fehlenden Werten verwendet und sich dann bis zur Spalte mit den meisten fehlenden Werten durchgearbeitet.

Der Ingenieur muss sich also mindestens mit folgenden festzulegenden Größen beim iterativen Schätzen auseinandersetzen:

- Wahl des zugrundeliegenden statistischen Modells (Lineare Regression, Entscheidungsbaum, RandomForest, BayesianRidge, KNeighbors...)
- Festlegung der Anzahl n der ähnlichsten Parameter, die für die Abschätzung von Parameter X herangezogen werden sollen
- Abbruchkriterium für Konvergenz
- Maximalanzahl Iterationen
- Strategie für das Schätzen des Startwerts festlegen (Mittelwert, Median, häufigster Wert...)
- Strategie für Reihenfolge der Spalten festlegen

2.5.5 Umgang mit Ausreißern

Unter dem Begriff „Ausreißer“ werden Messwerte verstanden, welche signifikant von den anderen Messwerten abweichen. Obgleich Ausreißer den Informationsgehalt eines Datensatzes stark verzerren oder reduzieren können, ist es nicht in allen Fällen sinnvoll, diese zu entfernen. Beruht der abweichende Wert nicht auf einem Messfehler, kann es sogar sein, dass er essenziell wichtig für die Interpretation der Daten ist. Daher ist im Umgang mit Ausreißern generell eine eingehende Prüfung und ein gutes Verständnis des zu untersuchenden Datensatzes von Nöten. Für die Detektion von Ausreißern existieren univariate und multivariate Methoden, von denen in dieser Arbeit lediglich die Ersteren verwendet werden, da dies bereits zu ausreichend guten Ergebnissen führte. Unter die univariaten Methoden fallen vor allem solche, die konventionell bei der Daten-Säuberung zum Einsatz kommen und dazu verwendet werden, fehlerhafte Messwerte zu eliminieren. Häufig wird hierzu die Z-Statistik verwendet, welche einer Standardisierung der Daten entspricht, und nach Gleichung (18) berechnet wird [67, 68].

$$|z_i| = \left| \frac{X_i - \mu}{\sigma} \right| \quad (18)$$

Dabei beschreibt X_i den jeweiligen Messwert, μ den Mittelwert und σ die Standardabweichung. Der Wert z_i beschreibt demnach, wie viele Standardabweichungen vom Mittelwert der jeweilige Messwert entfernt ist. Nach Celi et al. [69] dient ein Richtwert von $|z_i| \geq 3$ als guter Schwellwert, um Ausreißer zu detektieren. Alle Datenpunkte mit einem

$|z_i| < 3$ verbleiben demnach im Datensatz, alle anderen werden als Ausreißer deklariert und entfernt, da diese mehr als 3 Standardabweichungen vom Mittelwert ihrer Verteilung entfernt sind.

2.5.6 Transformationen

Transformationen bezeichnen Methoden, um die Wertebereiche des Datensatzes in einer bestimmten vorgegebenen Weise zu manipulieren, meistens mit dem Ziel einer Leistungs- oder Effizienzerhöhung des statistischen Modells. Generell wird zwischen linearen und nicht-linearen Transformationen unterschieden. Zu den linearen Transformationen gehören beispielsweise die Normalisierung oder Skalierung, welche bei Neuronalen Netzen praktisch immer zum Einsatz kommen. Hierbei werden die zugrundeliegenden Werte lediglich skaliert, wobei die Form der Verteilung gewahrt wird. Nicht-lineare Transformationen verändern stattdessen auch die Form der Verteilung, weswegen sie beispielsweise eingesetzt werden, um Datensätze „normalverteilter“ zu machen. Im Folgenden wird auf einige für diese Arbeit relevante Transformationen eingegangen [70].

Normalisierung

Daten-Normalisierung ist eine lineare Transformation und wird verwendet, um alle Merkmale eines Datensatzes in eine ähnliche Größenordnung zu bringen [71]. Anderenfalls würden Merkmale mit großen Werten (beispielsweise in der Größenordnung 1000) von manchen ML-Modellen als wichtiger eingestuft werden, als solche mit kleinen Werten (beispielsweise in der Größenordnung 10). Dies gilt besonders für Modelle, die eine gewichtete Summe für die Gradientenberechnung verwenden, wie bei den künstlichen neuronalen Netzen. Bei der Normalisierung wird der Wertebereich aller unabhängigen Variablen auf den Wertebereich $[0, 1]$ skaliert, wobei Gleichung (19) verwendet wird.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (19)$$

Standardisierung

Standardisierung ist eine weitere Möglichkeit der linearen Transformation von Werten. In diesem Fall wird der Datensatz so transformiert, dass er die Eigenschaften einer Standardnormalverteilung, also einen Mittelwert von 0 und eine Standardabweichung von 1 besitzt. Dabei wird nach Gleichung (20) vorgegangen. Die bereits zuvor besprochene Z-Statistik ist ein Beispiel einer Standardisierung.

$$X_{standard} = \frac{X - \mu}{\sigma} \quad (20)$$

Box-Cox-Transformation

Die Box-Cox-Transformation geht auf die Mathematiker George Box und David Cox zurück und zählt zu den nicht-linearen Transformationen. Ihr Ziel liegt darin, die Verteilung der Daten näher an eine Normalverteilung zu bringen, welche für viele statistische Untersuchungen vorausgesetzt wird [72, 52].

Die zugrundeliegende Rechenvorschrift ist in Gleichung (21) gegeben und gilt nur für positive Werte [73]:

$$Y^{(\lambda)} = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & (\lambda \neq 0) \\ \log(Y) & (\lambda = 0) \end{cases} \quad (21)$$

Dabei muss der optimale Wert für λ für jedes Problem geschätzt oder iterativ ermittelt werden [72].

2.5.7 Umgang mit kategorischen Variablen

Bei kategorischen Variablen handelt es sich um nicht-numerische Werte des Datensatzes, welche nicht direkt vom Algorithmus verwertet werden können. Darunter fallen beispielsweise Zeichenketten wie „MF43“, „HTA-Fasern“ oder „Vorbehandelt“. Um die Einflüsse dieser Parameter trotzdem in das Modell integrieren zu können, müssen kategorische Variablen durch Zahlenwerte verschlüsselt werden. Dafür gibt es eine Vielzahl an Möglichkeiten, welche ausführlich in Potdar et al. [74] beschrieben sind. In dieser Arbeit werden davon mit One Hot Encoding und Ordinal Encoding die in der Praxis am häufigsten vertretenen verwendet und im Folgenden erläutert.

Ordinal Encoding

Ordinal Encoding ist vermutlich die intuitivste Art, Zeichenketten in Zahlenwerte zu übersetzen. Dafür wird jeder Kategorie der Variable eine Ganzzahl zugewiesen, wie Tabelle 10 beispielhaft zeigt.

Tabelle 10: Fiktives Beispiel für Ordinal Encoding.

	Verwendetes Harz	Verwendetes Harz verschlüsselt
Probe 1	MF43	1
Probe 2	XP60	2
Probe 3	MF88	3

Diese relativ simple Verschlüsselungsmethode fügt dem Datensatz keine zusätzlichen Spalten zu, allerdings wird dadurch eine Reihenfolge zwischen den Kategorien impliziert, welche nicht unbedingt vorliegen muss. Ist in Wahrheit keine Reihenfolge vorhanden (wie im

Beispiel aus Tabelle 10), würde der Algorithmus fälschlicherweise annehmen, dass das Harz MF88 aufgrund seines höheren Verschlüsselungswerts von 3 einen größeren Wert besitzt als MF43 mit einem Verschlüsselungswert von 1, obwohl zwischen diesen Kategorien kein Zusammenhang besteht [74].

One Hot Encoding

One Hot Encoding ist die vermutlich am häufigsten verwendete Verschlüsselungsart für kategorische Variablen die keine Rangfolge besitzen. Hierbei wird eine Variable mit n Beobachtungen und k unterschiedlichen Werten in k binäre Variablen mit jeweils n Beobachtungen verschlüsselt. Ein Beispiel ist in Tabelle 11 gegeben. Dabei werden stets nur die Werte 0 (kein Vorhandensein) oder 1 (Vorhandensein) vergeben, wodurch keine Reihenfolge zwischen den unterschiedlichen Werten impliziert wird. Der Datensatz erhält jedoch zusätzliche Spalten, was besonders bei vielen Kategorien innerhalb der Variablen deutlich wird [74].

Tabelle 11: Fiktives Beispiel für One Hot Encoding.

	Verwendetes Harz	MF43	XP60	MF88
Probe 1	MF43	1	0	0
Probe 2	XP60	0	1	0
Probe 3	MF88	0	0	1

2.5.8 Normierte Entropie

Die normierte Entropie (im Folgenden auch abkürzend als ‚Entropie‘ bezeichnet) ist ein Streuungsmaß der Statistik und wird in dieser Arbeit verwendet, um die Variabilität innerhalb der Herstellungsparametern zu bewerten. Dabei ergibt die normierte Entropie immer einen Wert zwischen 0 und 1, wobei größere Werte für eine gleichmäßige Verteilung der Daten stehen und kleinere Werte für eine einseitige Verteilung. Dies ist schematisch in Abbildung 18 dargestellt, wobei für dieses Beispiel eine Variable mit zwei Klassen (blaue Punkte und orangefarbene Dreiecke) erfunden wurde. Ist eine der beiden Klassen deutlich überrepräsentiert, liegt eine niedrige Entropie vor, sind beide Klassen komplett ausgewogen, erreicht die Entropie den maximalen Wert von 1.

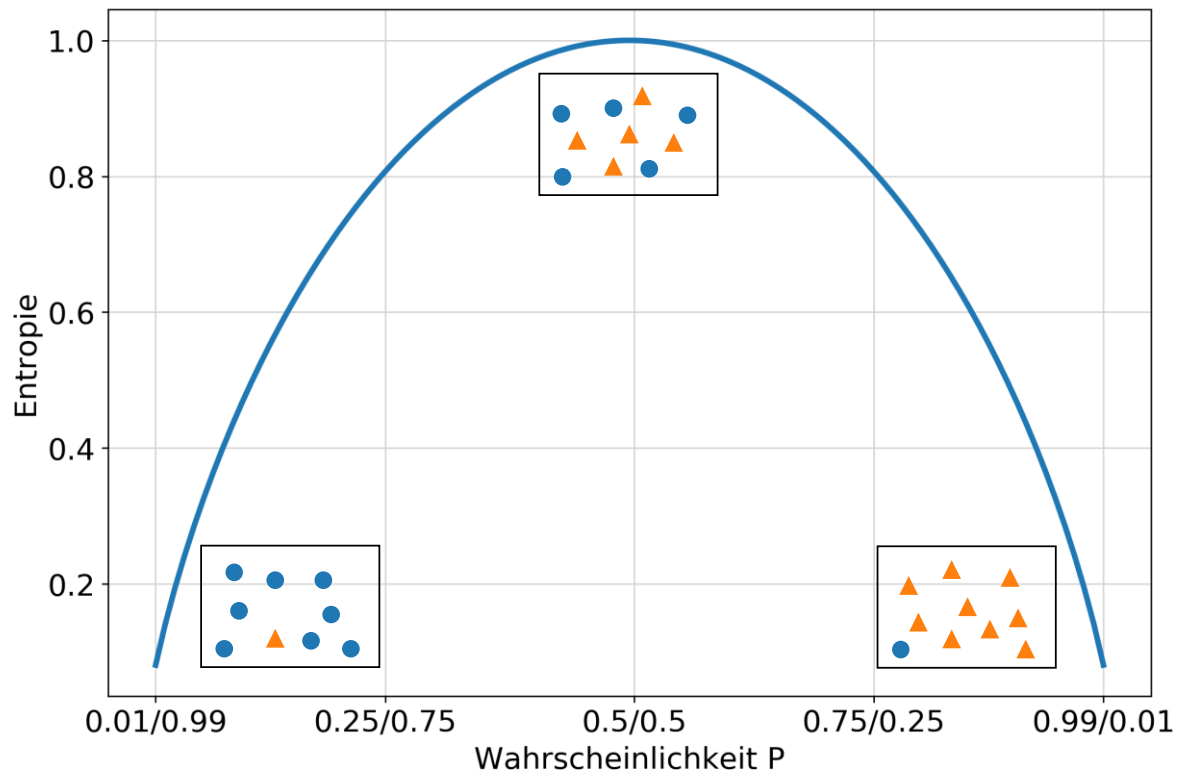


Abbildung 18: Beispielhafte Darstellung der Entropie eines Systems mit 2 Klassen (blaue Punkte und orangefarbene Dreiecke); je ungleichmäßiger die Verteilung, desto niedriger die Entropie und andersherum.

Die normierte Entropie wird nach Gleichung (22) berechnet [75]. Dabei ist n die Anzahl der unterschiedlichen Klassen und P_i die Wahrscheinlichkeit von Klasse i . Der Term $\sum_{i=1}^n P_i \log_2(P_i)$ steht dabei für die Shannon Entropie, welche sich zwischen 0 und $\log_2(n)$ bewegt. Der Koeffizient $\frac{1}{\log_2(n)}$ skaliert die Funktion lediglich auf den Wertebereich von 0 bis 1, wodurch man folglich von der „normierten“ Entropie spricht.

$$H = -\frac{1}{\log_2(n)} \sum_{i=1}^n P_i \log_2(P_i) \quad (22)$$

Die Gleichung soll an einem Beispiel verdeutlicht werden. Gegeben sei ein Datensatz, bei dem 10% der Proben bei der Produktion entschlichtet wurden und 90% der Proben nicht. Unter der Voraussetzung, dass es nur diese beiden Klassen gibt, erhält man für die Entropie einen Wert von $H = 0,47$. Würde ein ausgeglicheneres Verhältnis von 40% entschlichteten und 60% nicht-entschlichteten Proben vorliegen, würde man einen größeren Wert für die Entropie von $H = 0,97$ erhalten. Dabei kann die Entropie auch für Verteilungen mit mehr als zwei Klassen angewandt werden.

Für numerische Variablen kann ebenfalls eine normierte Entropie berechnet werden, sofern die Daten zuvor in abgegrenzte Bereiche, sog. „Bins“, unterteilt wurden, was bildlich gesprochen einer Annäherung der Verteilung durch ein Histogramm entspricht. Allerdings

muss hier beachtet werden, dass die Größe der Bins einen Einfluss auf die berechnete Entropie hat. Ein Beispiel ist in Abbildung 19 gegeben, in welcher die Entropie der Variable „Carbon Conversion Ratio (CCR)“ anhand einer Einteilung in 10 bzw. 20 Bins berechnet wurde. Es wird daher deutlich, dass für einen Vergleich verschiedener Verteilungen einheitliche Bin-Anzahlen verwendet werden sollten. Zusätzlich sollten bei kontinuierlichen numerischen Variablen auch weitere statistische Kenngrößen wie Mittelwert, Standardabweichung, Minimum und Maximum verwendet werden, um die Variabilität zu bewerten.

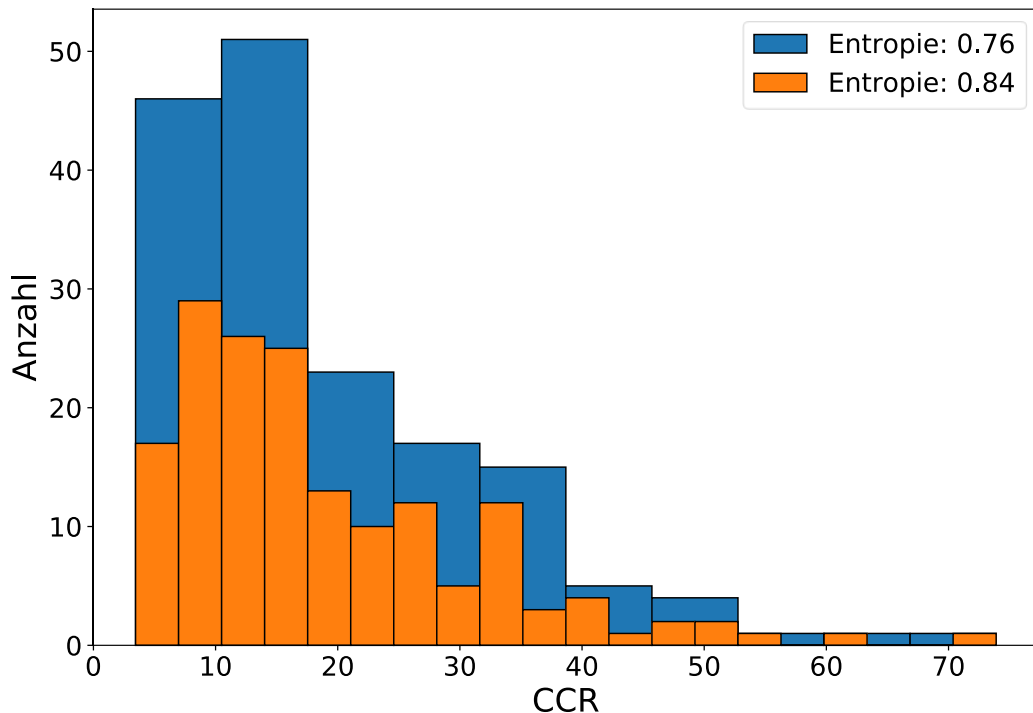


Abbildung 19: Normierte Entropie für eine (kontinuierliche) numerische Variable bei Einteilung in unterschiedliche Bins.

Es soll außerdem an dieser Stelle erläutert werden, warum die Standardabweichung zur Bewertung der Variabilität nicht ausreichend ist, obwohl sie ebenfalls ein statistisches Streuungsmaß darstellt. Einen Vorteil, den die Entropie gegenüber der Standardabweichung hat, ist die Tatsache, dass sie unabhängig von der Größe der Zahlen ist. In Abbildung 20 sind alle vier Kombinationen zwischen niedriger und hoher Entropie sowie niedriger und hoher relativer Standardabweichung in einer Matrix skizziert, wobei alle diese Fälle aus dem Datensatz dieser Arbeit entnommen wurden. Betrachtet man den Fall bei niedriger Entropie und hoher Standardabweichung (Abbildung 20 rechts oben), so sieht man, dass die überwältigende Mehrzahl an Proben eine Dicke unterhalb 10 mm aufweist. Einige wenige Ausnahmen weisen allerdings mit 30-35 mm eine deutlich größere Dicke auf, weshalb der Mittelwert verzerrt wird und eine hohe Standardabweichung von $\pm 108\%$ vorliegt. Hierdurch könnte man fälschlicherweise zu dem Schluss kommen, dass die Probendicke ausreichend

variiert wurde. Im Fall der Silizierungs-Temperatur (Abbildung 20 rechts unten) liegt ein ähnlicher Fall vor, nur dass hier die Werte näher beieinander liegen. Die Standardabweichung beträgt lediglich $\pm 3,5\%$, obwohl die Unausgeglichenheit der Klassen auch in diesem Fall sehr hoch ist. Dies wird in der Entropie wiedergespiegelt, welche in beiden Fällen niedrig ist, und damit anzeigt, dass beide Parameter sehr wenig variiert wurden. Kenntnis über die Standardabweichung ist für die Beurteilung der Variabilität bei kategorischen Variablen trotzdem wichtig, da bei ungünstiger Wahl der Bins auch verhältnismäßig hohe Entropie-Werten entstehen können, wenn die Bins einigermaßen ausgeglichen sind (Abbildung 20 links unten).

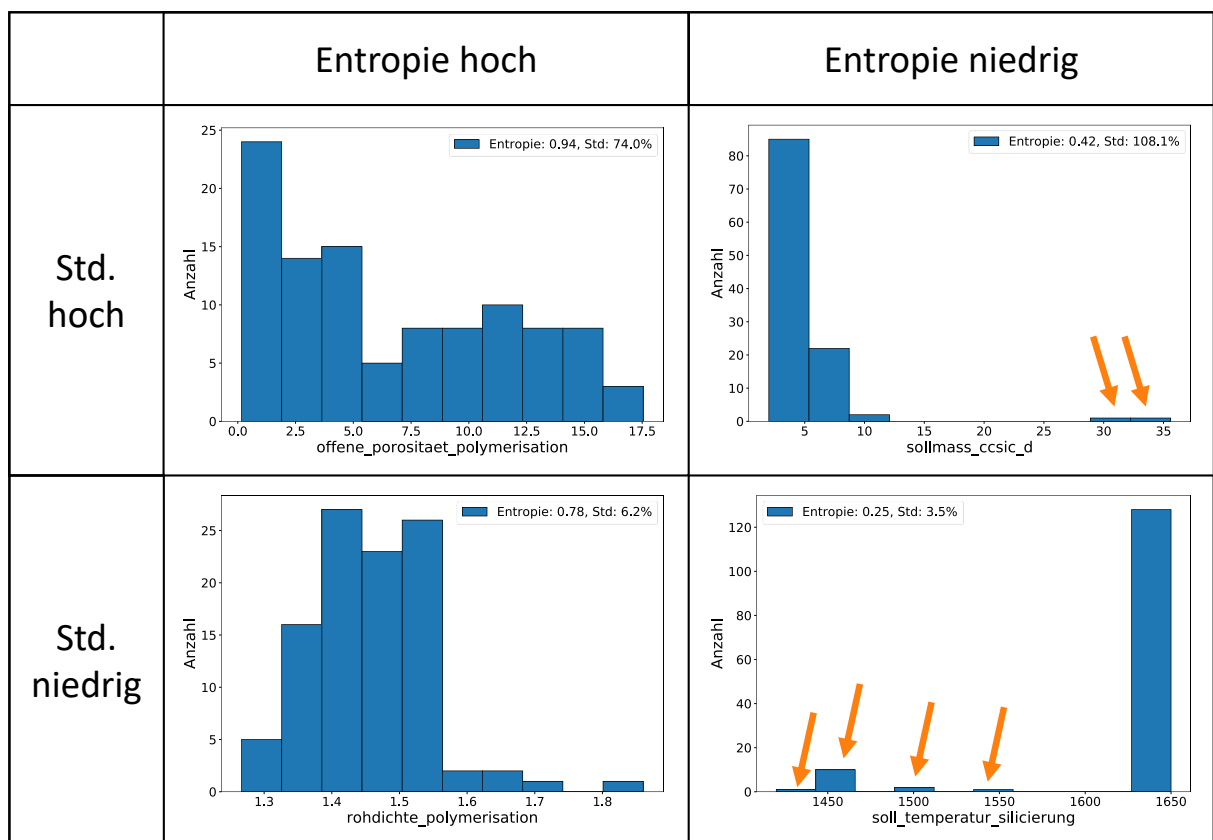


Abbildung 20: Kombinationsmatrix mit niedriger/hohere Entropie und niedriger/hohere Standardabweichung (Std.), orangefarbene Pfeile: Klassen mit sehr geringem Auftreten.

2.6 Machine-Learning Grundlagen

Da im Verlauf der Arbeit weitere Fachtermini zu bestimmten Methodiken und Zuständen aus dem Bereich Machine-Learning verwendet werden, sollen diese nun vorausgreifend erklärt werden.

2.6.1 Test- und Trainingsdaten

Unabhängig vom gewählten Algorithmus wird niemals der gesamte vorliegende Datensatz für den Trainingsprozess des KI-Modells verwendet, sondern gleich zu Beginn in Trainings-

und Test-Daten unterteilt [76]. Der Test-Datensatz wird zunächst außer Acht gelassen und erst wieder für die abschließende Genauigkeitsmessung des fertigen, trainierten Modells verwendet. Anderenfalls wäre es nicht möglich, die Genauigkeit des Modells bei neuen, bisher ungesehenen Datensätzen zu bestimmen, da das Modell die Lösung zu jedem Datenpunkt der Trainingsdaten bereits während des Trainingsprozesses gesehen hat. Die prozentuale Aufteilung zwischen Trainings- und Testdaten ist dabei nicht fest vorgeschrieben. Jedoch ist es aus Gründen der Effizienz vorteilhaft, den Testdatensatz lediglich so groß wie nötig zu machen. In der Praxis wird hierzu oft auf eine Unterteilung von 90:10, 80:20 oder 70:30 zugunsten des Trainings-Datensatzes gesetzt [77].

Zufälliges und stratifiziertes Ziehen

Zur Unterteilung eines Datensatzes in Test- und Trainingsdaten gibt es verschiedene Möglichkeiten. Meistens wird so lange zufällig aus dem gesamten Datensatz gezogen, bis alle Daten aufgeteilt sind, was besonders bei großen Datensätzen und ausgeglichenen Klassenverhältnissen innerhalb der Daten eine gute Strategie darstellt. Hat man aber sehr unausgeglichene Klassenverhältnisse, kann das zufällige Ziehen zu schlechten Ergebnissen führen, was anschaulich an folgendem Beispiel erklärt werden kann: es soll eine KI angelehrt werden, um festzustellen, ob es sich bei einer eingehenden E-Mail um Spam handelt oder nicht. Dazu liegt ein Datensatz von 100 E-Mails vor, von welchen es sich nur bei 2 um Spam handelt. Durch zufälliges Splitten könnte es sein, dass beide Spam-E-Mails im Test-Datensatz landen, weshalb die KI im Trainingsprozess nie mit Spam E-Mails konfrontiert wird und dadurch alle E-Mails als „kein Spam“ vorhersagt. In solchen unausgeglichenen Datensätzen wird häufig stratifiziertes Splitten angewendet. Hierbei werden die prozentualen Klassenanteile im Trainings- und Test-Datensatz gewahrt. Im obigen Beispiel würde das bedeuten, dass jeweils 2% der Mails im Test- und im Trainingsdatensatz Spam sein müssen.

Dieses Vorgehen funktioniert allerdings nur für kategoriale Daten, da bei numerischen Daten keine festen Grenzen existieren. Um auch numerische Daten stratifiziert splitten zu können, müssen diese zuvor in Bins unterteilt werden, welche dann zählbare Kategorien darstellen. Ein Beispiel dafür liefert die vorliegende Arbeit. Unter allen CCR-Messungen sind diejenigen mit niedrigen CCR-Werten deutlich in der Überzahl. CCR-Messungen über 40% sind dabei sehr selten. Deswegen wurde die gesamte Spannbreite der CCR (0% bis 100%) auf 5%-Pakete aufgeteilt, aus denen dann prozentual gleiche Anteile im Trainings- und Testdatensatz einfließen. Dadurch ist gewährleistet, dass die dem Training und Testen der KI zugrundeliegenden Daten repräsentativ für den gesamten Datenpool sind. Lediglich der letzte Bin (oberhalb CCR=40%) wurde vergrößert, da hier kaum noch Daten vorlagen. Grafisch ist

dies in Abbildung 21 dargestellt, in welcher die CCR in Bins aufgeteilt, und dann als Histogramm aufgetragen wurde, sodass die vertikale Achse die Anzahl der Proben im jeweiligen Bin repräsentiert. Blaue Bins stellen dabei Trainings-Daten und orangefarbene Bins Test-Daten dar. Es ist erkennbar, dass in diesem Fall beim zufälligen Splitten keine Proben mit $CCR > 35\%$ oder $CCR < 5\%$ im Test-Datensatz vorliegt, was zu einer Fehleinschätzung der Modell-Genauigkeit führen kann, welche über den Test-Datensatz bestimmt wird. Beim stratifizierten Splitten werden die Klassen-Proportionen besser gewahrt. Aufgrund der Vorgabe eines festen Prozentsatzes von Proben für den Trainings- und Test-Datensatz ist jedoch anzumerken, dass auch hier nicht immer eine perfekte Erhaltung der Proportionen für jeden Bin sichergestellt werden kann.

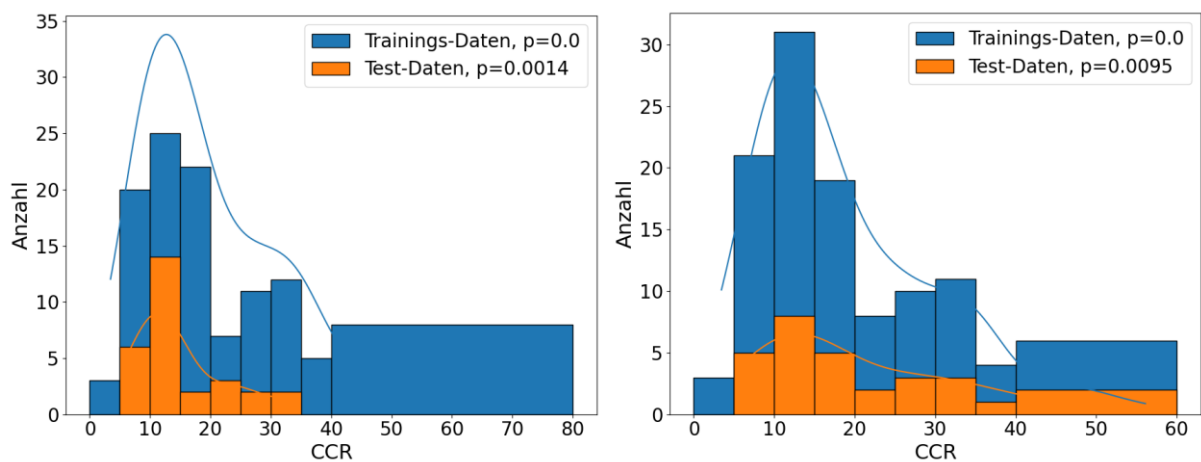


Abbildung 21: Vergleich von zufälligem Splitten (links) und stratifiziertem Splitten (rechts). Blau: Trainings-Datensatz, orange: Test-Datensatz. Zu beachten ist die bessere Wahrung der Klassen-Proportionen beim stratifizierten Splitten.

2.6.2 Overfitting und Underfitting

Die beiden Extreme Überanpassung und Unteranpassung (*engl.*: Overfitting und Underfitting) beschreiben zwei unerwünschte Zustände eines Machine-Learning Modells. Im Falle des Underfitting ist das Modell (noch) nicht gut genug an die Trainingsdaten angepasst und kann daher keine genauen Vorhersagen liefern. Im Falle des Overfitting besteht das entgegengesetzte Problem: hier ist das Modell so genau an die Trainingsdaten angepasst, dass jede kleinste Schwankung der Messdaten auswendig gelernt wurde. Die Genauigkeit bei noch ungesehenen Datensätzen verschlechtert sich dabei im Gegensatz zu einem optimal angepassten Modell wieder. Dies ist in Abbildung 22 skizziert. Im Falle des Underfitting kann die Krümmung der blauen Kurve durch die Annahme einer zu simplen linearen Beziehung (grüne Gerade) nie wirklich erfasst werden. Im Falle des Overfittings hat das Modell die Abweichung von den Trainingsdaten (weiß gefüllte Punkte) allerdings so weit minimiert, dass dies nur auf Kosten der Vorhersagegenauigkeit noch ungesehener Datenpunkte möglich war.

Das Verständnis für die wahre Beziehung, die sog. „Fähigkeit zur Verallgemeinerung“, ist durch die Überanpassung verloren gegangen [78, 79]. Overfitting kann durch die Anzahl der Datensätze oder durch Reduzierung der Modellkomplexität vermindert werden [79].

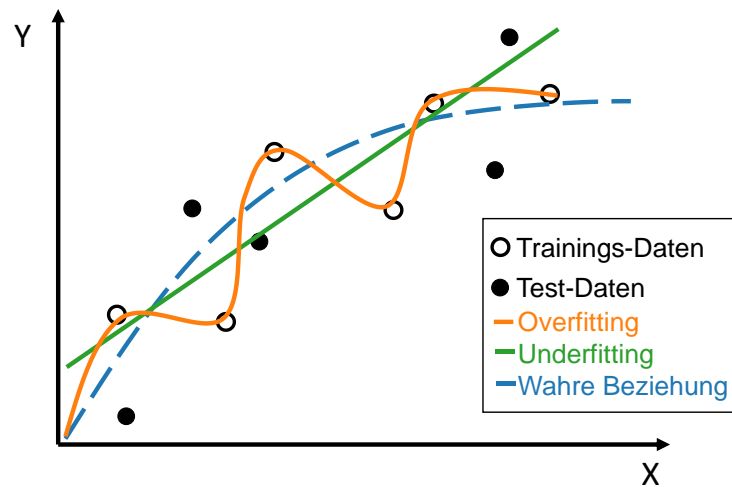


Abbildung 22: Overfitting und Underfitting an einem Beispiel; Legende: weiß gefüllte Punkte: Trainingsdaten, schwarz gefüllte Punkte: bisher ungesehene Datensätze, grün: Vorhersage des Modells im Falle des Underfitting, orange: Vorhersage des Modells im Falle des Overfitting, blau: wahre Beziehung, welche erlernt werden sollte.

2.6.3 Kreuzvalidierung

Kreuzvalidierung ist eine effiziente Methode, um die Genauigkeit eines Machine-Learning Modells zu bestimmen und gehört zu den Monte-Carlo Methoden [77]. Besonders bei kleineren Datensätzen tendieren Machine-Learning Modelle dazu, sich zu sehr an die Trainingsdaten anzupassen, was zu Overfitting führen kann. Um die Generalisierungsfähigkeit dieser Modelle beurteilen zu können, wird häufig Kreuzvalidierung verwendet [80]. Dabei wird der Datensatz in n Teile unterteilt (häufig wird $n = 10$ gesetzt [77]), wovon dann $n - 1$ Teil-Datensätze zum Trainieren des Modells verwendet werden und der letzte Teil-Datensatz zur Validierung verwendet wird, um die Modellgenauigkeit zu bestimmen. Da es aber möglich ist, dass zufällig ein besonders einfacher oder besonders schwerer Teil-Datensatz zur Bestimmung verwendet wurde, wird im nächsten Schritt der nächste Teil-Datensatz als Validierungs-Datensatz zurückbehalten und das Modell anhand der verbliebenen $n - 1$ Datensätze trainiert. Dann kann erneut die Genauigkeit bestimmt werden. Diese Schleife wird dann für alle n Datensätze wiederholt, wodurch Mittelwert und Standardabweichung der Genauigkeit berechnet werden können. Kreuzvalidierung ist daher effizienter als ein Unterteilen des gesamten Datensatzes in Test- und Trainingsdaten, wobei die Modellgenauigkeit anhand der Testdaten bestimmt wird, da durch Kreuzvalidierung der gesamte Datensatz für das Training verwendet werden kann [80, 81]. Das Verfahren ist schematisch in Abbildung 23 dargestellt, wobei $D_{train,1}$ den ersten gesamten Trainingsdatensatz darstellt und $D_{val,1}$ den ersten Validierungsdatensatz.

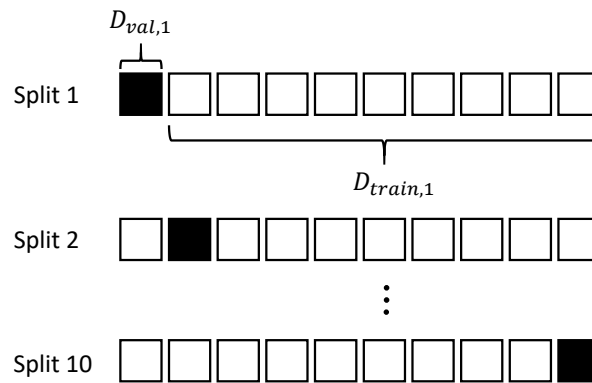


Abbildung 23: Schematische Darstellung des Kreuzvalidierungs-Verfahrens am Beispiel einer 90%/10% Aufteilung zwischen Trainings- und Test-Daten nach [77].

Allerdings darf Kreuzvalidierung nicht in allen Fällen zum Verzicht auf einen separaten Test-Datensatz führen. Wird Kreuzvalidierung beispielsweise verwendet, um die Hyperparameter eines Estimators zu optimieren, fließen dadurch Informationen über den Test-Datensatz in das Modell. Dadurch ist die berechnete Genauigkeit nicht mehr vertrauenswürdig und die Generalisierungsfähigkeit des Modells möglicherweise schlechter als berechnet. In diesem Fall sollte Kreuzvalidierung nur verwendet werden, um die besten Hyperparameter zu finden und die Genauigkeit des besten Modells anschließend anhand eines separaten Test-Datensatzes berechnet werden, wie in Abbildung 24 dargestellt.

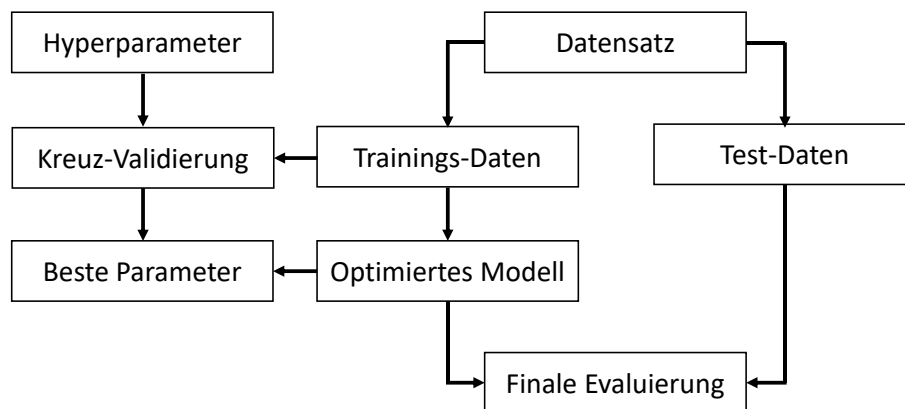


Abbildung 24: Verwendung der Kreuzvalidierung bei der Optimierung der Hyperparameter eines Estimators [82].

2.6.4 Bagging und Bootstrapping

Bagging (auch: Bootstrap aggregating) bezeichnet eine Methode zur Erstellung mehrerer Versionen eines Modells, um ein aggregiertes Modell zu erhalten. Durch die Aggregation wird ein Mittelwert aus allen Vorhersagen der unterschiedlichen Teilmodelle gebildet, mit dem Ziel die Varianz des Gesamtmodells zu verringern [83]. Die Variation der Modelle wird durch Stichproben-Datensätze k (*engl.*: bootstrapped datasets) erreicht, welche durch Ziehen aus dem vollständigen Datensatz D mit Zurücklegen gezogen werden. Bei einer Anzahl von n Proben im Datensatz wird beim Bagging ein neuer Datensatz mit k Proben erstellt, die zufällig aus D

gezogenen werden, wobei $k < n$ gilt. Durch das zufällige Ziehen mit Zurücklegen, kommen manche Proben möglicherweise gar nicht und manche sogar mehrfach in den jeweiligen Stichproben-Datensätzen D_i vor. Um die Diversifikation der einzelnen Modelle eines Ensembles zu erhöhen, wird dann jedes Modell mit einem anderen Stichproben-Datensatz trainiert. Die höhere Diversifikation führt dabei zu einer robusteren Vorhersage mit geringerer Neigung zum Overfitting [36]. Abbildung 25 zeigt das Vorgehen beim Bootstrapping anhand eines Schemas. Die Aggregierung bezeichnet dann das Mitteln über die Vorhersagen der so erstellten Modelle.

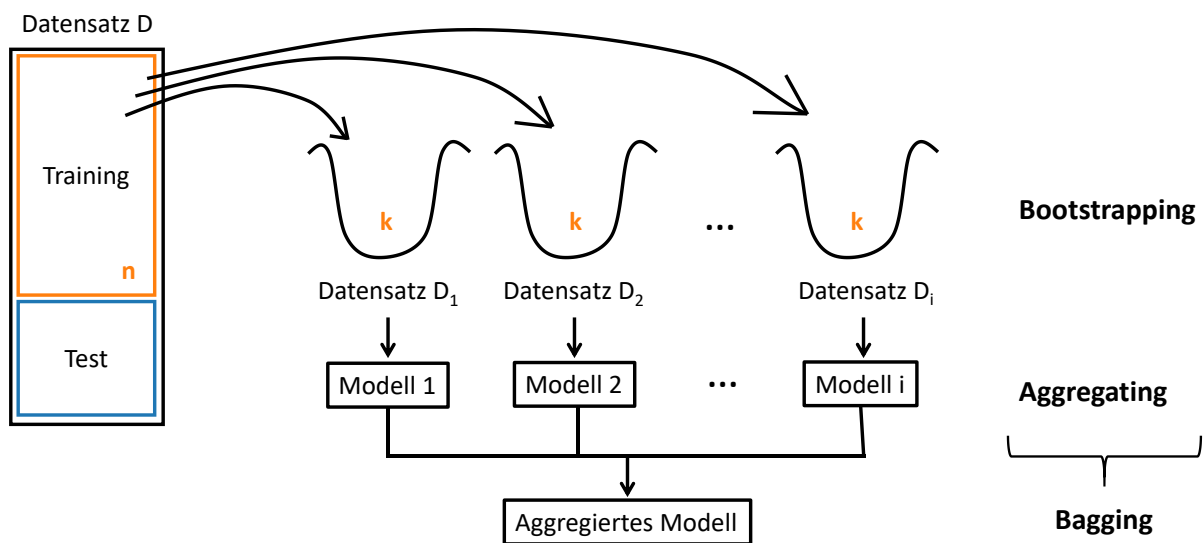


Abbildung 25: Erstellung von i Stichproben-Datensätzen D_i („Bootstrapping“) mit jeweils k Proben für das Training mehrerer Versionen desselben Modells und Aggregieren der Vorhersagen der Modelle.

2.6.5 Feature Selection

Die Trennung von wichtigen und unwichtigen Parametern aus dem gesamten zugrundeliegenden Datenpool vor dem eigentlichen Training eines Machine-Learning Modells wird als „Feature Selection“ bezeichnet. Feature Selection hat dabei mehrere positive Effekte. So werden dadurch bereits im Vorfeld unwichtige Parameter aus dem Datensatz entfernt, wodurch sich die Dimensionalität des Problems reduziert, was besonders bei kleinen Datensätzen mit einer großen Anzahl an Parametern hilfreich ist. Denn meist ergibt sich hieraus eine Genauigkeitserhöhung des Modells, da keine zufälligen Zusammenhänge zwischen unkorrelierten Daten als Trend missinterpretiert werden (Overfitting). Weiterhin dient Feature Selection der Verringerung der Rechendauer von KI-Modellen, indem Parameter, welche kaum oder gar keine Korrelationen mit der Zielgröße aufweisen, schon im Preprocessing aus dem Datensatz entfernt werden. Ein weiterer großer Vorteil ist die gesteigerte Interpretierbarkeit des

Modells aufgrund der reduzierten Dimensionalität, denn oft ist von Interesse, auf welcher Grundlage ein KI-Modell seine Prognosen tätigt [84].

Um Feature Selection anwenden zu können, muss zunächst die Wichtigkeit aller Parameter im Datensatz quantifiziert werden, um diese anschließend zu vergleichen. Dazu existieren mit Filter-Methoden, Wrapper-Methoden und Embedded-Methoden drei unterschiedliche Arten der Feature Selection, welche weiter in verschiedene Kategorien unterteilt werden können. Die folgenden vier Methoden werden dabei in der Praxis häufig benutzt und werden deswegen auch in dieser Arbeit verwendet und in DataTracker implementiert:

- Modell-Intrinsische Feature Selection (Embedded-Methode)
- Sequenzielle Feature Selection (Wrapper-Methode)
- Permutations-Feature Selection (Wrapper-Methode)
- Recursive Feature Elimination (Wrapper-Methode)

Mit der Pearson Korrelation und dem Variance Inflation Factor wurden in DataTracker außerdem zwei Filter-Methoden implementiert (siehe Kapitel 2.5.2). Da diese jedoch nicht für den Prozess der Feature-Selection verwendet wurden, sondern lediglich zur Prüfung auf Multikollinearität, werden die beiden Methoden in diesem Kapitel nicht aufgeführt.

Modell-Intrinsische Methode

Bei der Modell-Intrinsischen Methode wird ein unabhängiges KI-Modell dazu verwendet, um die relative Wichtigkeit der Herstellungsparameter zu berechnen. Dabei ergibt die Summe aller relativen Wichtigkeiten immer 100%. Je wichtiger ein Parameter im Vergleich zu anderen Parametern ist, desto höher fällt sein Prozentsatz aus. Die Berechnung basiert dabei auf der mittleren Reduktion des RMSE (Root Mean Squared Error) über alle Bäume des RandomForests. Die Formel für die Berechnung der Wichtigkeiten $Imp(X_j)$ für alle Merkmale X_j ist dabei in Gleichung (23) gegeben. Dabei beschreibt M die Anzahl der Bäume im RandomForest, φ_m den m -ten Baum, $p(t)$ den Anteil der Proben, die den Knoten t im Baum m erreichen, j_t die Variable welche für den Split in Knoten t verwendet wird und $i(t)$ das Fehlermaß, also in diesem Fall den RMSE [85, 25].

$$Imp(X_j) = \frac{1}{M} \sum_{m=1}^M \sum_{t \in \varphi_m} 1(j_t = j) [p(t) \Delta i(s_t, t)] \quad (23)$$

Das Vorgehen ist als Pseudocode in Schema 2 verdeutlicht.

Schema 2: Vorgehen bei der Intrinsischen Feature Selection.

Schritt 1: Für jeden Baum des RandomForest...

Schritt 1.1: Für jeden darin vorkommenden Parameter P_i ...

Schritt 1.1.1: Berechne die Varianz-Reduktion des Modells durch die Verwendung von P_i anhand RMSE

Schritt 1.1.2: Gewichte die Varianz-Reduktion anhand der Anzahl der betroffenen Proben

Schritt 2: Summiere Varianz-Reduktionen durch P_i über alle Knoten desselben Baums

Schritt 3: Summiere Varianz-Reduktion durch P_i über alle Bäume und normiere anhand der Anzahl der Bäume

Sequenzielle Feature Selection und Recursive Feature Elimination (RFE)

Anstelle eines zusätzlichen ML-Modells kann die Merkmals-Wichtigkeit auch durch eine sequenzielle Methode ermittelt werden. Es existieren vorwärts gerichtete und rückwärts gerichtete sequenzielle Methoden, von denen nun zuerst die vorwärts gerichtete Selektion erläutert wird. Dabei wird dem Modell zunächst nur ein einziger Herstellungsparameter gegeben, anhand dem es die Zielgröße (z.B. CCR) vorhersagen soll. Nachdem das Modell nacheinander mit jedem einzelnen Herstellungsparameter trainiert, und jeweils die Genauigkeit zur Bestimmung der Zielgröße berechnet wurde, ist bekannt, welcher Parameter die Genauigkeit am stärksten beeinflusst. Anschließend wird dem Modell ein weiterer Parameter hinzugefügt, indem wieder für alle möglichen Zusatzparameter die Genauigkeit berechnet und der Beste davon ausgewählt wird. Das Vorgehen ist schematisch in Schema 3 dargestellt [86, 87].

Schema 3: Vorgehen bei der Sequenziellen Vorwärts-Selektion.

Schritt 1: Für jeden Parameter P_i für $i = 1 \dots N$ des Modells...

Schritt 1.1: Trainiere das Modell ausschließlich anhand P_i

Schritt 1.2: Bewerte Genauigkeit des Modells mit P_i anhand R^2

Schritt 2: Behalte Parameter, welcher die höchste Modellgenauigkeit erzielt hat

Schritt 3: Füge so lange weitere Parameter nach dem Schema der Schritte 1 und 2 hinzu, bis sich die Genauigkeit nicht mehr signifikant erhöht (Abbruchkriterium angeben)

Bei der Rückwärts-Selektion, die auch als Recursive Feature Elimination (kurz: RFE) bezeichnet wird, wird dieselbe Idee verfolgt, jedoch entgegengesetzt vorgegangen. Hier wird zunächst mit allen Parametern des Datensatzes begonnen und so lange sequenziell einer davon entfernt, bis die Genauigkeit sich signifikant verschlechtert. Welcher Parameter entfernt wird, hängt davon ab, wie stark er sich auf die Genauigkeit auswirkt. So wird stets der Parameter mit der geringsten Auswirkung entfernt. Das Vorgehen ist in Schema 4 verdeutlicht.

Schema 4: Vorgehen bei der Sequenziellen Rückwärts-Selektion.

Schritt 1: Für jeden Parameter P_i für $i = 1 \dots N$ des Modells...

Schritt 1.1: Trainiere das Modell ohne P_i

Schritt 1.2: Bewerte Genauigkeit des Modells ohne P_i anhand R^2

Schritt 2: Entferne Parameter mit geringster Auswirkung auf die Modellgenauigkeit

Schritt 3: Entferne so lange weitere Parameter nach dem Schema der Schritte 1 und 2, bis sich die Genauigkeit sprunghaft verschlechtert (Abbruchkriterium angeben)

Permutation

Die vierte in DataTracker implementierte Möglichkeit, um die Wichtigkeit der Eingangsparameter zu bewerten, ist die Permutations-Methode. Hierbei werden die Daten eines beliebigen Eingangsparameters gemischt, sodass jede mögliche Korrelation mit der Zielgröße verloren geht. Danach wird die Genauigkeit des KI-Modells mit dem permutierten Parameter bestimmt. Anschließend wird das Vorgehen für jeden Parameter im Datensatz wiederholt, wobei immer nur ein Parameter gleichzeitig permutiert wird. Durch die errechneten Genauigkeiten der Modelle kann bestimmt werden, welche Permutation die größten Genauigkeitseinbußen nach sich zieht. Der so gefundene Parameter hat folglich auch die größte Wichtigkeit für das Modell [88].

Schema 5: Vorgehen bei der Permutation.

Schritt 1: Für jeden Parameter P_i für $i = 1 \dots N$ des Modells...

Schritt 1.1: Mische P_i , sodass die mögliche Korrelation zur Zielvariable verloren geht und trainiere Modell mit allen Parametern inklusive modifiziertem P_i

Schritt 1.2: Bewerte Genauigkeit des Modells anhand R^2

Schritt 2: Bewerte Wichtigkeit $Imp(P_i)$ aller Parameter aufgrund der jeweiligen Absenkung von R^2 durch Mischen von P_i

2.7 Mikrostruktur-Simulation

Die Simulation mikrostruktureller Vorgänge wird in dieser Arbeit durch das Finite Elemente Programm „Simcenter Multimech 2022“ der Firma Siemens (im Folgenden als „Multimech“ bezeichnet) durchgeführt. Die Finite Elemente Methode (FEM) ist eine in der Materialwissenschaft sehr häufig verwendete Methode, welche alle Konstituenten eines Verbundmaterials als Kontinuum betrachtet. Somit ist es möglich, die Einflüsse variabler mechanischer Eigenschaften von Fasern, Matrix und Interface auf die Gesamteigenschaften des Verbunds zu untersuchen [89].

In der FEM-Simulation wird die Geometrie durch Zerteilung in kleine (finite) Elemente angenähert, deren Eckpunkte als Knoten bezeichnet werden. Jedem Knoten wird ein individueller Festigkeitswert zugeordnet, dessen Wert durch praktische Tests bestimmt werden muss. In der Realität ist die Materialfestigkeit allerdings aufgrund von lokalen Defekten nicht überall im gesamten Bauteil exakt dieselbe. Defekte im Material entstehen bereits während der Herstellung, beispielsweise bei der Vernetzung des Harzes oder durch Ausgasen von Schlichte oder anderen flüchtigen Stoffen während des Aufheizvorgangs, und äußern sich meist in mikroskopisch kleinen Rissen. Dies hat stets eine lokale Absenkung der Materialfestigkeit zur Folge. In der Simulation wird dies durch eine Weibull-Verteilung der Festigkeiten berücksichtigt, welche eine zwei-parametrische Funktion ist und dadurch gut an alle beliebigen Wahrscheinlichkeitsverteilungen angepasst werden kann. Die Parameter λ und κ bestimmen dabei über die Form, Schiefe und Position der Kurve, so können durch ihre Auswahl auch andere bekannte Verteilungen wie die Normal- und die Exponentialverteilung nachgebildet werden. Beispiel-Graphen für verschiedene Parameterpaare der Weibullverteilung sind in Abbildung 26 zu sehen.

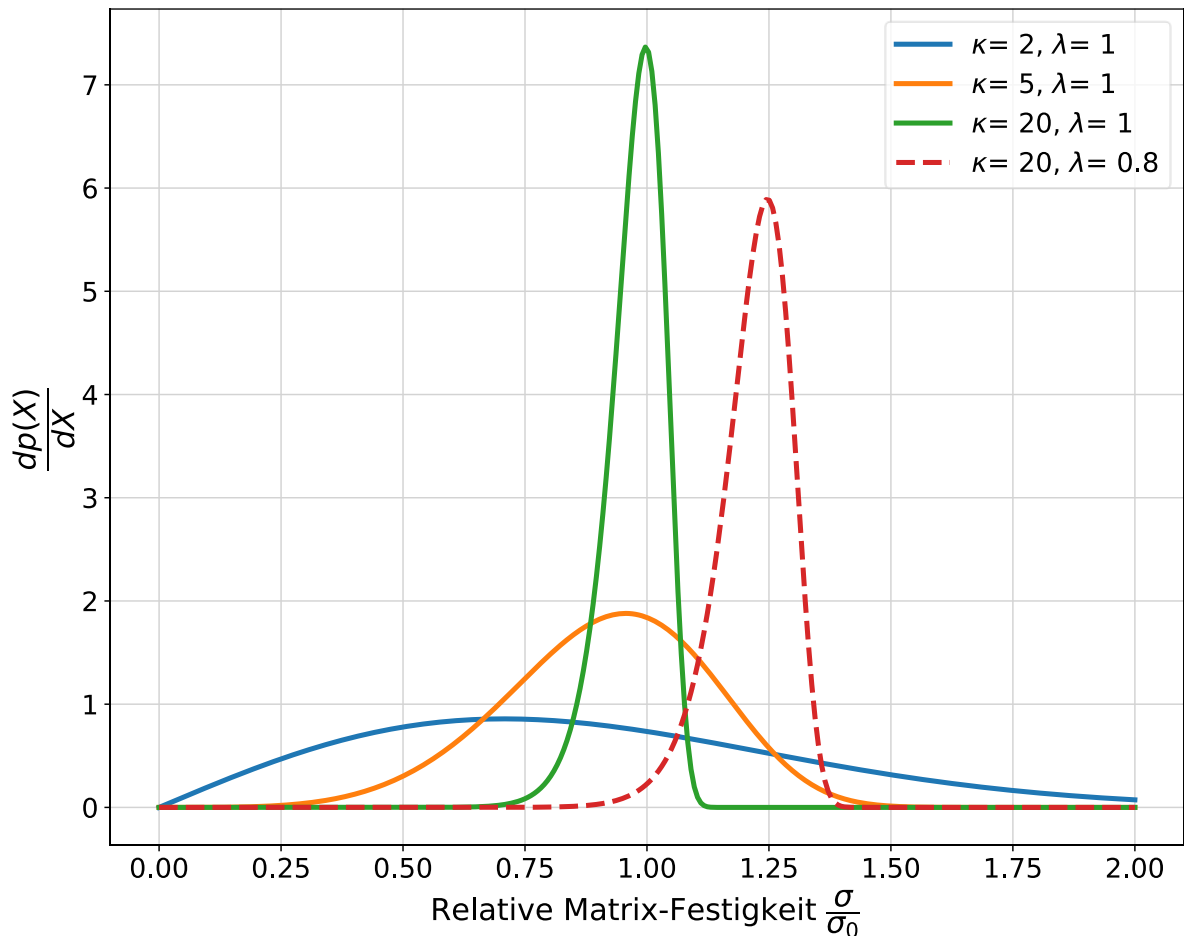


Abbildung 26: Dichtefunktion der Weibull-Verteilung für verschiedene Formparameter κ und Skalenparameter λ . Auf der horizontalen Achse ist die relative Matrix-Festigkeit aufgetragen, wobei der Wert $\frac{\sigma}{\sigma_0} = 1$ der durch Tests bestimmten mittleren Festigkeit entspricht.

Die der Dichtefunktion zugrundeliegende Formel ist in Gleichung (24) gegeben [90].

$$f(\sigma) = \lambda \kappa (\lambda \sigma)^{\kappa-1} \exp(-\lambda \sigma)^\kappa \quad (24)$$

Dabei steht σ_0 für die Matrix-Festigkeit, σ für die aktuell anliegende Spannung, λ für den Weibull-Skalierungsfaktor und κ für den Weibull-Formfaktor. Der Bruch $\frac{\sigma}{\sigma_0}$ beschreibt demnach den Anteil der Knoten-Festigkeit an der nominalen Festigkeit, und wird daher auch „relative Festigkeit“ genannt. Ein Wert für die relative Festigkeit von $\frac{\sigma}{\sigma_0} = 1$ würde bedeuten, dass die Festigkeit des beobachteten Knotens gerade der im Test ermittelten Festigkeit (Mittelwert aller Proben) entspricht. Würde diese jedoch für alle Knoten gleichermaßen angenommen werden, würden beim Erreichen der Bruchspannung alle Knoten gleichzeitig versagen, was weder realistisch, noch numerisch stabil ist. Der dimensionslose Formfaktor κ kann als Maß für die Irregularität im Werkstoff verstanden werden. Je höher der Wert, desto enger sammeln sich die Werte der Verteilung um den Mittelwert, je kleiner der Wert, desto größere Streuung liegt vor (vergleiche Abbildung 26). In der Werkstoffmodellierung werden

meist Formfaktoren von bis zu $\kappa = 2 \dots 10$ verwendet [91]. Eine Abschätzung von κ kann auch über Gleichung (24) berechnet werden, wobei die mittlere Festigkeit σ_0 und deren Varianz $var(\sigma)$ durch praktische Tests bestimmt werden müssen [90]. Daraus ergibt sich für diese Arbeit ein Weibull-Formfaktor von $\kappa = 16,6$. Der Skalierungsfaktor λ skaliert die Höhe und Breite der Verteilung bei gleicher Form. Er wird in dieser Arbeit analog zu Wittel et al. [91] mit dem Wert $\lambda = 1$ angenommen.

$$\kappa = \left(\frac{\sqrt{var(\sigma)}}{\sigma_0} \right)^{-1,086} \quad (25)$$

Die Verteilungsfunktion der Festigkeit berechnet sich nach Gleichung (26) und ist in Abbildung 27 dargestellt [91]. Sie beschreibt die kumulierte Anzahl Knoten, welche bei einer vorgegebenen relativen Festigkeit versagen.

$$P(\sigma) = 1 - \exp \left[- \left(\lambda \frac{\sigma}{\sigma_0} \right)^\kappa \right] \quad (26)$$

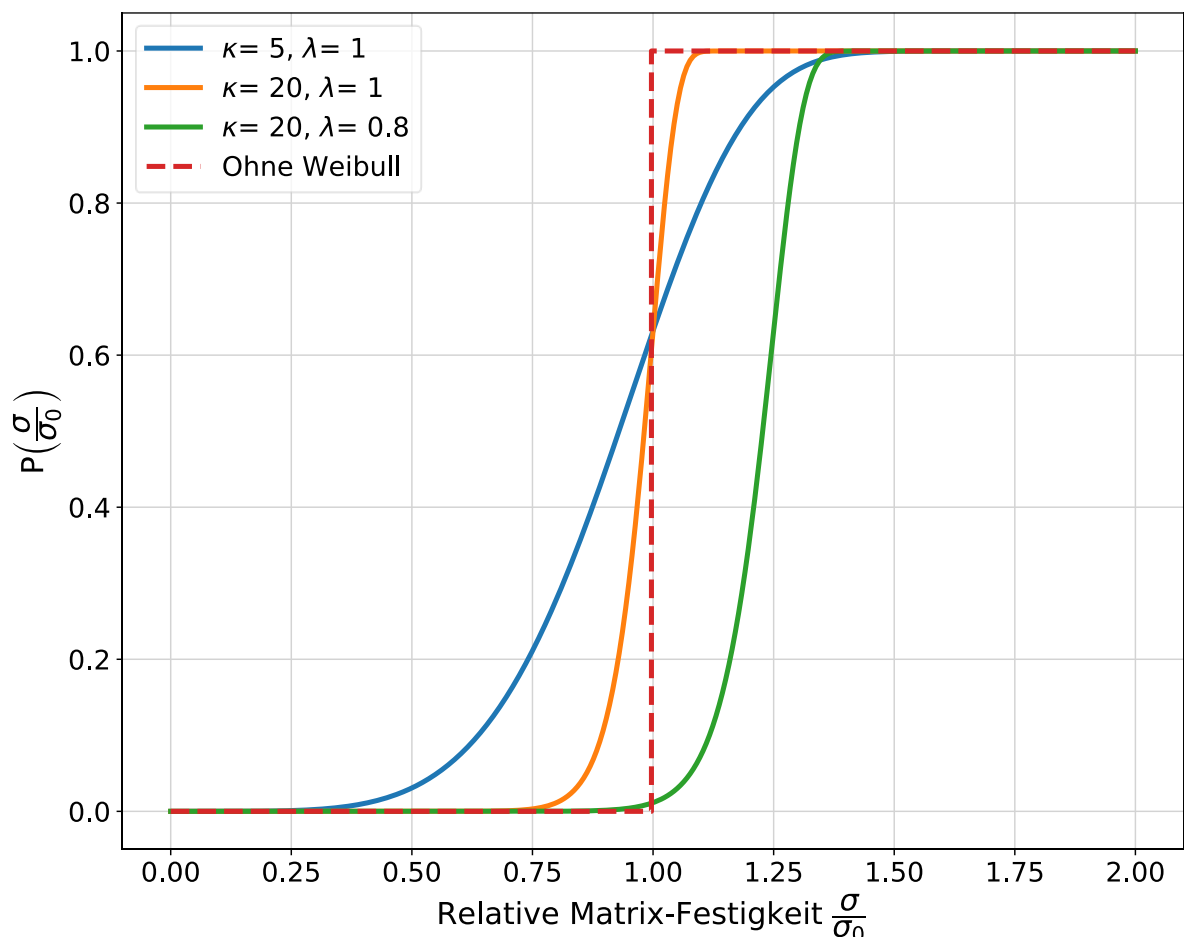


Abbildung 27: Verteilungsfunktion bei unterschiedlichen Formparametern. Im Gegensatz zur Stufenfunktion, haben bei der Weibullfunktion nicht alle Knoten desselben Materials dieselbe Festigkeit, wodurch Defekte im Material simuliert werden.

3 Entwicklung eines Web-Interfaces

In diesem Kapitel wird das Web-Interface beschrieben, welches für die Erfassung, Prozessierung und Speicherung der Prozessdaten, sowie als Informationsplattform für die Mitarbeiter verwendet wird. Die Unterkapitel befassen sich jeweils mit den verschiedenen Funktionalitäten der App.

3.1 Notwendigkeit und Ziele des Programms

Das Web-Interface stellt, neben dem GUI DataTracker, einen der beiden zentralen Bausteine dieser Arbeit dar. Die Ziele, welche durch dessen Einführung erreicht werden sollten, lassen sich wie folgt zusammenfassen:

- (Halb-)automatisches Einlesen von Datensätzen über einen lokalen Server, der für alle Abteilungsmitglieder über einen Web-Browser erreichbar ist
- Automatische Vorverarbeitung der Daten
- Speichern der Daten in einer verknüpften PSQL-Datenbank
- Erschaffung eines Kommunikationskanals zwischen technischen und wissenschaftlichen Mitarbeitern (Prozesse digital anfordern oder abschließen)
- Abteilungsübergreifende Konventionen einführen (durch Etablierung eines Standardvorgehens)

3.2 Überblick und Funktionen

Das Web-Interface wurde basierend auf dem Django-Framework geschrieben, welches über das DLR-Intranet erreichbar ist. Mitarbeitern ist somit der Zugriff über einen beliebigen Internet-Browser möglich. Das Programm ist in mehrere Apps unterteilt, welche in Python programmiert wurden und, je nach Benutzereingabe, über eine graphische Oberfläche unterschiedliche HTML-Seiten bereitstellt. Es dient hauptsächlich der Erfassung und Vorverarbeitung von Daten, welche anschließend im Hintergrund in einer PSQL-Datenbank gespeichert werden. Der Zugriff auf die Datenbank ist dabei nur durch Rechner aus der eigenen Abteilung möglich. Die Startseite des Programms ist in Abbildung 28 dargestellt.



Abbildung 28: Hauptmenü des erstellten Web-Interface.

Über das Web-Interface sind folgende Funktionen möglich, auf die im weiteren Verlauf dieser Arbeit detaillierter eingegangen wird:

- Virtuelle Proben erstellen
- Prozessschritte für Proben anfordern
- Proben teilen
- Dateien hochladen (Excel, CSV...)
- REM-Bilder hochladen
- Auftragsstatus der sich in Prozessierung befindlichen Proben einsehen
- Gespeicherte Daten ausgewählter Proben anzeigen
- Gespeicherte Daten von Proben aus der Datenbank löschen

3.2.1 Proben erstellen

In diesem kurzen Unterprogramm können Proben virtuell erstellt werden, sodass direkt zu Beginn des Produktlebenszyklus ein digitaler Zwilling der physikalischen Probe angelegt wird. Nachdem der Benutzer einen Namens-Präfix ausgewählt hat (z.B. WR-, HP-, IP-...), prüft das System automatisch, welche Laufzahl für die neue Probe vergeben werden muss und erstellt diese. Beispielsweise erstellt das System automatisch die Probe WR-437, wenn die Probe WR-436 bereits existiert. Außerdem wird automatisch ein Projektordner auf dem gemeinsamen Laufwerk der Abteilung angelegt. Alternativ kann eine manuelle Probenerstellung ausgewählt werden, worüber ein frei wählbarer Name vergeben werden kann. Allerdings funktioniert die Erstellung nur, wenn der frei gewählte Name nicht schon in der Datenbank enthalten ist. In diesem Fall wird der Benutzer durch eine Fehlermeldung informiert. Über den Haken „Aufbau

gleich wie“ kann außerdem der Aufbau einer bereits existierenden Probe für die neue Probe kopiert werden. Die zugehörige Eingabemaske ist in Abbildung 29 dargestellt.

Neue Probe erstellen

► Hilfe zum Proben erstellen (hier klicken)

Art der Probenerstellung:

Automatische Probenerstellung Manuelle Probenerstellung

Präfix wählen:

Präfix wählen...

Weitere Angaben:

Projekt Projektleiter tt. mm. jjjj Kostenträger

(Optional) Gleichen Aufbau für neu erstellte Probe verwenden wie bei folgender Probe:

Aufbau gleich wie: Probenamen eingeben...

Probe erstellen

[Hauptmenü](#)

Abbildung 29: Eingabemaske bei der automatischen oder manuellen Erstellung einer neuen Probe.

3.2.2 Prozessschritt anfordern

Diese Funktionalität ermöglicht es Projektleitern auf übersichtliche Weise eine Prozessierung ihrer Probe anzufordern. Die angeforderten Informationen werden im System gespeichert und können von Maschinenbedienern aufgerufen und erledigt werden.

Möchte der Projektleiter seine Probe beispielsweise zur Pyrolyse freigeben, kann dies über das Feld „Pyrolyse“ geschehen (siehe Abbildung 30). Soll stattdessen eine neue Probe erstellt werden, wird ein „Aufbau“ angefordert und die gewünschten Herstellungsparameter im Online-Laufzettel angegeben (siehe Abbildung 31). Außerdem können zusätzliche Messungen angefordert werden, beispielsweise eine Bauteilvermessung oder eine Dichte- und Porositätsmessung. Wurde der Prozessschritt angefordert, wird er grün angezeigt, und erscheint außerdem für den Techniker als offener zu bearbeitender Auftrag im Abschnitt „Auftrags-Status“. Sobald der Auftrag erledigt wurde, wird er auch dort grün gekennzeichnet. Die farbliche Kennzeichnung nach dem Ampelprinzip macht die Bearbeitung dabei besonders übersichtlich und effizient.



Abbildung 30: Abschnitt Prozessschritte anfordern des programmierten Web-Interfaces; grüne Felder sind bereits abschließend bearbeitet, graue Felder wurden noch nicht angefordert.

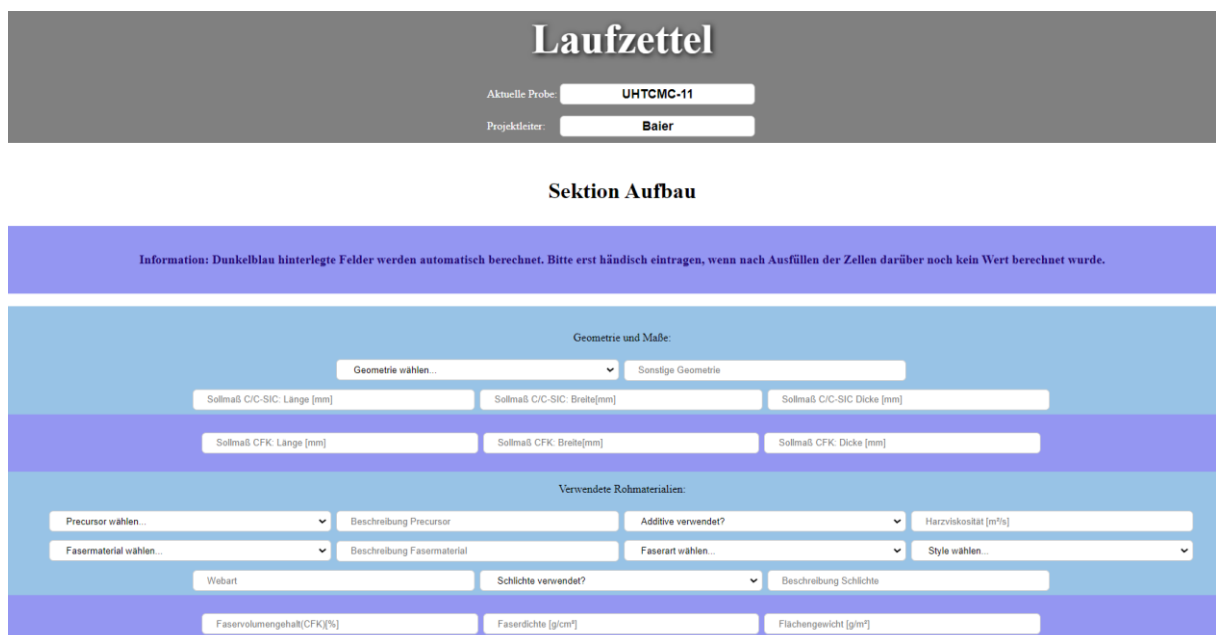


Abbildung 31: Sektion Aufbau des Abschnitts Prozessschritte anfordern.

3.2.3 Probe teilen

Dieser Abschnitt dient dazu, Proben, die sich bereits im System befinden, zu zerteilen und dabei die bereits vorhandenen Informationen auf alle neuen Teilstücke zu übertragen. Die Funktion ist damit das digitale Pendant zum physischen Zersägen von Proben in mehrere

Teilstücke. Wird beispielsweise Probe WR-300 nach der Polymerisation in zwei Teile zersägt, um diese anschließend unterschiedlich weiter zu prozessieren, kann diese Probe auch in der Datenbank auf einfache Weise geteilt werden. Dadurch entstehen automatisch die Proben WR-300-A und WR-300-B, wobei die Probe WR-300 verworfen wird. Die beiden neuen Proben enthalten jedoch alle Informationen, welche die Probe WR-300 bis zu diesem Prozessschritt besaß. Natürlich sind auch andere Teilstück-Anzahlen möglich, sowie die Auswahl, ob ein Reststück übrig bleibt oder nicht. Die zugehörige Eingabemaske ist in Abbildung 32 dargestellt.

Abbildung 32: Abschnitt Probe teilen.

3.2.4 Dateien hochladen

Die Möglichkeit, Dateien automatisch hochzuladen und einlesen lassen zu können, stellt eine Kernaufgabe des Web-Interfaces dar, da hier das größte Zeit-Einsparungspotenzial vorliegt. Zu den automatisch einlesbaren Dateiformaten gehören Excel- oder CSV-Dateien eines bestimmten Aufbaus, welche beispielsweise Pyrolyse-Kurven, Dichte- und Porositätsmessungen oder Bauteilvermessungen enthalten. Abbildung 33 zeigt, welche Daten nach welchem Prozessschritt hochgeladen werden können. Wird beispielsweise eine Dichte- und Porositätsmessung hochgeladen, welche die Proben HP-1, HP-2 und HP-3 enthält, werden alle Messdaten aufgenommen und automatisch bei den jeweiligen Proben in der Datenbank einsortiert. Dabei erfolgt zusätzlich im Hintergrund eine Korrektur der Probennamen, um häufige Flüchtigkeitsfehler abzufangen (Groß- und Kleinschreibung, Bindestriche, Unterstriche, Leerzeichen, Nummerierungen statt Buchstaben bei Teilstücken...). Ein Beispiel für eine Eingabemaske des Abschnitts Polymerisation ist in Abbildung 34 gezeigt.



Abbildung 33: Abschnitt Dateien hochladen des programmierten Web-Interfaces.

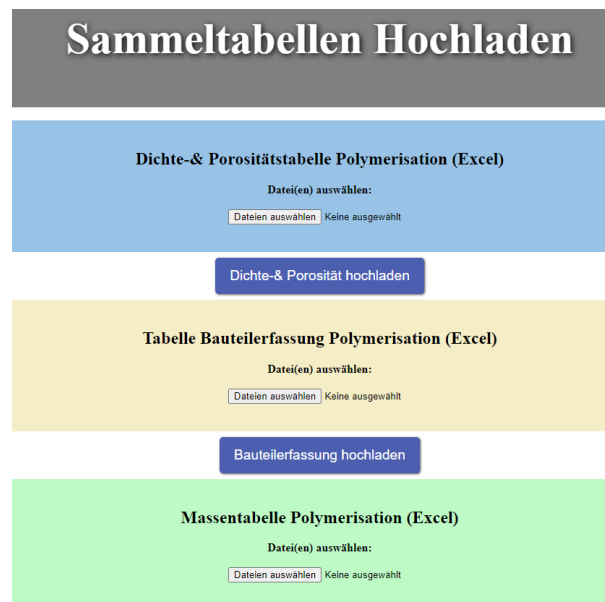


Abbildung 34: Sektion Polymerisation aus dem Abschnitt Dateien hochladen.

3.2.5 REM-Bilder hochladen

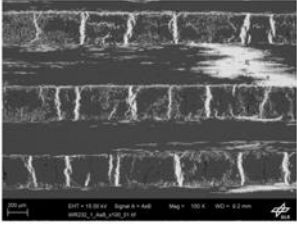
Ein weiterer, besonders wichtiger Abschnitt des Web-Interfaces, ist die Aufnahme von REM-Dateien. Dieser Arbeitsschritt besitzt absichtlich nur eine halb-automatische Bedienung, um eine bessere Kontrolle durch den Benutzer zu gewährleisten. Wurde das gewünschte Bild hochgeladen, werden die einzelnen Phasen durch eine automatische Grauerkennung voneinander getrennt und gemäß Abbildung 35 angezeigt. Außerdem wird aus jedem Bild eine spezifische Kennzahl berechnet, welche später als Zielvariable für die KI-Modelle verwendet wird. Eine ausführlichere Beschreibung der Bildererkennung wird in Kapitel 3.3 gegeben.

Segmentierung des Bildes

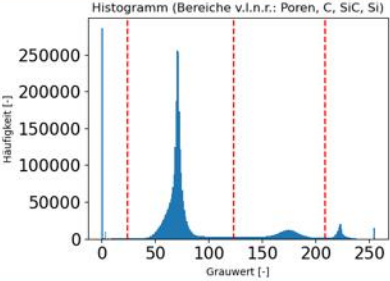
Aktuelle Probe: **VORVERSUCH-1**
 Projektleiter: **t**

Schwellen-Grauwerte werden automatisch berechnet. Manuelles Eingreifen ist über die drei Eingabefelder möglich.

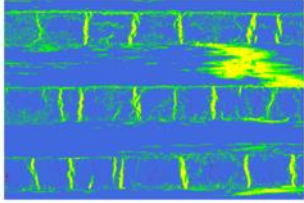
Original



Histogramm (Bereiche v.l.n.r.: Poren, C, SiC, Si)



Multi-Otsu Segmentierung



[Histogramm herunterladen](#) [Phasenanteile herunterladen](#)

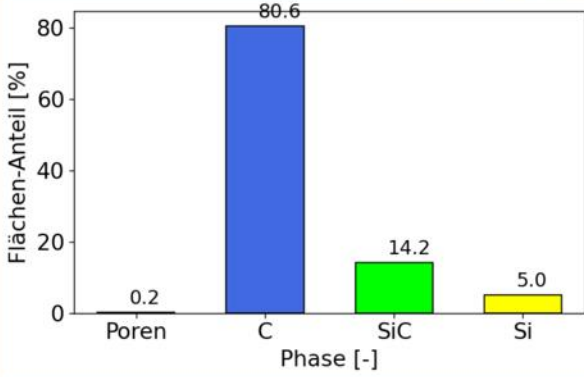
Die Grauwert-Schwellen, werden im Histogramm als rote Linien dargestellt.

Untere Schwelle vorgeben:
 Mittlere Schwelle vorgeben:
 Obere Schwelle vorgeben:

[Grauwert manuell festlegen](#)

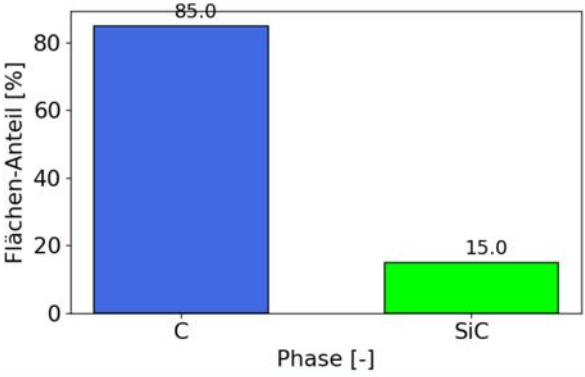
► Optional: Bild manuell zuschneiden (hier klicken)

Zusammenfassung (links: alle Phasen, rechts: nur C und SiC betrachtet):



Flächen-Anteil [%]

Phase [-]



Flächen-Anteil [%]

Phase [-]

[Linkes Diagramm als Excel-Tabelle herunterladen](#) [Rechtes Diagramm als Excel-Tabelle herunterladen](#)

Vorläufige Bewertungszahl: 9.2

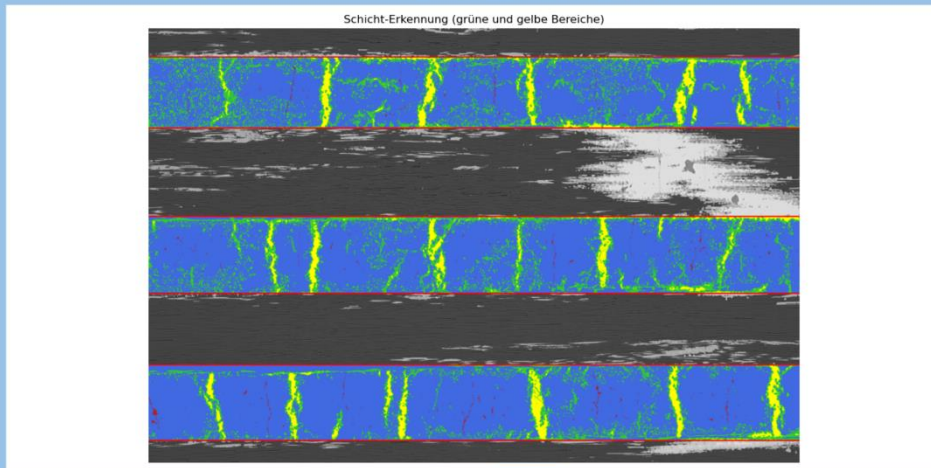
Vorläufige CCR: 8.4 %

[Speichern & weiter zur Lagen-Erkennung](#)

Abbildung 35: Abschnitt REM-Bilder hochladen des programmierten Web-Interface.

Weiterhin kann auch eine automatische Schichterkennung erfolgen, wie sie in Abbildung 36 gezeigt ist, um Längs- von Querlagen zu trennen. Diese wird ebenfalls in Kapitel 3.3 näher diskutiert und funktioniert derzeit nur für horizontale Schichtaufbauten. Die Schichterkennung hat einen geringen Einfluss auf die bestimmten Phasenanteile.

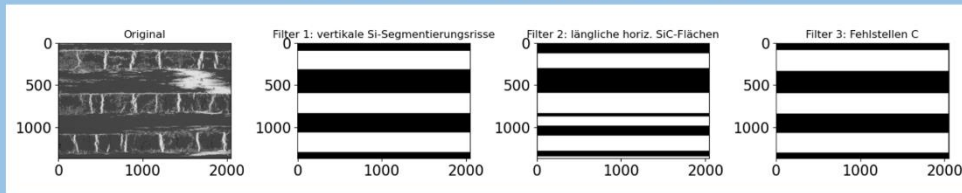
Querschnitte werden automatisch erkannt und von der Bewertung ausgeschlossen (graue Bereiche). Farbige Bereiche werden zur Berechnung der Bewertungszahl und der CCR herangezogen. Bei schlechter Erkennung kann entweder manuell in die Schichterkennung eingegriffen werden (Bereich durch Klicken ausklappen) oder es können stattdessen die vorläufigen Ergebnisse (letzte Seite, Ergebnisse aus gesamtem Bild) gespeichert werden.



Sollte die Schichterkennung keine akzeptablen Ergebnisse geliefert haben, kann hier manuell eingegriffen werden.

Die folgenden Bilder zeigen 3 berechnete Filter, von denen einer verwendet wurde, um die quer verlaufenden Schichten herauszufiltern. Sie sind jeweils genauso groß wie die Originalaufnahme und werden wie eine Schablone über das Originalbild gelegt. Schwarze Bereiche werden von der Bewertung ausgeschlossen, weiße Bereiche werden beibehalten (und erscheinen farbig im oberen Bild). Durch Aktivieren der jeweiligen Checkbox kann manuell ein anderer Filter oder eine Kombination aus Filtern ausgewählt werden.

Aktuell genutzter Filter: Keine Manuelle Vorgabe. Der Algorithmus hat sich für Filter 1 mit der Glättung über 30 px entschieden.



Folgende Filter verwenden (Mehrfachauswahl möglich):

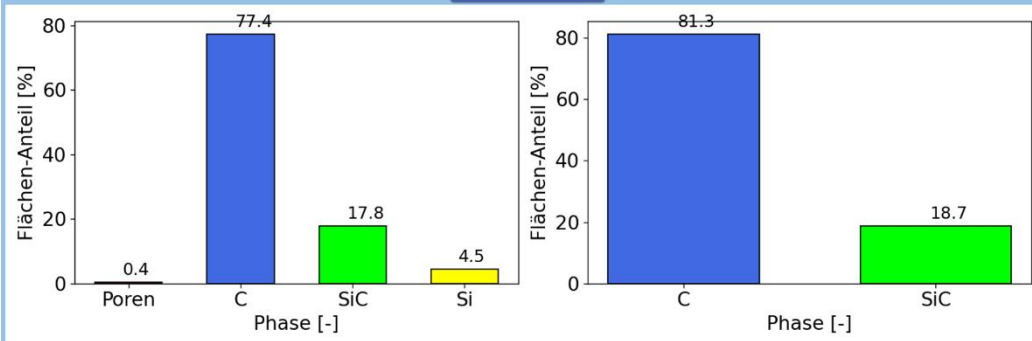
Filter 1 anwenden: Filter 2 anwenden: Filter 3 anwenden:

Eine weitere Optimierungsmöglichkeit ist über die Glättung der Filter gegeben. Ein höherer Wert filtert zu weniger aber größeren weißen Flächen im Filter. Ein geringerer Wert filtert zu mehr aber evtl. auch schmalere weißen Flächen im Filter. Die Glättung kann für jeden Filter im Bereich von 1-500 Pixel vergeben werden.

Folgende Glättung verwenden:

Glättung Filter 1: Glättung Filter 2: Glättung Filter 3:

Filter manuell festlegen



Linkes Diagramm als Excel-Tabelle herunterladen

Rechtes Diagramm als Excel-Tabelle herunterladen

Bewertungszahl: 8.6

CCR: 10.7 %

Falls die Schichterkennung ein akzeptables Ergebnis geliefert hat:

Ergebnisse der Schichterkennung speichern

Falls die Schichterkennung kein akzeptables Ergebnis geliefert hat, können stattdessen die Ergebnisse des ganzen Bildes (vorherige Seite) verwendet werden:

Stattdessen vorläufige Ergebnisse speichern

Abbildung 36: Sektion Schichterkennung des Abschnitts REM-Bilder hochladen.

3.2.6 Auftragsstatus anzeigen

Diese Funktion gibt dem User eine Übersicht darüber, welche Proben sich aktuell in welchem Prozessschritt befinden. Dabei ist für jeden Prozessschritt eine Tabelle angelegt, in der Proben automatisch angezeigt werden, sobald ein Projektleiter einen Arbeitsschritt anfordert. Außerdem kann direkt erkannt werden, welche Bearbeitungsschritte innerhalb dieses Prozessschrittes bereits erledigt wurden und welche noch ausstehend sind. Technische Mitarbeiter können die Arbeitsschritte durch Hochladen von Dateien oder manuelle Eingabe abschließen, was anschließend auch farblich markiert wird (Ampelprinzip). Wurden alle Arbeiten einer Probe zu einem Prozessschritt erledigt, verschwindet diese automatisch aus der aktuellen Tabelle und wird stattdessen im nächsten Prozessschritt angezeigt, sofern dieser angefordert wurde. Auf diese Weise können Verzögerungen in der Produktion durch unübersichtliches Probenmanagement vermieden werden. Ein Beispiel für eine Tabelle des Bereichs Auftragsstatus ist in Abbildung 37 gezeigt.

Offene Aufträge

Legende: ■ Nicht angefordert, ■ Ausstehend, ■ Abgeschlossen

▼ Aufbau / Wareneingang			
Proben-ID	Herstellung	Dichte- & Porositätsmessung	Bauteilvermessung
WR-204	Ja	Nein	Nein
I-907	Ja	Nein	Ja
I-685	Ja	Nein	Nein

► Polymerisation

► Temperung

Abbildung 37: Abschnitt Auftragsstatus des programmierten Web-Interface.

3.2.7 Daten anzeigen

Diese Funktion zielt darauf ab, den aktuellen Prozessstatus einer ausgewählten Probe sowie ihre detaillierte Historie möglichst übersichtlich darzustellen. Über das Web-Interface können die eingetragenen Daten zu gewünschten Proben und Prozessschritten angezeigt werden.

Digitaler Laufzettel

Aktuelle Probe:
 Projektleiter:

Aufbau

Geometrie und Maße

Geometrie: Geometrie Sonstiges:
 Sollmaß C/C-SiC Länge [mm]: Sollmaß C/C-SiC Innendurchmesser [mm]: Sollmaß C/C-SiC Außendurchmesser [mm]:
 Sollmaß CFK Länge [mm]: Sollmaß CFK Innendurchmesser [mm]: Sollmaß CFK Außendurchmesser [mm]:

Komponenten

Faservolumengehalt [%]: Precursor: Beschreibung Precursor:
 Additive verwendet: Additive Beschreibung: Viskosität Harz [m²/s]:
 Fasermaterial: Beschreibung Fasermaterial: Beschreibung Faserart:
 Filamentanzahl: Faserdichte [g/cm³]: Flächengewicht [g/m²]:
 Style: Webart: Schlichte Precursor:
 Beschreibung Schlichte: Vorbehandlung: Orientierung:
 Beschreibung Orientierung: Sonstige Bemerkungen:

Abbildung 38: Daten des Abschnitts „Aufbau“ des digitalen Laufzettels am Beispiel der Probe WR-151.

3.2.8 Daten löschen

In diesem Abschnitt können bereits vorhandene Daten aus der Datenbank gelöscht oder korrigiert werden. Je nach Auswahl des Benutzers öffnen sich verschiedene Optionen für eine Bearbeitung der Daten, wie Abbildung 39 zeigt.

Dateien korrigieren oder löschen

Aktuelle Probe:
 Projektleiter:

Aktion wählen:

Probe komplett aus Datenbank löschen
 Nur Daten aus bestimmter Tabelle löschen
 Projekt & KTR aktualisieren

Daten der ausgewählten Probe aus folgender Tabelle löschen:

Ausgewählte Anfrage ausführen

[Hauptmenü](#)

Abbildung 39: Abschnitt Daten löschen des programmierten Web-Interface.

3.3 Bilderkennung

Besonders wichtig ist auch das Einlesen und Auswerten von Rasterelektronenmikroskop-Aufnahmen (REM-Aufnahmen), denn die hieraus gewonnenen Daten werden später als Zielgröße für das Anlernen der KI-Modelle verwendet. Dafür ist es notwendig, dass die Aufnahmen nicht nur als Bilddatei gespeichert, sondern auch quantitative Aussagen daraus abgeleitet werden. Zum einen sollen die vier prozentualen Phasenanteile von C/C-SiC erfasst werden. Dazu gehören Poren, Kohlenstoff, Siliziumkarbid und Silizium. Zum anderen wurde

für jedes Bild eine spezifische Kennzahl berechnet, auf die im nächsten Unterkapitel genauer eingegangen wird.

3.3.1 Definition der CCR

Der Kohlenstoffkonvertierungsgrad, oder auch Carbon Conversion Ratio (CCR), stellt ein quantifizierbares Maß für die Güte einer Mikrostruktur dar. Sie beschreibt, wie viel Prozent des Kohlenstoffs während der Silizierung zu Siliziumcarbid konvertiert wurde, und bezieht dabei auch die molare Volumenänderung der beiden Stoffe ein. Die CCR berechnet sich nach Gleichung (27).

$$CCR = \frac{A_{SiC} \cdot K}{A_C + A_{SiC} \cdot K} \quad (27)$$

Dabei beschreibt A_{SiC} die betrachtete Siliziumcarbid-Fläche, A_C die betrachtete Kohlenstoff-Fläche und K das molare Volumenverhältnis von Kohlenstoff und Stickstoff, wobei gilt:

$$K = \frac{V_C}{V_{SiC}} = \frac{6,53 \frac{cm^3}{mol}}{12,45 \frac{cm^3}{mol}} = 0,52$$

Die CCR ist damit (nicht-linear) proportional zu A_{SiC} und (nicht-linear) indirekt proportional zu A_C . Der Verlauf der CCR ist graphisch in Abbildung 40 dargestellt.

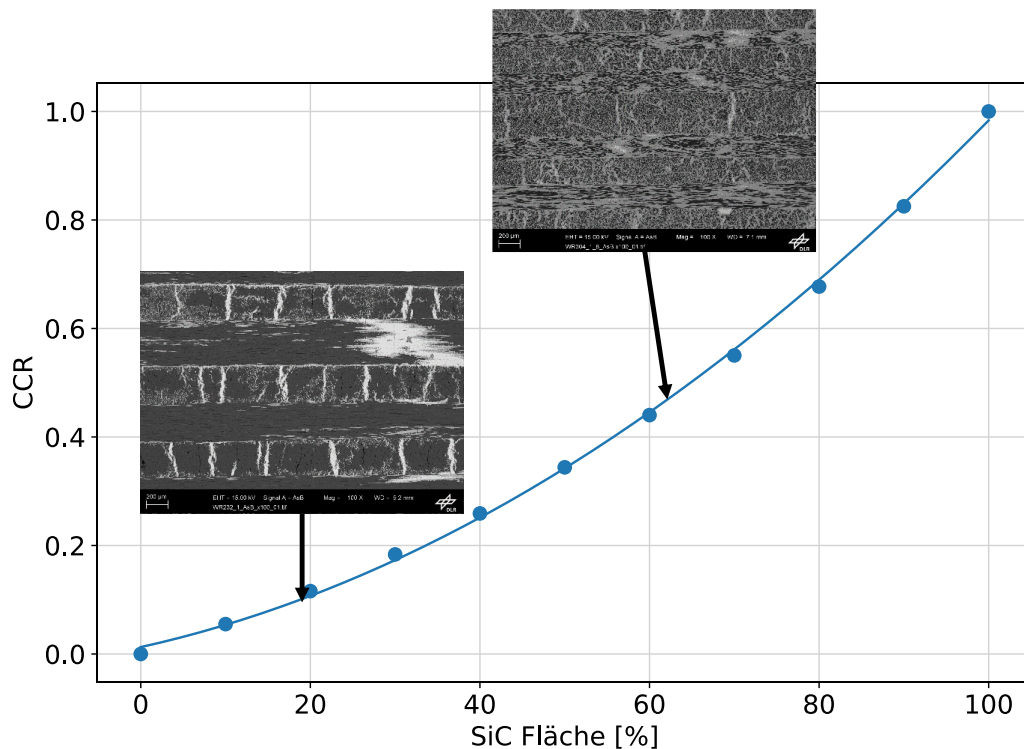


Abbildung 40: Visuelle Darstellung der CCR mit zwei Beispiel-Mikrostrukturen.

3.3.2 Phasenunterscheidung

Um die CCR eines Bildes zu berechnen, ist es notwendig, die vier Phasen Poren, Kohlenstoff, Siliziumkarbid und Silizium automatisch zu erkennen. Dabei wird nach dem folgenden Schema vorgegangen:

1. Erstellen eines Histogramms des Bildes: Häufigkeit über Grauwert der Pixel
2. Finden der Maxima im Histogramm und Ermittlung der Mitten zwischen den Maxima
3. Einteilung in Klassen basierend auf Maxima und Einfärbung der Pixel

In Abbildung 41 ist dafür beispielhaft eine REM-Aufnahme der Probe WR-232 dargestellt, von deren Mikrostruktur die CCR berechnet werden soll.

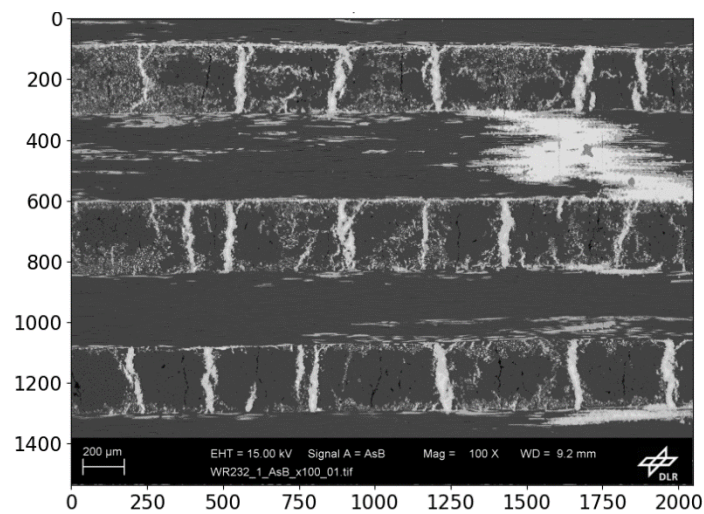


Abbildung 41: Originalbild der Probe WR-232.

Zunächst wird von der automatischen Auswertung ein Grauwert histogramm erstellt (siehe Abbildung 42) und anhand der Pixelanhäufungen die Grenzen für eine Unterteilung in die vier Phasen berechnet (rot gestrichelte Linien in Abbildung 42).

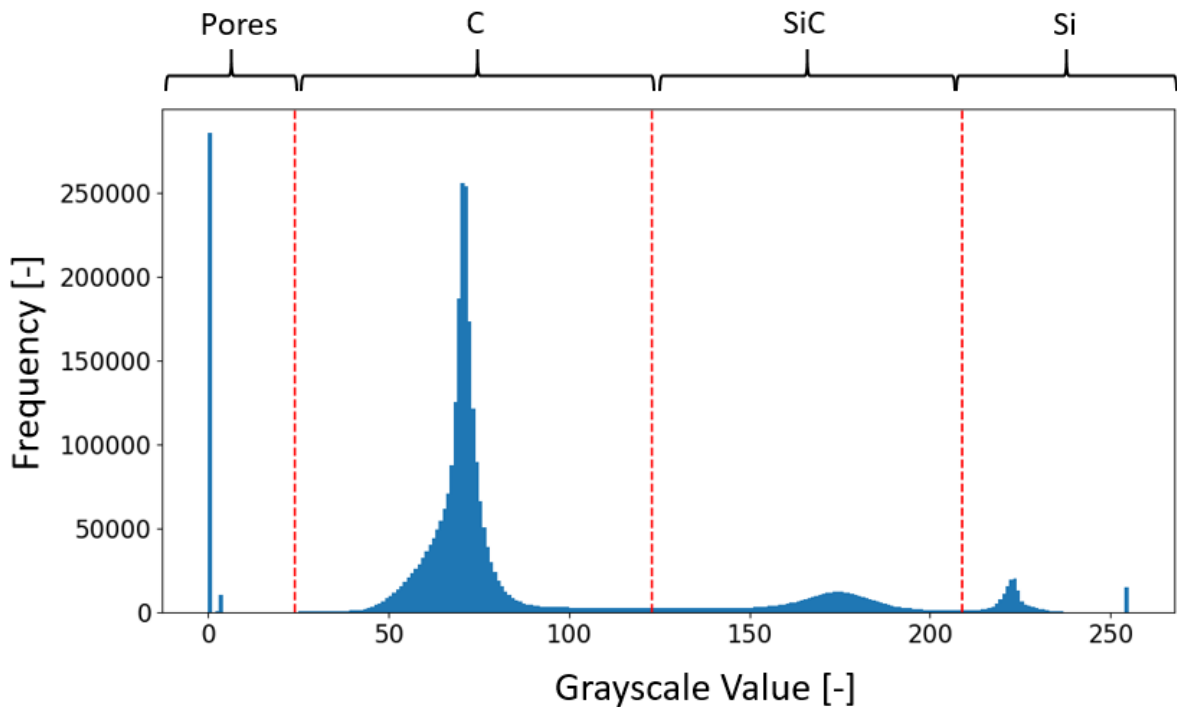


Abbildung 42: Histogramm der Grauwerte innerhalb des Bildes der Probe WR-232 mit automatisch bestimmten Schwellen für die unterschiedlichen Phasen (rot gestrichelt).

Nun können die prozentualen Anteile der Phasen berechnet werden. Letzteres wird auch visuell durch ein eingefärbtes REM-Bild ausgegeben (siehe Abbildung 43). Obwohl die automatische Berechnung der Grauwert-Schwellen durch den Algorithmus in den meisten Fällen gute Ergebnisse liefert, wird für schlechte Erkennungen eine manuelle Vorgabe durch den Benutzer ermöglicht.

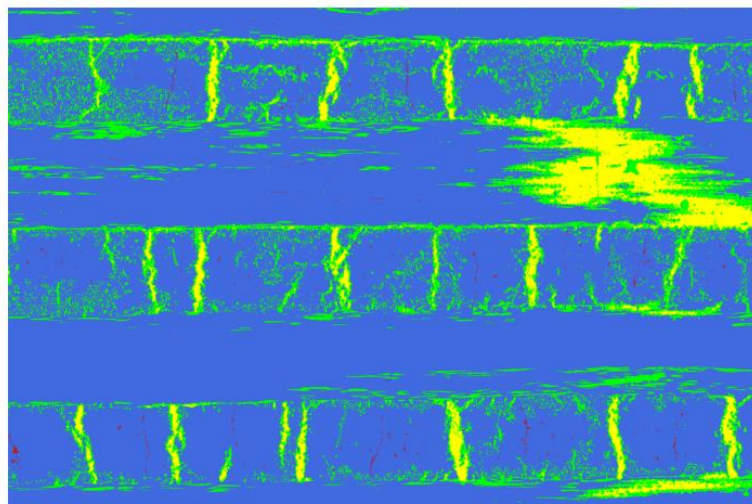


Abbildung 43: Beispielhaftes Ergebnis der automatischen Segmentierung, Blau: Kohlenstoff, Rot: Poren, Gelb: Silizium, Grün: Siliziumkarbid.

3.3.3 Lagenerkennung

Unabhängig von der Bestimmung der CCR des Gesamtbildes kann auf Wunsch des Benutzers auch die CCR nur in denjenigen Schichten der Probe berechnet werden, bei denen

die Fasern in die Bildebene hinein verlaufen („Längslagen“). Dies kann insofern nützlich sein, als dass die Phasenzusammensetzung der Lagen, bei denen die Fasern in der Bildebene verlaufen („Querlagen“) durch ungünstiges Anschneiden von Segmentierungsrissen verfälscht werden kann. Die Lagendefinition ist noch einmal in Abbildung 44 verdeutlicht. Allerdings funktioniert die Lagenerkennung zum aktuellen Stand der Arbeit nur für Proben, deren Schichten relativ horizontal verlaufen und wenig Ondulation aufweisen.

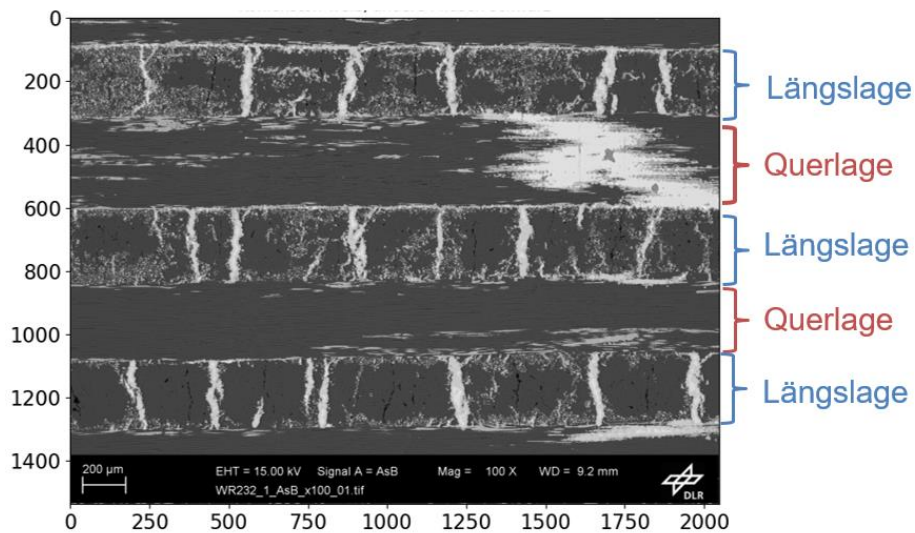


Abbildung 44: Definition der Längs- und Querlagen in einem REM-Bild.

Ist ein REM-Bild in seine vier Phasen segmentiert (vgl. Abbildung 43), können in einem weiteren Schritt die Querlagen herausgefiltert werden. Dies ist nicht zwingend notwendig, verbessert aber die Genauigkeit der berechneten CCR geringfügig, da Querlagen tendenziell weniger Siliziumkarbidanteile zeigen, als Längslagen. Letztere verfälschen somit die berechneten Phasenanteile, wenn das gesamte REM-Bild zur Berechnung der Kennwerte herangezogen wird. In der Praxis erweist sich dieser Unterschied aber als gering, wobei er von Bild zu Bild variiert und deswegen nur bedingt in Zahlenwerte gefasst werden kann. Sollte die Schichterkennung keine befriedigenden Ergebnisse liefern, kann auch manuell eingegriffen, und durch unterschiedliche Wahl von Filtern eine Optimierung erreicht werden. Sollte auch dies die Erkennung nicht verbessern, kann entschieden werden, stattdessen die Ergebnisse aus dem gesamten Bild zu verwenden. Generell ist das Unterteilen in Quer- und Längslagen aufgrund der Vielfalt der unterschiedlichen Mikrostrukturen eine Herausforderung. Die besten Ergebnisse lieferte hier ein flexibles Verfahren, welches je nach Merkmalen des segmentierten Bildes eine andere Berechnungsrouten durchläuft. Das Verfahren wird im Folgenden genauer beschrieben.

Längslagen können im Falle einer XB-ähnlichen Mikrostruktur gut über ihre vertikal verlaufenden Si-Segmentierungsrisse erkannt werden. In Querlagen kommen diese Formen

nicht vor. In XD-ähnlichen Mikrostrukturen funktioniert diese Methode aber nicht mehr, da hier kaum noch elementares Silizium vorliegt. Hier sind die Segmentierungsrisse allerdings häufig durch die Abwesenheit von Kohlenstofffasern in diesen Bereichen zu erkennen (ehemals Risse). Eine weitere Möglichkeit der Unterteilung bietet die Suche nach horizontal verlaufenden, schmalen Siliziumkarbid-Flächen einer bestimmten Größe, wie sie nur in Querlagen auftreten. Je nach Art der Mikrostruktur kann eine der drei Methoden – oder auch eine Kombination mehrerer Methoden – zielführend sein.

Jeder der drei möglichen Wege wird durch den Algorithmus verfolgt und ein Filter daraus generiert. Dabei ist ein Filter eine mathematische Matrix von der Größe der REM-Aufnahme (Breite in Pixeln x Höhe in Pixeln), welche den Wert 1 enthält, wenn der Pixel zu einer Längslage gehört und den Wert 0, wenn ein Pixel zu einer Querlage gehört. Bildlich dargestellt ist ein solcher Filter in Abbildung 45.

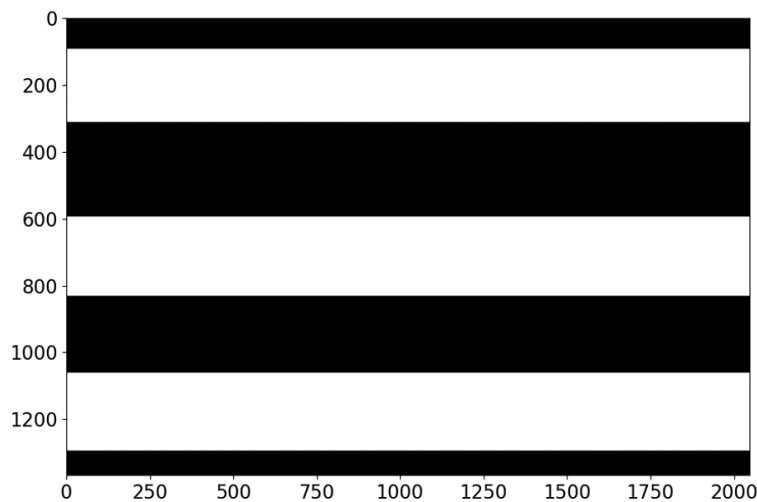


Abbildung 45: Filter zur Erkennung von Längslagen, weiß: Pixel gehört zu einer Längslage, schwarz: Pixel gehört zu einer Querlage.

Legt man einen solchen Filter wie eine Schablone über das Originalbild und behält nur die weißen Bereiche, erhält man alle Längslagen des Bildes. Dies ist, bildlich gesprochen, auch das Vorgehen des Algorithmus.

Im Folgenden wird nun auf die Erstellung der Filtermasken eingegangen. Dazu werden die oben genannten Merkmale im Bild erkannt. Am Beispiel der Silizium-Segmentierungsrisse wird zunächst ein Schwarz-Weiß-Bild erzeugt, welches nur elementares Silizium enthält, wie es in Abbildung 46 dargestellt ist. Dies ist möglich, da zuvor die verschiedenen Phasen des Bildes klassifiziert wurden (Abbildung 43).

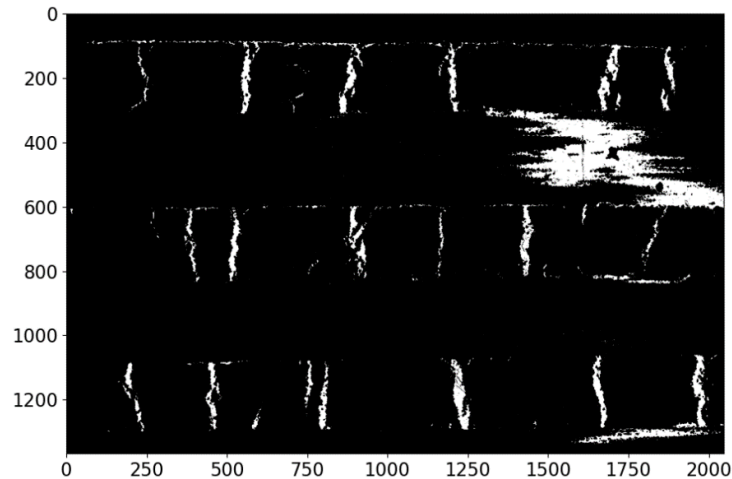


Abbildung 46: Elementares Silizium in weiß, alle anderen Phasen in schwarz.

Auffällig zu sehen sind die vertikalen Segmentierungsrisse. Weiterhin ist aber auch ein großer angeschnittener Segmentierungsriß in einer der Querlagen erkennbar (große weiße Fläche), sowie feine horizontale Linien, welche die Grenze zwischen einer Längs- und einer Querlage kennzeichnet. Diese sind bei der Lagenerkennung hinderlich und müssen herausgefiltert werden. Es wird daher nach Flächen eines bestimmten Seitenverhältnisses und einer bestimmten Größenordnung gefiltert, was in Abbildung 47 dargestellt ist.

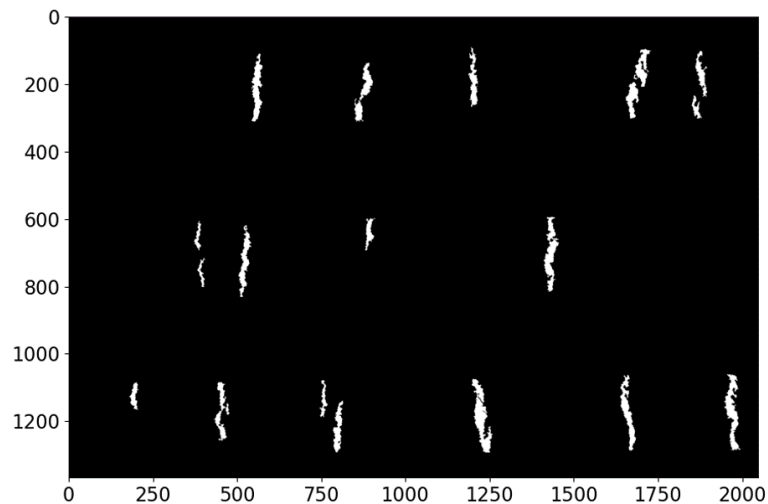


Abbildung 47: Extrahierte Si-Segmentierungsrisse durch mehrmaliges Filtern nach Seitenverhältnis und Größe der gefundenen Konturen.

Auffällig ist, dass Silizium-Segmentierungsrisse dieser Art nur in Längslagen auftreten. Um eine Schablone für Längslagen zu erstellen, müssen also alle Pixel einer Reihe, in der auch Segmentierungsrisse auftreten, weiß eingefärbt werden. Alle Pixel einer Reihe, in der keine Segmentierungsrisse auftreten, bleiben schwarz. Dies führt zu dem in Abbildung 48 gezeigten Filter.

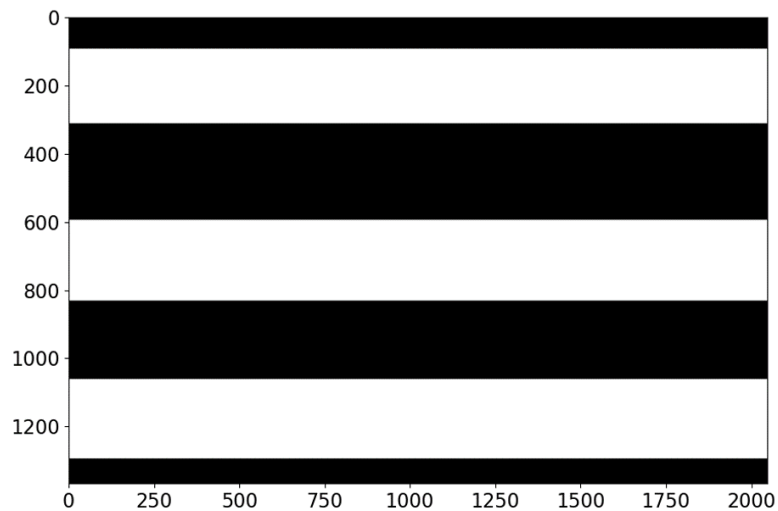


Abbildung 48: Filter mit weißen Streifen an Positionen, an denen Längslagen erkannt wurden (Si-Segmentierungsriss-Methode); schwarze Streifen kennzeichnen Querlagen.

Überlagert man den Filter aus Abbildung 48 mathematisch mit dem Originalbild, wird das Originalbild nur an den Stellen beibehalten, an denen der Filter weiße Pixel aufweist; alle anderen Bereiche werden verworfen. Damit stehen nur noch die Längslagen für die Auswertung der Kennzahlen zur Verfügung. Die erkannten Lagen sind in Abbildung 49 rot umrandet dargestellt. Weiterhin wurde Silizium innerhalb der erkannten Lagen gelb und Siliziumkarbid grün eingefärbt.

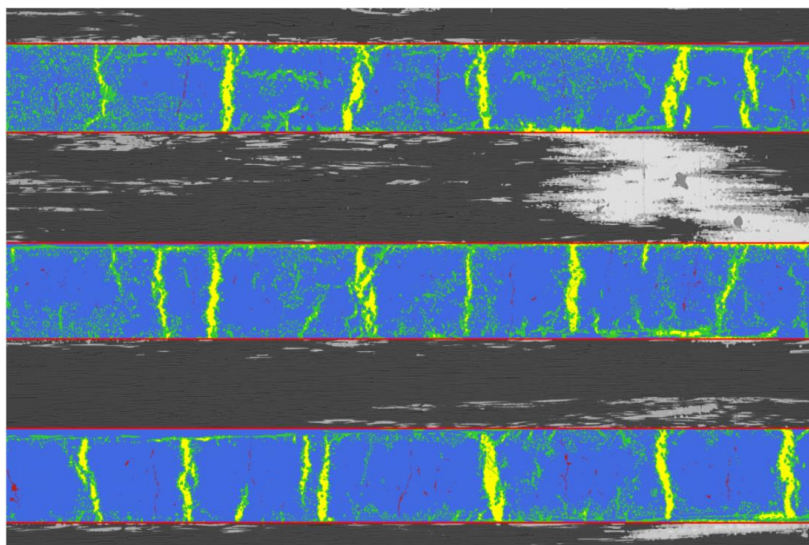


Abbildung 49: Erkannte Längslagen sind rot umrandet dargestellt; darin Silizium: Gelb, Siliziumkarbid: Grün, Poren: Rot und Kohlenstoff: Blau markiert (Si-Segmentierungsriss-Methode).

Ein ähnliches Vorgehen ergibt sich für die Gewinnung von Filtern aus den anderen beiden oben genannten Merkmalen. Das Vorgehen wird nun in verkürzter Form beschrieben.

Bei Werkstoffen, die wenig oder gar kein elementares Silizium enthalten, stößt die Filterung nach Silizium-Segmentierungsrissen an ihre Grenzen. Hier können aber Querlagen durch

längliche horizontal verlaufende Siliziumkarbidflächen identifiziert werden, wie sie in Abbildung 50 dargestellt sind.

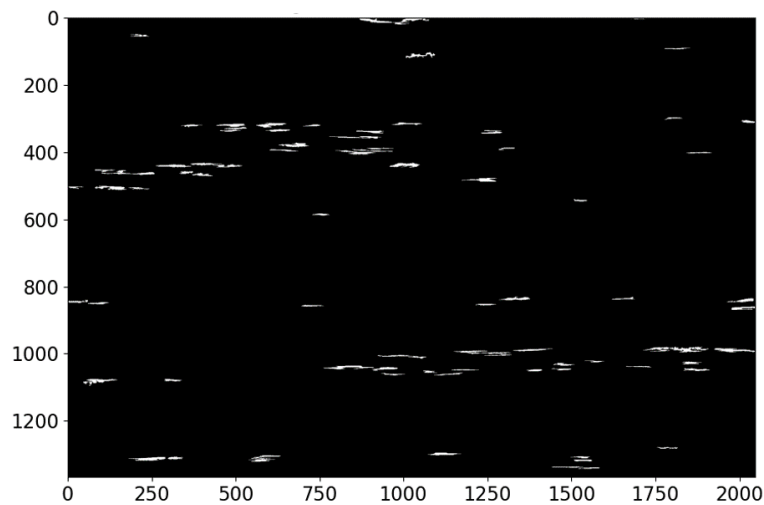


Abbildung 50: Horizontal verlaufende, längliche Flächen aus Siliziumkarbid mit einem Seitenverhältnis von Breite/Länge > 5.

Auch hieraus kann durch analoges Vorgehen ein Filter erstellt werden, wie in Abbildung 51 zu sehen.

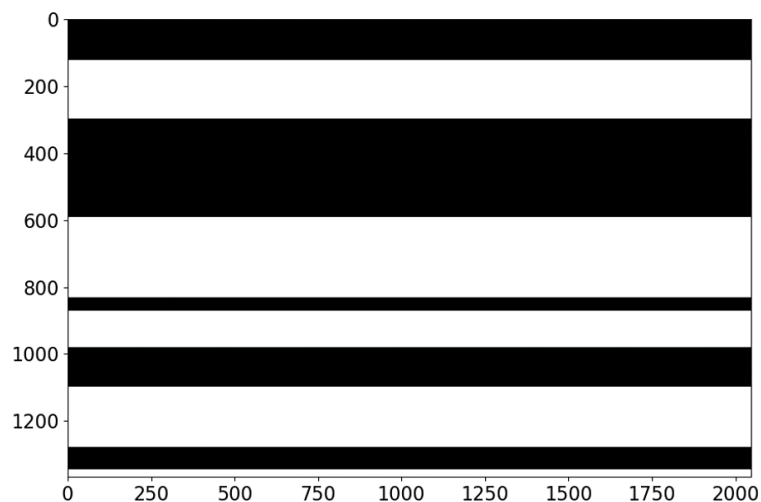


Abbildung 51: Filter auf Grundlage von Siliziumkarbidflächen; weiße Pixel kennzeichnen Bereiche in denen Längslagen vermutet werden, schwarze Pixel kennzeichnen Bereiche in denen Querlagen vermutet werden.

Deutlich erkennbar ist, dass die Längslagenerkennung in diesem Beispiel durch den Siliziumkarbid-Filter schlechter funktioniert, als durch den Silizium-Filter, was aber je nach Bild variiert.

Eine dritte Möglichkeit der Erkennung bietet die Suche nach Fehlstellen im Kohlenstoff in den Längslagen. Betrachtet man nur den Kohlenstoff in der Aufnahme (Abbildung 52), lassen sich die Segmentierungsrisse gut als Fehlstellen erkennen. Im Gegensatz zur Klassifizierung nach Silizium-Segmentierungsrisse bestehen diese Fehlstellen nicht zwangsläufig nur aus Silizium, sondern können auch mit Siliziumkarbid oder Poren gefüllt sein.

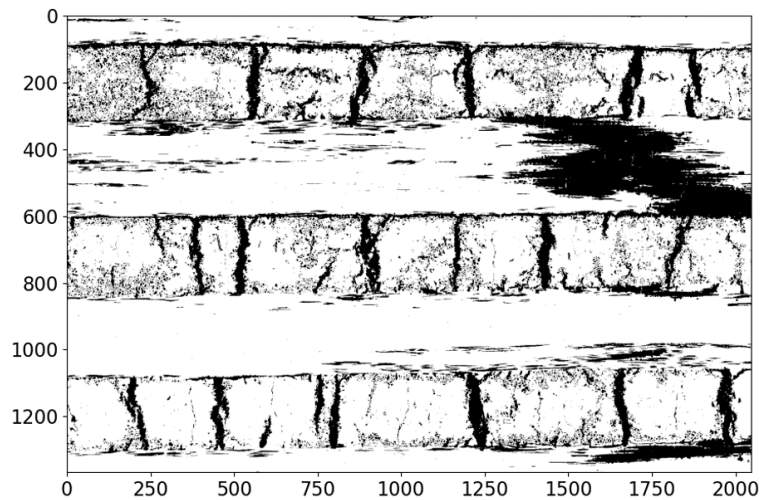


Abbildung 52: Kohlenstoff weiß, alle anderen Phasen schwarz eingefärbt.

Invertiert man die Farben in Abbildung 52, und führt einige glättende Berechnungen durch, ergibt sich wieder ein ähnliches Bild wie in Abbildung 46. Analog müssen weitere Filterschritte erfolgen, welche nur Flächen mit bestimmten Seitenverhältnissen und Größen zulassen, wodurch sich Abbildung 53 ergibt.

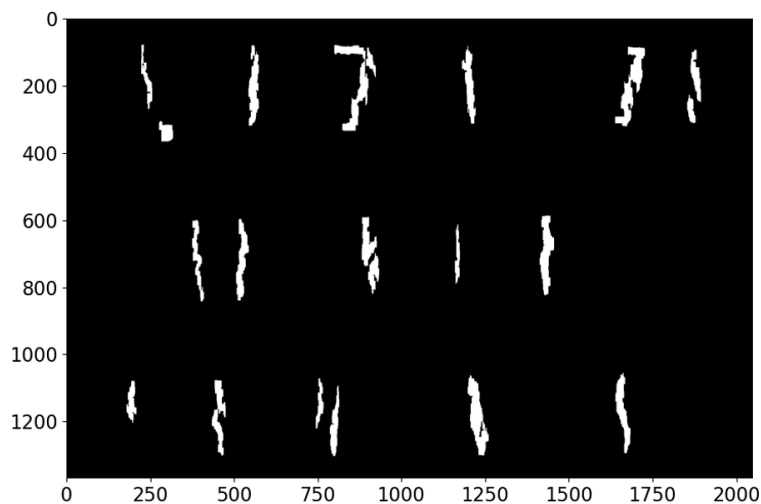


Abbildung 53: Negativ der Kohlenstoff-Fehlstellen gefiltert nach Flächen mit gewissen Seitenverhältnissen und Größen.

Aus Abbildung 53 lässt sich durch analoges Vorgehen wieder ein Filter erstellen, welcher Längslagen prognostiziert und in Abbildung 54 dargestellt ist.

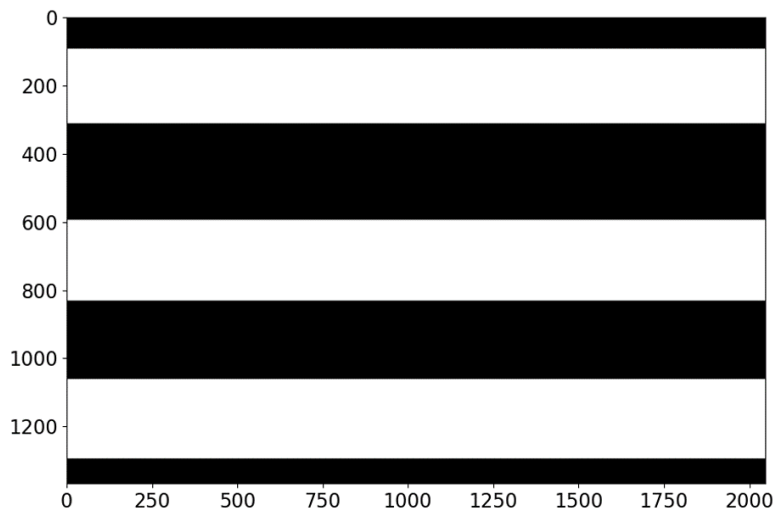


Abbildung 54: Längslagen-Filter durch Kohlenstoff-Fehlstellen; weiße Pixel kennzeichnen Bereiche, an denen im Originalbild Längslagen vermutet werden, schwarze Pixel kennzeichnen Bereiche, in denen Querlagen vermutet werden.

Nachdem die drei verschiedenen Filter für jedes Bild berechnet wurden, muss entschieden werden, welcher Filter (oder welche Filterkombination) zum Einsatz kommen soll. Dies kann aufgrund der Phasenanteile des Gesamtbildes (mit allen Lagen) abgeschätzt werden. Im Programmcode wurde bei der Auswahl der Unterteilungsmethode je nach Kohlenstoffanteil im Gesamtbild daher auf drei verschiedene Routen zurückgegriffen. Dies ist in Tabelle 12 dargestellt. Lag der Kohlenstoffgehalt im Gesamtbild über 75%, wurde dem REM-Bild die Kategorie 1 zugeordnet. Analog wurde dem Bild die Kategorie 2 zugeordnet, wenn der Kohlenstoffgehalt zwischen 75% und 60% lag, die Kategorie 3 wurde vergeben, wenn der Kohlenstoffgehalt im Gesamtbild unter 60% lag. Je nach Kategorie wurden die Filter in der in Tabelle 12 festgelegten Reihenfolge angewendet.

Tabelle 12: Vorgehen des Algorithmus bei der Unterteilung in Längs- und Querlagen, QF = Querlagenfilter, LF = Längslagenfilter.

Priorität	Kategorie 1 C-Gehalt $\geq 75\%$	Kategorie 2 $75\% > \text{C-Gehalt} \geq 60\%$	Kategorie 3 C-Gehalt $< 60\%$
1	LF nach Si-Rissen	QF nach SiC-Flächen	LF nach C-Flächen
2	QF nach SiC-Flächen	LF nach Si-Rissen	QF nach SiC-Flächen
3	LF nach C-Flächen	LF nach C-Flächen	LF nach Si-Rissen
4	Gebe Fehlermeldung	Gebe Fehlermeldung	Gebe Fehlermeldung

Eine geeignete Einteilung der Klassen nach Kohlenstoffgehalt wurde aufgrund von „Trial-and-Error“ unter Verwendung mehrerer REM-Aufnahmen getroffen. Da auch die optimal auf

die Gesamtheit aller Bilder abgestimmte Unterteilungsmethode im Einzelfall ungewollte Ergebnisse liefern kann, wurde auch hier eine Möglichkeit gegeben, manuell einzugreifen.

3.3.4 CCR mit und ohne Lagenerkennung

Der Einfluss der Lagenerkennung auf die CCR ist je nach Schwere der Einzelfasersilizierung unterschiedlich groß. Am größten ist der Einfluss bei stark ausgeprägten XB-Mikrostrukturen, verschwindend gering wird er stattdessen bei ausgeprägten XD-Mikrostrukturen. Nachteilig ist die Verwendung der Lagenerkennung aber in keinem Fall, vorausgesetzt, dass die Lagen richtig erkannt wurden, was durch den Ingenieur bewertet werden muss.

Am Beispiel der Probe WR-232 mit einer XB-Mikrostruktur kann der Unterschied zwischen Ein- und Ausschalten der Lagenerkennung gut verdeutlicht werden. Die berechneten Phasenanteile sind in Abbildung 55 gezeigt.

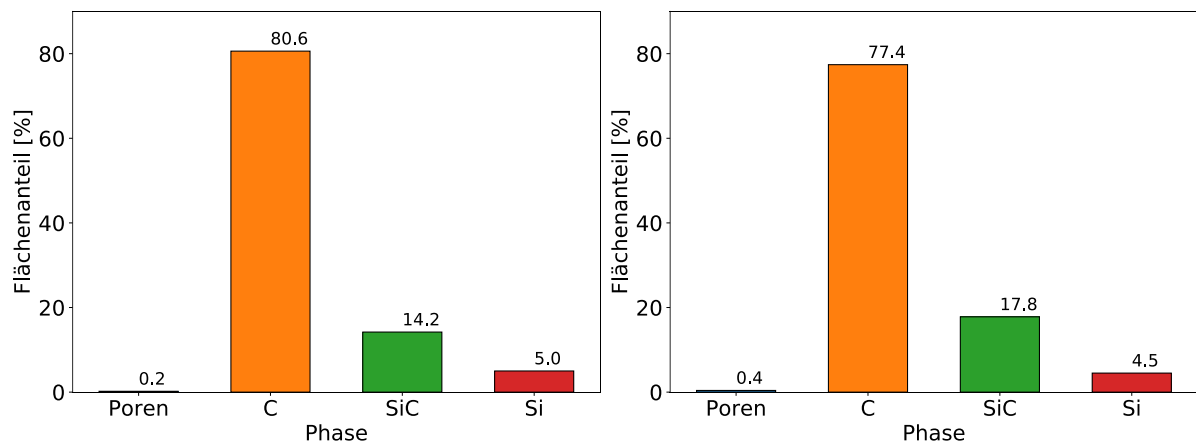


Abbildung 55: Ermittelte Phasenanteile der Probe WR-232; links: ohne Lagenerkennung, rechts: mit Lagenerkennung.

Die berechnete CCR ergibt sich zu $CCR = 8,4\%$ für die Betrachtung ohne Lagenerkennung und zu $CCR = 10,7\%$ für die Betrachtung mit Lagenerkennung.

Für XD-Mikrostrukturen fällt der Unterschied der Phasenzusammensetzung jeweils mit und ohne Lagenerkennung sehr gering aus, wie am Beispiel der Probe WR-304 in Abbildung 56 deutlich wird.

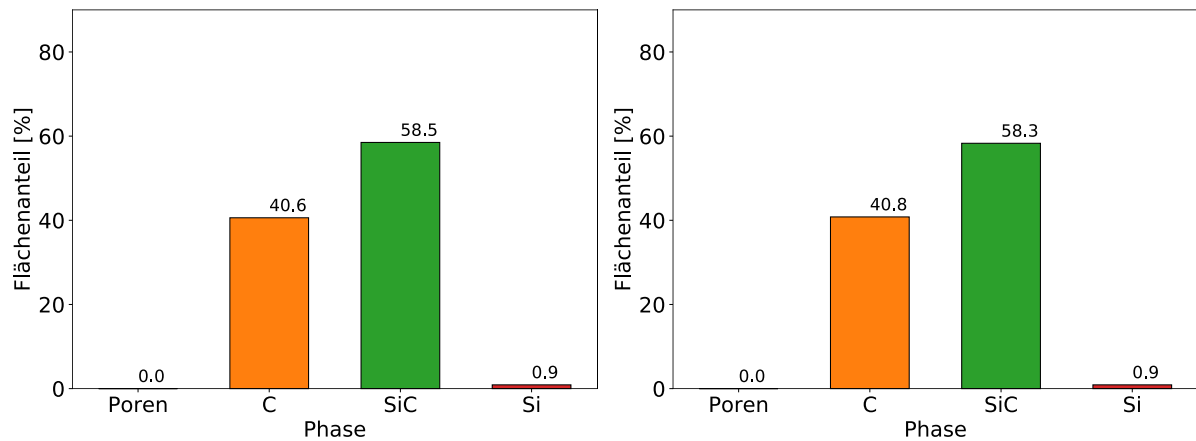


Abbildung 56: Ermittelte Phasenanteile der Probe WR-304; links: ohne Lagenerkennung, rechts: mit Lagenerkennung.

Hier liegt die resultierende CCR im Bild ohne Lagenerkennung bei $CCR = 42,8\%$, und im Bild mit Lagenerkennung bei $CCR = 42,6\%$.

3.3.5 Genauigkeit der CCR

Um eine aussagekräftige CCR zu erhalten, müssen die REM-Aufnahmen einen repräsentativen Anteil der Mikrostruktur enthalten. Die Genauigkeit der CCR hängt damit direkt von der Aussagekraft der zugrundeliegenden Aufnahmen ab. Die Entscheidung darüber, welche Aufnahmen repräsentativ sind, und wie viele davon zur Berechnung der CCR herangezogen werden sollten, liegt beim Ingenieur. Grundsätzlich sollten die Aufnahmen aber folgenden Richtlinien genügen:

- Die CCR sollte nicht allein auf Grundlage eines einzigen Bildes berechnet werden, sondern auf Grundlage mehrerer Bilder, um einen Mittelwert über die Mikrostruktur aus verschiedenen Bereichen zu erhalten
- Die verwendeten Aufnahmen sollten eine möglichst niedrige Vergrößerung (z.B. 100-fache Vergrößerung) verwenden, damit der betrachtete Ausschnitt möglichst groß wird und damit repräsentativer für die gesamte Probe
- Die REM-Aufnahmen sollten keine großen Belichtungsschwankungen innerhalb des Bildes (z. B. von links nach rechts) aufweisen, da hierdurch die Phasenerkennung erschwert wird

Weiterhin wurde untersucht, wie groß der Einfluss des Faserwinkels sowie der Entnahme-Position der Probe auf die gemessene CCR war. Dazu wurden zehn Proben an unterschiedlichen Positionen derselben Platte entnommen, jedoch der Schnittwinkel so variiert, dass die Fasern in unterschiedlichen Winkeln zur Probenoberfläche verlaufen. Die daraufhin erstellten REM-Bilder wurden zur Bildanalyse in das Web-Interface eingegeben, wodurch für jede Probe die

CCR ermittelt werden konnte. Mittelwert und Standardabweichung aus allen Proben ergaben sich zu $CRR = (5,9 \pm 1,0)\%$, was einer verhältnismäßig guten Übereinstimmung trotz unterschiedlicher Faserwinkel und Positionen entspricht, vor allem wenn man bedenkt, dass sich Unterschiede in der CCR aufgrund der Heterogenität des Materials auch bei konstantem Faserwinkel ergeben.

4 Entwicklung des GUI „DataTracker“

In diesem Kapitel wird das Programm „DataTracker“ beschrieben, über welches die Auswertung und Visualisierung der Daten aus der Datenbank erfolgte. Die Unterkapitel befassen sich jeweils mit den verschiedenen Funktionalitäten der App.

4.1 Notwendigkeit und Ziele des Programms

Um die in der PSQL-Datenbank gespeicherten Daten auch für Benutzer ohne Programmiererfahrung zugänglich zu machen, wurde eine graphische Benutzeroberfläche (GUI) in Python programmiert, deren Startmenü in Abbildung 57 zu sehen ist, und die im Folgenden als „DataTracker“ bezeichnet wird. DataTracker wurde in Python programmiert und der Abteilung als Executable (.exe) bereitgestellt, sodass eine Installation auch auf Rechnern ohne Python möglich ist. Als Sicherheitshürde wurde außerdem ein Login-System implementiert und Accounts ausschließlich für Abteilungsmitglieder erstellt.

In der DataTracker-Software können Informationen zu den in der SQL-Datenbank gespeicherten Proben abgefragt und automatisch als tabellarische Daten oder interaktive Diagramme angezeigt werden. Die Funktionalitäten umfassen:

- Anzeigen von Laufzetteldaten (Aufbau, Rohstoffe, Massenänderungen, Dichte- & Porositätsmessungen, Bauteilvermessungen)
- Anzeigen von Polymerisationsdaten (Temperaturkurven, Zyklen-Informationen)
- Anzeigen von Pyrolysedaten (Temperaturkurven, Zyklen-Informationen)
- Anzeigen von Silizierungsdaten (Temperaturkurven, Zyklen-Informationen)
- Anzeigen von Mikrostrukturdaten (Bild-Dateien, CCR-Werte, Phasenzusammensetzungen)
- Erstellen von PDF-Zusammenfassungen zu ausgewählten Proben
- Trainieren von KI-Algorithmen
- Verwendung von trainierten KI-Algorithmen

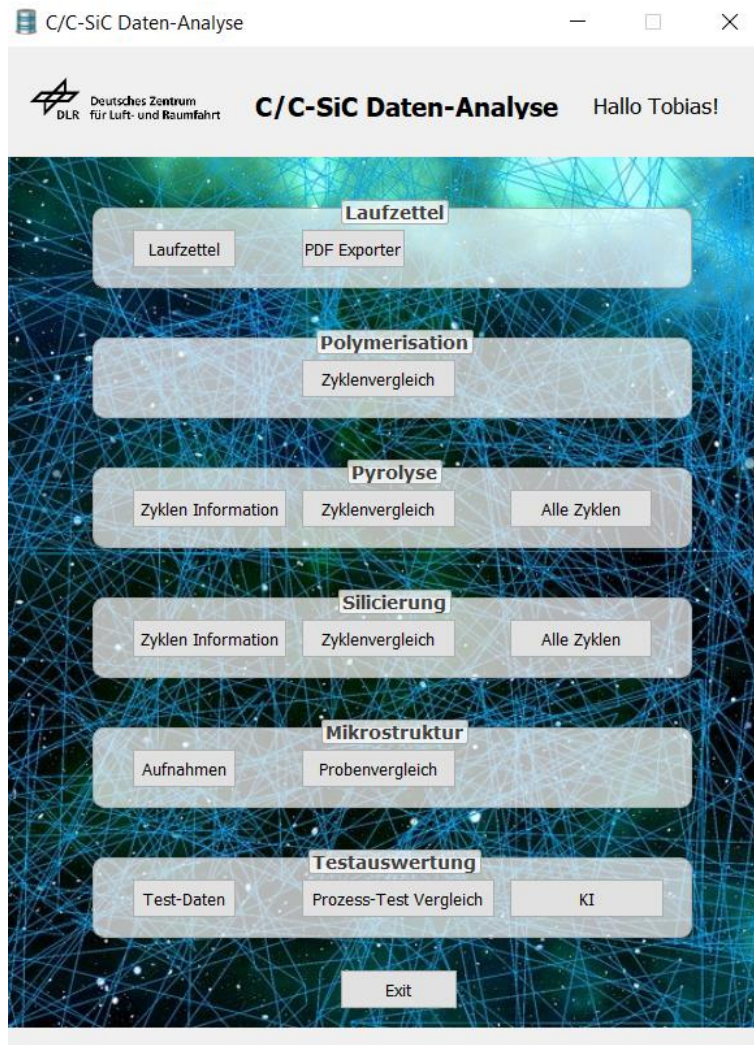


Abbildung 57: Startmenü der entwickelten GUI „DataTracker“.

4.2 Interaktiver Laufzettel

Eine recht naheliegende Funktion von DataTracker ist das Anzeigen von den in der SQL-Datenbank gespeicherten Datensätzen. Dabei handelt es sich in diesem Fall ausschließlich um Tabellendaten, das Erstellen von interaktiven Grafiken wird in späteren Kapiteln behandelt. Durch eine Vorauswahl kann der Suchbereich auf unterschiedliche Weise eingegrenzt werden. Somit können die Daten zu allen Proben, bestimmten Chargen oder einzelnen Proben angezeigt werden, wobei zusätzlich zwischen verschiedenen Prozessschritten und Typen von Messungen unterschieden werden kann. Zudem wurde eine Funktion implementiert, die es ermöglicht, die aktuell in der Tabelle angezeigten Daten als Excel-Datei herunterzuladen. Abbildung 58 zeigt eine beispielhafte Anfrage des interaktiven Laufzettels.

Laufzettel

DLR
C/C-SiC Daten-Analyse
Eingeloggt: Tobias

Laufzettel

Anzeigeeinstellungen

Was soll angezeigt werden? Laufzetteldaten Bauteilvermessung Dichte und Porosität Massentabelle

Prozessschritt vorgeben: Proben-ID:

Hinterlegte Daten

proben_id	projekt	projektleiter	liefertermin	kostentraeger	geometrie	geometrie_sonstiges	sollmass_ccsic_l	sollmass_ccsic_b
HP-1114	U	Klopsch	2017-08-28	2481024	Platte	None	200.0	200.0
HP-1116	U	Klopsch	2017-08-31	2481024	Platte	None	200.0	200.0
HP-1207	Masterarbeit Alina	Kessel	None	None	Platte	None	200.0	200.0
HP-1208	Masterarbeit Alina	Kessel	None	None	Platte	None	200.0	200.0
HP-1209	Masterarbeit Alina	Kessel	None	None	Platte	None	200.0	200.0
HP-1210	Masterarbeit Alina	Kessel	None	None	Platte	None	200.0	200.0
HP-1211	Masterarbeit Alina	Kessel	None	None	Platte	None	200.0	200.0
HP-1216	Stort	Dauth	None	None	Platte	None	400.0	280.0
HP-1289	Automatisches_Projekt	Automatisch_Erstellt	2000-01-01	999	None	None	None	None
HP-1290	KermiHt	Fiona Kessel	2021-11-04	3024906	None	None	None	None
HP-1312	KermiHt	Fiona Kessel	2021-11-04	3024906	Platte	None	290.0	90.0
HP-1330-A	LFBN	Friedrich	None	None	Platte	None	325.0	325.0
HP-1330-B	LFBN	Friedrich	None	None	Platte	None	325.0	325.0
HP-1330-C	LFBN	Friedrich	None	None	Platte	None	325.0	325.0
HP-1330-D	LFBN	Friedrich	None	None	Platte	None	325.0	325.0

Excel Export Home Exit

Abbildung 58: Beispielhafte Anwendung des interaktiven Laufzettels durch eine Datenbank-Suche nach allen Proben, deren ID mit den Ziffern „HP-1“ beginnt.

4.3 Erstellen von PDF-Zusammenfassungen

Um dem Nutzer einen schnellen Überblick über die gesamte Prozessierung speziell ausgewählter Proben zu ermöglichen, wurde in DataTracker eine weitere Funktion implementiert, welche PDF-Zusammenfassungen zu einzelnen vorgegebenen Proben erstellt. Die Zusammenfassung enthält die wichtigsten Eckdaten aller Prozessschritte, sowie Diagramme von Temperaturzyklen der Prozessschritte „Pyrolyse“ und „Silizierung“. Da zusätzlich jeweils eine Soll-Kurve mit in das Diagramm eingezeichnet wird, ist es außerdem leicht möglich, Abweichungen in den Prozessen zu identifizieren. Abbildung 59 zeigt beispielhaft einen Ausschnitt der PDF-Zusammenfassung der Probe HP-1330-A.

Zusammenfassung Probe HP-1330-A

Allgemeine Informationen

Projektleiter: Friedrich

Erstellungs-Datum: 21.11.2022

Geometrie: Platte

Sollmaß C/C-SiC (Länge x Breite x Dicke): 325.0 x 325.0 x 8.0 mm

Sollmaß CFK (Länge x Breite x Dicke):

FVG: 55.75 %

Faserart: Roving, Fasermaterial:T800

Precursor: JK60

Faser-Orientierung: 45°

Gesamt-Massenänderung durch Pyrolysen: 0.0 % (Insgesamt 0 Pyrolysen)

Gesamt-Massenänderung durch Silicierungen: 0.0 % (Insgesamt 0 Silicierungen inkl. Entsilicierung)

Polymerisation

Polymerisation stattgefunden am: 04.12.2022

Abbildung 59: Ausschnitt der automatischen PDF-Zusammenfassung der Probe WR-44-A.

4.4 Interaktive Diagramme

Die Messdaten der Prozesse „Polymerisation“, „Temperung“, „Pyrolyse“ und „Silizierung“ können in DataTracker als interaktive Diagramme angezeigt werden. Dabei ist es möglich, zwischen den verschiedenen Messinstrumenten und -positionen auszuwählen. Zusätzlich können auch verschiedene Kurven überlagert im selben Diagramm angezeigt werden. Speziell für Infiltrationsdaten kann außerdem eine Unterscheidung je nach Herstellungsverfahren erfolgen. Ein Beispiel hierfür ist in Abbildung 60 gegeben, wo eine Infiltrations-Temperaturkurve angezeigt wird. Eine weitere Funktionalität bietet außerdem die Möglichkeit, alle in der Datenbank verfügbaren Zyklen in einem einzigen Diagramm anzuzeigen.

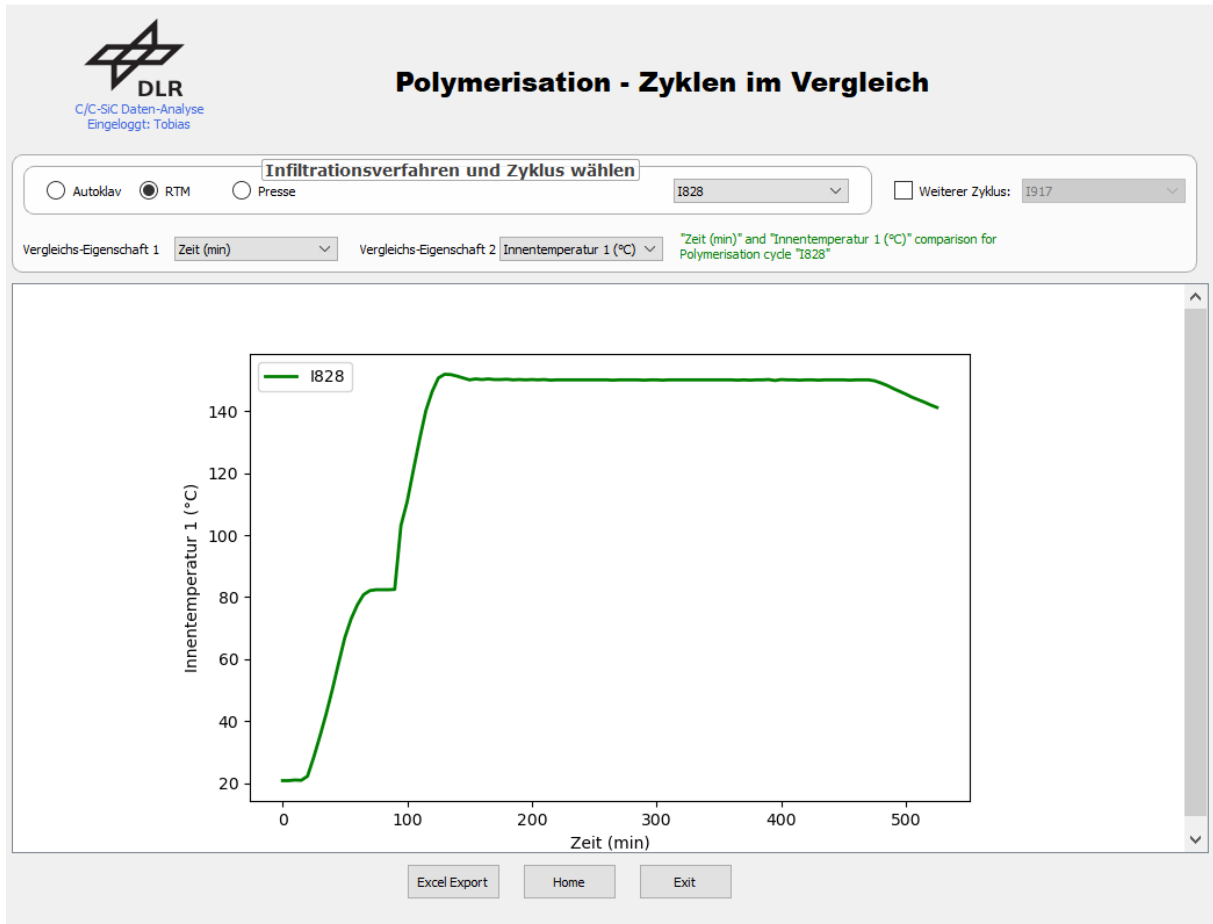


Abbildung 60: Graphische Darstellung einer Infiltrations-Temperaturkurve (RTM) in DataTracker. Durch Bedienung der Dropdown-Menüs und Schaltflächen lassen sich verschiedene Temperatur- und Druckkurven der Ist- und Soll-Werte, sowie mehrere Zyklen auf einmal anzeigen.

In Kapitel 3.3 wurde bereits beschrieben, wie REM-Aufnahmen in die Datenbank aufgenommen und bewertet werden können. Um die damit erzeugten Daten einsehen zu können, verfügt DataTracker über den Bereich „Mikrostruktur“, über welchen direkt eine Übersicht zu ausgewählten Proben gegeben werden kann. Diese umfasst dabei das originale REM-Bild selbst, ein Kuchendiagramm mit den darin enthaltenen Phasenanteilen, sowie die daraus extrahierte CCR. Sollten mehrere Aufnahmen zu der angegebenen Probe vorliegen, werden außerdem automatisch Mittelwert und Standardabweichung der CCR über alle vorliegenden Bilder berechnet und angezeigt. Des Weiteren kann über ein Dropdown-Menü zwischen den einzelnen Bildern derselben Probe gewechselt werden. Ein Beispiel ist in Abbildung 61 gegeben, welche Mikrostruktur-Informationen zur Probe WR-44-A zeigt.

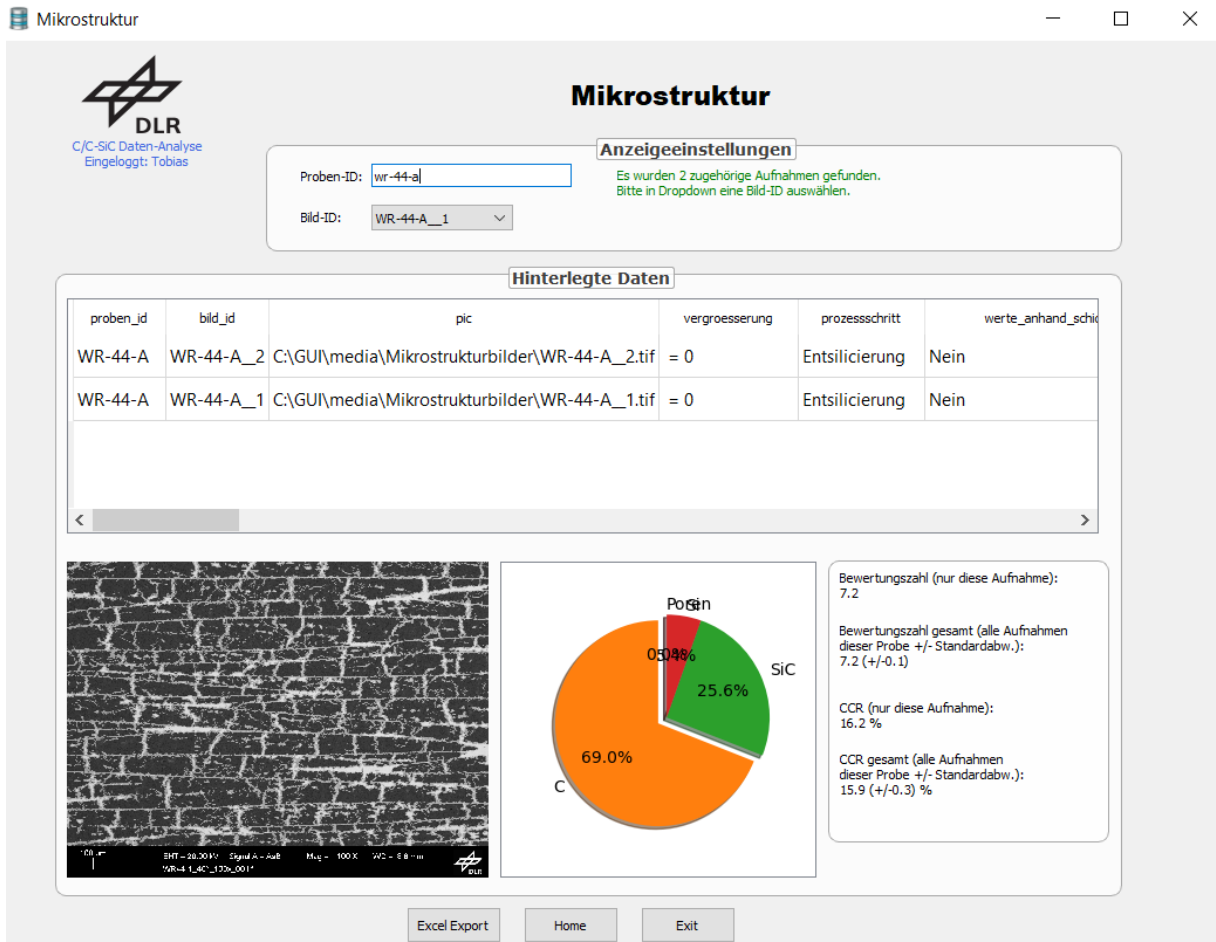


Abbildung 61: Anzeige einer Mikrostruktur zur Probe WR-44-A in DataTracker. Neben einer REM-Aufnahme wird weiterhin ein Phasenverteilungsdiagramm sowie die mittlere CCR inklusive Standardabweichung angegeben.

4.5 KI-Auswertung

Die KI-Auswertung nimmt in DataTracker eine zentrale Rolle ein und wird deshalb in diesem Kapitel ausführlich beschrieben. Grob kann die Abfolge der Arbeitsschritte in die drei in Abbildung 62 dargestellten Schritte eingeteilt werden, welche in den folgenden Unterkapiteln näher beschrieben werden.

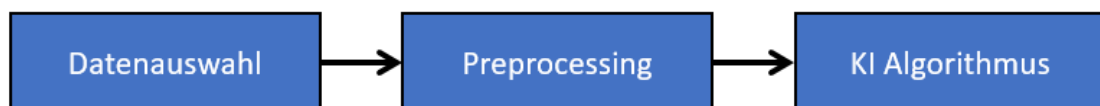


Abbildung 62: Vorgehen bei der Datenverarbeitung.

4.5.1 Datenauswahl

Allen nachfolgenden Untersuchungen lag ein Datensatz von 163 Proben zugrunde, welcher bereits über die letzten Jahre am DLR angesammelt worden, und teilweise im Verlauf dieser Arbeit durch Erstellung neuer Proben ergänzt worden war. Die Dokumentation war allerdings

lückenhaft und auf viele verschiedene Dokumente und Dateiformate verteilt, weswegen einige Daten-Vorverarbeitungsschritte angewendet werden mussten, auf die nun eingegangen wird.

Ausfüllgrad

Da die Ausführlichkeit der Dokumentation unter den einzelnen Proben stark schwankte, mussten einige davon komplett aus dem Datensatz entfernt werden. Das Entfernen von mangelhaft ausgefüllten Proben ist notwendig, da bei einer zu großen Fehlstellenrate das Schätzen der unbekanntenen Werte zu großen Fehlern führen kann, welche die Genauigkeit des KI-Modells verschlechtern. Um zu entscheiden, welche Proben als Datenbasis für die KI-Auswertung dienen sollen, wurde ein Ausfüllgrad definiert, welcher den Anteil ausgefüllter Felder im Laufzettel anhand aller verfügbaren Felder beschreibt (siehe Gleichung (28)).

$$A = \frac{\text{Anzahl ausgefüllter Felder}}{\text{Anzahl aller Felder}} \cdot 100\% \quad (28)$$

Dasselbe Vorgehen wurde auch für Herstellungsparameter durchgeführt. War ein bestimmter Herstellungsparameter unter allen Proben nur sehr selten ausgefüllt, wurde dieser komplett aus der Datenbasis entfernt. Folglich musste eine Untergrenze definiert werden, unterhalb welcher Proben und Herstellungsparameter (entspricht Zeilen und Spalten) des Datensatzes aussortiert werden. Diese sollte so gewählt werden, dass die maximale Menge an Daten bestehen bleibt, die gerade noch keine großen Genauigkeitseinbußen verursacht. Der Ausfüllgrad ist nicht nur bei Proben, sondern auch bei Herstellungsparametern relevant. Für beide Fälle wurde er als Variable in DataTracker implementiert, sodass unterschiedliche Schwellen schnell und einfach getestet werden konnten.

Ausreißer

Als nächstes wurden Ausreißer aus dem Datensatz entfernt. Analog zum Ausfüllgrad wurde auch dieses Feature als optionale Möglichkeit in DataTracker implementiert. Als Ausreißer wurden dabei jene Proben betitelt, welche in der CCR eine Z-Statistik von $|z| \geq 3$ besitzen, da dies in der Praxis als guter Orientierungswert gilt [69]. Die theoretischen Grundlagen zur Z-Statistik wurden bereits in Kapitel 2.5.5 erläutert und werden an dieser Stelle nicht weiter vertieft. Sollte sich für eine Entfernung von Ausreißern entschieden werden, wurden auch hierfür mehrere Optionen implementiert:

- Entfernung von Ausreißern nur innerhalb der CCR
- Entfernung von Ausreißern aus allen Herstellungsparametern des Datensatzes
- Entfernung von Ausreißern aus einer Liste vorgegebener Herstellungsparameter

Bei normalverteilten Daten liegen 99,73% der Werte innerhalb des Intervalls $-3 < z < 3$, folglich wurden die 0,27% der Daten, welche am weitesten vom Mittelwert entfernt liegen, als

Ausreißer deklariert und aus dem Datensatz entfernt. Um auf Normalverteilung zu testen, wurde der Shapiro-Wilk Test angewendet, welcher mit einem Wert von $p < 0,01$ jedoch eine Normalverteilung ausschloss. Um dem entgegenzuwirken, wurde eine Box-Cox-Transformation auf den Datensatz angewendet. Ein erneuter Shapiro-Wilk Test ergab einen Wert von $p = 0,09$, was zu einer Annahme einer Normalverteilung führt. Auch die Verteilung der Fehlerwerte des angelernten RandomForest Modells konnte damit von einem Shapiro-Wilk-Test Ergebnis von $p = 0,03$ auf $p = 0,77$ verbessert werden.

Tabelle 13: Ergebnisse des Shapiro-Wilk Tests vor und nach der Box-Cox-Transformation.

	p-Wert vor Box-Cox-Transformation	p-Wert nach Box-Cox-Transformation
Ganzer Datensatz	< 0,01	0,09
Fehlerterm RF	0,03	0,77

Neben diesen beiden wichtigsten Datenauswahl-Schritten, können zusätzlich auch andere Bedingungen als Filterkriterien vorgegeben werden, beispielsweise eine Liste zulässiger Proben-Präfixe oder das zwingende Vorhandensein bestimmter Pyrolyse- und Silizierungsdaten. Darüber hinaus können bestimmte Prozessschritte oder Proben explizit ausgeschlossen werden. Das gesamte Datenauswahl-Verfahren ist in Abbildung 63 dargestellt. Wurde der gewünschte Datensatz aus der SQL-Datenbank heruntergeladen, wird er zur Datenvorverarbeitung (Preprocessing) weitergeleitet.

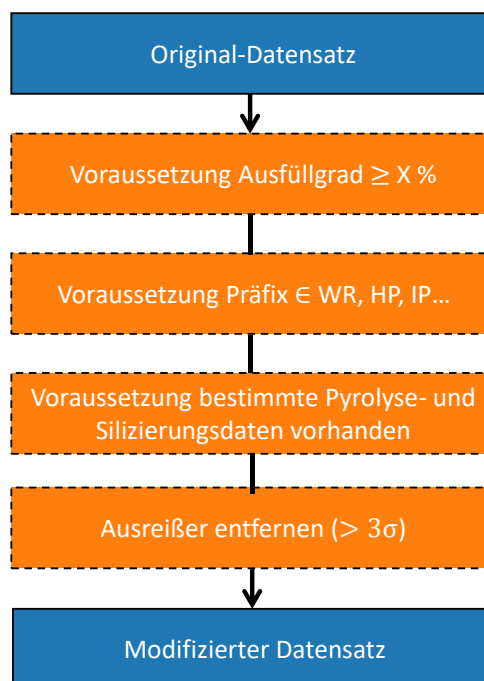


Abbildung 63: Optionen bei der Datenauswahl; optionale Schritte sind gestrichelt umrandet dargestellt.

4.5.2 Preprocessing

Wurde die Vorauswahl der Daten getroffen, wird nun in einem nächsten Schritt die Vorverarbeitung (*engl.*: Preprocessing) der Daten vorgenommen. Die durch das entwickelte Programm automatisierte Vorgehensweise ist in Abbildung 64 dargestellt.

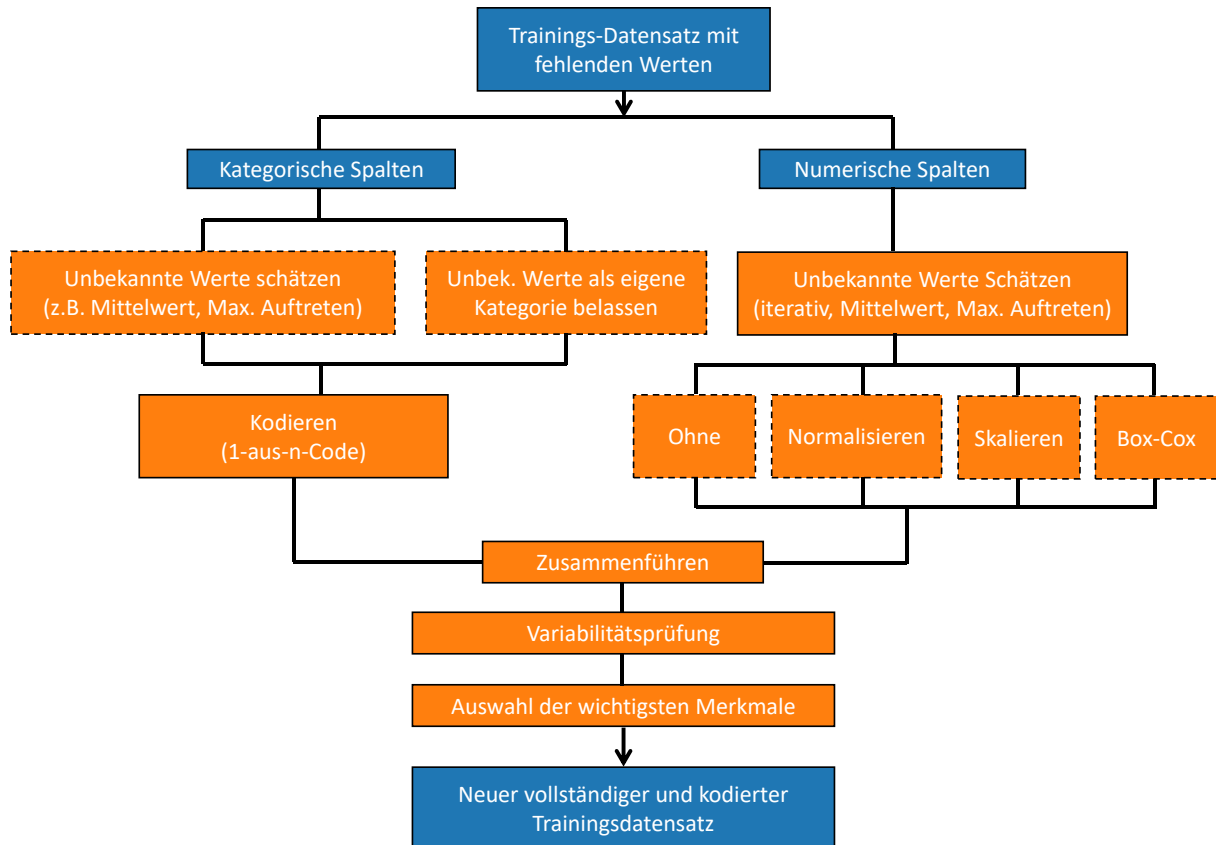


Abbildung 64: Schematisches Vorgehen beim Preprocessing der Daten; gestrichelt umrandete Schritte sind Entweder-Oder Entscheidungen.

Nachdem die Daten in kategorische und numerische Daten unterteilt wurden, müssen unbekannte Werte geschätzt werden. Dies kann auf unterschiedliche Weise geschehen, wie bereits in Kapitel 2.5.4 näher beschrieben. Im Fall von kategorischen Daten besteht die Möglichkeit einer einfachen univariaten Schätzung, oder alternativ eine eigene Kategorie für unbekannte Werte zu eröffnen. Im Fall von numerischen Daten kann zwischen einfacher univariater und iterativer multivariater Schätzung entschieden werden. Nachdem alle fehlenden Werte aus dem Datensatz eliminiert wurden, folgt für kategorische Variablen nun eine 1-aus-n-Kodierung (One Hot Encoding). Numerische Daten können an dieser Stelle linear oder nicht-linear transformiert werden (siehe Kapitel 2.5.6). Die eingebauten Möglichkeiten in DataTracker sind alle optional und umfassen Standardisierung, Normalisierung und Box-Cox-Transformation. Nach Kodierung und Transformation werden kategorische und numerische Daten wieder zusammengeführt. Anschließend wird geprüft, ob die vorhandenen Herstellungsparameter über ausreichend Variabilität verfügen, um statistisch untersucht werden

zu können. Zuletzt kann optional eine Bestimmung der wichtigsten Herstellungsparameter (Feature Selection) des Datensatzes erfolgen, und eine Schwelle bestimmt werden, unterhalb der ein Herstellungsparameter nicht mit in den Trainingsprozess aufgenommen wird. Dies spart Rechenzeit, verbessert die Interpretierbarkeit des Modells und kann verhindern, dass zufällige Korrelationen unkorrelierter Parameter die Genauigkeit verschlechtern. Zur Bewertung der Wichtigkeit aller Herstellungsparameter wurde, soweit nicht anders beschrieben, immer ein separat angelernter RandomForest Algorithmus verwendet. So können aus einer großen Fülle von Herstellungsparametern die n wichtigsten herausgefiltert werden.

4.5.3 Training der KI

Nach dem Preprocessing werden die Daten automatisch an eines von vier auswählbaren Machine-Learning Modellen weitergegeben. Dessen Hyperparameter sind durch das Programm frei wählbar. Alternativ kann auch eine Parametersuche über einen vorgegebenen Parameterraum gestartet werden, von dem die besten 5 Modelle behalten werden.

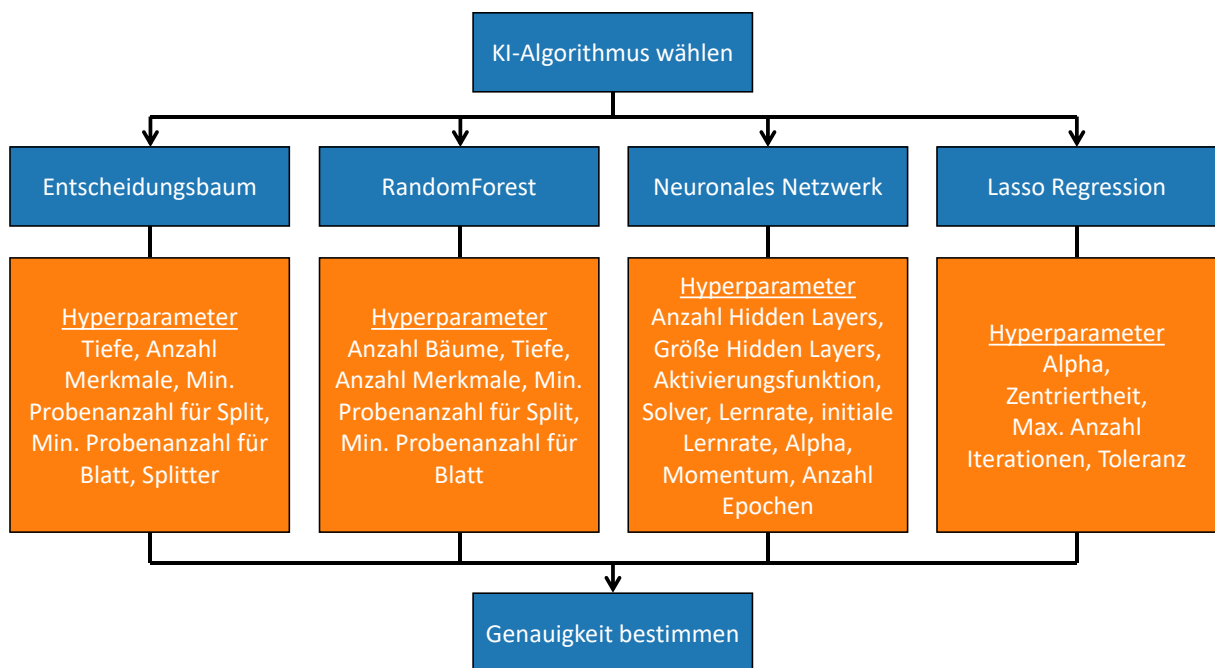


Abbildung 65: Implementierte Machine-Learning Modelle und deren Hyperparameter.

Abbildung 66 zeigt beispielhaft den Reiter „Einstellungen“ des Abschnitts „KI“ in DataTracker. Hier können umfassende Vorgaben zum anzulernenden Modell sowie zu dessen Hyperparameterräumen gemacht werden, auf die an dieser Stelle jedoch nicht im Detail eingegangen wird.

KI Auswertung

Mikrostruktur-Prognosen mittels KI

Daten wählen | **KI-Einstellungen** | Erweiterte Einstellungen | Feature Selection

Algorithmus wählen: RandomForest

Fehlende Daten schätzen: Einfacher Imputer Iterativer Imputer Max. Iterationen:

Imputer-Features begrenzen: Algo.: BRR LR RF KNN

Imputer Plot ausgeben Imputation in Plot unterscheiden

Kreuz-Validierung durchführen: Anzahl Wiederholungen (n): Anzahl Unterteilungen (k):

Hyperparameter Optimieren Anzahl Modelle:

Anzahl Bäume: Tiefe Bäume: Anzahl Features:

Min. Proben Split: Min. Proben Blatt:

Keine Bearbeitung Skalieren Normalisieren

Log erstellen KI speichern

KI trainieren Fortschritt: 100%

Info

Datenbasis wird aus DB heruntergeladen...
Anzahl gefundener Datensätze: 131

Für das Basis-Modell wurden 9 von 76 Features behalten.
CV Basis-Modell: $R^2=0.722 \pm 0.087$
Genauigkeit Bas.M. beim Test-Datensatz: $R^2=0.739$

Für das beste optimierte Modell wurden 6 von 76 Features behalten.
CV optimiertes Modell: $R^2=0.736 \pm 0.053$
Genauigkeit opt.M. beim Test-Datensatz: $R^2=0.708$

Datenbasis | Trainingsdaten | Testdaten | Beste 5 Modelle

proben_id	geometrie	sollmass_ccsic_d	faservolumengehalt_cfk	precursor	fasermaterial	faserart	faserdichte	schlichte	beschreibung_schlichte	vorbeh
WR-63-A	Platte	3.0	59.61	JK60	T800	Roving	1.81	Ja	0	Nein
HP-914	Platte	12.0	48.2	JK60	HTA	Gewebe	1.76	Ja	0	Nein
WR-68	sonst.	nan	nan	JK60	T800	Roving	1.76	Ja	0	Nein
WR-105	Platte	2.0	57.95	JK60	HTA	Roving	1.81	Ja	0	Nein
WR-69	sonst.	nan	nan	JK60	T800	Roving	1.76	Ja	0	Nein
WR-76	sonst.	nan	nan	JK60	T800	Roving	1.76	Ja	0	Nein
WR-80-A	Rohr	nan	50.0	JK60	HTA	Roving	1.81	Ja	0	Nein
WR-82-A	Rohr	nan	50.0	JK60	T800	Roving	1.81	Ja	0	Nein
HP-1116	Platte	nan	nan	JK63	HTA	Vlies	nan	Ja	0	Nein

Home Exit

Abbildung 66: Bereich KI in DataTracker; hier können umfassende Einstellungen zu Modell und verwendeten Daten getätigt werden.

Außerdem können über diese Schaltfläche Vorgaben zu folgenden Themen gemacht werden:

- Schätzmethode (falls iterativ auch: maximale Anzahl Iterationen und verwendeter Algorithmus)
- Feature-Selection-Methode
- Umgang mit Multikollinearität
- Methode zur Datenaufteilung (in Test- und Trainingsdaten)
- Auswahl automatisch erzeugter Diagramme und Dateien

Nachdem alle Einstellungen getätigt wurden, kann ein KI-Modell trainiert und als PKL-Datei abgespeichert werden. Zudem werden automatisch verschiedene Graphiken und diagnostische Dateien erzeugt und in einem vorgegebenen Ordner abgespeichert. Eine automatisch erzeugte Log-Datei zeichnet außerdem alle vom Programm vorgenommenen Schritte auf.

5 Physikalische Modellbildung und begleitende Tests

In diesem Kapitel wird auf die Mikrostruktur-Modellierung des Pyrolyseprozesses und die dazu nötigen praktischen Versuche eingegangen. Die Mikrostruktursimulation wurde anhand der Software Simcenter Multimech durchgeführt, mit dem Ziel, die Einflussfaktoren zu identifizieren, welche mit einer bestimmten Art von Rissmuster im C/C Zustand zusammenhängen. Für spezielle notwendige Funktionen, welche in der Standardsoftware nicht enthalten waren, wurden außerdem Python-Skripte entwickelt, welche diese auf manuellem Wege ermöglichen. Dadurch konnte eine Vergleichs- und Validierungsmöglichkeit für die Vorhersagen der datenbasierten Modelle geschaffen werden. Die praktischen Versuche dienen dabei der Bestimmung diverser notwendiger mechanischer Kennwerte des Phenolharzes XP60 als auch der in-situ Beobachtung der Pyrolyse durch ein Lichtmikroskop.

5.1 Mikrostruktursimulation

Die physikalische Modellbildung befasste sich ausschließlich mit dem Pyrolyseprozess, da eine vollständige Abbildung aller Prozessschritte den Rahmen dieser Arbeit sprengen würde, und die Pyrolyse als Schlüsselprozessschritt für die Entstehung des für die CCR wichtigen Rissmusters angesehen wird. Dabei wurde durch den Einsatz eines FEM-Mikrostruktur-Modells die Rissbildung innerhalb des Materials bei Aufbringen einer Temperaturlast simuliert. Hauptziel der Simulation war, ein ausreichend großes sowie stabiles Mikrostrukturmodell zu erschaffen, anhand dessen die beobachteten Phänomene der praktischen Tests nachgebildet werden konnten. Zudem sollten die gefundenen Korrelationen der datenbasierten Modelle auch durch physikalische Zusammenhänge erklärbar gemacht werden. Dabei ging es vorrangig um den Zusammenhang zwischen Porosität und entstehendem Rissmuster, da diese Korrelation zuvor von den KI-Modellen gefunden worden war. Vorausgesetzt wurde für alle Simulationen ein Faserverbundwerkstoff bestehend aus HTA-Fasern und XP60 Harz mit einem Faservolumengehalt von 59%. Folgende wissenschaftliche Fragestellungen werden durch die Mikrostruktursimulation beantwortet:

- Wie kommt es zu den vertikalen Segmentierungsrissen innerhalb der Matrix?
- Welchen Einfluss haben Defekte auf das entstehende Rissmuster?
- Wie unterscheidet sich das Rissmuster bei der Verwendung von HTA- und T800-Fasern?

5.1.1 Entstehung der vertikalen Segmentierungsrisse

Um die erste Fragestellung beantworten zu können, wurde folgende Vermutung aufgestellt: die Wärmeausdehnung des gesamten Laminats ist in Faserebene beinahe null, während sie in Dickenrichtung negativ ist, da dort keine Fasern verlaufen und das Verhalten damit matrixdominiert ist. Da das Phenolharz sich allerdings beim Aufheizen komprimieren möchte (negativer Wärmeausdehnungskoeffizient α), entstehen in der Faserebene Zugspannungen, welche beim Überschreiten der Matrix-Festigkeit zu Rissbildung führen [92]. Der Grund für dieses Verhalten ist also in den stark unterschiedlichen Wärmeausdehnungen von Fasern und Matrix zu sehen. Die Wärmeausdehnung wird in der Mechanik oft durch den Wärmeausdehnungskoeffizienten α beschrieben, welcher die reversible Ausdehnung eines Materials durch Temperatureinfluss beschreibt. In dieser Arbeit wird jedoch nicht zwischen der thermomechanisch bedingten und der chemisch bedingten Wärmeausdehnung unterschieden, sondern beide Effekte in Summe betrachtet. Der hier als Wärmeausdehnungskoeffizient α bezeichnete Kennwert bezieht sich somit nicht nur auf die reversible thermische Ausdehnung eines Materials, sondern auch auf seine irreversible Ausdehnung infolge chemischer Veränderung, wie es beispielsweise bei Phenolharzen während der Pyrolyse geschieht.

Abbildung 67 verdeutlicht die These der vertikalen Segmentierungsrisse aufgrund behinderter Wärmeausdehnung in Faserrichtung. Skizziert sind drei Lagen eines $(90^\circ/0^\circ/90^\circ)$ Schichtaufbaus, wobei 90° einem Faserverlauf parallel zur Bildebene und 0° einem Faserverlauf in die Bildebene hinein entspricht. Lediglich in Y-Richtung ist ein ungehindertes Schrumpfen der Matrixelemente beim Aufheizvorgang möglich (grüne Pfeile), weswegen sich in X- und Z-Richtung Spannungen aufbauen (orangefarbene Pfeile). Überschreiten die Spannungen die Festigkeitsgrenze, bilden sich Risse aus (blau).

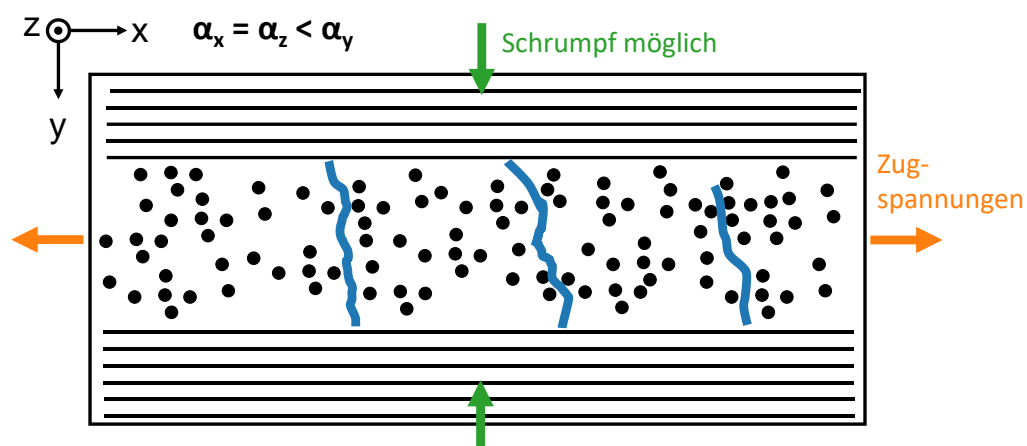


Abbildung 67: 3-Lagen Modell mit einer Faserorientierung von $(90^\circ/0^\circ/90^\circ)$. Aufgrund des niedrigen Wärmeausdehnungskoeffizienten der Fasern im Vergleich zur Matrix kann das Laminat nur in y-Richtung schrumpfen (grüne Pfeile); in X- und Z-Richtung wird dies durch die Fasern behindert, weshalb sich dort Risse ausbilden (blau).

Diese These wurde durch eine Simulation überprüft. Dazu war die Bestimmung von mechanischen Kennwerten für Faser- und Matrixmaterialien nötig, auf die in Kapitel 5.2 näher eingegangen wird. Zudem mussten einige Annahmen getroffen werden:

- Es tritt keine plastische Verformung auf (wird in Kapitel 5.2.2 näher untersucht)
- Alle Materialeigenschaften mit Ausnahme des E-Moduls der Matrix bleiben über die Temperatur im Bereich 20°C bis 600°C konstant
- Der Verlauf des E-Moduls des Phenolharzes XP60 über der Temperatur verhält sich analog zu dem für Phenolharze beschriebenen Verlauf aus Fan et al. [93]
- Die lokale Festigkeits-Verteilung der Matrix auf der mikroskopischen Ebene lässt sich durch eine Weibull-Verteilung annähern

Mit der oben definierten Wahrscheinlichkeitsverteilung für die Festigkeiten von Matrix und Interface, wurde ein (200µm x 200µm) großes 2D-Modell mit 3 Lagen erstellt, wobei die Größe der zu untersuchenden mittleren Lage die Maße 200µm x 67µm besitzt. Da es in Multimech nicht möglich ist, den Faserwinkel innerhalb derselben Skale zu verändern, wurde hierfür ein Python-Skript geschrieben, das die erstellten Vernetzungs-Dateien einlesen und entsprechend modifizieren kann. Dieses teilt die Mikrostruktur in drei gleiche Teile auf, berechnet die homogenisierten Eigenschaften einer Schicht, weist sie den Elementen der äußeren beiden Lagen zu und dreht diese anschließend um 90°, um Querlagen zu simulieren. Liest man die durch das Skript modifizierte Netz-Datei erneut in Multimech ein, erhält man die in Abbildung 68 gezeigte Struktur. Diese wurde für die Simulation an der Unterkante in vertikaler Richtung fixiert, sowie an der linken unteren Ecke in horizontaler Richtung, was durch die roten Loslager skizziert ist.

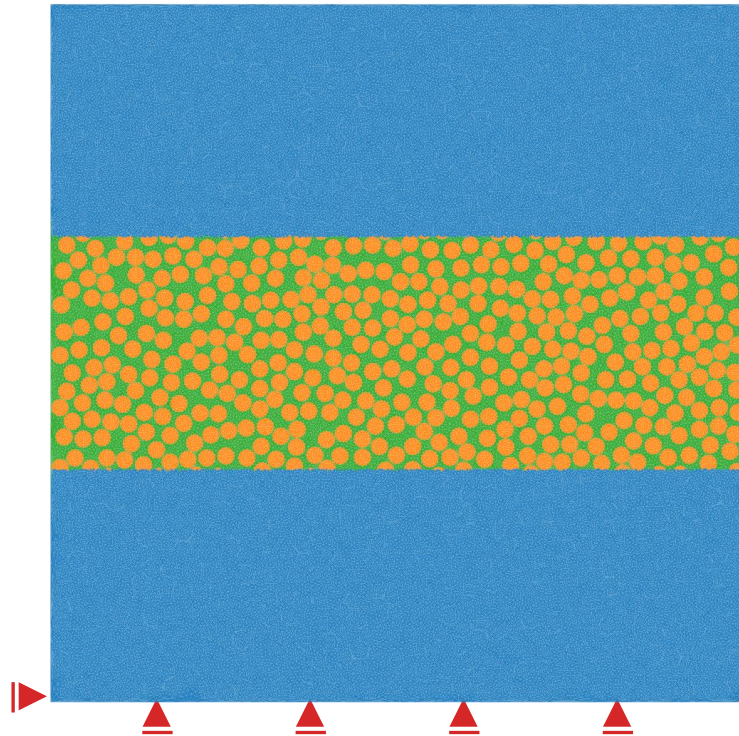


Abbildung 68: Mikrostruktur eines 3-Lagen Aufbaus mit Faserwinkel ($90^\circ/0^\circ/90^\circ$); Größe: $200\mu\text{m} \times 200\mu\text{m}$; blau: homogenisierte Querlagen, grün: Matrix, orange: Fasern; Faserdurchmesser: $5\mu\text{m}$; FVG=59%; Randbedingungen: vertikales Loslager entlang Unterkante und horizontales Loslager an linker unterer Ecke (rot).

Der Faserdurchmesser wurde dabei für T800-Fasern mit $d_F = 5\mu\text{m}$, und für HTA-Fasern mit $d_F = 7\mu\text{m}$ angenommen [94, 95]. Die verwendeten mechanischen und thermischen Eigenschaften für Fasern und Matrix sind in Tabelle 14 beschrieben. Messungen der Interface-Festigkeit für die verschiedenen Faser- und Harzmaterialien überstiegen aufgrund ihres großen Aufwands den Rahmen dieser Arbeit, weswegen die Interface-Festigkeit in dieser Arbeit mit der Matrix-Festigkeit gleichgesetzt wurde.

Tabelle 14: Mechanische und thermische Eigenschaften der CFK-Konstituenten mit Herkunft.

Kennwert	Symbol	Wert	Quelle
Matrix			
E-Modul	E_M	Temperaturabhängig [5,9 GPa bis 1,0 GPa]	Zugversuch
Poissonzahl	ν_M	$0,46 \pm 0,4$	Zugversuch
Festigkeit	σ_M	$13,6 \pm 5,3$ MPa	Zugversuch
Bruchzähigkeit	$K_{IC,M}$	$1,71$ MPa \sqrt{m}	SENB-Test
Wärmeausdehnung	α_M	Temperaturabhängig [-1,5e-4 bis 0,7e-4] [1/K]	Literatur [92]
Faser T800			
E-Modul	E_F	294 GPa	Literatur [95]

Poissonzahl	ν_F	0,36	Literatur [96]
Festigkeit	σ_F	5880 MPa	Literatur [95]
Wärmeausdehnung	α_F	-4e-7 [1/K]	Literatur [95]
Faservolumengehalt	FVG	59%	Datenbank
Faser HTA			
E-Modul	E_F	238	Literatur [94]
Poissonzahl	ν_F	0,36	Literatur [96]
Festigkeit	σ_F	3950 MPa	Literatur [94]
Wärmeausdehnung	α_F	-1e-7 [1/K]	Literatur [94]
Faservolumengehalt	FVG	59%	Datenbank

Die Matrix-Festigkeit wurde entsprechend einer Weibull-Verteilung angenommen, deren Mittelwert durch die in den Zugversuchen bestimmten Festigkeit von 13,6 MPa definiert wurde. Anhand im Test ermittelten Standardabweichung konnte außerdem der Formparameter der Weibull-Verteilung zu $\kappa = 16,6$ berechnet werden, wobei als Skalenparameter $\lambda = 1$ angenommen wurde (für Berechnung siehe Kapitel 2.7). Die mechanischen Matrix-Kennwerte wurden anhand von XP60 Proben über Zug- und SENB-Tests ermittelt, auf die im folgenden Kapitel näher eingegangen wird. Der Wärmeausdehnungskoeffizient α_M wurde bereits in einer vorhergehenden Masterarbeit am DLR Stuttgart durch Dilatometrie in Abhängigkeit der Temperatur bestimmt [92]. Alle anderen Kennwerte wurden aus den angegebenen Literaturen bezogen.

Anhand des Modells konnte die aufgestellte These für die Entwicklung von vertikalen Rissen aufgrund der unterschiedlichen thermischen Eigenschaften der Konstituenten gestützt werden. Das resultierende Rissmuster ist in Abbildung 69 dargestellt. Somit konnte der zugrundeliegende Mechanismus für die Ausbildung der vertikalen Segmentierungsrisse identifiziert werden.

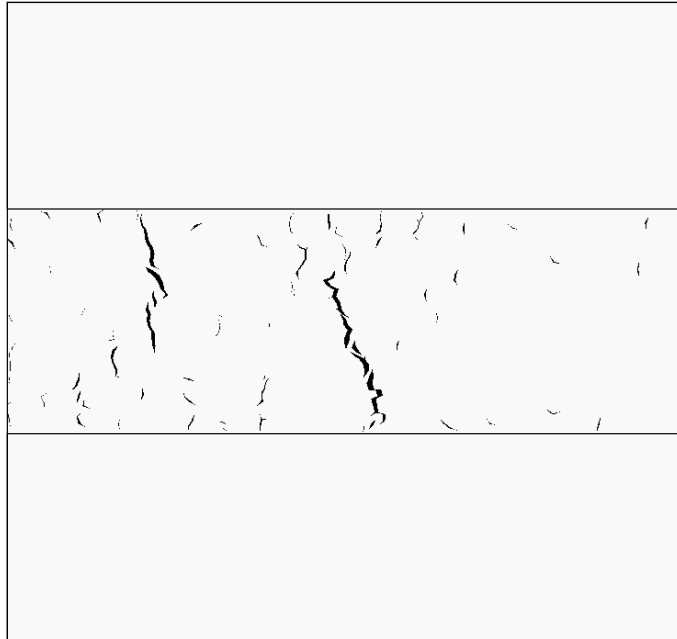


Abbildung 69: Entstehendes Rissmuster im 3-Lagen Modell mit HTA-Fasern bei einer Aufheizung von 20°C bis 600°C, weiß: intakte Struktur, schwarz: Risse.

5.1.2 Einfluss von Defekten und Fasermaterial auf das Rissystem

Dieser Mechanismus wurde nun auch auf eine weitere Simulation angewendet, wobei in dieser auf die Querlagen verzichtet, und stattdessen die Randbedingungen so angepasst wurden, dass die Ausdehnung des Laminats in Faserrichtung nicht möglich war. Auf diese Weise konnte ein größerer Bereich von 200µm x 200µm (L x B) abgedeckt werden, was in etwa den im Test als relevant erachteten Dimensionen zur Untersuchung eines Segmentierungsrisses entsprach, wie Abbildung 70 zeigt.

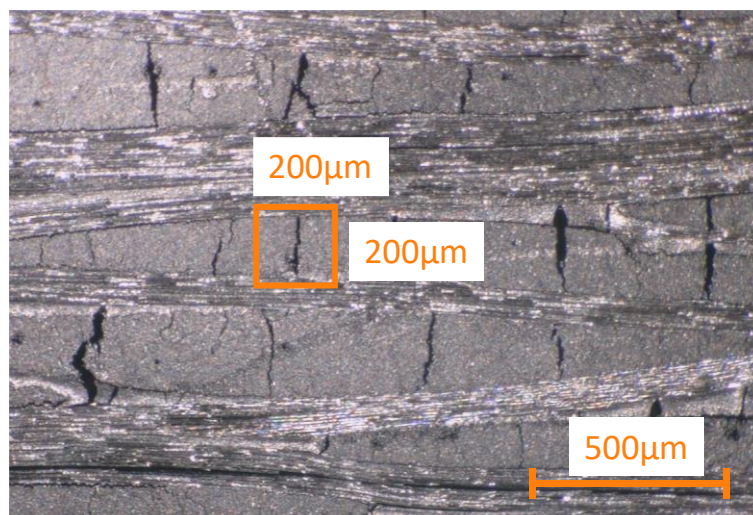


Abbildung 70: Mikrostrukturaufnahme einer Probe während der Pyrolyse mit eingezeichnetem Simulationsbereich.

Um den Einfluss von Defekten untersuchen zu können, wurde die Simulation bei sonst gleichen Einstellungen jeweils für einen Defektanteil von $D = 0\%$, $D = 5\%$ und $D = 10\%$ durchgeführt. Dabei wurden Defekte dadurch erzeugt, dass der E-Modul und die

Querkontraktionszahl von zufällig ausgewählten Elementen auf null gesetzt wurde, was einer Entfernung dieser Elemente gleichkommt, ohne dass die Stabilität der Rechnung litt. Die erstellten Mikrostrukturen inklusive Rissysteme für die Fälle $D = 0\%$ und $D = 10\%$ sind in Abbildung 71 dargestellt, wobei in a) und b) HTA-Fasern verwendet wurden, sowie in c) und d) T800-Fasern.

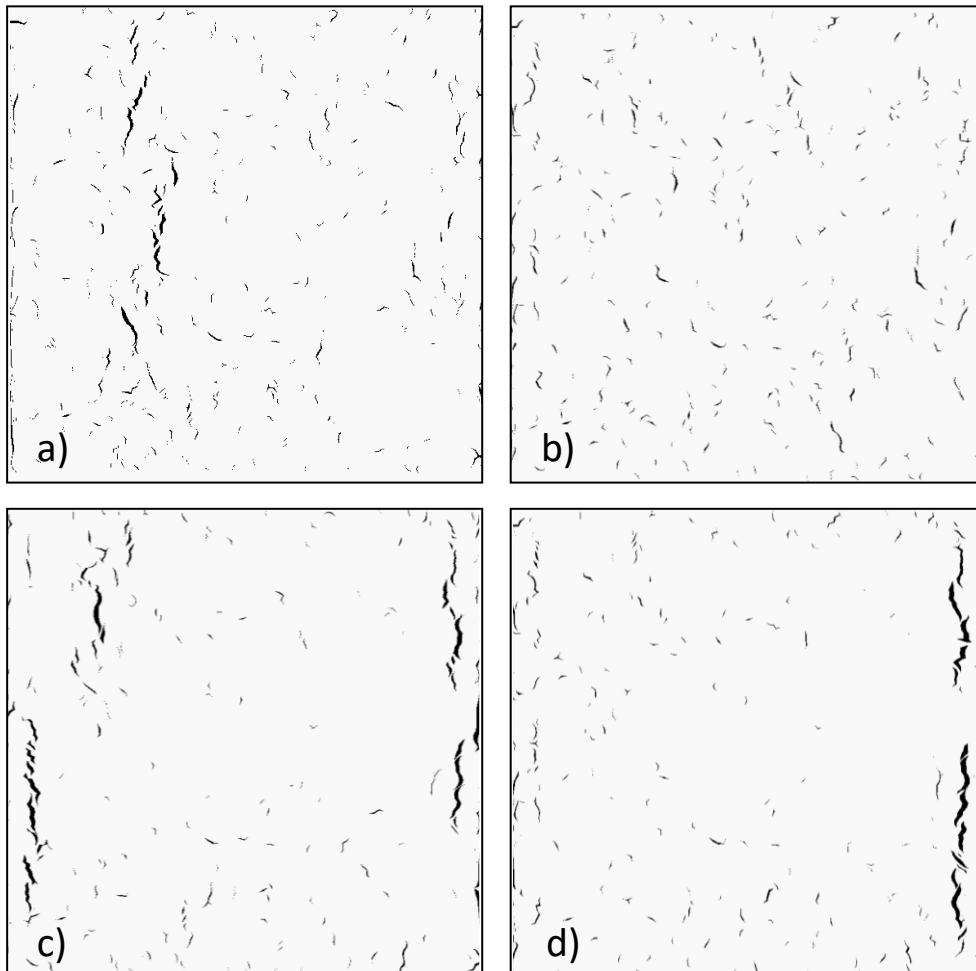


Abbildung 71: Entstehendes Rissmuster für a) HTA-Fasern bei $D=0\%$, b) HTA-Fasern bei $D=10\%$, c) T800-Fasern bei $D=0\%$ und d) T800-Fasern bei $D=10\%$ unter sonst gleichen Bedingungen; weiß: intakte Mikrostruktur, schwarz: Risse.

Um die oben gezeigten Bilder statistisch vergleichen zu können, wurde außerdem ein Python Skript geschrieben, welches Simulationsbilder einlesen und statistisch auswerten kann. Extrahiert wurden aus jedem Bild die Anzahl und Größe der Risse, sowie deren Dichte pro Quadratmikrometer Bildfläche, jeweils für eine Simulation mit HTA-Faser und mit T800 Fasern. Die Ergebnisse sind in Abbildung 72 festgehalten.

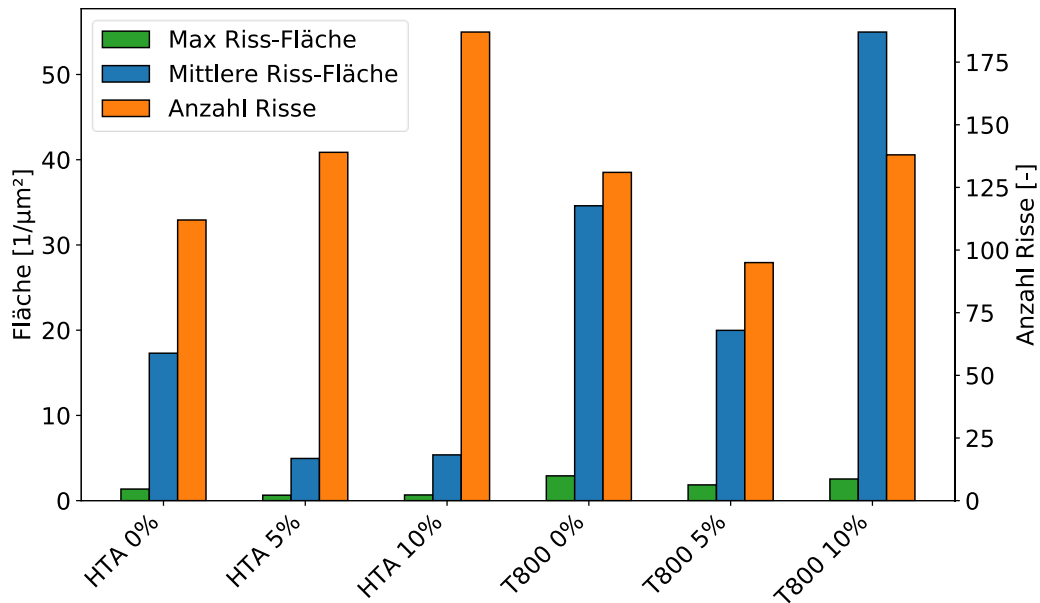


Abbildung 72: Ergebnisse der Bildauswertung der simulierten Mikrostrukturen, der Prozentsatz hinter dem Fasermaterial gibt die Defektdichte in der Matrix an.

Wie schon anhand der Mikrostrukturbilder erkennbar, treten bei den simulierten Mikrostrukturen mit HTA-Fasern ohne Defekte tendenziell weniger, aber dafür größere Risse auf. Im Vergleich dazu liegt die Rissdichte bei Mikrostrukturen mit Defektanteil von $D = 5\%$ etwa 21% höher, während sich die mittlere Rissgröße mehr als halbiert hat. Während die Größe der Risse bei einer Defektdichte von $D = 10\%$ im Vergleich zu $D = 5\%$ nicht weiter absinkt, steigt Anzahl der Risse in diesem Fall jedoch noch einmal beträchtlich an, und ist damit 64% höher als im Bild ohne Defekte. Die hohe Konzentration an kleinen Rissen im Falle der HTA-Simulationen mit Defekten lässt sich dadurch erklären, dass an den Rändern der Defekte Spannungsspitzen entstehen, wodurch die Matrixfestigkeit an vielen Stellen lokal überschritten wird. Das entstehende Rissystem ist fein verästelt und relativ homogen, wie es eher für XD-Strukturen typisch ist. In der Simulation ohne Defekte bilden sich hingegen größere Risskanäle bei gleichzeitig geringerer Anzahl von Rissen, wie es für XB-Strukturen typisch ist. Daraus lässt sich schließen, dass das Vorhandensein von Defekten wie Poren, Lunker oder Rissen im Material zu einer größeren Angriffsfläche für flüssiges Silizium während der Silizierung führt. Je mehr Defekte im Ausgangszustand vorhanden sind, desto feiner verteilt ist das entstehende Rissystem nach der Pyrolyse, was zu einer höheren Zahl an Einzelfasersilizierungen während der Silizierung und damit zu einer höheren CCR führt. Eine Beeinflussung des Rissmusters durch Mikrostruktureigenschaften wurde bereits bei Kosin [92] und Jain [5] simulativ festgestellt. Hier führte eine schwächere Faser-Matrix-Anbindung zu einer erhöhten Anzahl von kleineren Rissen, während eine stärkere Faser-Matrix Anbindung in weniger, dafür aber größeren Rissen resultierte.

Bei den T800-Fasern besteht der Trend zwischen Defektdichte und Anzahl der Risse nicht. Allerdings konnte beobachtet werden, dass sich durch die Verwendung von T800-Fasern bei einer vergleichbaren Anzahl Risse tendenziell größere Risse bildeten, als bei der Simulation mit HTA-Fasern. Da im Vergleich zur HTA-Simulation lediglich Faserdurchmesser, thermischer Ausdehnungskoeffizient und E-Modul variiert wurden, ist der Unterschied auf eine dieser Größen zurückführbar. Außerdem ist an dieser Stelle anzumerken, dass in der Realität Unterschiede in der Anbindungsfestigkeit zwischen den beiden Fasermaterialien und dem Matrixsystem existieren, welche in dieser Simulation nicht berücksichtigt wurden. Warum eine höhere Defektdichte bei T800 Fasern nicht zu einer größeren Anzahl Risse führte, konnte im Zuge dieser Arbeit nicht vollständig geklärt werden. Es wird jedoch vermutet, dass aufgrund des geringeren Faserdurchmessers der T800-Fasern ein homogeneres Gefüge entsteht, welches bei Vorhandensein von Defekten weniger Spannungskonzentrationen ausbildet. Der ausbleibende Zusammenhang zwischen Defekten und Rissanzahl bei Faserverbunden mit T800-Fasern wird aber auch im folgenden Kapitel durch die datenbasierten Modelle detektiert. Hier führte eine Erhöhung der Porosität im CFK-Zustand zu keiner signifikanten Erhöhung der CCR.

5.2 Praktische Versuche

Obwohl das Hauptaugenmerk dieser Arbeit auf der Modellierung und Digitalisierung lag, waren einige begleitende praktische Tests unerlässlich. Denn um eine aussagekräftige Mikrostruktursimulation durchführen zu können, waren diverse mechanische Kennwerte der Konstituenten zu ermitteln, oder aus der Literatur zu beziehen. Eine Zusammenfassung der Werte wurde bereits in Tabelle 14 aus dem vorherigen Kapitel gegeben.

5.2.1 Zugversuche

Die Ermittlung der zuvor beschriebenen Materialkennwerte wurde durch eine Materialprüfmaschine des Typs Zwick Roell UTS 10 und einem Prüfkopf der Stärke 2,5kN realisiert. Zunächst wurden 5 Harzproben des Typs XP60 hergestellt und auf die Dimensionen 70mm x 10mm x 3mm (L x B x H) zugeschnitten. Anschließend wurde jede Probe mit Längs- und Quer-Dehnmessstreifen versehen, um neben der reinen Zugfestigkeit auch eine Ermittlung des E-Moduls und der Querkontraktionszahl zu ermöglichen. Dabei wurde der E-Modul durch lineare Regression der Spannungs-Dehnungskurve im Auswertungsbereich zwischen 10% und 30% der maximalen Zugfestigkeit berechnet, wobei allgemein Gleichung (29) gilt.

$$E = \frac{\sigma}{\varepsilon} \quad (29)$$

Die Querkontraktionszahl kann durch Gleichung (30) ermittelt werden.

$$\nu = -\frac{\varepsilon_{quer}}{\varepsilon_{längs}} \quad (30)$$

5.2.2 SENB-Tests

Single-Edge-Notched-Bend-Tests (im Folgenden: SENB-Tests) dienen der Ermittlung bruchmechanischer Kennwerte und sind nach ASTM E 399 [97] genormt. Dabei wird eine gekerbte Probe durch zwei Auflager gestützt und anschließend durch eine Biegekraft belastet. Der Aufbau ist in Abbildung 73 dargestellt [98].

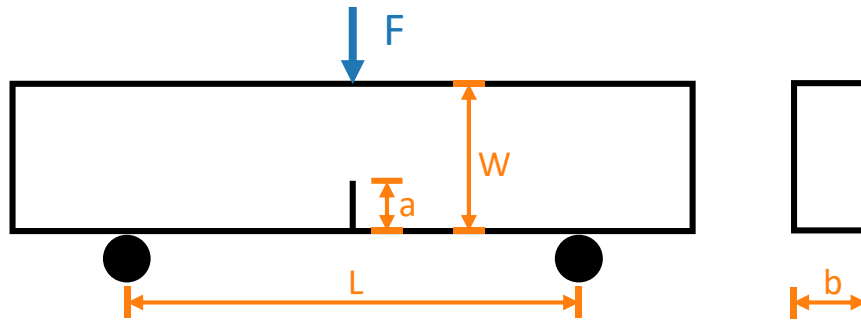


Abbildung 73: Versuchsaufbau eines SENB-Tests mit gekerbter Probe.

Anschließend kann über Gleichung (31) die Bruchzähigkeit K_{IC} des Prüflings errechnet werden [14].

$$K_{IC} = \sigma_b \sqrt{\pi a} F \left(\frac{a}{W} \right) \quad (31)$$

Dabei beschreibt a die Risslänge, W die Höhe der Probe, und σ_b die Biegespannung, welche für Rechteckquerschnitte nach Gleichung (32) berechnet werden kann [14].

$$\sigma_b = \frac{1.5FL}{bW^2} \quad (32)$$

Für ein $\frac{a}{W} = 4$ ergibt sich $F \left(\frac{a}{W} \right)$ nach Gleichung (33) [14].

$$F \left(\frac{a}{W} \right) = \frac{1}{\pi} \cdot \frac{1.99 - \frac{a}{W} \left(1 - \frac{a}{W} \right) \left(2.15 - \frac{3.93a}{W} + 2.7 \left(\frac{a}{W} \right)^2 \right)}{\left(1 + \frac{2a}{W} \right) \left(1 - \frac{a}{W} \right)^{\frac{3}{2}}} \quad (33)$$

Anhand der SENB-Versuche wurden fünf XP60 Proben mit den Maßen $W = 20\text{mm}$, $a = 5\text{mm}$, $L = 80\text{mm}$ und $F_{mittel} = 200\text{N}$ getestet, wodurch sich eine mittlere Bruchzähigkeit von $K_{IC} = 1,7 \text{ MPa}\sqrt{m}$ ergab. Im Vergleich dazu liegen Aluminiumlegierungen deutlich höher, bei etwa $22\text{-}33 \text{ MPa}\sqrt{m}$ und Stähle im Bereich $30\text{-}150 \text{ MPa}\sqrt{m}$ [99].

Die niedrige Bruchzähigkeit des verwendeten Phenolharzes macht deutlich, dass dieses Material ein sprödes Bruchverhalten aufweist. Damit ist die Annahme, dass Plastizität in den

physikalischen Modellen vernachlässigt werden kann, zumindest für Raumtemperatur bestätigt. Eine Durchführung von SENB-Tests bei Hochtemperatur war durch die Testanlagen am DLR nicht möglich. Aufgrund der sehr niedrigen Bruchzähigkeit wurde die Annahme aufgestellt, dass diese auch bei höheren Temperaturen nicht soweit steigt, dass plastisches Verhalten auftritt, was jedoch nicht endgültig bewiesen werden konnte.

5.2.3 In-situ Beobachtung der Pyrolyse

Um die Simulationsergebnisse validieren zu können, wurde der Pyrolyseprozess außerdem anhand des Digitalmikroskops VHX-5000 der Firma Keyence in Echtzeit beobachtet und gefilmt. Um die Pyrolyse zu beobachten, wurde ein kleiner Heiztisch verwendet, welcher Platz für etwa daumennagelgroße Proben bietet und an der Oberseite eine Glasabdeckung besitzt, durch den der Aufheizprozess mitverfolgt werden kann. Der Heiztisch ist mit einem Kühlwasserkreislauf und einem Schutzgasanschluss verbunden, durch den während der Pyrolyse Argon geleitet wurde, um die Reaktion mit Sauerstoff zu verhindern. Die für die Versuche verwendeten Objektive verfügen über eine zwischen 500-fache und 2000-fache Vergrößerung. Die angestrebte Maximaltemperatur lag für alle Versuche bei 1000°C. Obwohl dies unter der maximalen Pyrolysetemperatur für Standardproben am DLR liegt, ist diese Grenze für die Beobachtung der Rissentstehung im CFK ausreichend, da die pyrolytischen Vorgänge ab einer Temperatur von etwa 600°C abgeschlossen sind [1]. Als Beobachtungsobjekte dienten mehrere Proben derselben Prepreg-Platte, welche aus JK60 Harz und HTA-Fasern bestand, und vor der Pyrolyse getempert worden war. Um auch den Einfluss von schneller und langsamer Erhitzung abschätzen zu können, wurde der Versuch mit drei verschiedenen Temperaturprofilen an unterschiedlichen Proben derselben Platte durchgeführt. Die verwendeten Temperaturprofile sind in Abbildung 74 dargestellt. Es konnte jedoch keinerlei Einfluss der Aufheizrate auf die Mikrostruktur festgestellt werden.

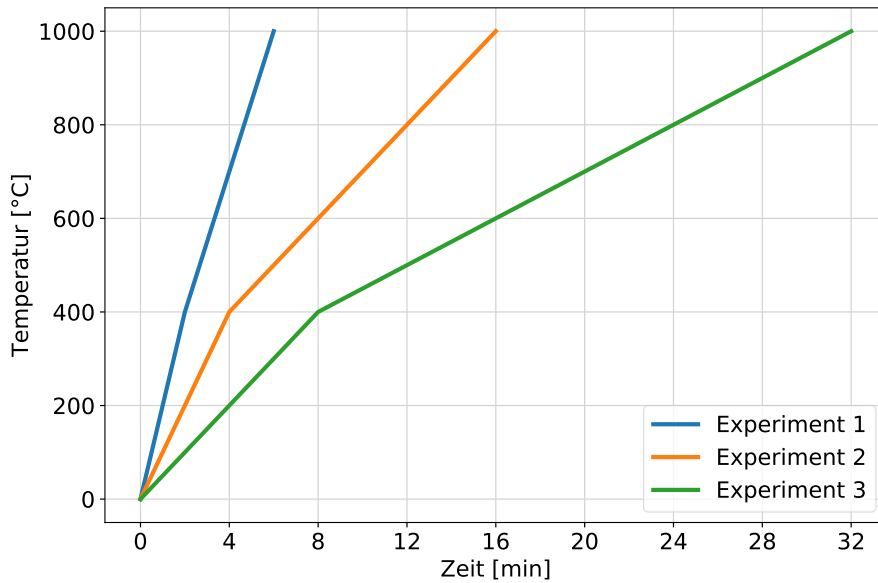


Abbildung 74: Temperaturprofile der Aufheizraten dreier untersuchter Pyrolysen.

Durch den Vergleich der Mikrostruktur derselben Probe vor und nach der Pyrolyse wurde ersichtlich, dass das Rissmuster, welches sich während der Pyrolyse deutlich sichtbar ausprägt, bereits davor vorhanden war (siehe Abbildung 75). Diese Beobachtung ist deswegen von Bedeutung, da sie den Schluss zulässt, dass die Position und Richtung der makroskopischen Segmentierungsrisse bereits während der Polymerisation oder Temperung festgelegt wird, und die Pyrolyse diese nur noch aufweitet. Ein Vergleich der simulierten und real beobachteten Risse macht darüber hinaus die gute Übereinstimmung der beiden Verfahren deutlich.

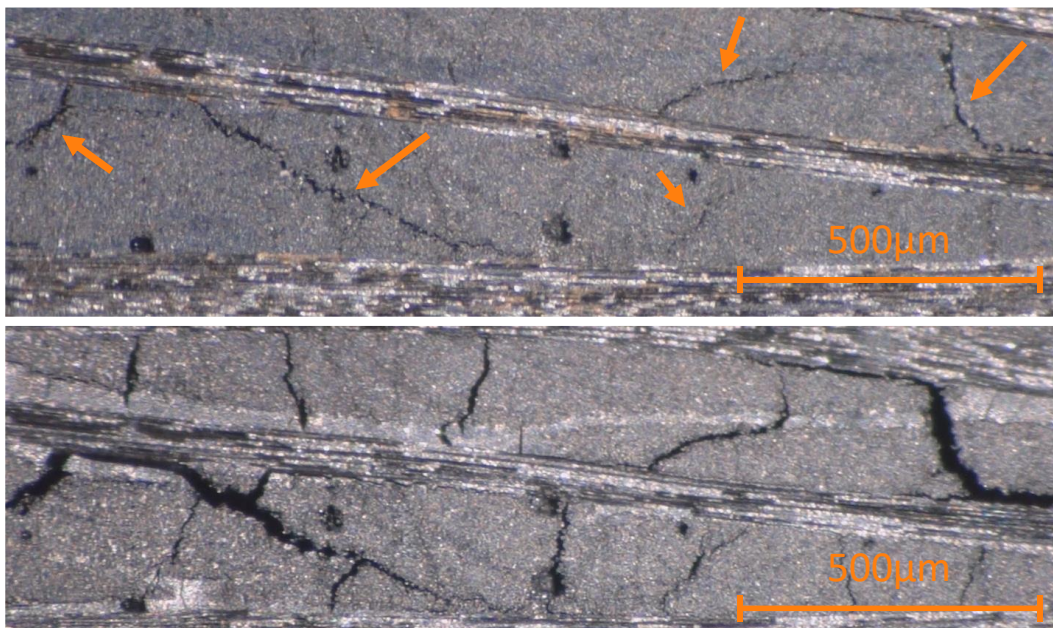


Abbildung 75: Vergleich der Mikrostruktur derselben Probe im getemperten Zustand vor der Pyrolyse (oben) und während der Pyrolyse (unten). Die Positionen der makroskopischen Risse sind bereits schon vor der Pyrolyse sichtbar (orange Pfeile).

6 Ergebnisse und Diskussion

Dieses Kapitel beschäftigt sich ausführlich mit den gewonnenen Ergebnissen dieser Arbeit und diskutiert deren Auswirkungen. Dabei wird sowohl auf Erkenntnisse zur Optimierung methodischer Vorgehensweisen als auch auf solche materialwissenschaftlicher Art eingegangen.

6.1 Methodische Erkenntnisse

Dieses Unterkapitel befasst sich ausschließlich mit der optimalen Auswahl methodischer Vorgehensweisen und enthält Parameterstudien zu den in DataTracker implementierten Funktionen. Materialwissenschaftliche Erkenntnisse werden hingegen erst im darauffolgenden Kapitel besprochen.

Für alle Parameterstudien in diesem Kapitel wurde eine einheitliche Vorgehensweise zur Bestimmung der mittleren Modellgenauigkeit verwendet. Dabei wurde stets ein RandomForest Algorithmus verwendet und die Genauigkeit anhand zehn unterschiedlicher Aufteilungen („Splits“) in Trainings- und Testdaten ermittelt, sodass Mittelwert und Standardabweichung berechnet werden konnten. Dabei wurden, sofern nicht anders beschrieben, zusätzlich die in Tabelle 15 aufgelisteten Einstellungen beibehalten.

Tabelle 15: Standard-Einstellungen für alle Parameterstudien aus Kapitel 6.1, sofern im jeweiligen Unterkapitel nicht anders beschrieben.

Method	Wert
Algorithmus	RandomForest
Anzahl Durchläufe (Splits)	10
Splitting-Methode	zufällig
Vorausgesetzter Ausfüllgrad	50% (Zeilen und Spalten)
Feature-Selection Methode	Modell-intrinsisch (RF)
Schätzmethode	iterativ
Ausreißer entfernen (3σ)	Nein
Multikollinearität behandeln	Nein

6.1.1 Optimale Untergrenze für Ausfüllgrad

Wie bereits in Kapitel 4.5.1 erläutert, beschreibt der Ausfüllgrad den Mindest-Prozentsatz ausgefüllter Felder einer Zeile (Probe) oder Spalte (Herstellungsparameter), damit diese im Datensatz beibehalten wird. Dabei handelt es sich um ein Optimierungsproblem. Wird der Ausfüllgrad zu hoch angesetzt, würden unnötig viele Proben aus dem Datensatz eliminiert werden, was eine Verschwendung von Information bedeuten würde. Wird sie zu niedrig angesetzt, würden auch sehr schlecht dokumentierte Proben in die Auswertung miteinbezogen,

was problematisch für die verwendeten Schätzmethoden sowie die Genauigkeit und Varianz der Modelle wäre. Deshalb wurde zur Bestimmung des optimalen Ausfüllgrads eine Parameterstudie unter Konstanthaltung aller anderen Modelleinstellungen durchgeführt. Diese umfasste Ausfüllgrade zwischen $A = 0\%$ und $A = 70\%$, welche anschließend anhand der resultierenden Modellgenauigkeit verglichen wurden. Das Ergebnis dieser Untersuchung ist in Abbildung 76 zu sehen.

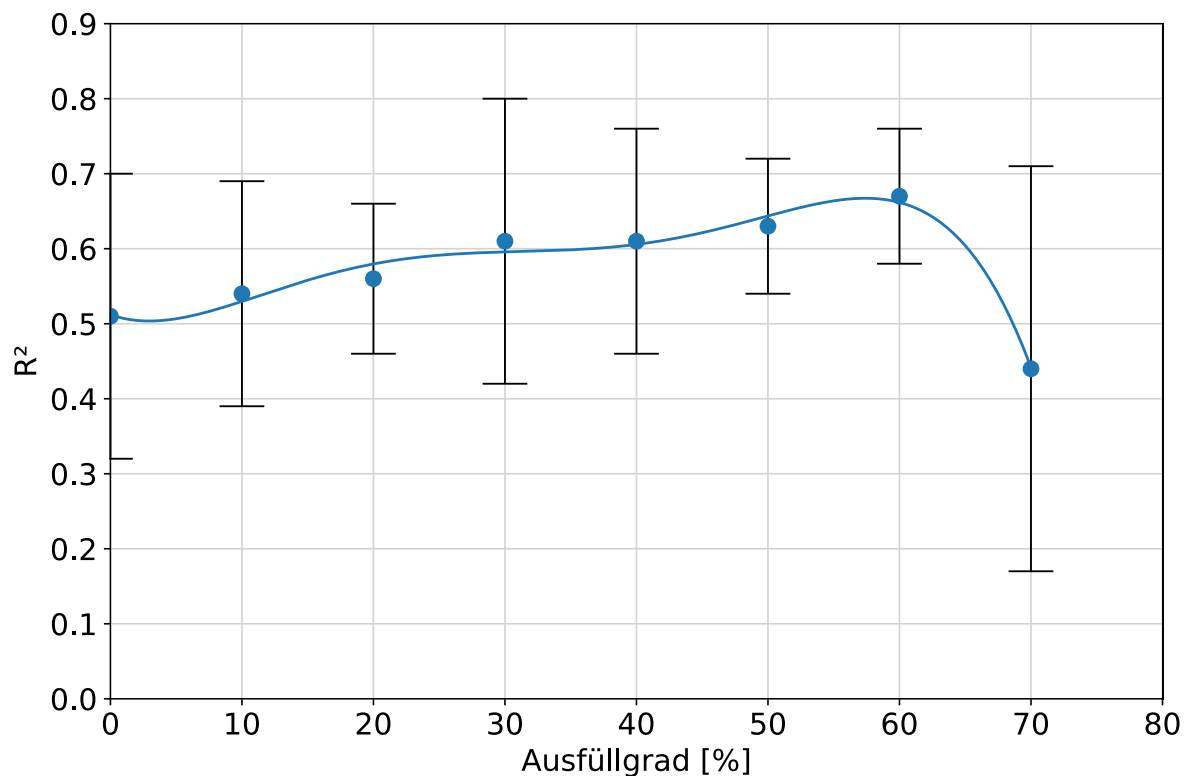


Abbildung 76: Modellgenauigkeit R^2 über Ausfüllgrad, das Optimum tritt bei $A=60\%$ auf.

Es wird ersichtlich, dass ein Ausfüllgrad von etwa $A = 60\%$ für Proben und Herstellungsparameter die besten Modellgenauigkeiten und geringsten Standardabweichungen liefert. Ab Ausfüllgraden $A > 60\%$ werden zu viele Zeilen und Spalten aus dem Datensatz entfernt, wodurch die Modellgenauigkeit aufgrund der unzureichenden Größe des Datensatzes stark abnimmt. Bei niedrigen Ausfüllgraden $A < 50\%$ werden hingegen so viele schlecht dokumentierte Proben in den Datensatz mit aufgenommen, dass ebenfalls mit Genauigkeitseinbußen und erhöhter Varianz zu rechnen ist, auch wenn diese vergleichsweise schwächer ausfallen als bei zu hohen Ausfüllgraden. Tabelle 16 gibt zusätzliche Informationen über die Anzahl an verbleibenden Proben und Herstellungsparametern je nach Ausfüllgrad an.

Tabelle 16: Auswirkungen unterschiedlicher Mindest-Ausfüllgrade auf die Modellgenauigkeit und die Anzahl der Datensätze.

Ausfüllgrad Proben	Ausfüllgrad Herstellungsparameter	Verbleibende Proben	Verbleibende Herstellungsparameter	Ausfüllgrad übriger Datensatz	Genauigkeit RF
0%	0%	163	45	68,0%	0,51
≥10%	≥10%	151	43	74,6%	0,54
≥20%	≥20%	149	39	76,0%	0,56
≥30%	≥30%	148	36	77,1%	0,61
≥40%	≥40%	144	36	82,2%	0,61
≥50%	≥50%	142	36	84,6%	0,63
≥60%	≥60%	132	29	86,4%	0,67
≥70%	≥70%	85	26	91,2%	0,44

Nach Festlegung eines Ausfüllgrad von mindestens 60%, resultiert ein Datensatz von 132 verwendbaren Proben mit 29 ausreichend dokumentierten Herstellungsparametern, wie Tabelle 16 zeigt. Der mittlere Ausfüllgrad des verbliebenen Datensatzes beträgt damit $A = 86,4\%$. Welche Parameter dadurch Einzug in die Auswertung erhalten und welche nicht, ist in Abbildung 77 und Abbildung 78 jeweils für die kategorischen und numerischen Variablen dargestellt. Die Untergrenzen für den Ausfüllgrad sind in beiden Abbildungen mit orange gestrichelten Linien eingezeichnet.

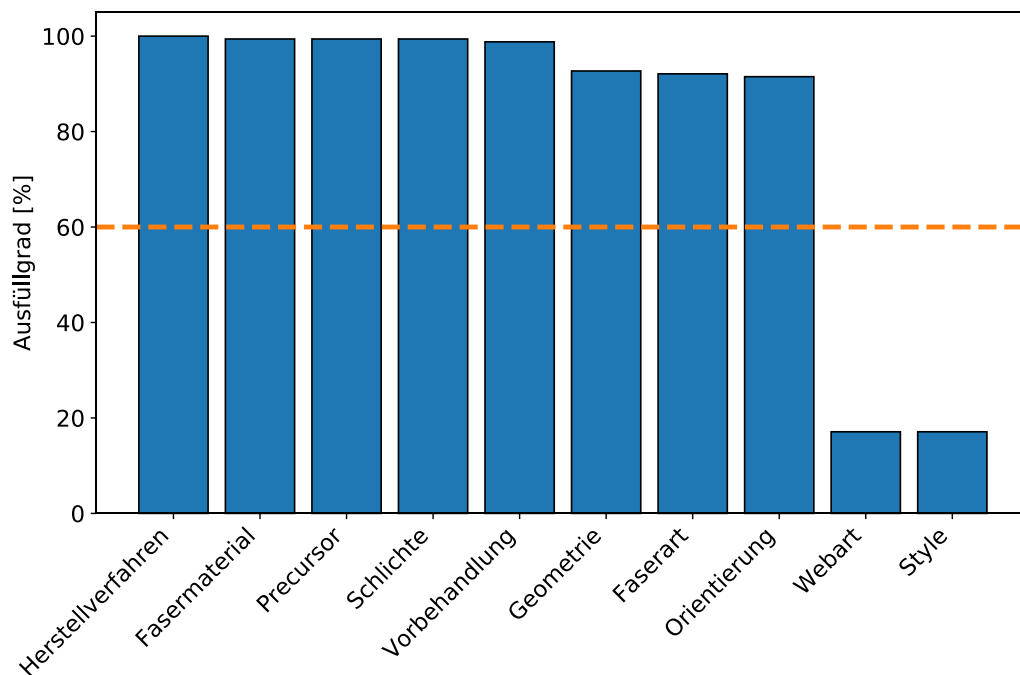


Abbildung 77: Ausfüllgrad der kategorischen Parameter; orange gestrichelte Linie: Grenze unterhalb derer ein Parameter aus dem Datensatz entfernt wird.

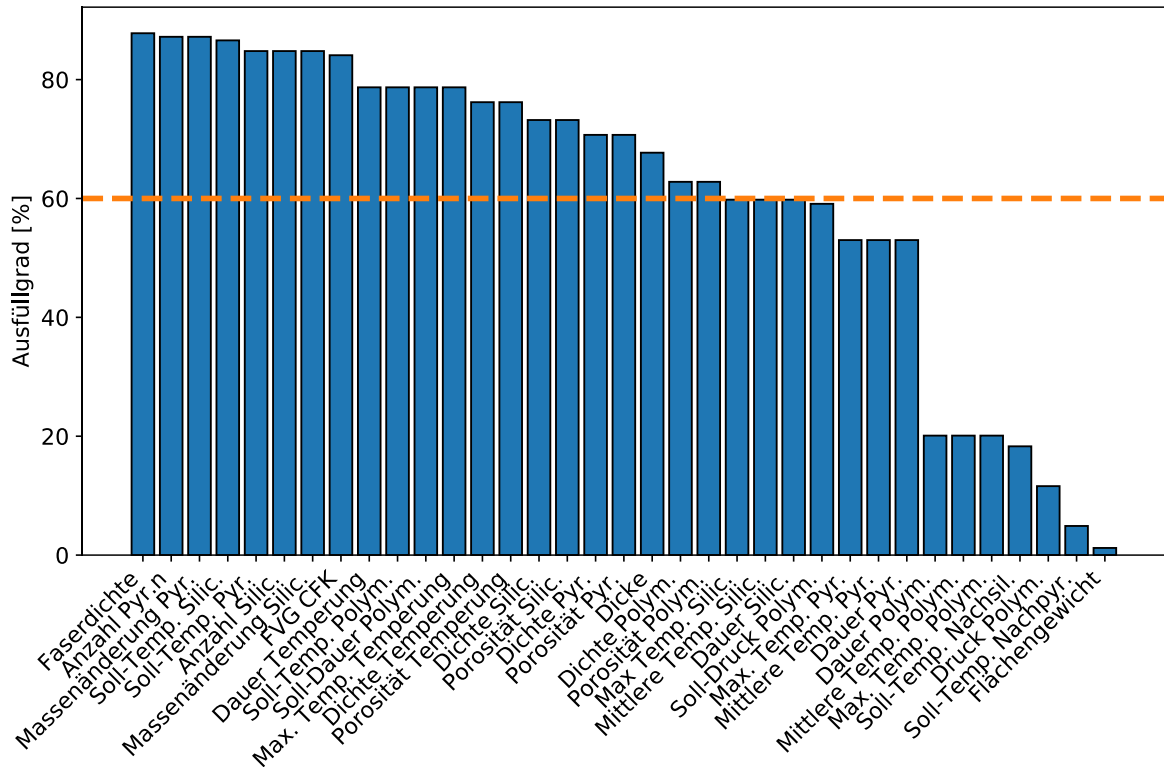


Abbildung 78: Ausfüllgrad der numerischen Parameter; orange gestrichelte Linie: Grenze unterhalb derer ein Parameter aus dem Datensatz entfernt wird.

6.1.2 Multikollinearität

Der originale Datensatz weist mehrere untereinander korrelierende Herstellungsparameter auf, beispielsweise die Porosität im polymerisierten und im getemperten Zustand. Aufgrund dessen wurde in DataTracker eine optionale Funktion implementiert, um mit kollinearen Parametern umzugehen (siehe auch Kapitel 2.5.2). Verglichen wurde jeweils die mittlere Genauigkeit von zehn verschiedenen RandomForest Algorithmen unter gleichen Bedingungen für die verschiedenen Methoden. Letztere bestanden aus der Entfernung von Parametern mit Pearson-Korrelationskoeffizienten von $|PK| > 0,80$, der Entfernung von Parametern mit einem $VIF > 10$, sowie die Beibehaltung aller Parameter unabhängig ihrer Korrelation untereinander. Abbildung 79 gibt Aufschluss über das Ergebnis dieser Untersuchung.

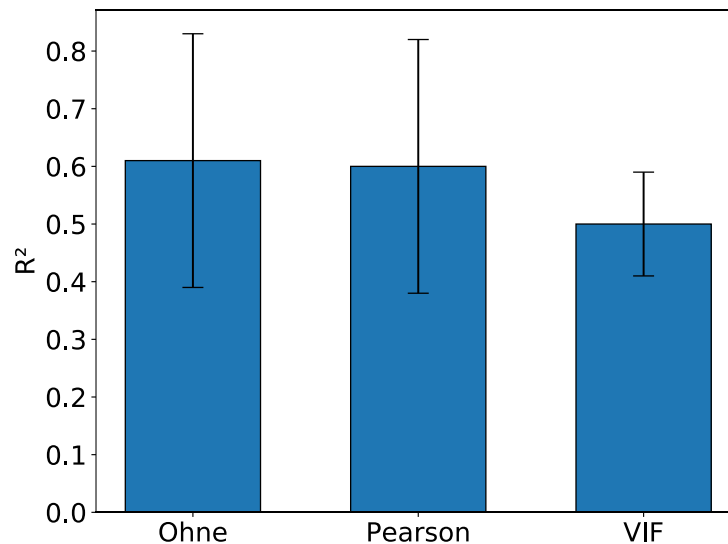


Abbildung 79: Modellgenauigkeiten bei verschiedenen Methoden mit Multikollinearität umzugehen.

Wie erwartet, ließ sich durch das Entfernen korrelierender Eingangsparameter zwar die Varianz innerhalb der Wichtigkeiten der einzelnen Parameter reduzieren, allerdings zog dieser Vorteil eine Senkung der mittleren Modellgenauigkeit nach sich. Deshalb wurde im weiteren Verlauf der Arbeit von einer Entfernung von Parametern aufgrund von Multikollinearität abgesehen. Der Grund für das bessere Abschneiden unter Beibehaltung aller Parameter ist vermutlich der geringe Ausfüllgrad im Datensatz. Auch wenn dieser mit mindestens 60% vorausgesetzt wurde, tragen untereinander korrelierende Parameter offenbar trotzdem noch zum Informationsgewinn bei, da die Chance besteht, dass fehlende Messungen eines Parameters einer Probe durch vorhandene Messungen eines anderen, kollinearen Parameters derselben Probe ausgeglichen werden können. Dies trifft besonders auf die beiden Parameter „Porosität im polymerisierten Zustand“ und „Porosität im getemperten Zustand“ zu.

6.1.3 Entfernen von Ausreißern

Um eine Verzerrung der Daten zu verhindern, wurde der Datensatz während des Preprocessing in DataTracker automatisch auf Ausreißer untersucht. Dazu wurde die bereits in Kapitel 2.5.5 erläuterte Z-Statistik benutzt und die in der Praxis häufig verwendete Schwelle von $|z_i| < 3$ angewandt. Folgende Parameterstudien wurden durchgeführt:

- Kein Entfernen von Ausreißern
- Entfernen von Ausreißern nur innerhalb des wichtigsten Parameters (Dichte im silizierten Zustand, wenn alle Parameter miteinbezogen werden)
- Entfernen von Ausreißern nur innerhalb der CCR
- Entfernen von Ausreißern innerhalb aller Parameter der Silizierung

- Entfernen von Ausreißern innerhalb der zehn wichtigsten Parameter

Dazu wurde jeweils das mittlere Bestimmtheitsmaß R^2 aus zehn verschiedenen Splits für den RandomForest Algorithmus unter sonst gleichen Einstellungen errechnet. Das Ergebnis ist in Abbildung 80 zu sehen.

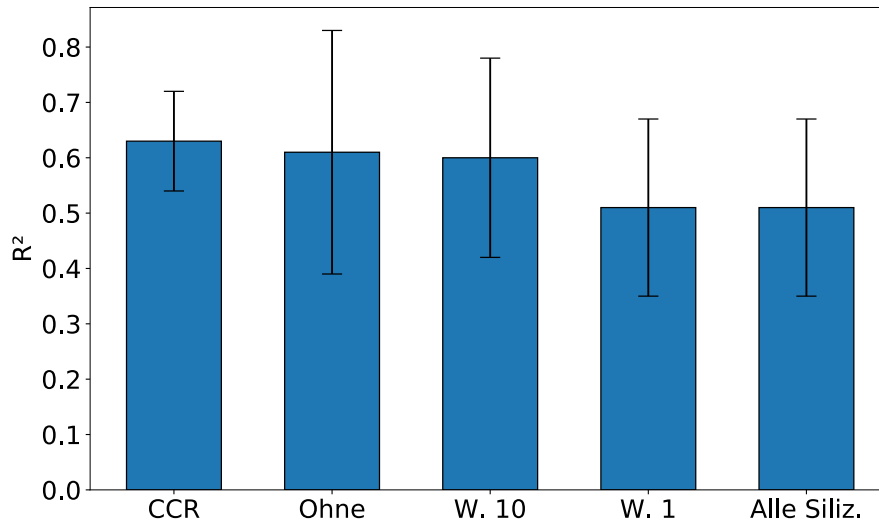


Abbildung 80: Mittlere Modellgenauigkeiten bei verschiedenen Arten mit Ausreißern umzugehen. CCR: Entfernung beim Parameter CCR, Ohne: keine Entfernung, W. 10: Entfernung bei den wichtigsten zehn Parametern, W. 1: Entfernung nur beim wichtigsten Parameter, Alle Siliz.: Entfernung bei allen Parametern der Silizierung.

Es wird ersichtlich, dass das Entfernen von Ausreißern lediglich aus der Spalte CCR zur besten mittleren Genauigkeit und zu den geringsten Standardabweichungen führt. Das vollständige Beibehalten aller Originalspalten (mit Ausreißern) führt zum zweitbesten Ergebnis, allerdings ist hier eine deutlich größere Standardabweichung erkennbar. Dies lässt den Schluss zu, dass Ausreißer innerhalb der unabhängigen Variablen für das ML-Modell nützlich sind, um die Fähigkeit zur Verallgemeinerung zu trainieren und Overfitting vorzubeugen. Innerhalb der Spalte CCR gibt es jedoch eine einzelne Probe, welche eine Z-Statistik > 3 besaß und durch deren Entfernung die Modellgenauigkeit und -standardabweichung verbessert werden kann. Eine weitere mögliche Erklärung für das beobachtete Verhalten ist, dass durch das Entfernen von Ausreißern aus zu vielen Spalten zu wenige Proben im Datensatz verbleiben, wodurch ein Genauigkeitsverlust auftritt.

6.1.4 Schätzen fehlender Werte

Der gesammelte Datensatz wies viele Lücken auf, was darauf zurückzuführen war, dass nicht jeder Prozessschritt bei allen Proben vermessen oder dokumentiert worden war. In Kapitel 2.5.4 wurden bereits unterschiedliche Methoden beschrieben, mit fehlenden Daten umzugehen. In dieser Arbeit wurde vor allem das Schätzen fehlender Werte auf Grundlage von univariaten oder multivariaten Methoden verwendet. Dabei wurden die fehlenden Werte bei den univariaten

Methoden durch den Mittelwert des jeweiligen Herstellungsparameters ersetzt. Bei den multivariaten iterativen Schätzern wurde im Gegensatz dazu die Ähnlichkeit zwischen verschiedenen Proben mit einbezogen, um durch inkrementelles Annähern an eine Konvergenzgrenze Prognosen für die fehlenden Werte zu erhalten. Dabei stellte sich heraus, dass multivariate iterative Schätzer zu realistischeren Ergebnissen, sowie höheren Genauigkeiten der verwendeten Modelle führten. Als Anzahl der Parameter, die in die Ähnlichkeitsbetrachtung von Proben einfließen, wurde dabei stets $k = 5$ gewählt. Dies ist in Abbildung 81 beispielhaft an der Korrelation zwischen CCR und der Dichte der Probe im silizierten Zustand verdeutlicht. Dabei stehen blaue Punkte für vorhandene Messwerte und orange Punkte für geschätzte Werte. Deutlich sichtbar ist, dass die Schätzwerte nur bei den multivariaten iterativen Methoden dem Trend der Messwerte folgen (siehe Abbildung 81 links), bei Verwendung des Mittelwerts der Messwerte für alle fehlenden Datenpunkte wird der Trend stattdessen verschmiert (siehe Abbildung 81 rechts).

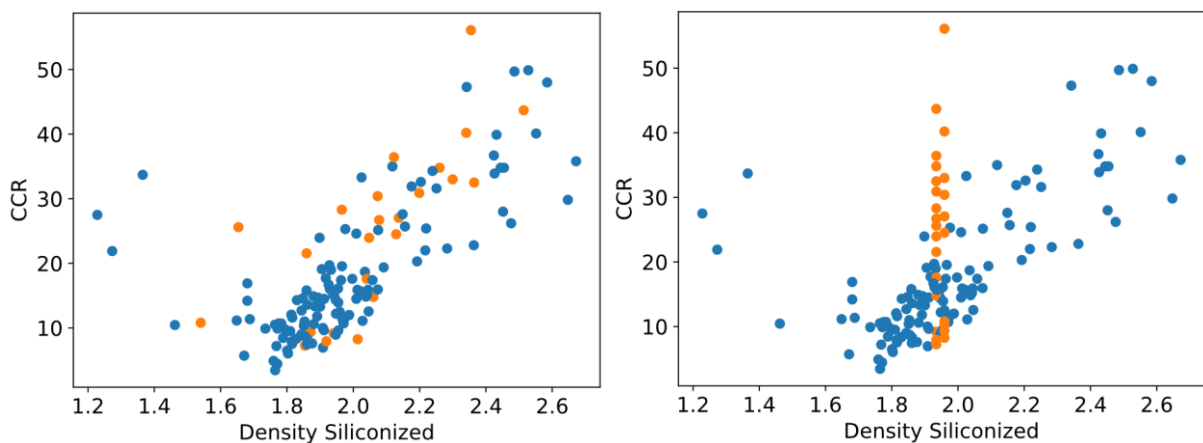


Abbildung 81: Vergleich von multivariater iterativer Schätzung (links) und univariater Schätzung (rechts) unter sonst gleichen Einstellungen. Blau: Messwerte, orange: Schätzwerte.

Durch das Verschmieren von Trends im Falle der simplen univariaten Schätzung ist auch die erreichte Genauigkeit des Machine-Learning Modells niedriger, wie anhand Tabelle 17 ersichtlich ist.

Tabelle 17: Vergleich univariater und multivariater iterativer Schätzmethoden durch das erreichte Bestimmtheitsmaß eines RandomForest bei sonst gleichen Randbedingungen (Mittelwerte aus zehn Durchläufen).

Univariater Mittelwert-Schätzer R^2	Multivariater iterativer Schätzer R^2
0.55 ± 0.13	0.60 ± 0.12

6.1.5 Unterteilung der Daten für KI-Auswertung

Eine weitere Erkenntnis dieser Arbeit bezieht sich auf die Art und Weise, den zugrundeliegenden Datensatz in Test- und Trainingsdaten zu unterteilen. Dies ist, wie bereits

in Kapitel 2.6.1 erläutert, auf zufällige oder stratifizierte Art und Weise möglich. Dabei wurden beide Möglichkeiten verglichen, indem für fünf verschiedene Splits jeweils die Mittelwerte der CCR im Test-Datensatz untersucht wurden (bei jeweils gleichem Random Seed). Um ein Modell sinnvoll auf einen Datensatz anlernen zu können, sollten sowohl Trainings- als auch Testdatensatz repräsentativ für den gesamten Datensatz sein. Um diese Eigenschaft testen zu können, wurde für jeden der fünf Splits pro Methode der Mittelwert der CCR im Test-Datensatz ermittelt und in Tabelle 18 dargestellt. Erkennbar ist, dass der CCR-Mittelwert im Test-Set beim zufälligen Splitten stärker schwankt, als beim stratifizierten Splitten. Das bedeutet, dass durch zufälliges Splitten weniger repräsentative Test-Sets entstehen, als durch stratifiziertes Splitten. Trotzdem hat letztere Methode nur eine geringe Verbesserung der Modell-Genauigkeit zufolge (vergleiche R^2 in Tabelle 18).

Tabelle 18: Vergleich zwischen zufälligem und stratifiziertem Splitten anhand fünf unterschiedlicher Splits.

	Zufälliges Splitten	Stratifiziertes Splitten
Mittelw. CCR je Test-Set	[15.0, 17.0, 18.3, 19.1, 21.3]	[17.6, 18.0, 18.0, 18.0, 18.3]
Mittelw. \pm Std. aller Test-Sets	18.1 \pm 2.4	18.0 \pm 0.3
R^2	0.54	0.56

6.1.6 Beste KI-Modelle

Eines der Kernziele dieser Arbeit war die Auswertung des Datensatzes durch KI-Methoden. Dafür wurden im Auswertetool DataTracker mit DT, RF, NN und LR vier verschiedene Algorithmen für den Benutzer zur Auswahl gestellt, deren Funktionsweisen bereits in Kapitel 2.3 erläutert wurden. In DataTracker wurde auch eine automatische Optimierung der Hyperparameter des jeweiligen ausgewählten Modells implementiert. Dazu wurde durch die Anwendung von Kreuzvalidierung auf den Trainingsdatensatz ein vorgegebener Parameterraum untersucht, um das optimal angepasste Modell für die vorliegende Anwendung zu finden. Der untersuchte Parameterraum für jeden Algorithmus ist in Tabelle 19 gegeben, wobei die Größe des jeweiligen Parameterraums auf 200 Kombinationen festgelegt wurde.

Tabelle 19: Parameterräume, in denen die besten Einstellungen für den jeweiligen Algorithmus gesucht wurden (Hyperparameteroptimierung); aus den angegebenen Kombinationen wurden für jeden Algorithmus 200 ausgewählt und evaluiert.

Algorithmus	Hyperparameterräume
DT	Maximalanzahl untersuchter Merkmale: (1, 2, 3, 4, 5, 6, 7), Maximale Baumtiefe: (4, 5, 6, 7, 8, 9, 10, 11), Minimale Probenanzahl für weiteren Split:

	(2, 6, 10, 14, 18, 22, 26), Minimale Probenanzahl pro Blatt: (1, 3, 5, 7, 9, 11), Splitting Kriterium: (best, random)
RF	Anzahl Bäume: (50, 60, 70, 80, 90, 100, 110, 120, 130), Maximalanzahl untersuchter Merkmale: (1, 2, 3, 4, 5, 6, 7), Maximale Baumtiefe: (4, 5, 6, 7, 8, 9, 10, 11), Minimale Probenanzahl für weiteren Split: (2, 6, 10, 14, 18, 22, 26), Minimale Probenanzahl pro Blatt: (1, 3, 5, 7, 9, 11), Bootstrapping: (with, without)
ANN	Aktivierungsfunktion: (logistic, reLu, tanh), Solver: (lbfgs, adam, sgd), Neuronenkonfiguration in versteckten Schichten: ((5, 5), (15, 15), (5, 5, 5), (50, 50, 50)), Trainingsepochen: (100, 200, 300, 400, 500), Lernrate: (konstant, adaptiv), Initiale Lernrate: (0.001, 0.004, 0.007, 0.01), Alpha: (0.0001, 0.0004, 0.0007, 0.001), Moment: (0.3, 0.6, 0.9)
LR	Lambda: (0.01, 0.05, 0.1, 0.3, 0.5, 0.8, 1)

Um die Genauigkeit der Modelle objektiv vergleichen zu können, wurden für jeden Algorithmus 20 unterschiedliche Trainings-Test-Splits untersucht und deren mittlere Genauigkeit und Standardabweichung berechnet. Das Ergebnis dieser Untersuchung ist in Abbildung 82 dargestellt.

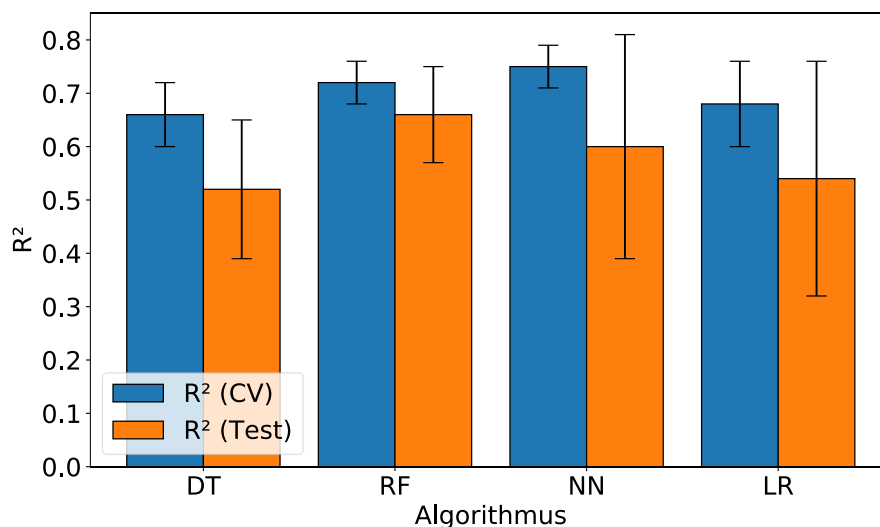


Abbildung 82: Erreichtes Bestimmtheitsmaß R^2 je nach Algorithmus (Mittelwert aus 20 verschiedenen Splits); DT: Decision Tree, RF: RandomForest, NN: Neuronales Netzwerk, LR: Lasso Regression; blau: R^2 während der Kreuzvalidierung, orange: R^2 bei den Test-Sets.

Es lässt sich erkennen, dass der RandomForest Algorithmus im Mittel beim vorliegenden Datensatz am besten abschneidet, gefolgt von Lasso Regression. Dabei kommen optimierte RF Modelle bei 20 unterschiedlichen Splits auf ein mittleres $R^2 = 0,67$ gemessen an den jeweiligen Test-Datensätzen, und einer Standardabweichung von $\sigma = 0,09$, sofern die Daten aller Prozessschritte verwendet wurden.

Die Vorhersagegüte eines RandomForest für einen einzelnen Split kann auch visuell beurteilt werden, indem die vorhergesagten CCR Werte über den wahren CCR Werten des Testdatensatzes aufgetragen werden (siehe Abbildung 83).

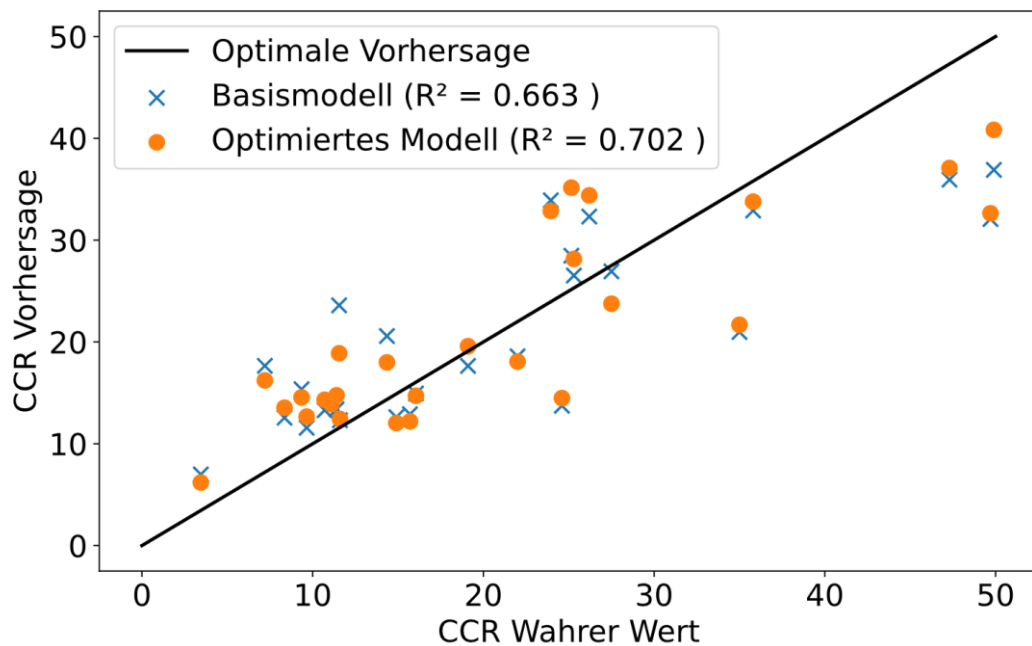


Abbildung 83: Vorhergesagte CCR eines RandomForest über wahrer CCR anhand eines Test-Datensatzes für einen einzelnen Split; blau: Basismodell, orange: optimiertes Modell, schwarz: Linie optimaler Vorhersage ($R^2=1$).

6.1.7 Auswahl wichtigster Parameter

Da die Ermittlung wichtiger Herstellungsparameter eines der Kernziele dieser Arbeit darstellt, kommt auch der verwendeten Bewertungs-Methode für die Wichtigkeit eine tragende Rolle zu. Wie bereits in Kapitel 2.6.5 beschrieben, wurde dies anhand von vier unterschiedlichen Methoden untersucht, welche in DataTracker implementiert wurden. Für jede Methode wurden zehn unterschiedliche RandomForest Modelle trainiert und jeweils Mittelwert und Standardabweichung der Modellgenauigkeit berechnet. Zusätzlich wurde untersucht, wie zeitintensiv deren jeweilige Bestimmung war. Abbildung 84 stellt den Vergleich der verschiedenen Methoden als Balkendiagramm dar. Für (vorwärts gerichtete) sequenzielle Feature-Selection und Recursive Feature Elimination (RFE) wurde jeweils eine Maximalanzahl von fünf Herstellungsparametern vorgegeben.

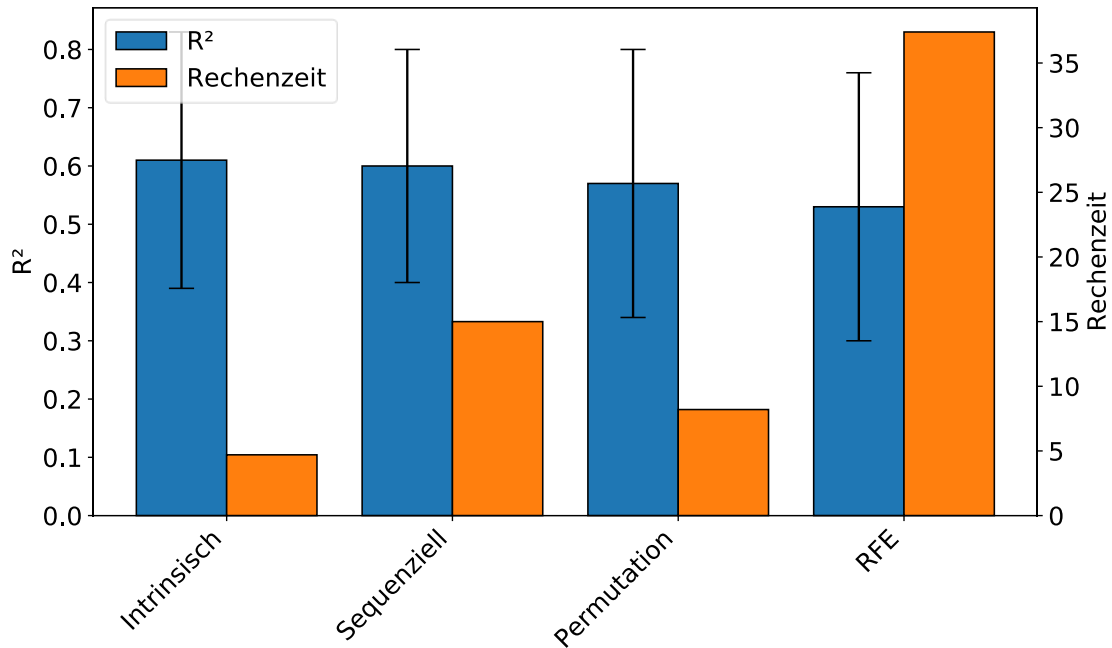


Abbildung 84: Modellgenauigkeit je nach verwendeter Feature-Selection Methode inklusive Rechenzeitanforderungen; die erzielten Genauigkeiten wurden aus zehn verschiedenen Splits gemittelt.

Insgesamt lässt sich feststellen, dass die Unterschiede zwischen den vier Methoden gering ausfallen. Die höchsten Modellgenauigkeiten werden dabei durch intrinsische und vorwärts gerichtete sequenzielle Methoden erreicht, wobei die intrinsische Methode deutlich schnellere Rechenzeiten aufweist. Letzteres ist dadurch begründet, dass die Bewertung der Wichtigkeiten im Falle des RandomForest-Algorithmus als natürlicher Prozess durch das ML-Modell selbst stattfindet und daher keine zusätzlichen Berechnungsschritte erfolgen müssen. Aufgrund der beiden genannten Vorteile wurde die Bewertung der Herstellungsparameter in dieser Arbeit durch Modell-intrinsische Methoden anhand eines unabhängigen RandomForest Algorithmus vorgenommen.

6.1.8 Verbesserung der Genauigkeit

Die im Rahmen dieser Arbeit erstellten KI-Modelle konnten die CCR anhand der Herstellungsparameter nur teilweise erklären ($R^2 = 0,67$). Daraus lässt sich schließen, dass es noch Produktions-Einflüsse gibt, welche in der bisherigen Auswertung nicht berücksichtigt sind. Dies kann folgende mögliche Gründe haben:

- Fehlende Messung wichtiger Herstellungsparameter
- Ausschluss von wichtigen Herstellungsparametern aufgrund zu niedrigem Ausfüllgrad oder zu geringer Variabilität
- Zu wenig Trainingsdaten (bestimmte Kombinationen von Herstellungsparametern zu selten in Datensatz)

Eine Herausforderung dieser Arbeit stellte die in manchen Fällen schlechte Dokumentation von Herstellungsparametern dar. Insgesamt wurden 16 der 45 gemessenen Parameter verworfen, da diese in weniger als 60% der Fälle dokumentiert wurden. Dies entspricht einer Quote von 36% der Produktionseinflüsse, deren Auswirkungen nicht erfasst werden konnten. Wie bereits in Kapitel 4.5.1 beschrieben, wurde der Ausfüllgrad von 60% für jeden Parameter vorausgesetzt, da sonst auch die angewandten Schätzmethoden für fehlende Werte zu starke Unsicherheiten aufwiesen. Zehn der verbleibenden 29 Parameter, welche ausreichend dokumentiert wurden, besaßen allerdings kaum Variation, sodass auch die Auswirkungen dieser Parameter auf die CCR nicht festgestellt werden konnten. Damit konnten insgesamt nur etwa 42% der ursprünglich aufgenommenen Herstellparameter tatsächlich untersucht werden. Da sich unter den ausgeschlossenen Parametern auch einige in empirischen Tests als wichtig eingestufte Parameter befinden, ist davon auszugehen, dass sich die Modellgenauigkeit R^2 durch die Miteinbeziehung dieser Größen verbessern lässt. Hier sind insbesondere die Information über die Faserentschlichtung sowie die maximale Pyrolysetemperatur oder die Anzahl der Pyrolysen als hilfreiche Kennwerte zu nennen.

Obwohl im bisherigen Prozess bereits eine große Anzahl an Parametern berücksichtigt wurden, ist es dennoch möglich, dass weitere wichtige Einflussfaktoren nicht erfasst wurden. Beispielsweise unterliegen Rohmaterialien wie Harze oder Fasern bei längerer Lagerung einem Alterungsprozess, welcher die chemischen und mechanischen Eigenschaften verändert. Insbesondere bei Harzlösungen spielen Lagerdauer, Umgebungstemperatur, UV-Einstrahlung, Verunreinigungen und Luftfeuchtigkeit eine große Rolle, sodass auch die Verwendung desselben Harzsystems zu unterschiedlichen mechanischen Eigenschaften (z.B. Viskosität) führen kann [100, 101]. Bei Fasern ist der Zustand der Schlichte ein wichtiges Merkmal für die spätere Faser-Matrix Anbindung im CFK. Wurde ein Entschlichtungsprozess vorgenommen, so wurde dies zwar im Laufzettel vermerkt, allerdings spielt hier zusätzlich die Entschlichtungs-Temperatur und -dauer eine Rolle. Letztere beide Informationen fehlten allerdings im Datensatz. All die genannten Kriterien haben Einfluss auf die Interphasenstärke, welche ihrerseits die Entwicklung der Mikrostruktur stark beeinflusst [102, 1].

Weiterhin bleibt festzuhalten, dass der Datenpool von 132 verwertbaren Proben für den CMC-Bereich zwar relativ groß erscheinen mag, jedoch angesichts der großen Zahl an Herstellungsparametern und Kombinationsmöglichkeiten vergleichsweise klein für ein Machine-Learning Problem ist. So kommt es beispielsweise vor, dass für gewisse Faser- oder Harztypen nur so wenige Datenpunkte vorhanden sind, dass eine statistische Auswertung wenig

Aussagekraft besitzt, zumal sich diese Datenpunkte auch noch in anderen Herstellungsparametern unterscheiden.

6.1.9 Zusammenfassung der methodischen Erkenntnisse

Abschließend ist festzuhalten, dass die in Tabelle 20 aufgelisteten Methoden und Einstellungen über zehn Durchläufe gemittelt zu den besten Ergebnissen führten.

Tabelle 20: Beste Einstellungen und Methoden für den in dieser Arbeit vorliegenden Datensatz.

Methode	Wert
Algorithmus	RandomForest
Splitting-Methode	stratifiziert
Vorausgesetzter Ausfüllgrad	60% (Zeilen und Spalten)
Feature-Selection Methode	Modell-intrinsisch (RF)
Schätzmethode	iterativ
Ausreißer entfernen (3σ)	Ja (CCR)
Multikollinearität behandeln	Nein

6.2 Materialwissenschaftliche Erkenntnisse

In diesem Kapitel werden ausschließlich die materialwissenschaftlichen Erkenntnisse diskutiert. Dabei geht es sowohl um Herstellungsparameter mit signifikantem Einfluss auf die CCR, als auch um solche mit vernachlässigbarem Einfluss, oder solche, bei denen aufgrund zu geringer Variabilität keine Untersuchung möglich war.

6.2.1 Allgemeines

Die große Anzahl von 45 Herstellungsparametern wurde bereits im Preprocessing auf 19 verringert, da 16 davon weniger als 60% Ausfüllgrad besaßen und weitere 10 eine zu geringe Variabilität aufwiesen. Festzuhalten ist hier, dass mögliche Einflüsse dieser Parameter auf die CCR grundsätzlich nicht festgestellt werden können. Von den 19 untersuchbaren Parametern korrelierten nur sieben signifikant mit der CCR, während die restlichen zwölf kaum Korrelation aufwiesen. Eine schematische Übersicht ist in Abbildung 85 dargestellt. Parameter mit zu geringer Variation werden eingehend in Kapitel 6.2.4 untersucht.

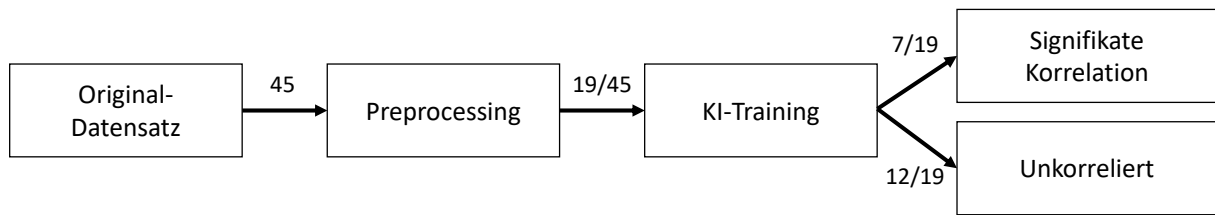


Abbildung 85: Schematische Darstellung der Ergebnisauswertung; Zahlen auf den Pfeilen zeigen die Anzahl der Herstellungsparameter an, welche im jeweiligen Schritt noch vorhanden sind (Zähler: aktuelle Anzahl, Nenner: Gesamt-Anzahl vom vorherigen Schritt).

6.2.2 Größte Einflüsse auf die CCR unter allen Parametern

Von den untersuchbaren signifikanten sieben Herstellungsparametern wurden je nach Modell nur etwa ein bis fünf Parameter beibehalten, ohne dass sich die Genauigkeit der Modelle dabei verschlechterte, die Rechenzeit sich aber verkürzte. Die gefundenen Korrelationen waren je nach Unterteilung der Daten in Test- und Trainings-Datensatz leicht unterschiedlich. Deshalb wurde zur Bestimmung der Merkmals-Wichtigkeit der Mittelwert für R^2 über 20 verschiedene Splits und Trainingszyklen eines RF-Algorithmus herangezogen. Das Ergebnis ist in Abbildung 86 dargestellt.

Je unwichtiger ein Merkmal war, und je höher seine Standardabweichung, desto eher konnte es aus dem Modell ausgeklammert werden, ohne dass dessen Vorhersagegenauigkeit darunter litt. Um beurteilen zu können, ab wann ein Merkmal aus dem Modell entfernt werden sollte, wurde ein künstliches Zufalls-Merkmal in den Datensatz eingefügt. Dieses bestand aus Zufallszahlen und korrelierte daher überhaupt nicht mit der CCR (oder nur in seltenen Fällen durch Zufall). Da auch für dieses Pseudo-Merkmal vom Algorithmus eine Wichtigkeit berechnet wird, konnte somit eine „Rauschgrenze“ festgelegt werden, um unwichtige Merkmale zu entfernen.

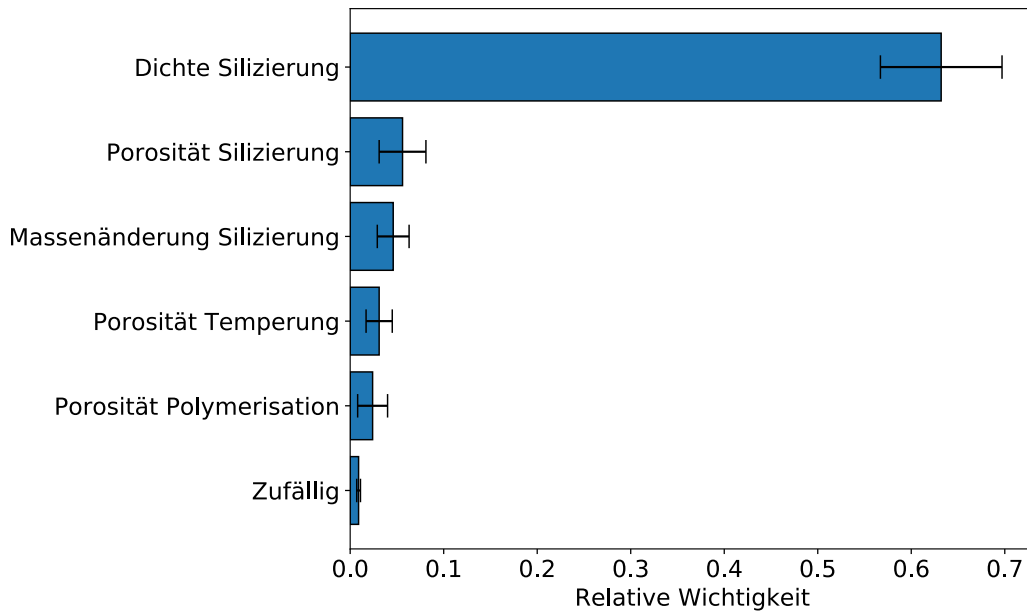


Abbildung 86: Mittlere relative Wichtigkeiten der bedeutendsten Herstellungsparameter für die Vorhersage der CCR (20 Messungen). Ein Merkmal mit Zufallszahlen wurde hinzugefügt, um die Rauschgrenze festzulegen (siehe „Zufällig“).

Die mit Abstand stärkste Korrelation mit der CCR hatte die Rohdichte mit 63% relativer Wichtigkeit, gefolgt von Porosität und Massenänderung, alle nach oder während des Prozessschrittes „Silizierung“. Das Kontrollmerkmal kam nur auf etwa 1% relative Wichtigkeit. Die starke Korrelation zwischen CCR und Dichte war zu erwarten, da Proben, welche eine ausgeprägte Einzelfasersilizierung aufweisen (entspricht hoher CCR), auch signifikant mehr Silizium aufnehmen, als Proben mit Blockstruktur (entspricht niedriger CCR). Auch wenn sich diese Information gut als Plausibilitätsüberprüfung für die Modelle eignete, ergab sie keinen besonderen Mehrwert hinsichtlich der Kernfragestellungen dieser Arbeit.

6.2.3 Größte Einflüsse auf die CCR unter Ausschluss der Silizierung

Interessantere Zusammenhänge ergaben sich, wenn der Prozessschritt Silizierung ausgeklammert, und das Modell anhand des so entstandenen Datensatzes neu angelernet wurde (vergleiche Abbildung 87). In diesem Fall kristallisierte sich die Porosität im getemperten Zustand mit ca. 27% relativer Wichtigkeit als bedeutendstes Vorhersagekriterium für die CCR heraus, gefolgt jeweils von den Porositäten im polymerisierten und im pyrolysierten Zustand. Außerdem hatte auch die Wahl der verwendeten Faser einen Einfluss auf die resultierende Mikrostruktur. Alle anderen Merkmale besaßen nur geringe relative Wichtigkeiten von unter 5% und näherten sich damit der Rauschgrenze des Kontrollmerkmals. Für letzteres wurde eine relative Wichtigkeit von ca. 2% berechnet.

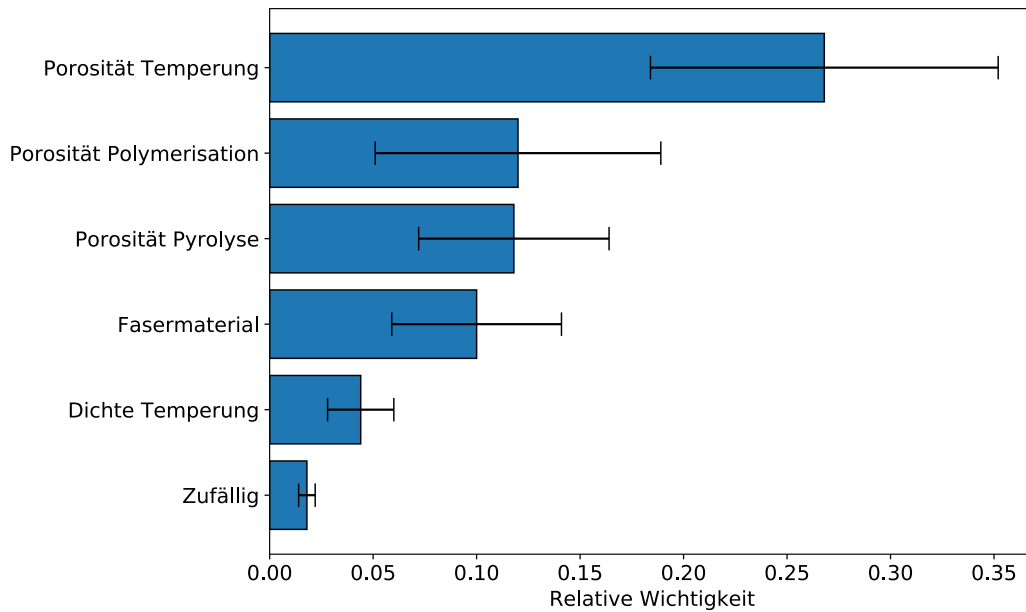


Abbildung 87: Relative Wichtigkeiten der bedeutendsten Herstellungsparameter für die Vorhersage der CCR unter Ausschluss des Silizierungsprozesses (20 Messungen). Ein Merkmal mit Zufallszahlen wurde hinzugefügt, um die Rauschgrenze festzulegen (siehe „Zufällig“).

Da sich die relative Wichtigkeit aller Merkmale immer zu 100% aufsummiert, ist ein Wert von 67% nicht pauschal besser, als ein Wert von 27%, wenn man verschiedene Modelle vergleicht. Herausragend hohe Werte bedeuten schlicht, dass der jeweilige Parameter, relativ zu den anderen betrachtet, wichtiger ist. Im Modell ohne den Prozessschritt „Silizierung“ ist folglich eine größere Anzahl an Parametern in hohem Maße an der CCR-Prognose beteiligt, während sich das Modell mit inkludierter Silizierung bei seinen Aussagen fast ausschließlich auf die Dichte im silizierten Zustand stützt. Weiterhin muss jedoch festgehalten werden, dass durch Ausschluss der Silizierungsdaten insgesamt niedrigere Modellgenauigkeiten erreicht wurden, was mit einer schlechteren Erklärbarkeit der CCR anhand der Herstellungsparameter gleichzusetzen ist. Dies ist in Tabelle 21 dargestellt.

Tabelle 21: Mittleres erreichtes R^2 aus 20 RadomForest Modellen für unterschiedliche Splits des Datensatzes mit und ohne Einbeziehung der Silizierung.

R^2 (RF, Silizierung eingeschlossen)	R^2 (RF, Silizierung ausgeschlossen)
0.67 ± 0.09	0.51 ± 0.10

Trägt man im Modell ohne Silizierung die CCR über der Porosität im getemperten Zustand (wichtigster Parameter) auf, erkennt man den Grund für das niedrigere Bestimmtheitsmaß von $R^2 \approx 0,5$: die Messwerte weisen eine starke Streuung auf, sodass ein Trend kaum zu erkennen ist. Hier wird nun die Stärke von KI-Algorithmen erkennbar: das gleichzeitige Einbeziehen einer Vielzahl an Parametern. So kann der Trend zwischen CCR und Porosität allein durch die

zusätzliche Unterscheidung nach verwendetem Fasermaterial deutlich besser sichtbar gemacht werden, wie Abbildung 88 zeigt. Dazu wurden aus Gründen der besseren Übersicht nur die Datenpunkte der beiden häufigsten beiden Fasermaterialien, T800 und HTA, sowie die Datenpunkte der häufigsten vier Matrixsysteme, JK60, XP60, MF43 und MF13 eingezeichnet.

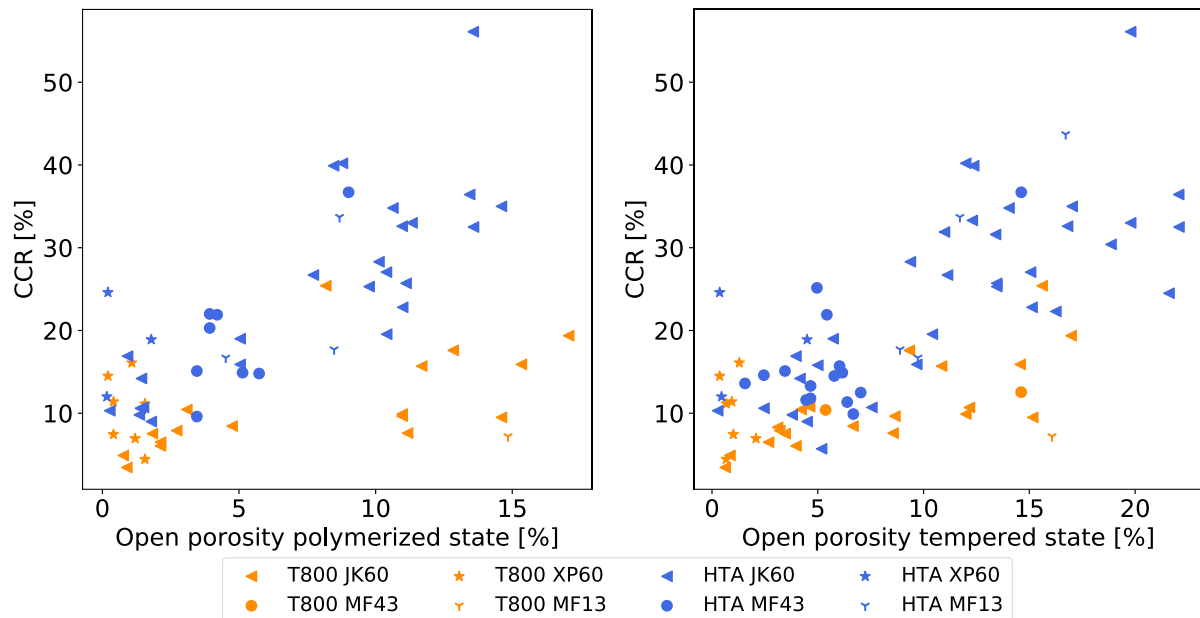


Abbildung 88: CCR über Porosität im polymerisierten Zustand (links) und im getemperten Zustand (rechts), aufgeteilt nach verwendeten Faser- und Matrixsystemen.

Es ist deutlich erkennbar, dass die Verwendung von Fasern des Typs T800 generell zu niedrigeren CCR-Werten führt, als die Verwendung von HTA-Fasern. Dies deckt sich mit den Beobachtungen aus Breede [3], wo ebenfalls ein geringerer Grad an Faserdegradation bei der Verwendung von T800-Fasern im Vergleich zur Verwendung von HTA-Fasern festgestellt wurde. Bei HTA-Fasern war außerdem ein deutlicher linearer Zusammenhang zwischen höherer Porosität und höheren CCR-Werten erkennbar, welcher bei T800-Fasern nur sehr schwach ausgeprägt zu beobachten war. Dies konnte sowohl für die Porositätsmessungen im polymerisierten als auch im getemperten Zustand gezeigt werden. Die Wahl des Harzsystems spielte dabei keine Rolle.

Bereits in Kapitel 5.1 konnte anhand von Mikrostruktursimulationen gezeigt werden, dass eine höhere Anzahl von Defekten in der Matrix bei gleichzeitiger Verwendung von HTA-Fasern zu einem fein verästelten Risssystem führt, welches auch bei Proben mit hoher CCR nach der Pyrolyse vorliegt. Für T800-Fasern wurde dieser Trend nicht beobachtet, allerdings entstanden hier tendenziell größere Risse als bei Laminaten mit HTA-Fasern. Beide Phänomene stimmen mit den gefundenen Korrelationen der datenbasierten Modelle überein. Wie bereits in

Kapitel 5.1 beschrieben, trägt vermutlich der unterschiedliche Durchmesser der Fasermaterialien maßgeblich zum unterschiedlichen Verhalten bei.

Ein weiterer möglicher Grund für den Einfluss des Fasermaterials auf die CCR könnte in unterschiedlicher Interface-Festigkeit zwischen Fasern und Matrix begründet sein. Bereits in Brandt et al. [103], Schulz [102] und Schulte-Fischedick [1] konnte gezeigt werden, dass die Anbindung von Fasern und Matrix einen großen Einfluss auf die Mikrostruktur und insbesondere die Einzelfasersilizierung hat. Dabei wurde in Schulz [102] gezeigt, dass eine schwächere Faser-Matrix Anbindung zu einer höheren Einzelfasersilizierung führt, und andersherum. Deshalb wurde vermutet, dass HTA-Fasern eine schwächere Anbindung zu den getesteten Harzen aufweisen, als T800 Fasern. Allerdings führt die Literatur keinen Vergleich der Interface-Festigkeit zwischen T800 und HTA Fasern bei gleicher Wahl des Harzsystems auf. Zur Überprüfung dieser Hypothese bedarf es demnach entsprechender Versuche in statistisch ausreichender Menge, welche jedoch den Rahmen dieser Arbeit übersteigen würden. Aufgrund dessen konnte die Hypothese in dieser Arbeit nicht abschließend geprüft werden.

6.2.4 Geringe Variabilität

Generell ist es erwähnenswert, dass einige der aufgenommenen Parameter zwar gut dokumentiert, allerdings kaum variiert wurden. Dieser Fall ist insofern problematisch, als dass sich daraus nicht ableiten lässt, ob ein Einfluss auf die CCR besteht oder nicht. Dabei ist es nicht zwingend notwendig, Parameter mit geringer Variabilität manuell auszusortieren. Ein großer Vorteil der KI-Methoden ist ja gerade, dass diese selbstständig aus Daten lernen können, und daher unwichtige Variablen bei ihren Prognosen ignorieren. Die Notwendigkeit für eine Bewertung der Variabilität ist vielmehr in der Interpretation der Ergebnisse zu sehen. Wird ein Herstellungsparameter von den KI-Modellen als unwichtig eingestuft, könnte es einerseits sein, dass dieser tatsächlich unwichtig ist. Andererseits wäre es auch möglich, dass die der KI zur Verfügung stehenden Trainingsdaten lediglich den Anschein erwecken lassen, dass der Parameter unwichtig sei, beispielsweise weil er überhaupt nicht variiert wurde. Auch diese Information ist für den Ingenieur wichtig, da sie die Schlussfolgerung von „Parameter ist unwichtig“ in „Parameter wurde nicht ausreichend untersucht“ ändert. Aus diesen Gründen wurden alle Eingangsparameter anhand einer Variabilitätsprüfung untersucht.

Geringe Variabilität in den Eingangsparametern wurde insbesondere bei Prozesstemperaturen und -zeiten beobachtet. Beispielsweise gab es bei der Pyrolyse im gesamten Datensatz fast ausschließlich Proben, welche über sieben Tage bei 1650 °C prozessiert wurden. Auch bei der Silizierung wurden beinahe ausschließlich über zwei Tage

andauernde Zyklen bis 1650 °C gefahren. Die Temperung unterschied sich zwar in der Prozesszeit zwischen 4-8 Stunden, jedoch wurden hier die Temperaturen auch nur geringfügig zwischen 220 °C und 240 °C variiert. Die einzig große Variation von Prozesstemperaturen zwischen 150°C und 240 °C lag im Polymerisationsprozess vor, allerdings waren die Temperaturen hier auf bestimmte Harze ausgelegt, sodass auch unter Proben desselben Harzes keinerlei Diversifikation stattfand. Allgemein ist festzuhalten, dass Korrelationen umso schlechter gefunden werden können, je weniger Variation innerhalb eines Herstellungsparameters vorliegt. Um herauszufinden, welche der 45 Herstellungsparameter aussagekräftig untersucht werden können, wurden für jeden von ihnen verschiedene statistische Kennzahlen ermittelt, deren Erläuterungen bereits in Kapitel 2.5.8 geliefert wurden.

Zur Bestimmung der Variabilität der kategorischen Variablen wurde die normierte Entropie als einziges Kriterium herangezogen. Für numerische Variablen wurde neben der normierten Entropie auch die relative Standardabweichung σ_{rel} berechnet, da in diesem Fall, wie bereits zuvor beschrieben, die Entropie alleine kein aussagekräftiges Kriterium bietet.

Variabilität der kategorischen Herstellungsparameter

Bei den kategorischen Parametern wurde ein Entropie-Wert von $H = 0,47$ als Untergrenze vorausgesetzt, da dieser Wert bei einer 2-parametrischen Variable einer Aufteilung von 90%/10% entspricht. Noch stärker unausgewogene Verteilungen führen daher zum Ausschluss des jeweiligen Parameters. Durch Anwenden dieses Schwellwerts wurde deutlich, dass drei von acht kategorischen Herstellungsparametern aufgrund unzureichender Variabilität aussortiert werden mussten, wie Abbildung 89 zeigt. Dies betraf die Information über die Faserart, die Faservorbehandlung, als auch die Information über die Faserentschlichtung. So waren beispielsweise von den 132 verwertbaren Proben nur sieben entschlichtet, wodurch extrem wenige Trainingsbeispiele für die KI vorlagen. Aufgrund der geringen Anzahl wird diesem Parameter während der Feature Selection eine geringere Wichtigkeit zugeordnet, da sich die gewichtete Varianz durch ihn nur wenig verbessern lässt. Bei der Faserart besaßen 110 der 132 Proben die Kategorie „Roving“ und bei der Vorbehandlung gab es nur 9-mal den Eintrag „Ja“.

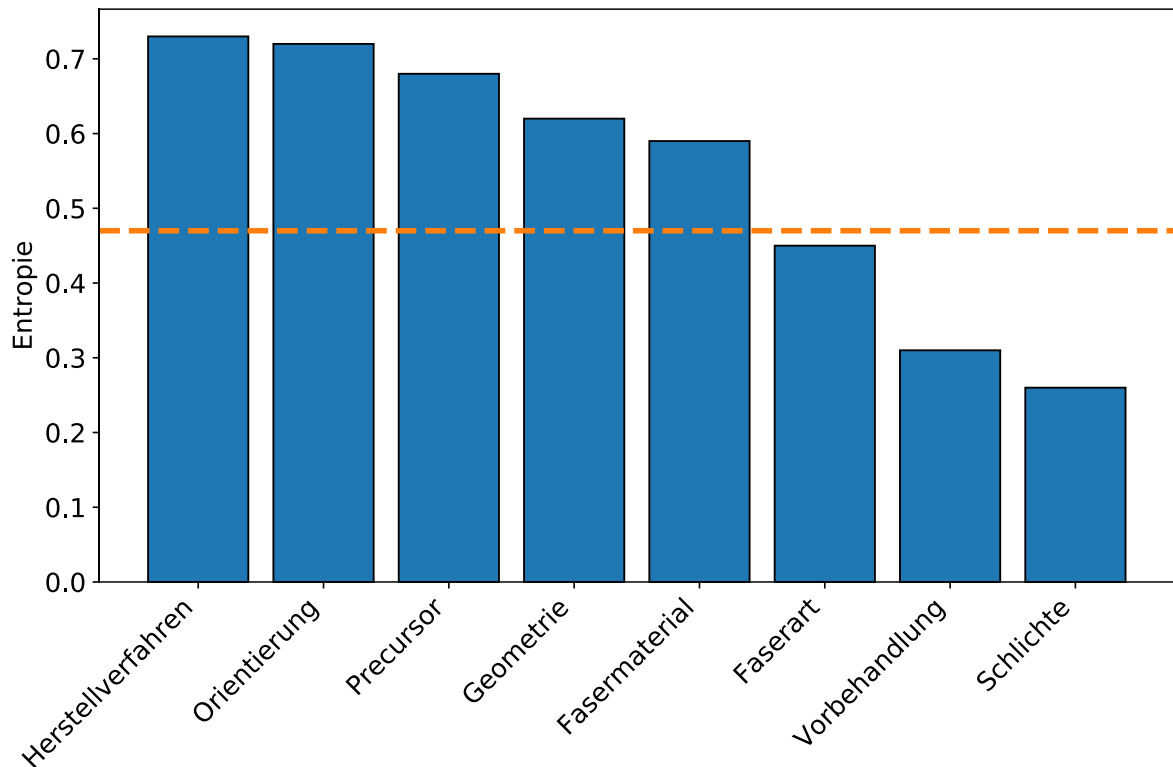


Abbildung 89: Normierte Entropie innerhalb der kategorischen Herstellungsparameter mit einem Ausfüllgrad über 60%; unterhalb einer Grenze von $H=0,47$ wurden Herstellungsparameter aufgrund unzureichender Variabilität entfernt.

Da die praktische Erfahrung jedoch zeigt, dass die Faser-Entschlichtung ein wichtiger Indikator für die CCR ist, wird deutlich, dass hierbei der zugrundeliegende Datensatz den Grund für die geringe Beachtung dieses Parameters darstellt. Hier könnte in zukünftigen Studien darauf geachtet werden, vermehrt Proben mit entschlichteten Fasern in den Datensatz aufzunehmen. Die Information über die Faserart, also beispielsweise ob zu Beginn ein Gewebe, ein Roving oder ein Prepreg vorlag, könnte aufgrund der unterschiedlichen Lösungsmittelgehalte ebenfalls einen Einfluss auf die Mikrostruktur haben; allerdings ist dieser Einfluss vermutlich geringer, als der Einfluss einer Entschlichtung.

Variabilität der numerischen Herstellungsparameter

Die Variabilität der numerischen Parameter wurde nicht nur anhand der Entropie, sondern zusätzlich anhand der relativen Standardabweichung σ_{rel} bewertet. Dies hatte den Grund, dass in manchen Fällen zwar eine ausgewogene Verteilung der Werte vorlag, diese aber insgesamt zu wenig um den Mittelwert schwankten, sodass auch hier nicht von einer hohen Variabilität in den Daten gesprochen werden konnte. Zur Berechnung der normierten Entropie wurden die Verteilungen der Herstellungsparameter jeweils durch zehn Bins angenähert. In Abbildung 90 sind die beiden eben beschriebenen Kennzahlen über jedem Parameter aufgetragen. Zusätzlich

sind Untergrenzen für Entropie (orange gestrichelte Linie) und relative Standardabweichung (blau gestrichelte Linie) eingezeichnet, auf die später im Text noch genauer eingegangen wird.

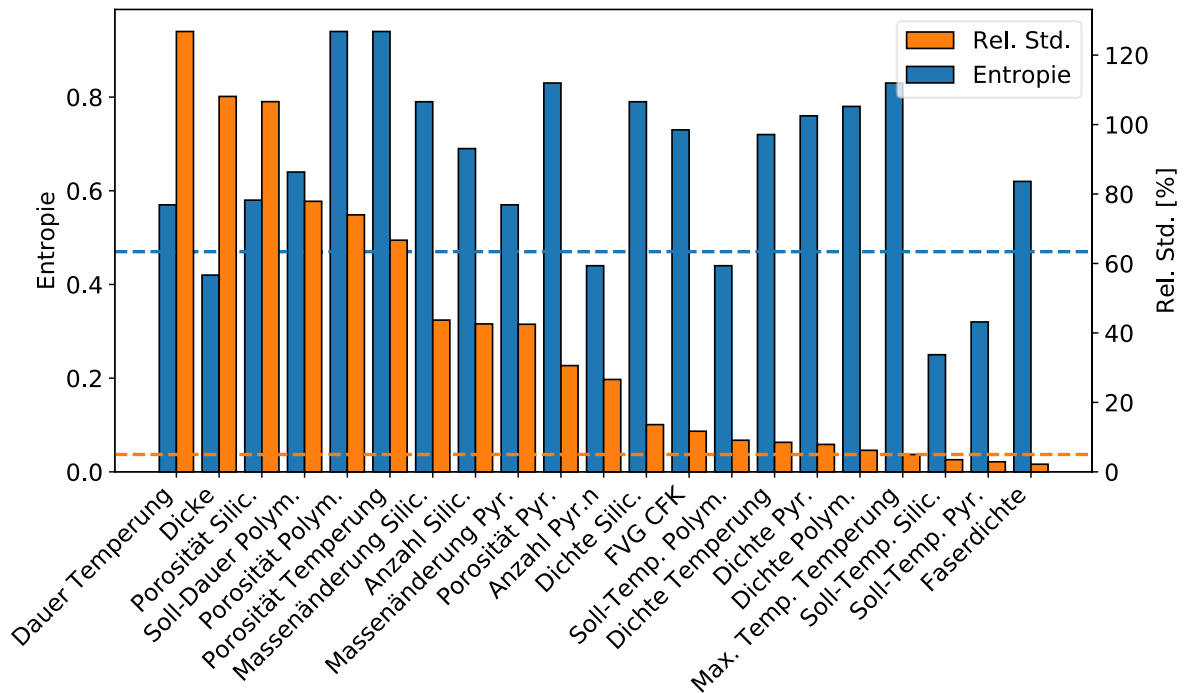


Abbildung 90: Relative Standardabweichung und Entropie der numerischen Herstellungsparameter mit einem Ausfüllgrad über 60%; blau gestrichelte Linie: Untergrenze für relative Standardabweichung; orangefarbene Linie: Untergrenze für Entropie.

Bei einigen Parametern, beispielsweise der Dicke der Proben, lag zwar eine hohe relative Standardabweichung von $\sigma_{rel} = 108\%$ vor, allerdings konnte hier aufgrund der sehr niedrigen Entropie von $H = 0,42$ trotzdem nicht von einer ausreichenden Variabilität ausgegangen werden. Vielmehr lässt sich daraus schließen, dass zwar sehr unterschiedliche Probendicken im Datensatz vorhanden waren, deren statistische Häufigkeiten allerdings sehr unausgewogen war. Betrachtet man nun auch die minimale und maximale Probendicke, welche bei $d_{min} = 2mm$ bzw. $d_{max} = 35,6mm$ lag, sowie den Mittelwert mit $\bar{X} = 4,2mm$, wird ersichtlich, dass der Mittelwert deutlich näher am Minimum liegt. Demnach wurden sehr viel mehr dünne als dicke Proben getestet, was sich folglich in einer geringen Entropie niederschlägt. Entsprechend ist davon auszugehen, dass die Auswirkung der Probendicke auf die CCR aufgrund zu geringer Variabilität nicht aussagekräftig untersucht werden konnte. Andererseits kann auch nicht ausschließlich die Entropie als Maßstab für die Variabilität verwendet werden, da in manchen Fällen zwar ein ausgewogenes Verhältnis zweier Bins vorlag, jedoch beide Bins so nahe beieinander lagen, dass kaum Streuung um den Mittelwert vorlag. Dies ist beispielsweise beim Parameter „Faserdichte“ der Fall, bei welchem ein ausreichender Entropiewert von $H = 0,62$ einer extrem kleinen relativen Standardabweichung von $\sigma_{rel} = 2,2\%$ gegenübersteht.

Folglich musste aus beiden Kennzahlen ein Bewertungsprozedere geschaffen werden, durch welches Herstellungsparameter als „aussagekräftig untersuchbar“ oder „nicht aussagekräftig untersuchbar“ bewertet werden konnten. Das Bewertungsprozedere wurde so definiert, dass ein Entropiewert von $H > 0,47$ erreicht werden musste, ansonsten wurde der Parameter direkt als „nicht aussagekräftig untersuchbar“ eingestuft. Allerdings mussten alle Parameter neben dem Entropie-Grenzwert zusätzlich einen Grenzwert bzgl. relativer Standardabweichung von $\sigma_{rel} > 5\%$ erreichen, ansonsten wurden sie ebenfalls als „nicht aussagekräftig untersuchbar“ eingestuft. Abbildung 91 beschreibt die schematische Vorgehensweise für diese Bewertung.

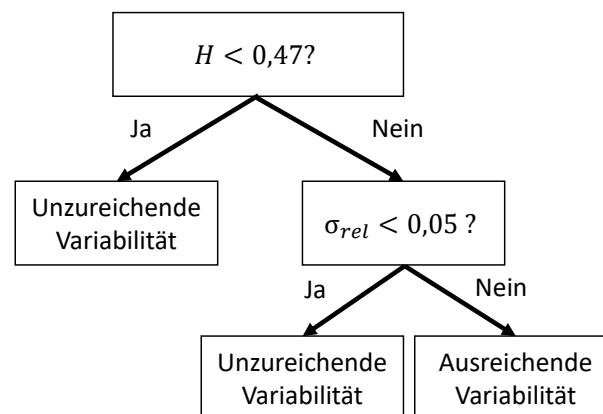


Abbildung 91: Schematisches Vorgehen bei der Bewertung der Variabilität der numerischen Parameter.

Durch Anwenden dieses Schemas wurden sieben der 21 numerischen Parameter aufgrund zu geringer Diversifikation als „nicht aussagekräftig auswertbar“ deklariert. Hier würde es sich in zukünftigen Herstellungsprozessen anbieten, eine größere Variation einfließen zu lassen, um auch die Einflüsse dieser Parameter erfassen zu können. Es ist zu vermuten, dass besonders die maximale Pyrolysetemperatur einen Einfluss das entstehende Rissystem und damit die CCR hat. Auch die Anzahl der Nachinfiltrationen und damit einhergehenden Nachpyrolysen (Parameter „Anzahl Pyrolysen“) hat vermutlich einen starken Einfluss auf die Kohlenstoffausbeute im C/C Zustand und damit indirekt auf die CCR. Weniger ausschlaggebend sind hingegen vermutlich die Parameter Faserdichte, Probendicke und die maximale Temperatur während der Temperung.

Insgesamt lässt sich festhalten, dass von den 29 kategorischen und numerischen Herstellungsparametern, welche nach dem Preprocessing übriggeblieben waren, lediglich 19 tatsächlich aussagekräftig untersuchbar waren, da die restlichen zehn eine zu geringe Variabilität aufwiesen. Tabelle 22 gibt Aufschluss über die aussagekräftig untersuchbaren Parameter mit und ohne starken Einfluss auf die CCR, sowie die nicht aussagekräftig untersuchbaren Parameter.

Tabelle 22: Aussagekräftig untersuchbare Parameter, unterteilt nach Stärke ihres Einflusses auf die CCR (Feature Selection), sowie nicht aussagekräftig untersuchbare Parameter.

Untersuchbar, einflussreich	Untersuchbar, einflusslos	Nicht untersuchbar
Dichte Silizierung	Massenänderung Pyrolyse	Faser-Vorbehandlung
Porosität Silizierung	Dauer Temperung	Faser-Entschlichtung
Porosität Polymerisation	Soll-Dauer Polymerisation	Max. T. Temperung
Porosität Temperung	Anzahl Silizierungen	Anzahl Pyrolysen
Massenänderung Silizierung	FVG CFK	Soll-T. Polymerisation
Porosität Pyrolyse	Dichte Temperung	Soll-T. Silizierung
Fasermaterial	Dichte Pyrolyse	Probendicke
	Dichte Polymerisation	Faserart
	Precursor	Faserdichte
	Verfahren Polymerisation	Soll-T. Pyrolyse
	Faserorientierung	
	Geometrie	

6.2.5 Ausbleibende Einflüsse auf die CCR

Neben den gefundenen Korrelationen liefern auch ausbleibende Korrelationen wertvolle Hinweise für die Optimierung des Herstellungsprozesses. Aus diesem Grund wird an dieser Stelle noch auf die Herstellungsparameter eingegangen, die zwar aussagekräftig untersuchbar waren, jedoch keinen signifikanten Einfluss auf die CCR hatten (siehe Tabelle 22).

Eine überraschende Erkenntnis ist, dass die Massenänderung während der Pyrolyse nicht mit der CCR korreliert. Dies lässt den Schluss zu, dass das für die Einzelfasersilizierung nötige Risssystem bereits nach der Temperung vorliegt und nicht erst durch die Pyrolyse entsteht. Diese Vermutung wurde bei der Beobachtung der Pyrolyse in Kapitel 5.2.3 durch Mikroskop-Aufnahmen unterstützt. Zudem deckt sie sich auch mit dem Ergebnis der KI-Modelle, dass die offene Porosität im CFK-Zustand bereits eine Aussage über die CCR erlaubt. Da zur Porositätsmessung das Archimedes-Verfahren verwendet wurde, können Proben mit einem fein ausgeprägten Risssystem nach der Temperung mehr Wasser aufnehmen als jene, welche über kein ausgeprägtes Risssystem verfügen. Dadurch würde eine höhere Porosität gemessen werden.

Auch die Wahl der verwendeten CFK-Herstellungsmethode (Autoklav, RTM, Heißpresse, Wickelanlage) ist nicht von Bedeutung für die CCR, ebenso wenig wie die Temperaturen und Prozessdauer bei der Polymerisation und Temperung. Hierdurch kann geschlussfolgert werden, dass das ausschlaggebende Risssystem sich unabhängig von diesen genannten Parametern ausbildet.

Die ausbleibenden Effekte in der Variation von Prozesstemperaturen und -zeiten auf die Mikrostruktur legen den Schluss nahe, dass an diesen Stellen zukünftig Kosten optimiert werden können, indem diese Parameter möglichst niedrig gehalten werden. In zukünftigen Prozessierungen könnte zudem untersucht werden, ob der Prozess der Temperung überhaupt notwendig ist, oder ob nach der CFK-Herstellung direkt mit der Pyrolyse fortgefahren werden kann, ohne dass Qualitätseinbußen hinsichtlich der CCR entstehen.

Die Wahl des verwendeten Harzes hat zwar keinen direkten Einfluss auf die CCR, allerdings in manchen Fällen auf die erreichbare Porosität im polymerisierten und getemperten Zustand, wie Abbildung 92 zeigt.

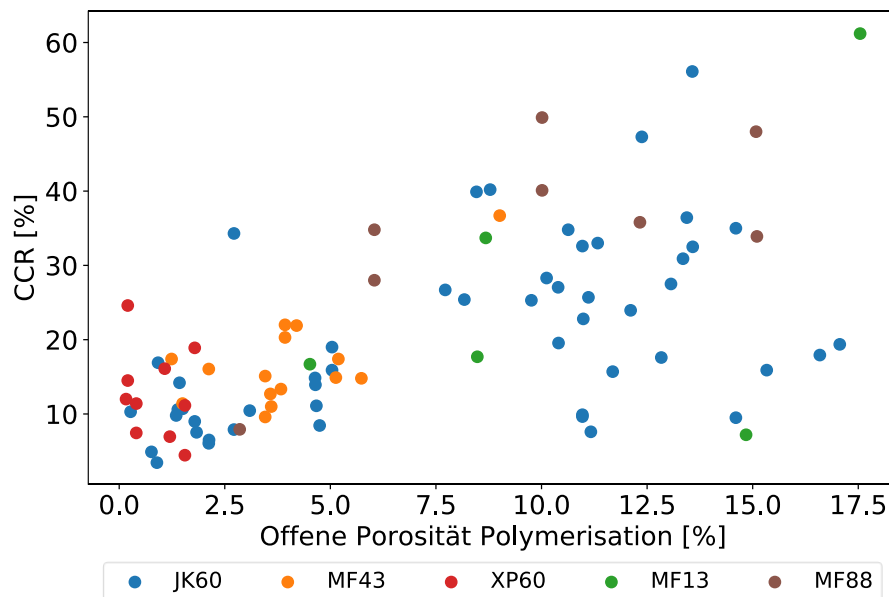


Abbildung 92: Einfluss verschiedener Harze auf die Porosität der Proben im polymerisierten

Hier wird ersichtlich, dass im Polymerzustand die Wahl von XP60 in einer Porosität von etwa 1% resultiert, wohingegen die Wahl von MF43 eine Porosität von etwa 4% ergibt. Sowohl JK60 als auch MF13 führen zu unvorhersagbaren Werten für die Porosität, die sich über die gesamte Spannbreite zwischen 1% und 17% erstrecken. Die Wahl des Harzsystems kann also in manchen Fällen ausschlaggebend für die Porosität sein, welche wiederum ausschlaggebend für die erreichbare CCR ist.

Eine weitere Erkenntnis dieser Arbeit liegt darin, dass die erreichbare CCR unabhängig von der Geometrie des Bauteils ist, wobei sich im Datensatz Proben aus Platten, Rohren, sowie Düsenbaugruppen wiederfinden. Auch der Faserwinkel hat keinen Einfluss auf die CCR. Beide Beobachtungen lassen sich so interpretieren, dass die Ausbildung des Rissystems unabhängig von den erwähnten Parametern ist. Folglich kann auch bei einer Hochskalierung von Proben auf größere Bauteile jede Form von Mikrostruktur erzielt werden.

Ein weiterer Parameter, der nicht mit der CCR korreliert, ist die Anzahl der Silizierungen. Dies kann damit begründet werden, dass während der ersten Silizierung noch eine ausreichende Kapillarwirkung vorliegt, welche flüssiges Silizium in das Bauteil zieht, wohingegen dieser Mechanismus bei nachfolgenden Silizierungen aufgrund der nun geschlossenen Riss- und Porenkanäle nur noch in stark abgeschwächter Form vorhanden ist. Dadurch treten bei der zweiten Silizierung kaum noch neue Reaktionen von Kohlenstoff und Silizium auf, wodurch sich auch keine weiteren Änderungen der Mikrostruktur ergeben. Demnach dienen Folgesilizierungen hauptsächlich dem Schließen offener Porosität und nicht der Gefügeveränderung.

7 Zusammenfassung und Ausblick

In diesem Kapitel werden die wesentlichen Ergebnisse der vorliegenden Arbeit zusammengefasst und ein Ausblick auf mögliche zukünftige Arbeiten gegeben. Dabei werden sowohl materialwissenschaftliche als auch methodische Erkenntnisse beleuchtet.

7.1 Zusammenfassung

Im Rahmen dieser Arbeit wurde der Zusammenhang zwischen Herstellungsparametern und resultierender Mikrostruktur von C/C-SiC Proben durch künstliche Intelligenz untersucht. Um diese Untersuchungen zu ermöglichen, wurde erstmals eine vollständige digitale Prozesskette zur Herstellung von C/C-SiC erstellt. Da zu Beginn keinerlei IT-Infrastruktur vorhanden war, wurden alle notwendigen Funktionalitäten im Zuge dieser Arbeit eigens entwickelt, wobei verschiedene Programmiersprachen zur Anwendung kamen. Ergänzend wurden Mikrostruktursimulationen und praktische Tests durchgeführt, sowie der Pyrolyseprozess unter dem Digitalmikroskop aufgezeichnet, um die Ergebnisse der KI-Modelle zu validieren.

Die digitale Prozesskette beinhaltete drei essenzielle Bausteine: eine PSQL-Datenbank, ein Web-Interface und das Auswertungstool DataTracker. Dabei diente das Web-Interface hauptsächlich der Eingabe, der Vorverarbeitung und dem Management von Daten, sowie der Übergabe an die Datenbank, während das Auswertungstool DataTracker für die KI-Auswertung und Visualisierung entwickelt wurde. Grundsätzlich lässt sich damit eine Probe von der Entstehung bis zur Fertigstellung auf übersichtliche Weise digital begleiten, was vor Beginn der Arbeit nur händisch über Excel-Tabellen möglich war. Um die Mikrostruktur als Zielgröße für die KI-Modelle zugänglich zu machen, wurde ein Verfahren zur Bewertung von REM-Aufnahmen entwickelt, sodass diese quantifiziert und objektiv miteinander verglichen werden konnten. Die so ermittelte Kennzahl, die Carbon Conversion Ratio (CCR), kann als Maß für die Ausprägung der Einzelfasersilizierung einer Probe verstanden werden. Eine besondere Herausforderung dieser Arbeit stellte die lückenhafte Dokumentation des zugrundeliegenden Datensatzes, sowie die teils geringe Variabilität der Herstellungsparameter dar. Trotzdem konnten einige methodische und materialwissenschaftliche Erkenntnisse gewonnen werden.

Zu den materialwissenschaftlichen Erkenntnissen zählte insbesondere die Bewertung der Wichtigkeit der 19 aussagekräftig untersuchbaren Herstellungsparameter. Hier kristallisierte sich bei Berücksichtigung aller Prozessschritte besonders die Dichte im silizierten Zustand als guter Indikator für die entstehende Mikrostruktur heraus, wobei höhere Werte für die Dichte mit höheren Werten für die CCR korrelierten. Weitere wichtige Indikatoren waren die Massenzunahme und die Porosität der Proben im silizierten Zustand, welche ebenfalls in einem

direkt proportionalen Zusammenhang mit der CCR standen. Aus Optimierungsgründen ist es jedoch besonders interessant, Einflussfaktoren zu einem möglichst frühen Zeitpunkt in der Prozesskette zu finden, weshalb ein weiteres Modell trainiert wurde, welches keine Daten aus dem Prozessschritt Silizierung erhielt. In diesem Fall erwies sich besonders die Porosität im CFK Zustand als guter Indikator für die CCR, was sowohl Messungen nach dem Prozessschritt Polymerisation als auch nach dem Prozessschritt Temperung betraf. Höhere Porositätswerte resultierten dabei in höheren Werten für die CCR und anders herum, wobei jedoch starke Unterschiede je nach gewählter Faser vorlagen. Derselbe Trend wurde außerdem für die Porosität im C/C-Zustand nach der Pyrolyse festgestellt. Die Wahl der verwendeten Faser beeinflusste ebenfalls die resultierende Mikrostruktur. So konnte gezeigt werden, dass die Verwendung von HTA-Fasern generell zu höheren Werten für die CCR führte, als die Verwendung von T800 Fasern, unabhängig vom verwendeten Harzsystem oder dem CFK-Herstellungsverfahren. Weiterhin war der lineare Trend zwischen Porosität und CCR bei HTA Fasern deutlich stärker ausgeprägt als bei T800 Fasern, bei welchen er kaum feststellbar war.

Kein signifikanter Zusammenhang bestand hingegen zwischen der CCR und dem verwendeten Harzsystem, dem Herstellungsverfahren, dem Faservolumengehalt, der Dauer von Polymerisation und Temperung, dem Massenverlust während der Pyrolyse, der Anzahl der Silizierungen, den Dichten im polymerisierten, getemperten und pyrolysierten Zustand, dem Faserwinkel und der Geometrie der Proben. Die Einflüsse von Faser-Vorbehandlung, Faser-Entschlichtung, Temperaturen während Polymerisation, Temperung, Pyrolyse und Silizierung, Anzahl der Pyrolysen, Probendicke, Faserart und Faserdichte konnten dahingegen nicht untersucht werden, da diese Parameter in zu geringem Maße variiert wurden, oder zu selten im Datensatz auftraten. Zu diesen kann in der vorliegenden Arbeit daher keine Optimierungsempfehlung gegeben werden. Da jedoch insbesondere die Hochtemperaturprozesse ein großes Kosteneinsparungspotenzial aufweisen, ist eine genauere Untersuchung hier zukünftig anzuraten.

Durch Anwendung der entwickelten Software konnten neben den materialwissenschaftlichen Erkenntnissen zudem einige methodische Erkenntnisse gesammelt werden. Da es im Bereich des Machine-Learning kein generelles Patentrezept gibt, das einheitlich auf alle Datensätze anwendbar ist, wurden in DataTracker für den überwiegenden Teil der Bearbeitungsschritte mehrere Möglichkeiten implementiert, welche einfach und schnell durch ein User-Interface änderbar sind. Die untersuchten Methoden umfassten neben der Wahl des optimalen Algorithmus auch die Festlegung eines geeigneten Ausfüllgrads, den Umgang mit Multikollinearität und Ausreißern, das Schätzen von fehlenden Werten, die Strategie bei der

Datenaufteilung in Test- und Trainingsdaten, die Bewertung der Wichtigkeiten bei der Feature-Selection und die Optimierung der Modell-Hyperparameter. Somit konnte durch diese Arbeit eine Plattform für die schnelle und effiziente statistische Auswertung von C/C-SiC Proben geschaffen werden, die von Benutzern ohne Programmiererfahrung verwendet werden kann. Als geeignetster Algorithmus für den vorliegenden Datensatz von 163 Proben kristallisierte sich RandomForest heraus, dessen Hyperparameter durch eine empirische Suchmatrix erhoben wurden. Die optimale Untergrenze für den Ausfüllgrad wurde in einer Parameterstudie als $A = 60\%$ identifiziert. Proben oder Herstellungsparameter, welche weniger als 60% ausgefüllt waren, wurden demnach verworfen. Das Schätzen der fehlenden Werte führte durch Verwendung eines iterativen multivariaten Schätzers zu den besten Ergebnissen. Unter den getesteten Möglichkeiten für die Bestimmung der wichtigsten Herstellungsparameter erwies sich die Modell-intrinsische Methode als am geeignetsten. Schließlich konnte gezeigt werden, dass eine Stratifizierung bei der Aufteilung in Test- und Trainingsdaten nur zu geringfügig besseren Ergebnissen hinsichtlich Genauigkeit und Standardabweichung führte, als die Verwendung von zufälligen Aufteilungen.

Weiterhin wurden im Rahmen dieser Arbeit FEM-Mikrostruktursimulationen durchgeführt, mit der Zielsetzung, die gefundenen Korrelationen der KI-Modelle durch die Simulation physikalischer Effekte zu erklären. In diesem Zuge konnten erstmals für die Pyrolyse relevante Größenordnungen bei realistischen Faserdurchmessern untersucht werden, wobei auch die Effekte unterschiedlicher Lagenwinkel auf die Rissentwicklung abgebildet wurden. Dazu wurden Python-Skripte geschrieben, welche die Lagenwinkel-Änderung innerhalb derselben Skale ermöglichten, da dies in der Standard-Software Simcenter Multimech 2022 nicht möglich war. Weiterhin wurde ein Python-Skript geschrieben, welches die durch die Simulation erstellten Mikrostrukturbilder einlesen und statistisch auswerten konnte, um die Anzahl und Größe der Risse zu quantifizieren. So konnte nachgewiesen werden, dass der bereits von den KI-Modellen gefundene Zusammenhang zwischen CCR und Porosität im CFK-Zustand auch physikalisch repliziert werden kann. Zusätzlich konnte die Korrelation zwischen verwendetem Fasermaterial und CCR, welcher schon durch die KI-Modelle detektiert worden war, erneut bestätigt werden. Um die für die Simulation nötigen mechanischen Kennwerte zu erhalten, wurden zudem begleitende mechanische Zug- und SENB-Tests an Harzproben durchgeführt. Um die Simulationen auch optisch anhand des beobachteten Rissmusters validieren zu können, wurde außerdem der Pyrolyseprozess von CFK-Proben unter einem Digitalmikroskop gefilmt und mit den Simulationsergebnissen verglichen.

Schlussendlich lässt sich festhalten, dass durch die in dieser Arbeit entwickelte digitale Prozesskette zum ersten Mal auf künstlicher Intelligenz basierende Auswertungen in verhältnismäßig großem Maßstab möglich werden. Hierdurch wurde das Potenzial geschaffen, zukünftige Prozesse zu optimieren, sowie die Qualität und Reproduzierbarkeit der Materialien zu verbessern. Neben der Beantwortung der wissenschaftlichen Forschungsfrage wurde ein Tool zur automatischen Datenauswertung bereitgestellt, welches auch in zukünftigen Untersuchungen angewendet werden kann.

7.2 Ausblick

Zukünftige Arbeiten könnten untersuchen, inwiefern sich die in dieser Arbeit nicht untersuchbaren Herstellparameter auf die erreichbare Mikrostruktur und die Energie- und Ressourceneffizienz der Herstellungskette auswirken. Hier sind besonders die langwierigen und teuren Hochtemperaturprozesse von Interesse, da diese das betriebswirtschaftlich größte Einsparungspotenzial bieten. Auch im Hinblick auf die Materialqualität könnten Parameter wie die maximale Pyrolysetemperatur und die Anzahl der Nachinfiltrationen und damit einhergehenden Pyrolysen einen großen Einfluss auf das entstehende Rissystem und somit auf die CCR haben. Neben den Hochtemperaturprozessen bietet aber auch die statistische Untersuchung von Temperatur- und Druckkurven während der Temperung eine Möglichkeit der Prozessoptimierung. In diesem Rahmen könnte außerdem untersucht werden, ob der Prozessschritt Temperung überhaupt zur Erreichung der gewünschten Mikrostruktur notwendig ist, oder ob möglicherweise sogar komplett darauf verzichtet werden kann, um Kosten und Energie zu sparen.

Diese Arbeit hat außerdem gezeigt, dass die Wahl des Fasermaterials einen großen Einfluss auf die erzielbare Mikrostruktur haben kann, wobei nur Proben mit T800- und HTA-Fasern in statistisch ausreichender Menge vorlagen. Folglich könnte in zukünftigen Arbeiten untersucht werden, ob ein Einfluss des Fasermaterials auch bei anderen Fasern feststellbar ist. Eine genauere Untersuchung der Ursache für dieses Verhalten wäre an dieser Stelle ebenfalls von Interesse.

Mit der Porosität im Polymerzustand konnte außerdem ein Parameter identifiziert werden, der bereits relativ früh in der Prozesskette einen Hinweis auf die zu erwartende Qualität der Mikrostruktur im silizierten Zustand ermöglicht. Auch hier sind die Ursachen für die Korrelation mit Hinblick auf die Optimierung der Materialqualität von Interesse und könnten in praktischen Folgearbeiten näher untersucht werden.

Schließlich bietet sich bezüglich der Mikrostruktursimulationen zukünftig eine Miteinbeziehung der Interface-Festigkeit an, welche in dieser Arbeit mangels realer Testwerte mit der Zugfestigkeit des Harzsystems gleichgesetzt wurde. Auch eine experimentelle Bestimmung des temperaturabhängigen E-Moduls des Harzes würde zu einer Verbesserung der Simulationsgenauigkeit führen.

8 Literatur

- [1] Schulte-Fischedick, J.: *Die Entstehung des Rissmusters während der Pyrolyse von CFK zur Herstellung von C/C-Werkstoffen* – PhD Thesis. Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen der Universität Stuttgart, 2006. <https://doi.org/10.1002/9783527622412>.
- [2] Walock, M.J.; Heng, V.; Nieto, A. et al.: *Ceramic Matrix Composite Materials for Engine Exhaust Systems on Next-Generation Vertical Lift Vehicles*. In: *Journal of Engineering for Gas Turbines and Power* 140 (2018), Heft 10. <https://doi.org/10.1115/1.4040011>.
- [3] Breede, F.: *Entwicklung neuartiger faserkeramischer C/C-SiC Verbundwerkstoffe auf Basis der Wickeltechnik für Raketendüsen* – PhD Thesis, Fakultät für Luft- und Raumfahrttechnik und Geodäsie, Universität Stuttgart. ELIB (Deutsches Zentrum für Luft- und Raumfahrt, 2017).
- [4] Krenkel, W.: *Ceramic Matrix Composites*. Wiley, 2008. ISBN 9783527313617. <https://doi.org/10.1002/9783527622412>.
- [5] Jain, N.: *Development of a digital manufacturing process chain for ceramic composites* – PhD Thesis. Universität Augsburg, Augsburg, 2022. <https://opus.bibliothek.uni-augsburg.de/opus4/94986>.
- [6] Corman, G.; Luthra K.: *Melt Infiltrated Ceramic Composites (Hipercomp) For Gas Turbine Engine Applications: Phase II Final Report* – DOE/CE/41000-2 (2006), https://digital.library.unt.edu/ark:/67531/metadc896795/m2/1/high_res_d/936318.pdf.
- [7] Deutsche Bundesregierung: *Klimaschutzgesetz - Generationenvertrag für das Klima*, 2022, <https://www.bundesregierung.de/breg-de/themen/klimaschutz/klimaschutzgesetz-2021-1913672#:~:text=Mit%20der%20C3%84nderung%20des%20Klimaschutzgesetzes,65%20Prozent%20gegen%20C3%BCber%201990%20sinken> [Zugriff am: 04.01.2023].
- [8] Sun, J.; Ye, D.; Zou, J. et al.: *A review on additive manufacturing of ceramic matrix composites*. In: *Journal of Materials Science & Technology* 138 (2023), S. 1-16. <https://doi.org/10.1016/j.jmst.2022.06.039>.
- [9] Choudhary, A.; Das Chakladar, N.; Paul, S.: *Identification and estimation of defects in high-speed ground C/SiC ceramic matrix composites*. In: *Composite Structures* 261 (2021), S. 113274. <https://doi.org/10.1016/j.compstruct.2020.113274>.
- [10] Raether, F.: *Ceramic Matrix Composites – an Alternative for Challenging Construction Tasks*. In: *Ceramic Applications* (2013). E-ISSN: 2196-2413, <https://www.htl.fraunhofer.de/content/dam/htl/de/%C3%9Cber%20uns/publikationen/Komposite/Ceramic%20Matrix%20Composites%20in%20Ceramic%20Applications%20Raether%20042013.pdf>.
- [11] Swiss-Composite.ch: *Produktbilder CFK Platten*, https://shop.swiss-composite.ch/shop/ProdukteBilder/ZZB100129_gr.jpg [Zugriff am: 22.03.2023].

- [12] DirectIndustry.de: *Produktbilder Faseroving*,
https://img.directindustry.de/images_di/photo-mg/193131-12102248.jpg [Zugriff am: 22.03.2023].
- [13] Yin, J.; Lee, S.-H.; Feng, L. et al.: *The effects of SiC precursors on the microstructures and mechanical properties of SiCf/SiC composites prepared via polymer impregnation and pyrolysis process*. In: *Ceramics International* 41 (2015), Heft 3, S. 4145-4153.
<https://doi.org/10.1016/j.ceramint.2014.11.112>.
- [14] Hofmann, S.: *Effect of interlaminar defects on the mechanical behaviour of carbon fibre reinforced silicon carbide* – PhD-Thesis. In: Universität Stuttgart (2014). ISSN: 1434-8454. <https://elib.dlr.de/87469/>, <https://elib.dlr.de/87469/>.
- [15] Singh, J.; Bansal, N.; Goto, T. et al.: *Processing and Properties of Advanced Ceramics and Composites IV* – Ceramic Transactions Volume 234. John Wiley & Sons, Inc, 2012. ISBN 978-1-118-27336-4.
- [16] Krenkel, W.: *Carbon Fiber Reinforced CMC for High-Performance Structures*. In: *International Journal of Applied Ceramic Technology* 1 (2004), Heft 2, S. 188-200.
<https://doi.org/10.1111/j.1744-7402.2004.tb00169.x>.
- [17] Heidenreich, B.: *Carbon Fibre Reinforced SiC Materials Based on Melt Infiltration*. In: *CORE* (2007), <https://elib.dlr.de/52517/>.
- [18] Arinez, J.F.; Chang, Q.; Gao, R.X. et al.: *Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook*. In: *Journal of Manufacturing Science and Engineering* 142 (2020), Heft 11. <https://doi.org/10.1115/1.4047855>.
- [19] Lee, J.; Davari, H.; Singh, J. et al.: *Industrial Artificial Intelligence for industry 4.0-based manufacturing systems*. In: *Manufacturing Letters* 18 (2018), S. 20-23.
<https://doi.org/10.1016/j.mfglet.2018.09.002>.
- [20] Deutsche Bundesregierung: *KI-Strategie Deutschland*, 2022, <https://www.ki-strategie-deutschland.de/home.html> [Zugriff am: 18.03.2023].
- [21] Li, B.; Hou, B.; Yu, W. et al.: *Applications of artificial intelligence in intelligent manufacturing: a review*. In: *Frontiers of Information Technology & Electronic Engineering* 18 (2017), Heft 1, S. 86-96. <https://doi.org/10.1631/FITEE.1601885>.
- [22] Aggour, K.S.; Gupta, V.K.; Ruscitto, D. et al.: *Artificial intelligence/machine learning in manufacturing and inspection: A GE perspective*. In: *MRS Bulletin* 44 (2019), Heft 7, S. 545-558. <https://doi.org/10.1557/mrs.2019.157>.
- [23] Ghayour, H.; Abdellahi, M.; Bahmanpour, M.: *Artificial intelligence and ceramic tools: Experimental study, modeling and optimizing*. In: *Ceramics International* 41 (2015), Heft 10, S. 13470-13479. <https://doi.org/10.1016/j.ceramint.2015.07.138>.
- [24] Xiang, G.A.; Guanghui, L.I.; Rong, T.A. et al.: *Using Deep Neural Networks to Predict the Tensile Property of Ceramic Matrix Composites Based on Incomplete Small Dataset*. In: *IOP Conference Series: Materials Science and Engineering* 647 (2019), Heft 1, S. 12004. <https://doi.org/10.1088/1757-899X/647/1/012004>.

- [25] Lehnert, T.; Heidenreich, B.; Koch, D.: *Investigation of process influences on the microstructural formation of C/C-SiC by machine-learning methods*. In: Open Ceramics (2022) [In Veröffentlichung].
- [26] Lexcellent, C.: *Artificial Intelligence*. In: Lexcellent, C. (Hrsg.): Artificial Intelligence versus Human Intelligence, SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, Cham, 2019, S. 5-21. https://doi.org/10.1007/978-3-030-21445-6_2.
- [27] Steinwendner, J.; Schwaiger, R.: *Neuronale Netze programmieren mit Python*, Rheinwerk computing, Rheinwerk Computing, Bonn, 2020. ISBN 978-3-8362-7450-0.
- [28] Jo, T.: *Machine Learning Foundations – Supervised, Unsupervised, and Advanced Learning*, Springer eBook Collection, Springer International Publishing; Imprint Springer, Cham, 2021. ISBN 978-3-030-65899-1. <https://doi.org/10.1007/978-3-030-65900-4>.
- [29] A.D.Dongare, R.R.Kharde, Amit D.Kachare: *Introduction to Artificial Neural Network*. In: International Journal of Engineering and Innovative Technology (IJEIT). ISSN: 2277-3754.
- [30] Bonaccorso, G.: *Machine learning algorithms – A reference guide to popular algorithms for data science and machine learning*. Packt, Birmingham, 2017. ISBN 978-1-78588-962-2.
- [31] Kiran, M.; Ozyildirim, M.: *Hyperparameter Tuning for Deep Reinforcement Learning Applications* Ausgabe 2022. <https://doi.org/10.48550/arXiv.2201.11182>.
- [32] Rodriguez-Galiano, V.; Sanchez-Castillo, M.; Chica-Olmo, M. et al.: *Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines*. In: Ore Geology Reviews 71 (2015), S. 804-818. <https://doi.org/10.1016/j.oregeorev.2015.01.001>.
- [33] Charbuty, B.; Abdulazeez, A.: *Classification Based on Decision Tree Algorithm for Machine Learning*. In: Journal of Applied Science and Technology Trends 2 (2021), Heft 01, S. 20-28. <https://doi.org/10.38094/jastt20165>.
- [34] Narayanan, R.; Honbo, D.; Memik, G. et al.: *An FPGA implementation of decision tree classification*. In: 2007 Design, Automation & Test in Europe Conference & Exhibition (pp. 1-6) (2007). <https://doi.org/10.1109/DATE.2007.364589>.
- [35] Boulesteix, A.-L.; Janitza, S.; Kruppa, J. et al.: *Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics*. In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2 (2012), Heft 6, S. 493-507. <https://doi.org/10.1002/widm.1072>.
- [36] Kotsiantis, S.; Tsekouras, G.; Pintelas, P.: *Bagging Model Trees for Classification Problems*. In: E.N. (eds) Advances in Informatics, S. 328-337. https://doi.org/10.1007/11573036_31.
- [37] Guo, L.; Chehata, N.; Mallet, C. et al.: *Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests*. In: ISPRS Journal of

- Photogrammetry and Remote Sensing 66 (2011), Heft 1, S. 56-66.
<https://doi.org/10.1016/j.isprsjprs.2010.08.007>.
- [38] Markus Schlüter: *Neuronale Netze* Ausgabe 2016.
<https://doi.org/10.13140/RG.2.2.16261.55523>.
- [39] *Proceedings of the 30th Chinese Control and Decision Conference (2018 CCDC)* – 09-11 June 2018, Shenyang, China. IEEE, Piscataway, NJ, 2018. ISBN 978-1-5386-1243-9.
- [40] B. Ding, H. Qian and J. Zhou: *Activation functions and their characteristics in deep neural networks*. In: Chinese Control And Decision Conference (CCDC), 2018, pp. 1836-1841, (2018). <https://doi.org/10.1109/CCDC.2018.8407425>.
- [41] Xu, J.; Li, Z.; Du, B. et al.: *Reluplex made more practical: Leaky ReLU*. In: : 2020 IEEE Symposium on Computers and Communications (ISCC). IEEE, Rennes, France, 2020, S. 1-7. <https://doi.org/10.1109/ISCC50000.2020.9219587>.
- [42] Ide, H.; Kurita, T.: *Improvement of learning for CNN with ReLU activation by sparse regularization*. In: : 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, Anchorage, AK, USA, 2017, S. 2684-2691.
<https://doi.org/10.1109/IJCNN.2017.7966185>.
- [43] *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects – COMITCon 2019* : 14th-16th February 2019. IEEE, Piscataway, NJ, 2019. ISBN 978-1-7281-0211-5.
- [44] Ranstam, J.; Cook, J.A.: *LASSO regression*. In: British Journal of Surgery 105 (2018), Heft 10, S. 1348. <https://doi.org/10.1002/bjs.10895>.
- [45] Grant, S.W.; Hickey, G.L.; Head, S.J.: *Statistical primer: multivariable regression considerations and pitfalls*. In: European journal of cardio-thoracic surgery : official journal of the European Association for Cardio-thoracic Surgery, Vol. 55 (2019), Iss. 2, pp. 179-185. <https://doi.org/10.1093/ejcts/ezy403>.
- [46] Tibshirani, R.: *Regression Shrinkage and Selection via the Lasso*. Journal of the Royal Statistical Society. Series B (Methodological), 1996. <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>.
- [47] Kotz, S. (ed.): *Encyclopedia of statistical sciences*. Wiley, Hoboken, N.J, 2005. ISBN 978-0-471-15044-2.
- [48] Hamada, M.; Tanimu, J.J.; Hassan, M. et al.: *Evaluation of Recursive Feature Elimination and LASSO Regularization-based optimized feature selection approaches for cervical cancer prediction*. In: : 2021 IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc). IEEE, Singapore, Singapore, 2021, S. 333-339. <https://doi.org/10.1109/MCSoc51149.2021.00056>.
- [49] Chicco, D.; Warrens, M.J.; Jurman, G.: *The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation*. In: PeerJ. Computer science, Vol. 7 (2021), e623.
<https://doi.org/10.7717/peerj-cs.623>.

- [50] Gonzalez Zelaya, C.V.: *Towards Explaining the Effects of Data Preprocessing on Machine Learning*. In: : 2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE, Macao, Macao, 2019, S. 2086-2090. <https://doi.org/10.1109/ICDE.2019.00245>.
- [51] Razali, N.M.; Yap, B.W.: *Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests*. In: Journal of Statistical Modeling and Analytics (2011). ISBN 978-967-363-157-5.
- [52] Pernot, P.; Huang, B.; Savin, A.: *Impact of non-normal error distributions on the benchmarking and ranking of quantum machine learning models*. In: Machine Learning: Science and Technology, Volume 1, Number 3 (2020). <https://doi.org/10.1088/2632-2153/aba184>.
- [53] D'Agostino, R.B. (ed.): *Goodness-of-fit techniques*, Statistics no. 68, Dekker, New York, NY, 1986. ISBN 0-8247-7487-6.
- [54] Hanusz, Z.; Tarasińska, J.: *Normalization of the Kolmogorov–Smirnov and Shapiro–Wilk tests of normality*. In: Biometrical Letters 52 (2015), Heft 2, S. 85-93. <https://doi.org/10.1515/bile-2015-0008>.
- [55] D'Agostino, R.B.: *An omnibus test of normality for moderate and large size samples*. In: Biometrika 58 (1971), Heft 2, S. 341-348. <https://doi.org/10.1093/biomet/58.2.341>.
- [56] Shapiro, S.S.; Wilk, M.B.: *An Analysis of Variance Test for Normality (Complete Samples)*. In: Biometrika 52 (1965), 3/4, S. 591. <https://doi.org/10.2307/2333709>.
- [57] Dodge, Y.: *The concise encyclopedia of statistics – With 247 tables*, Springer reference, Springer, New York, NY, 2008. ISBN 978-0-387-31742-7.
- [58] Alin, A.: *Multicollinearity*. In: Wiley Interdisciplinary Reviews: Computational Statistics 2 (2010), Heft 3, S. 370-374. <https://doi.org/10.1002/wics.84>.
- [59] Benesty, J.; Chen, J.; Huang, Y. et al.: *Pearson Correlation Coefficient*. In: Cohen, I.; Huang, Y.; Chen, J. et al. (Hrsg.): Noise Reduction in Speech Processing, Springer Topics in Signal Processing. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, S. 1-4. https://doi.org/10.1007/978-3-642-00296-0_5.
- [60] Lee Rodgers, J.; Nicewander, W.A.: *Thirteen Ways to Look at the Correlation Coefficient*. In: The American Statistician 42 (1988), Heft 1, S. 59-66. <https://doi.org/10.1080/00031305.1988.10475524>.
- [61] Sedgwick, P.: *Pearson's correlation coefficient*. In: BMJ 345 (2012), jul04 1, e4483-e4483. <https://doi.org/10.1136/bmj.e4483>.
- [62] Stine, R.A.: *Graphical Interpretation of Variance Inflation Factors*. In: The American Statistician 49 (1995), Heft 1, S. 53-56. <https://doi.org/10.1080/00031305.1995.10476113>.
- [63] Emmanuel, T.; Maupong, T.; Mpoeleng, D. et al.: *A survey on missing data in machine learning*. In: Journal of big data, Vol. 8 (2021), Iss. 1, p. 140. <https://doi.org/10.1186/s40537-021-00516-9>.

- [64] van Buuren, S.: *Flexible imputation of missing data*, Chapman & Hall/CRC interdisciplinary statistics series, CRC Press, Boca Raton, Fla., 2012. ISBN 1439868247.
- [65] Rubin, D.B.: *Inference and Missing Data*. In: *Biometrika* 63 (1976), Heft 3, S. 581. <https://doi.org/10.2307/2335739>.
- [66] Pedregosa, F.; Varoquaux, G.; Gramfort: *Scikit-learn: Machine Learning in Python – Imputation of Missing Values*, 2011, <https://scikit-learn.org/stable/modules/impute.html#iterative-imputer> [Zugriff am: 12.09.2022].
- [67] Hawkins, D.M.: *Identification of Outliers*, Springer eBook Collection Mathematics and Statistics, Springer, Dordrecht, 1980. ISBN 978-94-015-3994-4. <https://doi.org/10.1007/978-94-015-3994-4>.
- [68] Wada, K.: *Outliers in official statistics*. In: *Japanese Journal of Statistics and Data Science* 3 (2020), Heft 2, S. 669-691. <https://doi.org/10.1007/s42081-020-00091-y>.
- [69] Celi, L.A.; Charlton, P.; Ghassemi, M.: *Secondary Analysis of Electronic Health Records*. Springer, Cham (CH), 2016. ISBN 9783319437408. <https://doi.org/10.1007/978-3-319-43742-2>.
- [70] Peterson, R.: *Finding Optimal Normalizing Transformations via bestNormalize*. In: *The R Journal* Vol. 13/1 (2021). ISSN 2073-4859, <https://journal.r-project.org/archive/2021/RJ-2021-041/RJ-2021-041.pdf>.
- [71] Cabaneros, S.M.; Calautit, J.K.; Hughes, B.R.: *A review of artificial neural network models for ambient air pollution prediction*. In: *Environmental Modelling & Software* 119 (2019), S. 285-304. <https://doi.org/10.1016/j.envsoft.2019.06.014>.
- [72] Sakia, R.M.: *The Box-Cox Transformation Technique: A Review*. In: *The Statistician* 41 (1992), Heft 2, S. 169. <https://doi.org/10.2307/2348250>.
- [73] Box, G.E.P.; Cox, D.R.: *An Analysis of Transformations*. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 26, no. 2 (1964): 211–52. (1964), <http://www.jstor.org/stable/2984418>.
- [74] Potdar, K.; S., T.; D., C.: *A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers*. In: *International Journal of Computer Applications* 175 (2017), Heft 4, S. 7-9. <https://doi.org/10.5120/ijca2017915495>.
- [75] Polasek, W. (Hrsg.): *Explorative Daten-Analyse*, Heidelberger Lehrtexte Wirtschaftswissenschaften, Springer Berlin Heidelberg, Berlin, Heidelberg, 1988. ISBN 978-3-540-19417-0.
- [76] Batta Mahesh: *Machine Learning Algorithms - A Review*. In: *International Journal of Science and Research (IJSR)* (2019). <https://doi.org/10.21275/ART20203995>.
- [77] Berrar, D.: *Cross-Validation*. In: *Dalia*, A. (Hrsg.): *Encyclopedia of Bioinformatics and Computational Biology*. Elsevier, 2019, S. 542-545. <https://doi.org/10.1016/B978-0-12-809633-8.20349-X>.
- [78] Chen, L. (Hrsg.): *Deep Learning and Practice with MindSpore*, Cognitive Intelligence and Robotics, Springer Singapore, Singapore, 2021. ISBN 978-981-16-2232-8.

- [79] Chaoji, V.; Rastogi, R.; Roy, G.: *Machine learning in the real world*. In: Proceedings of the VLDB Endowment 9 (2016), Heft 13, S. 1597-1600.
<https://doi.org/10.14778/3007263.3007318>.
- [80] King, R.D.; Orhobor, O.I.; Taylor, C.C.: *Cross-validation is safe to use*. In: Nature Machine Intelligence 3 (2021), Heft 4, S. 276. <https://doi.org/10.1038/s42256-021-00332-z>.
- [81] Blockeel, H.: *Efficient algorithms for decision tree cross-validation*. In: Journal of Machine Learning Research 3.Dec (2002): 621-650 2002.
<https://doi.org/10.48550/arXiv.cs/0110036>.
- [82] Pedregosa, F.; Varoquaux, G.; Gramfort, A. et al.: *Scikit-learn: Machine Learning in Python*, 2011, https://scikit-learn.org/stable/modules/cross_validation.html [Zugriff am: 30.08.2022].
- [83] Breiman, L.: *Bagging predictors*. In: Machine Learning 24 (1996), Heft 2, S. 123-140.
<https://doi.org/10.1023/A:1018054314350>.
- [84] Jovic, A.; Brkic, K.; Bogunovic, N.: *A review of feature selection methods with applications*. In: : 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, Opatija, Croatia, 2015, S. 1200-1205. <https://doi.org/10.1109/MIPRO.2015.7160458>.
- [85] Gilles Louppe: *Understanding Random Forests: From Theory to Practice* 2014.
<https://doi.org/10.13140/2.1.1570.5928>.
- [86] Shafiee, S.; Lied, L.M.; Burud, I. et al.: *Sequential forward selection and support vector regression in comparison to LASSO regression for spring wheat yield prediction based on UAV imagery*. In: Computers and Electronics in Agriculture 183 (2021), S. 106036.
<https://doi.org/10.1016/j.compag.2021.106036>.
- [87] Sebastian Rashka; Vahid Mirjalili: *Python Machine Learning – Machine Learning and Deep Learning with Python scikit learn and TensorFlow*. Packt Publishing Ltd. Birmingham, 2017. ISBN 9781787125933.
- [88] Radivojac, P.; Obradovic, Z.; Dunker, A.K. et al.: *Feature Selection Filters Based on the Permutation Test*. In: Hutchison, D.; Kanade, T.; Kittler, J. et al. (Hrsg.): Machine Learning: ECML 2004, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, S. 334-346. https://doi.org/10.1007/978-3-540-30115-8_32.
- [89] Belytschko, T.; Gracie, R.; Ventura, G.: *A review of extended/generalized finite element methods for material modeling*. In: Modelling and Simulation in Materials Science and Engineering 17 (2009), Heft 4, S. 43001. <https://doi.org/10.1088/0965-0393/17/4/043001>.
- [90] Weisser, D.: *A wind energy analysis of Grenada: an estimation using the 'Weibull' density function*. In: Renewable Energy 28 (2003), Heft 11, S. 1803-1812.
[https://doi.org/10.1016/s0960-1481\(03\)00016-8](https://doi.org/10.1016/s0960-1481(03)00016-8).
- [91] Wittel, F.K.; Schulte-Fischedick, J.; Kun, F. et al.: *Discrete element simulation of transverse cracking during the pyrolysis of carbon fibre reinforced plastics to*

- carbon/carbon composites*. In: Computational Materials Science 28 (2003), Heft 1, S. 1-15. [https://doi.org/10.1016/S0927-0256\(03\)00035-1](https://doi.org/10.1016/S0927-0256(03)00035-1).
- [92] Kosin, M.: *Development of Virtual Micro-Models for Prediction of Macroscopic Properties of Ceramic Matrix Composites (CMCs) and Optimization of Manufacturing Process* – Master Thesis (2018), https://www.researchgate.net/publication/330497334_Development_of_Virtual_Micro-Models_for_Prediction_of_Macroscopic_Properties_of_Ceramic_Matrix_Composites_CMCs_and_Optimization_of_Manufacturing_Process.
- [93] Fan, J. (ed.): *Advances in heterogeneous material mechanics* – Proceedings of the Third International Conference on Heterogeneous Material Mechanics, May 22 - 26, 2011, Shanghai, China. DEStech Publ, Lancaster, 2011. ISBN 9781605950549.
- [94] Swiss-Composite: *Carbon-(Kohlenstoff-Fasern) Fasertabelle* – TORAYCA-Fasern, TENAX-Fasern, MITSUBISHI-Fasern, <https://www.swiss-composite.ch/pdf/i-Carbon-Fasertabelle.pdf> [Zugriff am: 15.03.2023].
- [95] Toray Composite Materials America, Inc.: *T800S Intermediate Modulus Fiber*, 2018, <https://www.toraycma.com/wp-content/uploads/T800S-Technical-Data-Sheet-1.pdf.pdf> [Zugriff am: 15.03.2023].
- [96] Rouway, M.; Nachtane, M.; Tarfaoui, M. et al.: *Prediction of Mechanical Performance of Natural Fibers Polypropylene Composites: a Comparison Study*. In: IOP Conference Series: Materials Science and Engineering 948 (2020), Heft 1, S. 12031. <https://doi.org/10.1088/1757-899X/948/1/012031>.
- [97] E08 Committee: *Test Method for Linear-Elastic Plane-Strain Fracture Toughness of Metallic Materials*. <https://doi.org/10.1520/E0399-22>.
- [98] Bakker, A.: *Elastic-plastic fracture mechanics analysis of an SENB specimen*. In: International Journal of Pressure Vessels and Piping 10 (1982), Heft 6, S. 431-449. [https://doi.org/10.1016/0308-0161\(82\)90004-7](https://doi.org/10.1016/0308-0161(82)90004-7).
- [99] Hahn, G. T., M. F. Kanninen, Rosenfield, A. R.: *Fracture toughness of materials*. In: Annual Review of Materials Science 2.1 (1972), <https://www.annualreviews.org/doi/pdf/10.1146/annurev.ms.02.080172.002121>.
- [100] Ma, J.; Wu, W.; Ke, Z. et al.: *Viscosity master curves and predictions of phenolic resin solutions through early aging*. In: Polymer 261 (2022), S. 125405. <https://doi.org/10.1016/j.polymer.2022.125405>.
- [101] Mijatovic, J.; Binder, W.H.; Kubel, F. et al.: *Studies on the stability of MF resin solutions: investigations on network formation*. In: Macromolecular Symposia 181 (2002), Heft 1, S. 373-382. [https://doi.org/10.1002/1521-3900\(200205\)181:1<373::AID-MASY373>3.0.CO;2-J](https://doi.org/10.1002/1521-3900(200205)181:1<373::AID-MASY373>3.0.CO;2-J).
- [102] Schulz, M.: *Einfluss der Faser-Matrix-Anbindung auf die Ausbildung der Mikrorissstruktur bei der Herstellung von keramischen Faserverbundwerkstoffen im Flüssigsilizierverfahren* – PhD (2021), <https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/index/index/docId/89791>.

- [103] Brandt, R.; Frieß, M.; Neuer, G.: *Thermal conductivity, specific heat capacity, and emissivity of ceramic matrix composites at high temperatures*. In: High Temperatures-High Pressures 35/36 (2003), Heft 2, S. 169-177. <https://doi.org/10.1068/htjr105>.
- [104] Smith, L.N.: *Cyclical Learning Rates for Training Neural Networks*. In: : 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, Santa Rosa, CA, USA, 2017, S. 464-472. <https://doi.org/10.1109/WACV.2017.58>.

Anhang

A) Beispielhafte Anwendung des Gradientenabstiegsverfahrens auf ein einfaches Perzeptron

In diesem Beispiel wird das Perzeptron in seiner Grundform verwendet, welches ein sehr simples neuronales Netzwerk mit nur einem einzigen Neuron und einem einzelnen Gewicht darstellt, welches in Abbildung 93 skizziert ist.

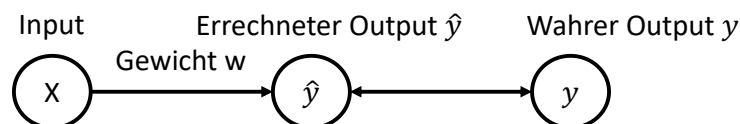


Abbildung 93: Schematische Darstellung eines Perzeptrons in seiner grundlegendsten Form [27].

Zur Berechnung des Fehlers wird folgende quadratische Kostenfunktion verwendet:

$$E(w) = (y - \hat{y})^2 \quad (34)$$

Dabei beschreibt \hat{y} den Output des Perzeptrons und y den wahren (gewünschten) Wert der Zielgröße. Zur Vereinfachung wurde auf eine Aktivierungsfunktion verzichtet. Es werden außerdem folgende Annahmen aufgestellt:

- Der Input des Perzeptrons beträgt $X = 0,2$
- Der Output des Perzeptrons soll $\hat{y} = 1$ betragen (da auch der wahre Wert $y = 1$ beträgt)

Nun soll durch Gradientenabstieg das Gewicht w gefunden werden, welches den Fehler minimiert. In Abbildung 94 ist der Fehler des Perzeptrons E über dem Gewicht w aufgetragen. Da Kostenfunktionen oft als der quadratische Fehler zwischen wahren Wert y und vorhergesagtem Wert \hat{y} gewählt werden, ist diese Form realistisch. Der vorhergesagte Wert wird dabei durch Multiplikation von Input- und Gewichtswert berechnet:

$$\hat{y} = X \cdot w \quad (35)$$

Die Änderung des Gewichts wird dann anhand der partiellen Ableitung der Kostenfunktion nach dem Gewicht berechnet und ergibt sich zu:

$$\Delta w = -1 \cdot \eta \frac{\partial}{\partial w} E(w) = \frac{\partial}{\partial w} (\eta \cdot (y - X \cdot w)^2) = -2X \cdot \eta \cdot (y - \hat{y}) \quad (36)$$

Wobei die Lernrate η klein gewählt werden sollte. Diese bestimmt über die Schrittlänge beim Gradientenabstieg und kann bei falscher Wahl die Konvergenz erschweren oder sogar verhindern [104]. Für dieses Beispiel wird $\eta = 1$ gesetzt.

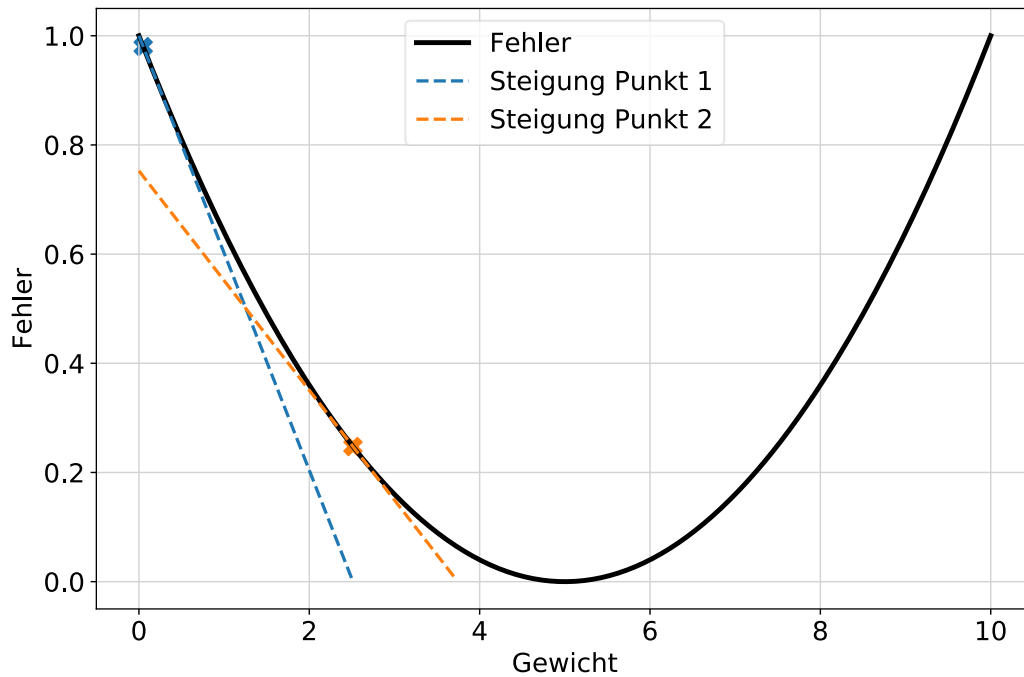


Abbildung 94: Quadratische Kostenfunktion (schwarz) mit Steigungen (gestrichelte blaue und orangefarbene Linien) an unterschiedlichen Punkten (blaue, orangefarbene Kreuze) [27].

Es ist erkennbar, dass die Kostenfunktion ihr Minimum beim Gewichtswert $w = 5$ erreicht. Um diesen Wert durch das Gradientenabstiegsverfahren zu berechnen, sind folgende Schritte vorzunehmen [27]:

1. Bei einem beliebigen Gewichtswert starten (beispielsweise $w = -4$)
2. Gradienten an diesem Punkt ermitteln
3. Einen Schritt in die entgegengesetzte Richtung des steilsten Anstiegs machen (Schrittlänge durch Lernrate bestimmen) und Gewichte aktualisieren
4. Punkt 2 und 3 so lange wiederholen, bis der Fehler kleiner als eine selbst festgelegte Schwelle ist

Das Vorgehen für das obige Beispiel mit Lernrate $\eta = 1$ ist in Tabelle 23 festgehalten:

Tabelle 23: Beispielhafte iterative Berechnung des Gewichts w mit Aktualisierung des Fehlers $E(w)$.

Iteration	\mathbf{X}	\mathbf{w}	$\hat{\mathbf{y}}$	\mathbf{y}	$\mathbf{E(w)}$	$\Delta\mathbf{w}$
0	0,2	-4	-0,8	1	3,24	0,72
1	0,2	-3,28	-0,656	1	2,74	0,6624
...	0,2	1
n	0,2	4,999	0,999	1	1,2E-4	0,000024

Nach n Iterationen liegt der Fehler bei $E(w) = 1,2E - 4$ und wird als vernachlässigbar angesehen. Der Ausgabewert liegt mit $\hat{y} = 0,999$ sehr nahe am wahren Wert $y = 1$, das dazu

notwendige Gewicht von $w \approx 5$ ist nun bestimmt. Abbildung 95 zeigt den Fehler $E(w)$ sowie die Gewichtsanzpassung Δw bei jeder Iteration. Nach 125 Iterationen ist der Fehler vernachlässigbar gering, allerdings wird auch ersichtlich, dass die Gewichtsanzpassung immer langsamer geschieht, je kleiner der Fehler wird. Dies kann durch verschiedene Techniken verbessert werden, auf die an dieser Stelle nicht genauer eingegangen wird.

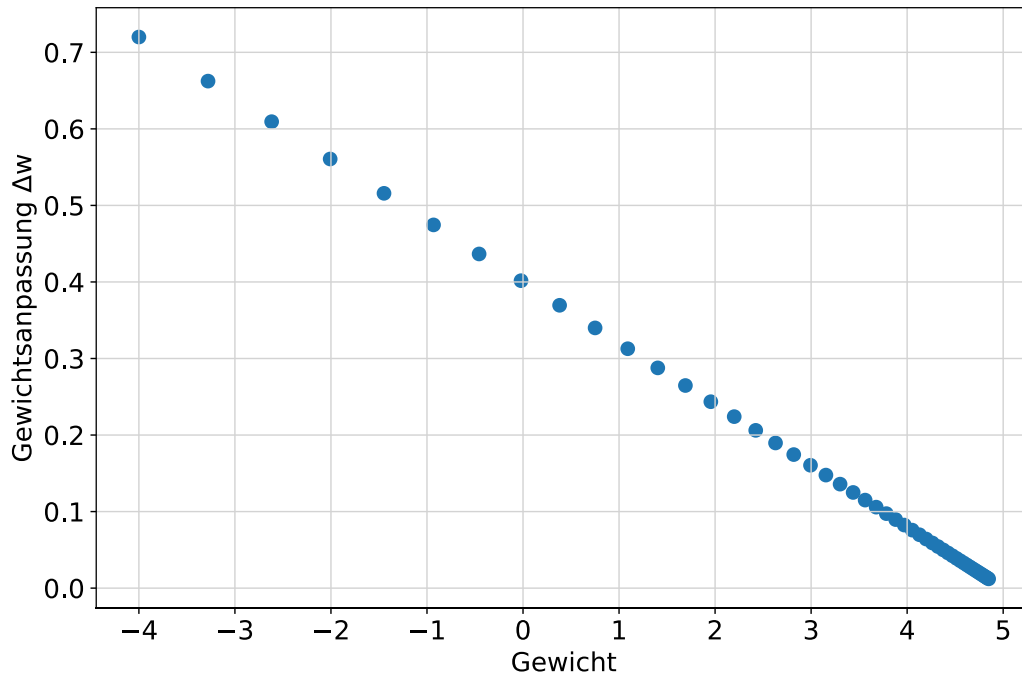


Abbildung 95: Gewichtsanzpassung Δw über dem gewählten Gewicht ohne Aktivierungsfunktion.

In realen neuronalen Netzwerken werden aber, wie bereits beschrieben, Aktivierungsfunktionen verwendet, was die Gradientenberechnung beeinflusst. Mit der Sigmoidfunktion als Aktivierungsfunktion würde sich für den Output des Neurons ergeben [27]:

$$\hat{y} = \text{sigmoide}(X \cdot w) = \frac{1}{1 + e^{-(X \cdot w)}} \quad (37)$$

Leitet man den Gesamtfehler $E(w) = (y - \hat{y})^2$ nach dem Gewicht ab, um Δw zu berechnen, ergibt sich:

$$\begin{aligned} \frac{\partial}{\partial w} E(w) &= 2(y - \hat{y}) \cdot \frac{\partial}{\partial w} \text{sigmoide}(X \cdot w) \\ &= 2(y - \hat{y}) \cdot \left[-1 \cdot \frac{1}{1 + e^{-X \cdot w}} \cdot \left(1 - \frac{1}{1 + e^{-X \cdot w}} \right) \right] \end{aligned} \quad (38)$$

$$\Delta w = -1 \cdot \eta \frac{\partial}{\partial w} E(w) = 2\eta(y - \hat{y}) \cdot \frac{1}{1 + e^{-X \cdot w}} \cdot \left(1 - \frac{1}{1 + e^{-X \cdot w}} \right) \quad (39)$$

Trägt man im selben Beispiel den Gesamtfehler $E(w)$ und die Gewichtsänderung Δw wieder über dem Gewichtswert auf, erkennt man, dass sich ein anderer Wert für das optimale Gewicht ergibt (siehe Abbildung 96).

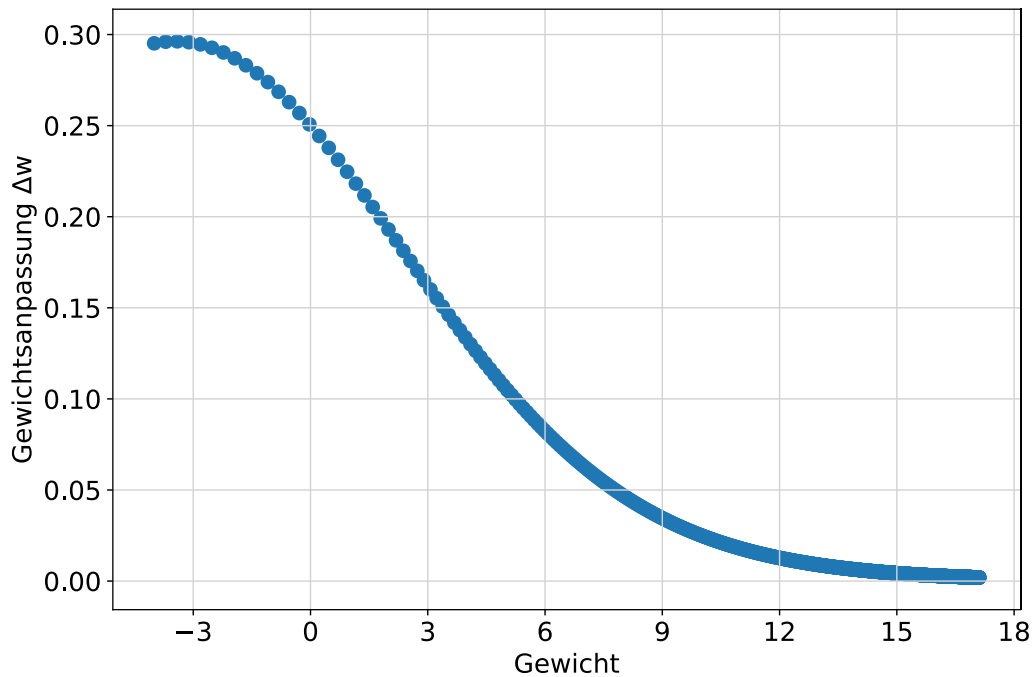


Abbildung 96: Gewichtsanzpassung Δw über dem gewählten Gewicht unter Verwendung der Sigmoidfunktion als Aktivierungsfunktion.

Bereits hier ist das langsame Konvergieren der Sigmoidfunktion aufgrund des verschwindenden Gradienten bei niedrigen Fehlerwerten ersichtlich. Weiterhin wird deutlich, dass die Gewichte bei unterschiedlicher Aktivierungsfunktion voneinander abweichen, woraus resultiert, dass man durch Auswechseln der Aktivierungsfunktion auch den Trainingsprozess eines Neuronalen Netzwerks wiederholen muss.

B) Durchführung eines Vorwärts- und Rückwärtsdurchgangs durch ein mehrschichtiges Neuronales Netzwerk

In diesem Beispiel wird ein vollständiger Vorwärts- sowie ein Rückwärtsdurchgang durch ein neuronales Netzwerk erklärt, wobei die verwendete Netzwerk-Architektur möglichst simpel gehalten wurde, wie in Abbildung 97 dargestellt.

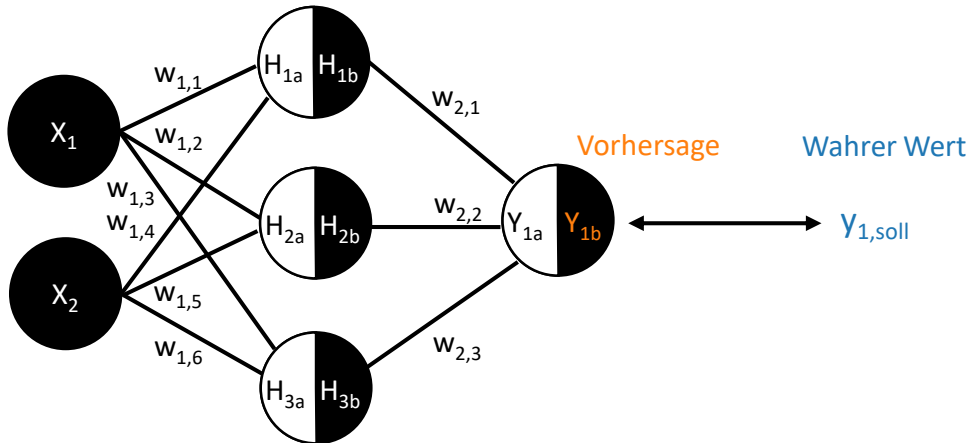


Abbildung 97: Schematischer Aufbau des angenommenen Neuronales Netzkes für die Beispielberechnung der Backpropagation.

Vorwärtsdurchgang

Mithilfe der initiierten Gewichte w_{ij} und Inputs X_i können die Werte für die Neuronen der versteckten Schicht H_{ia} berechnet werden. Über die Aktivierungsfunktion kann daraus der Output H_{ib} der Neuronen in der versteckten Schicht ermittelt werden, wobei in diesem Beispiel die ReLu Funktion verwendet wird. Dasselbe Vorgehen wird angewendet, um aus H_{ib} und w_{ij} über Y_{ia} den Output des Neuronales Netzkes Y_{ib} zu berechnen. Dieser wird anschließend mit den korrekten Lösungen der Trainingsdaten $y_{i,soll}$ verglichen, um den Fehler $E(w)$ zu ermitteln, welcher anschließend für die Anpassung aller Gewichte während der Backpropagation herangezogen wird. Dabei gelten die folgenden Ausgangsgleichungen:

$$H_{1a} = X_1 \cdot w_{1,1} + X_2 \cdot w_{1,4} \quad (40)$$

$$H_{2a} = X_1 \cdot w_{1,2} + X_2 \cdot w_{1,5} \quad (41)$$

$$H_{3a} = X_1 \cdot w_{1,3} + X_2 \cdot w_{1,6} \quad (42)$$

$$H_{1b} = \max(0, H_{1a}) \quad (43)$$

$$H_{2b} = \max(0, H_{2a}) \quad (44)$$

$$H_{3b} = \max(0, H_{3a}) \quad (45)$$

$$Y_{1a} = H_{1b} \cdot w_{2,1} + H_{2b} \cdot w_{2,2} + H_{3b} \cdot w_{2,3} \quad (46)$$

$$Y_{1b} = \max(0, Y_{1a}) \quad (47)$$

$$E(w) = \frac{1}{2} (y_{1,soll} - Y_{1b})^2 \quad (48)$$

Rückwärtsdurchgang (Backpropagation)

Nun kann über die partiellen Ableitungen berechnet werden, wie die Gewichte w_{ij} geändert werden müssen, um den Fehler $E(w)$ zu minimieren. Im Folgenden wird nun eine einzelne Iteration für ein einzelnes Gewicht händisch durchgeführt, um den Vorgang zu beschreiben. Alle weiteren Iterationen und Gewichte werden analog berechnet.

Es wird nun das aktualisierte Gewicht $w_{1,1,neu}$ berechnet, wofür nach Formel (49) vorgegangen wird [27]:

$$w_{1,1,neu} = w_{1,1,alt} + \Delta w_{1,1} \quad (49)$$

$$\text{mit } \Delta w_{1,1} = -\eta \cdot \frac{\partial E(w)}{\partial w_{1,1}}$$

Die partielle Ableitung $\frac{\partial E(w)}{\partial w_{1,1}}$ beschreibt dabei, wie sehr sich der Fehler ändert, wenn sich das Gewicht $w_{1,1}$ um einen gewissen Betrag ändert. Diese ist aber nicht unmittelbar berechenbar, sondern muss folgendermaßen aufgeteilt werden:

$$\frac{\partial E(w)}{\partial w_{1,1}} = \frac{\partial E(w)}{\partial Y_{1b}} \cdot \frac{\partial Y_{1b}}{\partial Y_{1a}} \cdot \frac{\partial Y_{1a}}{\partial H_{1b}} \cdot \frac{\partial H_{1b}}{\partial H_{1a}} \cdot \frac{\partial H_{1a}}{\partial w_{1,1}} \quad (50)$$

Die in Gleichung (50) beschriebenen partiellen Ableitungen können nun gebildet werden. Dabei wurde eine ReLu-Aktivierungsfunktion angenommen.

$$\frac{\partial E(w)_1}{\partial Y_{1b}} = -(y_{soll} - Y_{1b}) \quad (51)$$

$$\frac{\partial Y_{1b}}{\partial Y_{1a}} = \begin{cases} 1 & \text{falls } Y_{1a} > 0 \\ 0 & \text{falls } Y_{1a} \leq 0 \end{cases} \quad (52)$$

$$\frac{\partial Y_{1a}}{\partial H_{1b}} = w_{2,1} \quad (53)$$

$$\frac{\partial H_{1b}}{\partial H_{1a}} = \begin{cases} 1 & \text{falls } H_{1a} > 0 \\ 0 & \text{falls } H_{1a} \leq 0 \end{cases} \quad (54)$$

$$\frac{\partial H_{1a}}{\partial w_{1,1}} = X_1 \quad (55)$$

Mit der so ermittelten Gewichtsänderung $\Delta w_{1,1}$ kann das Gewicht $w_{1,1}$ aktualisiert werden. Dieser Vorgang muss innerhalb einer Epoche entsprechend für alle Gewichte des Netzwerks wiederholt werden. Anschließend wird mit der nächsten Epoche fortgefahren, bis die Maximalanzahl an Epochen erreicht wird, oder der Fehler unter ein bestimmtes Mindestmaß fällt. Dies soll nun durch das Einsetzen von Zahlenwerten anhand eines konkreten Beispiels verdeutlicht werden. Dazu soll das oben skizzierte Neuronale Netzwerk darauf trainiert werden, anhand zweier Messungen, nämlich der Dichte und der Porosität einer Probe, die Carbon Conversion Ratio (CCR) vorherzusagen. Die Werte für die Gewichte werden zunächst zufällig initiiert. Dadurch ergibt sich folgende Ausgangssituation:

- Dichte: $1,7 \text{ kg/m}^3$ (in X_1 berücksichtigt)
- Porosität: 15% (in X_2 berücksichtigt)
- $w_{1,1} = 0,8$; $w_{1,2} = 0,5$; $w_{1,3} = 0,2$; $w_{1,4} = 0,1$; $w_{1,5} = 0,4$; $w_{1,6} = 0,7$
 $w_{2,1} = 0,9$; $w_{2,2} = 0,3$; $w_{2,3} = 0,5$
- Soll-Ausgabe Wert für CCR: $y_{soll} = 43$

Damit lässt sich der Vorwärtsthrough durch das Neuronale Netzwerk berechnen. Zunächst werden die Werte der versteckten Schicht berechnet:

$$H_{1a} = 1,7 \cdot 0,8 + 15 \cdot 0,1 = 2,86$$

$$H_{2a} = 1,7 \cdot 0,5 + 15 \cdot 0,4 = 6,85$$

$$H_{3a} = 1,7 \cdot 0,2 + 15 \cdot 0,7 = 10,80$$

Da alle Werte größer als 0 sind, gibt die ReLu-Aktivierungsfunktion den ankommenden Wert einfach weiter, woraus folgt:

$$H_{1b} = H_{1a} = 2,86$$

$$H_{2b} = H_{2a} = 6,85$$

$$H_{3b} = H_{3a} = 10,80$$

Dadurch lassen sich nun Y_{1a} und Y_{1b} berechnen:

$$Y_{1a} = 2,86 \cdot 0,9 + 6,85 \cdot 0,3 + 10,80 \cdot 0,5 = 10,03 = Y_{1b}$$

Das Neuronale Netzwerk sagt demnach mit den zufällig initiierten Gewichten einen Wert von $CCR = 10,03$ voraus, obwohl der gewünschte Wert 43 beträgt. Anhand des Ist- und des Sollwerts kann nun der Fehler berechnet werden, anhand dessen das Netzwerk trainiert werden kann. Dies geschieht während dem Rückwärtsthrough (Backpropagation). Werden die Zahlenwerte in die Formeln (50)-(55) eingesetzt, können somit die Änderungen für die jeweiligen Gewichte berechnet werden. Dies wird beispielhaft für die Änderung des Gewichts $w_{1,1}$ gezeigt:

$$\frac{\partial E(w)}{\partial w_{1,1}} = -(43 - 10,03) \cdot 1 \cdot 0,9 \cdot 1 \cdot 1,7$$

$$\frac{\partial E(w)}{\partial w_{1,1}} = -50,49$$

Setzt man eine Lernrate von $\eta = 0,0001$ voraus, berechnet sich damit die Gewichtsanzpassung $\Delta w_{1,1}$ des ersten Gewichts zu:

$$\Delta w_{1,1} = -\eta \cdot \frac{\partial E(w)}{\partial w_{1,1}} \approx 0,005$$

Und das neue Gewicht hat damit einen Wert von $w_{1,1} = 0,805$. Dasselbe Vorgehen muss nun für alle Gewichte und Iterationen durchgeführt werden. Programmiert man das beschriebene Neuronale Netz in Python und zeichnet die Abweichung zwischen vorhergesagtem und wahren Wert für die CCR über der Anzahl der Epochen auf, erkennt man, dass der Fehler innerhalb von ca. 120 Epochen vernachlässigbar klein wird (siehe Abbildung 98).

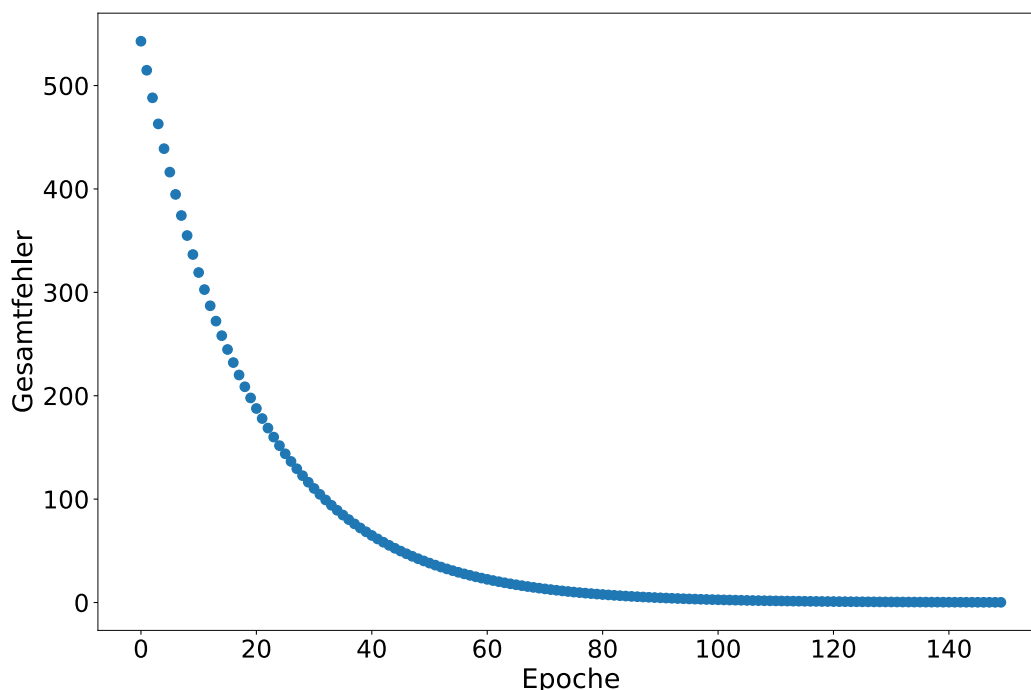


Abbildung 98: Fehler über Epoche beim Training des Neuronalen Netzwerks. Der relativ große initiale Fehler von $E(w) = 544$ wird stetig kleiner und tendiert schließlich gegen 0

In der Praxis wird ein Neuronales Netzwerk jedoch nicht nur anhand eines einzelnen Trainingsbeispiels trainiert. Stattdessen muss der Fehler über alle Trainingsbeispiele minimiert werden. Durch die hohe Anzahl an Rechenschritten ist schnell ersichtlich, warum große Neuronale Netzwerke leistungsstarke Computerhardware voraussetzen.