

Institut für Parallele und Verteilte Systeme  
Abteilung Anwendersoftware

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Masterarbeit

**Entwicklung einer Methode zur  
datengetriebenen Vorhersage der  
Ausgaben einer  
Kaltmassivumformsimulation**

Florian Niebling

<b>Studiengang:</b>	Informatik
<b>Prüfer/in:</b>	Prof. Dr.-Ing. habil. Bernhard Mitschang
<b>Betreuer/in:</b>	Dr. rer. nat. Peter Reimann, Julius Voggesberger, M.Sc.
<b>Beginn am:</b>	7. März 2023
<b>Beendet am:</b>	21. September 2023



## Kurzfassung

Im Bereich der Kaltmassivumformung werden physikalische Simulationen eingesetzt, um den Fertigungsprozess zu optimieren. Sie sind kostengünstiger als physische Experimente und ermöglichen es, die Eigenschaften des Produkts der Umformung anhand von Prozessparametern präzise zu bestimmen. Eine lange Laufzeit von bis zu mehreren Tagen schränkt den Nutzen von Simulationen für die Prozessoptimierung jedoch ein. An ihrer Stelle können datengetriebene Surrogatmodelle eingesetzt werden. Diese erlernen Zusammenhänge zwischen Eingangs- und Ausgangsgrößen der Simulation anhand simulierter Daten. Anschließend können sie zur schnellen Approximation der Ausgangsgrößen verwendet werden. Eine Methode zur Erzeugung solcher Modelle für die Vorhersage der Ausgaben einer Kaltmassivumformsimulation wird in dieser Arbeit entwickelt. Anhand einer Untersuchung verwandter Arbeiten und einer ausführlichen Datenanalyse wird eine Vorgehensweise erarbeitet, die datenspezifische Vorverarbeitung, Feature Engineering und eine Gaußprozess-Regression umfasst. Sie wird mithilfe von Testdaten evaluiert und erreicht eine hohe Vorhersagequalität für sechs von einem Fachexperten ausgewählte Ausgangsgrößen der Simulation. Im Rahmen einer Ablationsstudie werden der Einfluss einzelner Schritte der Vorgehensweise auf die Vorhersagequalität und die Laufzeit des Verfahrens untersucht. Es wird festgestellt, dass sich durch Überspringen des Feature Engineerings der zeitliche Aufwand der Vorgehensweise auf wenige Minuten reduzieren lässt. Dabei bleibt die Vorhersagequalität unverändert hoch. Mit diesen Anpassungen benötigen die Erzeugung eines Surrogatmodells und die Vorhersagen für über hundert Prozesskonfigurationen des Testdatensatzes weniger Zeit, als einen einzelnen Datenpunkt zu simulieren. Da die Vorhersagen komplexer Surrogatmodelle nur schwer nachzuvollziehen sind, werden mit verschiedenen XAI-Verfahren Erklärungen für das Verhalten der erzeugten Modelle produziert. Dabei werden Zusammenhänge zwischen Eingangsgrößen und Vorhersagen aufgedeckt, die zuvor im Rahmen einer Untersuchung der Werteverteilungen und Korrelationen zwischen Eingangs- und Ausgangsgrößen der Simulation nicht aufgefallen sind.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>15</b>
<b>2</b>	<b>Grundlagen</b>	<b>17</b>
2.1	Anwendungsbereich Kaltmassivumformung . . . . .	17
2.2	Datengetriebene Surrogatmodelle . . . . .	20
2.3	Erklärbarkeit . . . . .	30
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>35</b>
3.1	Surrogatmodelle in der Fertigung . . . . .	35
3.2	Surrogatmodelle in der Umformtechnik . . . . .	36
3.3	Erklärbarkeit in der Fertigungstechnik . . . . .	38
3.4	Zentrale Erkenntnisse . . . . .	38
<b>4</b>	<b>Ergebnisse der explorativen Datenanalyse</b>	<b>41</b>
4.1	Datenbeschreibung . . . . .	41
4.2	Datenqualität . . . . .	44
4.3	Datenvorbereitung . . . . .	48
4.4	Datenvisualisierung und Korrelationsanalyse . . . . .	49
4.5	Erste Approximationen . . . . .	55
<b>5</b>	<b>Erarbeitete Vorgehensweise</b>	<b>67</b>
5.1	Datenspezifische Vorverarbeitung . . . . .	68
5.2	Transformation der Ausgangsgröße . . . . .	69
5.3	Aufteilung in Trainings- und Testdaten . . . . .	70
5.4	ML-Pipeline . . . . .	70
<b>6</b>	<b>Implementierung</b>	<b>73</b>
6.1	Genutzte Module . . . . .	74
6.2	Implementierung der Datenanalyse . . . . .	74
<b>7</b>	<b>Diskussion der Evaluationsergebnisse</b>	<b>81</b>
7.1	Evaluation der vollständigen Vorgehensweise . . . . .	82
7.2	Einfluss von Hyperparameterwerten auf die Vorhersagequalität . . . . .	84
7.3	Ablationsstudie . . . . .	85
7.4	Approximation anderer Ausgangsgrößen . . . . .	92
7.5	Erklärbarkeit . . . . .	94
7.6	Zentrale Schlussfolgerungen . . . . .	100
<b>8</b>	<b>Zusammenfassung</b>	<b>103</b>
8.1	Übersicht der Ergebnisse . . . . .	103

8.2	Ausblick . . . . .	104
	<b>Literaturverzeichnis</b>	<b>107</b>

# Abbildungsverzeichnis

2.1	Beispiel für Werkstück und Umformprozess des Napf-Rückwärts-Fließpressens. . .	18
2.2	Darstellung des CRISP-DM. . . . .	21
2.3	Beispiel für einen Boxplot. . . . .	23
2.4	Drei Beispiele für Residualplots. . . . .	29
2.5	Einteilung von XAI-Methoden. . . . .	31
4.1	Darstellung der relativen Veränderung der Stempelgeometrie. . . . .	43
4.2	Boxplots für die Eingangsgrößen. . . . .	46
4.3	Übersicht über die Schritte der Datenvorbereitung. . . . .	48
4.4	Histogramme der sechs untersuchten Ausgangsgrößen. . . . .	50
4.5	Ausschnitt der Streudiagramm-Matrix für die Eignangsgröße Stempelgeometrie und die Ausgangsgröße Napfbodendicke. . . . .	51
4.6	Ausschnitt der Streudiagramm-Matrix für die Eingangsgröße Zyklus und die Ausgangsgröße Temperatur. . . . .	52
4.7	Ausschnitt der Streudiagramm-Matrix für die Eingangsgröße Werkstückgeometrie und die Ausgangsgröße Kraft. . . . .	53
4.8	Streudiagramm für die Ausgangsgröße Wandstärke und die Eingangsgrößen Stempelgeometrie und Matrizengeometrie. . . . .	53
4.9	Zwei Ausschnitte der Streudiagramm-Matrix, die die jeweils stärkste Korrelation der Ausgangsgrößen Umformgrad und Schädigung mit einer Eingangsgröße zeigen. . . . .	54
4.10	Aufbau der AutoML-Modelle. . . . .	59
4.11	Vergleich der Werte der Ausreißerdatenpunkte mit entsprechenden Vorhersagen für die Ausgangsgröße Schädigung. . . . .	62
4.12	Residualplots für die Vorhersagen der Ausgangsgröße Schädigung von Modellen, die mit und ohne Ausreißerdatenpunkte trainiert werden. . . . .	64
4.13	Histogramme der von Ausreißern betroffenen Ausgangsgrößen Kraft und Napfbodendicke in den alten und den neu simulierten Daten. . . . .	65
5.1	Schematische Darstellung der im Rahmen dieser Arbeit entwickelten Vorgehensweise. . . . .	67
5.2	Histogramme der Ausgangsgröße Schädigung vor und nach einer logarithmischen Transformation. . . . .	69
5.3	ML-Pipeline der erarbeiteten Vorgehensweise. . . . .	71
6.1	Darstellung des Python-Packets mit Hilfsfunktionen. . . . .	75
7.1	Residualplots für die Ausgangsgrößen Umformgrad und Schädigung bei Vorhersage durch das jeweils beste gefundene Modell im Rahmen der entwickelten Vorgehensweise. . . . .	83
7.2	Einfluss der Werte ausgewählter Hyperparameter auf den $R^2$ Mittelwert einer Kreuzvalidierung mit fünf Teilmengen. . . . .	85

7.3	Residualplots für die Ausgangsgrößen Umformgrad und Schädigung bei Vorhersage durch das beste Modell, das mit skalarer Bandbreite trainiert wurde. . . . .	87
7.4	Residualplots für die Vorhersage auf Testdaten mit Ausreißerdatenpunkten. . . .	91
7.5	Entscheidungsbäume für die Ausgangsgrößen Umformgrad und Schädigung. . . .	95
7.6	Partial Dependence Plots für die Ausgangsgröße Umformgrad. . . . .	96
7.7	Partial Dependence Plots für die Ausgangsgröße Schädigung. . . . .	97
7.8	Gegenüberstellung von Korrelationskoeffizienten und SHAP Werten. . . . .	99
7.9	Schematische Darstellung der angepassten Vorgehensweise. . . . .	100



# Tabellenverzeichnis

2.1	Beispiel für zwei Ausgangsgrößen und zugehörige Vorhersagen. . . . .	28
4.1	Datendokumentation . . . . .	42
4.2	Eingangsgrößen der Simulation. . . . .	42
4.3	Ausgangsgrößen der Simulation. . . . .	44
4.4	Übersicht über Ausreißer in den Daten. . . . .	47
4.5	Übersicht über die Dateneigenschaften vor und nach der Vorverarbeitung. . . . .	49
4.6	Gegenüberstellung der jeweiligen fünf Werte der Eingangsgröße Stempelgeometrie und der Ausgangsgröße Napfbodendicke. . . . .	56
4.7	Erreichte $R^2$ Werte für Approximationen auf Basis von Wertetabellen. . . . .	56
4.8	Übersicht über die mit simplen Modellen erreichten $R^2$ Werte für die Ausgangsgrößen Umformgrad und Schädigung. . . . .	57
4.9	Übersicht über die mit AutoML-Verfahren erreichten $R^2$ Werte für die Ausgangsgrößen Umformgrad und Schädigung. . . . .	59
6.1	Übersicht über die verwendete Python Distribution sowie relevanter Module. . . . .	73
6.2	Gegenüberstellung eines Teils der Dateinamen und der entsprechenden Werte der Eingangsgröße Reibung. . . . .	76
6.3	Parameterraster für die Rastersuche. . . . .	78
7.1	Charakterisierung der besten Modelle, die für die Ausgangsgrößen Umformgrad und Schädigung durch eine Rastersuche gefunden wurden. . . . .	82
7.2	Werte für die besten Modelle bei Verwendung von Skalaren für die Bandbreite. . . . .	86
7.3	Werte für die besten Modelle, wenn keine polynomiellen Eingangsgrößen gebildet werden. . . . .	87
7.4	Werte für die besten Modelle, wenn die Eingangsgrößen nicht standardisiert werden. . . . .	88
7.5	Werte für die besten Modelle, wenn keine Dimensionsreduktion durchgeführt wird. . . . .	89
7.6	Werte für das besten Modelle zur Vorhersage der Schädigung ohne Transformation der Ausgangsgröße. . . . .	90
7.7	Werte für die besten Modelle, die mit Ausreißerdatenpunkten trainiert wurden. . . . .	90
7.8	Werte für die besten Modelle, die mit der erarbeiteten Vorgehensweise für die vier Ausgangsgrößen Napfbodendicke, Wandstärke, Temperatur und Kraft erzeugt werden können. . . . .	93
7.9	Absolute Korrelationskoeffizienten zwischen Eingangsgrößen und den Ausgangsgrößen Umformgrad und Schädigung. . . . .	98
7.10	Vergleich der genutzten Hardware für Simulation und Surrogatmodell. . . . .	101



## Verzeichnis der Listings

6.1	Codeausschnitt, der die Verwendung des Moduls preprocessing veranschaulicht und die Erzeugung des Testdatensatzes darstellt. . . . .	77
6.2	Beispiel für einen Aufruf von auto-sklearn. . . . .	77
6.3	Beispiel für eine Rastersuche für die Ausgangsgröße Schädigung. . . . .	79



# Abkürzungsverzeichnis

- AutoML** automatisiertes maschinelles Lernen. 16
- CRISP-DM** Cross-Industry Standard Process for Data Mining. 21
- DM** Data Mining. 20
- DSGVO** Datenschutz-Grundverordnung. 30
- GP** Gaußprozess. 16
- KRR** Kernel Ridge Regression. 26
- MAE** Mean Absolute Error. 28
- ML** maschinelles Lernen. 15
- PDP** Partial Dependence Plot. 32
- RBF** radiale Basisfunktion. 27
- RMSE** Root Mean Squared Error. 28
- RSS** Residual Sum of Squares. 24
- SHAP** Shapley additive explanations. 16
- VM** virtuelle Maschine. 73
- XAI** eXplainable Artificial Intelligence. 16



# 1 Einleitung

Die Kaltmassivumformung ist ein Fertigungsverfahren der Hauptgruppe Umformen [DIN8580] und zeichnet sich im Gegensatz zu trennenden Verfahren durch den Erhalt von Stoffzusammenhalt und Masse aus [DIN8582]. Eine Unterkategorie dieser Verfahren stellt das Napf-Rückwärts-Fließpressen dar, das symmetrische, becherförmige Teile produziert [PGK+23]. Vorteile dieses Verfahrens sind eine effiziente Werkstoffausnutzung, kurze Fertigungszeiten und hohe Produktqualität [Fri22; LFM17]. Jedoch handelt es sich um einen komplexen Prozess, in dem große Kräfte wirken. Es kann daher zu Brüchen des Werkzeugs oder des zu bearbeitenden Teils kommen [PGK+23]. Dabei muss das Formveränderungsvermögen dieses Verfahrens aufgrund wirtschaftlicher Zwänge umfassend ausgenutzt werden [Bre07]. Denn die Fertigung steht vor der Herausforderung, immer vielfältigere, hochqualitative Produkte herzustellen, während immer weniger Zeit für die Entwicklung zur Verfügung steht [DDL+20]. Daher ist die Optimierung von Fertigungsprozessen unerlässlich [DB16; KCL21]. Zu diesem Zweck hat sich der Einsatz numerischer Simulationen durchgesetzt [AAM20; CET21]. Diese können das Materialverhalten während der Fertigung präzise vorhersagen und zu einer kostengünstigen Produktion beitragen, indem sie physische Experimente ersetzen. Genaue Simulationen sind jedoch rechenintensiv und können Laufzeiten von mehreren Tagen aufweisen [DB16; Fri22; ZZHM23]. Darüber hinaus müssen sie von geschultem Fachpersonal durchgeführt werden und können mit hohen Lizenzkosten verbunden sein [DB16].

Eine Alternative zu Computersimulationen stellen datengetriebene Surrogatmodelle dar, die mit Methoden des maschinellen Lernens (ML) erzeugt werden können. Diese sind in der Lage, Zusammenhänge zwischen den Eingangs- und Ausgangsgrößen einer Simulation zu erlernen. Anschließend können sie für weitere Werte der Eingangsgrößen eine Annäherung der Simulationsergebnisse ausgeben [CCC19; DMA17; QSJ+05]. Die Anwendung von Surrogatmodellen kann aufgrund ihrer geringen Anforderungen an Rechenzeit und Computerressourcen die Kosten der Optimierung senken [PA05] und gewinnt daher an Bedeutung für die Fertigungstechnik [BAC+19; HSSK05; WWIT16]. Der Einsatz solcher Modelle im Anwendungsbereich der Kaltmassivumformung ist bisher allerdings nicht weit verbreitet [ACS22].

Ziel dieser Arbeit ist es, eine Vorgehensweise zu entwickeln, anhand derer datengetriebene Surrogatmodelle zur Approximation der Ausgangsgrößen einer Kaltmassivumformsimulation erzeugt werden können. Als Grundlage dient dabei ein vom Institut für Umformtechnik der Universität Stuttgart zur Verfügung gestellter Datensatz einer Simulation eines Prozesses des Napf-Rückwärts-Fließpressens. Im Rahmen dieser Simulation werden verschiedene Prozessparameter wie die Werkzeuggeometrie oder ein Reibfaktor des verwendeten Schmiermittels variiert und ein vollständiger Versuchsplan durchgeführt. Sechs Ausgangsgrößen der Simulation wurden von einem Fachexperten des Instituts für Umformtechnik ausgewählt, die approximiert werden sollen. Dazu zählen unter anderem die Dicke des Bodens, der Umformgrad und die Schädigung des becherförmigen Produkts.

Zur Entwicklung der Vorgehensweise werden verwandte Arbeiten gesichtet, die den Einsatz von Surrogatmodellen im Bereich der Fertigung und der Kaltmassivumformung im Speziellen behandeln. Darüber hinaus werden die zur Verfügung stehenden Daten umfassend untersucht. Dazu gehört eine Analyse der Verteilungen und Korrelationen der Größen im Datensatz sowie eine Aufbereitung der Daten, um ihre Eignung für die Anwendung von ML-Verfahren zu erhöhen. Das anschließende Training simpler Modelle, zum Beispiel auf Basis der linearen Regression, sowie die Anwendung von Verfahren des automatisierten maschinellen Lernens (AutoML) [FKE+15] geben überdies Aufschluss über geeignete Vorverarbeitungsschritte und ML-Algorithmen. Anhand der gesammelten Erkenntnisse wird ein Vorgehen bestehend aus datenspezifischen Vorverarbeitungsschritten, Feature Engineering sowie einer Gaußprozess-Regression (GP-Regression) entwickelt. Dieses wird mittels der zur Verfügung stehenden Daten erprobt und hinsichtlich der Vorhersagequalität und des zeitlichen Aufwands zur Durchführung von Training und Vorhersagen evaluiert. Dabei wird festgestellt, dass die erzeugten Surrogatmodelle sechs von einem Fachexperten ausgewählte Ausgangsgrößen der Simulation präzise approximieren können. Die Ergebnisse der entwickelten Vorgehensweise werden mit Ergebnissen von Modellen auf Basis von Korrelationen oder AutoML-Verfahren verglichen. Diese Vergleiche ergeben, dass die Modelle der erarbeiteten Vorgehensweise die höchste Vorhersagequalität erreichen und weniger komplex als AutoML-Modelle sind. In Rahmen einer Ablationsstudie wird darüber hinaus der Einfluss einzelner Schritte des Verfahrens auf die Vorhersagequalität und die Laufzeit überprüft und sinnvolle Anpassungen der Vorgehensweise identifiziert. Diese ermöglichen eine Reduzierung der Laufzeit des Verfahrens auf wenige Minuten bei gleichbleibender Vorhersagequalität. Zuletzt wird die Erklärbarkeit der erzeugten Modelle behandelt. Mithilfe von Ansätzen der eXplainable Artificial Intelligence (XAI) [Fri01; MPC22] werden Erklärungen für das Verhalten der Modelle erzeugt. Dabei werden verschiedene Verfahren angewendet und ihre Ergebnisse verglichen. Die Berechnung von Shapley additive explanations (SHAP) [LL17] offenbart Zusammenhänge zwischen Eingangsgrößen und Vorhersagen, die andere Verfahren nicht erkennen.

Diese Arbeit ist wie folgt aufgebaut: In Kapitel 2 werden Grundlagen des Anwendungsgebiets der Kaltmassivumformung umrissen sowie die Hintergründe von mit ML-Verfahren erzeugten datengetriebenen Surrogatmodellen und eingesetzte Verfahren zur Datenanalyse erläutert. Kapitel 3 befasst sich mit Arbeiten anderer Autoren, die Surrogatmodelle im Bereich der Fertigungstechnik und der Kaltmassivumformung im Speziellen einsetzen. Anschließend werden in Kapitel 4 die Ergebnisse einer explorativen Datenanalyse vorgestellt, in deren Rahmen Verteilungen und Korrelationen in den Daten untersucht und erste Surrogatmodelle erzeugt und begutachtet werden. Daraufhin werden in Kapitel 5 die erarbeitete Vorgehensweise vorgestellt und in Kapitel 6 Implementierungsdetails besprochen. In Kapitel 7 werden die Ergebnisse der Evaluation des Vorgehens auf den zur Verfügung gestellten Daten präsentiert. Dazu gehört auch die Ablationsstudie sowie die die Anwendung von XAI-Verfahren. Zuletzt wird in Kapitel 8 eine Zusammenfassung der Arbeit und ein Ausblick auf noch zu untersuchende Aspekte gegeben.



## 2 Grundlagen

In diesem Kapitel werden die Hintergründe dieser Arbeit genauer beleuchtet sowie relevante Begriffe und Konzepte eingeführt. Dabei wird sowohl auf das Anwendungsgebiet der Umformtechnik und die in diesem Rahmen eingesetzten Simulationen eingegangen, als auch auf die Grundlagen datengetriebener Surrogatmodelle, die mit Methoden des maschinellen Lernens (ML) erstellt werden. Zudem wird die Erklärbarkeit solcher Modelle thematisiert.

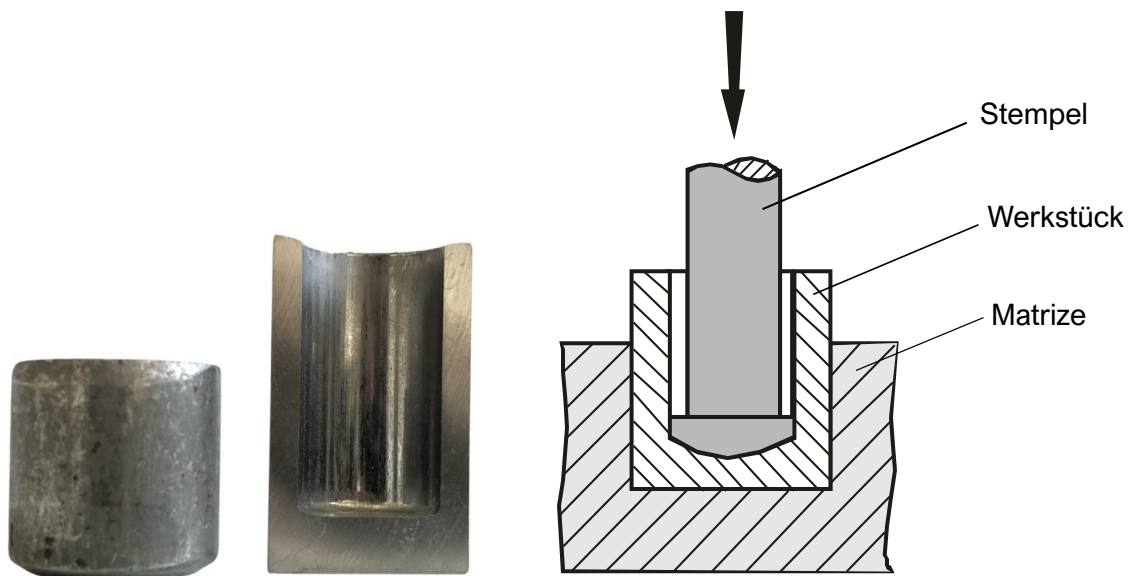
### 2.1 Anwendungsbereich Kaltmassivumformung

Dieser Arbeit liegen die Daten der Simulation eines Prozesses des Napf-Rückwärts-Fließpressens zugrunde. Dabei handelt es sich um einen Vorgang, der zu den Verfahren der Kaltmassivumformung gezählt wird. Im Folgenden wird dieser Anwendungsbereich näher beschrieben und sowohl eine kurze Einführung in die Kaltmassivumformung gegeben als auch der aktuelle Einsatz von Simulationen in diesem Kontext beleuchtet.

#### 2.1.1 Fertigungstechnik Kaltmassivumformung

Die Kaltmassivumformung ist eine Unterkategorie des Umformens, einer von sechs per DIN Norm definierten Hauptgruppen von Fertigungsverfahren [Bre07; DIN8580]. Die Präfixe kalt und massiv beziehen sich auf Merkmale des Werkstücks, also des Teils, das umgeformt wird. Bei der Kaltumformung wird das Werkstück im Gegensatz zur Warmumformung, nicht von außen erwärmt [Bre07]. Werden Bleche als Werkstück eingesetzt, spricht man von Blechumformung. Die Massivumformung bearbeitet dagegen massive Rohteile. Diese zeichnen sich durch in allen drei Raumrichtungen ähnliche Abmessungen aus [DB16; DIN8582]. Ob es sich um Massiv- oder Blechumformung handelt, bestimmt dementsprechend die geometrische Form des Werkstücks. Beispiele für Werkstücke der Massivumformung sind Stangenabschnitte oder Vorpresslinge [Fri22], wie beispielhaft in Abbildung 2.1a dargestellt. Beim Umformen verändern sich Stoffzusammenhalt sowie Masse des Werkstücks nicht [DIN8582; Küh22] und sein Volumen kaum [Fri22]. Zu den Vorteilen dieser Fertigungstechnik gegenüber trennenden Verfahren, also solchen, die den Stoffzusammenhalt auflösen, gehören eine bessere Werkstoffausnutzung, kürzere Fertigungszeit und höhere Werkstückqualität. Darüber hinaus kann die Kaltumformung die Festigkeit des Werkstücks erhöhen [Fri22]. Das Umformen eignet sich insbesondere für Bereiche, in denen eine hohe Zuverlässigkeit und Lebensdauer der Produkte benötigt wird [Bre07].

Zu den Verfahren der Kaltmassivumformung zählt das Napf-Rückwärts-Fließpressen. Diese Technik ist insbesondere unter den Zulieferern der Automobilindustrie verbreitet [Fri22; PGK+23]. Sie zeichnet sich durch hohe Festigkeit der Erzeugnisse, Stoffeinsparungen gegenüber anderen



(a) Beispiel für ein Werkstück vor (links) und Querschnitt des Werkstücks nach (rechts) der Umformung. Abbildung zur Verfügung gestellt vom Institut für Umformtechnik der Universität Stuttgart.

(b) Schematische Darstellung des Umformprozesses. Eigene Darstellung nach [DB16].

**Abbildung 2.1:** Beispiel für Werkstück und Umformprozess des Napf-Rückwärts-Fließpressens.

Verfahren und einen hohen Durchsatz aus [Fri22; LFM17; PGK+23]. Ein einfacher Napf-Rückwärts-Fließpressen-Prozess ist beispielhaft in Abbildung 2.1 dargestellt. In Abbildung 2.1a sieht man links ein Werkstück. Der Prozess läuft wie in Abbildung 2.1b dargestellt ab. Das Werkstück wird in das Werkzeug gegeben. Daraufhin fährt der obere Teil des Werkzeugs, auch Stempel genannt, herunter und wirkt eine Kraft auf das Werkstück. Dabei beschränkt die Matrize die Formveränderung des Werkstücks seitlich und nach unten. Das Werkstück verformt sich in die einzig mögliche Richtung, entgegen der Kraftereinwirkung seitlich am Stempel vorbei. Es entsteht ein becherförmiges Endprodukt, wie auf der rechten Seite von Abbildung 2.1a zu sehen.

Das Napf-Rückwärts-Fließpressen ist ein komplexer Prozess, in dem enorme Kräfte wirken, was zu Brüchen von Werkstück oder Werkzeug führen kann [PGK+23]. Doch um in einem kompetitiven Markt, der immer vielfältigere Produkte bei immer kürzerer Entwicklungszeit fordert, erfolgreich zu sein, muss dieser Prozess so effizient wie möglich gestaltet werden [DDL+20; HYDF16; TKE+17]. Dementsprechend hat die Optimierung von Fertigungsprozessen einen hohen Stellenwert [KCL21]. Dabei sind folgende Maßnahmen von Bedeutung [KBT11]:

- Identifizierung von Faktoren mit maßgeblichem Einfluss auf die Produktqualität.
- Aufdecken der Zusammenhänge zwischen Prozessparametern und Qualitätseigenschaften.
- Vorhersage der Produktqualität bei gegebenen Prozessparametern.

Auch die Beziehungen zwischen den Prozessparametern sind von Interesse [GFW+19]. Die so gewonnenen Informationen sind unerlässlich für die Entwicklung von Umformprozessen [KBT11] und können genutzt werden, um geeignete Maschinenparameter zu finden [TKE+17].

In der Fertigungstechnik sind Optimierungsprobleme häufig hochdimensional und nicht linear. Zusätzlich steigt ihre Komplexität beständig [TKE+17]. Die Anzahl der zu optimierenden Parametern variiert je nach Prozess zwischen weniger als zehn und mehreren hundert [PZL+18]. Diese werden häufig durch eine werkstückspezifische Versuch-und-Irrtum-Vorgehensweise festgelegt. Der Erfolg dieser Methode hängt stark von dem Fachwissen und der Erfahrung der durchführenden Person ab [DDL+20; PZL+18; TGH+18]. Häufig sind viele Iterationen notwendig, bis eine zufriedenstellende Konfiguration gefunden wird. Diese Form der Optimierung benötigt daher viel Zeit und verursacht hohe Kosten [DDL+20; PZL+18]. Eine Alternative zu ineffizienten Experimenten stellen numerische Simulationen dar [TGH+18].

### 2.1.2 Simulationen in der Kaltmassivumformtechnik

Computersimulationen ermöglichen präzise Vorhersagen des Materialverhaltens, wie sie für die Optimierung von Fertigungsprozessen benötigt werden. Ihr Einsatz zur Analyse physikalischer Prozesse hat sich etabliert [AAM20], mitunter werden sie als „Goldstandard“ für die Prozessoptimierung bezeichnet [CET21]. Sie ermöglichen es, den Einfluss von veränderbaren Prozessparametern auf Werkzeug und Werkstück zu untersuchen [TKE+17].

Simulationen werden erfolgreich zur Modellierung von Umformprozessen eingesetzt, haben allerdings auch ihre Nachteile. Insbesondere im Bereich der Kaltmassivumformung sind die verwendeten Simulationen häufig deterministisch [ACS22; Bre07; VYC+23]. In der Praxis, das heißt in der tatsächlichen Fertigung, unterliegen Größen wie Materialeigenschaften, geometrische Gestaltungen und Schmierung jedoch Schwankungen, die nur schwer kontrolliert werden können. Diese als deterministisch zu betrachten, kann zu Problemen führen, wenn eine Prozesskonfiguration am Rand des Machbaren gewählt wird. Abweichungen der oben genannten Größen können dann zu Fehlern wie Brüchen im Werkstück führen [ACS22]. Darüber hinaus werden bei Simulationen physikalischer Vorgänge zwangsweise vereinfachende Annahmen gemacht, sodass ein Teil der zugrundeliegenden Zusammenhänge ignoriert wird [BCNA21; TGH+18]. Auch die numerische Natur von Simulationen kann das Endergebnis verfälschen [RSK20]. Eine solche muss daher anhand von physischen Experimenten validiert werden [KCL21]. Aus diesen Gründen sind Fachexperten erforderlich, um Simulationen von Umformprozessen durchzuführen. Diese definieren relevante Prozessparameter, entsprechende Wertebereiche und gesuchte Ausgangsgrößen. Sie variieren in einem iterativen Vorgang die Eingangswerte auf Basis begründeter Vermutungen, um ein zuvor spezifiziertes Ergebnis zu erreichen [FBS15]. Diese Entscheidungen basieren auf Fachwissen und Erfahrung, weshalb sie nur von Fachexperten getroffen werden können [DDL+20; PZL+18; TGH+18].

Detaillierte Simulationen physikalischer Zusammenhänge sind bekannt dafür, viel Rechenzeit zu benötigen [BCNA21; CCC19; HBFP22; RSK20; TKE+17; ZZHM23]. Trotz der steten Weiterentwicklung von Computersystemen und der damit einhergehenden Steigerung der zur Verfügung stehenden Rechenkapazitäten kann ein Simulationsdurchlauf mehrere Tage andauern [DMA17; PZL+18]. Hinzu kommt, dass Simulationen zur Prozessoptimierung wiederholt ausgeführt werden, mitunter tausende Male [AAM20; FBS15; QSJ+05]. Daher ist es notwendig, die Abdeckung

von Parameterräumen bei der Simulation von Umformprozessen einzuschränken [RSK20]. Das erschwert es wiederum, die Gesetzmäßigkeiten des zugrundeliegenden physikalischen Prozesses zu erkennen [TKE+16], was die Eignung von Simulationen für die Prozessoptimierung einschränkt [VYC+23].

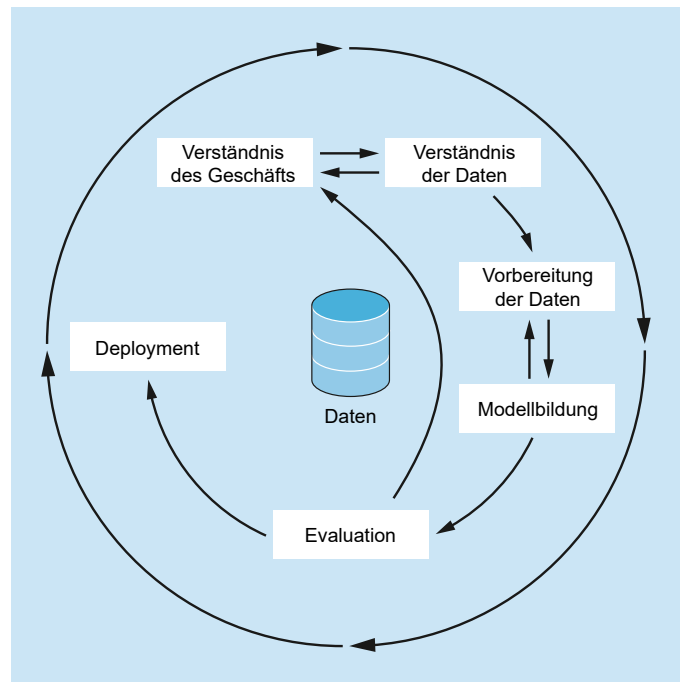
## 2.2 Datengetriebene Surrogatmodelle

Im Folgenden werden die datengetriebenen Surrogatmodelle näher beleuchtet. Zunächst wird auf den Einsatz solcher Modelle im Bereich der Kaltmassivumformung eingegangen. Anschließend werden die Konzepte Data Mining (DM) sowie maschinelles Lernen (ML) und deren Zusammenhänge mit Surrogatmodellen thematisiert. Es folgt ein Abschnitt über die Theorie verschiedener Modelle, deren Komponenten und Eigenschaften, einschließlich zugrundeliegender Gleichungen. Zuletzt wird auf die Methoden zum Test von Surrogatmodellen eingegangen.

### 2.2.1 Surrogatmodelle in der Kaltmassivumformtechnik

Erste Anwendungen von DM im Bereich der Umformtechnik haben in den 1990er Jahren stattgefunden [DB21; ICFQ93]. Seither haben DM-Verfahren an Bedeutung für die Fertigungstechnik gewonnen [HSSK05; PA05; TKE+16; WWIT16]. Sie können zur Optimierung von Produkten und Fertigungsprozessen eingesetzt werden [BAC+19; BCNA21] und Unternehmen dadurch einen kompetitiven Vorteil verschaffen [DB21; TKE+17]. Der Einsatz von DM bietet sich an, wenn konventionellere Ansätze wie Simulationen impraktikabel sind [GC03]. Dabei können mit ML-Methoden datengetriebene Surrogatmodelle erstellt werden, die in der Lage sind, Zusammenhänge zwischen Ein- und Ausgangsgrößen zu erlernen, zugrundeliegende Gesetzmäßigkeiten vereinfacht darzustellen und schnelle Vorhersagen zu machen [CCC19; QSJ+05]. Es ist möglich, Surrogatmodelle anhand von Daten physischer Experimente zu trainieren. Allerdings bieten sie sich insbesondere an, um die Ausgaben von detaillierten Simulationen anzunähern [BAC+19; BCNA21; DMA17]. Dabei wird kein Fachwissen des Anwendungsbereichs, beispielsweise der Umformtechnik, benötigt [XZZ21b]. Allerdings sind die Ausgaben von Surrogatmodellen lediglich eine Annäherung an die wahren Werte der Ausgangsgrößen und liefern möglicherweise keine optimale Prozesskonfiguration [PZL+18]. Trotzdem können die Vorhersagen aufgrund ihrer geringeren Anforderungen an Rechenzeit und Computerressourcen die Kosten der Optimierung senken [PA05].

Die Fähigkeit von Surrogatmodellen, die Ausgaben von Simulationen zu approximieren, kann dazu genutzt werden, den Raum der Eingangsgrößen zu erkunden. Im Fall der Umformtechnik können dadurch vielversprechende Prozessparameter identifiziert werden [AAM20; BCNA21; PGK+23]. Eine häufige Herangehensweise ist es, diese Kandidaten im Anschluss mit aufwendigeren Methoden, wie zum Beispiel einer Simulation, zu untersuchen [DMA17; ZXNL21]. Surrogatmodelle eignen sich aufgrund ihrer geringeren Laufzeit insbesondere für die iterative Natur einer Prozessoptimierung [VYC+23]. Sie sind auch in der Lage, die komplexen, nicht linearen Zusammenhänge, wie sie in der Umformtechnik vorherrschen, zu erlernen [WWIT16]. Daher gelten sie als vielversprechend in diesem Anwendungsgebiet [ZXNL21], ihr Einsatz ist jedoch noch nicht weit verbreitet [ACS22]. Als größte Herausforderungen dabei werden häufig Eigenschaften der Daten genannt, wie zum Beispiel die Umwandlung von kategorischen in numerische Werte [DRF+22] oder die schwer



**Abbildung 2.2:** Eigene Darstellung des CRISP-DM nach [Alb22].

zu interpretierenden Größen in den Daten [DB21; GFW+19]. Wird ein datengetriebenes Modell anhand von Simulationsdaten erstellt, ist auch die Genauigkeit der Simulation von entscheidender Bedeutung [TGH+18].

### 2.2.2 Begriffsbestimmungen von Data Mining und maschinellem Lernen

In der Literatur sind die Definitionen von DM nicht eindeutig und übereinstimmend. Eine präzise Auseinandersetzung mit dieser Mehrdeutigkeit und der Abgrenzung von anderen Begriffen wie Knowledge Discovery geht über den Rahmen dieser Arbeit hinaus. Im Folgenden wird DM als Überbegriff für einen Prozess verwendet, welcher der Extraktion von Mustern und Zusammenhängen aus Daten dient [CHT09; Han12]. Beispielhaft kann der weit verbreitete Cross-Industry Standard Process for Data Mining (CRISP-DM) herangezogen werden [HSSK05]. Dieser ist in Abbildung 2.2 dargestellt. Von den sechs Schritten sind für diese Arbeit lediglich drei relevant:

- Verständnis der Daten, das mithilfe einer explorativen Datenanalyse, wie sie in Kapitel 4 durchgeführt wird, erlangt werden kann [Alb22].
- Vorbereitung der Daten, inklusive notwendiger Verarbeitungs- und Transformationsschritte.
- Modellbildung, was die Modellselektion, das Training und eine Bewertung des erstellten Modells umfasst.

Die Vielzahl an Pfeilen in der Grafik verdeutlicht die iterative Natur eines DM-Projekts. Häufig werden Schritte nicht strikt hintereinander ausgeführt und wiederholt.

Im Bereich von DM bedient man sich häufig Verfahren des maschinellen Lernens [Han12]. Diese werden dem Gebiet der künstlichen Intelligenz zugeordnet [Alb22]. Beim Einsatz von ML geht es darum, Computer dazu zu verwenden, Aufgaben zu lösen, wobei lediglich Daten zur Verfügung gestellt werden [Han12; NM15]. Insbesondere grenzt sich diese Vorgehensweise von sogenannten Expertensystemen ab, in denen beispielsweise von Fachwissen abgeleitete Regeln gespeichert sind.

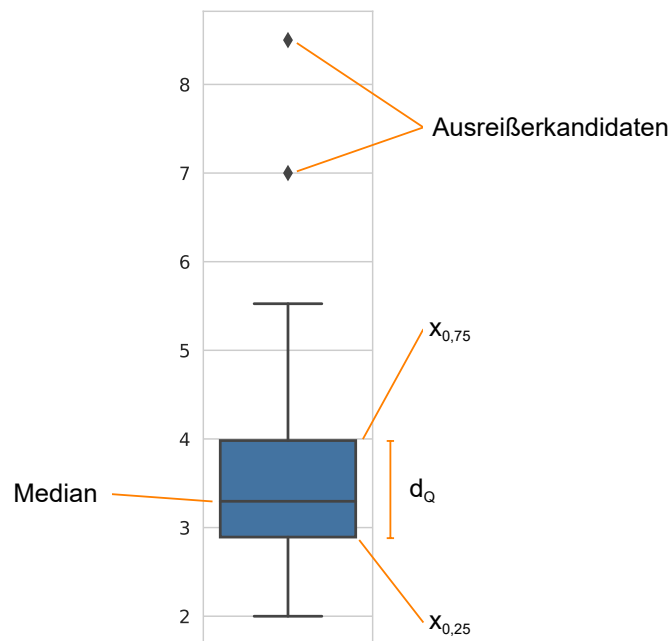
### 2.2.3 Datenvorbereitung

Da die hier beschriebenen Surrogatmodelle lediglich anhand von Daten trainiert werden, haben die Eigenschaften des Datensatzes erheblichen Einfluss auf die zu erreichende Approximation. Die Daten können so angepasst werden, dass eine Vorhersage durch ein Surrogatmodell begünstigt wird. Einige dabei verwendete Verfahren werden im Folgenden vorgestellt. Zuerst wird auf Ausreißer eingegangen, anschließend wird die Anpassung der Werteverteilung von Größen thematisiert.

Daten können Ausreißer beinhalten. Unter einem solchen versteht man einen einzelnen Wert, der sich wesentlich von den übrigen unterscheidet [FHK+16]. Zusätzlich wird im Folgenden auch der Begriff Ausreißerdatenpunkt verwendet. Damit werden Datenpunkte bezeichnet, die mindestens einen solchen auffälligen Wert, also einen Ausreißer in einem ihrer Attribute beinhalten. Ausreißer können fehlerhafte Daten sein und die Vorhersage mittels ML-Verfahren erschweren [Pla21]. Eine Methode zur Identifizierung von Ausreißerkandidaten, also Werten, bei denen es sich möglicherweise um Ausreißer handelt, stützt sich auf den Interquartilsabstand [Alb22]. Das untere Quartil ist über einen Schwellenwert  $x_{0,25}$  definiert, wobei mindestens 25 % aller Werte einer Größe kleiner oder gleich  $x_{0,25}$  und mindestens 75 % größer oder gleich  $x_{0,25}$  sind. Für das obere Quartil  $x_{0,75}$  gilt das entsprechend umgekehrt. Anhand von  $x_{0,25}$  und  $x_{0,75}$  kann der Interquartilsabstand  $d_Q = x_{0,75} - x_{0,25}$  berechnet werden. Liegt ein Wert um mehr als  $1,5 \cdot d_Q$  über dem oberen Quartil oder unter dem unteren Quartil, so handelt es sich dabei um einen Ausreißerkandidaten [FHK+16; Pla21].

Zur grafischen Identifikation solcher Werte eignen sich Boxplots [Alb22; Han12]. In Abbildung 2.3 ist beispielhaft ein Boxplot abgebildet. Die blaue Box deckt den Bereich zwischen dem unteren und dem oberen Quartil ab, ihre Länge stellt folglich den Interquartilsabstand  $d_Q$  dar. Der horizontale Strich in der Box markiert den Median. Die Striche außerhalb der Boxen nennt man Antennen, auf Englisch auch whiskers. Ihre Bedeutung ist in der Literatur nicht eindeutig bestimmt [Han12]. Hier zeigen sie den minimalen, beziehungsweise maximalen Wert, wobei sie nicht länger als  $1,5 \cdot d_Q$  sind. Dementsprechend endet die untere Antenne im Beispiel bei dem minimalen Wert  $x_{min}$ . Die obere Antenne endet dagegen bei  $x_{0,75} + 1,5 \cdot d_Q$ . Der Maximalwert  $x_{max}$  liegt noch darüber und ist als Raute markiert. Die Rauten stellen Ausreißerkandidaten dar.

Es stellt sich die Frage, wie mit Ausreißern und den betroffenen Datenpunkten verfahren werden soll. Eine Möglichkeit ist, die Ausreißer im Rahmen einer Imputation mit möglichst plausiblen Werten zu ersetzen [Pla21]. Dabei ist das Ziel, die Verzerrung von Analyseergebnissen zu minimieren, nicht jedoch, die korrekten Werte zu finden. Eine Imputation kann durch das Einsetzen von Lageparametern wie dem Mittelwert oder dem Modus erfolgen. Darüber hinaus sind auch aufwendigere Methoden mittels Regressionsverfahren denkbar. Eine einfache und häufig eingesetzte Alternative zur Imputation ist die Complete-Case-Analyse [DB21; KBT11; Pla21]. Dabei werden die betroffenen Datenpunkte entfernt und bei der weiteren Untersuchung nicht berücksichtigt. Ihr Nachteil liegt darin, einen Teil



**Abbildung 2.3:** Beispiel für einen Boxplot.

der Daten für die Analyse zu verlieren. Immerhin sind etwa 10 % der zur Verfügung stehenden Datenpunkte von Ausreißern in den Ausgangsgrößen betroffen. Dabei gilt neben der Datenqualität auch die Datenmenge als Indikator für den Erfolg eines ML-Projekts [BH21].

Mitunter liegt den Verfahren, mit denen Surrogatmodelle erzeugt werden, die Annahme zugrunde, dass die Ausgangsgröße normal verteilt ist. Das trifft zum Beispiel auf die lineare Regression zu. Daher ist empfehlenswert, lediglich normal verteilte Ausgangsgrößen zu verwenden [FHK+16]. Falls notwendig kann eine annähernde Normalverteilung durch eine Transformation der Größe, beispielsweise durch  $Y^m$  oder  $\ln Y$  erreicht werden. Auch für die Eingangsgrößen empfiehlt sich eine Skalierung, um den Einfluss einzelner Größen auf die zu optimierende Funktion unabhängig vom jeweiligen Wertebereich zu machen. Jede Eingangsgröße kann beispielsweise so angepasst werden, dass alle Werte in ein spezifisches Intervall wie  $-1$  bis  $1$  oder  $0$  bis  $1$  fallen. Diesen Vorgang nennt man Normalisierung. Der Einsatz solcher Verfahren ist insbesondere dann angezeigt, wenn ML-Methoden verwendet werden, die Distanzen berechnen. Dazu gehören unter anderen die später in diesem Kapitel beschriebenen Kernel Methoden [Han12]. Standardisierung wird ebenfalls zu diesem Zweck verwendet und funktioniert ähnlich: Hier wird von jedem Wert einer Größe der Mittelwert  $\mu$  subtrahiert, anschließend wird durch die Standardabweichung  $\sigma$  dividiert [FHK+16; Pla21]:

$$(2.1) \quad X_{\text{stand}} = \frac{X - \mu}{\sigma}$$

## 2.2.4 Theorie datengetriebener Surrogatmodelle

Im Rahmen dieser Arbeit werden Surrogatmodelle mithilfe von überwachten Lernverfahren, einer Unterkategorie von ML-Methoden, erstellt. Dabei besteht das Problem darin, eine Abbildung  $f : X \rightarrow Y$  von Eingangsgrößen  $x_1, x_2, \dots, x_D = \mathbf{x} \in X$  auf eine Ausgangsgröße  $y \in Y$  zu finden. Diese Abbildung muss anhand von  $N$  Datenpunkten  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$  erlernt werden [Mur22]. Ist  $Y$  kategorisch, so spricht man von einer Klassifikation. Handelt es sich bei  $Y$  dagegen um eine numerische Größe, so handelt es sich um eine Regression. Da in den im Rahmen dieser Arbeit untersuchten Daten lediglich numerische Ausgangsgrößen enthalten sind, wird im Folgenden die Regression betrachtet.

### Modellerzeugung

Die Abbildung  $f$ , auch Hypothese genannt, wird aus einem Hypothesenraum der Form

$$(2.2) \quad \mathcal{F} = \{f(\cdot; \theta) : X \rightarrow Y \mid \theta \in \mathcal{P}_\alpha\}_{\alpha \in \mathcal{H}}$$

gewählt. Dieser ist durch die Hyperparameter  $\alpha \in \mathcal{H}$  parametrisiert, wobei  $\mathcal{H}$  den Raum der Hyperparameter beschreibt. Der Raum der Modellparameter  $\mathcal{P}_\alpha$  wird bestimmt durch die Hyperparameter  $\alpha$ . Nachfolgend wird die Optimierung der Modellparameter im Rahmen des Trainings beschrieben. Anschließend wird auf die Modellauswahl, also die Wahl eines Hypothesenraums inklusive der Hyperparameter, eingegangen. Diese baut in der hier verwendeten Variante auf der Optimierung der Modellparameter auf.

Im Rahmen des Trainings werden die Modellparameter  $\theta \in \mathcal{P}_\alpha$  festgelegt. Häufig wird dazu die Methode der kleinsten Quadrate, auf Englisch Residual Sum of Squares (RSS), verwendet:

$$(2.3) \quad RSS = \sum_{i=1}^N (f(\mathbf{x}_i; \theta) - y_i)^2$$

RSS quantifiziert den quadratischen Fehler des Modells durch die Differenz zwischen den Ausgaben  $f(\mathbf{x}_i; \theta)$  und der Ausgangsgröße  $y_i$ . Um diesen möglichst gering zu halten, werden diejenigen Modellparameter  $\theta$  gesucht, die RSS minimieren. Das zur Optimierung von Modellparameters eingesetzte Verfahren hängt von der Hypothese ab und wird hier nicht näher behandelt.

Zur Auswahl von Hyperparametern kann eine Rastersuche eingesetzt werden [Pla21]. Dabei wird ein  $m$ -dimensionales Raster erzeugt, wobei eine Dimension einem von  $m$  Hyperparametern entspricht. Dieses wird mit Kandidatenwerten für die Parameter befüllt und es werden alle Kombinationen getestet. Das Modell und damit die gewählten Parameter werden unter Zuhilfenahme eines Validierungsdatensatzes oder einer Kreuzvalidierung überprüft. Bei Letzterer werden die Trainingsdaten in  $n$  gleich große Teilmengen aufgeteilt. Dabei ist die Schnittmenge zweier beliebiger Teilmengen die leere Menge und die Vereinigung aller Teilmengen die ursprüngliche Menge. Anschließend wird auf den Daten von  $n - 1$  Teilmengen ein Modell trainiert, und auf der übrigen Teilmenge getestet. Dieses Vorgehen wird  $n$ -mal wiederholt, wobei jede Teilmenge einmal zur Überprüfung des Modells verwendet wird. Ausgegeben wird dabei eine Liste von Werten einer oder mehrerer Metriken, anhand derer die Vorhersagen unter unterschiedlichen Hyperparameterwerten verglichen werden können.



### Modellkomplexität

Ein ML-Modell sollte so simpel wie möglich und so komplex wie nötig sein [NM15]. Zu den Methoden mit geringer Komplexität zählt die lineare Regression. Dabei wird jeder Eingangsgröße ein Gewicht  $w \in \mathbf{w}$  zugeordnet. Die Vorhersage wird dann als Linearkombination von Eingangsgrößen erzeugt:

$$(2.4) \quad f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

In der in Gleichung (2.4) dargestellten, einfachen Form können nur Zusammenhänge abgedeckt werden, die linear im Raum der Eingangsgrößen sind. Um diese Einschränkung zu umgehen, kann die Komplexität des Verfahrens gesteigert werden, indem die Eingangsgrößen mithilfe einer Funktion  $\phi(\mathbf{x})$  in einen höher-dimensionalen Raum projiziert werden:

$$(2.5) \quad f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

Ein Beispiel für eine Transformation wie sie eine Funktion  $\phi$  darstellen kann, ist die Erzeugung polynomieller Eingangsgrößen bis zum Grad 2. Dabei werden aus je zwei Eingangsgrößen  $x_1$  und  $x_2$  drei zusätzliche Größen berechnet. Bei letzteren handelt es sich jeweils um das Quadrat von  $x_1$  und  $x_2$  sowie die Multiplikation  $x_1 x_2$ . Die ursprünglichen Eingangsgrößen  $x_1$  und  $x_2$  bleiben dabei erhalten. Unabhängig von der Anzahl der Eingangsgrößen kommt zusätzlich eine konstante Größe hinzu. Diese Transformation ist nachfolgend für zwei Eingangsgrößen dargestellt:

$$(2.6) \quad [x_1, x_2] \mapsto [1, x_1, x_2, x_1^2, x_1 x_2, x_2^2]$$

Durch eine solche Transformation steigt die Anzahl der auszuwertenden Größen und der zu berechnenden Gewichte. Zusätzlich muss die höher-dimensionale Repräsentation der Eingangsgrößen berechnet werden. Die gesteigerte Komplexität des Verfahrens kann eine bessere Anpassung an die Daten ermöglichen, hat aber einen erhöhten Rechenaufwand zur Folge. Es muss daher ein Ausgleich zwischen Vorhersagegenauigkeit und Komplexität des Modells gefunden werden [AAM20].

Dieser Zusammenhang gilt prinzipiell für alle Surrogatmodelle [AAM20]. Mit der Anzahl der Modellparameter steigt die Fähigkeit einer ML-Methode, sich an die Daten anzupassen. Es ist dabei zu erwarten, dass das resultierende Modell nur eine fehlerbehaftete Annäherung an die tatsächlichen Werte liefern kann. Im Allgemeinen ist davon auszugehen, dass die Zielgröße gewissen Schwankungen unterliegt. Die entsprechenden Werte in den Trainingsdaten sind demnach nur eine Realisierung einer Verteilung. Diesen Fehler kann man auch mit mehr Trainingsdaten oder komplexeren Methoden nicht vermeiden. Die Hypothese verhält sich entsprechend dieser Annahme wie folgt zur Ausgangsgröße:

$$(2.7) \quad Y = f(X) + \epsilon$$

Das  $\epsilon$  wird als Fehler- oder Störgröße bezeichnet [FHK+16]. Darüber hinaus sind bei ML-Modellen Verzerrung und Varianz zu erwarten. Die Verzerrung (englisch: bias) beschreibt den Unterschied zwischen der bestmöglichen Annäherung der verwendeten Methode und des tatsächlichen Zielwerts. Unter Varianz versteht man die Abweichungen bei wiederholter Anwendung eines Verfahrens aufgrund geringer Schwankungen in den Daten [Pla21]. Bei hoher Varianz kann es passieren, dass das Modell Eigenheiten des spezifischen Datensatzes erlernt, die nicht für die Grundgesamtheit der Daten gelten. Dabei ist das Modell perfekt an die Trainingsdaten angepasst, kann aber womöglich

nicht generalisieren. Die Vorhersagequalität gemessen auf neuen, bislang ungesehenen Daten ist dann deutlich schlechter als für die Trainingsdaten. In einem solchen Fall spricht man von Überanpassung. Um dem entgegenzuwirken, kann ein Term eingefügt werden, der eine starke Anpassung der Modellparameter an die Daten sanktioniert. Dabei wird zugleich die Verzerrung erhöht und die Varianz verringert. Dieses Vorgehen nennt man Regularisierung. Für die lineare Regression und die Methode der kleinsten Quadrate kann eine Norm der Gewichte zur Regularisierung eingesetzt werden. Bei Verwendung der euklidischen Norm ergibt sich so die Ridge Regression [HK70; Mur22], die durch folgende Gleichung charakterisiert wird:

$$(2.8) \quad RSS = \sum_{i=1}^N (\mathbf{w}^T \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2$$

Bei der Anwendung von Regularisierung muss beachtet werden, dass die Verzerrung nicht zu groß wird. Ansonsten können die zu erlernenden Zusammenhänge in den Daten nicht mehr abgebildet werden. Man spricht auch von einem Kompromiss zwischen Verzerrung und Varianz [Mur22]. Dieser Kompromiss wird bei der Ridge Regression durch Anpassung des Hyperparameters  $\lambda$  erreicht.

Eine andere Möglichkeit als die Transformation von Eingangsgrößen, die Komplexität von Modellen zu erhöhen, stellen Ensembles dar. Dabei werden mehrere Modelle zusammengefasst und die endgültige Vorhersage aus den Vorhersagen der einzelnen Modelle gebildet, zum Beispiel durch Mittelung [WRG+23]. Im optimalen Fall gleichen sich die Vorhersagefehler der zugrundeliegenden Modelle gegenseitig aus [FSQ+17; VRM23]. Diese Modelle sollten dazu divers sein, also nicht identische Fehler auf den gleichen Dateninstanzen machen [PA05]. Ensembles können die Varianz senken und die Verzerrung erhöhen [SF12]. Sie lassen sich in vielen Bereichen einsetzen und ermöglichen eine höhere Vorhersagequalität als einzelne Modelle [BLFS16; FKE+15; Mur22].

### Kernel Methoden

Kernel Methoden bauen auf ML-Verfahren auf, die sich dadurch auszeichnen, dass sie nur Skalarprodukte der Eingangsdaten verwenden. Beispielsweise kann Gleichung (2.4) so umgeformt werden, dass nur noch Skalarprodukte von  $\mathbf{x}$  berechnet werden [Mur22]:

$$(2.9) \quad f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_{i=1}^N \alpha_i \mathbf{x}_i^T \mathbf{x}$$

Die Berechnung der  $\alpha_i$  wird hier nicht dargestellt und kann der Literatur entnommen werden [Mur22]. Anstelle von Skalarprodukten können Kernelfunktionen eingesetzt werden. Nach diesem Austausch, den man Kernel Trick nennt, sind Kernel Methoden benannt [Cho17]. Diese Methoden sind vielseitig einsetzbar und erreichen präzise Approximationen [STV04]. Ein Beispiel für eine Kernel Methode ist die Kernel Ridge Regression (KRR) [Mur22; SGV98], die aus der Kombination von Ridge Regression mit dem Kernel Trick entsteht. Die Anwendung des Kernel Tricks auf Gleichung (2.9) ergibt:

$$(2.10) \quad f(\mathbf{x}) = \sum_{i=1}^N \alpha_i k(x_i, x)$$

Eine Kernelfunktion ist identisch mit einem Skalarprodukt in einem anderen mathematischen Raum und ermöglicht so die implizite Transformation in potenziell unendliche Räume, ohne die Repräsentation der Datenpunkte in diesen Räumen berechnen können zu müssen. An einen Kernel  $k : X \times X \rightarrow \mathbb{R}$  werden dabei lediglich die Anforderungen der Symmetrie

$$(2.11) \quad k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$$

und der positiven Definitheit

$$(2.12) \quad \sum_{i=1}^N \sum_{j=1}^N c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \quad \forall n > 0 \quad \forall c_i, c_j \in \mathbb{R} \quad \forall \mathbf{x}_i, \dots, \mathbf{x}_N \in X$$

gestellt [STV04]. Es wird häufig eine radiale Basisfunktion (RBF) als Kernel verwendet [BLFS16; Mur22; Ras06], auch im Anwendungsbereich der Umformtechnik [XZZ21b]. RBFs sind reelle, radialsymmetrische Funktionen, die als Basis einer Approximation genutzt werden können. Radialsymmetrie besteht dann, wenn der Wert einer Funktion nur von der Entfernung vom Ursprung abhängt.

$$(2.13) \quad \varphi(x) \text{ ist eine radiale Basisfunktion} \Rightarrow \varphi(x) = \varphi(\|x\|)$$

Als Norm wird hier die euklidische Distanz verwendet. Ein Beispiel dafür ist der RBF-Kernel, auch Gaußkernel oder auf englisch squared exponential kernel genannt:

$$(2.14) \quad k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2}{2l^2}\right)$$

Die in Gleichung (2.14) definierte Funktion ist eine radiale Basisfunktion unter vielen. Jedoch handelt es sich um den für metrische Größen am häufigsten eingesetzten Kernel [Mur22], weshalb sich die Bezeichnung RBF-Kernel durchgesetzt hat [BLFS16; Buh03]. Dieser Kernel gibt die Ähnlichkeit zweier Vektoren basierend auf der skalierten euklidischen Distanz an. Der implizierte Merkmalsraum ist hier unendlich-dimensional. Dabei beschreibt  $l$  die sogenannte Bandbreite. Diese Größe beeinflusst die Distanz, über die zwei Datenpunkte einander beeinflussen. Der Faktor  $\sigma_f$  dient der Skalierung.

Eine andere Interpretation von Kernelfunktionen verwendet man im Rahmen der GP-Regression. Diese Technik basiert auf dem Satz von Bayes aus der Wahrscheinlichkeitstheorie. Ein GP kann als Verteilung über Funktionen verstanden werden. Dabei wird zunächst eine a-priori Wahrscheinlichkeitsverteilung über alle Funktionen definiert [Ras06]. Diese stützt sich auf die Annahme, dass die gesuchte Funktion gewisse, allgemeine Eigenschaften, wie zum Beispiel Glattheit, besitzt. Die a-priori Verteilung wird durch die Wahl einer Mittelwertfunktion, die häufig auf  $\mathbf{0}$  gesetzt wird, und eines Kernels, in diesem Kontext auch Kovarianzfunktion genannt, festgelegt:

$$(2.15) \quad f \sim \mathcal{N}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$$

In Kombination mit Trainingsdaten lässt sich über den Satz von Bayes die a-posteriori Verteilung berechnen. Wie sich herausstellt, entspricht der Mittelwert dieser Verteilung der Hypothese, die RSS minimiert. Das Training von GPs besteht aus der Wahl eines Kernels und seiner Parameter [Ras06].

$y_1$	$f(\mathbf{x}_1)$	$y_2$	$f(\mathbf{x}_2)$
4,93	5	4 930	5 000
6,24	7	6 240	7 000
10,44	9	10 440	9 000
11,41	11	11 410	11 000
12,47	13	12 470	13 000

**Tabelle 2.1:** Beispiel für zwei Ausgangsgrößen und zugehörige Vorhersagen.

Wird die Mittelwertfunktion auf  $\mathbf{0}$  gesetzt, so entspricht GP-Regression einer alternativen Formulierung von KRR. Ein Vorteil von GPs ist jedoch, dass die Hyperparameter, also die Parameter der Kernelfunktion, analytisch berechnet werden können. Bei KRR dagegen muss auf eine Rastersuche mit entsprechend höherer Komplexität zurückgegriffen werden [Mur22; Ras06]. Darüber hinaus dient die im Kernel kodierte Anforderung, dass eine Lösung glatt sein muss, als Mittel gegen Überanpassung [STV04]. Zusätzlich kann mit einem GP ein Konfidenzintervall für Vorhersagen ausgegeben werden.

### 2.2.5 Modelltest

Zur Beurteilung der Vorhersagequalität eines ML-Modells können verschiedene Metriken genutzt werden. Sie werden für den Modelltest auf einem Testdatensatz berechnet. Diese Testdaten bestehen wie die Trainingsdaten aus Eingangs- und Ausgangsgrößen, werden aber nicht im Training verwendet. Eine Metrik zur Beurteilung der Vorhersagequalität ist die Wurzel der mittleren quadratischen Abweichung, auf Englisch Root Mean Squared Error (*RMSE*) genannt, die wie folgt definiert ist:

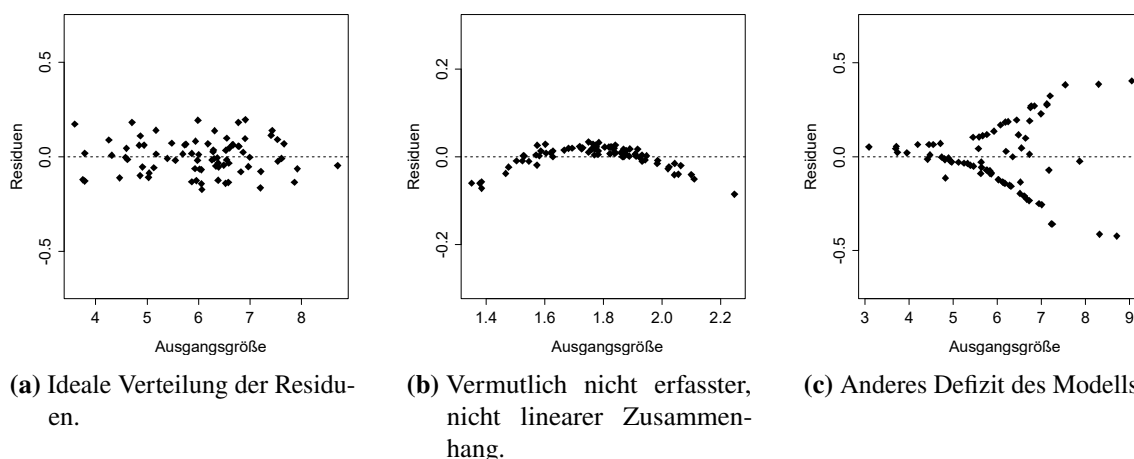
$$(2.16) \quad RMSE = \sqrt{\frac{\sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2}{N}}$$

Diese Metrik ist abhängig vom Quadrat der Differenz zwischen Vorhersage und tatsächlichem Wert. Somit haben große Differenzen größeren Einfluss auf sie als kleine. Anders verhält es sich für die mittlere absolute Abweichung, auf Englisch Mean Absolute Error (*MAE*) genannt. Sie beschreibt den Mittelwert der absoluten Differenz zwischen tatsächlichen und vorhergesagten Werten [RPA+23]. Diese Größe ist wie folgt definiert [BCNA21; DB21]:

$$(2.17) \quad MAE = \frac{\sum_{i=1}^N |f(\mathbf{x}_i) - y_i|}{N}$$

*MAE* vermittelt einen Eindruck des Vorhersagefehlers im Wertebereich der Ausgangsgröße. Damit ist sie aber auch abhängig von der Skalierung der Größe und Werte für unterschiedliche Ausgangsgrößen oder Problemstellungen lassen sich im Allgemeinen nicht vergleichen. Das trifft auch auf *RMSE* zu.

In Tabelle 2.1 sind zwei Ausgangsgrößen  $y_1$  und  $y_2 = 1\,000 \cdot y_1$  angegeben. Diese enthalten jeweils dieselbe relative Störgröße. Zu jeder Ausgangsgröße ist auch eine Approximation angegeben. Diese Vorhersagen  $f(\mathbf{x}_1)$  und  $f(\mathbf{x}_2)$  entsprechen exakt den Ausgangsgrößen ohne Störgrößen. Sie sind damit die bestmögliche Approximation der Ausgangsgrößen ohne Überanpassung. Für  $y_1$  und  $f(\mathbf{x}_1)$



**Abbildung 2.4:** Drei Beispiele für Residualplots [FHK+16].

beträgt  $RMSE$  0,79 und  $MAE$  0,64. Beide Metriken sind für  $y_2$  und  $f(\mathbf{x}_2)$  um den Faktor 1 000 höher. Dabei kann der Eindruck entstehen, dass die Vorhersagen für  $y_2$  schlechter sind als für  $y_1$ , obwohl sich Ausgangsgrößen und Vorhersagen lediglich in der Skalierung unterscheiden.

Aus diesem Grund werden die beiden Metriken  $RMSE$  und  $MAE$  nicht zur Evaluation im Rahmen dieser Arbeit eingesetzt. Stattdessen kommt die  $R^2$  Metrik zum Einsatz. Das Bestimmtheitsmaß  $R^2$ , das auch als Determinationskoeffizient bezeichnet wird, wird wie folgt berechnet:

$$(2.18) \quad R^2 = 1 - \frac{\sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2}{\sum_{i=1}^N (\bar{y} - y_i)^2}$$

Diese Maßzahl beschreibt den Anteil der Abweichungen der Zielgröße  $y$  von ihrem Mittelwert  $\bar{y}$ , der durch das Modell erklärt wird. Bestenfalls ist das die gesamte Abweichung und der  $R^2$  Wert ist 1. Dagegen kann das Bestimmtheitsmaß beliebig schlecht sein und auch Werte kleiner 0 annehmen [FHK+16]. Der Wert 0 wird erreicht, wenn unabhängig von den Eingangsgrößen  $\mathbf{x}$  der Mittelwert  $\bar{y}$  ausgegeben wird. Ein Vorteil dieser Metrik ist, dass ihr Wert unabhängig von der Skalierung der untersuchten Ausgangsgröße ist. Für beide Ausgangsgrößen und jeweilige Vorhersagen in Tabelle 2.1 beträgt die  $R^2$  Metrik 0,93. Sie eignet sich daher zum Vergleich von Modellen über Ausgangsgrößen und Datensätze hinweg.

Auch Residualplots können verwendet werden, um die Vorhersagequalität eines Regressionsmodells zu beurteilen. Dabei handelt es sich um ein Streudiagramm, wobei die x-Achse den tatsächlichen Werten der Ausgangsgröße  $y$  entspricht und die y-Achse den entsprechenden Residuen  $f(\mathbf{x}) - y$ . In Abbildung 2.4 sind drei Beispiele für Residualplots gegeben. Kennzeichen eines guten Modells ist eine unsystematische Schwankung der Punkte um die horizontale Achse im Bereich  $y = 0$ , wie sie in Abbildung 2.4a zu sehen ist. Das bedeutet, dass die Fehler der Vorhersage unabhängig von dem Wert der Ausgangsgröße sind. Können darüber hinaus Muster in einem Residualplot erkannt werden, so ist eine Anpassung des Modells zu erwägen. Diese Muster können unterschiedliche Formen annehmen. Abbildung 2.4b zum Beispiel zeigt einen nicht linearen Zusammenhang, der von dem Modell nicht erfasst ist. Das kann womöglich durch eine Erhöhung der Komplexität des Modells kompensiert werden. Bei anderen Mustern wie beispielsweise dem in Abbildung 2.4c kann das Modell eventuell durch Transformationen der Ein- oder Ausgangsgrößen verbessert werden [FHK+16; Mur22].

### 2.3 Erklärbarkeit

Im Folgenden wird die Erklärbarkeit von datengetriebenen Surrogatmodellen thematisiert. Dabei wird auf die Motivation zur Verbesserung der Erklärbarkeit eingegangen und eine Kategorisierung von Verfahren vorgestellt. Zuletzt werden ausgewählte Methoden zur Steigerung der Erklärbarkeit umrissen.

#### 2.3.1 Motivation

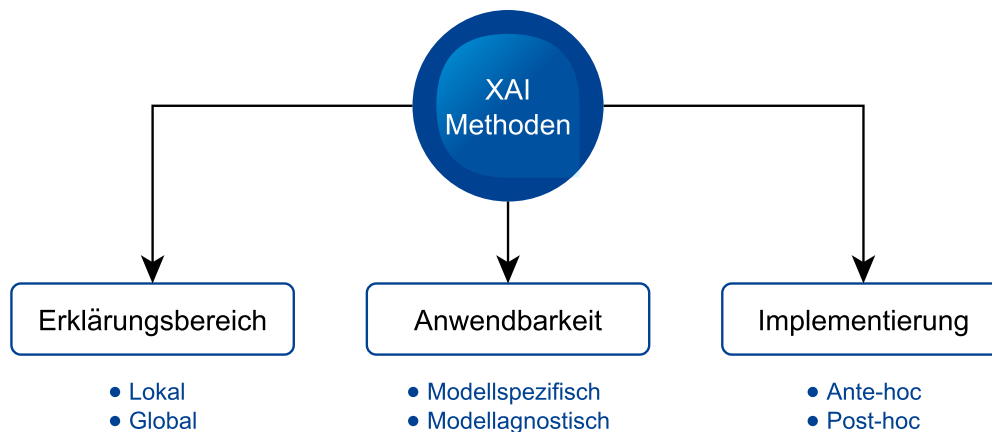
Modelle, die mit ML-Verfahren erstellt werden, sind häufig intransparent. Wie sie ihre Vorhersagen treffen ist dabei für den Anwender nicht oder nur sehr eingeschränkt nachvollziehbar [AB18; BH21; Rus19]. Ensembles von Modellen, die Vorhersagen auf den Ausgaben mehrerer einzelner Modelle aufbauen, sind besonders davon betroffen [Fre14]. Man spricht hier auch von black-box Modellen [AB18; BH21; TKE+16]. Zur Lösung komplexer Probleme sind jedoch häufig komplexe Modelle notwendig [BH21]. Einfachere Modelle dagegen sind einfacher zu erklären, erreichen aber je nach Anwendungsfall nicht die gleiche Vorhersagequalität. Dementsprechend wird häufig von einem Kompromiss zwischen Vorhersagequalität und Erklärbarkeit gesprochen [AB18; Fre04; Rus19]. Es besteht im Allgemeinen die Gefahr, dass die Vorhersagequalität von Modellen nachlässt, wenn die Erklärbarkeit einen zu hohen Stellenwert einnimmt [Lip18]. Gleichzeitig ist Erklärbarkeit von hoher Bedeutung für ML-Methoden [BH21; RSG16b].

- Erklärbarkeit hat großen Einfluss auf das **Vertrauen** der Nutzer in das Modell. [DBH18; DK17; Mil19; RSG16b]. Ist dieses Vertrauen nicht groß genug, so werden die Nutzer das Modell nicht verwenden oder zumindest ihre eigenen Entscheidungen nicht darauf stützen. Mit nachvollziehbaren Erklärungen kann das Vertrauen der Nutzer in das Modell gesteigert werden [AB18; BVV+18; RSG16a].
- Erklärbare Modelle haben das Potenzial, das **Verständnis der zugrundeliegenden Zusammenhänge** zu stärken. Datengetriebene Surrogatmodelle können Beziehungen zwischen Eingangs- und Ausgangsgrößen erkennen, die von bekannten Regeln abweichen. Diese gelernten Zusammenhänge zu analysieren kann daher neue Hypothesen begründen [BH21; DBH18; Fre14].
- Ist erkennbar, anhand welcher Kriterien ein Modell seine Entscheidungen trifft, können diese von Fachexperten auf ihre Plausibilität untersucht werden. So kann Erklärbarkeit zur Überprüfung oder **Anpassung eines Modells** herangezogen werden und die Vorhersagequalität erhöhen [AB18; BH21; LL17].

Teilweise ist die Erklärbarkeit von Modellen auch rechtlich vorgeschrieben, wie beispielsweise von der Datenschutz-Grundverordnung (DSGVO) der Europäischen Union. [BH21; RSK20]

#### 2.3.2 Begriffe und Kategorien

Einige Autoren kritisieren das Fehlen einer eindeutigen Definition der Konzepte Erklärbarkeit und Interpretierbarkeit, verwenden diese Begriffe dann aber selbst, ohne klarzustellen, was sie genau damit meinen [DBH18; Lip18]. Allgemein herrschen verschiedene Interpretationen dieser Begriffe vor, und auch über die Konzepte, die zur Herleitung der Erklärbarkeit verwendet werden, sind Autoren



**Abbildung 2.5:** Einteilung von XAI-Methoden anhand der drei Dimensionen Erklärungsreichweite, Anwendbarkeit und Implementierung.

geteilter Meinung [CK22; DK17]. Die Relevanz von Erklärbarkeit auf Vertrauen zurückzuführen, führt zum Beispiel zu den Problemen, dass eine einheitliche Definition sowie Methoden zur Operationalisierung von Vertrauen fehlen [DBH18; Lip18]. Eine mögliche Beschreibung von Erklärbarkeit lautet, dass mit ihrer Steigerung bestenfalls eine für Menschen annähernd verständliche, vereinfachte Darstellung des Entscheidungsprozesses von ML-Modellen gelingen kann [BH21].

Der Forschungsbereich, der sich mit der Erklärbarkeit von ML-Modellen beschäftigt, wird XAI genannt. Dabei gibt es Überschneidungen mit den Themen und Methoden der Psychologie, der Künstlichen Intelligenz und der Mensch-Computer Interaktion [Mil19]. Das Interesse der Forschungsgemeinschaft, aber auch der Industrie, an XAI ist in den letzten Jahren gestiegen [AB18; Mil19]. Sie verfolgen das Ziel, Methoden zu entwickeln, die besser erklärbare Modelle hervorbringen [AB18]. Das kann sich auf den gesamten Entscheidungsprozess der Modelle [BH21] beziehen, oder auf die Begründung einzelner Vorhersagen [KM22]. Anhand dieser und anderer Merkmale können die Methoden der XAI in Klassen unterteilt werden, wie in Abbildung 2.5 dargestellt.

Eine grundlegende Unterscheidung bietet der Bereich, bzw. der Umfang der Erklärbarkeit. Kann eine XAI-Methode das gesamte Modell, also die zugrundeliegende Logik, erklären, spricht man von globaler Erklärbarkeit. Im Gegensatz dazu erlauben lokale Methoden lediglich Aussagen über einzelne Vorhersagen. Ist ein XAI-Verfahren nur für eine Teilmenge von Modellen anwendbar, so handelt es sich um ein modellspezifisches Verfahren. Vom Modell unabhängige Herangehensweisen werden modellagnostisch genannt. Bei der Implementierung unterscheidet man, ob die Erklärbarkeit in das Lernverfahren direkt integriert ist (ante-hoc), oder erst nach dem Training in einem zusätzlichen Schritt erreicht wird (post-hoc) [KM22].

Die Wahl einer adäquaten XAI-Methode ist nicht trivial [LL17]. Eine einzelne, beste Herangehensweise gibt es nicht [BVV+18]. Je nach Anforderungen an die erzeugten Erklärungen und an die zu verwendenden Lernverfahren existiert womöglich gar keine geeignete Methode [Mil19]. Beispielsweise konzentriert sich die Forschung hauptsächlich auf Klassifikationsaufgaben und die Ergebnisse lassen sich nur eingeschränkt auf die Regression übertragen [DBH18].

### 2.3.3 Methoden

Ein häufiges Ziel bei der Anwendung von XAI ist es, die Bedeutung von Eingangsgrößen auf die Approximation der Ausgangsgröße zu ermitteln [AB18; BH21; GMR+18]. Ausgewählte Methoden, die zu diesem Zweck eingesetzt werden können, werden nachfolgend vorgestellt. Dabei liegt der Fokus auf einfach anzuwendenden Verfahren, die für Regressionsmodelle geeignet sind.

Ein simpler Ansatz, der als global, modellspezifisch und ante-hoc eingestuft werden kann, ist, die Komplexität von Modellen zu verringern [RSK20]. Sind Modelle einfach genug, dann können Menschen nachvollziehen, wie sie ihre Entscheidungen treffen. Häufig werden Entscheidungsbäume oder die lineare Regression als Beispiele für einfache Modelle genannt, wobei einige Autoren bestreiten, dass diese Erklärbarkeit prinzipiell inhärent ist [BH21; Lip18]. Bei der Einschränkung auf solche Herangehensweisen ist die erhebliche Limitierung der Vorhersagequalität bei komplexeren Zusammenhängen zu beachten [RSG16b]. Wenn die Erklärbarkeit von höherer Relevanz ist als die Vorhersagequalität, dann sind ante-hoc Methoden zu bevorzugen [RSG16b]. Modellagnostische und post-hoc Methoden limitieren die Vorhersagequalität nicht. Sie ermöglichen es, sowohl das ML- als auch das spezifische XAI-Verfahren flexibel zu wechseln, da die beiden die Wahl des jeweils anderen nicht einschränken. Somit kann theoretisch die bestmögliche Approximation mit einem komplexen ML-Modell erreicht werden. Gleichzeitig wird das Problem umgangen, dieses potenziell komplizierte Modell erklären zu müssen. Außerdem erlauben solche XAI Methoden, die Erklärungen für zwei unterschiedliche Modelle zu vergleichen. Jedoch sind sie mitunter nicht für alle Datendomänen und ML-Modelle anwendbar.

Ein Beispiel für einen globalen, modellagnostischen, post-hoc Ansatz sind Partial Dependence Plots (PDPs) [BH21]. Sie stellen die Beziehungen zwischen einer Eingangsgröße und den Vorhersagen eines Modells grafisch dar und zeigen die Veränderung der durchschnittlichen Vorhersage auf, wenn die untersuchte Eingangsgröße über ihren Wertebereich variiert wird [AB18]. PDPs werden in verschiedenen Anwendungsgebieten erfolgreich eingesetzt, um die Erklärbarkeit von Modellen zu erhöhen [GKBP15].

Eine andere, häufig angewendete Methode sind Shapley additive explanations (SHAP). Auch bei diesem Ansatz wird die Auswahl von ML-Modellen nicht eingeschränkt. Er ist als lokal, modellagnostisch und post-hoc einzuordnen. Dabei liegt folgende Vorgehensweise zugrunde: Nachdem ein komplexes Modell gebildet wurde, wird in der Umgebung einzelner Datenpunkte ein simpleres Modell trainiert, das einfacher zu erklären ist. Häufig ist das einfachere, erklärbare Modell linearer Natur. Dann kann anhand der Gewichte der einzelnen Eingangsgrößen der Einfluss dieser auf die Ausgabe abgelesen werden [Rus19]. Es gibt eine Vielzahl leicht unterschiedlicher Implementierungen dieser Herangehensweise. SHAP vereint jedoch mehrere andere Methoden und ist dabei theoretisch optimal unter diesen Verfahren. Mit SHAP werden gute Ergebnisse bei Experimenten erreicht, bei denen Menschen die gelieferten Erklärungen bewerten [LL17]. Die Ausgaben dieser Methode können auch leicht visualisiert werden, was für die Darstellung von Erklärungen zu bevorzugen ist [Mil19]. Aus der Kombination mehrerer lokaler Erklärungen können auch globale Erklärungen erzeugt werden [RPA+23].

Die Evaluation von XAI-Methoden stellt im Allgemeinen eine Herausforderung dar. Dabei ist nicht nur die Korrektheit einer Erklärung zu beachten, sondern auch ihre Relevanz, Verständlichkeit und weitere Eigenschaften [Mil19]. Eine Möglichkeit, die Erklärbarkeit zu überprüfen, ist es, die



Ergebnisse von Menschen beurteilen zu lassen [BH21; LL17]. Weitere, auch weniger aufwendige Methoden zu entwickeln, ist Gegenstand aktueller Forschung. Häufig wird daher auf eine umfassende Evaluation der erzeugten Erklärungen verzichtet [Lip18; WMR17].



## 3 Verwandte Arbeiten

In diesem Kapitel werden verwandte Arbeiten vorgestellt, die Surrogatmodelle für Problemstellungen der Fertigung verwenden. Von diesen gibt es nur wenige [ACS22]. Ihre Anzahl reduziert sich weiter, begrenzt man die Suche auf Anwendungen in der Umformtechnik.

### 3.1 Surrogatmodelle in der Fertigung

Eine der ersten Anwendungen von Surrogatmodellen in der Fertigung wird von Irani et al. [ICFQ93] beschrieben. Die Autoren setzen Entscheidungsbäume zur Prozessoptimierung in verschiedenen Bereichen der Halbleiterherstellung ein, in denen das Verständnis der zugrundeliegenden Prozesse gering ist. Zu den von ihnen verwendeten Eingangsgrößen zählen Maschinenspezifikationen und Materialeigenschaften. Als besondere Herausforderungen bezeichnen sie die geringe Verfügbarkeit von Daten hoher Qualität. Die von den Entscheidungsbäumen abgeleiteten Entscheidungsregeln finden Zuspruch bei den involvierten Fachexperten. Die Autoren verwenden nicht Simulationsdaten, sondern solche aus Experimenten. Das trifft ebenso auf die Arbeit von Gellrich et al. [GFW+19] zu. Sie verwenden ML-Verfahren, um die Prozessparameter bei der Herstellung von Stahlstrukturen, bei dem auch Umformen eingesetzt werden kann, zu optimieren. Dabei stellen sie fest, dass die Qualität der Vorhersagen stark von der Güte der Daten abhängt. Ihr Vorgehen orientiert sich an CRISP-DM und die Autoren bezeichnen diese Herangehensweise als nützlich für alle Arten von Fertigungsprozessen. Zunächst ermitteln sie Korrelationen zwischen Eingangsgrößen und verwenden anschließend lediglich die unabhängigen. Auch teilen sie ihre 785 650 Datenpunkte einer Zeitreihe in Trainings- und Testdaten im Verhältnis 80 : 20 auf. Anhand des Trainingsdatensatzes erzeugen sie mit unterschiedlichen Algorithmen Entscheidungsbäume zur Klassifikation von Maschinenzuständen.

Dass Surrogatmodelle im Bereich der Fertigung auch auf Simulationsdaten trainiert werden können, zeigen Viisainen et al. [VYC+23]. Sie verwenden diese zur Approximation eines Qualitätskriteriums von Geweben. Im Vergleich zur Simulation können Surrogatmodelle die Zeit für die Erzeugung einer Vorhersage um den Faktor 6 000 senken. Die Autoren verwenden dazu ein Neuronales Netz, das auf 90 % von 1 802 Datenpunkten trainiert wurde. Zwar erreicht dieses Modell nur einen  $R^2$  Wert von 0,53 auf dem Testdatensatz, doch ist dies ausreichend, um eine Vorauswahl an Prozessparametern zu treffen. Diese können anschließend simuliert werden, um präzisere Werte zu erhalten. Auch Karadogan et al. [KCL21] trainieren Neuronale Netze auf Simulationsdaten. Sie approximieren eine Eigenschaft von Blechen, die für die Simulation eines Blechumformprozesses relevant ist. Als Eingangsgrößen stehen dabei 1 080 bereits bekannte Materialspezifikationen zur Verfügung. Die 1 029 Datenpunkte werden in 70 % Trainings-, 15 % Validierungs- und 15 % Testdaten aufgeteilt, bevor ein Neuronales Netz trainiert wird. Die Autoren erreichen damit  $R^2$  Werte bis zu 0,71.

Auch Dey et al. [DMA17] beschäftigen sich mit der Approximation von Simulationsergebnissen. Ihnen geht es um die Vorhersage der Eigenschaften von Verbundwerkstoffen. Sie geben an, dass sich aus der Literatur keine Wahl eines besten ML-Verfahrens für die Erzeugung von datengetriebenen Surrogatmodellen ableiten lässt. Stattdessen müssen mehrere Ansätze angewendet und verglichen werden. Für eine Vorauswahl von Verfahren sind die Komplexität des Modells, die Datenqualität und die Anzahl an Eingangsgrößen zu beachten. Auch die Popularität einer Methode wird von den Autoren berücksichtigt. Unter den erprobten Methoden erreicht polynomielle Regression die beste Vorhersagequalität für eine reduzierte Anzahl von Eingangsgrößen. Auch ein auf GP basierendes Verfahren kann die Ausgangsgrößen approximieren. Gilt es eine Vielzahl an Eingangsgrößen zu beachten, so ermöglichen Neuronale Netze präzisere Vorhersagen. Allerdings müssen die Autoren die Anzahl der Datenpunkte für diese Methode im Vergleich zur polynomiellen Regression um den Faktor 20 auf über 2 000 erhöhen. Zu ähnlichen Erkenntnissen gelangen auch Pfommer et al. [PZL+18], die ebenfalls Verbundstoffe untersuchen. Mithilfe von ML möchten die Autoren deren Herstellung optimieren. Auch sie wählen Neuronale Netze aufgrund ihrer häufigen Verwendung im Bereich der Fertigung und trainieren sie auf Simulationsdaten. Zu den 50 verwendeten Eingangsgrößen zählen die Form des Werkstücks und die des Werkzeugs. In acht Wochen haben die Autoren lediglich 584 Datenpunkte simulieren können. Weder mit linearer oder polynomieller Regression noch mit Neuronalen Netzen können sie die gesuchte Ausgangsgröße präzise vorhersagen und begründen das mit der geringen Anzahl an Daten. Auf einem deutlich größeren, 24 000 Datenpunkte umfassenden Datensatz trainieren sie ein Neuronales Netz. Die damit identifizierten Prozessparameter übertreffen zuvor manuell gefundene bezüglich der untersuchten Ausgangsgröße.

Denkena et al. [DDL+20] beschäftigen sich mit der Vorhersage von Formabweichungen in einem Fräsprozess. Dabei verwenden sie Daten von anderen Autoren, was ihnen einen Vergleich mit deren Methodik ermöglicht. Darüber hinaus simulieren sie den Prozess selbst und erzeugen so neue Daten. Die Autoren stellen fest, dass die Modellauswahl großen Einfluss auf die Vorhersagequalität hat, aber ein zeitaufwendiger, fehleranfälliger und auf Anwendererfahrung beruhender Vorgang ist. Überdies geben sie an, dass die Modellauswahl in der Literatur selten beschrieben wird. Sie wenden daher ein AutoML Verfahren an, das die Modellauswahl selbstständig anhand der Trainingsdaten trifft. Dabei bedienen sie sich des Python-Moduls `auto-sklearn`. Wird es zur Optimierung der Hyperparameter eines Support-Vektor-Regressionsmodells verwendet, kann ein RMSE Wert von 0,001 mm erreicht werden. Überlassen die Autoren `auto-sklearn` die Wahl des Hypothesenraums komplett, inklusive Datenvorverarbeitungsschritten, kann sich dieser Wert noch um den Faktor 100 verringern. Mit `auto-sklearn` trainieren sie auch Ensembles, diese erreichen jedoch keine bessere Vorhersagequalität als ein einzelnes Modell. Die Autoren geben an, dass ein AutoML-Ansatz bereits bei einer Laufzeit von unter zwei Minuten die Vorhersagequalität manuell erzeugter Modelle erreichen und übertreffen kann. Sie empfehlen prinzipiell, AutoML als Ausgangsgrundlage für DM-Projekte einzusetzen.

## 3.2 Surrogatmodelle in der Umformtechnik

Während die im vorherigen Abschnitt behandelten Arbeiten Anwendungen von Surrogatmodellen in anderen Fertigungsverfahren beschreiben, steht im Folgenden spezifisch die Umformtechnik im Fokus. So wie bei Bustillo et al. [BLFS16], die sich mit der Herstellung von Stahlgewinden durch Umformen beschäftigen. Ihr Ziel ist es, den Werkzeugverschleiß vorherzusagen. Dazu stehen ihnen 435 Datenpunkte physischer Experimente zur Verfügung. Sie betrachten alle Eingangsgrößen als

metrisch, selbst wenn sie nicht derart in den Daten kodiert sind und trainieren unterschiedliche ML-Modelle. Dabei erzielen sie mit Ensembles von Entscheidungsbäumen die besten Ergebnisse. Auch mit der Kernel Methode Support-Vektor-Regression und dem RBF-Kernel erreichen die Autoren präzise Vorhersagen. Zur Evaluation ihrer Modelle verwenden sie *RMSE* und eine Kreuzvalidierung mit zehn Teilmengen. Auch Dai et al. [DRF+22] wenden Kreuzvalidierung an und bezeichnen sie als eine dateneffiziente Möglichkeit, die Vorhersagequalität eines Modells auf neuen, ungesehenen Daten anzunähern. Die Autoren untersuchen das Biegen von durch Kaltumformung erzeugten Stahlträgern und verwenden Kreuzvalidierung zur Optimierung der Modellparameter. Für die Evaluation wird ein Testdatensatz aus 30 % der 1 620 Datenpunkte gebildet. Die Autoren erreichen ihre besten Ergebnisse,  $R^2$  Werte bis zu 0,99, mit einem Ensemble aus Entscheidungsbäumen und sind in der Lage, aus diesen bisher unbekannte Gleichungen für physikalische Zusammenhänge in den Umformprozessen abzuleiten. Letztere können die zugrundeliegenden Gesetzmäßigkeiten besser abbilden als zuvor bekannte. Die größten Hürden für den Einsatz von ML im Bereich der Fertigungstechnik sind die Transformation von Eingangsgrößen in metrische Werte und die Evaluation von Modellen, so die Autoren.

Auch Xu et al. [XZZ21a] untersuchen die Belastbarkeit von Stahlteilen. Sie trainieren Ensembles von Entscheidungsbäumen auf 262 Datenpunkten mit 13 Eingangsgrößen. Die anzunähernde Ausgangsgröße ist ein Wert, der die Tragfähigkeit von Stahlträgern beschreibt. Außerdem verwenden sie eine Kreuzvalidierung mit zehn Teilmengen und die Metrik *RMSE* zur Evaluation ihrer Modelle. Mithilfe ihres besten Modells ermitteln sie Prozessparameter, die bessere Erzeugnisse ermöglichen, als die bisher verwendeten Annäherungen. Sie merken an, dass die Präzision der Vorhersage stark von der Anzahl zur Verfügung stehender Daten in den jeweiligen Wertebereichen abhängt. In einer anderen Arbeit vergleichen dieselben Autoren verschiedene ML-Ansätze zur Vorhersage der Belastbarkeit von Stahlteilen [XZZ21b]. Sie verwenden dabei 20 bis 30 % ihrer Daten als Testdatensatz und greifen erneut auf Kreuzvalidierung mit zehn Teilmengen zurück. Als Metrik zur Evaluation kommt wieder *RMSE* zum Einsatz. Ensembles von Entscheidungsbäumen und Kernel Methoden schneiden am besten ab und übertreffen selbst etablierte Gleichungen. Erneut wird auf die Bedeutung der Datenqualität für präzise Vorhersagen hingewiesen.

Auch anhand von Simulationsdaten kann die Belastbarkeit von mit Umformen produzierten Stahlteilen vorhergesagt werden, wie Fang et al. [FRC+21] zeigen. Bevor sie ein Neuronales Netz trainieren, normalisieren sie die Eingangs- und Ausgangsgrößen. Aufgrund ihrer vergleichsweise hohen Anzahl an Daten ist es ihnen zudem möglich, neben 26 % Testdaten auch 24 % Validierungsdaten der 50 000 simulierten Datenpunkte beiseite zu legen. Zur Optimierung von Hyperparametern verwenden sie eine Rastersuche und die  $R^2$  Metrik. Hema et al. [HCN22] wenden ebenfalls Neuronale Netze zur Untersuchung der Belastbarkeit von Stahlteilen an. Sie verwenden eine Mischung aus 12 Experiment- und 48 Simulationsdatenpunkten und teilen diese in 70 % Trainings- und 30 % Testdaten auf. Manuell testen sie verschiedene Modellparameter und überprüfen die Vorhersagequalität mit den Metriken *RMSE* und  $R^2$ . Ein lineares Modell übertrifft dabei Neuronale Netze wie auch Ensembles aus Entscheidungsbäumen.

Rajneesh et al. [RPA+23] beschäftigen sich ebenfalls mit der Belastbarkeit von Stahlteilen. Sie beschreiben im Gegensatz zu den meisten anderen Arbeiten ihre Datenvorverarbeitung, wie auch die Modellauswahl in größerem Ausmaß. Im Rahmen der Implementierung verwenden sie das Python-Modul *scikit-learn* [PVG+11]. Zur Grundlage des Trainings dienen 70 % der 18 720 simulierten Datenpunkte mit elf Eingangs- und einer Ausgangsgröße. Validierungs- und Testdatensatz bestehen jeweils aus 15 % der Gesamtdaten. Die Autoren untersuchen zunächst

Korrelationen unter den Eingangsgrößen und wenden eine Skalierung auf das Intervall 0 bis 1 an. Sie vergleichen verschiedene ML-Verfahren, darunter die einfache lineare Regression, Kernel Methoden und Ensembles von Entscheidungsbäumen. Zur Optimierung von Hyperparametern kommt eine Rastersuche zum Einsatz. Bis auf die einfache lineare Regression und eines der Ensembles erreichen alle Modelle einen  $R^2$  Wert über 0,90.

Das Rückwärts-Fließpressen als spezifische Kaltmassivumformtechnik untersuchen Petkar et al. [PGK+23]. Sie geben an, dass das Werkstück in diesen Umformprozessen enormen Kräften ausgesetzt ist und daher oft beschädigt wird. Auch die Eigenschaften des Endprodukts sind nur schwer vorherzusagen, hängen aber maßgeblich von der Reibung, der Geometrie des Werkzeugs und des Werkstücks sowie des verwendeten Materials ab. Die Autoren streben an, die Ausgaben einer Simulation eines Umformprozesses des Rückwärts-Fließpressens durch datengetriebene Surrogatmodelle zu approximieren. Sie wählen dazu Neuronale Netze als ML-Ansatz, da diese die vorherrschenden, nicht linearen Zusammenhänge abbilden können. Die notwendigen Daten erzeugen sie mithilfe eines vollständigen Versuchsplans. Eine Simulation dauert dabei etwas mehr als zehn Minuten. Die Autoren beschränken sich auf vier Eingangsgrößen mit ebenso vielen Werten. Ihr Datensatz besteht damit aus 256 Datenpunkten, von denen 12 % als Testdaten zum Einsatz kommen. Zur Evaluation verwenden sie die Metrik  $MSE$ , also  $RMSE$  ohne die Wurzel. Außerdem vergleichen die Autoren ihre Vorhersagen mit den Ergebnissen physischer Experimente und stellen fest, dass die Abweichungen in einem akzeptablen Bereich liegen. Die Autoren stellen darüber hinaus die Vorhersagen einzelnen Eingangsgrößen grafisch gegenüber, um so Aussagen über die Beziehungen zwischen den Größen treffen zu können. Sie stellen fest, dass die Annahme der Nichtlinearität der Zusammenhänge und damit auch die Wahl ihres ML-Verfahrens gerechtfertigt ist.

### 3.3 Erklärbarkeit in der Fertigungstechnik

Häufig eingesetzte ML-Verfahren im Bereich der Fertigung sind nicht zu durchschauen, aber Einblicke in ihre Funktionsweise müssen möglich sein [HSSK05]. Auch einfachere Modelle wie Entscheidungsbäume können zu groß und komplex sein, um von Menschen erfasst zu werden [TKE+16].

Es kommen im Bereich der Fertigungstechnik jedoch nur selten Verfahren zum Einsatz, welche die Erklärbarkeit erzeugter Modelle steigern. Unter den zuvor beschriebenen Arbeiten wurden lediglich in einer XAI-Methoden angewendet [RPA+23]. Mit SHAP fiel die Wahl dabei auf ein Verfahren, das die Bedeutung einzelner Eingangsgrößen für die Approximation der Ausgangsgröße bestimmt. Unter diesen ist SHAP populär und fußt auf einer stabilen theoretischen Grundlage [LL17]. Damit sind lokale und globale Erklärungen möglich [RPA+23].

### 3.4 Zentrale Erkenntnisse

Zusammenfassend lässt sich sagen, dass datengetriebene Surrogatmodelle im Bereich der Umformtechnik erfolgreich eingesetzt werden. Die Daten dafür können sowohl durch physische Experimente [BLFS16], als auch durch Simulationen erzeugt werden [FRC+21]. Folgende Punkte sind darüber hinaus von besonderem Interesse für die vorliegende Arbeit:

- **Eingangsgrößen:** Maschinenspezifikationen wie die Werkzeuggeometrie sowie Reibung und Materialparameter sind wichtige Größen zur Vorhersage relevanter Größen eines Fertigungsprozesses [ICFQ93; PGK+23].
- **Umwandlung kategorischer in metrische Werte:** Arbeiten Autoren mit kategorischen Größen, so versuchen sie, diese in metrische zu übertragen. Das vereinfacht die spätere Anwendung von ML-Methoden [BLFS16; DRF+22].
- **Validierung und Test:** Zur Optimierung von Modellen kann eine Rastersuche verwendet werden [FRC+21; RPA+23]. Dabei ist der Einsatz einer Kreuzvalidierung [BLFS16; DRF+22] sowie die Verwendung von 12 % bis 30 % der Daten als Testdatensatz verbreitet [PGK+23; XZZ21b]
- **CRISP-DM:** Eine Möglichkeit, das eigene Vorgehen zu strukturieren, ist es, Vorgehensmodelle wie CRISP-DM heranzuziehen [GFW+19].
- **AutoML:** Zwar ist die Anwendung von AutoML im Bereich der Umformtechnik nicht verbreitet, kann jedoch als Entscheidungsgrundlage herangezogen werden [DDL+20].
- **Voruntersuchung:** Einige Autoren behandeln die Datenqualität und bestätigen ihre Bedeutung für die präzise Vorhersage durch Surrogatmodelle [DMA17; ICFQ93]. Wie diese festzustellen oder zu verbessern ist, darauf gehen sie jedoch nicht ein. Andere Autoren untersuchen Korrelationen und verwenden sie zur Auswahl von Eingangsgrößen [GFW+19]. Allgemein sind aus einer Voruntersuchung abgeleitete Vorverarbeitungsschritte essentiell für den Erfolg von DM-Projekten, ein solches Vorgehen wird aber nur selten beschrieben [KBT11; RPA+23].
- **Wahl von ML-Algorithmen:** Die Wahl eines ML-Algorithmus sollte nicht allein auf einer Literaturrecherche basieren. Auch die Datencharakteristika müssen dabei berücksichtigt werden [DMA17]. Viele Autoren verwenden Neuronale Netze, auch aufgrund ihrer Popularität [DMA17; PZL+18]. Dabei werden sie mitunter auch auf kleinen Datensätzen mit etwa 1 000 Datenpunkten trainiert und erreichen durchwachsene Ergebnisse [HCN22; VYC+23].

Aufbauend auf der vorgestellten Literatur wird sich im Rahmen dieser Arbeit an obigen Erkenntnissen orientiert. Insbesondere wird hier, abweichend von den beschriebenen Arbeiten, die Voruntersuchung im Rahmen einer explorativen Datenanalyse vertieft betrachtet. Grund dafür ist die hohe Bedeutung dieses Vorgehens für den Erfolg einer Anwendung von ML-Verfahren und die seltene Beschreibung dieses Schritts in der Literatur [KBT11; RPA+23].





## 4 Ergebnisse der explorativen Datenanalyse

Die Entwicklung einer Methode zur datengetriebenen Approximation einer Kaltmassivumformsimulation bedarf einer gründlichen Prüfung der Anforderungen. Neben relevanter Literatur sind die Eigenschaften der zu verarbeitenden Daten eine wichtige Informationsquelle. Zu diesem Zweck wurde vom Institut für Umformtechnik der Universität Stuttgart ein Datensatz bereitgestellt. Der Fachexperte, der diesen erzeugt hat, stand auch für Rückfragen im Rahmen dieser Arbeit zur Verfügung. Die Charakteristiken der Daten werden nachfolgend analysiert.

Im Allgemeinen ist zu Beginn eines ML-Projekts eine explorative Datenanalyse durchzuführen [Mur22]. Dabei steht die Suche nach Strukturen und Besonderheiten in den Daten im Vordergrund [Alb22]. Es handelt sich um einen kreativen Prozess, in dem Ideen nachgegangen werden soll und der neue Fragestellungen, Hypothesen oder Erkenntnisse hervorbringen kann. Die Grenzen zum Feature Engineering und Data Preprocessing sind fließend [Han12]. Hier wird zunächst die Datenqualität untersucht und soweit möglich gesteigert. Dabei liegt besonderes Augenmerk auf der Behandlung von Ausreißern. Anschließend werden die Daten visualisiert und Korrelationen zwischen Eingangs- und Ausgangsgrößen untersucht. Auch erste ML-Modelle werden trainiert. Es kommen also deskriptive, wie Korrelationsanalysen, und induktive Verfahren zum Einsatz [Han12]. Die Ergebnisse dieser Untersuchungen dienen als Grundlage für die Ausgestaltung der Methode zur Erzeugung eines datengetriebenen Surrogatmodells.

Im Folgenden werden zunächst die Eckdaten des Datensatzes beschrieben und die Datenqualität untersucht. Anschließend werden Maßnahmen zur Vorverarbeitung der Daten diskutiert. Es folgt ein Abschnitt, in dem Visualisierungen der Daten vorgestellt und interpretiert werden. Abschließend werden erste Modelle mit einfachen Methoden, aber auch unter Zuhilfenahme von AutoML erzeugt.

### 4.1 Datenbeschreibung

Ausgangspunkt der Daten ist die Simulation einer Kaltmassivumformung. Es handelt sich dabei im Speziellen um einen einfachen Napf-Rückwärts-Fließpressen-Prozess, wie in Abschnitt 2.1.1 beschrieben. Eine Übersicht über Herkunft und Form der zur Verfügung gestellten Daten gibt Tabelle 4.1. Der Datensatz wurde im Frühjahr 2023 mit der Software DEFORM 2D<sup>1</sup> in der Version 12.1 am Institut für Umformtechnik der Universität Stuttgart erzeugt. Er wurde in Form einer Ordnerstruktur aus 54 Dateien des Formats CSV mit einer Gesamtgröße von 284 Kilobyte übergeben. Dabei sind manche Eingangsgrößen in den Datei- und Verzeichnisnamen kodiert. In jeder Datei befinden sich 20 Zeilen, wobei jede Zeile einem Datenpunkt entspricht. Ein solcher repräsentiert einen Simulationsdurchlauf des untersuchten Umformprozesses von etwa zehn Minuten Dauer.

---

<sup>1</sup><https://www.deform.com/products/deformd/>

#### 4 Ergebnisse der explorativen Datenanalyse

Eigenschaft	Wert
Erzeugungsdatum	Februar 2023
Erzeugungsort	Institut für Umformtechnik, Universität Stuttgart
Erzeugungsmethode	DEFORM 2D V12.1
Simulationsdauer pro Datenpunkt	etwa 10 Minuten
Anzahl Dateien	54
Dateiformat	CSV
Gesamtgröße	284 KB
Anzahl Datenpunkte pro Datei	20
Anzahl Datenpunkte gesamt	1 080
Anzahl Eingangsgrößen	7, davon 4 in Verzeichnisstruktur kodiert
Anzahl Ausgangsgrößen	bis zu 33

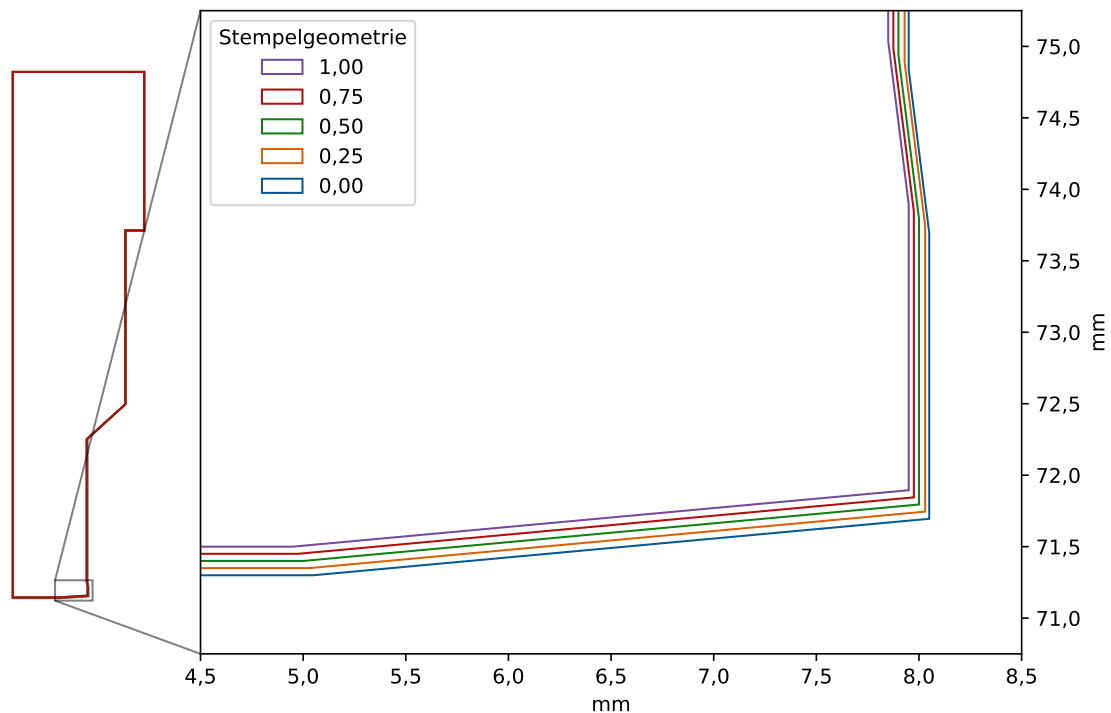
**Tabelle 4.1:** Datendokumentation

Jeder Datenpunkt steht für sich allein und ist unabhängig von den anderen. Insgesamt stehen 1 080 Datenpunkte zur Verfügung. Die Spalten entsprechen Eingangs-, beziehungsweise Ausgangsgrößen der Simulation.

Über die Simulationsdurchläufe hinweg wurden von einem Fachexperten des Instituts für Umformtechnik die Werte von den sieben in Tabelle 4.2 beschriebenen Eingangsgrößen variiert. Anhand dieser und ähnlicher Größen trainieren andere Autoren erfolgreich Surrogatmodelle [ICFQ93; PGK+23]. Es wurde ein vollständiger Versuchsplan durchgeführt. Das bedeutet, dass für jede Kombination der gegebenen Werte der Eingangsgrößen je eine Simulation durchgeführt wurde. Die Größe Zyklus beschreibt die Anzahl der Umformhübe pro Zeiteinheit und steht somit im Zusammenhang mit der Geschwindigkeit des Umformvorgangs. Zwei verschiedene Geschwindigkeiten wurden berücksichtigt, *Norm* und *High*. Durch den Einsatz unterschiedlicher Schmiermittel kann die Reibung zwischen Werkzeug und Werkstück variiert werden. Die Werte 0,08, 0,10 und 0,12 entsprechen dabei einem Reibfaktor. Für die beiden Größen Werkzeugmaterial und Werkstückmaterial entsprechen die Werte *Low*, *Norm* und *High* einer relativen Verschiebung der Fließkurve der verwendeten Materialien. Aus einer Fließkurve lassen sich mehrere Kennzahlen für ein Material ableiten, im Allgemeinen beschreiben sie das Materialverhalten bei der Umformung [DB16].

Eingangsgröße	Werteliste	Beschreibung
Zyklus	<i>Norm</i> und <i>High</i>	Anzahl Hübe pro Sekunde
Reibung	0,08, 0,10 und 0,12	Reibfaktor
Werkstückmaterial	<i>Low</i> , <i>Norm</i> und <i>High</i>	Relative Anpassung der Fließkurve des Materials
Werkzeugmaterial	<i>Low</i> , <i>Norm</i> und <i>High</i>	ebenso
Werkstückgeometrie	0 und 1	Relative Veränderung der Geometrie
Stempelgeometrie	0,00, 0,25, 0,50, 0,75 und 1,00	ebenso
Matrizengeometrie	0 und 1	ebenso

**Tabelle 4.2:** Eingangsgrößen der Simulation.



**Abbildung 4.1:** Darstellung der relativen Veränderung der Stempelgeometrie in den Trainingsdaten. Jede Farbe steht für eine andere, getestete Stempelgeometrie. Da die Veränderungen bei Betrachtung des gesamten Stempels nur gering ausfallen, ist auf der rechten Seite der relevante Teil des Stempels vergrößert dargestellt.

Die Werte der Geometriegrößen stellen dagegen jeweils eine relative Veränderung der Geometrie des Querschnitts von Stempel, Matrize und Werkstück zwischen den Simulationen dar. Genauer gesagt beschreiben sie relative Verschiebungen relevanter Punkte gemäß einem Vektor in einem zweidimensionalen Raum. Die entsprechenden Realisierungen sind für den Stempel in Abbildung 4.1 dargestellt. Diese zeigt die Hälfte des symmetrischen Querschnitts der verschiedenen untersuchten Stempelgeometrien. Die Veränderungen sind bei Betrachtung des gesamten Stempels gering, daher wurde der relevante Teil der Geometrie vergrößert auf der rechten Seite abgebildet. Jede getestete Geometrie ist mit einer eigenen Farbe dargestellt. Dabei entsprechen die Grenzwerte 0 und 1 den beiden extremen Ausprägungen, in der Grafik jeweils violett und blau dargestellt. Die Werte dazwischen repräsentieren proportionale Veränderungen. Nimmt man den Wert 0 als Ausgangspunkt, so erhält man die Geometrie des Wertes 0,5, indem die Hälfte der Veränderungen, die zum Wert 1 führen, umgesetzt werden. Für die Werkstück- und Matrizegeometrie verhält es sich ähnlich, jedoch wurden jeweils nur zwei verschiedene Ausprägungen berücksichtigt. Die tatsächlichen, absoluten Maße der Geometrien sind aus den Daten nicht direkt abzulesen. Die gegebenen Werte beschreiben diese jedoch korrekt und eignen sich für das Training eines Surrogatmoduls. Die vorliegende Kodierung kann darüber hinaus als Ergebnis einer Dimensionsreduktion in Kombination mit einer Skalierung, wie sie häufig vor Einsatz von ML-Methoden eingesetzt wird, verstanden werden [DB21]. Daher wurden diese Größen so belassen.

## 4 Ergebnisse der explorativen Datenanalyse

Bezeichnung	Plausibler Wertebereich	Beschreibung
Napfbodendicke	6,0 mm bis 6,5 mm	Maximale Dicke des Bodens des Werkstücks nach der Umformung, in mm
Wandstärke	3,0 mm bis 3,5 mm	Maximale Dicke der Wand des Werkstücks nach der Umformung, in mm
Kraft	< 500 000 N	Maximale Kraft im Stempel in y-Richtung während des Umformprozesses, in N
Temperatur	100 °C bis 400 °C	Maximale Temperatur des Werkstücks während des Umformprozesses, in °C
Umformgrad	3,0 bis 5,0	Maximaler Umformgrad des Werkstücks, ohne Einheit
Schädigung	< 1,0	Maximale Schädigung des Werkstücks, ohne Einheit

**Tabelle 4.3:** Ausgangsgrößen der Simulation.

In den Daten befinden sich etwa 30 Ausgangsgrößen der Simulation, die genaue Anzahl schwankt über die verschiedenen Dateien. Der Fachexperte vom Institut für Umformtechnik hat davon sechs als besonders interessant gekennzeichnet und für sie realistische Wertebereiche definiert. Daher konzentriert sich die weitere Untersuchung auf diese, in Tabelle 4.3 beschriebenen, Größen. Davon ist die Napfbodendicke eines der wichtigsten Qualitätsmerkmale des hier angewendeten Umformprozesses und gibt die maximale Dicke des Bodens des Werkstücks nach dem Umformprozess an. Plausible Werte dieser Größe liegen im Bereich 6,0 mm bis 6,5 mm. Die Wandstärke beschreibt analog die Stärke der Wand des Werkstücks auf der Höhe 31,6 mm und enthält Werte zwischen 3,0 mm und 3,5 mm. Die Kraft in y-Richtung, die der Stempel aushält, sowie die Temperatur des Werkstücks werden über die Geometrie und die Zeit maximiert in den Ausgangsgrößen Kraft und Temperatur angegeben. Plausible Werte der Kraft überschreiten 500 000 N nicht. Es sind Temperaturen zwischen 100 °C und 400 °C zu erwarten. Der Umformgrad, auch logarithmische Dehnung genannt, berechnet sich aus den Abmessungsänderungen und den Anfangsabmessungen und verfügt über keine Einheit. Werte dieser Größe liegen zwischen 3,0 und 3,5. Die Schädigung ist eine Kennzahl, die mit der Verformung des Werkstücks wächst. Erreicht sie einen kritischen Wert, der durch Experimente ermittelt wird, so kommt es zum Bruch. Auch die Schädigung ist einheitenlos und enthält Werte kleiner 1.

## 4.2 Datenqualität

Plaue [Pla21] beschreibt Datenqualität wie folgt: „Die Qualität von Daten bemisst, inwieweit diese bestimmungsgemäß verwendet werden können.“ In der Praxis sind Daten oft von geringer Qualität [BH21; Han12]. Falsche Werte, beziehungsweise der Einfluss von Störgrößen auf den Datensatz, sind kaum zu vermeiden [Han12; PA05]. Dabei ist die Datenqualität von hoher Bedeutung für die Vorhersagegüte erzeugter Modelle und damit für den Erfolg eines jeden ML-Projekts [AB18; Mur22]. Sie wird auch als das größte Problem bei der Anwendung von DM-Methoden bezeichnet [BH21]. Dementsprechend nimmt die Untersuchung und Steigerung der Datenqualität einen erheblichen zeitlichen Aufwand in Anspruch [KM06; WWIT16]. Das gilt auch für das Anwendungsgebiet der Fertigungstechnik [WWIT16].

Im Folgenden wird die Qualität der zur Verfügung gestellten Daten strukturiert untersucht. Dabei werden sechs definierte Merkmale der Datenqualität berücksichtigt [Pla21]: Vollständigkeit, Konsistenz, Gültigkeit, Eindeutigkeit, Richtigkeit und Aktualität.

**Vollständigkeit** Bezüglich der Vollständigkeit wird überprüft, ob alle relevanten Daten zur Verfügung stehen oder Informationen fehlen. In den Daten sind keine Einträge enthalten, die auf fehlende Werte hinweisen. Allerdings befinden sich in vier Dateien nur 34 Spalten, während die übrigen 36 aufweisen. Beide zusätzliche Spalten bestehen aus denselben Werten und sind nicht identisch mit einer der übrigen Spalten. Es existiert also in 50 von 54 Dateien effektiv eine weitere Größe. Dabei handelt es sich um eine nicht näher spezifizierte Ausgabe der Simulation, der vonseiten des Fachexperten keine gesteigerte Bedeutung beigemessen wird. Entsprechend wurde sie in weiteren Untersuchungen nicht beachtet. Die übrigen Daten sind vollständig.

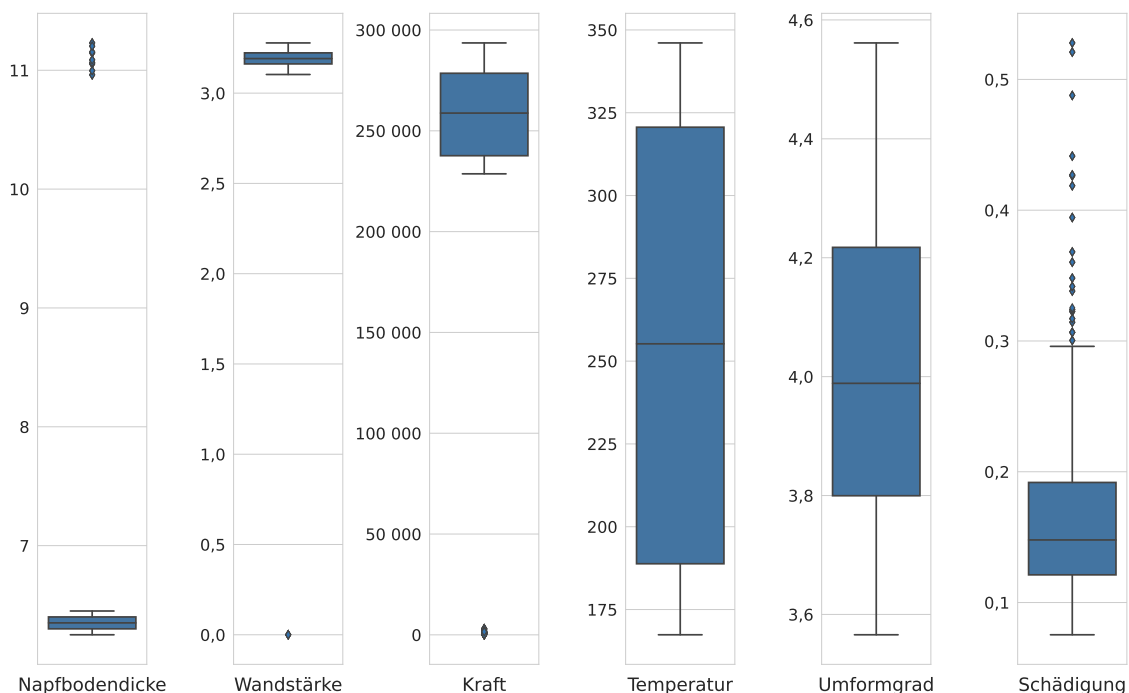
**Konsistenz** Ein Problem mit der Konsistenz liegt vor, wenn redundant gespeicherte Daten sich widersprechen. Laut Fachexperten befinden sich unter den 34 Spalten der Daten fünf redundante Paare. Für zwei dieser Paare stimmen die Werte jeweils nicht überein. Eine erneute Prüfung durch den Fachexperten ergab, dass es sich nicht um redundante Größen handelt. Die Semantik der Werte ist zwar ähnlich, aber nicht identisch. Somit liegt kein Konsistenzproblem vor, sodass alle Spalten beibehalten werden können.

**Gültigkeit** Die Gültigkeit beschreibt die formale Korrektheit der Daten. Dazu gehört, ob die Werte in einem plausiblen Bereich liegen. Dies ist für 2 855 Werte in 281 von insgesamt 1 080 Datenpunkten nicht der Fall. Diese Werte sind dem Fachexperten des Instituts für Umformtechnik zufolge zwar physikalisch möglich, aber unter den gegebenen Bedingungen unglaubwürdig. Zu dieser Zahl tragen alle Spalten der Datensätze bei. Betrachtet man lediglich die vom Fachexperten ausgewählten Eingangs- und Ausgangsgrößen, die in Tabelle 4.2 und Tabelle 4.3 festgehalten sind, so reduziert sich die Zahl der betroffenen Datenpunkte auf 62 von insgesamt 1 080. Diese 62 Werte gehören zu den Ausgangsgrößen Napfbodendicke, Wandstärke und Kraft. Die Gültigkeit der Daten ist daher zu etwa 94 % gegeben.

**Eindeutigkeit** Unter Eindeutigkeit versteht man, dass jede Entität eindeutig identifizierbar ist. Das betrifft hier die Simulationsdurchläufe. Ein solcher ist gekennzeichnet durch eine Kombination von Eingangsgrößenwerten. Keine Kombination kommt in den Daten mehrfach vor und identifiziert somit einen Simulationsdurchlauf eindeutig. Die Eindeutigkeit ist gegeben.

**Richtigkeit** Die Richtigkeit der Daten wird daran gemessen, ob sie den wahren Stand der Dinge wiedergeben. Hier stellen die Daten Ein- und Ausgaben einer Simulation dar und es wird davon ausgegangen, dass sie fehlerfrei übertragen und seit ihrer Erzeugung nicht verändert wurden. Dementsprechend wird die Richtigkeit als gegeben angesehen. Letztendlich ist bei der Simulation eines Umformprozesses interessant, inwieweit die Ergebnisse mit physischen Experimenten übereinstimmen. Dies zu überprüfen ist nicht Bestandteil der vorliegenden Arbeit.

## 4 Ergebnisse der explorativen Datenanalyse



**Abbildung 4.2:** Boxplots für die Eingangsgrößen.

**Aktualität** Sind Informationen veraltet oder überholt, so sind sie nicht aktuell. Die Daten wurden wenige Monate vor dieser Untersuchung erzeugt. Darüber hinaus stellen sie die Simulation eines physikalischen Prozesses dar. Unter der Annahme, dass die Simulation die zur Zeit bekannten, zugrundeliegenden Gesetzmäßigkeiten adäquat abbildet, ist die Aktualität gegeben.

Abgesehen von der Gültigkeit sind die Daten somit von hoher Qualität. Die abweichenden Einträge sind genauer zu untersuchen. Extreme Werte, die sich wesentlich von den übrigen unterscheiden, werden Ausreißer genannt [FHK+16]. Zusätzlich wird im Folgenden auch der Begriff Ausreißerdatenpunkt verwendet. Damit werden Datenpunkte bezeichnet, die mindestens einen solchen auffälligen Wert, also einen Ausreißer beinhalten. Ausreißer können Ausdruck fehlerhafter Daten sein und die Vorhersage mittels ML-Methoden erschweren [Pla21]. Eine Methode zur Identifizierung von Ausreißerkandidaten, also Werten, bei denen es sich möglicherweise um Ausreißer handelt, stützt sich auf den Interquartilsabstand [Alb22]. Wie in Abschnitt 2.2.3 besprochen werden zur grafischen Anwendung dieser Methode Boxplots eingesetzt.

In den Boxplots in Abbildung 4.2 finden sich Ausreißerkandidaten für vier der gezeigten Ausgangsgrößen. Sie sind als Rauten dargestellt. Besonders auffällig sind die Werte für die Napfbodendicke, die Wandstärke und die Kraft, da sie einen deutlichen Abstand zur Verteilung der jeweiligen, übrigen Werte haben. Die jeweiligen Wertebereiche sind Tabelle 4.4 zu entnehmen. Diese insgesamt 114 Werte können als Ausreißer bezeichnet werden. Insbesondere liegen die Ausreißer der Napfbodendicke und der Wandstärke nicht in den vom Fachexperten definierten, sinnvollen Wertebereichen. Für die Schädigung ist das nicht der Fall. Die für diese Ausgangsgröße als Rauten dargestellten Werte haben keine so große Distanz zur Verteilung der übrigen Werte. Zusätzlich liegen sie im definierten Wertebereich, weshalb es sich bei diesen Werten möglicherweise nicht

Ausgangsgröße	Anzahl Ausreißerdatenpunkte	Wertebereich der Ausreißer	Wertebereich ohne Ausreißer
Napfbodendicke	27	10,96 bis 11,23	6,25 bis 6,45
Wandstärke	6	0,00 bis 0,00	3,10 bis 3,28
Kraft	81	40,54 bis 3 099,83	228 671 bis 293 596

**Tabelle 4.4:** Übersicht über Ausreißer in den Daten.

um Ausreißer handelt. Sie könnten beispielsweise Ausdruck einer schiefen Verteilung sein. Das wird in Abschnitt 4.4 untersucht. Zur weiteren Analyse werden die 27 Datenpunkte, die auffällige Werte in der Ausgangsgröße Napfbodendicke beinhalten, identifiziert und vorübergehend entfernt. Dabei ist zu bemerken, dass dadurch auch die sechs Ausreißer der Größe Wandstärke wegfallen. Offenbar existiert ein Zusammenhang zwischen den extremen Werten beider Ausgangsgrößen. Für die Ausreißer der Größe Kraft ist das nicht der Fall. Es fällt darüber hinaus auf, dass Ausreißer in der Napfbodendicke lediglich für hohe Temperaturen und geringe Schädigungen auftreten. Es findet sich kein derartiger Zusammenhang für die anderen Ausgangsgrößen, Kraft und Umformgrad. Die Verteilungen der Ausgangsgrößen Temperatur, Umformgrad und Schädigung verändern sich nicht wesentlich, wenn die Ausreißerdatenpunkte entfernt werden. Neben Korrelationen zwischen Ausgangsgrößen sind auch solche zwischen Ausgangsgrößen und Eingangsgrößen von Interesse. Dabei ist festzustellen, dass es für Ausreißer in der Napfbodendicke eine notwendige Bedingung ist, dass der Zyklus *High* und die Werkstückgeometrie sowie die Matrizengeometrie 1 sind. Es ist aber keine hinreichende Bedingung, in 80 % der Fälle tritt für diese Eingabewerte kein Ausreißer in der Napfbodendicke auf.

Es stellt sich die Frage, wie mit Ausreißern und den insgesamt 105 betroffenen Datenpunkten verfahren werden soll. Eine Möglichkeit ist, die Ausreißer im Rahmen einer Imputation mit möglichst plausiblen Werten zu ersetzen [Pla21]. Dabei ist das Ziel, die Verzerrung von Analyseergebnissen zu minimieren, nicht jedoch, die korrekten Werte zu finden. Eine Imputation kann durch das Einsetzen von Lageparametern wie dem Mittelwert oder dem Modus erfolgen. Darüber hinaus sind auch aufwendigere Methoden mittels Regressionsverfahren denkbar. Doch bereits die Auswahl des Lageparameters kann eine Hürde sein. So ist laut des Fachexperten keiner geeignet, um die Ausreißer der Napfbodendicke zu ersetzen. Auch ist nicht geklärt, ob die Werte der Ausreißerdatenpunkte für die Ausgangsgrößen Temperatur, Umformgrad und Schädigung korrekt sind. Zwar passen diese zur gesamten Werteverteilung der jeweiligen Größen, jedoch ist nicht auszuschließen, dass auch sie fehlerbehaftet sind und die Approximation durch ein Surrogatmodell negativ beeinflussen können.

Eine einfache und häufig eingesetzte Alternative zur Imputation ist die Complete-Case-Analyse [DB21; KBT11; Pla21]. Dabei werden die betroffenen Datenpunkte entfernt und bei der weiteren Untersuchung nicht berücksichtigt. Ihr Nachteil liegt darin, einen Teil der Daten für die Analyse zu verlieren. Immerhin sind etwa 10 % der zur Verfügung stehenden Datenpunkte von Ausreißern in den Ausgangsgrößen betroffen. Dabei gilt neben der Datenqualität auch die Größe der Datenmenge als Indikator für den Erfolg eines ML-Projekts [BH21]. Der Einfluss von Ausreißerdatenpunkten und dem Entfernen derselben wird in dieser Arbeit überprüft, indem Modelle mit und ohne die entsprechenden Datenpunkte trainiert und deren Vorhersagen verglichen werden (vgl. Abschnitt 4.5.5).

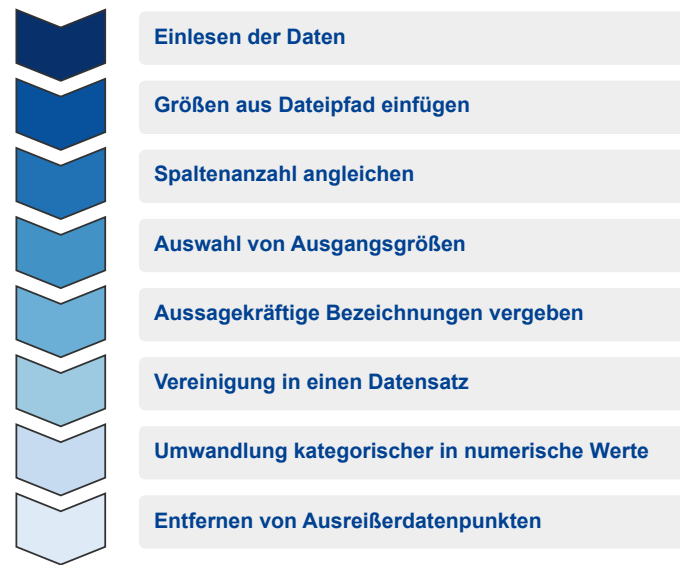


Abbildung 4.3: Übersicht über die Schritte der Datenvorbereitung.

### 4.3 Datenvorbereitung

Bestimmte datenspezifische Verarbeitungsschritte sind notwendig, um die Daten für die weitere Untersuchung vorzubereiten. Dazu gehören die Datenintegration in eine Form, welche die spätere Verwendung begünstigt, sowie Maßnahmen zur Steigerung der Datenqualität. Ähnliche Verfahrensweisen werden Datenbereinigung [Pla21] oder Datenvorverarbeitung [Han12] genannt. Eine Übersicht über die hier durchgeführten Maßnahmen ist in Abbildung 4.3 gegeben.

Zuerst müssen die Daten, die über 54 Dateien in zwei Ordnern verteilt sind, eingelesen werden. Dabei sind Ordnerstruktur und Dateinamen zu beachten, da diese wichtige Informationen über Eingangsgrößen beinhalten. Ein Beispiel für einen Dateipfad sieht wie folgt aus, wobei die Bedeutungen der jeweiligen Abschnitte unten annotiert sind:

$\frac{\text{/Cycle\_Norm}}{\text{Zyklus=Norm}}$    
  $\frac{\text{/F12}}{\text{Reibung=0,12}}$    
  $\frac{\text{CHigh}}{\text{Werkstückmaterial=High}}$    
  $\frac{\text{AiSiLow}}{\text{Werkzeugmaterial=Low}}$    
 .csv

Diese Werte müssen aus dem Dateipfad extrahiert und den sonstigen Daten hinzugefügt werden. Das heißt, das entsprechende Spalten erzeugt und mit den korrekt extrahierten Werten befüllt werden müssen. Die eingelesenen Datensätze – einer pro Datei – enthalten unterschiedlich viele Spalten. Solche, die nicht in allen Dateien vorhanden sind, werden wie in Abschnitt 4.2 beschrieben entfernt. Abgesehen von den sieben Eingangsgrößen befinden sich einige Ausgangsgrößen in den Daten. Von letzteren werden alle bis auf die sechs laut Fachexperten besonders interessanten entfernt (vgl. Tabelle 4.3). Die Spalten haben zu diesem Zeitpunkt teils schwer zu interpretierende Namen wie „Object Load.2“, die sich vereinzelt auch nur durch ihre Nummerierung unterscheiden. Zur einfacheren Handhabung und Visualisierung werden aussagekräftige Namen für die verbleibenden Größen gewählt. Anschließend werden die einzelnen Datensätze zu einem zusammengefügt.



Dateneigenschaft	Vor Vorverarbeitung	Nach Vorverarbeitung
Anzahl Datensätze	54	1
Anzahl Datenpunkte	1 080	975
Anzahl Eingangsgrößen	3 (+4 in Verzeichnisstruktur)	7
Anzahl Ausgangsgrößen	> 29	6
Datentypen	numerisch und kategorisch	numerisch

**Tabelle 4.5:** Übersicht über die Dateneigenschaften vor und nach der Vorverarbeitung.

Das Skalenniveau der Eingangsgrößen ist nicht eindeutig. Für den Zyklus, das Werkzeugmaterial und das Werkstückmaterial liegen kategorische Werte vor, die einer Ordinalskala zugeordnet werden können. Die Geometrien sind zwar als Zahlen dargestellt, ob es sich dabei allerdings auch um Kategorien handelt ist nicht ohne Weiteres ersichtlich. Für die Visualisierung und die Anwendung von ML-Methoden ist es jedoch von Vorteil, lediglich mindestens intervallskalierte, metrische Werte zu verwenden. Diese Skala erlaubt Vergleiche zwischen den Differenzen unterschiedlicher Werte [Han12], die bei der Visualisierung grafisch dargestellt und von ML-Algorithmen berechnet werden. Wie vom Fachexperten bestätigt, repräsentieren die Werte aller sieben Eingangsgrößen äquidistante, numerische Veränderungen. Somit können die kategorischen Werte der drei Größen Zyklus, Werkzeugmaterial und Werkstückmaterial jeweils in eine Intervallskala mit gleichmäßig über das Intervall  $[0, 1]$  verteilten, metrischen Werten übertragen werden. Die übrigen Eingangsgrößen liegen bereits in dieser Form vor. Dabei ist der Nullpunkt der Größen nicht natürlich, aber Unterschiede zwischen Differenzen sind aussagekräftig. Werden die Eingangsgrößen im Rahmen des Trainings eines Surrogatmodells gleich skaliert, ist der Nullpunkt ohnehin nicht weiter von Bedeutung. Eine solche Umformung wird häufig durchgeführt [Han12; STV04]. In einem letzten, optionalen Schritt werden die in Abschnitt 4.2 identifizierten Ausreißerdatenpunkte entfernt. Tabelle 4.5 zeigt einige Dateneigenschaften vor und nach der hier beschriebenen Vorverarbeitung.

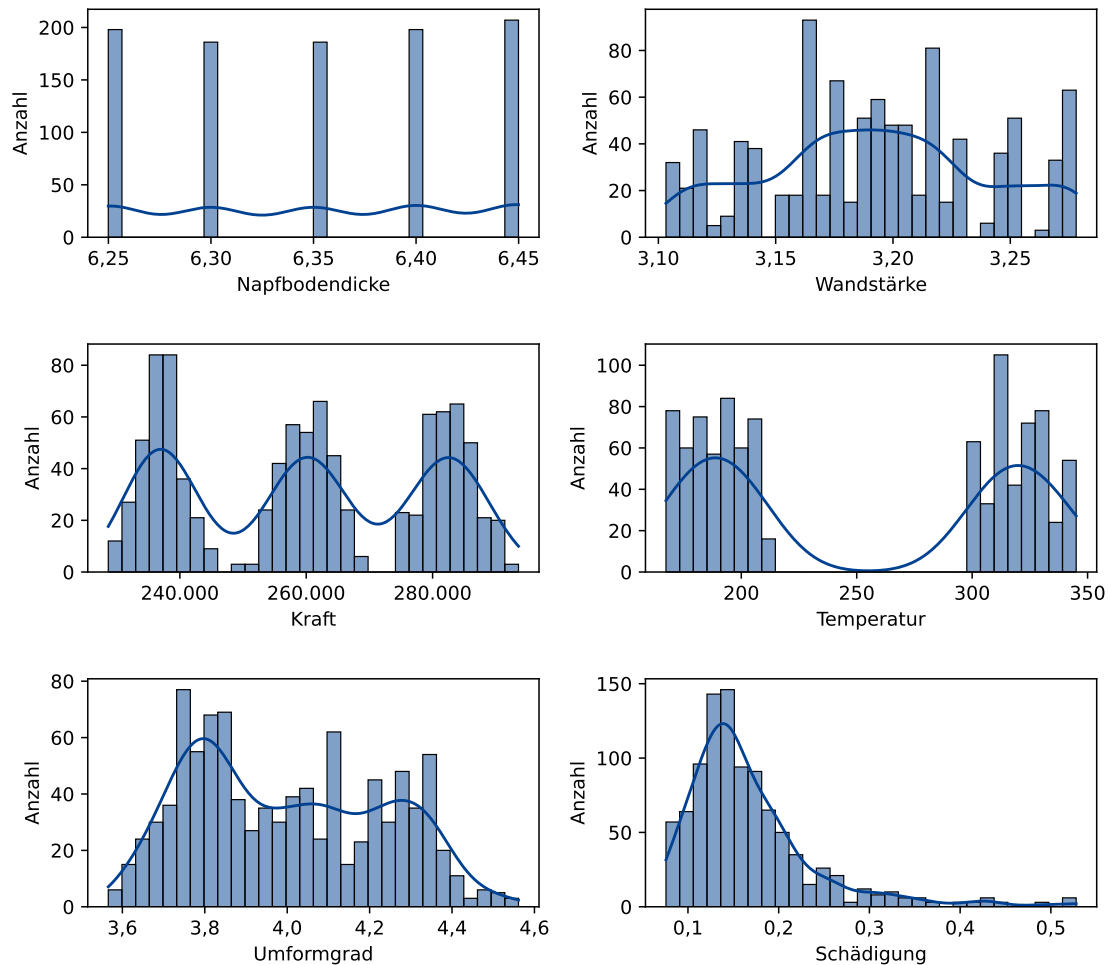
## 4.4 Datenvisualisierung und Korrelationsanalyse

Visualisierungen sind wichtige und häufig eingesetzte Hilfsmittel im Rahmen der Untersuchung von Daten [FBS15; Mur22]. Sie ermöglichen es, ein Verständnis der Daten und der Beziehungen unterschiedlicher Größen in ihnen zu erlangen [Alb22; Han12]. Dabei ist es sinnvoll, zunächst einzelne Größen darzustellen und im Anschluss Visualisierungsverfahren einzusetzen, die Beziehungen zwischen den Größen erkennen lassen [Alb22]. Vor der grafischen Darstellung werden die Daten wie in Abschnitt 4.3 beschrieben vorbereitet, inklusive der Entfernung von Ausreißerdatenpunkten. Mit Ausreißern sind die Eigenschaften von Verteilungen schwerer zu erkennen, da ein erheblicher Teil der Grafik den leeren Raum zwischen den Ausreißern und den übrigen Daten darstellt.

### 4.4.1 Histogramme

Histogramme, eine Form von Säulendiagrammen, stellen eine gute Möglichkeit dar, die Werteverteilung einzelner Größen zu untersuchen [Alb22]. Zur Vorbereitung werden die Werte einer zumindest ordinalskalierten Größe in  $k$  gleich große Intervalle aufgeteilt. Diese werden auf der  $x$ -Achse abgebildet. Pro Intervall wird eine Säule erstellt, die in der  $x$ -Dimension dem Intervall entspricht.

## 4 Ergebnisse der explorativen Datenanalyse



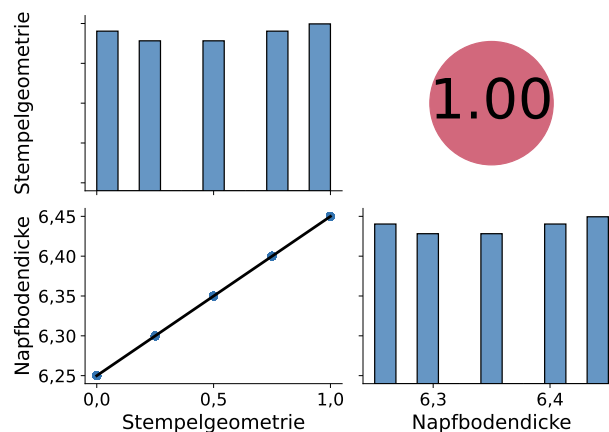
**Abbildung 4.4:** Histogramme der sechs untersuchten Ausgangsgrößen inklusive Kerndichteschätzung.

Für die Intervallbreite wird die Faustregel  $\sqrt{N}$  empfohlen, wobei  $N$  der Anzahl an Datenpunkten entspricht [FHK+16]. Die Fläche einer Säule ist direkt proportional zur Anzahl der Datenpunkte, deren Werte der jeweiligen Größe im Intervall der Säule liegen. Da hier alle Säulen die gleiche Breite haben, gilt das auch für die Höhe der Säulen. Zusätzlich kann eine Dichtefunktion mittels einer Kerndichteschätzung erzeugt und eingezeichnet werden. So kann ein Histogramm Aufschluss über die Verteilung von Werten über den Wertebereich einer Größe geben. Abbildung 4.4 zeigt sechs Histogramme, jeweils eins pro untersuchter Ausgangsgröße. Die Anzahl der Intervalle wurde in Anlehnung an oben genannte Faustregel auf 30 festgelegt. Für die Napfbodendicke fällt auf, dass nur fünf etwa gleich hohe Säulen sichtbar sind. Das bedeutet, dass alle Werte nur in diesen fünf Intervallen liegen. Die Verteilung der Wandstärke zeigt einige Häufungen und Lücken, die Abbildung lässt sonst keine Besonderheiten erkennen. Für die Kraft und die Temperatur sind drei beziehungsweise zwei getrennte Verteilungen zu sehen. Der Umformgrad zeigt keine deutliche Trennung mehrerer Verteilungen, allerdings können ebenfalls mehrere lokale Maxima ausgemacht werden. Für die Schädigung ist eine rechtsschiefe Verteilung erkennbar.

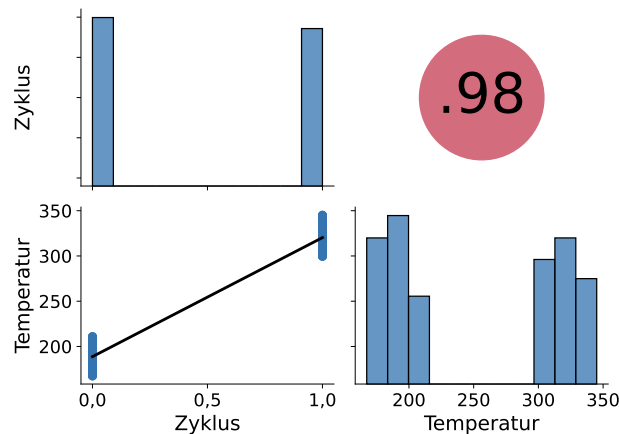
### 4.4.2 Streudiagramm-Matrizen

Histogramme können die Eigenschaften der Verteilungen einzelner Größen wiedergeben, aber Beziehungen zwischen Größen lassen sich aus ihnen nicht ablesen. Zu diesem Zweck kann eine Streudiagramm-Matrix eingesetzt werden [Han12; Mur22]. Das ist eine Matrix von Abbildungen, wobei für alle Größen jeweils eine Zeile und eine Spalte vorgesehen ist. Auf der Diagonalen – also dort, wo sich Zeilen und Spalten derselben Größe schneiden – sind Histogramme der zugehörigen Größe abgebildet. Ansonsten sind in der Matrix Streudiagramme enthalten, die die jeweiligen Größen aus der Zeile und der Spalte gegenüberstellen. Streudiagramme vermitteln einen Eindruck von den Zusammenhängen zwischen zwei Größen [FHK+16]. Für solche Diagramme aller Paare von Größen werden nur die Einträge unter der Diagonalen benötigt. Daher kann der Bereich über dieser für andere Visualisierungen genutzt werden. Beispielsweise kann hier pro Eintrag ein Kreis gezeichnet werden, der in Farbe und Ausmaß der Korrelation der beiden entsprechenden Größen aus Zeile und Spalte entspricht. Hier wird ein für metrische Größen üblicher Korrelationskoeffizient nach Bravais-Pearson verwendet [FHK+16]. Korrelationen zwischen Eingangs- und Ausgangsgrößen beschreiben den Zusammenhang beider und können zur Auswahl von Größen für das Training eines Surrogatmodells herangezogen werden [GFW+19; HYDF16; VGG21].

Streudiagramm-Matrizen können sehr groß werden, weshalb hier nur Ausschnitte abgebildet sind. Einen solchen Ausschnitt zeigt Abbildung 4.5 für die Eingangsgröße Stempelgeometrie und die Ausgangsgröße Napfbodendicke. Wie schon bei der Untersuchung der Histogramme der Ausgangsgrößen festgestellt, enthält die Napfbodendicke abgesehen von Ausreißern nur Werte aus fünf disjunkten Intervallen. Aus der Abbildung ist abzulesen, dass es sich um sehr kleine Intervalle handelt. Tatsächlich sind es lediglich fünf diskrete Werte, wie im Streudiagramm von Abbildung 4.5 zu sehen ist. Das Gleiche gilt für die Stempelgeometrie. Interessanterweise ist das Streudiagramm beider Größen eine Gerade, was von einem starken linearen Zusammenhang zwischen den beiden zeugt. Ein Korrelationskoeffizient von 1, was dem maximalen Wert entspricht, bestätigt diesen Eindruck. Daraus ist zu schließen, dass lediglich anhand der Eingangsgröße Stempelgeometrie die Ausgangsgröße Napfbodendicke perfekt vorhergesagt werden kann. Eine einfache Tabelle, in der die jeweiligen fünf Werte der Größen gegenübergestellt sind, reicht als Surrogatmodell aus. Auf ML-Methoden muss in diesem Fall nicht zurückgegriffen werden.



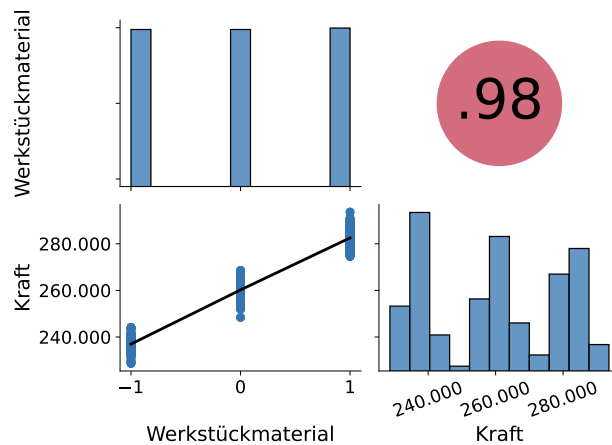
**Abbildung 4.5:** Ausschnitt der Streudiagramm-Matrix für die Eingangsgröße Stempelgeometrie und die Ausgangsgröße Napfbodendicke.



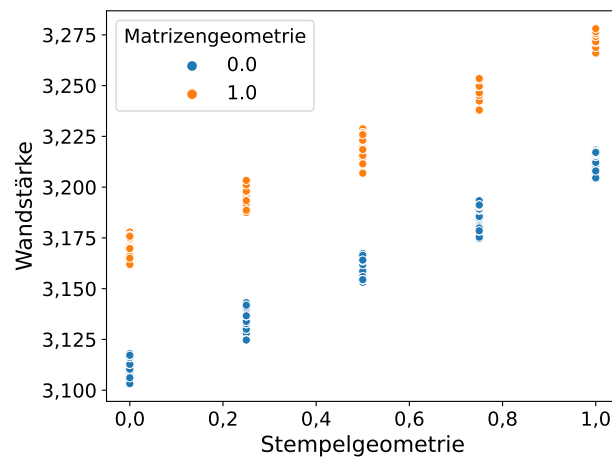
**Abbildung 4.6:** Ausschnitt der Streudiagramm-Matrix für die Eingangsgröße Zyklus und die Ausgangsgröße Temperatur.

Die Histogramme der Größen Temperatur und Kraft zeigen Verteilungen mit mehreren lokalen Maxima. Dabei kann es sich um mehrere getrennte Verteilungen handeln. Aufgrund dieser Auffälligkeiten ist die Untersuchung ihrer Beziehungen zu Eingangsgrößen von besonderem Interesse. Tatsächlich zeigt Abbildung 4.6 einen bemerkenswerten Zusammenhang zwischen dem Zyklus und der Temperatur. Mit einem Koeffizienten von 0,98 ist die Korrelation zwischen beiden Größen als stark einzustufen. Das Streudiagramm zeichnet ein ähnliches Bild: Die zwei Verteilungen in der Ausgangsgröße Temperatur sind auf die zwei Werte der Eingangsgröße Zyklus zurückzuführen. Zwar ist der Wert der Temperatur anhand des Zyklus nicht eindeutig bestimmbar, aber es ist denkbar, jeweils den Mittelwert der beiden Verteilungen als Approximation zu verwenden. Ähnlich verhält es sich für die Ausgangsgröße Kraft, deren Histogramm als drei getrennte Verteilungen interpretiert werden kann. Wird sie der Eingangsgröße Werkstückmaterial entgegengestellt, so ist dabei ein starker Zusammenhang mit einem Korrelationskoeffizienten von 0,98 zu erkennen. Auch hier gilt, dass die drei Werte von Werkstückmaterial genutzt werden können, um den Mittelwert der entsprechenden Verteilung der Kraft auszugeben und diese Ausgangsgröße so zu approximieren.

Untersucht man Teilmengen der Daten, können die Zusammenhänge innerhalb einer der beiden erkennbaren Verteilungen der Ausgangsgröße Temperatur untersucht werden. Zu diesem Zweck werden nur Datenpunkte mit jeweils einem Wert der Eingangsgröße Zyklus betrachtet. Dabei sind weitere starke Korrelationen der Temperatur mit Eingangsgrößen festzustellen. Die Eingangsgröße Werkstückmaterial scheint innerhalb der beiden Verteilungen von Temperatur mit einem Korrelationskoeffizienten von 0,94 den größten Ausschlag zu machen. Analog dazu erlaubt eine Einschränkung der Eingangsgröße Werkstückmaterial eine Analyse der einzelnen Verteilungen der Ausgangsgröße Kraft. In diesem Fall sind starke Korrelationen zwischen der Kraft und den Eingangsgrößen Werkstückgeometrie sowie Stempelgeometrie zu erkennen. Daraus ist zu schließen, dass die Ausgangsgrößen Temperatur und Kraft nicht ausschließlich von den jeweils dominierenden Eingangsgrößen, die in Abbildung 4.6 und Abbildung 4.7 dargestellt sind, bestimmt werden. Innerhalb der einzelnen Verteilungen haben auch andere Größen beachtlichen Einfluss. Ein Surrogatmodell, das auf der gesamten Datenmenge trainiert wird, könnte diese Zusammenhänge innerhalb der einzelnen Verteilungen beider Ausgangsgrößen womöglich nicht erfassen. Um das zu umgehen ist es denkbar, für jede Verteilung ein einzelnes Modell zu trainieren [HRM20]. Allerdings stünden dann im Falle der Größe Kraft nur etwa 300 Datenpunkte pro Modell zur Verfügung. Für die



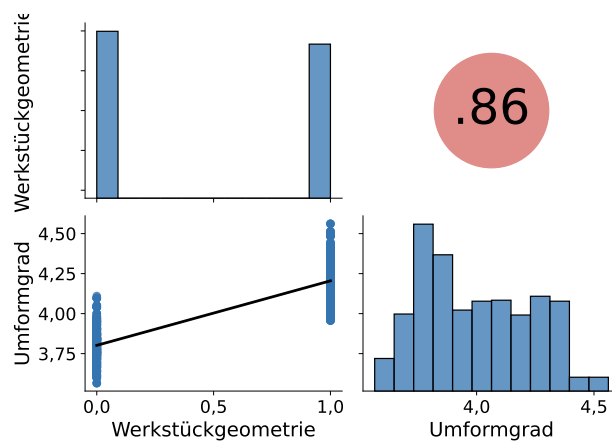
**Abbildung 4.7:** Ausschnitt der Streudiagramm-Matrix für die Eingangsgröße Werkstückgeometrie und die Ausgangsgröße Kraft.



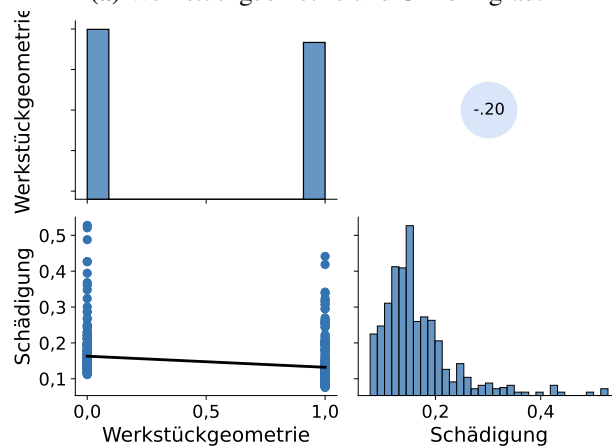
**Abbildung 4.8:** Streudiagramm für die Ausgangsgröße Wandstärke und die Eingangsgrößen Stempelgeometrie und Matrizengeometrie.

Ausgangsgröße Temperatur wären es etwa 400 Datenpunkte. Derart kleine Datensätze können die Effektivität von ML-Methoden einschränken [KBT11]. Daher wurden solche Modelle in dieser Arbeit nicht trainiert.

In Abbildung 4.8 ist ein Streudiagramm abgebildet, das die Zusammenhänge zwischen der Ausgangsgröße Wandstärke und den beiden Eingangsgrößen Stempel- und Matrizengeometrie visualisiert. Die Abbildung zeigt, dass pro Wert der Stempelgeometrie lediglich Werte in zwei disjunkten Intervallen der Wandstärke auftreten. Darüber hinaus gibt die Farbe der Punkte den Wert der Matrizengeometrie an. Dabei ist zu erkennen, dass die Matrizengeometrie bestimmt, in welchem der beiden Intervalle die Werte der Wandstärke liegen. Es handelt sich bei der Wandstärke demnach um zwei sich überlagernde Verteilungen mit jeweils fünf disjunkten Intervallen, in denen Werte liegen. Anhand der Eingangsgrößen Stempel- und Matrizengeometrie kann der Wert der Wandstärke auf ein begrenztes Intervall eingeschränkt werden. Eine Verwendung des Mittelwerts dieses Intervalls zum Zwecke der Approximation der Wandstärke ist möglich.



(a) Werkstückgeometrie und Umformgrad.



(b) Werkstückgeometrie und Schädigung.

**Abbildung 4.9:** Zwei Ausschnitte der Streudiagramm-Matrix, die die jeweils stärkste Korrelation der Ausgangsgrößen Umformgrad und Schädigung mit einer Eingangsgröße zeigen.

Der Umformgrad korreliert stark mit der Werkstückgeometrie, wie in Abbildung 4.9a zu sehen ist. Jedoch ergeben sich dabei keine getrennten Verteilungen wie zum Beispiel bei der Temperatur. Die stärkste Korrelation der Ausgangsgröße Schädigung besteht mit der Eingangsgröße Werkstückgeometrie (vgl. Abbildung 4.9b). Der Korrelationskoeffizient liegt jedoch lediglich bei  $-0,20$ . Für diese zwei Ausgangsgrößen zeigen sich keine auffälligen Zusammenhänge mit anderen Größen, aus denen sich direkt eine mögliche Approximation ableiten lässt.

Die Untersuchungen der Werteverteilungen und Korrelationen in diesem Kapitel ergeben, dass für die Ausgangsgrößen Napfbodendicke, Wandstärke, Kraft und Temperatur Approximationen auf Basis von Korrelationen möglich sind. Für die Ausgangsgrößen Umformgrad und Schädigung ergibt sich kein solcher Zusammenhang. Im folgenden Abschnitt werden mögliche Approximationen für alle sechs Ausgangsgrößen untersucht.

## 4.5 Erste Approximationen

Eine häufige Herangehensweise bei der Auswahl von ML-Methoden zur Erzeugung eines datengetriebenen Surrogatmodells ist es, verschiedene Verfahren anzuwenden und die durch diese Verfahren erzielbaren Approximationen zu evaluieren [Mur22]. Dabei ist es sinnvoll, zunächst mit simplen Methoden wie linearer Regression oder einfachen Entscheidungsbäumen zu beginnen [GE03]. Solche sind bei guter Approximation prinzipiell komplexeren Verfahren vorzuziehen und sind grundsätzlich eher in der Lage, bei einer geringen Anzahl von Trainingsdaten zu generalisieren [BAC+19; Mur22]. Ein komplexeres Modell hat möglicherweise genug Freiheitsgrade, um sich den Trainingsdaten perfekt anzupassen. Häufig entspricht die Verteilung der Trainingsdaten jedoch nicht exakt der Verteilung der Grundgesamtheit der Daten. Daher ist es möglich, dass solche Modelle auf bisher ungesehenen Daten aus der gleichen Grundgesamtheit deutlich schlechtere Vorhersagen machen als für die Trainingsdaten [Mur22].

Bevor Modelle trainiert werden können, ist festzulegen, wie der Modelltest und die Validierung durchgeführt werden. Entsprechend sind die Daten zum Zweck des Trainings, der Validierung und des Tests zu partitionieren. Alternativ kann auch eine Kreuzvalidierung wie in Abschnitt 2.2.4 beschrieben angewendet werden, was es ermöglicht, einen höheren Anteil der Daten zum Training zu verwenden. Das bietet sich bei einer geringen Anzahl von Datenpunkten an. Dann muss lediglich ein Testdatensatz gebildet werden, der nicht im Training oder zur Auswahl von Modellen verwendet werden darf. Stattdessen dient dieser dazu, die Vorhersagequalität auf ungesehenen Daten durch die Vorhersagequalität auf den Testdaten anzunähern [Mur22]. Typische Größen des Testdatensatzes liegen zwischen 10 % [VYC+23; ZXNL21] und etwa 30 % [FRC+21; HRM20]. Für diese Arbeit wurden aus den insgesamt 1 080 Datenpunkten 162 zufällig ausgewählt, um als Testdaten zu verwendet zu werden. Es befinden sich auch Ausreißer in den Testdaten. Mit den Ausreißerdatenpunkten entspricht die Größe des Testdatensatzes 15 %, ohne etwa 16 % der gesamten zur Verfügung stehenden Daten. Wenn nicht explizit anders angegeben, werden die Ausreißerdatenpunkte jedoch im Rahmen der datenspezifischen Vorverarbeitung (vgl. Abschnitt 4.3) aus Trainings- und Testdaten entfernt. Diese Vorverarbeitung wird wie schon für die Visualisierung auch für die Erzeugung erster Modelle durchgeführt.

### 4.5.1 Approximationen auf Basis von Korrelationen

Ein denkbar einfaches Modell stellt eine Wertetabelle dar, wie sie in Tabelle 4.6 zu sehen ist. Auf Basis dieser Wertetabelle kann anhand des Werts der Eingangsgröße Stempelgeometrie direkt der zugehörige Wert der Ausgangsgröße Napfbodendicke abgelesen werden. Wenn also beispielsweise die Stempelgeometrie 0,25 beträgt, dann wird für die Napfbodendicke der Wert 6,30 vorhergesagt. Die Wertetabelle lässt sich darüber hinaus in eine simple Geradengleichung der Form  $f(x) = 6,25 + 0,2 \cdot x$  umwandeln, die wiederum als lineares Regressionsmodell verstanden werden kann. Hierbei entspricht  $x$  der Stempelgeometrie und  $f$  der Annäherung der Napfbodendicke. Wie erwartet erreicht dieses Modell sowohl auf den Trainings- wie auch den Testdaten einen perfekten  $R^2$  Wert von 1. Da sich dieses Ergebnis nicht verbessern und das Modell nicht sinnvoll vereinfachen lässt, ist die Behandlung der Napfbodendicke damit abgeschlossen.

Auch für die Temperatur und die Kraft werden im Rahmen der Visualisierung vielversprechende, einfache Herangehensweisen identifiziert. Der Zyklus bestimmt, in welches von zwei Intervallen die Werte der Temperatur fallen. Sagt man für den Wert des Zyklus den Mittelwert des entsprechenden

Stempelgeometrie	Napfbodendicke
0,00	6,25
0,25	6,30
0,50	6,35
0,75	6,40
1,00	6,45

**Tabelle 4.6:** Gegenüberstellung der jeweiligen fünf Werte der Eingangsgröße Stempelgeometrie und der Ausgangsgröße Napfbodendicke. Die Werte wurden zur Steigerung der Übersichtlichkeit auf zwei Nachkommastellen gerundet.

Ausgangsgröße	$R^2$
Napfbodendicke	1,00
Temperatur	0,96
Kraft	0,96
Wandstärke	0,99

**Tabelle 4.7:** Erreichte  $R^2$  Werte für Approximationen auf Basis von Wertetabellen. Die Werte wurden auf den Testdaten ermittelt.

Intervalls der Temperatur voraus, so wird damit ein  $R^2$  Wert von 0,96 sowohl auf den Trainings- als auch auf den Testdaten erzielt. Dieser Zusammenhang kann durch die Geradengleichung  $f(x) = 189,13 + 131,38 \cdot x$  dargestellt werden. Diese Vorgehensweise kann analog auf die Ausgangsgröße Kraft übertragen werden, deren Wert vornehmlich durch das Werkstückmaterial bestimmt wird. Anhand dieser Eingangsgröße kann der Mittelwert des zugehörigen Intervalls der Kraft als Approximation ausgegeben werden. Das ergibt annähernd die Geradengleichung  $f(x) = 260\,115,30 + 22\,857,59 \cdot x$ . Dabei beträgt der  $R^2$  Wert 0,96 auf Trainings- und Testdaten. Für beide Ausgangsgrößen Temperatur und Kraft wird damit eine hohe Vorhersagequalität erreicht. Bezüglich der Wandstärke sind die Zusammenhänge komplizierter. Hier hängt das Intervall, in das die Werte dieser Größe fallen, sowohl von der Stempel- als auch von der Matrizengeometrie ab. Approximiert man die Wandstärke durch die Mittelwerte der entsprechenden Intervalle anhand der beiden Eingangsgrößen, so erreicht man  $R^2$  Werte von bis zu 0,99. Dabei ergibt sich für die Stempelgeometrie als  $x_1$  und die Matrizengeometrie als  $x_2$  annähernd die Gleichung  $f(x_1, x_2) = 3,11 + 0,10 \cdot x_1 + 0,06 \cdot x_2$ . Insgesamt sind für diese vier Ausgangsgrößen damit gute Approximationen gefunden worden. Die Ergebnisse sind in Tabelle 4.7 noch einmal zusammengefasst. Für die Ausgangsgrößen Umformgrad und Schädigung ergeben sich aus den Visualisierungen und den Korrelationskoeffizienten keine Zusammenhänge, die für eine derartige Vorhersage genutzt werden können. Daher werden für diese Größen im folgenden alternative Vorhersagemodelle diskutiert.

#### 4.5.2 Approximation durch einfache ML-Modelle

Für zwei Ausgangsgrößen wird im vorherigen Abschnitt kein Modell beschrieben: Umformgrad und Schädigung. Zur Approximation dieser Größen wird in diesem Abschnitt auf ML-Methoden zurückgegriffen, die von dem Python-Modul scikit-learn [PVG+11] in der Version 0.24.2 bereitgestellt



Ausgangsgröße	Lineare Regression		Entscheidungsbaum	
	Training	Mittelwert Kreuzvalidierung	Training	Mittelwert Kreuzvalidierung
Umformgrad	0,88	0,84	1,00	0,93
Schädigung	0,11	0,05	1,00	0,51

**Tabelle 4.8:** Übersicht über die mit simplen Modellen erreichten  $R^2$  Werte für die Ausgangsgrößen Umformgrad und Schädigung.

werden. Die  $R^2$  Metrik wird zum Vergleich dieser Methoden verwendet und in zweifacher Weise berechnet. Einmal wird der Wert auf den gesamten Trainingsdaten ermittelt. Daran lässt sich ablesen, wie gut sich das Modell an die Trainingsdaten anpasst. Zur Modellauswahl wird als Annäherung der Vorhersagequalität auf bisher ungesehenen Daten der Mittelwert einer Kreuzvalidierung mit zehn Teilmengen herangezogen. Zu diesem Zweck wird nicht der Testdatensatz eingesetzt, damit nicht ein für die Testdaten optimiertes Modell gebildet wird. Zusätzlich zu den Vorverarbeitungsschritten aus Abschnitt 4.3 werden die Daten vor dem Training von Modellen standardisiert. Das dient dem Ausgleich des Einflusses von Größen unterschiedlicher Wertebereiche und kann die Approximation durch ML-Verfahren verbessern [Han12]. Im Anschluss daran wird zunächst ein Modell der linearen Regression trainiert. Als simples Verfahren ist es für einen ersten Versuch geeignet. Die berechneten Werte der  $R^2$  Metrik sind Tabelle 4.8 zu entnehmen. Für die Größe Umformgrad wird damit ein  $R^2$  Wert von 0,84 erreicht. Dies ist der Mittelwert, der im Rahmen einer Kreuzvalidierung mit zehn Teilmengen auf den Trainingsdaten ermittelt wurde. Das erreicht nicht die Vorhersagequalität der im vorherigen Abschnitt beschriebenen Vorgehensweisen für andere Ausgangsgrößen. Zur Vorhersage der Schädigung eignet sich die lineare Regression dagegen nicht. Hier kann lediglich ein  $R^2$  Wert von 0,11 auf den Trainingsdaten und 0,05 in der Kreuzvalidierung erzielt werden. Ein solches Modell ist kaum besser, als unabhängig von den Eingangsgrößen stets den Mittelwert der Ausgangsgröße auszugeben. Es ist demnach anzuraten, andere Verfahren in Erwägung zu ziehen.

Als Nächstes kommt daher ein Entscheidungsbaum zum Einsatz, wobei die Hyperparameter nicht optimiert und für sie Standardwerte von scikit-learn 0.24.2 verwendet werden. Damit wird im Training ein perfekter  $R^2$  Wert von 1 erreicht. Für die Ausgangsgröße Umformgrad ist der  $R^2$  Mittelwert in der Kreuzvalidierung mit 0,93 deutlich höher als bei der linearen Regression. Das gilt auch für die Schädigung, jedoch wird hier lediglich ein  $R^2$  Mittelwert von 0,51 in der Kreuzvalidierung erreicht. Dieser Wert liegt deutlich unter dem  $R^2$  Wert, der für die Trainingsdaten erzielt wird. Mehrere Erklärungen dafür sind plausibel. Die Kreuzvalidierung beruht auf der Annahme, der Algorithmus ist stabil. Das heißt, er generiert ähnliche Modelle bei geringer Veränderung der Trainingsdaten. Obwohl bei einer Kreuzvalidierung mit zehn Teilmengen, wie sie hier durchgeführt wird, auch gute Ergebnisse für Entscheidungsbäume erzielt werden, ist diese Eigenschaft für diese Klasse von ML-Modellen im Allgemeinen nicht erfüllt [Bre93]. Ein weiterer Grund kann die fehlende Regularisierung sein, denn eine sehr gute Vorhersagequalität auf den Trainingsdaten bei gleichzeitig deutlich schwächerem Abschneiden in der Validierung ist ein Anzeichen von Überanpassung. Das ist bei Entscheidungsbäumen ein häufig auftretendes Problem [KBT11]. In diesen kann Regularisierung durch die Begrenzung der Tiefe oder durch nachgelagertes Kürzen des Baumes erreicht werden [Bre93]. Keine dieser Maßnahmen wurde hier ergriffen. Eine nähere Untersuchung des Baumes für die Ausgangsgröße Schädigung ergab, dass dieser mit einer Tiefe von 13 und insgesamt 439 Blattknoten tatsächlich sehr groß ist. Insbesondere enthält er mehr Blattknoten als es diskrete Werte der Schädigung im Trainingsdatensatz gibt.

Die in diesem Abschnitt erzeugten Modelle erreichen nicht die Vorhersagequalität der Modelle aus Abschnitt 4.5.1. Darüber hinaus weist eine erhebliche Differenz zwischen dem  $R^2$  Wert auf den Trainingsdaten und dem Mittelwert der Kreuzvalidierung auf Überanpassung hin. Eine Verwendung der trainierten Modelle auf Basis von linearer Regression oder Entscheidungsbäumen ist daher insbesondere zur Approximation der Schädigung nicht zu empfehlen. Im folgenden Abschnitt werden dementsprechend andere Verfahren zur Erzeugung von Modellen für die Approximation der Ausgangsgrößen Umformgrad und Schädigung untersucht.

### 4.5.3 Approximation durch AutoML-Verfahren

Die ungenaueren Vorhersagen der linearen Regression im Vergleich mit dem Entscheidungsbaum können auf nicht lineare Zusammenhänge in den Daten hinweisen [XZZ21b]. Entsprechend ist der Einsatz komplexerer Methoden, die solche Beziehungen abbilden können, angezeigt. Darüber hinaus hat die Auswertung des Entscheidungsbaums ergeben, dass eine Optimierung der Hyperparameter eines Modells geboten ist. Um den Hypothesenraum derart zu erkunden, bietet sich AutoML an [DDL+20]. Entsprechende Verfahren können eine Vielzahl von ML-Methoden in Kombination mit einigen Vorverarbeitungsschritten testen und die Hyperparameter der Vorverarbeitungsschritte und des verwendeten ML-Algorithmus effizienter als mit einer Rastersuche optimieren. Eine Implementierung von AutoML, die auf dem verwendeten Python-Modul scikit-learn aufbaut, findet sich in auto-sklearn [FKE+15]. Dabei wird die Modellauswahl selbst mit ML-Methoden angegangen. Für einen begrenzten Zeitraum von jeweils 18 Stunden wurden mithilfe von auto-sklearn Modelle zur Approximation der Ausgangsgrößen Umformgrad und Schädigung erzeugt. Darüber hinaus wurde lediglich festgelegt, dass kein Validierungsdatensatz, sondern Kreuzvalidierung mit fünf Teilmengen einzusetzen ist. Die Verwendung von fünf Teilmengen ist Standard für auto-sklearn und hat im Vergleich zu einer höheren Anzahl den Vorteil, die Laufzeit der Kreuzvalidierung zu verkürzen und so in der zur Verfügung stehenden Zeit mehr Algorithmen und Hyperparameterwerte testen zu können. Zur Vergleich der Modelle, die das AutoML-Verfahren als die besten erzeugten Modelle ansieht, wird wieder eine Kreuzvalidierung mit zehn Teilmengen eingesetzt, die eine präzisere Annäherung an den  $R^2$  Wert auf ungesehenen Daten erreichen kann. Das ist damit zu erklären, dass für die einzelnen Modelle in der Kreuzvalidierung mehr Daten im Training zur Verfügung stehen. Darüber hinaus wird der Mittelwert der Kreuzvalidierung aus mehr Werten berechnet und ist damit stabiler. Die durch AutoML-Verfahren erzeugten Modelle werden nachfolgend beschrieben.

Die durch AutoML erreichten Vorhersagequalitäten sind in Tabelle 4.9 abgebildet. Zur Vorhersage beider Ausgangsgrößen Umformgrad und Schädigung wendet auto-sklearn die GP-Regression (vgl. Abschnitt 2.2.4) an. Für den Umformgrad wird damit ein  $R^2$  Wert von 1 auf den Trainingsdaten und einen Mittelwert in einer Kreuzvalidierung mit zehn Teilmengen von 0,96 erreicht. Das ist eine deutliche Steigerung um 0,12 im Vergleich zur linearen Regression und liegt in derselben Größenordnung wie die Vorhersagequalitäten, welche für die vier anderen Ausgangsgrößen erzielt wurden. Auch für die Ausgangsgröße Schädigung kann das AutoML-Verfahren eine gute Approximation hervorbringen. Hier liegt der  $R^2$  Wert im Training bei 0,99, der Mittelwert der Kreuzvalidierung bei 0,87. Im Vergleich zur linearen Regression mit 0,05 und dem Entscheidungsbaum mit 0,58 ist das eine erhebliche Steigerung.

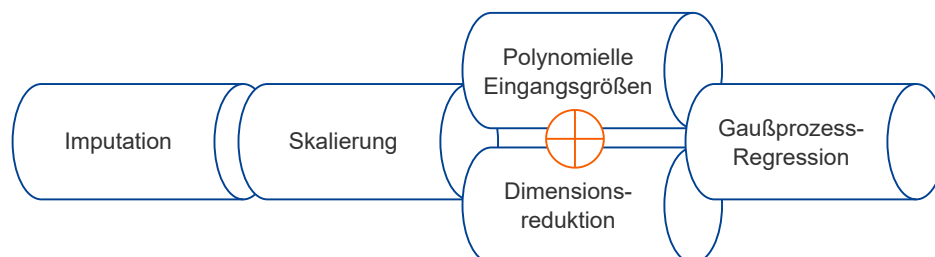
Bei der Anwendung von auto-sklearn werden zahlreiche sogenannte Pipelines erzeugt und getestet. Unter einer Pipeline versteht man eine Folge von Datenverarbeitungsschritten oder Anwendungen von ML-Algorithmen. Auto-sklearn bildet aus ausgewählten Pipelines, die in Kombination eine

Ausgangsgröße	Training	Mittelwert Kreuzvalidierung	Gewählte ML-Methode
Umformgrad	1,00	0,96	Ensemble aus GP-Regressionen
Schädigung	0,99	0,87	Ensemble aus GP-Regressionen

**Tabelle 4.9:** Übersicht über die mit AutoML-Verfahren erreichten  $R^2$  Werte für die Ausgangsgrößen Umformgrad und Schädigung.

gute Approximation ermöglichen, ein Ensemble zur Vorhersage der Ausgangsgröße. Jede von auto-sklearn erstellte Pipeline ist prinzipiell gleich aufgebaut. Die erste Komponente wird als Data Preprocessor bezeichnet. Mehrere Schritte können darin zusammengefasst sein, zum Beispiel Imputation und die Umwandlung von kategorischen Werten in numerische. Auch die Skalierung fällt in diesen Bereich. Anschließend wird ein sogenannter Feature Preprocessor eingesetzt. Im Gegensatz zur vorherigen Komponente, die die Werte der Eingangsgrößen verändert, wird hier die Menge der Eingangsgrößen variiert. Es können dabei einzelne Größen verworfen oder neue gebildet werden. Die letzte Komponente ist entweder ein Klassifikations- oder Regressionsalgorithmus [FKE+15]. Für die Approximationen der Ausgangsgrößen Umformgrad und Schädigung kann eine Untersuchung aller erzeugten Pipelines Aufschluss darüber geben, welche Algorithmen und Datentransformationen sich hinsichtlich der Vorhersagequalität als förderlich erwiesen haben. Es gestaltet sich jedoch schwierig, die von auto-sklearn trainierten Modelle im Detail zu untersuchen. Das liegt darin begründet, dass die gesuchten Informationen über eine Vielzahl unterschiedlicher Funktionsaufrufe verteilt sind. Dabei ist es kaum möglich, die Teilergebnisse einander zuzuordnen und ein möglichst vollständiges Bild zu erhalten, da die identifizierenden Angaben inkonsistent verwendet werden. Eine umfassende manuelle Suche und Zuordnung ist bei hunderten von Pipelines in vertretbarer Zeit nicht durchführbar. Beschränkt man sich auf die Pipelines, aus denen auto-sklearn zur Vorhersage ein Ensemble bildet, ist der zeitliche Aufwand tolerabel.

Die Ensembles für die Ausgangsgrößen Umformgrad und Schädigung bestehen aus jeweils acht, beziehungsweise drei Modellen. Diese sind wie in Abbildung 4.10 dargestellt aufgebaut. Zunächst werden fehlende Werte im Rahmen einer Imputation mit Lageparametern ersetzt. Während des Trainings wird die Wahl dieses Parameters optimiert. Welcher Lageparameter im Detail gewählt wurde, ist nicht weiter von Bedeutung, da in den verwendeten Eingangs- und Ausgangsgrößen keine Werte fehlen. Die Imputation wird also von auto-sklearn als Schritt angegeben, hat aber keine Auswirkungen. Anschließend erfolgt eine Skalierung der Eingangsgrößen, hauptsächlich durch Standardisierung. Lediglich in einem Modell wurde eine andere Methode verwendet. Der dritte Schritt unterscheidet sich je nach Ausgangsgröße. Für den Umformgrad werden polynomielle



**Abbildung 4.10:** Folge von Verarbeitungsschritten der einzelnen AutoML-Modelle, aus denen ein Ensemble gebildet wird.

Kombinationen der Eingangsgrößen bis zum Grad 2 gebildet, auf denen später das Modell trainiert wird. Dadurch vervielfacht sich die Anzahl der Eingangsgrößen. Bei den Modellen für die Schädigung wird dagegen die Anzahl verwendeter Größen durch eine Dimensionsreduktion verringert. Unabhängig von der Ausgangsgröße wird zuletzt eine GP-Regression, wie sie in Abschnitt 2.2.4 beschrieben ist, durchgeführt.

### 4.5.4 Einfluss kategorischer oder numerischer Kodierung

Ein manueller Vorverarbeitungsschritt, der vor der Visualisierung und Erzeugung von Modellen durchgeführt wird, ist die Umwandlung von kategorischen in numerische Werte. In Abschnitt 4.3 wurde die Bedeutung dieses Schritts durch theoretische Überlegungen hergeleitet. Da auch auto-sklearn diese Umformung durchführen kann, ist es von Interesse, wie sich die entstehenden Ensembles zueinander verhalten, wenn jeweils welche mit und ohne diese manuelle Vorverarbeitung erstellt werden. Es wurde entsprechend ein zusätzliches Modell von auto-sklearn für die Ausgangsgröße Umformgrad trainiert.

Dieses Ensemble erreicht auf dem Trainingsdatensatz einen  $R^2$  Wert von 1 und in der Kreuzvalidierung einen Mittelwert von 0,96. Diese Werte sind, von der Rundung abgesehen, mit denen identisch, die zuvor mit rein numerischen Werten erzielt wurden. Das mit kategorischen Werten erzeugte Ensemble besteht erneut aus acht Pipelines, wobei nur drei keine Umwandlung von kategorischen in numerische Werte beinhalten. Da die GP-Regression lediglich numerische Werte verarbeiten kann, wird sich in diesen Fällen auch anderer ML-Algorithmen bedient. In einem Modell kommt ein k-nächste Nachbarn Ansatz zum Einsatz, bei den übrigen handelt es sich um Ensembles von Entscheidungsbäumen. Das heißt, dass einzelne Modelle des Gesamtensembles ihrerseits Ensembles sind. Eine mögliche Erklärung dafür ist, dass bei kategorischen Werten mehr Pipelines mit diversen ML-Verfahren notwendig sind, um eine gleichwertige Approximation zu erreichen. Die unterschiedlichen Algorithmen, die im Falle von kategorischen Werten zum Einsatz kommen, sowie die Tatsache, dass ein Ensemble von Ensembles gebildet wird, bedeuten eine Steigerung der Komplexität. Insbesondere sind mehr und unterschiedliche Hyperparameter zu optimieren. Dieser zusätzliche Aufwand spiegelt sich jedoch nicht in einer deutlich besseren Approximation wieder. Da bei der Umwandlung in numerische Werte, zumindest im vorliegenden Fall, keine Informationen verloren gehen, ist die manuelle Durchführung dieses Vorverarbeitungsschritts zu empfehlen. Sie ermöglicht es, ohne erhebliche Einbußen bei der Vorhersagequalität, lediglich GP-Regression zu verwenden.

### 4.5.5 Einfluss von Ausreißern

Wie in Abschnitt 4.2 festgestellt, gibt es einige Ausreißerdatenpunkte in den Daten. Zum Zwecke der in Abschnitt 4.4 Visualisierung wurden diese entfernt, um die eigentlichen Verteilungen besser darstellen zu können. Auch für die bisher trainierten Modelle wurden Daten ohne Ausreißer verwendet. Dadurch verringert sich jedoch die Menge der im Training zur Verfügung stehenden Daten. Dabei ist zu beachten, dass sich mit mehr Daten prinzipiell bessere Ergebnisse erzielen lassen [BH21; Mur22]. Daher wurden die Ausreißerdatenpunkte mit ML-Methoden näher untersucht.

### Modelle mit und ohne Ausreißerdatenpunkte

Eines der wichtigsten Qualitätsmerkmale bei dem hier zugrundeliegenden Napf-Rückwärts-Fließpressen ist die Ausgangsgröße Napfbodendicke. Diese enthält im verwendeten Datensatz unglaubliche Werte, die laut Fachexperte auf Fehler der Simulation zurückzuführen sind und nicht sinnvoll mit Lageparametern der Verteilung ersetzt werden können. Zur Analyse der Auswirkungen der Ausreißer auf Modelle und Vorhersagen werden zwei AutoML-Modelle zur Vorhersage der Ausgangsgröße Napfbodendicke trainiert. Dabei werden für eines dieser Modelle die Ausreißerdatenpunkte aus den Trainingsdaten entfernt, für das andere nicht.

Trainiert man ein AutoML-Modell zur Vorhersage der Napfbodendicke auf allen Daten – inklusive der Ausreißerdatenpunkte – so wird im Training ein  $R^2$  Wert von 1 und in einer Kreuzvalidierung mit zehn Teilmengen ein Mittelwert von ebenfalls 1 erreicht. Diese Werte unterscheiden sich lediglich in der Größenordnung  $10^{-3}$  von denen, die ohne Ausreißerdatenpunkte erzielt werden. Das Modell mit Ausreißern erreicht geringfügig niedrigere  $R^2$  Werte auf dem Trainingsdatensatz, in der Kreuzvalidierung ist das Modell mit Ausreißern aber besser und robuster als das andere Modell. Die Standardabweichung über die im Rahmen der Kreuzvalidierung berechneten Werte beträgt ohne Ausreißer  $1,20 \cdot 10^{-2}$ , mit Ausreißern  $8,11 \cdot 10^{-4}$ . Für beide Modelle ist die Vorhersagequalität jedoch als sehr gut zu bewerten. Es fällt darüber hinaus auf, dass das Modell mit Ausreißern diese auch auf dem Testdatensatz mit hoher Genauigkeit vorhersagen kann. Hier wurde also der Fehler der Simulation von der ML-Methode gelernt.

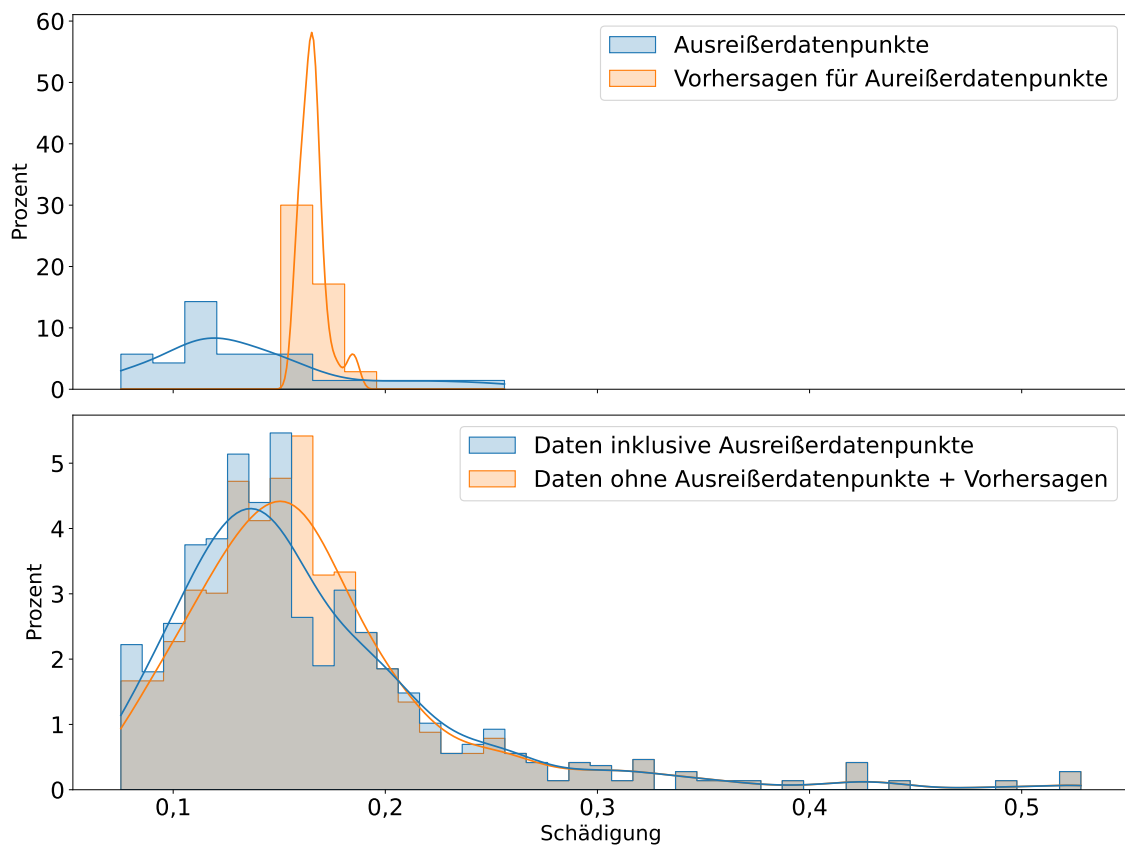
Betrachtet man den Aufbau beider Modelle, werden größere Unterschiede offenbar. Das Modell, welches ohne Ausreißerdatenpunkte trainiert wurde, ist ein Ensemble aus 31 Ensembles von Entscheidungsbäumen. Bedenkt man, dass die Napfbodendicke ohne Ausreißer allein anhand der Werte einer Eingangsgröße perfekt vorhergesagt werden kann, ist diese enorme Komplexität überraschend und unnötig. In einem Modell mit vielen Entscheidungsbäumen können auch viele Datenpunkte direkt in Blattknoten der Bäume kodiert sein. Hier ist also definitiv die Möglichkeit für Überanpassung gegeben. Das wird noch bestärkt durch den Zustand, dass ohne Ausreißer lediglich fünf diskrete Werte in der Ausgangsgröße Napfbodendicke enthalten sind. Das mit Ausreißerdatenpunkten trainierte Modell basiert dagegen auf fünf GP-Regressionen und ist damit deutlich simpler. Die Vorverarbeitungsschritte wurden nicht verglichen.

Ausreißer befinden sich nur in Ausgangsgrößen, die schon anhand einfacher, durch Visualisierung identifizierter Zusammenhänge vorhergesagt werden können. Dass sie jedoch laut Fachexperte auf Fehlern der Simulation beruhen, stellt auch die Glaubwürdigkeit der Werte für andere Größen dieser Datenpunkte infrage. Daher wurden für die Ausgangsgrößen Umformgrad und Schädigung weitere AutoML-Modelle auf Daten mit Ausreißerdatenpunkten trainiert. Für den Umformgrad erreicht ein solches Modell einen  $R^2$  Wert von 1 auf den Trainingsdaten. Der Mittelwert der Kreuzvalidierung beträgt 0,97 und ist damit etwa um 0,01 besser als ohne Ausreißerdatenpunkte. Der Aufbau beider Modelle ist ähnlich: es handelt sich jeweils um Ensembles aus sechs und acht GP-Regressionen für Training mit und ohne Ausreißerdatenpunkte. Auch die Vorverarbeitung beider Modelle ist größtenteils übereinstimmend. In den Pipelines für beide Ensembles werden polynomielle Eingangsgrößen gebildet und vornehmlich durch Standardisierung skaliert. Für die Ausgangsgröße Schädigung wird mit Ausreißerdatenpunkten ein  $R^2$  Wert von 0,98 im Training und 0,87 im Mittel in der Kreuzvalidierung erreicht. Lediglich der Wert im Training ist 0,01 schlechter als zuvor. Ansonsten sind die  $R^2$  Werte für beide Modelle identisch. Auch die Ensembles sind für diese Größe sehr ähnlich. Sie bestehen je aus drei GP-Regressionen. Im Rahmen der Vorverarbeitung

wird in beiden Fällen lediglich eine Skalierung der Eingangsgrößen durchgeführt. Zusammenfassend lässt sich folgendes sagen. Ob mit oder ohne Ausreißerdatenpunkte trainiert: ein AutoML-Modell kann in beiden Fällen die Ausgangsgrößen Umformgrad und Schädigung approximieren. Dabei werden sehr ähnliche Vorhersagequalitäten erreicht. Auch der Aufbau der Modelle ist annähernd übereinstimmend. Daraus lässt sich nicht eindeutig ableiten, ob die Ausreißerdatenpunkte im Training verwendet werden sollten oder nicht.

### Vorhersagen für Ausreißerdatenpunkte

Die Vorhersagen für Ausreißerdatenpunkte durch ein Modell, das ohne Ausreißerdatenpunkte trainiert wird, können mit den Werten in den Daten verglichen werden. Dabei kann überprüft werden, ob die Vorhersagen oder die Werte der Ausreißerdatenpunkte besser zu der Verteilung der übrigen Werte einer Größe passen. Hier wird die Ausgangsgröße Schädigung untersucht, die keine Ausreißer beinhaltet. Jedoch ist auch die Korrektheit der Werte dieser Größe in den Ausreißerdatenpunkten fraglich, da die Simulation dieser Datenpunkte laut Fachexperte fehlerhaft ist.



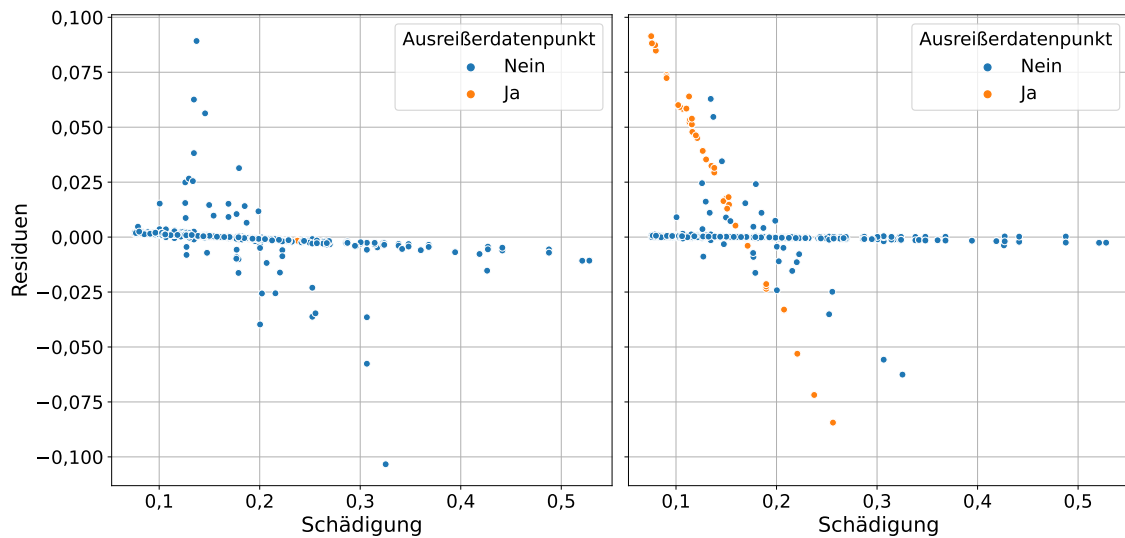
**Abbildung 4.11:** Vergleich der Werte der Ausreißerdatenpunkte mit entsprechenden Vorhersagen für die Ausgangsgröße Schädigung. Im oberen Teil sind lediglich die Ausreißerdatenpunkte dargestellt, im unteren Teil alle Datenpunkte.

Ein Vergleich zwischen Vorhersagen und den Werten in den Trainingsdaten ist für die Ausgangsgröße Schädigung in Abbildung 4.11 visualisiert. Es handelt sich bei der Darstellungsform um Histogramme in Kombination mit einer durch Kerndichteschätzung errechneten Dichtefunktion. Im oberen Teil der Grafik sind nur die Werte der Ausgangsgröße Schädigung abgebildet, die den Ausreißerdatenpunkten entsprechen. In Blau sind die Werte aus den Daten, in Orange entsprechende Vorhersagen dargestellt. Beide Verteilungen unterscheiden sich stark. Im unteren Teil der Grafik sind zusätzlich zu den oben dargestellten Vorhersagen und Werten die Werte der Datenpunkte abgebildet, die keine Ausreißer enthalten. Die Verteilung der Schädigung in den Daten inklusive Ausreißerdatenpunkten ist in blau dargestellt. Ihr in Orange gegenübergestellt ist die Verteilung derselben Größe ohne die tatsächlichen Werte der Ausreißerdatenpunkte, dafür mit Vorhersagen für ebendiese. Da die meisten Datenpunkte keine Ausreißer beinhalten, ähneln sich diese Verteilungen. Im Bereich, in dem die Werte, beziehungsweise Vorhersagen für die Ausreißerdatenpunkte liegen, sind dagegen Unterschiede auszumachen.

Eine Frage, die sich bei den Unterschieden in der Visualisierung von Ausreißerdatenpunkten und Vorhersagen ergibt, ist, welche Werte besser zur übrigen Verteilung der Ausgangsgröße passen. Das ist anhand der Abbildung nicht eindeutig festzustellen. Zur Beantwortung werden statistische Tests herangezogen. Zunächst wird mit dem Wilcoxon-Vorzeichen-Rang-Test überprüft, ob die Werte in den Daten und die Vorhersagen der gleichen Verteilung entstammen [FHK+16]. Die Nullhypothese  $H_0$  sagt dabei aus, dass die Werte aus derselben Verteilung kommen. Der ermittelte  $p$ -Wert liegt mit  $2,10 \cdot 10^{-9}$  weit unter dem gebräuchlichen Signifikanzniveau von 0,05.  $H_0$  ist damit abzulehnen, die Werte resultieren nicht aus derselben Verteilung. Mit dem Cramér-von-Mises-Test wird anschließend überprüft, ob sich die Verteilung der Werte für die Ausreißerdatenpunkte - aus den Daten wie auch vorhergesagt - von der Verteilung der übrigen Werte unterscheidet. Hierbei besagt die Nullhypothese  $H_0$ , dass sich die Verteilungen nicht unterscheiden. Für die Werte der Ausreißerdatenpunkte in den Daten ergibt sich ein  $p$ -Wert von  $1,42 \cdot 10^{-8}$ , für die Vorhersagen von  $4,75 \cdot 10^{-10}$ . In beiden Fällen ist  $H_0$  bei einem Signifikanzniveau von 0,05 abzulehnen. Das heißt, dass beide Wertemengen für Ausreißerdatenpunkte nicht der Verteilung der übrigen Werte entsprechen. Das könnte daran liegen, dass die Wertemengen zu klein sind oder dass sie nicht zufällig aus den verfügbaren Werten ausgewählt wurden. Schlussendlich lassen die statistischen Tests keinen Schluss darauf zu, ob die Datenpunkte, die Ausreißer enthalten, im Training verwendet werden sollten oder nicht.

Zuletzt wurden zum Vergleich der Vorhersagen der Modelle, die mit und ohne Ausreißerdatenpunkte trainiert wurden, Residualplots angefertigt. Diese beschreiben die Approximationen der Trainingsdaten und sind in Abbildung 4.12 dargestellt. Ausreißerdatenpunkte sind durch ihre orange Färbung gekennzeichnet. In der linken Abbildung für das Modell, welches mit Ausreißerdatenpunkten trainiert wurde, zeigen die Residuen einen leicht abfallenden, linearen Verlauf über die Werte der Ausgangsgröße. Auch sind einige Punkte mit deutlich erhöhtem Abstand zum Wert 0 der y-Skala zu sehen. Diese treten häufiger für niedrige Werte der Schädigung auf. Der rechte Residualplot für das Modell, das ohne Ausreißerdatenpunkte trainiert wurde, zeigt keine so starke Steigung in dem linearen Verhalten der Residuen. Die Werte sind näher um den Wert 0 der Residuen verteilt. Doch auch in dieser Abbildung lassen sich Punkte mit deutlichem Abstand zu diesem erkennen. Wie zuvor treten diese häufiger für niedrige Schädigungen auf. Hinzu kommt ein klar erkennbares Muster hinsichtlich der Ausreißerdatenpunkte. Während diese in der linken Grafik kaum zwischen den übrigen Datenpunkten auszumachen sind, formen sie in der rechten beinahe eine Gerade mit starker negativer Steigung. Es ist zu schließen, dass das Modell, welches ohne Ausreißerdatenpunkte trainiert wurde, die blauen Werte etwas besser approximieren kann wie das

## 4 Ergebnisse der explorativen Datenanalyse



**Abbildung 4.12:** Residualplots für die Vorhersagen der Ausgangsgröße Schädigung von Modellen, die mit (links) und ohne Ausreißerdatenpunkte (rechts) trainiert werden.

andere Modell. Jedoch bilden die Residuen für die Ausreißerdatenpunkte ein deutliches Muster. Das kann bedeuten, dass diese einem anderen Zusammenhang unterliegen, den das Modell nicht erlernt hat. Da die objektive Richtigkeit der Werte der Ausreißerdatenpunkte im Rahmen dieser Arbeit jedoch nicht überprüft werden kann, ist auch nach Auswertung der Residualplots unklar, ob ein Modell mit Ausreißerdatenpunkten trainiert werden sollte oder nicht.

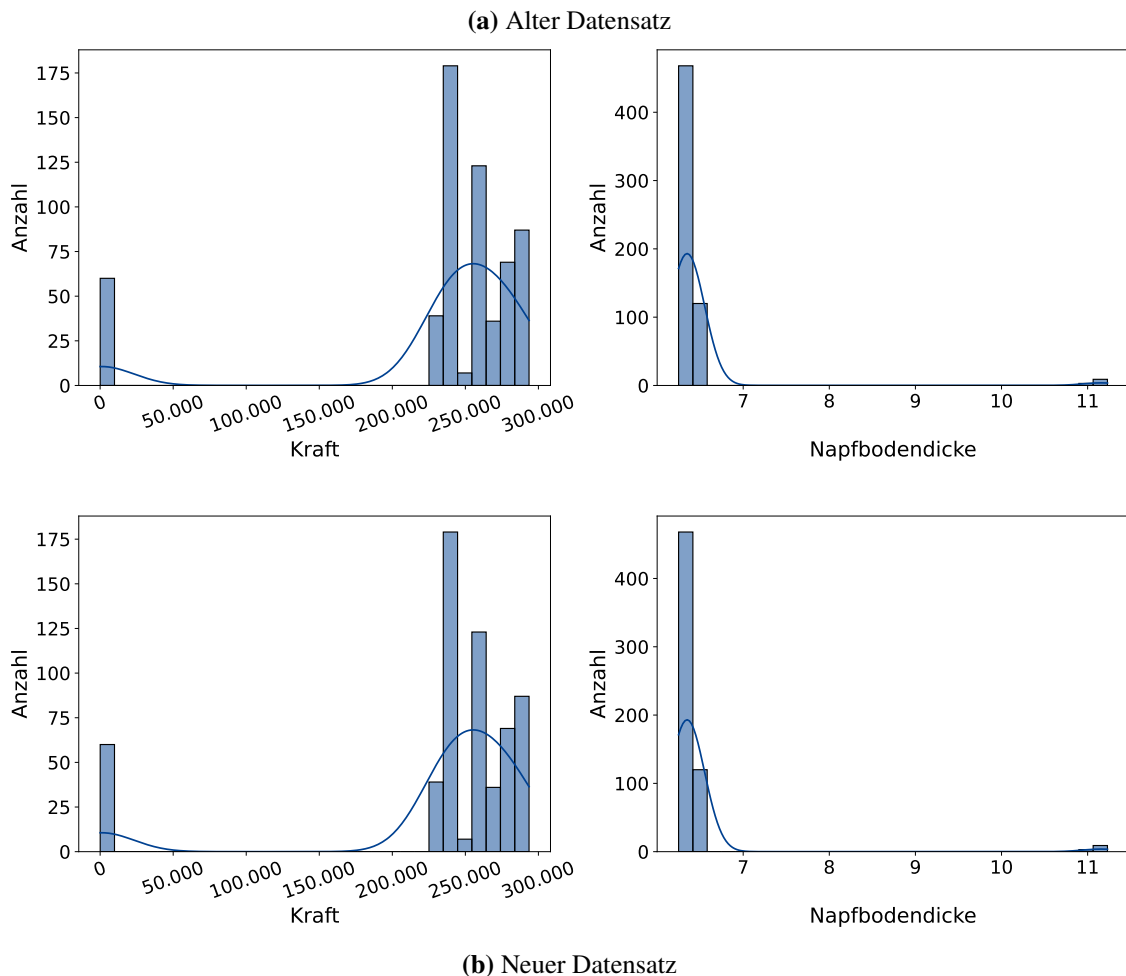
### 4.5.6 Erneute Simulation einer Teilmenge der Daten

Die Ausreißer in den Daten lassen sich laut Fachexperte durch Fehler der Simulationsoftware begründen. Eine Fehlerquelle könnte dabei die numerische Natur dieser Simulationen sein [RSK20]. Treten diese Fehler bei einer erneuten Simulation der Daten nicht auf, so erübrigt sich die Behandlung der Ausreißer. In diesem Fall können stattdessen die entsprechenden Werte der neuerlichen Simulation verwendet werden.

Zunächst wurden 600 der insgesamt 1 080 Datenpunkte neu simuliert. Hintergrund ist die Hypothese, dass Ausreißer unabhängig von den Eingangsgrößen auftreten. Zum Vergleich mit den neuen Daten werden die 600 alten Datenpunkte herangezogen, welche die gleichen Werte für die Eingangsgrößen aufweisen. Die entsprechenden Histogramme sind in Abbildung 4.13 für die zwei von Ausreißern betroffenen Ausgangsgrößen Kraft und Napfbodendicke dargestellt. Im oberen Teil der Grafik, Abbildung 4.13a sind die alten Daten visualisiert. Im unteren Bereich der Darstellung, Abbildung 4.13b, sind die neu simulierten Datenpunkte zu sehen. Dabei sind keine wesentlichen Unterschiede zwischen den Verteilungen der beiden Teildatensätze festzustellen. Beide enthalten auch etwa gleich viele Ausreißer, deren Werte in denselben Intervallen liegen.

Die Ergebnisse beider Simulationsreihen enthalten nicht die exakt gleichen Werte. Allerdings unterscheiden sich auch nur 20 Datenpunkte um mehr als  $10^{-4}$ . Die maximale Differenz zwischen neuen und alten Werten der beiden Größen liegt dabei in der Größenordnung von  $10^5$ . In den neu simulierten, 600 Datenpunkten befinden sich 60 Ausreißer der Ausgangsgröße Kraft und





**Abbildung 4.13:** Histogramme der von Ausreißern betroffenen Ausgangsgrößen Kraft und Napfbodendicke in den alten (oben) und den neu simulierten (unten) Daten.

12 für die Napfbodendicke. Insgesamt beinhalten die neuen Daten 72 Ausreißerdatenpunkte. In den entsprechenden alten Daten mit denselben Werten in den Eingangsgrößen sind ebenfalls 72 Ausreißerdatenpunkte enthalten. Für die Ausgangsgröße Kraft ergab eine nähere Untersuchung, dass ein einzelner Ausreißer in den alten Daten in den neuen Daten einen Wert enthält, der im definierten Wertebereich liegt. Zusätzlich ist dafür ein weiterer, neuer Ausreißer hinzugekommen. Bezüglich der Napfbodendicke sind die neuen und alten Daten identisch. Nimmt man folglich aus den alten und neuen Daten die jeweils „besten“ Datenpunkte, so hat man insgesamt einen Ausreißer weniger, 71 statt 72. Insofern hält sich der Mehrwert der erneuten Simulation in Grenzen.

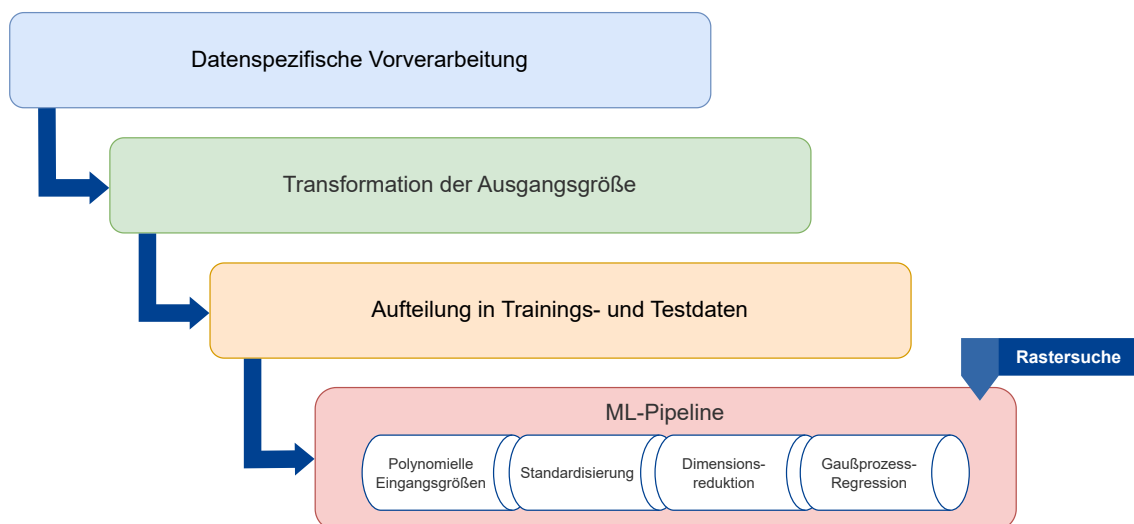
Darüber hinaus fällt auf, dass bis auf zwei Datenpunkte die Ausreißer beider Simulationsdurchläufe jeweils für dieselben Werte der Eingangsgrößen vorkommen. Geht man davon aus, dass bei fixer Anzahl der Fehler der Simulationssoftware diese zufällig gleichverteilt über die Werte der Eingangsgrößen auftreten, so liegt die Wahrscheinlichkeit, diese beiden Teildatensätze zu erhalten, in der Größenordnung von  $10^{-94}$ . Es ist daher mit sehr hoher Wahrscheinlichkeit davon auszugehen, dass die Ausreißer abhängig von den Eingangsgrößen sind. Das erklärt auch, warum ein Modell

zur Approximation der Napfbodendicke in Abschnitt 4.5.5 die Ausreißerwerte präzise ausgeben kann. Liegt dem Auftreten von Ausreißern eine Systematik zugrunde, so kann diese durch ein ML-Modell erlernt werden. Auch das Auftreten eines klar erkennbaren Musters im Residualplot der Abbildung 4.12 kann darin begründet liegen. Ein Teil dieses Zusammenhangs ist möglicherweise bei der Untersuchung der Korrelationen aufgedeckt worden. Für das Auftreten von Ausreißern in der Napfbodendicke wurde als notwendige Bedingung identifiziert, dass die Werte des Zyklus bei *High* und die Werkstückgeometrie bei 1 liegen müssen. Allerdings handelt es sich dabei nicht um eine hinreichende Bedingung.

## 5 Erarbeitete Vorgehensweise

Ziel dieser Arbeit ist es, eine Methodik zu entwickeln, mit der die Ausgangsgrößen einer Kaltmasivumformsimulation approximiert werden können. Zu diesem Zweck wird ein datengetriebenes Surrogatmodell erzeugt. Sechs Ausgangsgrößen hat ein Fachexperte ausgewählt, für die eine Approximation von besonderem Interesse ist (vgl. Abschnitt 4.1). Für vier davon ist wie in Abschnitt 4.5 beschrieben eine Vorhersage mit einfachen Methoden auf Basis von Korrelationen möglich. Für die Ausgangsgrößen Umformgrad und Schädigung ist das nicht der Fall. Daher zielt die Entwicklung der Vorgehensweise im Rahmen dieser Arbeit auf die Approximation dieser Größen ab. Die Vorgehensweise stützt sich auf die Erkenntnisse der Literaturrecherche aus Kapitel 3 und der explorativen Datenanalyse, die in Kapitel 4 beschrieben ist. Es wird im Folgenden zunächst eine Übersicht über den Aufbau des Verfahrens gegeben. Daran schließt sich eine Erläuterung der einzelnen Schritte an.

In Abbildung 5.1 sind die einzelnen Schritte des erarbeiteten Verfahrens dargestellt. Es erfolgt zunächst eine datenspezifische Vorverarbeitung, um die Daten für die weiteren Schritte vorzubereiten. Diese muss individuell dem Datenbestand angepasst werden. Je nach Verteilung der Ausgangsgröße wird diese transformiert, um möglichst normalverteilt zu sein. Eine solche Verteilung kann die Vorhersage durch ein Surrogatmodell begünstigen [FHK+16]. Anschließend erfolgt eine Aufteilung der Daten in Trainings- und Testdatensatz. Trainiert man ein Modell für den Produktiveinsatz, ist das nicht zwingend notwendig. Es erlaubt jedoch, die Vorhersagequalität des Surrogatmodells auf ungesehenen Daten zu beurteilen. Ist sie zufriedenstellend, kann das Modell im Anschluss auf



**Abbildung 5.1:** Schematische Darstellung der im Rahmen dieser Arbeit entwickelten Vorgehensweise.

allen zur Verfügung stehenden Daten trainiert werden. Zuletzt wird eine ML-Pipeline gebildet und ihre Hyperparameter mittels einer Rastersuche optimiert. Die Pipeline besteht aus der Erzeugung polynomieller Eingangsgrößen, der Standardisierung, eines Verfahrens zur Dimensionsreduktion und der Gaußprozess-Regression. Die Wahl des ML-Algorithmus fällt auf die GP-Regression, da sie bei der Anwendung von AutoML-Verfahren gute Vorhersagen erreicht. Darüber hinaus verwenden auch andere Autoren im Bereich der Fertigungstechnik Kernel Methoden, zu denen die GP-Regression zählt [BLFS16; CCA+15; XZZ21b]. Die Pipeline kann nach dem Training zur Vorhersage einzelner Ausgangsgrößen verwendet werden. Sind mehrere Größen zu approximieren, so können dazu weitere Pipelines gemäß dem beschriebenen Aufbau trainiert werden. Zur Approximation der Ausgangsgrößen Napfbodendicke, Wandstärke, Temperatur und Kraft kann statt des hier beschriebenen Vorgehens ein einfaches Verfahren auf Basis von Korrelationen angewendet werden. Dabei muss lediglich die datenspezifische Vorverarbeitung dieser Vorgehensweise durchgeführt werden.

In den folgenden Abschnitten werden die einzelnen Schritte, welche die erarbeitete Vorgehensweise ausmachen, beschrieben. Ihre Ausgestaltung wird anhand von verwandten Arbeiten oder Ergebnissen der Voruntersuchung begründet. Teilweise erfolgt eine Abwägung mehrerer Alternativen gegeneinander.

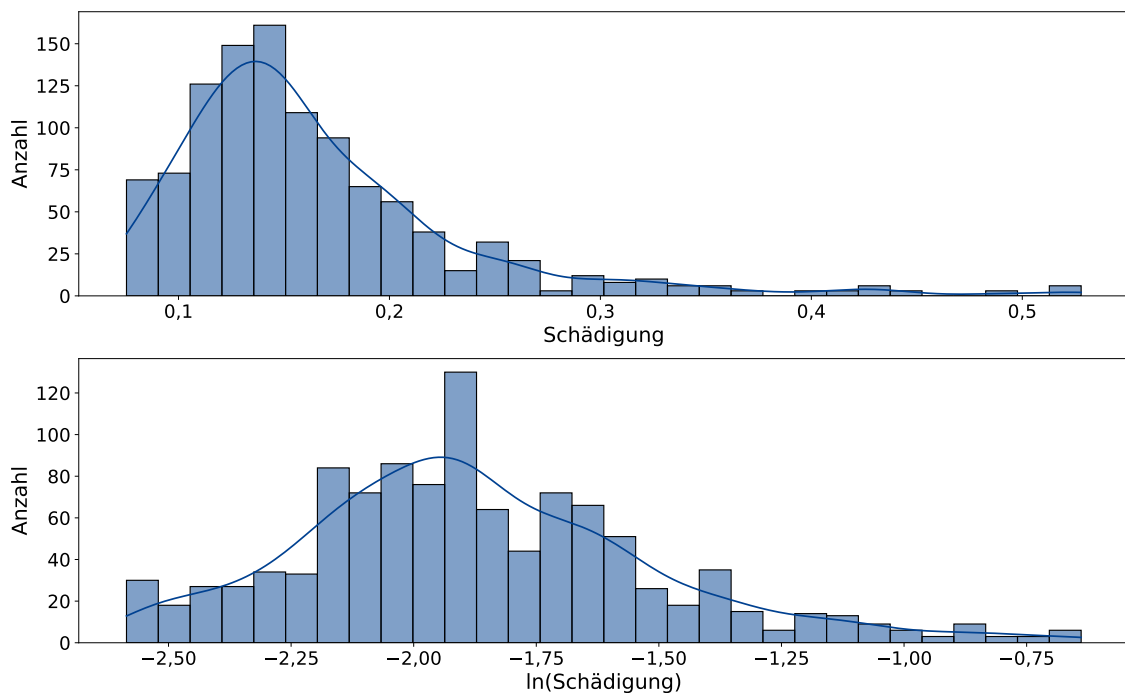
### 5.1 Datenspezifische Vorverarbeitung

Zunächst ist eine datenspezifische Vorverarbeitung durchzuführen. Während einzelne Autoren die Bedeutung dieses Schritts infrage stellen [FBS15], wird er von anderen als wesentlich für den Erfolg einer Anwendung von ML bezeichnet [DB21; KBT11]. Hier wurden im Rahmen einer ausgedehnten Voruntersuchung acht Verarbeitungsschritte identifiziert und umgesetzt, die die Daten in eine Form bringen, die das Training eines Surrogatmodells begünstigt. Diese sind in Abschnitt 4.3 beschrieben. Die letzten beiden – Umwandlung von kategorischen Werten in numerische und Entfernen von Ausreißerdatenpunkten – sind besonders relevant für eine Anwendung der Vorgehensweise auf anderen Daten als den getesteten. Diese zwei werden im Folgenden noch einmal beleuchtet.

Die in Abschnitt 2.2.4 beschriebene GP-Regression, die im Rahmen der Vorgehensweise zum Einsatz kommt, kann lediglich mit metrischen Größen durchgeführt werden. Daher ist es eine Voraussetzung für die Anwendung der entwickelten Methodik, dass ausschließlich metrische Werte in den Daten enthalten sind. Im Allgemeinen schränkt es den Hypothesenraum ein, wenn sich kategorische Werte in den Daten befinden. Die einfachste Methode, diese Einschränkung zu umgehen, besteht in der Umwandlung von kategorischen in numerische Werte [JCL16]. Insbesondere wenn dabei keine Informationen verloren gehen, ist diese Herangehensweise angezeigt und wird daher in der erarbeiteten Vorgehensweise eingesetzt. Ein weiterer Verarbeitungsschritt behandelt Ausreißerdatenpunkte. Diese wurden in Kapitel 4 ausgiebig untersucht, wovon sich aber keine Maßnahmen zur Handhabung ableiten lassen. Häufig werden Ausreißerdatenpunkte verworfen [GE03; KBT11]. Da diese Datenpunkte im vorliegenden Fall falsche Werte enthalten und sich keine bessere Vorhersagequalität der ML-Modelle ergibt, wenn sie beibehalten werden, werden sie aus dem Datensatz entfernt.

## 5.2 Transformation der Ausgangsgröße

Nach der Vorverarbeitung findet eine Transformation der Ausgangsgröße statt. Verfahren auf Basis von GP liegt die Annahme zugrunde, dass jede Teilmenge von Eingangs- und Ausgangsgrößen gemeinsam normalverteilt ist. Prinzipiell empfiehlt es sich, möglichst normal verteilte Größen zu approximieren [FHK+16]. Die Verteilung einer anzunähernden Größe  $x$  kann mithilfe eines Histogramms überprüft werden. Ist dabei eine schiefe Verteilung zu erkennen, kann eine Transformation wie  $e^x$  oder  $\ln x$  Abhilfe schaffen. Vorliegend betrifft das hauptsächlich die Ausgangsgröße Schädigung, für die bisher lediglich durch ein AutoML-Verfahren eine Approximation hoher Güte erreicht worden ist. Ihre Verteilung ist rechtsschief und wird durch einen Logarithmus in annähernde Normalverteilung gebracht. Die Histogramme in Abbildung 5.2 zeigen die Verteilung der Ausgangsgröße vor und nach der Anpassung. Für die Ausgangsgröße Umformgrad (vgl. Abbildung 4.4) ist eine solche Transformation nicht zielführend. Die Verteilung dieser Größe ist weder links- noch rechtsschief und wird durch eine Transformation, wie sie für die Schädigung durchgeführt wird, nicht wesentlich verändert. Es ist aber auch kein nachteiliger Effekt zu erwarten, wenn die Transformation trotzdem angewendet wird. Vor der Ausgabe der Vorhersage muss die Transformation rückgängig gemacht werden. Durch die mehrfache Umwandlung von Größen verlieren die Werte an Präzision, jedoch bewegen sich diese Veränderungen für die Schädigung und den Umformgrad in der Größenordnung von  $10^{-16}$ .



**Abbildung 5.2:** Histogramme der Ausgangsgröße Schädigung vor und nach einer logarithmischen Transformation.

### 5.3 Aufteilung in Trainings- und Testdaten

Als letzter Schritt vor dem Training eines ML-Modells wird ein Testdatensatz von den Gesamtdaten abgespalten, auf dem die Vorhersagequalität geprüft werden kann. Wie die Untersuchung verwandter Arbeiten in Kapitel 3 ergab, ist es üblich, 12 % bis 30 % der Daten dafür zu verwenden [PGK+23; XZZ21b]. Der hier verwendete Datensatz aus 162 von insgesamt 1 080 Datenpunkten liegt mit 15 % im unteren Bereich dieses Intervalls. Das ist der geringen Menge an verfügbaren Daten geschuldet. Ein geringerer Prozentsatz erlaubt es, mehr Daten im Training zu verwenden. Das kann es ermöglichen, bessere Modelle zu trainieren [Alb22; BH21]. Insbesondere sinkt die Wahrscheinlichkeit für Überanpassung [Mur22] und es können mehr Eingangsgrößen verwendet werden [VGG21].

### 5.4 ML-Pipeline

Zuletzt wird eine ML-Pipeline angewendet. Diese ist in Abbildung 5.3 dargestellt. Sie besteht aus den vier Schritten der Erzeugung polynomieller Eingangsgrößen, der Standardisierung, einer Form der Dimensionsreduktion und der GP-Regression. Ihre Hyperparameter werden im Rahmen einer Rastersuche optimiert. Die Ausgestaltung der Pipeline wird nachfolgend diskutiert.

#### Polynomielle Eingangsgrößen

Der erste Schritt der Pipeline beschäftigt sich mit der Menge an Eingangsgrößen. In den Daten sind davon sieben enthalten. Sie ähneln denen, die andere Autoren erfolgreich im Bereich der Umformtechnik verwenden, um Surrogatmodelle zu trainieren [PGK+23]. Die Wahl der Eingangsgrößen hat erheblichen Einfluss auf die Vorhersagequalität eines Modells. Mitunter lassen sich mit mehr Eingangsgrößen bessere Ergebnisse erzielen [XZZ21b]. Zu diesem Zweck können weitere Größen aus bestehenden berechnet werden, beispielsweise durch Multiplikation [KSS16; PS09]. In einem ersten Schritt werden daher polynomielle Eingangsgrößen bis zum Grad 2 erzeugt. Ein Beispiel dafür ist in Abschnitt 2.2.4 dargestellt. Bei der Verwendung von AutoML wurden polynomielle Eingangsgrößen zwar nicht in vielen Pipelines eingesetzt. Durchschnittlich hatten sie jedoch in solchen, in denen sie zu Einsatz kamen, einen positiven Einfluss auf die Vorhersagequalität. Im Anschluss an diesen Schritt liegen 37 Eingangsgrößen vor.

#### Standardisierung

In einem zweiten Schritt werden die Eingangsgrößen standardisiert. Das wurde bereits für das Training von ersten Modellen in Kapitel 4 umgesetzt und auch durch das AutoML-Verfahren als Vorverarbeitungsschritt gewählt. Allgemein ist es eine häufig durchgeführte Maßnahme bei der Erzeugung datengetriebener Surrogatmodelle [Mur22]. Dabei wird von jedem Wert einer Größe der Mittelwert der Größe subtrahiert und anschließend durch die Standardabweichung geteilt, wie in Gleichung (2.1) angegeben. Das dient dem Ausgleich des Einflusses einzelner Eingangsgrößen auf die Vorhersage. Jedes Verfahren, das Distanzen berechnet, wird von der Skalierung der Daten beeinflusst. Zu solchen Verfahren gehört auch die Dimensionsreduktion, die im Rahmen



**Abbildung 5.3:** ML-Pipeline, die im Rahmen der erarbeiteten Vorgehensweise zum Einsatz kommt.

der erarbeiteten Vorgehensweise durchgeführt wird. Aus diesem Grund werden die Daten vor der Dimensionsreduktion standardisiert. Darüber hinaus liegt der GP-Regression die Annahme zugrunde, dass jede Teilmenge der beteiligten Größen eine multivariate Normalverteilung darstellt. Für den unbekanntem Mittelwert dieser Verteilung wird üblicherweise  $\mathbf{0}$  gewählt [Mur22; Ras06]. Unter Verwendung standardisierter Größen ist das korrekt.

### Dimensionsreduktion

Werden die Größen wie beschrieben konstruiert, ist im Anschluss eine Reduktion ihrer Anzahl angezeigt [PS09]. Das dient der Verbesserung der Vorhersagen, der schnelleren Berechnung dieser sowie der Verringerung des zeitlichen Aufwands für das Training [GFW+19]. Auch der Erklärbarkeit kann es zuträglich sein, wenn ein Modell auf weniger Eingangsgrößen aufbaut [GE03; HYDF16]. Zur Auswahl von Größen werden prinzipiell zwei Ansätze unterschieden. Ein Filter bewertet Eingangsgrößen anhand einer Metrik, zum Beispiel ihrer Korrelation mit der Ausgangsgröße. Anhand dieser können die Eingangsgrößen geordnet und so die  $n$  höchstbewerteten gewählt werden [HYDF16]. Diese Herangehensweise zeichnet sich hauptsächlich durch ihre geringe Laufzeit aus. Jedoch werden dabei alle Größen unabhängig voneinander bewertet, was möglicherweise suboptimal ist. Eine Größe kann allein nutzlos, aber in Kombination mit anderen ausschlaggebend für die Vorhersage sein [GE03]. Dieses Problem besteht beim Einsatz von sogenannten Wrappern nicht. Hierbei werden auf Teilmengen der Eingangsgrößen ML-Modelle trainiert und die Vorhersagequalität dieser verglichen. Da letztendlich die Genauigkeit der Vorhersage, in Kombination mit dem spezifischen Lernalgorithmus und dem Datensatz, zählt, können mit Wrappern präzisere Ergebnisse als mit Filtern erzielt werden. Das Training der Modelle ist jedoch zeitaufwendig. Mit einem Wrapper die beste Teilmenge von Eingangsgrößen auszuwählen ist ein NP-hartes Problem [GE03]. Daher wird hier im Rahmen der Optimierung der Eingangsgrößenanzahl ein Wrapper mit einem gierigen Vorgehen kombiniert. Dabei wird auf eine Methode zur Dimensionsreduktion zurückgegriffen, die auf hierarchischem Clustering beruht. Der Einsatz solcher Verfahren ist im Rahmen von ML-Anwendungen weit verbreitet [KBT11]. Auch in den AutoML-Modellen kommt eine derartige Methode zum Einsatz. Hierbei startet jede Eingangsgröße in ihrem eigenen Cluster und es werden so lange die nächsten Cluster kombiniert, bis eine bestimmte Anzahl an Größen erreicht ist. Das genaue Verfahren der Berechnung der Distanzen wird im Rahmen des Trainings optimiert, baut jedoch auf der euklidischen Distanz auf.

### Gaußprozess-Regression

Zuletzt wird ein ML-Algorithmus angewendet. Die Wahl eines solchen ist entscheidend [DB21]. Die Literatur bietet dafür zwar Hinweise, aber kein klar umrissenes Auswahlverfahren [BCNA21]. Eine Orientierung daran, welche Algorithmen andere Autoren für ähnliche Anwendungen eingesetzt

haben, kann ein guter Ansatz sein [WWIT16]. Jedoch garantiert das keinen Erfolg, denn die Vorhersagequalität, die mit einem ML-Algorithmus erreicht werden kann, hängt auch von den Daten und den gewählten Vorverarbeitungsschritten ab [WWIT16]. Da letztere selten beschrieben werden, ist ein Vergleich mit den eigenen schwierig [KBT11]. Schließlich muss ein geeigneter Algorithmus durch das Testen unterschiedlicher Verfahren gefunden werden [WWIT16], wie es in Abschnitt 4.5 durchgeführt wurde. Daraus ergibt sich, dass die GP-Regression hier geeignet ist. Auch in der Literatur werden diese und ähnliche Kernel Methoden erfolgreich eingesetzt [QSJ+05; VGG21; XZZ21b]. Sie sind flexibel und liefern bei geringer Dimensionalität, das heißt bei weniger als 20 Eingangsgrößen, gute Vorhersagen [BCNA21]. Als Kernel wird der RBF-Kernel eingesetzt, welcher in Gleichung (2.14) dargestellt ist. Dieser ist weit verbreitet [BLFS16; Mur22; Ras06] und wird auch im Bereich der Umformtechnik mit Erfolg verwendet [XZZ21b]. Die Glattheit, die durch den RBF-Kernel forciert wird, kann zusätzlich als Schutz gegen Überanpassung wirken [STV04]. Zwar werden im Anwendungsbereich der Fertigungstechnik auch vielfach Neuronale Netze eingesetzt [KCL21; VYC+23], diese benötigen jedoch häufig große Datenmengen [DMA17; PZL+18]. Im Rahmen dieser Arbeit stehen für das Training der Modelle weniger als 1 000 Datenpunkte zur Verfügung. Dieser Trainingsdatensatz kann als klein bezeichnet werden [HRM19; KBT11]. Dementsprechend kommen Neuronale Netze nicht zum Einsatz. Auch Ensembles von Modellen werden für Problemstellungen im Bereich der Fertigung trainiert [FRC+21; PGK+23]. Ob mit ihnen bessere Ergebnisse als mit einzelnen Modellen erreicht werden können, ist jedoch fraglich [BLFS16; DDL+20]. Sicher ist dagegen, dass ein Ensemble eine gesteigerte Komplexität der Modelle mit sich bringt. Allgemein sollte diese so gering wie möglich gehalten werden [BCNA21]. Daher wurde entschieden, zunächst einzelne Modelle einer GP-Regression zu trainieren und nur dann auf Ensembles zurückzugreifen, wenn damit eine erhebliche Steigerung der Vorhersagequalität erreichbar scheint.

### Rastersuche

Die Hyperparameter der Pipeline müssen für eine möglichst hohe Vorhersagequalität optimiert werden. Dies umfasst die Parameter des Kernels der GP-Regression sowie die der Dimensionsreduktion, inklusive der Anzahl an gewünschten Dimensionen (vgl. Tabelle 6.3). Ein Test verschiedener Werte dieser Hyperparameter ist notwendig [DB21], was im Rahmen dieser Arbeit durch eine Rastersuche umgesetzt wird (vgl. Abschnitt 2.2.4). Zur Validierung von Hyperparameterwerten kann ein separater Datensatz eingesetzt werden. Jedoch wird empfohlen bei weniger als 1 000 Trainingsdatenpunkten eine Kreuzvalidierung durchzuführen [FEF+22]. Dieses Verfahren wird auch bei anderen Autoren im Anwendungsgebiet der Fertigungstechnik eingesetzt [DRF+22; VGG21; XZZ21a] und in Abschnitt 2.2.4 beschrieben. Dadurch ist es möglich, mehr Daten im Training und jeden Datenpunkt zur Validierung zu nutzen. Lediglich der erhöhte Zeitaufwand ist zu bemängeln, da für jede der  $n$  gebildeten Teilmengen ein Modell erzeugt werden muss. Übliche Werte von  $n$  liegen zwischen 3 und 10 [Han12; Ras06]. In Kapitel 4 wurde zur Evaluation eine Kreuzvalidierung mit zehn Teilmengen durchgeführt. Dabei stand die Annäherung an die Vorhersagequalität auf ungesehenen Daten im Vordergrund. Im Rahmen der Rastersuche sollen jedoch lediglich Modelle miteinander verglichen werden, um deren Hyperparameter zu optimieren. Zu diesem Zweck ist auch ein geringeres  $n$  ausreichend, das zusätzlich die Laufzeit des Verfahrens reduziert. Daher wird in der Rastersuche eine Kreuzvalidierung mit fünf Teilmengen durchgeführt. Wie bei anderen Autoren wird die  $R^2$  Metrik berechnet und zum Vergleich von Modellen herangezogen [VGG21]. Die Hyperparameter, die den höchsten Mittelwert in der Kreuzvalidierung erreichen, werden für das Training des zur Vorhersage zu verwendenden Modells auf den gesamten Trainingsdaten genutzt.



## 6 Implementierung

Die in Kapitel 5 entwickelte Vorgehensweise wurde in der Programmiersprache Python in der Version 3.10.9 implementiert und auf einer virtuellen Maschine (VM) an der Universität Stuttgart getestet. Die VM verfügt über vier logische Prozessoren, 8 GB RAM und 44 GB Massenspeicher (vgl. Tabelle 7.10). Auch andere Autoren nutzen Python [CET21; Mur22; XZZ21a]. Diese Sprache wird mitunter als Standard für ML-Anwendungen bezeichnet [BAC+19], da sie über eine Vielzahl von ML- und XAI-Bibliotheken verfügt [KM22; Mur22] und mit grundlegenden Programmierkenntnissen das Training von modernen Modellen in wenigen Zeilen ermöglicht [Cho17]. Informationen über die verwendete Python Distribution sowie über relevante Module sind Tabelle 6.1 zu entnehmen. Diese werden im folgenden Abschnitt vorgestellt. Anschließend wird auf die Implementierung der Datenanalyse eingegangen. Dazu gehören die Darstellung erstellter Module inklusive Anwendungsbeispiele sowie spezifische Hürden bei der Umsetzung der Vorgehensweise.

Bezeichnung	Version	Beschreibung	Link
Anaconda3	2023.03	Python Distribution	<a href="https://www.anaconda.com/">https://www.anaconda.com/</a>
CPython	3.10.9	Python Implementierung	<a href="https://www.python.org/">https://www.python.org/</a>
scikit-learn	0.24.2	ML-Algorithmen	<a href="https://scikit-learn.org/">https://scikit-learn.org/</a>
auto-sklearn	0.15.0	AutoML-Implementierung	<a href="https://automl.github.io/auto-sklearn/">https://automl.github.io/auto-sklearn/</a>
PipelineProfiler	0.1.18	Analyse von AutoML-Pipelines	<a href="https://github.com/VIDA-NYU/PipelineVis">https://github.com/VIDA-NYU/PipelineVis</a>
pandas	2.0.1	Datenanalyse und Manipulation	<a href="https://pandas.pydata.org/">https://pandas.pydata.org/</a>
numpy	1.24.3	Numerische Berechnungen	<a href="https://numpy.org/">https://numpy.org/</a>
scipy	1.10.1	Mathematische Algorithmen, Erweiterung von numpy	<a href="https://scipy.org/">https://scipy.org/</a>
matplotlib	3.7.1	Visualisierung	<a href="https://matplotlib.org/">https://matplotlib.org/</a>
seaborn	0.12.2	Visualisierung	<a href="https://seaborn.pydata.org/">https://seaborn.pydata.org/</a>
plotly	5.14.1	Visualisierung	<a href="https://plotly.com/">https://plotly.com/</a>
mpl_axes_aligner	1.3	Visualisierung	<a href="https://github.com/ryutok/mpl_axes_aligner">https://github.com/ryutok/mpl_axes_aligner</a>

**Tabelle 6.1:** Übersicht über die verwendete Python Distribution sowie relevanter Module, die nicht Teil der Python Standard Library sind, inklusive ihrer Versionen.

### 6.1 Genutzte Module

Im Folgenden werden die zur Implementierung verwendeten Module beschrieben. Soweit zutreffend wird die Wahl des Moduls begründet sowie Zusammenhänge zwischen den Modulen aufgedeckt. Zusätzlich zu den in Tabelle 6.1 dargestellten Modulen wurde noch auf solche zurückgegriffen, die Teil der Python Standard Library sind. Letztere werden im weiteren Verlauf der Arbeit nicht diskutiert.

Das Modul scikit-learn [PVG+11] wurde gewählt, um Implementierungen von ML-Algorithmen bereitzustellen. Es ist populär, leicht anzuwenden und bietet eine große Bandbreite an Algorithmen [DB21; FKE+15; Mur22]. Es eignet sich besonders für den Einsatz auf kleinen Datensätzen [FKE+15] und wird im Bereich der Fertigungstechnik auch von anderen Autoren verwendet [DDL+20; RPA+23; XZZ21a]. Als AutoML-Lösung bietet sich dementsprechend auto-sklearn [FKE+15] an, das auf scikit-learn aufbaut. Dieses wurde kürzlich auf eine Version 2.0 aktualisiert [FEF+22], die unter anderem durch eine Erweiterung des Verfahrens zur Modellauswahl bessere Modelle in kürzerer Zeit erzeugen kann als Version 1.0. Die neuere Version kann leider nicht verwendet werden, da sie bislang keine Regression unterstützt. Es wird daher im Rahmen dieser Arbeit auf Version 1.0 zurückgegriffen [FKE+15]. Der Einsatz von auto-sklearn ist abhängig von einer spezifischen scikit-learn Version, weshalb keine aktuellere zum Einsatz kommt. PipelineProfiler [OCL+20] ist ein Modul zur Untersuchung des Aufbaus von Modellen. Unter anderem kann es dafür genutzt werden, Modelle, die von auto-sklearn erzeugt wurden, zu erforschen. Ohne dieses Modul können diese Modelle nur schwerlich analysiert werden, denn die Interpretation der Ausgaben von auto-sklearn ist aufwendig. PipelineProfiler ermöglicht es, sowohl die Ergebnisse der im Rahmen der Rastersuche durchgeführten Kreuzvalidierung als auch den Aufbau der Modelle zu untersuchen. Gleichzeitig zeigt es auch den mittleren Einfluss eines Schrittes über alle im Rahmen eines auto-sklearn Durchlaufs erzeugten Pipelines auf den  $R^2$  Wert an. Für die Wahl von Verarbeitungsschritten ist solch eine Aggregation von großem Interesse. Sonstige in Tabelle 6.1 dargestellte Module dienen der Datenverarbeitung, stellen mathematische Werkzeuge zur Verfügung oder ermöglichen verschieden Formen der Visualisierung.

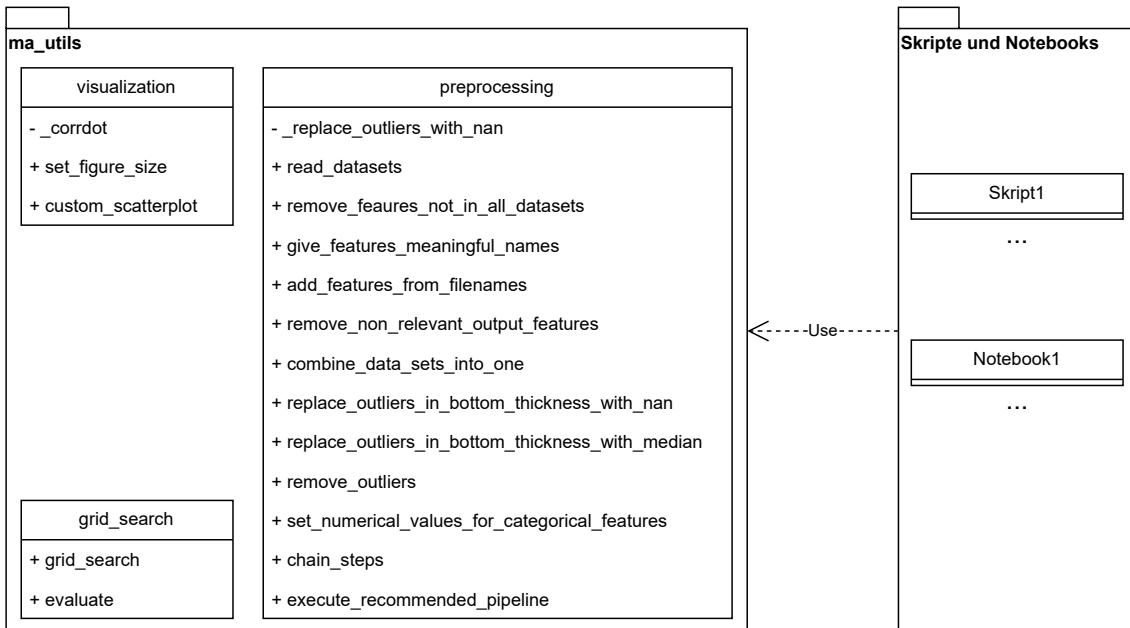
### 6.2 Implementierung der Datenanalyse

Es wurde im Rahmen der vorliegenden Arbeit eigener Code geschrieben. Dieser kann im zugehörigen Repository<sup>1</sup> eingesehen werden. Er erstreckt sich über ein Modulpaket mit Hilfsfunktionen sowie zahlreiche Jupyter Notebooks und Python Skripte. Die explorativen Untersuchungen sowie Auswertungen wurden in Jupyter Notebooks<sup>2</sup> durchgeführt. Es handelt sich dabei um eine Webapplikation, in der formatierter Text, Code und interaktive Ausgaben gemeinsam dargestellt werden können. Die Anaconda3 Distribution stellt eine entsprechende Serveranwendung bereit, die auf der zur Verfügung gestellten VM ausgeführt wurde. Neben den Notebooks wurden auch mehrere Python Skripte geschrieben, hauptsächlich zur Ausführung von Programmen, die sich aufgrund ihrer Laufzeit nicht für die Umsetzung in einem Notebook anbieten.

---

<sup>1</sup><https://gitlab-as.informatik.uni-stuttgart.de/nieblifn/masterarbeit>

<sup>2</sup><https://jupyter.org/>



**Abbildung 6.1:** Darstellung des Python-Packages mit Hilfsfunktionen in Anlehnung an ein UML-Klassendiagramm.

### 6.2.1 Python Paket mit Hilfsfunktionen

Zusätzlich zu Skripten und Notebooks wurde das Modulpaket `ma_utils` mit nützlichen Hilfsfunktionen erstellt. Eine Übersicht über die Module und Funktionen ist in Abbildung 6.1 dargestellt. Das Modul `visualization` besteht aus drei Funktionen, die genutzt werden können, um angepasste Visualisierungen der Daten wie Streudiagramm-Matrizen zu erzeugen und diese ästhetisch ansprechend zu dimensionieren. Da im Rahmen dieser Arbeit auch einige Rastersuchen mit leicht unterschiedlichen Hyperparametern durchgeführt werden, wurde das Modul `grid_search` geschrieben. Die Funktionen darin erzeugen automatisch Werteraster für die Rastersuche gemäß der gegebenen Parameter und können die Ergebnisse einer umfassenden Auswertung der erzeugten Modelle ausgeben. Zuletzt ist noch das Modul `preprocessing` im Paket enthalten, in dem die in Abschnitt 4.3 beschriebenen Vorverarbeitungsschritte implementiert sind. Wird es importiert, können diese Schritte mithilfe eines einzigen Funktionsaufrufs angewendet oder einzeln mit angepassten Argumenten aufgerufen werden. Die Laufzeit der Vorverarbeitungsschritte ist, zumindest mit den zur Verfügung stehenden Daten, so kurz, dass sie zur Ausführzeit eines Notebooks durchgeführt werden können. Es wird daher auch keine vorverarbeitete Datenversion zwischengespeichert. Die Daten sind auf der VM sowie im Repository unverändert hinterlegt. Für die Ausführung der Vorgehensweise auf anderen Daten, insbesondere wenn diese auch nur geringfügig anders kodiert sind, muss die Implementierung der Vorverarbeitung im Modulpaket angepasst werden. Dies war bereits für die Daten der erneuten Simulation, wie sie in Abschnitt 4.5.6 beschrieben sind, der Fall.

Ausschnitt des Dateinamens	Wert der Reibung
F08	0,08
F8	0,08
F1	0,10
F12	0,12

**Tabelle 6.2:** Gegenüberstellung eines Teils der Dateinamen und der entsprechenden Werte der Eingangsgröße Reibung.

### 6.2.2 Erzeugen des Trainingsdatensatzes und der Testdaten

Die Daten werden zunächst mithilfe des Moduls `preprocessing` eingelesen und vorbereitet. Zur möglichst objektiven Beurteilung erzeugter Modelle wird eine Menge an Daten als Testdaten vor dem Training der ML-Modelle abgespalten. Auf diesen Testdaten kann die Vorhersagequalität der Modelle auf ungesehenen Daten bewertet und eine Überanpassung erkannt werden [Mur22; Pla21]. Der Testdatensatz wird mithilfe der Funktion `train_test_split` des Moduls `scikit-learn` erzeugt (vgl. Listing 6.1 Zeile 4ff). Dieser zeigt neben der Auswahl der Testdaten die Verwendung des Moduls `preprocessing`. Wie gezeigt können die erarbeiteten Verarbeitungsschritte mit einem einzelnen Funktionsaufruf automatisiert durchgeführt werden (vgl. Listing 6.1 Zeile 2). Dabei ist jeder Schritt in einer Funktion implementiert, wie in Abbildung 6.1 zu sehen ist. Dementsprechend ist es auch möglich, einzelne Funktionen mit individuellen Argumenten in abweichender Reihenfolge auszuführen. Im dargestellten Fall wurden die Ausreißerdatenpunkte nicht entfernt. Daher sind auch im Testdatensatz Ausreißer enthalten. Sofern nicht explizit anders angegeben, werden Modelle, die auf Daten ohne Ausreißer trainiert werden, auf einem von Ausreißerdatenpunkten befreiten Testdatensatz evaluiert.

Bei der Implementierung der Vorverarbeitung hat sich eine bemerkenswerte Hürde ergeben. Sie besteht in der Extraktion von Eingangsgrößen aus der Verzeichnisstruktur. Zur Umsetzung dieser Funktionalität wurde auf reguläre Ausdrücke zurückgegriffen. Besondere Aufmerksamkeit ist dabei der Eingangsgröße Reibung zu widmen. Die Kodierung der Werte dieser Größe ist nicht intuitiv nachvollziehbar und hat im Rahmen der vorliegenden Arbeit für erheblichen Mehraufwand gesorgt. Da die Werte zunächst so interpretiert wurden, wie sie in den Dateinamen stehen, wurden die ersten Untersuchungen mit einer grundlegend falschen Verteilung der Reibungswerte durchgeführt. Diese mussten nach der eventuellen Korrektur wiederholt werden. Eine Wertetabelle, die Teile der Dateinamen den tatsächlichen Werten der Größe gegenüberstellt, ist in Tabelle 6.2 wiedergegeben.

### 6.2.3 Auto-sklearn

Im Rahmen der Untersuchungen in Kapitel 4 wurden zahlreiche Modelle durch `auto-sklearn` trainiert. Ein entsprechender Ausschnitt ist in Listing 6.2 angegeben. Die Angabe eines Wertes für das Argument `time_left_for_this_task` ist notwendig, wenn `auto-sklearn` nicht nach einer Stunde abbrechen soll. Standardmäßig wird zur Validierung ein extra Datensatz abgespalten, jedoch kann über das passende Argument eine Kreuzvalidierung erzwungen werden. Bei einer solchen werden ohne anderweitige Spezifikation fünf Teilmengen gebildet. Weitere Argumente stehen zur Verfügung, sind aber im Rahmen dieser Arbeit nicht notwendig.

---

**Listing 6.1** Codeausschnitt, der die Verwendung des Moduls preprocessing veranschaulicht und die Erzeugung des Testdatensatzes darstellt.

---

```

1     # Load data and do basic preprocessing
2     df = ma_utils.preprocessing.execute_recommended_pipeline(version='keep_outliers')
3
4     df_train, df_test = sklearn.model_selection.train_test_split(
5         df,
6         test_size=0.15,
7         random_state=123,
8     )
9
10    # Saved for later use in
11    ma_utils.preprocessing.TEST_INDICES == sorted(df_test.index)
12

```

---

Jedoch gibt es bei der Ausführung von auto-sklearn einiges zu beachten. Als ein Vorverarbeitungsschritt kann automatisch eine Imputation gewählt werden. Das impliziert, dass auto-sklearn mit fehlenden Werten umgehen kann. Wie diese in den Daten darzustellen sind, wird in der Dokumentation des Moduls nicht erwähnt. Auch die Dokumentation von scikit-learn, dessen Funktionen durch auto-sklearn aufgerufen werden, lässt die Spezifikation einer Repräsentation für fehlende Werte vermissen. In einigen offiziellen Beispielen wird jedoch `numpy.nan` verwendet. Wie in Abschnitt 4.2 diskutiert können Ausreißer durch Imputation ersetzt werden. Dabei ist ein geeigneter Lageparameter als Ersatz zu wählen. Es war beabsichtigt, zu analysieren, welche Lageparameter auto-sklearn wählt. Dazu werden zunächst die Ausreißer durch `numpy.nan` ersetzt und damit als fehlende Werte markiert, die durch Imputation ersetzt werden sollen. Wird anschließend auto-sklearn wie in Listing 6.2 gezeigt aufgerufen, so bricht die Ausführung ab und es wird eine Fehlermeldung ausgegeben.

Ein weiteres Problem bei der Nutzung von auto-sklearn ergibt sich durch begrenzten Speicherplatz. Grund dafür ist, dass bei einer Ausführung über mehrere Stunden zahlreiche temporäre Dateien angelegt werden. Diese können einige Gigabyte an Festplattenspeicher beanspruchen. Steht von diesem nicht ausreichend zur Verfügung, um alle Dateien abzuspeichern, so bricht die Ausführung mit einem Fehler ab. Im vorliegenden Fall waren die insgesamt 24 GB an Speicherplatz, die der VM zur Verfügung standen, nicht ausreichend. Es musste weiterer Speicher eingehängt und auto-sklearn so konfiguriert werden, dass dieser für temporäre Dateien verwendet wird. So ist es gelungen, die Fehlermeldung zu vermeiden. Nach einer erfolgreichen Ausführung über die definierten 18 Stunden wird, wie in Listing 6.2 zu sehen, das resultierende Objekt mithilfe des Moduls `pickle`

---

**Listing 6.2** Beispiel für einen Aufruf von auto-sklearn.

---

```

1     automl = autosklearn.regression.AutoSklearnRegressor(
2         time_left_for_this_task=64800, # 18 h
3         resampling_strategy='cv',
4         resampling_strategy_arguments={'folds': 5}, # default
5         tmp_folder="<path to additional disk space>",
6     )
7     automl.fit(X_train, Y_train)
8     pickle.dump(automl, open("<path to model directory>", 'wb'))
9

```

---

abgespeichert. Will man dieses zur späteren Analyse wieder laden, so müssen die gleichen Versionen relevanter Module importiert werden. Um Modelle im Rahmen der Modellauswahl zu evaluieren, wird auf eine Kreuzvalidierung zurückgegriffen. Zu diesem Zweck stellt `scikit-learn` die Funktion `cross_val_score` zur Verfügung. Obwohl `auto-sklearn` auf `scikit-learn` aufbaut, sind die Modelle nicht mit dieser Funktion kompatibel. Die verwendete Kreuzvalidierung wurde daher in den Notebooks selbst implementiert.

### 6.2.4 Rastersuche

Die in Kapitel 5 erarbeitete Vorgehensweise umfasst die Optimierung von Hyperparametern der ML-Pipeline mittels einer Rastersuche. Die getesteten Hyperparameter und ihre Werte sind Tabelle 6.3 zu entnehmen. Die Dimensionsanzahl, die Verknüpfung und die Aggregatfunktion sind Hyperparameter des Verfahrens zur Dimensionsreduktion. Über die Dimensionsanzahl wird festgelegt, wie viele Eingangsgrößen zum Training des Modells verwendet werden. Das Verfahren zur Dimensionsreduktion kombiniert iterativ die nächsten Eingangsgrößen, bis die gewünschte Anzahl erreicht ist. Die Berechnung der Distanz zweier Größen wird dabei durch den Hyperparameter Verknüpfung festgelegt. Bei `ward` wird die gemeinsame Varianz der Größen berechnet, `complete` und `average` beziehen sich jeweils auf die maximale oder durchschnittliche euklidische Distanz. Wie aus zwei Größen eine neue berechnet wird, bestimmt der Hyperparameter Aggregatfunktion. Es wird entweder der Mittelwert oder der Median verwendet. Die übrigen zwei Hyperparameter aus Tabelle 6.3 gehören zur GP-Regression. Ob die Ausgangsgröße vor dem Training normalisiert wird, bestimmt der Hyperparameter Normalisierung. Der `alpha` Wert dient hauptsächlich der numerischen Stabilität des Verfahrens, kann aber auch als Hinzufügen einer Störgröße zur Berechnung des Kernels interpretiert werden.

In Listing 6.3 ist beispielhaft eine Ausführung einer Rastersuche dargestellt. Dabei wird auch die Dimension des Eingangsgrößenraums angepasst. Entsprechend wird der Hyperparameter `n_clusters` des Verfahrens zur Dimensionsreduktion variiert (Zeile 14). Die Anzahl der Eingangsgrößen ist entscheidend für mehrere Modellparameter der GP-Regression, die im Rahmen des Trainings

Hyperparameter	Getestete Werte	Beschreibung
Dimensionsanzahl	$\{i \in \mathbb{N} : 2 \leq i \leq 36\}$	Anzahl der Eingangsgrößen, die nach der Dimensionsreduktion vorliegen sollen.
Verknüpfung	<code>ward</code> , <code>complete</code> , <code>average</code>	Art der Berechnung der Distanzen zwischen Eingangsgrößen im Rahmen der Dimensionsreduktion.
Aggregatfunktion	Mittelwert, Median	Art der Berechnung der neuen Werte für durch die Dimensionsreduktion zusammengeführte Eingangsgrößen.
Normalisierung	Nein, Ja	Normalisierung der Ausgangsgröße.
alpha	$\{10^i : i \in \{-2, -3, -4, -6, -8\}\}$	Entspricht einer normalverteilten Störgröße, die der Berechnung des Kernels hinzugefügt wird und unter anderem der numerischen Stabilität des Verfahrens dient.

**Tabelle 6.3:** Parameterraster für die Rastersuche.

**Listing 6.3** Beispiel für eine Rastersuche für die Ausgangsgröße Schädigung.

```

1   pipe = Pipeline(steps=[
2       ('poly', PolynomialFeatures(2)),
3       ('scaler', StandardScaler()),
4       ('dim_red', FeatureAgglomeration(affinity='euclidean')),
5       ('gp', TransformedTargetRegressor(
6           regressor=GaussianProcessRegressor(kernel=ConstantKernel() * RBF()),
7           func=np.log,
8           inverse_func=np.exp,
9       )),
10      ])
11  param_grid = []
12  for feature_dim in range(2, 37):
13      param_grid.append(dict(
14          dim_red__n_clusters=[feature_dim],
15          gp__kernel__k2__length_scale=[[1.0] * feature_dim],
16          gp__kernel__k2__length_scale_bounds=[[10 ** -6, 10 ** 6] * feature_dim],
17          dim_red__linkage=['ward', 'complete', 'average'],
18          dim_red__pooling_func=[np.mean, np.median],
19          gp__alpha=[0.01, 0.001, 0.0001, 0.000001, 0.00000001],
20          gp__normalize_y=[False, True],
21      ))
22  grid_search = GridSearchCV(pipe, param_grid=param_grid, cv=5)
23  grid_search.fit(
24      df_train.loc[:, ma.preprocessing.INPUT_FEATURES],
25      df_train.loc[:, 'Max-Damage']
26  )
27

```

optimiert werden. Zum Beispiel für den Wert von `length_scale` (Zeile 15), der die Bandbreite des RBF-Kernels beschreibt (vgl. Abschnitt 2.2.4). Sie wird im Rahmen des Trainings an die Daten angepasst. Jedoch kann vorgegeben werden, ob die Bandbreite für alle Eingangsgrößen gleich oder pro Dimension eigenständig gewählt werden soll. Im Rahmen der erarbeiteten Vorgehensweise wird ein Vektor für `length_scale` angegeben, um eine Bandbreite pro Eingangsgröße zu verwenden. Dieses Vorgehen wird in der Rastersuche nicht variiert. Da die Dimension des Vektors jedoch abhängig von der Anzahl der Eingangsgrößen ist, müssen beide übereinstimmen. Die Dimension des Vektors für `length_scale` wird daher in der Rastersuche abhängig von `n_clusters` gesetzt. Dabei kann nicht jede Kombination von `length_scale` und `n_clusters` getestet werden, wie es eigentlich in der Rastersuche der Fall ist. Dieser Zusammenhang gilt analog für `length_scale_bounds`, was die Wertebereiche der Bandbreite festlegt. Um dieses Problem zu umgehen, besteht die Möglichkeit, der Rastersuche von `scikit-learn` nicht nur ein Dictionary zu übergeben, sondern eine Liste von Dictionaries. Jeder Eintrag dieser Liste enthält dabei die Werte, die für eine Anzahl an Eingangsgrößen getestet werden sollen.

Nicht nur für die Rastersuche stellt `scikit-learn` ein geeignetes Werkzeug zur Verfügung. Auch für die Transformation der Ausgangsgröße wird eine Klasse angeboten. Der `TransformedTargetRegressor` wird im vorliegenden Fall mit einer Instanz der GP-Regression, sowie zwei Funktionen initialisiert. Eine der Funktionen transformiert die Ausgangsgröße, bevor sie

## 6 Implementierung

---

dem Lernalgorithmus übergeben wird. Die andere führt die inverse Transformation auf den Ausgaben desselben aus. So kann der Lernalgorithmus transparent mit einer transformierten Ausgangsgröße arbeiten.



## 7 Diskussion der Evaluationsergebnisse

Im Rahmen dieser Arbeit wurde eine Vorgehensweise entwickelt, um die Ausgaben einer Kaltmasivumformsimulation zu approximieren. Diese wird anhand eines Datensatzes, der vom Institut für Umformtechnik der Universität Stuttgart zur Verfügung gestellt wurde, erprobt. Dabei wird die Approximation aller sechs vom Fachexperten als besonders interessant markierten Ausgangsgrößen untersucht. Da für die Größen Umformgrad und Schädigung zu diesem Zeitpunkt jedoch lediglich mit ML-Modellen eine hochqualitative Vorhersage möglich ist, stehen sie im Zentrum der Auswertung. Für die anderen vier Ausgangsgrößen kann wie in Kapitel 4 beschrieben eine Approximation direkt aus den Korrelationen zwischen Eingangs- und Ausgangsgrößen abgeleitet werden.

Im Rahmen der Evaluation wird die Vorhersagequalität durch die Metrik  $R^2$  bewertet, auf die auch andere Autoren verwandter Arbeiten zurückgreifen [DRF+22; HCN22; RPA+23; VYC+23]. Die Angaben werden auf zwei Nachkommastellen gerundet. Darüber hinaus werden Residualplots erzeugt und ausgewertet. Sie stellen die Abweichungen der Vorhersagen den tatsächlichen Werten gegenüber. Dabei sollten die Punkte unsystematisch und gleichmäßig um den Wert 0 der Abweichungen verteilt sein. Sind klare Muster in den Residualplots erkennbar, so besteht für das Modell Verbesserungspotenzial (vgl. Abschnitt 2.2.5). Neben der Vorhersagequalität wird auch auf die Dauer der Anwendung der Vorgehensweise eingegangen.

Als Teil der Vorgehensweise werden Testdaten vom Datensatz abgespalten. Auf den Trainingsdaten wird anschließend für die Ausgangsgrößen Umformgrad und Schädigung jeweils eine Rastersuche durchgeführt, um Hyperparameter zu optimieren. Dabei wird für jedes erstellte Modell der durchschnittliche  $R^2$  Wert einer Kreuzvalidierung auf fünf Teilmengen, im Folgenden  $R_{KV5}^2$  genannt, erfasst. Als die besten Hyperparameterwerte werden jene bezeichnet, mit denen der höchste  $R_{KV5}^2$  erreicht wird. Mit ihnen wird auf den gesamten Trainingsdaten ein Modell trainiert und auf dem Testdatensatz evaluiert. Der dabei ermittelte  $R^2$  Wert auf den Testdaten wird als  $R_{\text{test}}^2$  bezeichnet. Die Ergebnisse dieser Evaluation sind in Abschnitt 7.1 dargestellt. Im Anschluss wird in Abschnitt 7.3 mittels einer Ablationsstudie der Einfluss einzelner Schritte der Vorgehensweise auf die Approximation überprüft. Dabei wird jeweils ein Schritt der erarbeiteten Vorgehensweise ausgelassen, bevor durch eine Rastersuche die besten Hyperparameterwerte ermittelt werden. Das resultierende Modell wird auf dem Testdatensatz evaluiert. Die Vorhersagequalität auf den Testdaten und die Dauer der Rastersuche werden mit denen verglichen, die das Modell der vollständigen Vorgehensweise erreicht. Danach werden in Abschnitt 7.4 mit der in Kapitel 5 vorgestellten Vorgehensweise Modelle für die anderen vier Ausgangsgrößen erzeugt und denen gegenübergestellt, die in Kapitel 4 beschrieben sind. Daraufhin wird diskutiert, wie die Erklärbarkeit der erzeugten Modelle durch XAI-Verfahren gesteigert werden kann (vgl. Abschnitt 7.5). Zuletzt werden die wichtigsten Erkenntnisse der Evaluation zusammengefasst (vgl. Abschnitt 7.6).

	Umformgrad	Schädigung
Dimensionsreduktion: Dimensionsanzahl	24	13
Dimensionsreduktion: Verknüpfung	complete	average
Dimensionsreduktion: Aggregatfunktion	Mittelwert	Median
GP-Regression: alpha	0,01	0,001
GP-Regression: Normalisierung	Ja	Ja
$R_{KV5}^2$	0,95	0,71
$R_{test}^2$	0,98	0,99
Dauer Rastersuche	100 560 s	100 500 s

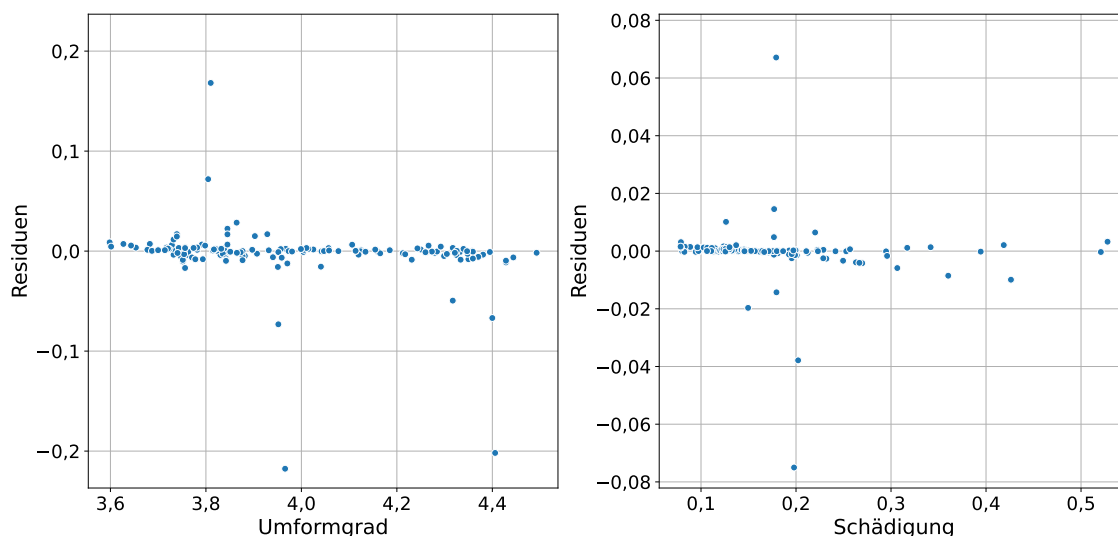
**Tabelle 7.1:** Charakterisierung der besten Modelle, die für die Ausgangsgrößen Umformgrad und Schädigung durch eine Rastersuche gefunden wurden.

## 7.1 Evaluation der vollständigen Vorgehensweise

Zunächst werden entsprechend der erarbeiteten Vorgehensweise aus Kapitel 5 die Hyperparameter für die zwei Ausgangsgrößen Umformgrad und Schädigung durch eine Rastersuche optimiert. Im Rahmen einer solchen werden 2 100 Modelle mit jeweils verschiedenen Werten der Hyperparameter erzeugt und validiert. Die getesteten Werte sind in Kapitel 6 beschrieben. Neben den Hyperparametern werden die Ergebnisse einer Kreuzvalidierung auf fünf Teilmengen sowie die Zeitangaben für die Rastersuche erfasst. Diese Angaben werden im Folgenden ausgewertet.

Die besten gefundenen Hyperparameter sowie die damit erreichten Vorhersagequalitäten sind in Tabelle 7.1 dargestellt. Je nach Ausgangsgröße unterscheidet sich die Konfiguration der Dimensionsreduktion. Es fällt auf, dass für die Vorhersage des Umformgrads elf Eingangsgrößen mehr verwendet werden als für die der Schädigung. Die Hyperparameter der GP-Regression sind für beide ähnlich. Normalisierung der Ausgangsgröße wird in beiden Fällen angewendet, der Wert des Parameters alpha unterscheidet sich lediglich um den Faktor 10. In der Kreuzvalidierung mit fünf Teilmengen erreichen beide Modelle unterschiedliche  $R^2$  Werte. Für den Umformgrad beträgt dieser  $R_{KV5}^2 = 0,95$  und liegt damit in einem hohen Bereich. Der Wert von 0,71 für die Schädigung erreicht dagegen nicht dasselbe Niveau. Werden die Modelle auf den gesamten Trainingsdaten trainiert und auf dem Testdatensatz ausgewertet, erzielen sie bessere  $R^2$  Werte. Bei Approximation der Testdaten wird für den Umformgrad ein  $R_{test}^2$  Wert von 0,98 erreicht, für die Schädigung liegt dieser bei 0,99. Das bedeutet, dass lediglich 2, beziehungsweise 1 Prozent der Abweichung eines Wertes vom Mittelwert der Ausgangsgröße nicht von den Modellen erfasst wird. Die Angaben zur Dauer in Tabelle 7.1 beziehen sich auf die Rastersuche. Diese hat insgesamt für den Umformgrad 100 560 s und für die Schädigung 100 500 s gedauert. Beides entspricht über 27 Stunden.

Des Weiteren können die in Abbildung 7.1 dargestellten Residualplots zur Analyse der angefertigten Modelle herangezogen werden. Für die Ausgangsgröße Umformgrad sind die Residuen gleichmäßig um den Wert 0 der y-Achse verteilt. Lediglich einzelne Punkte liegen weiter entfernt. Auch bei der Schädigung gibt es vereinzelt betragsmäßig relativ hohe Residuen. Jedoch liegen hier ebenfalls die meisten Werte nahe  $y = 0$ . Entsprechend der rechtsschiefen Verteilung der Ausgangsgröße Schädigung befindet sich die Mehrheit der Punkte auf der linken Diagrammseite. Der durchschnittliche Residuenbetrag wächst mit der Schädigung. Dass das Modell weniger präzise Vorhersagen für hohe Werte macht, kann damit erklärt werden, dass nur wenige solcher Werte im Datensatz



**Abbildung 7.1:** Residualplots für die Ausgangsgrößen Umformgrad und Schädigung bei Vorhersage durch das jeweils beste gefundene Modell im Rahmen der entwickelten Vorgehensweise.

enthalten sind. Dementsprechend ist der Einfluss solcher Werte auf den  $R^2$  Wert relativ gering. Sie fallen daher bei der Optimierung des Verfahrens anhand des  $R^2$  Wertes nicht stark ins Gewicht. Weiterhin fällt auf, dass einige Punkte annähernd auf einer Geraden mit sichtbar negativer Steigung liegen. Für diese konnte kein Zusammenhang mit Eingangsgrößen gefunden werden, daher ist das Zustandekommen dieses Musters ungeklärt. Jedoch ist es nur schwach ausgeprägt und die Punkte, welche es ausmachen, haben betragsmäßig keine übermäßig großen Residuen. Es herrscht daher diesbezüglich kein Handlungsbedarf. Jedoch kann das darauf hindeuten, dass sich weitere Zusammenhänge aus den Daten ergeben, die das erzeugte Modell nicht erfasst. Auch fällt auf, dass die Residuen für den Umformgrad betragsmäßig größere Werte annehmen als für die Schädigung. Das lässt sich jedoch mit dem größeren Wertebereich des Umformgrads erklären. Darüber hinaus sind für beide Ausgangsgrößen keine Abhängigkeiten der Residuen von den Werten der x-Achse und damit Defizite der Modelle zu erkennen.

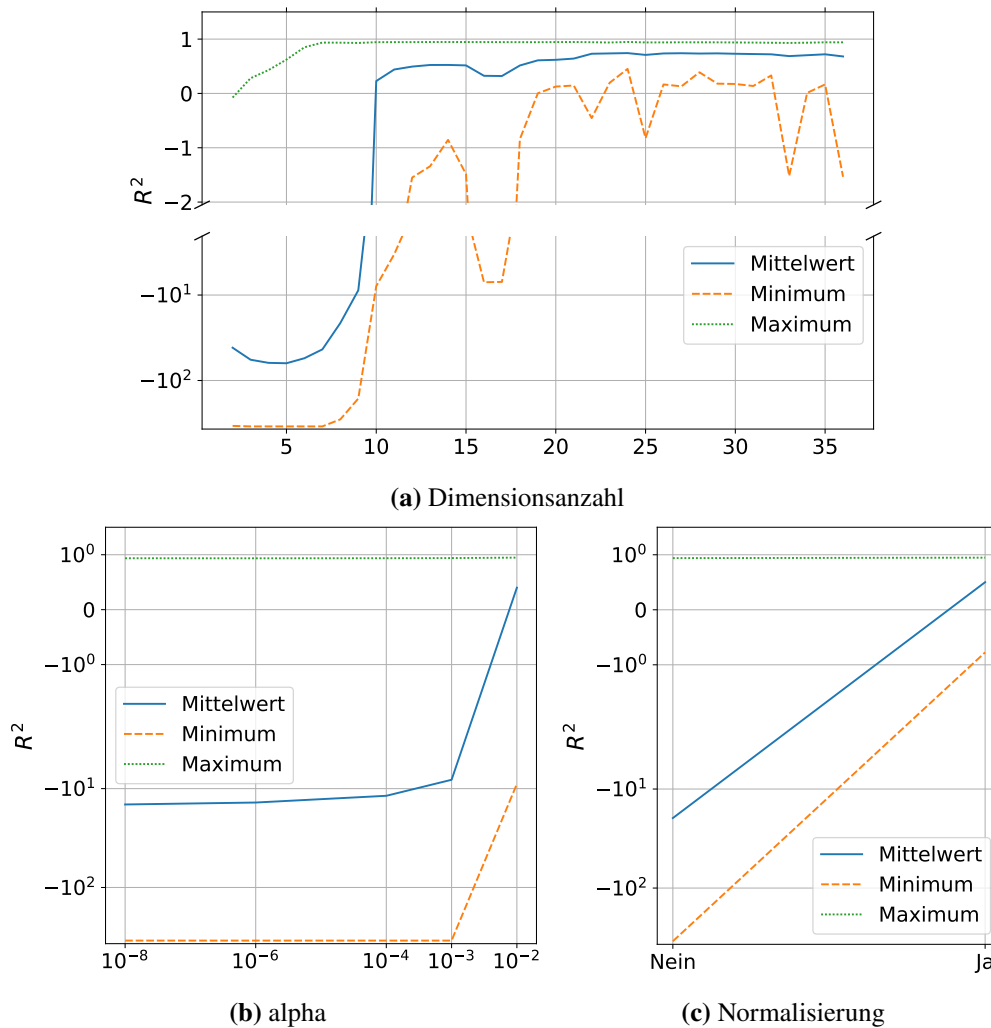
Zum besseren Vergleich mit den Modellen aus Kapitel 4 wird für diese ebenfalls der  $R_{\text{test}}^2$  Wert berechnet. Dabei ergibt sich, dass für beide Ausgangsgrößen zuvor die beste Vorhersagequalität mit den AutoML-Modellen erreicht wurde. Für den Umformgrad wird mit AutoML-Verfahren die gleiche Vorhersagequalität erzielt wie mit der erarbeiteten Vorgehensweise. Bei Approximation der Schädigung ist das eigene Verfahren dagegen besser. Das AutoML-Modell erreicht für diese Ausgangsgröße einen um 0,02 niedrigeren  $R_{\text{test}}^2$  Wert von 0,97. Daraus lässt sich schließen, dass die Modelle, welche durch die im Rahmen dieser Arbeit entwickelten Methode erzeugt wurden, keine schlechtere Approximation liefern als die AutoML-Modelle. Darüber hinaus ist festzustellen, dass letztere sehr viel komplexer aufgebaut sind. Statt nur einer Pipeline trainieren sie viele und verwenden zur Vorhersage des Umformgrads und der Schädigung acht, beziehungsweise drei davon in einem Ensemble. Diese gesteigerte Komplexität führt nicht zu einer besseren Approximation, sodass die mit dem in dieser Arbeit entwickelten Verfahren erzeugten Modelle zu bevorzugen sind.

## 7.2 Einfluss von Hyperparameterwerten auf die Vorhersagequalität

In diesem Abschnitt wird der Einfluss von Hyperparametern auf die Vorhersagequalität anhand der in der Kreuzvalidierung erhobenen Werte untersucht. Der  $R_{KV5}^2$  Wert wird zur Auswahl der Hyperparameter verwendet und bietet eine Annäherung an den  $R_{\text{test}}^2$  Wert, der auf bisher ungesehenen Daten erreicht wird. Da  $R_{KV5}^2$  für jedes Modell, das im Rahmen der Rastersuche erstellt wurde, zur Verfügung steht, kann für jeden einzelnen Wert eines Hyperparameters jeweils der maximal, der minimal und der im Durchschnitt erreichte Wert ermittelt werden. Diese Werte sind über die zwei Ausgangsgrößen Umformgrad und Schädigung aggregiert in Abbildung 7.2 für die Dimensionsanzahl, den alpha Wert sowie die Normalisierung der Ausgangsgröße dargestellt. Das ermöglicht es, den Einfluss dieser Hyperparameter auf die Vorhersagequalität zu untersuchen.

Bei der Darstellung für die Anzahl der Dimensionen in Abbildung 7.2a ist zu beachten, dass der untere Teil der y-Achse logarithmisch skaliert ist. Diese Grafik lässt erkennen, dass mit der Dimensionsanzahl der maximal erreichbare  $R_{KV5}^2$  Wert deutlich steigt, bis er bei 7 Eingangsgrößen annähernd seinen Höchstwert erreicht. Ähnlich verhält es sich für den durchschnittlichen  $R_{KV5}^2$  Wert. Die entsprechende Kurve flacht bei 10 Dimensionen deutlich ab und erreicht bei 22 annähernd ihr Maximum. Im Anschluss verändern sich beide Werte kaum noch. Der höchste mittlere Wert beträgt 0,74 und wird mit 24 Dimensionen erreicht. Der minimal erreichbare Wert schwankt sehr viel stärker als die beiden anderen Größen. Bei 19 Dimensionen erreicht er zum ersten Mal 0 und fällt danach nicht mehr unter  $-2$ . Jedoch sind die minimal zu erreichenden  $R_{KV5}^2$  Werte für alle Dimensionsanzahlen durchwachsen und liegen stets unter 0,5. Es lässt sich ableiten, dass mit steigender Dimensionsanzahl auch  $R_{KV5}^2$  durchschnittlich tendenziell steigt. Jedoch kann nicht allein durch eine hohe Anzahl an konstruierten Eingangsgrößen eine hochqualitative Vorhersage sichergestellt werden. Weiterhin fällt auf, dass bereits mit 7 Eingangsgrößen annähernd die maximal mögliche Approximationsqualität erreicht werden kann. Das entspricht der Anzahl der originalen Eingangsgrößen, die in den Daten kodiert sind. Jedoch handelt es sich bei den hier verwendeten Größen nicht um diese. Stattdessen werden auf Basis der Eingangsgrößen in den Daten polynomielle Größen bis zum Grad 2 erzeugt und anschließend eine Dimensionsreduktion durchgeführt. Letztere kombiniert die Werte mehrerer Größen und berechnet daraus neue.

Für die Hyperparameter alpha sowie Normalisierung der GP-Regression sind die dargestellten Tendenzen eindeutig. In beiden Fällen wird mit einem Wert ein deutlich besseres mittleres Ergebnis erzielt als mit den anderen. Auch der minimal erreichte  $R_{KV5}^2$  Wert ist für alpha =  $10^{-2}$  und Normalisierung = Wahr um Größenordnungen besser als mit den Alternativen. Wieder ist zu beachten, dass bei den entsprechenden Abbildungen 7.2b und 7.2c die y-Achse logarithmisch skaliert ist. Interessant ist ebenso, dass der Maximalwert von  $R_{KV5}^2$  über die Werte beider Hyperparameter lediglich um 0,01 schwankt. Daraus ist zu schließen, dass die Optimierung der Hyperparameter alpha und Normalisierung dazu beiträgt, eine hochqualitative Approximation zu erzielen. Sie ist zu diesem Zweck jedoch nicht notwendig. Nicht erforderlich ist auch die Optimierung der Aggregatfunktion sowie die Art der Kopplung. Beides sind Hyperparameter des Verfahrens zur Dimensionsreduktion. Die entsprechenden Graphen – hier nicht abgebildet – zeigen lediglich horizontal verlaufende Geraden. Weder auf den maximalen, minimalen oder durchschnittlichen  $R_{KV5}^2$  Wert ist ein nennenswerter Einfluss dieser Hyperparameter zu erkennen.



**Abbildung 7.2:** Einfluss der Werte ausgewählter Hyperparameter auf den  $R^2$  Mittelwert einer Kreuzvalidierung mit fünf Teilmengen.

## 7.3 Ablationsstudie

Wie zuvor für einzelne, in der Rastersuche optimierte Hyperparameter soll im Folgenden für weitere Elemente der Vorgehensweise untersucht werden, welchen Einfluss sie auf die Vorhersagequalität des resultierenden Modells haben. Zu diesem Zweck wurde eine Ablationsstudie durchgeführt. Im Rahmen dieser wird die Vorgehensweise wiederholt angewendet, wobei jedes Mal einzelne Schritte ausgelassen werden. Zu den untersuchten Komponenten zählen die Verwendung von Skalaren oder Vektoren für die Bandbreite der GP-Regression, die Erzeugung polynomieller Eingangsgrößen, die Standardisierung, die Dimensionsreduktion, die Transformation von Ausgangsgrößen in annähernde Normalverteilung sowie die Entfernung von Ausreißerdatenpunkten. Wie zuvor wird eine Rastersuche durchgeführt und die Ergebnisse, inklusive der  $R^2$  Werte und zeitlicher Angaben, werden gespeichert. Es folgt eine Analyse dieser Ergebnisse.

	Umformgrad		Schädigung	
$R_{KV5}^2$	0,87	(-0,08)	0,04	(-0,67)
$R_{\text{test}}^2$	0,89	(-0,09)	0,01	(-0,98)
Dauer Rastersuche	4 560 s	(-96 000 s)	3 900 s	(-96 600 s)

**Tabelle 7.2:** Werte für die besten Modelle bei Verwendung von Skalaren für die Bandbreite. In Klammern stehen Veränderungen im Vergleich zur vollständigen Vorgehensweise.

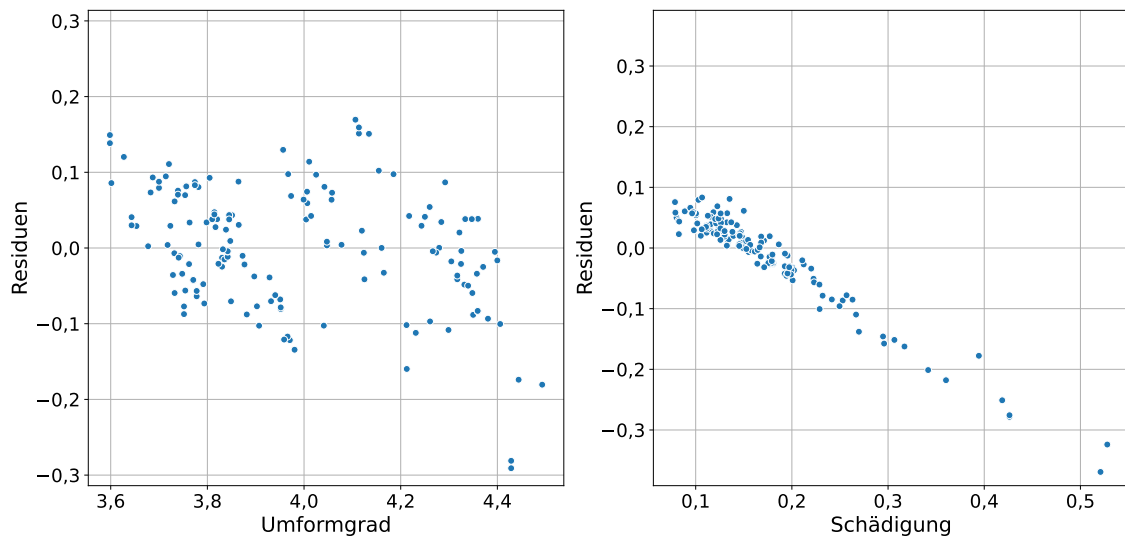
### 7.3.1 Format der Bandbreite

Die Bandbreite des verwendeten RBF-Kernels steuert, wie stark sich weit entfernte Datenpunkte gegenseitig beeinflussen. Es ist möglich, für diesen Modellparameter einen Skalar zu verwenden. In diesem Fall ist die Bandbreite für alle Eingangsgrößen gleich. Stattdessen kann auch ein Vektor im Raum der Eingangsgrößen angegeben werden. Das hat zur Folge, dass für jede Dimension ein eigener Bandbreitenwert gilt. Dies erlaubt eine bessere Anpassung des Modells an die Daten und wurde daher für die im Rahmen dieser Arbeit entwickelte und in Kapitel 5 beschriebene Vorgehensweise ausgewählt. Abweichend von der erarbeiteten Vorgehensweise werden in diesem Teilabschnitt Ergebnisse untersucht, die mit skalaren Bandbreiten erreicht werden können. Diese Ergebnisse sind in Tabelle 7.2 dargestellt.

Insgesamt sinkt die Vorhersagequalität bei der Verwendung von skalaren Bandbreiten. Für die Ausgangsgröße Umformgrad fällt  $R_{\text{test}}^2$  auf 0,89. Dieser Wert liegt um 0,09 unter dem, der mit der vollständigen Vorgehensweise erreicht wird. Der Residualplot für den Umformgrad ist in Abbildung 7.3 dargestellt. Im Vergleich mit dem Modell der vollständigen Vorgehensweise sind die Residuen ebenfalls unsystematisch um den Wert 0 verteilt und es ist kein starkes Muster über die x-Werte erkennbar. Jedoch sind die Residuen um ein Vielfaches größer als zuvor. Dieses Modell erreicht demnach keine präzise Approximation.

Das beste Modell mit skalarer Bandbreite erreicht für die Ausgangsgröße Schädigung lediglich einen  $R_{\text{test}}^2$  Wert von 0,01, ein Rückgang um 0,98 im Vergleich zu dem Modell mit vektorieller Bandbreite. Damit ist das Modell mit skalarer Bandbreite kaum besser als eines, das unabhängig von den Eingangsgrößen den Mittelwert der Schädigung ausgibt. Darüber hinaus ergibt eine detaillierte Untersuchung, dass der  $R^2$  Wert auf den Trainingsdaten bei 1,00 liegt. Ein derart hoher Wert und eine Differenz zu  $R_{\text{test}}^2$  von 0,99 deuten auf eine Überanpassung des Modells hin. Im Residualplot in Abbildung 7.3 ist zusätzlich ein deutliches, lineares Muster mit negativer Steigung auszumachen, das alle Testdatenpunkte umfasst. Auch anhand dieser Abbildung wird demnach deutlich, dass die Vorhersagequalität unter Verwendung eines Skalars für die Bandbreite des RBF-Kernels im Vergleich zu einem Vektor nachlässt.

Eine positive Entwicklung lässt sich jedoch auch aus Tabelle 7.2 ablesen. Die Zeit, die für die Rastersuche benötigt wird, sinkt bei skalaren Bandbreiten erheblich. Weniger als eine Stunde und 20 Minuten ist hier erforderlich. Dieser Wert liegt für die vollständige Vorgehensweise bei über 27 Stunden. Das ist nicht überraschend, da statt für jede der bis zu 36 Eingangsgrößen einen Wert zu optimieren hier lediglich ein einziger angepasst werden muss. Ob geringere  $R^2$  Werte bei der Approximation des Umformgrads bei deutlich verkürzter Laufzeit zu tolerieren sind, muss je nach Anwendungsfall entschieden werden. Für die Vorhersage der Schädigung sollte keine skalare Bandbreite verwendet werden.



**Abbildung 7.3:** Residualplots für die Ausgangsgrößen Umformgrad und Schädigung bei Vorhersage durch das beste Modell, das mit skalarer Bandbreite trainiert wurde.

### 7.3.2 Polynomielle Eingangsgrößen

Im Rahmen der erarbeiteten Vorgehensweise werden polynomielle Eingangsgrößen erzeugt. Bei diesem Vorgang wird aus jeweils zwei Größen mittels Multiplikation eine weitere berechnet. Ein Beispiel dafür ist in Abschnitt 2.2.4 dargestellt. Durch diesen Vorgang wird die Dimension des Raums der Eingangsgrößen von 7 auf 36 erhöht. Das kann eine bessere Approximation ermöglichen [KSS16; XZZ21b] und wurde durch AutoML-Verfahren zur Vorhersage der Ausgangsgrößen Umformgrad und Schädigung durchgeführt. Daher wurde dieser Schritt als Teil der erarbeiteten Vorgehensweise ausgewählt.

In diesem Abschnitt werden Modelle untersucht, die ohne die Erzeugung polynomieller Eingangsgrößen trainiert werden. Tabelle 7.3 zeigt die Ergebnisse, die mit solchen Modellen erreicht werden. Die  $R^2$  Werte zur Beurteilung der Vorhersagequalität auf dem Testdatensatz unterscheiden sich nicht von denen, die mithilfe der vollständigen Vorgehensweise erzielt werden. Auch der  $R_{KV5}^2$  Wert sinkt maximal um 0,01. Jedoch ist die Dauer für die Rastersuche um einiges kürzer. Ohne die Erzeugung polynomieller Eingangsgrößen benötigt diese weniger als 40 Minuten, eine Reduktion von über

	Umformgrad		Schädigung	
$R_{KV5}^2$	0,94	(-0,01)	0,71	(±0)
$R_{\text{test}}^2$	0,98	(±0)	0,99	(±0)
Dauer Rastersuche	2 280 s	(-98 280 s)	2 040 s	(-98 460 s)

**Tabelle 7.3:** Werte für die besten Modelle, wenn keine polynomiellen Eingangsgrößen gebildet werden. In Klammern stehen Veränderungen im Vergleich zur vollständigen Vorgehensweise.

97 % im Vergleich zum zeitlichen Aufwand für die Rastersuche bei Anwendung der vollständigen Vorgehensweise. Demzufolge ist es zu empfehlen, die Erzeugung polynomieller Eingangsgrößen für die Approximation der Ausgangsgrößen Umformgrad und Schädigung auszulassen.

Untersucht man weitere Modelle, die im Rahmen der Rastersuchen ohne polynomielle Eingangsgrößen trainiert wurden, fällt auf, dass für beide Ausgangsgrößen jeweils sechs von ihnen exakt die gleichen  $R_{KV5}^2$  Werte erreichen, wie sie in Tabelle 7.3 dargestellt sind. Diese Modelle unterscheiden sich in der Parametrisierung des Verfahrens der Dimensionsreduktion. Gemeinsam ist ihnen jedoch der Wert 7 für den Hyperparameter Dimensionsanzahl. Das ist auffallend, da bereits vor der Dimensionsreduktion lediglich sieben Eingangsgrößen vorliegen. Das bedeutet, dass die Anzahl der Dimensionen nicht reduziert wird. In einem solchen Fall hat dieses Verfahren keine Auswirkungen. Daraus lässt sich schließen, dass bei dem Verzicht auf die Erzeugung von polynomiellen Größen die besten Ergebnisse gerade mit den sieben originalen Eingangsgrößen erzielt werden. Ohne den Schritt polynomielle Eingangsgrößen ist es demnach optimal, auch den Schritt Dimensionsreduktion auszulassen. Dabei werden gleichwertige Approximationen erreicht wie mit der vollständigen Vorgehensweise, jedoch benötigt die Erzeugung der Modelle nur einen Bruchteil der Zeit.

### 7.3.3 Standardisierung

Im Rahmen der Standardisierung werden Eingangsgrößen so skaliert, dass sie den Mittelwert 0 und die Standardabweichung 1 haben. Eine solche Skalierung wird bei Anwendung von ML-Verfahren häufig eingesetzt [Mur22]. Die Ergebnisse, die mit der erarbeiteten Vorgehensweise, aber ohne diese Maßnahme erzielt werden können, werden in diesem Abschnitt diskutiert und sind in Tabelle 7.4 dargestellt.

Auffallend ist, dass die Werte der Metriken für die Vorhersagequalität bei der Approximation des Umformgrads mit denen identisch sind, die mithilfe der vollständigen Vorgehensweise erreicht wurden. Für die Ausgangsgröße Schädigung fallen die  $R^2$  Werte etwas schlechter aus. Dabei weicht der Wert der Kreuzvalidierung am stärksten ab und liegt 0,13 unter dem Vergleichswert. Dagegen ist nur ein deutlich geringerer Rückgang von 0,02 für  $R_{\text{test}}^2$  festzustellen. Folglich kann ohne Standardisierung der  $R^2$  Mittelwert einer Kreuzvalidierung nicht im gleichen Maße zur Annäherung des  $R^2$  Werts auf den Testdaten herangezogen werden wie mit Standardisierung. Die Approximation des Umformgrads ist davon allerdings nicht betroffen. Hier trennen  $R_{KV5}^2$  und  $R_{\text{test}}^2$  lediglich 0,03.

Tabelle 7.4 ist auch zu entnehmen, dass die Dauer der Rastersuche ohne Standardisierung deutlich unter den Werten für die vollständige Vorgehensweise liegt. Im Falle des Umformgrads werden etwa 49 % der Zeit eingespart. Für die Schädigung liegt die Verringerung der Laufzeit mit über 18

	Umformgrad		Schädigung	
$R_{KV5}^2$	0,95	(±0)	0,58	(−0,13)
$R_{\text{test}}^2$	0,98	(±0)	0,97	(−0,02)
Dauer Rastersuche	51 480 s	(−49 080 s)	33 120 s	(−67 380 s)

**Tabelle 7.4:** Werte für die besten Modelle, wenn die Eingangsgrößen nicht standardisiert werden. In Klammern stehen Veränderungen im Vergleich zur vollständigen Vorgehensweise.



Stunden bei rund 67 %. Die Verkürzung der Laufzeit kann dabei nicht durch die Laufzeitkomplexität der Rastersuche erklärt werden. Da die Standardisierung keine Hyperparameter besitzt, die optimiert werden, sinkt die Anzahl der trainierten Modelle nicht im Vergleich zur vollständigen Vorgehensweise. Auch die Standardisierung selbst hat für alle Trainingsdaten mit der maximalen Anzahl konstruierter Eingangsgrößen lediglich eine Laufzeit von wenigen Millisekunden. Eine weitergehende Untersuchung deckt auf, dass ohne Standardisierung die Zeit für das Training von Modellen im Durchschnitt sinkt. Diese Verringerungen stimmen prozentual mit den oben angegebenen Werten für die relative Veränderung der Dauer der Rastersuchen überein. Zusammenfassend kann festgestellt werden, dass die Standardisierung bei der Approximation der Ausgangsgröße Umformgrad ausgelassen werden kann. Für die Ausgangsgröße Schädigung muss im Einzelfall abgewogen werden, ob eine Verringerung der Laufzeit des Verfahrens die oben genannten Nachteile ausgleicht.

### 7.3.4 Dimensionsreduktion

Ein Verfahren zur Dimensionsreduktion wird eingesetzt, um die Anzahl der Eingangsgrößen zu optimieren. Das hier gewählte Verfahren erzeugt jeweils aus zwei euklidisch nahen Größen eine neue durch Aggregation. Es werden in diesem Abschnitt Modelle analysiert, die ohne eine Dimensionsreduktion erzeugt werden. Wie Tabelle 7.5 zeigt, hat dieses Vorgehen keinen nennenswerten Einfluss auf die Vorhersagequalität für beide getesteten Ausgangsgrößen. Lediglich die  $R_{KV5}^2$  Werte weichen von den Vergleichswerten ab, wenn auch nur um maximal 0,02. Die  $R_{test}^2$  Werte stimmen mit denen für die vollständige Vorgehensweise überein. Jedoch verändert es erheblich die Zeit, welche für die Rastersuche erforderlich ist. Die verringerte Laufzeit der Rastersuche lässt sich dadurch erklären, dass deutlich weniger Hyperparameterwerte getestet werden. Über die Hyperparameter aller Schritte der Pipeline hinweg werden 47 verschiedene Werte erprobt. 40 davon entfallen auf Hyperparameter des Verfahrens zur Dimensionsreduktion. Wird dieses Verfahren nicht ausgeführt, müssen dementsprechend lediglich Kombinationen von sieben Werten überprüft werden. Da der Aufwand polynomiell mit der Anzahl der Werte wächst, ist hier eine große Zeitersparnis zu erwarten. In der Tat kann die Rastersuche ohne Dimensionsreduktion in etwa 20 Minuten durchgeführt werden. Sie umfasst das Training von lediglich zehn Modellen. Für die vollständige Vorgehensweise sind es 2 100 Modelle, deren Training über 27 Stunden in Anspruch nimmt. Diese Erkenntnisse decken sich mit denen der Untersuchung polynomieller Eingangsgrößen in Abschnitt 7.3.2. Hierbei wurde bereits festgestellt, dass die Schritte zur Erzeugung polynomieller Eingangsgrößen und zur Dimensionsreduktion ausgelassen werden können. Dabei tritt keine Verringerung der Vorhersagequalität auf, aber der zeitliche Aufwand wird gesenkt. Werden beide Schritte aus der Vorgehensweise entfernt, so wird die Laufzeit sogar um 99,98 % auf unter zwei Minuten reduziert.

	Umformgrad		Schädigung	
$R_{KV5}^2$	0,94	(−0,01)	0,69	(−0,02)
$R_{test}^2$	0,98	(±0)	0,99	(±0)
Dauer Rastersuche	1 320 s	(−99 240 s)	1 080 s	(−99 420 s)

**Tabelle 7.5:** Werte für die besten Modelle, wenn keine Dimensionsreduktion durchgeführt wird. In Klammern stehen Veränderungen im Vergleich zur vollständigen Vorgehensweise.

		Schädigung	
$R_{KV5}^2$	0,75	(+0,04)	
$R_{test}^2$	0,98	(-0,01)	

**Tabelle 7.6:** Werte für das besten Modelle zur Vorhersage der Schädigung ohne Transformation der Ausgangsgröße. In Klammern stehen Veränderungen im Vergleich zur vollständigen Vorgehensweise.

### 7.3.5 Transformation der Ausgangsgröße

Die Transformation der Ausgangsgröße Schädigung dient der Veränderung der Werteverteilung in eine annähernde Normalverteilung. Als Transformationsfunktion wird der natürliche Logarithmus eingesetzt. Für den Umformgrad wird im Rahmen der vollständigen Vorgehensweise keine derartige Anpassung vorgenommen. Die Werteverteilung des Umformgrads ist weder links- noch rechtsschief und verändert sich nicht wesentlich, wenn eine Transformation wie für die Schädigung angewendet wird. In diesem Abschnitt wird untersucht, wie sich die Vorhersagequalität verändert, wenn die Ausgangsgröße Schädigung nicht transformiert wird. Die entsprechenden  $R^2$  Werte sind in Tabelle 7.6 dargestellt. Angaben zur Dauer erfolgen hier nicht, da keine zusätzliche Rastersuche durchgeführt wurde. Der Einfluss der Transformation der Ausgangsgröße auf die Dauer der Rastersuche wurde nicht untersucht. Stattdessen werden die Hyperparameter wie bei der vollständigen Vorgehensweise gewählt. Ohne Transformation ist der  $R^2$  Mittelwert der Kreuzvalidierung um 0,04 höher, das spiegelt sich jedoch nicht in einem besseren  $R_{test}^2$  Wert wider. Dieser sinkt dagegen um 0,01. Daraus kann gefolgert werden, dass die Transformation der Ausgangsgröße Schädigung einen positiven Einfluss auf die Vorhersagequalität des Surrogatmodells hat. Darüber hinaus hat sie den Effekt, dass  $R_{KV5}^2$  eine schlechtere Annäherung an  $R_{test}^2$  bietet.

### 7.3.6 Entfernung von Ausreißerdatenpunkten

Die Ausreißerdatenpunkte wurden ausführlich in Kapitel 4 untersucht. Letztendlich wurde entschieden, sie für das Training der vollständigen Vorgehensweise aus dem Datensatz zu entfernen. Die Auswirkungen davon, Modelle abweichend von der erarbeiteten Vorgehensweise auf Daten mit Ausreißerdatenpunkten zu trainieren, sind in Tabelle 7.7 beziffert.

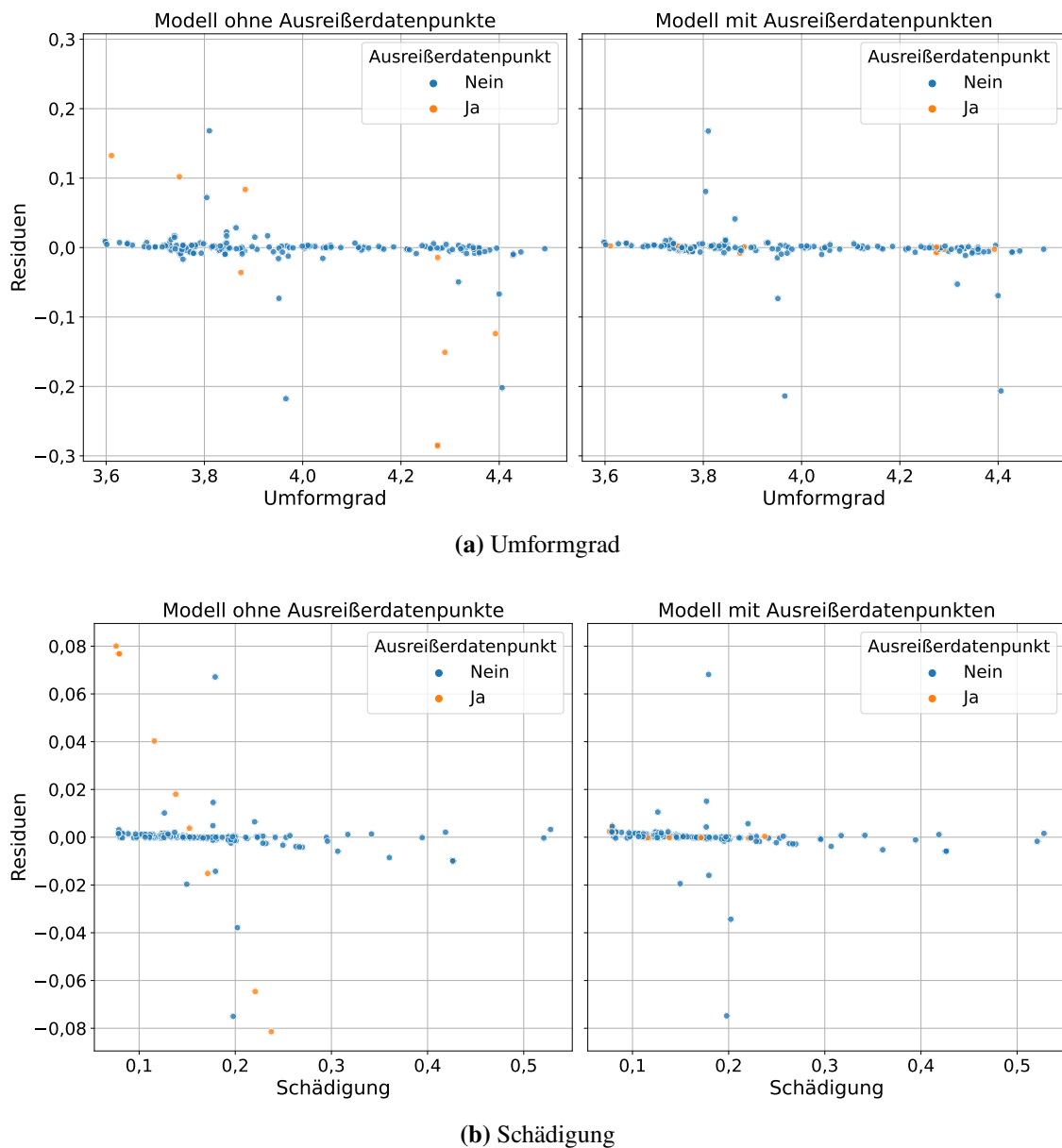
Auf die Vorhersagequalität haben die Ausreißerdatenpunkte in den Trainingsdaten offenbar kaum Einfluss. Der  $R^2$  Wert auf dem Testdatensatz, der keine Ausreißerdatenpunkte enthält, steigt bei Vorhersage des Umformgrads um 0,01 im Vergleich zur vollständigen Vorgehensweise. Ansonsten

	Umformgrad		Schädigung	
$R_{KV5}^2$	0,95	(±0)	0,71	(±0)
$R_{test}^2$	0,99	(+0,01)	0,99	(±0)
Dauer Rastersuche	131 880 s	(+31 320 s)	111 420 s	(+10 920 s)

**Tabelle 7.7:** Werte für die besten Modelle, die mit Ausreißerdatenpunkten trainiert wurden. In Klammern stehen Veränderungen im Vergleich zur vollständigen Vorgehensweise.

sind die Werte für die  $R^2$  Metrik unverändert, auch bei Approximation der Ausgangsgröße Schädigung. Die längere Dauer der Rastersuche kann über die höhere Anzahl an verwendeten Datenpunkten erklärt werden.

In Abbildung 7.4 sind Residualplots für die Vorhersage auf Testdaten mit Ausreißern dargestellt. Auf der linken Seite sind dabei jeweils die Residuen für ein Modell, das ohne Ausreißerdatenpunkte trainiert wurde, abgebildet. Analog wurden die Daten für die rechten Abbildungen durch Modelle erzeugt, die mit Ausreißerdatenpunkten trainiert wurden. Die blauen Datenpunkte repräsentieren die Residuen für Datenpunkte, die keine Ausreißer beinhalten. Im Gegensatz dazu sind in Orange die Residuen der Ausreißerdatenpunkte abgebildet. Abbildung 7.4a zeigt die Residuen bei Vorhersage



**Abbildung 7.4:** Residualplots für die Vorhersage auf Testdaten mit Ausreißerdatenpunkten.

der Ausgangsgröße Umformgrad. Hierbei ist zwischen den unterschiedlichen Modellen rechts und links für die blauen Punkte eine sehr ähnliche Verteilung zu erkennen. Auch einzelne, größere Residuen treten für dieselben Werte der Ausgangsgröße auf und haben den selben Betrag. Betrachtet man jedoch die in Orange dargestellten Ausreißerdatenpunkte, lassen sich Unterschiede feststellen. Das Modell, das ohne Ausreißerdatenpunkte trainiert wurde, zeigt relativ große Residuen für diese Punkte. Für geringe Werte des Umformgrads sind die Residuen positiv, für hohe Umformgrade eher negativ. Hier ist also ein Muster über die Ausreißerdatenpunkte erkennbar. Dagegen liegen die orangenen Residuen für das Modell, das mit Ausreißerdatenpunkten trainiert wurde, nahe 0. Ähnlich verhält es sich auch für die Ausgangsgröße Schädigung in Abbildung 7.4b. Die blauen Punkte sind für beide Modelle sehr ähnlich verteilt. Das Modell, das mit Ausreißerdatenpunkten trainiert wurde, zeigt für die orangenen Datenpunkte sehr geringe Residuen. Auf der linken Seite ist jedoch wieder ein lineares Muster für die Ausreißerdatenpunkte erkennbar. Hier ist es sogar noch deutlicher zu sehen als in Abbildung 7.4a. Das lässt vermuten, dass den Ausreißerdatenpunkten eine Systematik zugrunde liegt, die das Modell, welches ohne Ausreißer trainiert wird, nicht erfasst. Diese Systematik ist damit nicht aus den übrigen Datenpunkten abzuleiten. Diese Erkenntnis deckt sich mit den Ergebnissen der Untersuchung von Ausreißerdatenpunkten in Kapitel 4. Ob es für die Umformtechnik sinnvoll ist, dieses Muster zu lernen, kann an dieser Stelle nicht beurteilt werden. Auf die Vorhersage der übrigen Daten hat es nur wenig Einfluss, ob Ausreißerdatenpunkte im Training verwendet werden oder nicht.

### 7.4 Approximation anderer Ausgangsgrößen

Bisher wurden in diesem Kapitel Modelle zur Approximation des Umformgrads und der Schädigung untersucht. Dieser Fokus ist dadurch bedingt, dass für diese Ausgangsgrößen in Abschnitt 4.5 lediglich mit ML-Methoden eine hohe Vorhersagequalität erreicht wird. Für die übrigen, vom Fachexperten als besonders interessant gekennzeichneten Ausgangsgrößen können dagegen im Rahmen der explorativen Datenanalyse in Kapitel 4 bereits einfache Modelle abgeleitet werden, die eine hochqualitative Approximation ermöglichen. Für sie wird im Folgenden untersucht, ob auch die erarbeitete Vorgehensweise präzise Vorhersagen liefern kann. Die Ergebnisse dieser Untersuchung sind in Tabelle 7.8 abgebildet.

**Napfbodendicke** Die Napfbodendicke beschreibt, wie dick der Boden des Werkstücks nach dem Umformen ist. Abgesehen von Ausreißern, sind lediglich fünf unterschiedliche Werte dieser Ausgangsgröße in den Daten enthalten. Diese korrelieren perfekt mit den fünf Werten der Eingangsgröße Stempelgeometrie. Anhand einer Wertetabelle lässt sich die Napfbodendicke daher fehlerfrei vorhersagen. Mit dieser einfachen Herangehensweise kann ein  $R_{\text{test}}^2$  Wert von 1,00 erzielt werden. Wie in Tabelle 7.8 gezeigt wird, bringt auch die erarbeitete Vorgehensweise ein Modell hervor, das dieselbe Vorhersagequalität erreicht. Dieses Modell ist aus einer Rastersuche hervorgegangen. Zwar werden in dieser über elf Stunden lang Hyperparameter optimiert, aber der schlechteste erreichte  $R_{\text{KV5}}^2$  Wert liegt dabei bei 1. Eine Anpassung der Hyperparameter ist daher bei dieser Ausgangsgröße nicht notwendig. Aufgrund der geringeren Komplexität bei gleicher Vorhersagequalität ist das Modell auf Basis einer Wertetabelle dem Modell der erarbeiteten Vorgehensweise vorzuziehen.

	Napfbodendicke		Wandstärke		Temperatur		Kraft	
$R_{KV5}^2$	1,00		1,00		1,00		0,99	
$R_{test}^2$	1,00	(1,00)	1,00	(0,99)	1,00	(0,96)	1,00	(0,96)
Dauer Rastersuche	39 720 s		87 660 s		128 520 s		132 000 s	

**Tabelle 7.8:** Werte für die besten Modelle, die mit der erarbeiteten Vorgehensweise für die vier Ausgangsgrößen Napfbodendicke, Wandstärke, Temperatur und Kraft erzeugt werden können. In Klammern stehen Werte, die mit Modellen auf Basis von Wertetabellen erreicht werden (vgl. Abschnitt 4.5.1). Für diese stehen keine  $R_{KV5}^2$  Werte oder Dauern der Rastersuchen zur Verfügung.

**Wandstärke** Analog zur Napfbodendicke beschreibt die Wandstärke die Dicke der Wand des Werkstücks am Ende des Umformprozesses. Der Wert dieser Ausgangsgröße wird von den Eingangsgrößen Stempel- und Matrizengeometrie dominiert. Anhand dieser lässt sich die Wandstärke auf Basis einer Wertetabelle mit einem  $R_{test}^2$  Wert von 0,99 vorhersagen. Dasselbe Niveau erreicht auch ein Modell auf Basis der vollständigen Vorgehensweise. Wie in Tabelle 7.8 zu sehen, ist dabei  $R_{test}^2 = 1,00$ . Das Modell, welches diese Werte erreicht, nutzt durch eine Rastersuche optimierte Hyperparameterwerte. Diese Optimierung dauert über 24 Stunden, ist im Gegensatz zur Napfbodendicke jedoch notwendig. Der schlechteste, erfasste  $R_{KV5}^2$  Wert liegt etwa bei  $-4\,600$ . Die gesteigerte Komplexität des Modells der erarbeiteten Vorgehensweise resultiert lediglich in einer geringen Steigerung der Vorhersagequalität. Daher ist auch für die Ausgangsgröße Wandstärke das Modell auf Basis von Wertetabellen zu bevorzugen.

**Temperatur** Die maximal erreichte Temperatur während des Umformens ist in der Ausgangsgröße Temperatur enthalten. Anhand der Eingangsgröße Zyklus kann sie mit einem  $R_{test}^2$  Wert von 0,96 vorhergesagt werden. Dieses Ergebnis wird mit einem Modell auf Basis einer Wertetabelle erreicht. Das Modell der vollständigen Vorgehensweise kann auch noch die letzten 4 %-Punkte der Abweichung vom Mittelwert dieser Größe erklären und ist somit etwas besser. Für das Erreichen dieser Vorhersagequalität ist eine knapp 36 Stunden dauernde Rastersuche notwendig. Die erarbeitete Vorgehensweise erreicht eine bessere Vorhersagequalität als das Modell auf Basis von Wertetabellen. Daher sollte zur Approximation der Temperatur die erarbeitete Vorgehensweise gewählt werden, wenn ausreichend Zeit zur Durchführung aufgewendet werden kann.

**Kraft** In der Ausgangsgröße Kraft steht die maximale Kraft, die der Stempel im Umformprozess erfährt. Dieser ist beim Umformen großen Kräften ausgesetzt. Der Wert der entsprechenden Ausgangsgröße Kraft wird hauptsächlich durch das Werkstückmaterial bestimmt. Mit einer Wertetabelle mit Werten für das Werkstückmaterial und die Kraft kann letztere mit einem  $R_{test}^2$  Wert von 0,96 vorhergesagt werden. Wie für die Ausgangsgröße Temperatur ist das Modell der erarbeiteten Vorgehensweise 4 Prozentpunkte besser. Auch zur Erzeugung dieses Modells musste eine Rastersuche eingesetzt werden, die über 36 Stunden Rechenzeit in Anspruch genommen hat. Wie bei der Temperatur erreicht die erarbeitete Vorgehensweise für die Approximation der Ausgangsgröße Kraft eine bessere Vorhersagequalität als ein einfaches Modell auf Basis von Wertetabellen. Es

sollte daher die im Rahmen dieser Arbeit entwickelte Methodik angewendet werden. Wenn die Verkürzung der Laufzeit von hoher Bedeutung und eine geringere Vorhersagequalität akzeptabel ist, kann auch eine Wertetabelle als Modell verwendet werden.

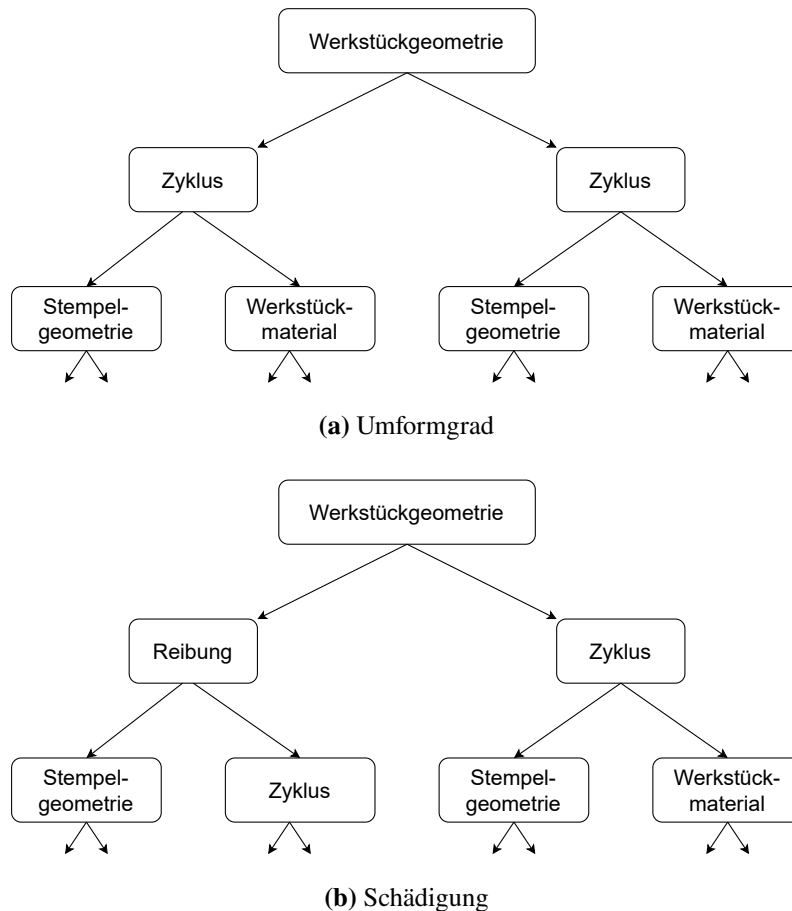
### 7.5 Erklärbarkeit

Im Rahmen dieser Arbeit wird die Erklärbarkeit der hervorgebrachten Modelle gesteigert. Für die vier Ausgangsgrößen Napfbodendicke, Wandstärke, Temperatur und Kraft sind bereits in Kapitel 4 Modelle mit hoher Vorhersagequalität erzeugt worden, die allein auf Korrelationen beruhen. Dies entspricht einem globalen ante-hoc Ansatz (vgl. Abschnitt 2.3) zur Steigerung der Erklärbarkeit, bei dem möglichst simple Modelle verwendet werden [RSK20]. Der Fokus liegt hier daher auf der Untersuchung des Einflusses von Eingangsgrößen auf die Vorhersage der Ausgangsgrößen Umformgrad und Schädigung. Zu diesem Zweck werden zunächst die Entscheidungsbäume aus Abschnitt 4.5.2 untersucht. Sie gelten aufgrund ihres nachvollziehbaren Aufbaus als erklärbare Modelle [BH21; DK17; TKE+16]. Im Anschluss wird eine Herangehensweise betrachtet, die von dem verwendeten Python-Modul scikit-learn zur Verfügung gestellt wird: Partial Dependence Plots. Zuletzt werden SHAP Werte berechnet und mit den Ergebnissen der anderen Verfahren verglichen. Zur Überprüfung von Erklärungen steht neben Nutzerstudien kein standardisiertes Verfahren zur Verfügung [BH21]. Eine Evaluation der verwendeten XAI-Verfahren bleibt daher aus.

#### 7.5.1 Interpretation der Entscheidungsbäume

Die wichtigsten Eingangsgrößen für die Vorhersage einer Ausgangsgröße sind daran zu erkennen, dass sie in einem Entscheidungsbaum zur Approximation der Ausgangsgröße auf niedriger Tiefe, das heißt nahe am Wurzelknoten, zur Aufteilung der Daten verwendet werden. Dieser Zusammenhang gilt auch für das Anwendungsgebiet der Fertigungssysteme [DB21]. Die Wahl der Eingangsgrößen auf niedrigen Tiefen des Baumes ist oft eindeutig, weshalb sie auch bei geringer Vorhersagequalität des Entscheidungsbaums aussagekräftig ist [Bre93]. Je geringer die Tiefe, auf der die Daten zum ersten Mal anhand einer Eingangsgröße aufgeteilt werden, desto wichtiger die Eingangsgröße für die Vorhersage [Fre14]. Dementsprechend werden die ersten drei von 13 Level der in Kapitel 4 erstellten Entscheidungsbäume untersucht. Diese sind in Abbildung 7.5 dargestellt.

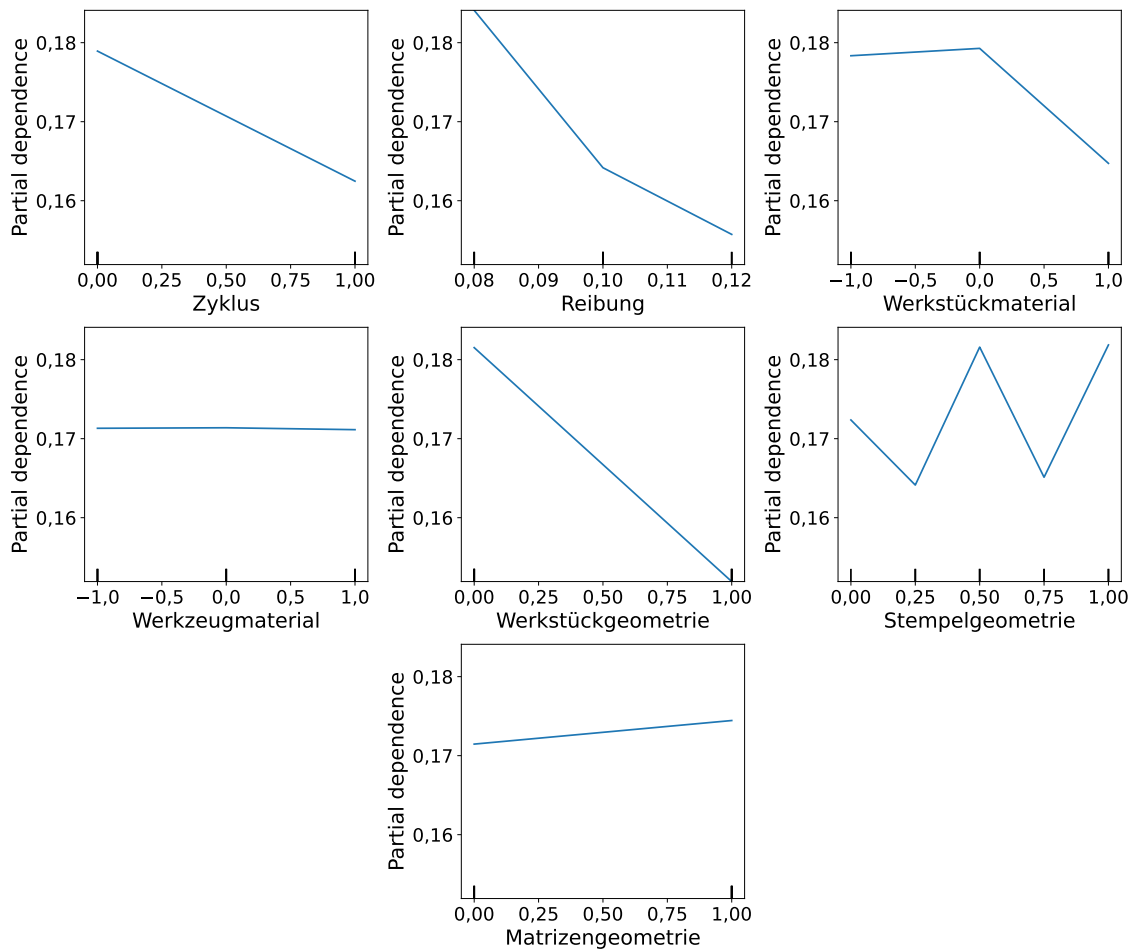
Für die Ausgangsgröße Umformgrad werden die Daten zunächst anhand der Größe Werkstückgeometrie aufgeteilt. Es folgt darauf der Zyklus und auf Tiefe 2 die Stempelgeometrie sowie das Werkstückmaterial. Das entspricht der Reihenfolge der Eingangsgrößen, wenn man diese nach ihrer absoluten Korrelation mit der Ausgangsgröße sortiert. Auch für die Schädigung wird der Datensatz zuerst anhand der Ausgangsgröße Werkstückgeometrie aufgeteilt. Es folgen die Reibung und der Zyklus auf Tiefe 1. Diese drei Eingangsgrößen haben die betragsmäßig höchsten Korrelationen mit der Schädigung. Auf dem nächsten Level wird vom Entscheidungsbaum zusätzlich zum Zyklus die Stempelgeometrie betrachtet. Für diese liegt der Korrelationskoeffizient bei 0,01, der zweitniedrigste Wert unter allen Eingangsgrößen. Der Grund dafür ist an dieser Stelle nicht direkt ersichtlich. Die Untersuchungen der nächsten Abschnitte geben jedoch mehr Aufschluss über den Zusammenhang zwischen der Eingangsgröße Stempelgeometrie und der Ausgangsgröße Schädigung.



**Abbildung 7.5:** Vereinfachte Darstellung der Entscheidungsbäume aus Abschnitt 4.5.2 für die Ausgangsgrößen Umformgrad und Schädigung bis zur Tiefe 2. Die Werte in den Knoten geben an, anhand welcher Eingangsgröße die Daten aufgeteilt werden.

### 7.5.2 Partial Dependence Plots

PDPs visualisieren die durchschnittliche Veränderung der Ausgangsgröße, wenn eine Eingangsgröße über ihren Wertebereich variiert wird. Anhand dieser Darstellung kann der Einfluss von Eingangsgrößen auf die Vorhersage der Ausgangsgröße analysiert werden. In Abbildung 7.6 sind PDPs für die Ausgangsgröße Umformgrad dargestellt. Es ist zu erkennen, dass die Eingangsgröße Werkstückgeometrie den größten Einfluss aller Eingangsgrößen auf den Umformgrad hat. Der entsprechende Graph ist eine Gerade mit der höchsten Steigung aller dargestellten Kurven. Jedoch kann der tatsächliche Zusammenhang auch nicht linear sein, denn es stehen lediglich zwei Werte für diese Eingangsgröße zur Verfügung. Der Zyklus, die Stempelgeometrie und das Werkstückmaterial sind nach der Werkstückgeometrie die einflussreichsten Größen. Das deckt sich mit den Korrelationen und den Erkenntnissen aus der Analyse der Entscheidungsbäume. Für die Ausgangsgrößen Reibung, Werkzeugmaterial und Matrizengeometrie ist der mittlere Einfluss auf die Ausgangsgröße gering. Die entsprechenden Graphen variieren kaum im Wert.



**Abbildung 7.6:** Partial Dependence Plots für die Ausgangsgröße Umformgrad.

Für die Schädigung sind PDPs in Abbildung 7.7 dargestellt. Auch für diese Ausgangsgröße ist ein starker Zusammenhang mit der Werkstückgeometrie erkennbar. Die Reibung hat ebenfalls großen Einfluss auf die Schädigung. Es folgen der Zyklus, die Stempelgeometrie und das Werkstückmaterial. Für die Stempelgeometrie fällt darüber hinaus auf, dass der Zusammenhang nicht monoton ist. Das erklärt auch, warum die Korrelation zwischen der Schädigung und der Stempelgeometrie mit 0,01 so gering ist, aber der Entscheidungsbaum trotzdem bereits auf Tiefe 2 die Daten anhand der Stempelgeometrie aufteilt. Der verwendete Korrelationskoeffizient nach Bravais-Pearson [FHK+16] erkennt lediglich lineare Zusammenhänge. Für die Ausgangsgrößen Werkzeugmaterial und Matrizengeometrie sind erneut keine starken Auswirkungen auf die Ausgangsgröße zu erkennen. Die vier Eingangsgrößen, die die größten Veränderungen der Ausgangsgröße Schädigung im PDP aufweisen, sind auch jene, die bei der Untersuchung der Entscheidungsbäume aufgefallen sind.



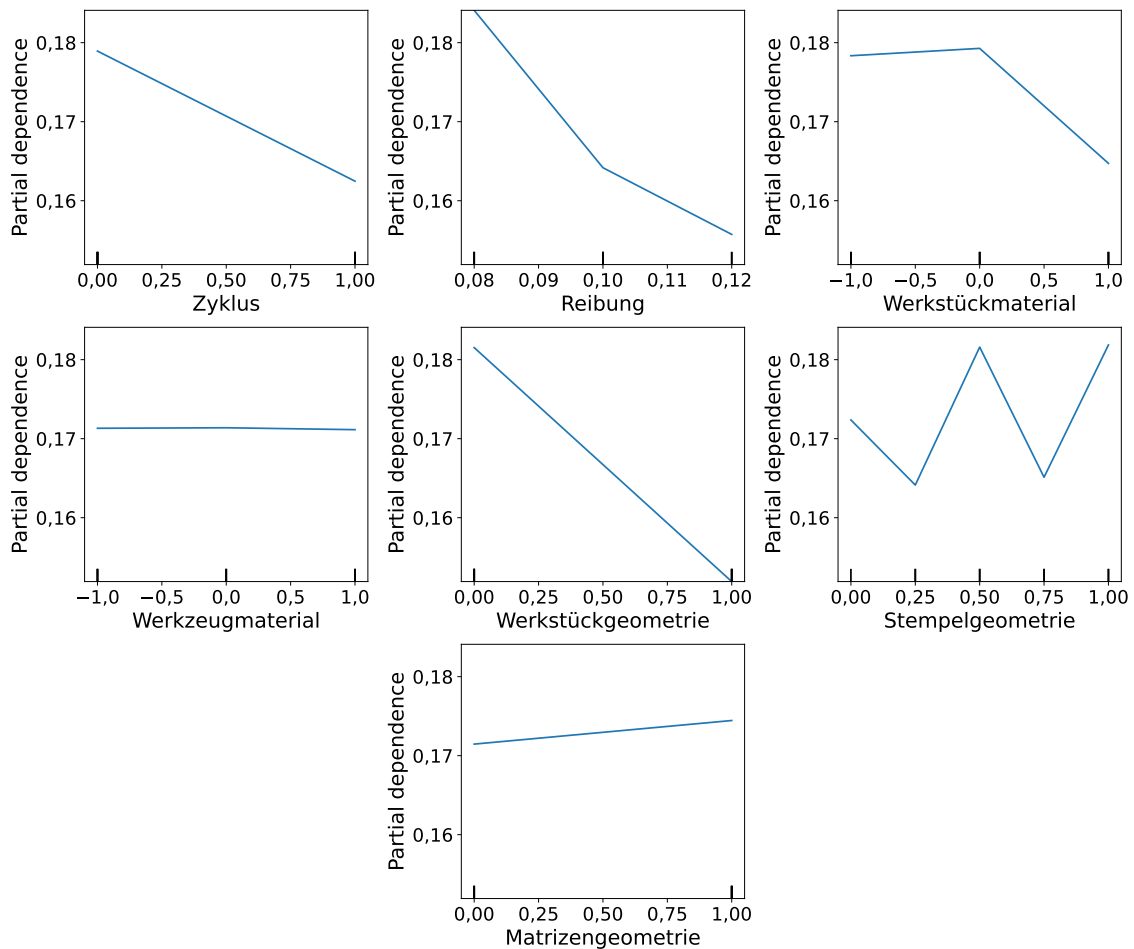


Abbildung 7.7: Partial Dependence Plots für die Ausgangsgröße Schädigung.

### 7.5.3 SHAP

Lokale Erklärungen für einzelne Vorhersagen eines Modells können erzeugt werden, indem in der näheren Umgebung eines spezifischen Datenpunktes ein lineares Modell erzeugt wird (vgl. Abschnitt 2.3). Unter den Vorgehensweisen, die solche linearen Modelle zur Steigerung der Erklärbarkeit trainieren, ist SHAP theoretisch optimal [LL17]. Anhand der lokalen Erklärungen, die dieses Verfahren produziert, kann auch eine globale Erklärung erzeugt werden [LEC+20]. Ähnlich wie für PDPs können die SHAP Werte gegen die Eingangsgrößen abgebildet werden. Im Fall dieser Arbeit ähneln diese Abbildungen denen der PDPs sehr, weshalb auf eine solche Darstellung verzichtet wird. Stattdessen werden die absoluten SHAP Werte über die Werte der Datenpunkte gemittelt, um so eine Kennzahl zu erhalten, die den durchschnittlichen, absoluten Einfluss einer Eingangsgröße auf die Ausgangsgröße quantifiziert. Diese Kennzahlen sind den absoluten Werten der Korrelationen (vgl. Tabelle 7.9) zwischen Eingangs- und Ausgangsgrößen in Abbildung 7.8 gegenübergestellt. Dabei ist zu beachten, dass der absolute Wert des Korrelationskoeffizienten unabhängig von der Skalierung der beteiligten Größen ist und im Bereich 0 bis 1 liegt. Der

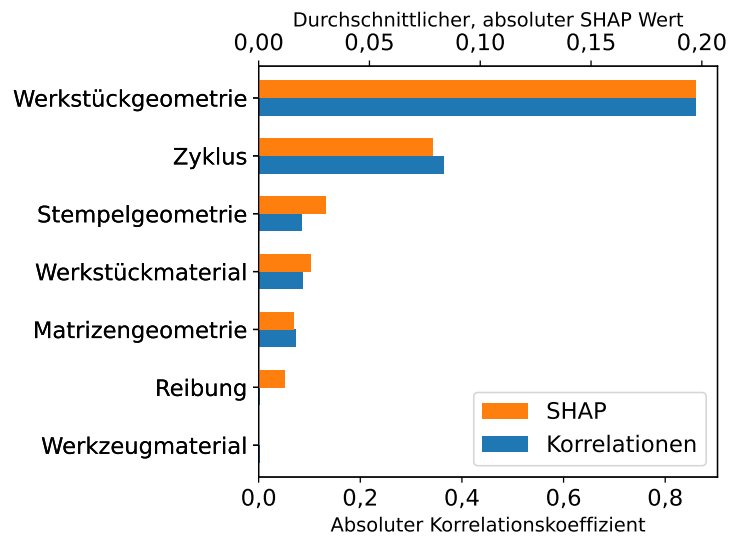
Eingangsgrößen	Korrelationen mit	
	Umformgrad	Schädigung
Zyklus	0,36	0,16
Reibung	0,00	0,17
Werkstückmaterial	0,09	0,08
Werkzeugmaterial	0,00	0,00
Werkstückgeometrie	0,86	0,20
Stempelgeometrie	0,09	0,01
Matrizengeometrie	0,07	0,03

**Tabelle 7.9:** Absolute Korrelationskoeffizienten zwischen Eingangsgrößen und den Ausgangsgrößen Umformgrad und Schädigung.

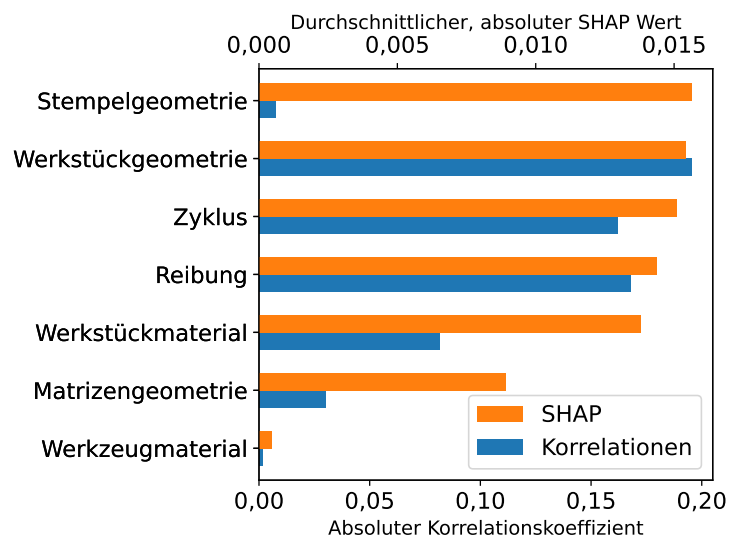
mittlere, absolute SHAP Wert dagegen gibt die absolute Veränderung der Ausgangsgröße in ihrem Wertebereich an. Daher ist die Skala für Korrelationen in den Abbildungen unten, die Skala für SHAP Werte oben angegeben.

Abbildung 7.8a kann entnommen werden, dass die absoluten Korrelationen der Ausgangsgröße Umformgrad mit den Eingangsgrößen annähernd proportional zu den SHAP Werten sind. Das ist für die Schädigung nicht der Fall. Wie in Abbildung 7.8b zu sehen ist, weicht die Verteilung der SHAP Werte von der Verteilung der Korrelationen mit den Eingangsgrößen ab. Insbesondere die Werte für die Eingangsgröße Stempelgeometrie fallen auf. Diese Größe verfügt über den höchsten durchschnittlichen, absoluten SHAP Wert von 0,02 bezüglich der Ausgangsgröße Schädigung. Die Korrelation zwischen der Schädigung und der Stempelgeometrie beträgt jedoch lediglich 0,01. Die höchste Korrelation der Ausgangsgröße Schädigung liegt für die Eingangsgröße Werkstückgeometrie vor hat einen absoluten Betrag von 0,20. Der SHAP Wert für die Werkstückgeometrie liegt jedoch unter dem der Stempelgeometrie. Auch die Untersuchungen der Entscheidungsbäume und der PDPs haben einen Einfluss der Stempelgröße auf die Ausgangsgröße Schädigung gezeigt. Jedoch war dabei nicht zu erkennen, dass die Stempelgeometrie im Mittel den größten Einfluss auf die Ausgangsgröße hat. Auch das Werkstückmaterial und die Matrizengeometrie haben gemessen an dem jeweiligen mittleren, absoluten SHAP Wert einen deutlich höheren Einfluss auf die Vorhersage der Ausgangsgröße Schädigung, als die Korrelationen zwischen Ein- und Ausgangsgröße vermuten lassen.

Zuletzt wird für die Ausgangsgrößen Napfbodendicke, Wandstärke, Temperatur und Kraft überprüft, wie sich die Korrelationen mit den Eingangsgrößen zu den jeweiligen SHAP Werten verhalten. Dabei werden die SHAP Werte anhand der Modelle berechnet, die für diese Ausgangsgrößen mit der erarbeiteten Vorgehensweise erzeugt wurden. Wie für den Umformgrad sind die Korrelationen und die SHAP Werte annähernd proportional. Damit decken sich die Ergebnisse zwischen den unterschiedlichen Methoden zur Steigerung der Erklärbarkeit größtenteils. Lediglich für die Ausgangsgröße Schädigung hat die Berechnung von SHAP Werten zusätzliche Erkenntnisse hervorgebracht. Für eine weitere Untersuchung der Zusammenhänge mit dieser Ausgangsgröße können darüber hinaus die lokalen Erklärungen, die das SHAP-Verfahren bereitstellt, verwendet werden. Das geht jedoch über den Rahmen der vorliegenden Arbeit hinaus.



(a) Umformgrad



(b) Schädigung

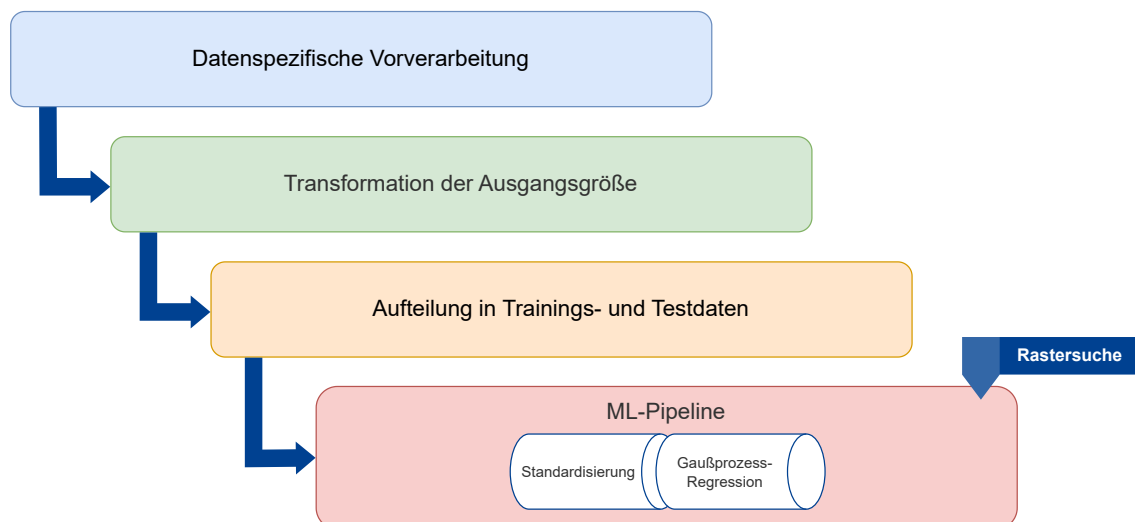
**Abbildung 7.8:** Gegenüberstellung von absoluten Korrelationskoeffizienten und durchschnittlichen, absoluten SHAP Werten.

## 7.6 Zentrale Schlussfolgerungen

Für die vier Ausgangsgrößen Napfbodendicke, Wandstärke, Temperatur und Kraft wird mit der erarbeiteten Vorgehensweise eine bestmögliche Approximation erreicht. Die Vorhersagen sind genauso gut oder besser als jene, die in Abschnitt 4.5.1 aus den Korrelationen zwischen Eingangs- und Ausgangsgrößen abgeleitet werden. Auch der Umformgrad und die Schädigung werden präzise approximiert. Die erzielten Vorhersagequalitäten übertreffen die der einfachen Modelle, die in Abschnitt 4.5.2 getestet wurden. Auch die angewendeten AutoML-Verfahren erreichen keine bessere Approximation. Die erarbeitete Vorgehensweise ist damit in der Lage, die vom Fachexperten als besonders interessant markierten Ausgangsgrößen der Simulation eines Kaltmassivumformprozesses präzise zu approximieren.

Aus der Ablationsstudie ergeben sich mögliche Anpassungen der Vorgehensweise. Die Erzeugung von polynomiellen Eingangsgrößen sowie die Dimensionsreduktion können ausgelassen werden. Das hat keine Reduktion der Vorhersagequalität zur Folge und verringert die Laufzeit der Vorgehensweise von über 27 Stunden auf unter zwei Minuten. Die Standardisierung hat einen positiven Einfluss auf die  $R^2$  Werte für mindestens eine Ausgangsgröße. Zwar kann durch Auslassen dieses Schritts ebenfalls die Laufzeit verringert werden, jedoch nicht im gleichen Umfang wie für die Erzeugung polynomieller Eingangsgrößen und die Dimensionsreduktion. Nur wenn eine zusätzliche Verringerung des zeitlichen Aufwands dringend erforderlich ist, sollte auf die Standardisierung verzichtet werden. Auch skalare Bandbreiten zu verwenden kann die Laufzeit zusätzlich verringern, ist aber aufgrund des starken negativen Effekts auf die Vorhersagequalität im Allgemeinen nicht zu empfehlen. Eine verkürzte Vorgehensweise mit sinnvollen Anpassungen ist in Abbildung 7.9 dargestellt.

Die Trainings- und Testdaten enthalten jeweils Ausreißerdatenpunkte. Als Teil der Vorgehensweise werden diese aus beiden Datensätzen entfernt. Im Anschluss kann ein Modell lediglich die Testdaten ohne Ausreißerdatenpunkte präzise approximieren. Werden diese Punkte wieder in den Testdatensatz eingefügt, weichen die Vorhersagen für die Ausreißerdatenpunkte relativ stark von den Daten



**Abbildung 7.9:** Durch im Rahmen der Ablationsstudie identifizierte Anpassungen abgewandelte, verkürzte Vorgehensweise.

Komponente	Simulation	Surrogatmodell
CPU Modell	Intel(R) Core(TM) i9-10900 @ 2,80 GHz	AMD EPYC(TM) 7003 @ 2,45 GHz
Anzahl logischer Prozessoren	20	4
Arbeitsspeicher	64 GB	8 GB

**Tabelle 7.10:** Vergleich der genutzten Hardware für Simulation und Surrogatmodell.

ab. Dagegen kann ein Modell, das mit Ausreißerdatenpunkten trainiert wurde, diese auch auf dem Testdatensatz gut approximieren. Daraus ist zu schließen, dass den Ausreißerdatenpunkten Zusammenhänge zugrunde liegen, die nicht aus den übrigen Daten abgeleitet werden.

Die teils hohe Laufzeit von über 27 Stunden der Vorgehensweise wird hauptsächlich für die Optimierung der Hyperparameter aufgewendet. Das Training des finalen Modells erfordert nur wenige Sekunden und die Dauer für Vorhersagen liegt im Bereich von Millisekunden. Dagegen benötigt die Simulation etwa zehn Minuten, um einen Datenpunkt zu erzeugen. Darüber hinaus läuft die Simulation auf deutlich leistungsstärkerer Hardware. Einen Vergleich der eingesetzten Computersysteme erlaubt Tabelle 7.10. Zur Durchführung der Simulation wird ein Prozessor mit höherem Takt eingesetzt. Auch stehen deutlich mehr logische Prozessoren und Arbeitsspeicher zur Verfügung als zur hier durchgeführten Anwendung der Vorgehensweise. Eine Verringerung der Laufzeit der Vorgehensweise unter Verwendung von mehr Prozessoren ist denkbar. Darüber hinaus erfordert die Anwendung der Simulation den Erwerb einer Lizenz. Dagegen sind die im Rahmen der Vorgehensweise eingesetzten Softwarekomponenten entgeltfrei zu verwenden, auch für kommerzielle Zwecke.

Mit XAI-Verfahren können globale Erklärungen für die Vorhersage der Ausgangsgrößen Umformgrad und Schädigung erzeugt werden. Für den Umformgrad ergeben die angewandten Methoden ähnliche Ergebnisse. Die Eingangsgrößen, die die höchsten Korrelationen mit der Ausgangsgröße aufweisen, haben auch den größten Einfluss auf die Vorhersage. Am einflussreichsten ist die Eingangsgröße Werkstückgeometrie. Für die Schädigung offenbart die Berechnung von SHAP Werten einen Zusammenhang, der mit den anderen Verfahren nicht zu erkennen ist. Die Stempelgeometrie hat im Mittel den größten absoluten Einfluss auf die Vorhersage dieser Ausgangsgröße. Dabei liegt der Betrag des Korrelationskoeffizienten für diese beiden Größen lediglich bei 0,01.



## 8 Zusammenfassung

In diesem Kapitel werden die wichtigsten Eckpunkte der Arbeit präsentiert. Nach einer Beschreibung der Problemstellung werden die durchgeführten Untersuchungen zusammengefasst und zentrale Ergebnisse angeführt. Zuletzt wird ein Ausblick darauf gegeben, welche Fragestellungen im Verlauf dieser Arbeit aufgekommen sind und zukünftig behandelt werden können.

### 8.1 Übersicht der Ergebnisse

Zur Optimierung von Fertigungsprozessen wie der Kaltmassivumformung werden häufig Simulationen eingesetzt. Diese ermöglichen eine präzise Vorhersage des Verhaltens der physischen Vorgänge [Bre07; CET21]. Jedoch können einzelne Simulationsdurchläufe mehrere Tage andauern [DMA17; PZL+18]. Schnellere Ergebnisse lassen sich mit Surrogatmodellen erreichen [CCC19; QSJ+05]. Im Rahmen dieser Arbeit wurde eine Vorgehensweise entwickelt, mit der datengetriebene Surrogatmodelle erzeugt werden, die die Ausgaben einer Kaltmassivumformsimulation approximieren. Sie wurde anhand der Daten einer Simulation eines Napf-Rückwärts-Fließpressen-Prozesses, der zu den Verfahren der Kaltmassivumformung gehört, erprobt. Ein Fachexperte hat sechs Ausgangsgrößen in den Daten als besonders interessant markiert: die Napfbodendicke, die Wandstärke, die Temperatur, die Kraft, den Umformgrad und die Schädigung.

Zu Approximation dieser Größen wurde zunächst eine explorative Datenanalyse (vgl. Kapitel 4) auf dem vom Institut für Umformtechnik der Universität Stuttgart zur Verfügung gestellten Datensatz durchgeführt. Eine solche Analyse gilt als essenziell für den erfolgreichen Einsatz von Methoden des maschinellen Lernens, wird aber in der Literatur nur selten beschrieben [KBT11]. Es wurden die Datenqualität, die Verteilung von Eingangs- und Ausgangsgrößen sowie die Korrelationen zwischen Größen untersucht. Daraus wurden Datenvorverarbeitungsschritte wie das Entfernen von Ausreißerdatenpunkten und erste, simple Modelle abgeleitet. Diese Modelle auf Basis von Korrelationen erreichen für die vier Ausgangsgrößen Napfbodendicke, Wandstärke, Temperatur und Kraft mit  $R^2$  Werten von 0,95 bis 1,00 genaue Vorhersagen. Daraufhin wurden einfache ML-Modelle wie die lineare Regression zu Approximation der anderen Ausgangsgrößen Umformgrad und Schädigung erprobt. Die lineare Regression erreicht für den Umformgrad eine hohe Vorhersagequalität. Für die Schädigung ist das jedoch nicht der Fall. Das kann an komplexen Zusammenhängen in den Daten liegen, die nur durch komplexere Methoden erfasst werden können [XZZ21b]. Daher wurden Verfahren des automatisierten maschinellen Lernens eingesetzt, um den Hypothesenraum und mögliche, weitere Vorverarbeitungsschritte zu erkunden. Mit diesen Verfahren kann auch die Ausgangsgröße Schädigung mit hoher Vorhersagequalität approximiert werden. Die AutoML-Modelle erreichen die zu diesem Punkt besten Vorhersagen und werden neben der Literatur als Grundlage für die eigene Vorgehensweise verwendet.

Die erarbeitete Vorgehensweise, die in Kapitel 5 beschrieben ist, besteht im Wesentlichen aus einer datenspezifischen Vorverarbeitung, der Transformation von Ausgangsgrößen, Verfahren des Feature Engineerings, der Standardisierung und einer Gaußprozess-Regression. Sie ermöglicht die Approximation aller sechs Ausgangsgrößen mit einem  $R^2$  Wert größer als 0,97. Diese Vorhersagen sind damit genauso gut oder besser als jene, die auf Korrelationen basieren. Auch mit einfachen ML-Modellen und AutoML-Verfahren werden keine besseren Ergebnisse erzielt. Zusätzlich sind die durch AutoML erzeugten Modelle komplexer als die der erarbeiteten Vorgehensweise. Diese anzuwenden hat jedoch deutlich länger gedauert als die Erzeugung von AutoML-Modellen. Im Rahmen einer Ablationsstudie wurde daher der Einfluss einzelner Schritte der Vorgehensweise auf die Vorhersagequalität und die Laufzeit untersucht. Dabei wurde festgestellt, dass ohne die Erzeugung von weiteren Eingangsgrößen und ohne eine anschließende Dimensionsreduktion die Laufzeit der Vorgehensweise bei gleichbleibender Vorhersagequalität um 94 % von über 27 Stunden auf unter zwei Minuten reduziert werden kann. Anschließende Vorhersagen dauern nur Millisekunden. Im Vergleich dazu benötigt die Simulation, auf deren Ausgaben Modelle trainiert werden, etwa zehn Minuten zur Erzeugung eines Datenpunktes. Die datengetriebenen Surrogatmodelle können daher den Raum der Prozessparameter deutlich effizienter erkunden als das mit der Simulation möglich ist.

Nach der Entwicklung der Vorgehensweise wurde die Erklärbarkeit der erzeugten Modelle durch XAI-Verfahren erhöht. Eine einfache und häufig eingesetzte Vorgehensweise dabei ist es, möglichst simple Modelle zu verwenden [RSK20]. Zu diesen gehören die lineare Regression und einfache Entscheidungsbäume. Solche wurden im Rahmen der explorativen Datenanalyse erzeugt und basieren auf den identifizierten Korrelationen zwischen Eingangs- und Ausgangsgrößen der Simulation. Darüber hinaus wurden komplexere XAI-Verfahren zur Behandlung der Modelle der erarbeiteten Vorgehensweise verwendet. Diese bestätigen die Erkenntnisse der Untersuchung der Korrelationen, können darüber hinaus jedoch zusätzliche, für die Vorhersage einer Ausgangsgröße relevante Eingangsgrößen identifizieren. Hierbei haben insbesondere Shapley additive explanations [LL17] weitere, quantifizierbare Erkenntnisse über die Zusammenhänge zwischen Eingangsgrößen und der Vorhersage der Ausgangsgröße Schädigung offenbart (vgl. Abschnitt 7.5.3).

Zusammenfassend lässt sich sagen, dass die im Rahmen dieser Arbeit entwickelte Vorgehensweise datengetriebene Surrogatmodelle hervorbringt, die die Ausgaben einer Kaltmassivumformung mit hoher Vorhersagequalität approximieren können. Mit den in Abschnitt 7.6 beschriebenen Anpassungen benötigen die Erzeugung eines Surrogatmodells und die Vorhersagen für über hundert Prozesskonfigurationen des Testdatensatzes weniger Zeit, als einen einzelnen Datenpunkt zu simulieren. Die Untersuchung der Erklärbarkeit der erzeugten Modelle liefert darüber hinaus Erkenntnisse über den Einfluss einzelner Eingangsgrößen auf die Vorhersagen.

## 8.2 Ausblick

Der in dieser Arbeit zur Erprobung der entwickelten Vorgehensweise verwendete Datensatz verfügt lediglich über 1 080 Datenpunkte. Die Zahl der zum Training von ML-Modellen verwendeten Daten liegt mit 822 noch darunter, da 15 % als Testdatensatz verwendet und Ausreißerdatenpunkte entfernt werden. Bei einem kleinen Datensatz wie diesem besteht die Gefahr der Überanpassung. Die Modelle bilden womöglich nicht die Zusammenhänge der Grundgesamtheit der Daten ab, sondern die spezifischen Eigenschaften der untersuchten Stichprobe. Anhand der Vorhersagequalität



auf einem größeren Datensatz kann das untersucht werden. Darüber hinaus enthalten die Eingangsgrößen in den Daten nur wenige unterschiedliche Werte. Da alle Kombinationen der getesteten Werte der Eingangsgrößen simuliert wurden, müssen bei einem größeren Datensatz auch weitere Werte einzelner Größen verwendet werden. Das ist insbesondere für die Eingangsgrößen Zyklus, Werkstückgeometrie und Matrizengeometrie angezeigt. Für diese sind im verwendeten Datensatz lediglich jeweils zwei Werte enthalten. Mehr Werte für diese Eingangsgrößen zu verwenden, kann weitere und komplexere Zusammenhänge offenbaren. Darüber hinaus wurden anhand der wenigen Werte dieser Eingangsgrößen Zusammenhänge mit einigen Ausgangsgrößen erkannt und genutzt, um einfache Modelle zu erzeugen (vgl. Abschnitt 4.5.1). Es ist von Interesse, zu untersuchen, ob diese Zusammenhänge auch für weitere Werte der Eingangsgrößen Bestand haben. Auch die Kodierung der Geometrien von Werkzeug und Werkstück in den Daten kann in zukünftigen Untersuchungen angepasst werden. Hier sind Stempel-, Matrizen- und Werkstückgeometrie in jeweils einer metrischen Eingangsgröße kodiert. Mit dieser Repräsentation sind viele Geometrien nicht darstellbar. Um das zu umgehen, können weitere Eingangsgrößen zur Beschreibung der Geometrien verwendet und getestet werden.

Eine weitere Auffälligkeit des zur Verfügung gestellten Datensatzes stellen die Ausreißer dar, die in Kapitel 4 beschrieben werden. Laut Fachexperte handelt es sich dabei um falsche Werte, die auf Fehler der Simulation zurückzuführen sind. Da bei einer erneuten Simulation eines Teils der Daten annähernd dieselben Fehler zu beobachten sind, ist auszuschließen, dass diese zufällig und unabhängig von den Eingangsgrößen auftreten. Ausreißerdatenpunkte enthalten Ausreißer in mindestens einer Ausgangsgröße. In Abschnitt 7.3.6 wird festgestellt, dass für diese Datenpunkte auch die Werte anderer Ausgangsgrößen ein Muster aufweisen, das von den übrigen Datenpunkten abweicht. An dieser Stelle empfiehlt sich eine weitere Untersuchung des Zustandekommens und der Auswirkungen der Ausreißer. Im Sinne einer Überprüfung der Korrektheit der Vorhersagen kommt auch eine Evaluation anhand von Daten physischer Experimente in Betracht. Dabei kann untersucht werden, ob ein Modell, das ohne Ausreißerdatenpunkte trainiert wird, Werte für Ausreißer vorhersagt, die den Ergebnissen der physischen Experimente nahekommen.

In dieser Arbeit werden anhand von Prozessparametern Eigenschaften des Produkts einer Kaltmassivumformung vorhergesagt. Anhand dieser Vorhersagen, die durch Simulationen oder Surrogatmodelle erzeugt werden, soll der Umformprozess optimiert werden. Zu diesem Zweck werden in der Praxis zahlreiche Vorhersagen für unterschiedliche Prozesskonfigurationen erzeugt und verglichen [FBS15]. Stattdessen ist es denkbar, Surrogatmodelle zu erzeugen, die anhand der gewünschten Qualitätsmerkmale des Endprodukts die dafür erforderlichen Prozessparameter ausgeben.

Einen anderen Bereich, in dem sich zusätzliche Untersuchungen anbieten, stellt die Steigerung der Erklärbarkeit durch weitere XAI-Methoden dar. Aufgrund des begrenzten Umfangs der vorliegenden Arbeit konnte darauf nicht umfassend eingegangen werden. So wurde lediglich der mittlere Einfluss von Eingangsgrößen auf die Vorhersage untersucht. Die berechneten SHAP Werte erlauben darüber hinaus lokale Erklärungen einzelner Vorhersagen, die für Fachexperten von Interesse sein können. Auch weitere XAI-Ansätze können verfolgt werden. Darüber hinaus empfiehlt sich eine Evaluation der erzeugten Erklärungen, beispielsweise durch Nutzerstudien.



# Literaturverzeichnis

- [AAM20] R. Alizadeh, J. K. Allen, F. Mistree. „Managing computational complexity using surrogate models: a critical review“. en. In: *Research in Engineering Design* 31.3 (Juli 2020), S. 275–298. ISSN: 1435-6066. DOI: [10.1007/s00163-020-00336-7](https://doi.org/10.1007/s00163-020-00336-7) (zitiert auf S. 15, 19, 20, 25).
- [AB18] A. Adadi, M. Berrada. „Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)“. In: *IEEE Access* 6 (2018), S. 52138–52160. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2018.2870052](https://doi.org/10.1109/ACCESS.2018.2870052) (zitiert auf S. 30–32, 44).
- [ACS22] A. Andrade-Campos, S. Coppieters, M. Strano. „Optimization and inverse analysis in metal forming: scientific state-of-the-art and recent trends“. In: *International Journal of Material Forming* 15.3 (Apr. 2022). DOI: [10.1007/s12289-022-01690-8](https://doi.org/10.1007/s12289-022-01690-8) (zitiert auf S. 15, 19, 20, 35).
- [Alb22] T. Alby. *Data Science in der Praxis: Eine verständliche Einführung in alle wichtigen Verfahren*. Bonn, GERMANY: Rheinwerk Verlag, 2022. ISBN: 9783836284646 (zitiert auf S. 21, 22, 41, 46, 49, 70).
- [BAC+19] F. E. Bock, R. C. Aydin, C. J. Cyron, N. Huber, S. R. Kalidindi, B. Klusemann. „A Review of the Application of Machine Learning and Data Mining Approaches in Continuum Materials Mechanics“. In: *Frontiers in Materials* 6 (Mai 2019). DOI: [10.3389/fmats.2019.00110](https://doi.org/10.3389/fmats.2019.00110) (zitiert auf S. 15, 20, 55, 73).
- [BCNA21] Á. Bárkányi, T. Chován, S. Németh, J. Abonyi. „Modelling for Digital Twins—Potential Role of Surrogate Models“. en. In: *Processes* 9.3 (März 2021), S. 476. ISSN: 2227-9717. DOI: [10.3390/pr9030476](https://doi.org/10.3390/pr9030476). URL: <https://www.mdpi.com/2227-9717/9/3/476> (zitiert auf S. 19, 20, 28, 71, 72).
- [BH21] N. Burkart, M. F. Huber. „A Survey on the Explainability of Supervised Machine Learning“. In: *J. Artif. Int. Res.* 70 (Mai 2021), S. 245–317. ISSN: 1076-9757. DOI: [10.1613/jair.1.12228](https://doi.org/10.1613/jair.1.12228) (zitiert auf S. 23, 30–33, 44, 47, 60, 70, 94).
- [BLFS16] A. Bustillo, L. N. López de Lacalle, A. Fernández-Valdivielso, P. Santos. „Data-mining modeling for the prediction of wear on forming-taps in the threading of steel components“. In: *Journal of Computational Design and Engineering* 3.4 (Okt. 2016), S. 337–348. ISSN: 2288-5048. DOI: [10.1016/j.jcde.2016.06.002](https://doi.org/10.1016/j.jcde.2016.06.002) (zitiert auf S. 26, 27, 36, 38, 39, 68, 72).
- [Bre07] D. Breuer. *Bestimmung des Formänderungsvermögens bei der Kaltmassivumformung*. 2007. ISBN: 9783832264277 (zitiert auf S. 15, 17, 19, 103).
- [Bre93] L. Breiman. *Classification and regression trees*. Chapman & Hall, 1993, S. 358. ISBN: 0412048418 (zitiert auf S. 57, 94).

- [Buh03] M. D. Buhmann. *Radial Basis Functions Theory and Implementations. Theory and Implementations*. Cambridge University Press, 2003. ISBN: 9780511040207 (zitiert auf S. 27).
- [BVV+18] R. Binns, M. Van Kleek, M. Veale, U. Lyngs, J. Zhao, N. Shadbolt. „It’s Reducing a Human Being to a Percentage’: Perceptions of Justice in Algorithmic Decisions“. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI ’18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, S. 1–14. ISBN: 9781450356206. DOI: [10.1145/3173574.3173951](https://doi.org/10.1145/3173574.3173951) (zitiert auf S. 30, 31).
- [CCA+15] C. Ciancio, T. Citrea, G. Ambrogio, L. Filice, R. Musmanno. „Design of a High Performance Predictive Tool for Forging Operation“. en. In: *Procedia CIRP*. 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME ’14 33 (Jan. 2015), S. 173–178. ISSN: 2212-8271. DOI: [10.1016/j.procir.2015.06.032](https://doi.org/10.1016/j.procir.2015.06.032). URL: <https://www.sciencedirect.com/science/article/pii/S2212827115006757> (zitiert auf S. 68).
- [CCC19] T. Chatterjee, S. Chakraborty, R. Chowdhury. „A Critical Review of Surrogate Assisted Robust Design Optimization“. en. In: *Archives of Computational Methods in Engineering* 26.1 (Jan. 2019), S. 245–274. ISSN: 1886-1784. DOI: [10.1007/s11831-017-9240-5](https://doi.org/10.1007/s11831-017-9240-5) (zitiert auf S. 15, 19, 20, 103).
- [CET21] S. Chabanet, H. B. El-Haouzi, P. Thomas. „Coupling digital simulation and machine learning metamodel through an active learning approach in Industry 4.0 context“. In: *Computers in Industry* 133 (Dez. 2021), S. 103529. DOI: [10.1016/j.compind.2021.103529](https://doi.org/10.1016/j.compind.2021.103529) (zitiert auf S. 15, 19, 73, 103).
- [Cho17] F. Chollet. *Deep Learning with Python*. Manning Publications, 2017, S. 384. ISBN: 9781617294433. URL: <https://www.manning.com/books/deep-learning-with-python> (zitiert auf S. 26, 73).
- [CHT09] A. K. Choudhary, J. A. Harding, M. K. Tiwari. „Data mining in manufacturing: a review based on the kind of knowledge“. en. In: *Journal of Intelligent Manufacturing* 20.5 (Okt. 2009), S. 501–521. ISSN: 1572-8145. DOI: [10.1007/s10845-008-0145-x](https://doi.org/10.1007/s10845-008-0145-x) (zitiert auf S. 21).
- [CK22] D. O. Chergykalo, D. A. Klyushin. „Fundamental Fallacies in Definitions of Explainable AI: Explainable to Whom and Why?“ In: *Explainable AI: Foundations, Methodologies and Applications*. Springer International Publishing, Okt. 2022, S. 25–42. DOI: [10.1007/978-3-031-12807-3\\_2](https://doi.org/10.1007/978-3-031-12807-3_2) (zitiert auf S. 31).
- [DB16] E. Doege, B.-A. Behrens. *Handbuch Umformtechnik*. Springer Berlin Heidelberg, 2016. DOI: [10.1007/978-3-662-43891-6](https://doi.org/10.1007/978-3-662-43891-6) (zitiert auf S. 15, 17, 18, 42).
- [DB21] A. Dogan, D. Birant. „Machine learning and data mining in manufacturing“. en. In: *Expert Systems with Applications* 166 (März 2021), S. 114060. ISSN: 0957-4174. DOI: [10.1016/j.eswa.2020.114060](https://doi.org/10.1016/j.eswa.2020.114060). URL: <https://www.sciencedirect.com/science/article/pii/S095741742030823X> (zitiert auf S. 20–22, 28, 43, 47, 68, 71, 72, 74, 94).
- [DBH18] F. K. Došilović, M. Brčić, N. Hlupić. „Explainable artificial intelligence: A survey“. In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Mai 2018, S. 0210–0215. DOI: [10.23919/MIPRO.2018.8400040](https://doi.org/10.23919/MIPRO.2018.8400040) (zitiert auf S. 30, 31).

- [DDL+20] B. Denkena, M.-A. Dittrich, M. Lindauer, J. Mainka, L. Stürenburg. *Using AutoML to Optimize Shape Error Prediction in Milling Processes*. en. Techn. Ber. 3724234. Rochester, NY, Nov. 2020. DOI: [10.2139/ssrn.3724234](https://doi.org/10.2139/ssrn.3724234). URL: <https://papers.ssrn.com/abstract=3724234> (zitiert auf S. 15, 18, 19, 36, 39, 58, 72, 74).
- [DIN8580] *DIN 8580:2022-12, Fertigungsverfahren\_- Begriffe, Einteilung*. Techn. Ber. Beuth Verlag GmbH. DOI: [10.31030/3217872](https://doi.org/10.31030/3217872). URL: <https://www.beuth.de/de/-/-/331759492> (zitiert auf S. 15, 17).
- [DIN8582] *DIN 8582:2003-09, Fertigungsverfahren Umformen\_- Einordnung; Unterteilung, Begriffe, Alphabetische Übersicht*. Techn. Ber. Beuth Verlag GmbH. DOI: [10.31030/9500643](https://doi.org/10.31030/9500643). URL: <https://www.beuth.de/de/-/-/65028536> (zitiert auf S. 15, 17).
- [DK17] F. Doshi-Velez, B. Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. Feb. 2017. DOI: [10.48550/ARXIV.1702.08608](https://doi.org/10.48550/ARXIV.1702.08608). URL: <https://www.semanticscholar.org/paper/Towards-A-Rigorous-Science-of-Interpretable-Machine-Doshi-Velez-Kim/5c39e37022661f81f79e481240ed9b175dec6513> (zitiert auf S. 30, 31, 94).
- [DMA17] S. Dey, T. Mukhopadhyay, S. Adhikari. „Metamodel based high-fidelity stochastic analysis of composite laminates: A concise review with critical comparative assessment“. en. In: *Composite Structures* 171 (Juli 2017), S. 227–250. ISSN: 0263-8223. DOI: [10.1016/j.compstruct.2017.01.061](https://doi.org/10.1016/j.compstruct.2017.01.061). URL: <https://www.sciencedirect.com/science/article/pii/S0263822316328793> (zitiert auf S. 15, 19, 20, 36, 39, 72, 103).
- [DRF+22] Y. Dai, K. Roy, Z. Fang, B. Chen, G. M. Raftery, J. B. P. Lim. „A novel machine learning model to predict the moment capacity of cold-formed steel channel beams with edge-stiffened and un-stiffened web holes“. en. In: *Journal of Building Engineering* 53 (Aug. 2022), S. 104592. ISSN: 2352-7102. DOI: [10.1016/j.jobe.2022.104592](https://doi.org/10.1016/j.jobe.2022.104592). URL: <https://www.sciencedirect.com/science/article/pii/S23527102222006052> (zitiert auf S. 20, 37, 39, 72, 81).
- [FBS15] N. Feldkamp, S. Bergmann, S. Strassburger. „Knowledge Discovery in Manufacturing Simulations“. In: *Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. SIGSIM PADS '15. New York, NY, USA: Association for Computing Machinery, Juni 2015, S. 3–12. ISBN: 9781450335836. DOI: [10.1145/2769458.2769468](https://doi.org/10.1145/2769458.2769468) (zitiert auf S. 19, 49, 68, 105).
- [FEF+22] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, F. Hutter. *Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning*. Techn. Ber. arXiv:2007.04074 [cs, stat] type: article. arXiv, Okt. 2022. DOI: [10.48550/arXiv.2007.04074](https://doi.org/10.48550/arXiv.2007.04074). arXiv: [2007.04074](https://arxiv.org/abs/2007.04074) (zitiert auf S. 72, 74).
- [FHK+16] L. Fahrmeir, C. Heumann, R. Künstler, I. Pigeot, G. Tutz. *Statistik. Der Weg zur Datenanalyse*. 8. Aufl. Springer Berlin Heidelberg, 2016. ISBN: 9783662503720. DOI: [10.1007/978-3-662-50372-0](https://doi.org/10.1007/978-3-662-50372-0) (zitiert auf S. 22, 23, 25, 29, 46, 50, 51, 63, 67, 69, 96).
- [FKE+15] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, F. Hutter. „Efficient and Robust Automated Machine Learning“. In: *Advances in Neural Information Processing Systems*. Bd. 28. Curran Associates, Inc., 2015. URL: [https://papers.nips.cc/paper\\_files/paper/2015/hash/11d0e6287202fced83f79975ec59a3a6-Abstract.html](https://papers.nips.cc/paper_files/paper/2015/hash/11d0e6287202fced83f79975ec59a3a6-Abstract.html) (zitiert auf S. 16, 26, 58, 59, 74).

- [FRC+21] Z. Fang, K. Roy, B. Chen, C.-W. Sham, I. Hajirasouliha, J. B. P. Lim. „Deep learning-based procedure for structural design of cold-formed steel channel sections with edge-stiffened and un-stiffened holes under axial compression“. en. In: *Thin-Walled Structures* 166 (Sep. 2021), S. 108076. ISSN: 0263-8231. DOI: [10.1016/j.tws.2021.108076](https://doi.org/10.1016/j.tws.2021.108076). URL: <https://www.sciencedirect.com/science/article/pii/S0263823121003888> (zitiert auf S. 37–39, 55, 72).
- [Fre04] A. A. Freitas. „A critical review of multi-objective optimization in data mining: a position paper“. In: *ACM SIGKDD Explorations Newsletter* 6.2 (Dez. 2004), S. 77–86. ISSN: 1931-0145. DOI: [10.1145/1046456.1046467](https://doi.org/10.1145/1046456.1046467) (zitiert auf S. 30).
- [Fre14] A. A. Freitas. „Comprehensible classification models: a position paper“. In: *ACM SIGKDD Explorations Newsletter* 15.1 (März 2014), S. 1–10. ISSN: 1931-0145. DOI: [10.1145/2594473.2594475](https://doi.org/10.1145/2594473.2594475) (zitiert auf S. 30, 94).
- [Fri01] J. H. Friedman. „Greedy Function Approximation: A Gradient Boosting Machine“. In: *The Annals of Statistics* 29.5 (2001), S. 1189–1232. ISSN: 0090-5364. DOI: [10.1214/aos/1013203451](https://doi.org/10.1214/aos/1013203451). URL: <https://www.jstor.org/stable/2699986> (zitiert auf S. 16).
- [Fri22] A. H. Fritz. „Umformen“. In: *Fertigungstechnik*. Springer Berlin Heidelberg, 2022, S. 141–234. DOI: [10.1007/978-3-662-64875-9\\_3](https://doi.org/10.1007/978-3-662-64875-9_3) (zitiert auf S. 15, 17, 18).
- [FSQ+17] J. Fang, G. Sun, N. Qiu, N. H. Kim, Q. Li. „On design optimization for structural crashworthiness and its state of the art“. en. In: *Structural and Multidisciplinary Optimization* 55.3 (März 2017), S. 1091–1119. ISSN: 1615-1488. DOI: [10.1007/s00158-016-1579-y](https://doi.org/10.1007/s00158-016-1579-y) (zitiert auf S. 26).
- [GC03] M. D. Giess, S. J. Culley. „Investigating Manufacturing Data for Use within Design“. en. In: *DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm* (2003), 193–194 (exec.summ.), full paper no. DS31\_1408FPB. URL: <https://www.designsociety.org/publication/24045/INVESTIGATING+MANUFACTURING+DATA+FOR+USE+WITHIN+DESIGN> (zitiert auf S. 20).
- [GE03] I. Guyon, A. Elisseeff. „An introduction to variable and feature selection“. In: *The Journal of Machine Learning Research* 3.null (März 2003), S. 1157–1182. ISSN: 1532-4435. URL: <https://dl.acm.org/doi/10.5555/944919.944968> (zitiert auf S. 55, 68, 71).
- [GFW+19] S. Gellrich, M.-A. Filz, J. Wölper, C. Herrmann, S. Thiede. „Data Mining Applications in Manufacturing of Lightweight Structures“. en. In: *Technologies for economical and functional lightweight design*. Hrsg. von K. Dröder, T. Vietor. Zukunftstechnologien für den multifunktionalen Leichtbau. Berlin, Heidelberg: Springer, 2019, S. 15–27. ISBN: 9783662582060. DOI: [10.1007/978-3-662-58206-0\\_2](https://doi.org/10.1007/978-3-662-58206-0_2) (zitiert auf S. 19, 21, 35, 39, 51, 71).
- [GKBP15] A. Goldstein, A. Kapelner, J. Bleich, E. Pitkin. „Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation“. In: *Journal of Computational and Graphical Statistics* 24.1 (Jan. 2015), S. 44–65. DOI: [10.1080/10618600.2014.907095](https://doi.org/10.1080/10618600.2014.907095) (zitiert auf S. 32).
- [GMR+18] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi. „A Survey of Methods for Explaining Black Box Models“. In: *ACM Computing Surveys* 51.5 (Aug. 2018), 93:1–93:42. ISSN: 0360-0300. DOI: [10.1145/3236009](https://doi.org/10.1145/3236009) (zitiert auf S. 32).

- [Han12] J. Han. *Data mining concepts and techniques. concepts and techniques*. 3. Aufl. Elsevier, 2012, S. 703. ISBN: 9780123814791. DOI: [10.1016/c2009-0-61819-5](https://doi.org/10.1016/c2009-0-61819-5) (zitiert auf S. 21–23, 41, 44, 48, 49, 51, 57, 72).
- [HBFP22] S. Huzni, I. B.M. Ibrahim, S. Fonna, R. Pidaparti. „Physics-based surrogate model for reinforced concrete corrosion simulation“. en. In: *Results in Engineering* 16 (Dez. 2022), S. 100659. ISSN: 2590-1230. DOI: [10.1016/j.rineng.2022.100659](https://doi.org/10.1016/j.rineng.2022.100659). URL: <https://www.sciencedirect.com/science/article/pii/S2590123022003292> (zitiert auf S. 19).
- [HCN22] H. Hema, H. G. N. Chakravarthy, S. R. Naganna. „Prediction of ultimate load carrying capacity of short cold-formed steel built-up lipped channel columns using machine learning approach“. en. In: *Sādhanā* 47.4 (Okt. 2022), S. 207. ISSN: 0973-7677. DOI: [10.1007/s12046-022-01979-z](https://doi.org/10.1007/s12046-022-01979-z) (zitiert auf S. 37, 39, 81).
- [HK70] A. E. Hoerl, R. W. Kennard. „Ridge Regression: Biased Estimation for Nonorthogonal Problems“. In: *Technometrics* 12.1 (Feb. 1970), S. 55–67. DOI: [10.1080/00401706.1970.10488634](https://doi.org/10.1080/00401706.1970.10488634) (zitiert auf S. 26).
- [HRM19] V. Hirsch, P. Reimann, B. Mitschang. „Data-Driven Fault Diagnosis in End-of-Line Testing of Complex Products“. In: *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, Okt. 2019. DOI: [10.1109/dsaa.2019.00064](https://doi.org/10.1109/dsaa.2019.00064) (zitiert auf S. 72).
- [HRM20] V. Hirsch, P. Reimann, B. Mitschang. „Exploiting domain knowledge to address multi-class imbalance and a heterogeneous feature space in classification tasks for manufacturing data“. In: *Proceedings of the VLDB Endowment* 13.12 (Aug. 2020), S. 3258–3271. DOI: [10.14778/3415478.3415549](https://doi.org/10.14778/3415478.3415549) (zitiert auf S. 52, 55).
- [HSSK05] J. A. Harding, M. Shahbaz, Srinivas, A. Kusiak. „Data Mining in Manufacturing: A Review“. In: *Journal of Manufacturing Science and Engineering* 128.4 (Dez. 2005), S. 969–976. ISSN: 1087-1357. DOI: [10.1115/1.2194554](https://doi.org/10.1115/1.2194554) (zitiert auf S. 15, 20, 21, 38).
- [HYDF16] K. Hansson, S. Yella, M. Dougherty, H. Fleyeh. „Machine Learning Algorithms in Heavy Process Manufacturing“. en. In: *American Journal of Intelligent Systems* 6.1 (2016), S. 1–13. ISSN: 2165-8994. URL: <http://article.sapub.org/10.5923.j.ajis.20160601.01.html> (zitiert auf S. 18, 51, 71).
- [ICFQ93] K. Irani, J. Cheng, U. Fayyad, Z. Qian. „Applying machine learning to semiconductor manufacturing“. In: *IEEE Expert* 8.1 (Feb. 1993), S. 41–47. ISSN: 2374-9407. DOI: [10.1109/64.193054](https://doi.org/10.1109/64.193054) (zitiert auf S. 20, 35, 39, 42).
- [JCL16] H. Jia, Y.-m. Cheung, J. Liu. „A New Distance Metric for Unsupervised Learning of Categorical Data“. In: *IEEE Transactions on Neural Networks and Learning Systems* 27.5 (Mai 2016), S. 1065–1079. ISSN: 2162-2388. DOI: [10.1109/TNNLS.2015.2436432](https://doi.org/10.1109/TNNLS.2015.2436432) (zitiert auf S. 68).
- [KBT11] G. Köksal, İ. Batmaz, M. C. Testik. „A review of data mining applications for quality improvement in manufacturing industry“. en. In: *Expert Systems with Applications* 38.10 (Sep. 2011), S. 13448–13467. ISSN: 0957-4174. DOI: [10.1016/j.eswa.2011.04.063](https://doi.org/10.1016/j.eswa.2011.04.063). URL: <https://www.sciencedirect.com/science/article/pii/S0957417411005793> (zitiert auf S. 18, 19, 22, 39, 47, 53, 57, 68, 71, 72, 103).



- [KCL21] C. Karadogan, P. Cyron, M. Liewald. „Potential use of machine learning to determine yield locus parameters“. In: *IOP Conference Series: Materials Science and Engineering* 1157.1 (Juni 2021), S. 012064. DOI: [10.1088/1757-899x/1157/1/012064](https://doi.org/10.1088/1757-899x/1157/1/012064) (zitiert auf S. 15, 18, 19, 35, 72).
- [KM06] L. A. Kurgan, P. Musilek. „A survey of Knowledge Discovery and Data Mining process models“. en. In: *The Knowledge Engineering Review* 21.1 (März 2006), S. 1–24. ISSN: 1469-8005, 0269-8889. DOI: [10.1017/S0269888906000737](https://doi.org/10.1017/S0269888906000737). URL: <https://www.cambridge.org/core/journals/knowledge-engineering-review/article/survey-of-knowledge-discovery-and-data-mining-process-models/368D6AFE435EB5E30378398D34D61C17> (zitiert auf S. 44).
- [KM22] D. Kumar, M. A. Mehta. „An Overview of Explainable AI Methods, Forms and Frameworks“. In: *Explainable AI: Foundations, Methodologies and Applications*. Springer International Publishing, Okt. 2022, S. 43–59. DOI: [10.1007/978-3-031-12807-3\\_3](https://doi.org/10.1007/978-3-031-12807-3_3) (zitiert auf S. 31, 73).
- [KSS16] G. Katz, E. C. R. Shin, D. Song. „ExploreKit: Automatic Feature Generation and Selection“. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. ISSN: 2374-8486. Dez. 2016, S. 979–984. DOI: [10.1109/ICDM.2016.0123](https://doi.org/10.1109/ICDM.2016.0123) (zitiert auf S. 70, 87).
- [Küh22] K.-D. Kühn. „Einteilung der Fertigungsverfahren“. In: *Fertigungstechnik*. Springer Berlin Heidelberg, 2022, S. 1–6. DOI: [10.1007/978-3-662-64875-9\\_1](https://doi.org/10.1007/978-3-662-64875-9_1) (zitiert auf S. 17).
- [LEC+20] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee. „From local explanations to global understanding with explainable AI for trees“. In: *Nature Machine Intelligence* 2.1 (Jan. 2020), S. 56–67. DOI: [10.1038/s42256-019-0138-9](https://doi.org/10.1038/s42256-019-0138-9) (zitiert auf S. 97).
- [LFM17] M. Liewald, A. Felde, R. Meißner. „Kaltmassivumformung von Stahl“. In: *Handbuch Stahl*. Hrsg. von W. Bleck, E. Moeller. Carl Hanser Verlag GmbH & Co. KG, Nov. 2017. DOI: [10.3139/9783446449626](https://doi.org/10.3139/9783446449626) (zitiert auf S. 15, 18).
- [Lip18] Z. C. Lipton. „The mythos of model interpretability“. In: *Communications of the ACM* 61.10 (Sep. 2018), S. 36–43. ISSN: 0001-0782. DOI: [10.1145/3233231](https://doi.org/10.1145/3233231) (zitiert auf S. 30–33).
- [LL17] S. M. Lundberg, S.-I. Lee. „A unified approach to interpreting model predictions“. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., Dez. 2017, S. 4768–4777. ISBN: 9781510860964. URL: <https://dl.acm.org/doi/10.5555/3295222.3295230> (zitiert auf S. 16, 30–33, 38, 97, 104).
- [Mil19] T. Miller. „Explanation in artificial intelligence: Insights from the social sciences“. en. In: *Artificial Intelligence* 267 (Feb. 2019), S. 1–38. ISSN: 0004-3702. DOI: [10.1016/j.artint.2018.07.007](https://doi.org/10.1016/j.artint.2018.07.007). URL: <https://www.sciencedirect.com/science/article/pii/S0004370218305988> (zitiert auf S. 30–32).



- [MPC22] M. Mehta, V. Palade, I. Chatterjee, Hrsg. *Explainable AI: Foundations, Methodologies and Applications*. Springer International Publishing, Okt. 2022. 256 S. ISBN: 978-3-031-12807-3. DOI: [10.1007/978-3-031-12807-3](https://doi.org/10.1007/978-3-031-12807-3). URL: [https://www.ebook.de/de/product/44679829/explainable\\_ai\\_foundations\\_methodologies\\_and\\_applications.html](https://www.ebook.de/de/product/44679829/explainable_ai_foundations_methodologies_and_applications.html) (zitiert auf S. 16).
- [Mur22] K. P. Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022, S. 848. ISBN: 9780262046824. URL: <http://probml.ai> (zitiert auf S. 24, 26–29, 41, 44, 49, 51, 55, 60, 70–74, 76, 88).
- [NM15] I. E. Naqa, M. J. Murphy. „What Is Machine Learning?“ In: *Machine Learning in Radiation Oncology*. Springer International Publishing, 2015, S. 3–11. DOI: [10.1007/978-3-319-18305-3\\_1](https://doi.org/10.1007/978-3-319-18305-3_1) (zitiert auf S. 22, 25).
- [OCL+20] J. P. Ono, S. Castelo, R. Lopez, E. Bertini, J. Freire, C. Silva. *PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines*. 2020. DOI: [10.48550/ARXIV.2005.00160](https://doi.org/10.48550/ARXIV.2005.00160) (zitiert auf S. 74).
- [PA05] D. T. Pham, A. A. Afify. „Machine-learning techniques and their applications in manufacturing“. en. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 219.5 (Mai 2005), S. 395–412. ISSN: 0954-4054. DOI: [10.1243/095440505X32274](https://doi.org/10.1243/095440505X32274) (zitiert auf S. 15, 20, 26, 44).
- [PGK+23] P. M. Petkar, V. N. Gaitonde, V. N. Kulkarni, R. S. Karnik, J. P. Davim. „A Comparative Study in Forming Behavior of Different Grades of Steel in Cold Forging Backward Extrusion by Integrating Artificial Neural Network (ANN) with Differential Evolution (DE) Algorithm“. en. In: *Applied Sciences* 13.3 (Jan. 2023), S. 1276. ISSN: 2076-3417. DOI: [10.3390/app13031276](https://doi.org/10.3390/app13031276). URL: <https://www.mdpi.com/2076-3417/13/3/1276> (zitiert auf S. 15, 17, 18, 20, 38, 39, 42, 70, 72).
- [Pla21] M. Plaeue. *Data Science*. Springer Berlin Heidelberg, 2021. DOI: [10.1007/978-3-662-63489-9](https://doi.org/10.1007/978-3-662-63489-9) (zitiert auf S. 22–25, 44–48, 76).
- [PS09] S. Piramuthu, R. T. Sikora. „Iterative feature construction for improving inductive learning algorithms“. In: *Expert Systems with Applications* 36.2 (März 2009), S. 3401–3406. DOI: [10.1016/j.eswa.2008.02.010](https://doi.org/10.1016/j.eswa.2008.02.010) (zitiert auf S. 70, 71).
- [PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830. URL: <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html> (zitiert auf S. 37, 56, 74).
- [PZL+18] J. Pfrommer, C. Zimmerling, J. Liu, L. Kärger, F. Henning, J. Beyerer. „Optimisation of manufacturing process parameters using deep neural networks as surrogate models“. en. In: *Procedia CIRP*. 51st CIRP Conference on Manufacturing Systems 72 (Jan. 2018), S. 426–431. ISSN: 2212-8271. DOI: [10.1016/j.procir.2018.03.046](https://doi.org/10.1016/j.procir.2018.03.046). URL: <https://www.sciencedirect.com/science/article/pii/S221282711830146X> (zitiert auf S. 19, 20, 36, 39, 72, 103).
- [Q SJ+05] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, C. F. Jeff Wu. „Building Surrogate Models Based on Detailed and Approximate Simulations“. In: *Journal of Mechanical Design* 128.4 (Juni 2005), S. 668–677. ISSN: 1050-0472. DOI: [10.1115/1.2179459](https://doi.org/10.1115/1.2179459) (zitiert auf S. 15, 19, 20, 72, 103).

- [Ras06] C. E. Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006. ISBN: 026218253X. URL: <https://gaussianprocess.org/gpml/> (zitiert auf S. 27, 28, 71, 72).
- [RPA+23] K. Rajneesh, V. E. Parvathi, S. Aswin, V. Aswin, A. Anisha, P. J. Usman Arshad, S. Mangalathu, R. Davis. „Prediction of axial load capacity of cold formed lipped channel section using machine learning“. en. In: *Structures* 50 (Apr. 2023), S. 1429–1446. ISSN: 2352-0124. DOI: [10.1016/j.istruc.2023.02.102](https://doi.org/10.1016/j.istruc.2023.02.102). URL: <https://www.sciencedirect.com/science/article/pii/S2352012423002606> (zitiert auf S. 28, 32, 37–39, 74, 81).
- [RSG16a] M. T. Ribeiro, S. Singh, C. Guestrin. „"Why Should I Trust You?": Explaining the Predictions of Any Classifier“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, S. 1135–1144. ISBN: 9781450342322. DOI: [10.1145/2939672.2939778](https://doi.org/10.1145/2939672.2939778) (zitiert auf S. 30).
- [RSG16b] M. T. Ribeiro, S. Singh, C. Guestrin. *Model-Agnostic Interpretability of Machine Learning*. Techn. Ber. arXiv:1606.05386 [cs, stat] type: article. arXiv, Juni 2016. DOI: [10.48550/arXiv.1606.05386](https://doi.org/10.48550/arXiv.1606.05386). arXiv: [1606.05386](https://arxiv.org/abs/1606.05386) (zitiert auf S. 30, 32).
- [RSK20] A. Rasheed, O. San, T. Kvamsdal. „Digital Twin: Values, Challenges and Enablers From a Modeling Perspective“. In: *IEEE Access* 8 (2020), S. 21980–22012. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2020.2970143](https://doi.org/10.1109/ACCESS.2020.2970143) (zitiert auf S. 19, 20, 30, 32, 64, 94, 104).
- [Rus19] C. Russell. „Efficient Search for Diverse Coherent Explanations“. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. FAT\* '19. New York, NY, USA: Association for Computing Machinery, Jan. 2019, S. 20–28. ISBN: 9781450361255. DOI: [10.1145/3287560.3287569](https://doi.org/10.1145/3287560.3287569) (zitiert auf S. 30, 32).
- [SF12] R. E. Schapire, Y. Freund. *Boosting: Foundations and Algorithms*. The MIT Press, Mai 2012. ISBN: 9780262301183. DOI: [10.7551/mitpress/8291.001.0001](https://doi.org/10.7551/mitpress/8291.001.0001) (zitiert auf S. 26).
- [SGV98] C. Saunders, A. Gammernan, V. Vovk. „Ridge Regression Learning Algorithm in Dual Variables“. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Juli 1998, S. 515–521. ISBN: 9781558605565. URL: <https://dl.acm.org/doi/10.5555/645527.657464> (zitiert auf S. 26).
- [STV04] B. Schölkopf, K. Tsuda, J.-P. Vert. „A Primer on Kernel Methods“. In: *Kernel Methods in Computational Biology*. The MIT Press, 2004, S. 35–70. DOI: [10.7551/mitpress/4057.003.0004](https://doi.org/10.7551/mitpress/4057.003.0004) (zitiert auf S. 26–28, 49, 72).
- [TGH+18] H. Tercan, A. Guajardo, J. Heinisch, T. Thiele, C. Hopmann, T. Meisen. „Transfer-Learning: Bridging the Gap between Real and Simulation Data for Machine Learning in Injection Molding“. In: *Procedia CIRP* 72 (2018), S. 185–190. DOI: [10.1016/j.procir.2018.03.087](https://doi.org/10.1016/j.procir.2018.03.087) (zitiert auf S. 19, 21).

- [TKE+16] H. Tercan, T. A. Khawli, U. Eppelt, C. Büscher, T. Meisen, S. Jeschke. „Use of Classification Techniques to Design Laser Cutting Processes“. en. In: *Procedia CIRP*. The Sixth International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2016) 52 (Jan. 2016), S. 292–297. ISSN: 2212-8271. DOI: [10.1016/j.procir.2016.08.001](https://doi.org/10.1016/j.procir.2016.08.001). URL: <https://www.sciencedirect.com/science/article/pii/S2212827116308435> (zitiert auf S. 20, 30, 38, 94).
- [TKE+17] H. Tercan, T. A. Khawli, U. Eppelt, C. Büscher, T. Meisen, S. Jeschke. „Improving the laser cutting process design by machine learning techniques“. en. In: *Production Engineering* 11.2 (Apr. 2017), S. 195–203. ISSN: 1863-7353. DOI: [10.1007/s11740-017-0718-7](https://doi.org/10.1007/s11740-017-0718-7) (zitiert auf S. 18–20).
- [VGG21] F. A. C. Viana, C. Gogu, T. Goel. „Surrogate modeling: tricks that endured the test of time and some recent developments“. en. In: *Structural and Multidisciplinary Optimization* 64.5 (Nov. 2021), S. 2881–2908. ISSN: 1615-1488. DOI: [10.1007/s00158-021-03001-2](https://doi.org/10.1007/s00158-021-03001-2) (zitiert auf S. 51, 70, 72).
- [VRM23] J. Voggesberger, P. Reimann, B. Mitschang. „Towards the Automatic Creation of Optimized Classifier Ensembles“. In: *Proceedings of the 25th International Conference on Enterprise Information Systems*. SCITEPRESS - Science und Technology Publications, 2023. DOI: [10.5220/0011988800003467](https://doi.org/10.5220/0011988800003467) (zitiert auf S. 26).
- [VYC+23] J. V. Viisainen, F. Yu, A. Codolini, S. Chen, L. T. Harper, M. P. F. Sutcliffe. „Rapidly predicting the effect of tool geometry on the wrinkling of biaxial NCFs during composites manufacturing using a deep learning surrogate model“. en. In: *Composites Part B: Engineering* 253 (März 2023), S. 110536. ISSN: 1359-8368. DOI: [10.1016/j.compositesb.2023.110536](https://doi.org/10.1016/j.compositesb.2023.110536). URL: <https://www.sciencedirect.com/science/article/pii/S1359836823000392> (zitiert auf S. 19, 20, 35, 39, 55, 72, 81).
- [WMR17] S. Wachter, B. Mittelstadt, C. Russell. *Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR*. en. Techn. Ber. 3063289. Rochester, NY, Okt. 2017. DOI: [10.2139/ssrn.3063289](https://doi.org/10.2139/ssrn.3063289). URL: <https://papers.ssrn.com/abstract=3063289> (zitiert auf S. 33).
- [WRG+23] Y. Wilhelm, P. Reimann, W. Gauchel, S. Klein, B. Mitschang. „Pusion - A Generic and Automated Framework for Decision Fusion“. In: *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. ISSN: 2375-026X. Apr. 2023, S. 3282–3295. DOI: [10.1109/ICDE55515.2023.00252](https://doi.org/10.1109/ICDE55515.2023.00252) (zitiert auf S. 26).
- [WWIT16] T. Wuest, D. Weimer, C. Irgens, K.-D. Thoben. „Machine learning in manufacturing: advantages, challenges, and applications“. In: *Production & Manufacturing Research* 4.1 (Jan. 2016), S. 23–45. DOI: [10.1080/21693277.2016.1192517](https://doi.org/10.1080/21693277.2016.1192517) (zitiert auf S. 15, 20, 44, 72).
- [XZZ21a] Y. Xu, M. Zhang, B. Zheng. „Design of cold-formed stainless steel circular hollow section columns using machine learning methods“. en. In: *Structures* 33 (Okt. 2021), S. 2755–2770. ISSN: 2352-0124. DOI: [10.1016/j.istruc.2021.06.030](https://doi.org/10.1016/j.istruc.2021.06.030). URL: <https://www.sciencedirect.com/science/article/pii/S2352012421005300> (zitiert auf S. 37, 72–74).

- [XZZ21b] Y. Xu, B. Zheng, M. Zhang. „Capacity prediction of cold-formed stainless steel tubular columns using machine learning methods“. en. In: *Journal of Constructional Steel Research* 182 (Juli 2021), S. 106682. issn: 0143-974X. doi: [10.1016/j.jcsr.2021.106682](https://doi.org/10.1016/j.jcsr.2021.106682). URL: <https://www.sciencedirect.com/science/article/pii/S0143974X21001644> (zitiert auf S. 20, 27, 37, 39, 58, 68, 70, 72, 87, 103).
- [ZXNL21] H. Zhou, Q. Xu, Z. Nie, N. Li. „A Study on Using Image-Based Machine Learning Methods to Develop Surrogate Models of Stamp Forming Simulations“. In: *Journal of Manufacturing Science and Engineering* 144.021012 (Aug. 2021). issn: 1087-1357. doi: [10.1115/1.4051604](https://doi.org/10.1115/1.4051604) (zitiert auf S. 20, 55).
- [ZZHM23] Q. Zhou, M. Zhao, J. Hu, M. Ma. *Multi-fidelity Surrogates*. Springer Nature Singapore, 2023. doi: [10.1007/978-981-19-7210-2](https://doi.org/10.1007/978-981-19-7210-2) (zitiert auf S. 15, 19).

Alle URLs wurden zuletzt am 20.09.2023 geprüft.

### **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift