

Bestimmung der Kerbformzahlen von Schweißverbindungen aus Oberflächenscans durch künstliche neuronale Netze

Masterarbeit

Eingereicht von:

Öner Aydogan
Informatik (Master)
Matrikelnummer: 4947195
aydogan@tf.uni-freiburg.de

Gutachter:

Prof. Dr. Joschka Boedecker
Prof. Dr. Christoph Eberl

Betreuer:

Matthias Jung
Dr. Jan Schubnell

2023

„Bestimmung der Kerbformzahlen von Schweißverbindungen aus Oberflächenscans durch künstliche neuronale Netze“

Version vom 21. August 2023

ERKLÄRUNG

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Symbolverzeichnis	vi
1 Einleitung	1
2 Grundlagen	3
2.1 Kerbformzahl	3
2.2 Machine Learning	5
2.2.1 Überwachtes Lernen und Regression	6
2.2.2 Convolutional Neural Networks	8
2.2.3 Generalisierung und Über-/Unteranpassung	9
3 Stand der Technik	11
3.1 Geometriebestimmung von realen Schweißnähten	11
3.2 Bestimmung der Kerbformzahl von realen Schweißnähten	14
3.2.1 Näherungsformeln	15
3.2.2 Methoden basierend auf künstlichen neuronalen Netzen	16
3.3 Machine Learning auf Punktwolken	16
3.3.1 PointNet	17
3.3.2 PointNet++	18
3.3.3 2DLaserNet	19
4 Forschungslücken & eigener Ansatz	22
4.1 Diskussion vorhandener Ansätze	22
4.2 Eigener Ansatz	24
5 Methodik	26
5.1 Reale Daten	26
5.2 Künstlich erzeugte Daten	28
5.2.1 Stumpfstoß	29

Inhaltsverzeichnis

5.2.2	Kreuzstoß	33
5.2.3	FE-Modelle	36
5.3	Datenaugmentierung	39
5.4	Machine Learning Modelle	40
5.4.1	Baseline Modelle	40
5.5	Training und Validierung	41
5.6	Implementierung	43
6	Ergebnisse und Diskussion	44
6.1	Baseline	44
6.2	R2-Loss	50
6.3	Vergleich verschiedener Punktabstände	53
6.4	Datenaugmentierung	55
6.5	Transfer künstliche Daten auf reale Daten	58
6.5.1	Transfer mit Datenaugmentierung	61
6.6	Vergleich mit anderen Methoden	62
7	Zusammenfassung und Ausblick	68
7.1	Ausblick	69
A	Anhang	71
A.1	Näherungsformeln	71
A.1.1	Anthes	71
A.1.2	Rainer	72
A.1.3	Kiyak	73
A.2	Generierung Punktwolke Stumpfstoß	73
A.3	Ergebnisse mit R2-Loss	75
A.4	Ergebnisse mit Augmentierungen	78
B	Nomenklatur	80
	Literaturverzeichnis	81

Symbolverzeichnis

K_t	Kerbformzahl
σ_{max}	Höchstwert der Spannung im Kerbgrund in Pa
σ_{nk}	Nennspannung in Pa
r	Nahtübergangsradius in mm
α	Nahtanstiegswinkel in Grad
w	Nahtbreite in mm
t	Bleckdicke in mm
h	Nahtüberhöhung in mm
a	Nahtdicke in mm

1 Einleitung

Schweißverbindungen spielen eine wichtige Rolle im Maschinenbau, in der Automobilindustrie und bei vielen anderen industriellen Anwendungen. Sie ermöglichen es, dass mehrere Komponenten zu einer stabilen Einheit verbunden werden können und so komplexe Strukturen entstehen. Daher ist die Qualität und Zuverlässigkeit von Schweißverbindungen von entscheidender Bedeutung, um die Sicherheit und Langlebigkeit von Bauteilen und Strukturen zu gewährleisten. In diesem Zusammenhang spielt die Kerbwirkung in Schweißnähten eine besonders wichtige Rolle, da sie zu Spannungskonzentrationen führen und somit vorzeitiges Versagen verursachen kann, falls diese nicht bei der Bauteilauslegung berücksichtigt werden [45]. Dem entsprechend ist die Bestimmung der Kerbformzahl, welche die Spannungskonzentration und die Kerbwirkung quantifiziert, von großer Bedeutung, um potenzielle Schwachstellen zu erkennen und das Versagensrisiko zu minimieren.

Zur Bestimmung der Kerbformzahl von Schweißnähten existieren verschiedene Ansätze. Eine häufig verwendete Methode besteht in der Anwendung von analytischen Approximationsformeln [12, 28, 41] oder Künstlichen Neuronalen Netzen (KNN) [35, 36]. Die Kerbformzahl wird hierbei anhand einer idealisierten Näherung der Schweißnahtgeometrie und basierend auf geometrischen Parameter wie Nahtübergangsradius und Nahtanstiegswinkel bestimmt. Diese Geometrieparameter dienen entweder als Eingangsparameter für die Approximationsformel oder als Eingabedaten für das KNN. Die Anwendung dieser Verfahren ist jedoch stark von der Bestimmung der Geometrieparameter abhängig. Diese variieren je nach Messsystem und Auswertealgorithmus [49] und sind mit hohen Streuungen verbunden [16]. Hinzu kommt dass die geometrischen Parameter, besonders der Nahtübergangsradius nicht immer eindeutig zuzuordnen ist und von der Scanauflösung abhängt [46]. Eine weitere Methode ist die Nutzung von 3D-Oberflächenscans, welche in Finite-Elemente-Simulationen überführt werden und mit Hilfe dieser die Kerbformzahl bestimmt wird [25]. Dies ermöglicht zwar eine realistischere Abbildung der Schweißnahtgeometrie, ist jedoch mit hohem Zeit-, Rechen- & Modellierungsaufwand verbunden.

1 Einleitung

Das Ziel dieser Arbeit besteht darin, neue Methoden zur direkten Bestimmung der Kerbformzahlen von Schweißverbindungen mit Hilfe von KNN zu entwickeln, welche auf Oberflächen-scans basieren. Diese Herangehensweise soll die direkte Nutzung von Oberflächenscans zur Kerbformzahlberechnung ermöglichen, sodass eine detailliertere Repräsentation der Schweißnahtgeometrie erreicht wird. Die Nutzung von trainierten KNN bietet außerdem den Vorteil, dass auf die Bestimmung von Geometrieparametern oder die Modellierung von FE-Modellen verzichtet wird, was sowohl Ressourcen spart als auch mögliche Fehlerquellen und Störfaktoren durch zwischengeschaltete Schritte minimiert. Durch das selbstständige Erlernen von Merkmalen werden die hochdimensionalen Eingangsdaten auf einen Ausgabewert, die Kerbformzahl, abgebildet. Es gibt verschiedene typische Rissausgangsorte bei Schweißverbindungen, im Rahmen dieser Arbeit wird jedoch nur der Nahtübergangsbereich behandelt, da dieser in der Regel den ermüdungskritischen Bereich einer Schweißverbindungen darstellt [24].

2 Grundlagen

In diesem Kapitel werden die Grundlagen und Konzepte erläutert, die das Verständnis und die Umsetzung dieser Arbeit ermöglichen sollen. Zunächst werden die Begriffe der Spannungskonzentration und Kerbformzahl und deren Bedeutung im Kontext von Schweißverbindungen behandelt.

Des Weiteren werden grundlegende Konzepte aus dem Bereich des Machine Learning (ML) und Deep Learning (DL) diskutiert. Die Grundprinzipien des überwachten Lernens werden erläutert und eine Einführung in Convolutional Neural Networks (CNNs) gegeben, die einen integralen Bestandteil der verwendeten DL-Architekturen darstellen. In diesem Zusammenhang werden auch die Themen der Generalisierung, Über- und Unteranpassung behandelt.

2.1 Kerbformzahl

In der Mechanik bezeichnet eine Spannungskonzentration eine Stelle in einem Bauteil, an der die Spannung deutlich größer ist als in den umgebenden Bereichen. Spannungskonzentrationen können unter anderem durch Kerben oder Unregelmäßigkeiten in der Geometrie oder im Material eines Bauteils entstehen, die eine Unterbrechung des Kraftflusses verursachen. Als Kerbe wird eine plötzliche Querschnittsänderung in Bauteilen bezeichnet [45]. Diese können bewusst in das Material eingefügt werden oder durch die Fertigung oder den Betrieb eines Bauteils entstehen. Erhöhte Spannungskonzentrationen können die Festigkeit und Belastbarkeit eines Bauteils stark beeinträchtigen und dazu führen, dass es unter bestimmten Belastungen leichter bricht oder versagt [45]. In Abbildung 2.1 a) sind die Kraftflüsse in einem gekerbten Bauteil entlang des Querschnitts skizziert. Die Kraftlinien verlaufen nicht homogen, sondern sie zeigen eine erhöhte Dichte im Bereich der Kerbe auf,

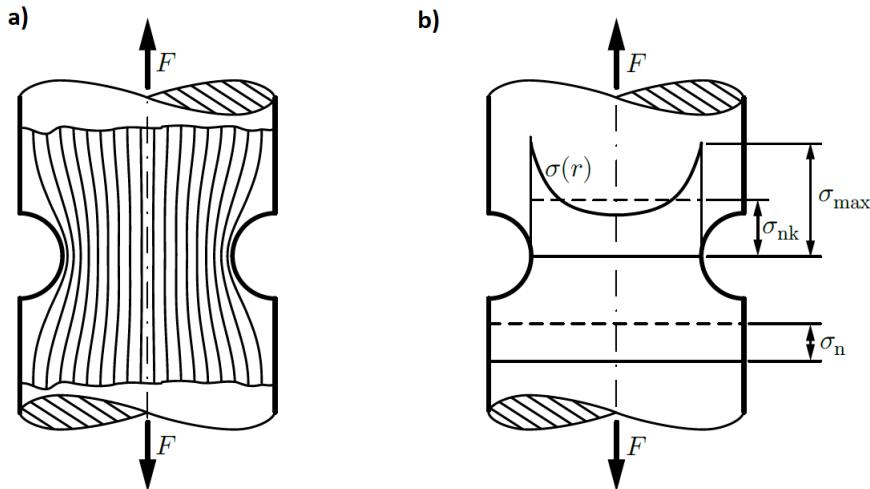


Abbildung 2.1: a) Kraftfluss in einem gekerbten Bauteilen bei Zugbelastung. Die Dichte der Kraftflusslinien ist ein Maß für die Höhe der Spannung. Im Kerbgrund ergibt sich eine Spannungskonzentration [45]
 b) Verteilung der Längsspannung bei Zugbelastung [45]

was zu einer Konzentration von Spannungen an dieser Stelle führt. Diese lokale Spannungskonzentration im Kerbgrund ist in Abbildung 2.1 b) zu sehen und ist als Maximalspannung σ_{max} bezeichnet. Die Intensität der Spannungskonzentration hängt hierbei von den spezifischen geometrischen Eigenschaften des Kerbgrundes ab und wird durch die *Kerbformzahl* beschrieben, welche für linear-elastisches Werkstoffverhalten folgendermaßen definiert ist

$$K_t = \frac{\sigma_{max}}{\sigma_{nk}} \quad (2.1)$$

wobei σ_{max} die maximale Spannung im Kerbgrund und σ_{nk} die nominelle Spannung im Kerbquerschnitt sind [45]. Die nominelle Spannung stellt die Spannung dar, welche ohne Kerbe in einem Querschnitt vorhanden wäre, der dem Kerbquerschnitt entspricht.

Die Kerbformzahl K_t ist essenziell für die Gestaltung und Analyse von Bauteilen, die Kerben aufweisen. Schweißverbindungen weisen durch ihre geometrische Form und aufgrund des Schweißvorgangs Spannungskonzentrationen und eine starke Kerbwirkung auf [6]. Das Ermüdungsverhalten bei Schweißnähten ist stark von der spezifischen Geometrie und der Art der Bauteilkasse abhängig [43]. Die zwei maßgeblichen Parameter, welche das Verhalten

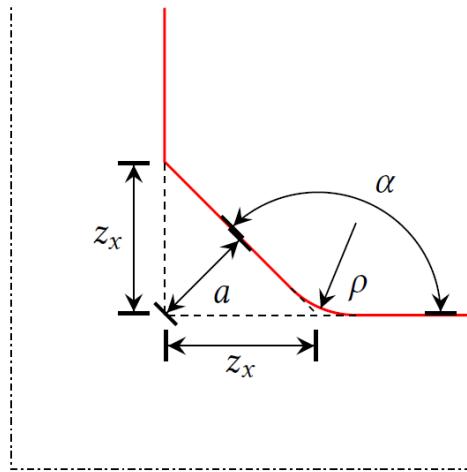


Abbildung 2.2: Geometrieparameter einer Kehlnaht

auf Belastung beeinflussen, sind der Nahtanstiegswinkel α und der Nahtübergangsradius ρ (Abbildung 2.2) [47].

Die Kerbformzahl ist eine wichtige Größe um den Einfluss von Kerben auf die Spannungsverteilung zu quantifizieren und das Verhalten der Schweißnaht unter Belastung besser zu verstehen. Zur Bestimmung der Kerbformzahl von Schweißnähten gibt es verschiedene Ansätze. Die Methoden von Anthes et al. [12], Kiyak et al. [28] und Rainer [41] basieren auf idealisierten Modellen der Schweißnahtgeometrie und geben Approximationsformeln an, welche anhand der Geometrieparameter die Kerbformzahl berechnen. Diese werden in 3.2.1 näher beschrieben.

2.2 Machine Learning

In diesem Abschnitt werden bestimmte Bereiche und Begriffe des Machine Learnings und künstlicher neuronaler Netze erklärt, welche für das Verständnis dieser Arbeit sinnvoll sind.

2.2.1 Überwachtes Lernen und Regression

Auf dem Gebiet des ML gibt es verschiedene Methoden und Ansätze, welche darauf abzielen, aus Daten zu lernen und Muster sowie Beziehungen zu erkennen. Die Art des Lernens kann hierbei grob in drei Hauptkategorien unterteilt werden, die unterschiedliche Lernkonzepte darstellen: überwachtes Lernen, unüberwachtes Lernen und bestärkendes Lernen. Aufgrund der Relevanz für diese Arbeit beschränkt sich dieser Abschnitt auf den Bereich des überwachten Lernens und auf den Fall der Regression.

Beim überwachten Lernen werden einem Algorithmus Trainingsdaten zur Verfügung gestellt, die aus Eingabedaten und den entsprechenden Ausgabedaten bestehen. Der Algorithmus versucht anhand dieser Daten eine Funktion zu erlernen, welche die Eingabedaten auf die richtige Ausgabe abbildet. Wenn die gewünschte Ausgabe aus einer oder mehreren stetigen Variablen besteht, dann wird die Aufgabe als Regression bezeichnet [14]. Der Trainingsablauf eines Algorithmus zur Regression kann anhand eines künstlichen neuronalen Netzes erläutert werden (Abbildung 2.3). Ein KNN besteht hierbei aus mehreren Schichten von künstlichen Neuronen, die miteinander verbunden sind. Jedes Neuron bekommt Eingangsdaten, bestehend aus ein oder mehreren *Features* (in der Abbildung für die Eingangsschicht mit x_1 und x_2 bezeichnet) und verrechnet diese mit Hilfe der eigenen Gewichte (*Weights*) und einem *Bias*-Term (in der Abbildung $Bias = 1$). Dies kann mathematisch folgendermaßen ausgedrückt werden [14]:

$$z_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (2.2)$$

wobei j das j -te Neuron in der Schicht, D die Anzahl der *Features*, w_{ji} die *Gewichte* und w_{j0} die *Biases* darstellt. Die Schreibweise ⁽¹⁾ gibt an, dass sich die entsprechenden Parameter in der ersten Schicht des Netzes befinden. Auf das Ergebnis z_j dieser Berechnung wird eine sogenannte Aktivierungsfunktion f angewandt und an die nächste Schicht weitergeleitet. Dies wird folgendermaßen beschrieben

$$o_j = f(z_j) \quad (2.3)$$

2 Grundlagen

Bei der Funktion f handelt es sich oft um nichtlineare Funktionen wie zum Beispiel die Sigmoid- oder $tanh$ -Funktion [14]. Dieser Vorgang kann bis zur letzten Ausgabeschicht fortgeführt werden. Die Ausgabe, also die Vorhersage des Modells, wird schließlich verglichen mit dem Zielwert bzw. Wahrheitswert. Dies geschieht anhand der sogenannten Loss-Funktion (auch Verlustfunktion genannt), welche eine mathematische Funktion darstellt, die den Fehler oder die Differenz zwischen den vorhergesagten Werten eines Modells und den tatsächlichen Werten quantifiziert [14]. Das Ziel des Lernprozesses ist es den *Loss* zu minimieren, also die Differenz zwischen der Vorhersage des Modells und den tatsächlichen Werten. Nach der Berechnung des *Losses* wird dieser durch das Modell zurück propagiert und dabei wird der Gradient der Loss-Funktion bezüglich der Gewichte und Bias-Werte jedes Neurons berechnet [14]. Anschließend wird dieser Gradient verwendet um die Gewichte und Bias-Werte anzupassen. Dieser Prozess der Vorwärts- und Rückwärtspropagierung wird während des Trainings für eine bestimmte Anzahl an *Epochen* wiederholt, sodass der Fehler bestenfalls reduziert und optimiert wird und das Modell lernt Vorhersagen anhand der Eingangsdaten zu treffen.

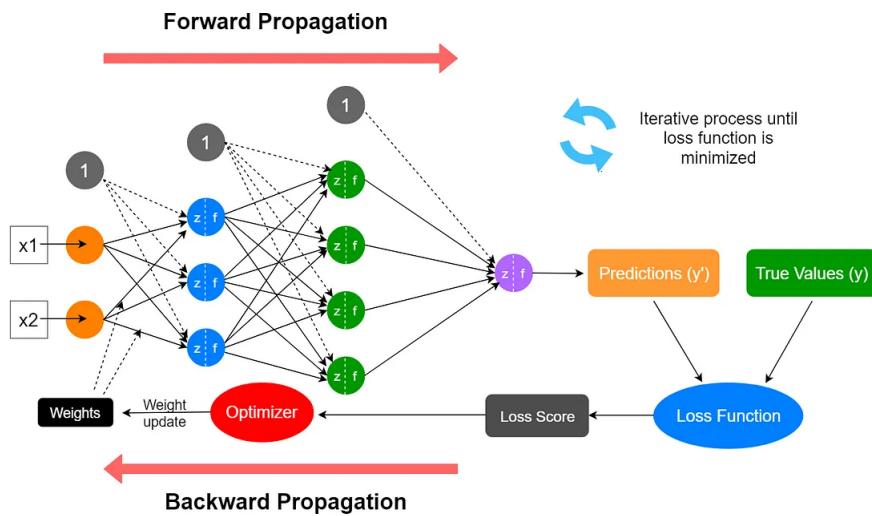


Abbildung 2.3: Beispielhaftes Training und Aufbau eines KNN mit Vorwärts- und Rückwärts-propagierung [8]

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) sind eine spezielle Art von künstlichen neuronalen Netzwerken die vor allem für die Verarbeitung von Daten, die eine gitterartige Struktur aufweisen, angewandt werden [21]. Beispielsweise können Daten von Zeitreihen als 1D-Gitter betrachtet werden, bestehend aus Stichproben in regelmäßigen Zeitabständen, und Bilddaten als 2D-Gitter von Pixeln.

Essenziell für CNNs ist die Verwendung einer speziellen linearen mathematischen Operation namens *Faltung* oder *Convolution*. Hierbei besteht die Faltung aus *Eingangsdaten* und einem *Kernel* (Filter). Bei Anwendungen im ML sind die Eingangsdaten in der Regel ein mehrdimensionales Feld von Daten und der Kernel ist in der Regel ein mehrdimensionales Feld von Parametern, die durch den Lernalgorithmus angepasst werden [21]. Die Faltungsoperation kann anhand eines Beispiel mit einem 2D-Input und einem Kernel der Größe 3×3 verdeutlicht werden (Abbildung 2.4). In diesem Fall wird ein Ausschnitt der Eingangsdaten betrachtet (lokales rezeptives Feld genannt) und mit dem Kernel gefaltet. Dabei wird jedes Element des rezeptiven Feldes mit den korrespondierenden Gewichten des Filters multipliziert und anschließend aufsummiert. Es entsteht im Endeffekt eine neue 2D-Matrix, weil der Kernel über den Input verschoben wird, sodass das rezeptive Feld stets einen anderen Ausschnitt betrachtet.

Der Grund für die Nutzung der Faltung ergibt sich aus der Tatsache, dass ein vollständig verbundenes Netzwerk schnell mit der Größe eines Bildes wächst und zu Überanpassung führen kann (siehe Abschnitt 2.2.3). Außerdem führt dies zu größerem Speicherverbrauch und zu einer erhöhten Rechenzeit [4]. Faltungen bieten außerdem den Vorteil, dass sie translationsinvariant sind, d.h. das Ergebnis der Faltung ist bei verschobenem Input das selbe [14]. CNNs verwenden außerdem sogenannte *Pooling-Schichten*. Diese ersetzen die Ausgabe des Netzes an einer bestimmten Stelle durch eine zusammenfassende Statistik der nahe gelegenen Werte [21]. Hierdurch wird die räumliche Dimension der Daten reduziert. Beispielsweise gibt die *max-Pooling* Funktion den maximalen Output innerhalb einer rechteckigen Nachbarschaft an (zu sehen in Abbildung 2.4). Das Einsetzen von mehreren Faltungs- und Pooling-Schichten führt dazu, dass die Größe des rezeptiven Felds in den tieferen Schichten steigt, eine abstraktere Darstellung der Daten erzeugt und komplexe *Features* erkannt werden. Sowohl die Größe der rezeptiven Felder als auch die Komplexität der

erfassten Merkmale ermöglichen es CNNs, lokale Muster in den Eingangsdaten zu erkennen und Merkmale effizient zu extrahieren [55].

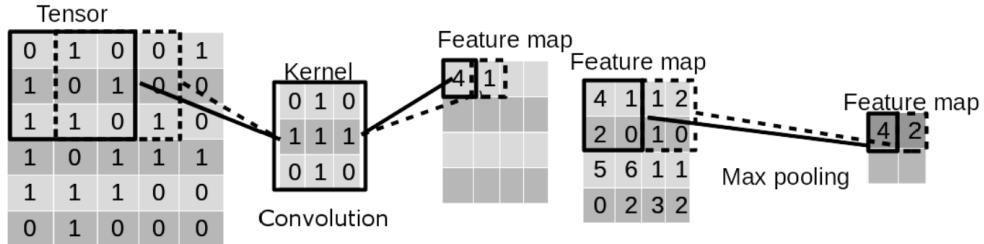


Abbildung 2.4: Faltungsoperation und Max pooling am Beispiel eines 2D-Inputs und einem Kernel der Größe 3×3 [50]

2.2.3 Generalisierung und Über-/Unteranpassung

Eines der Hauptziele eines ML-Modells ist es gute Vorhersagen auf neuen, ungesiehenen Daten zu treffen, also die Fähigkeit der *Generalisierung*. Um diese Fähigkeit zu quantifizieren, werden ML-Modelle auf einem separaten Datensatz evaluiert, welcher Testdatensatz genannt wird. Die Leistungsfähigkeit eines Modells hängt davon ab wie gut es den Trainingsfehler minimiert und die Differenz zwischen Trainings- und Testfehler klein hält. Diese beiden Aspekte stellen die beiden zentralen Herausforderungen der *Unter- und Überanpassung* im Machine Learning dar [21]. Falls das Modell den Trainingsfehler nicht effektiv minimieren kann, spricht man von Unteranpassung. Dies kann auftreten, falls das Modell nicht ausreichend Kapazität verfügt, d.h. die nicht in der Lage ist eine Vielzahl von Datenmustern zu erfassen [21]. In diesem Fall ist die Leistung des Modells begrenzt und Muster in den Daten werden nicht gut genug gelernt. Die Folge ist eine unzureichende Anpassung an die Trainingsdaten und eine schlechte Generalisierung auf neuen, ungesiehenen Daten. Das Gegenteil dieses Phänomens, die Überanpassung, bezeichnet die Spezialisierung eines Modells auf den Trainingsdatensatz. Dies bedeutet, dass das Modell sich sehr genau auf die Trainingsdaten anpasst und die Fähigkeit verliert auf neue Daten zu verallgemeinern. Ersichtlich wird dies durch einen niedrigen Trainings- und einem hohen Testfehler. Insbesondere bei Modellen mit zu hoher Kapazität kann es zur Überanpassung kommen aufgrund der Tatsache, dass die Eigenschaften der Daten auswendig gelernt werden, statt Muster zu erkennen und zu erlernen [21].

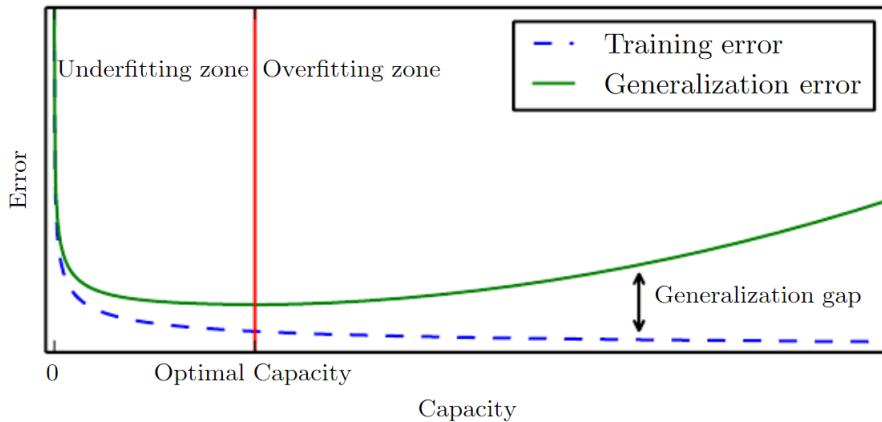


Abbildung 2.5: Beispielhafte Darstellung des typischen Verhältnis zwischen Modellkapazität und Fehler [21]

Eine Möglichkeit die Kapazität und damit potenziell die Generalisierungsfähigkeit zu beeinflussen, ist das Anpassen der Modellarchitektur. Beispielsweise das Hinzufügen oder Entfernen von Schichten in einem KNN. Das Verhältnis zwischen Kapazität und Fehler kann anhand von Abbildung 2.5 verdeutlicht werden. Bei geringer Kapazität ist sowohl der Trainings- als auch der Generalisierungsfehler groß, d.h. das Modell ist unterangepasst. Mit größer werdender Kapazität können beide Fehler gesenkt werden. Dies kann bis zur *optimalen Kapazität* fortgeführt werden. Das Überschreiten dieser Kapazität führt jedoch zur Überanpassung. Der Generalisierungsfehler wird größer und die Differenz zum Trainingsfehler steigt.

3 Stand der Technik

Wie in Abschnitt 2.1 erwähnt, quantifiziert die Kerbformzahl lokale Spannungskonzentrationen von gekerbten Bauteilen, worunter auch Schweißverbindungen fallen. Die Kerbformzahl ist von großer Bedeutung um das Verhalten der Schweißnaht unter Belastung besser zu verstehen, sodass zum Beispiel die Gestaltung des entsprechenden Bauteils angepasst werden kann. Die zwei maßgeblichen Geometrieparameter einer Schweißnaht, welche das Verhalten auf Belastung beeinflussen, sind der Nahtanstiegswinkel α und der Nahtübergangsradius r (Abbildung 2.2). In diesem Kapitel werden Vorgehensweisen zur Bestimmung der Geometrieparameter an realen Schweißnähten aufgezeigt und deren Probleme und Limitierungen diskutiert. Anschließend werden unterschiedliche Methoden zur Bestimmung der Kerbformzahl von realen Schweißnähten vorgestellt. Ziel dieser Arbeit ist es Kerbformzahlen von Schweißverbindungen mit Hilfe künstlicher neuronaler Netze zu bestimmen. Hierfür werden Oberflächenscans von Schweißverbindungen, welche in Form von Punktwolken vorliegen, verwendet. In diesem Zusammenhang werden Methoden im Bereich des Machine Learnings auf Punktwolken präsentiert, welche relevant für diese Arbeit sind.

3.1 Geometriebestimmung von realen Schweißnähten

Die Bestimmung geometrischer Parameter an realen Schweißnähten kann aufgrund der komplexen Geometrie, welche je nach Schweiß- und Behandlungsprozess stark variieren kann, eine herausfordernde Aufgabe darstellen. Zur Bestimmung der Geometrie von realen Schweißnähten existieren verschiedene Methoden. Eine Möglichkeit besteht darin, die Geometrieparameter wie zum Beispiel den Nahtübergangsradius r und Nahtanstiegswinkel α mit Hilfe von Radienlehren direkt an der Schweißnaht auszumessen. Während die Messung in der Produktion einfach möglich ist, hängt sie jedoch stark vom Anwender ab [22]. Andere Methoden verwenden häufig Schnitte quer zur Schweißrichtung und werten

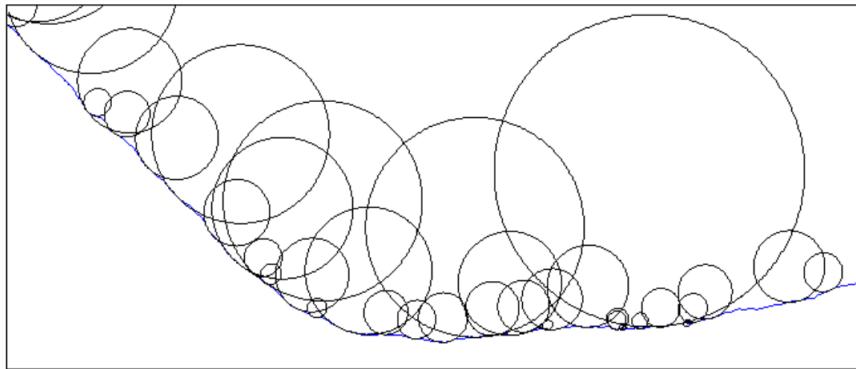


Abbildung 3.1: Mehrdeutigkeit von Nahtübergangsradien einer Schweißverbindung [34]

diese aus. Beispielsweise kann ein Silikonabdruck der Schweißnaht erstellt und daran ein Schnitt gemacht und ausgewertet werden [32]. Alternativ ist über das Tastschnittverfahren die Geometriebestimmung einer Schnittposition direkt an der Schweißnaht möglich [33, 46]. Um zuverlässige und wiederholbare Ergebnisse zu erhalten, sind manuelle Auswertungen der Schweißnahtgeometrie aufgrund des hohen Aufwands nicht immer gut geeignet. Aus diesem Grund werden computergestützte Ansätze bevorzugt [42]. Hierbei wird die Oberfläche der Schweißnaht oder eines Abdrucks mit Hilfe von optischen 3D-Messmethoden in Form einer Punktwolke oder eines Polygonnetzes erfasst. Daraus kann dann wiederum ein digitaler 2D-Schnitt an einer bestimmten Stelle erzeugt werden um weitere detailliertere Auswertungen durchzuführen. Die Bestimmung der Parameter α und r kann dann beispielsweise manuell durch das Anpassen von Kreisen mit Hilfe von Bildbearbeitungs-[32] oder CAD Programmen [20, 31, 52] erfolgen. Auch bei diesen Verfahren ist jedoch die Bestimmung der Geometrie vom Anwender abhängig [54]. Trotz der Tatsache, dass viele Methoden vorhanden sind gibt es weder eine standardisierte Definition noch ein allgemein anerkanntes Verfahren zur Bestimmung des Nahtübergangsradius [46]. Ein Problem das damit einhergeht ist, dass der Radius oft nicht eindeutig zuzuordnen ist bei realen Schweißnähten. Dies lässt sich am Beispiel in Abbildung 3.1 verdeutlichen. Hinzu kommt, dass das System, mit dem die Schweißnaht gescannt wurde, ebenfalls einen großen Einfluss auf die Bestimmung des Radius hat. Je nach Auflösung und Abstand der Punkte kann der ermittelte Radius und Winkel stark variieren (siehe Abbildung 3.2).

Andere Verfahren basieren auf der automatischen Bestimmung der Parameter mit Hilfe von Algorithmen. In der Arbeit von Jung wurden zwei verschiedene automatisierte Methoden

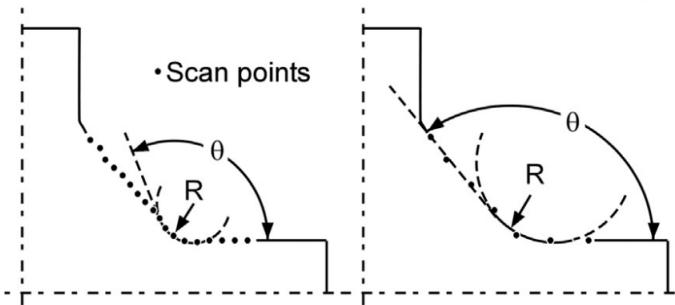


Abbildung 3.2: Einfluss des Punktabstandes auf die Bestimmung der Geometrieparameter [57]

hervorgebracht [26]. Zum einen die *Curvature-Methode* und zum anderen die *Optimierungsmethode*. Die Curvature-Methode wurde im Rahmen dieser Arbeit verwendet, indem Geometrieparameter von realen Kehlnähten extrahiert und diese zum Vergleich mit Approximationsformeln herangezogen wurden. Zur Bestimmung der Nahtgeometrie bei Curvature-Methode berechnet diese zunächst die Krümmung eines ebenen Schweißnahtprofils und lokaliert den Nahtübergang an der Stelle der stärksten Krümmung (siehe Abbildung 3.3). Der Nahtübergangsradius wird schließlich über den Kehrwert der Krümmung bestimmt. Die Optimierungsmethode hingegen passt ein parametrisches Modell an ein ebenes Schweißnahtprofil an. Evaluiert wurden die beiden Methoden indem die Kerbformzahlen, welche mit Hilfe der bestimmten Geometrieparameter berechnet wurden, mit den ermittelten Kerbformzahlen aus geometrienahen FEM-Modellen verglichen wurde. Die Curvature-Methode zeigte eine gute Übereinstimmung mit den geometrienahen FEM-Modellen auf, während die Optimierungsmethode eine schlechtere Übereinstimmung aufwies, die auf zu groß ermittelten Radien zurückzuführen ist. Bei einem Vergleich mit Proben und Scandaten anderer Forschungseinrichtungen zeigte sich außerdem, dass trotz vergleichbarer Auflösungen der Scans die unterschiedlichen Messmethoden einen Einfluss auf den ermittelten Radius hatte, was auf eine Schwachstelle der Curvature-Methode hinweist. Außerdem zeigte sich, dass die Auflösung des Messsystems einen signifikanten Einfluss auf die Ermittlung des Radius hat. Es konnten sogar Abweichungen um das Doppelte des Radius beobachtet werden bei variierendem Messsystem [49].

Ein weiterer Ansatz zur automatischen Bestimmung der Geometrieparameter wurde von Lindgren und Stenberg entwickelt [34]. Ähnlich wie bei Jung berechnet ihr Algorithmus die Krümmung entlang der Schweißnaht und lokaliert hierdurch den Nahtübergang. Anschließend wird ein Kreis anhand der Least-Square Methode angepasst, welcher den

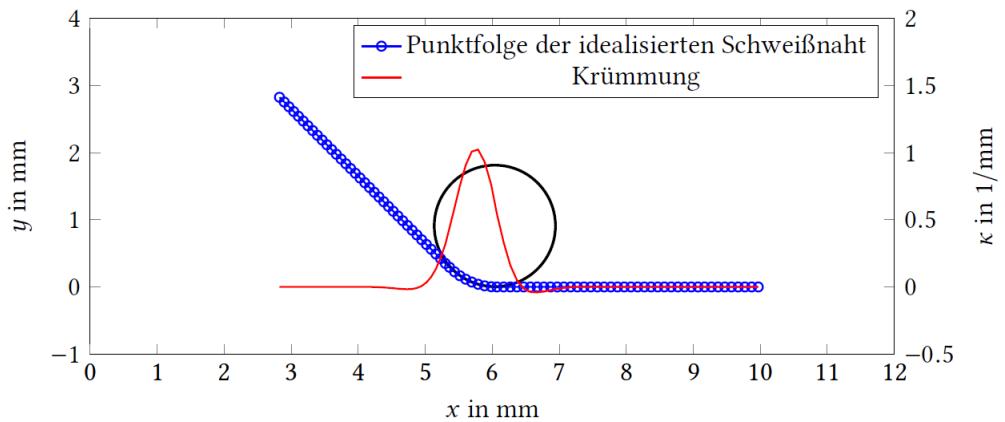


Abbildung 3.3: Krümmungsverlauf am Beispiel einer idealisierten Schweißnaht [26]

Nahtübergangsradius beschreibt. Der Nahtanstiegswinkel kann daraufhin anhand der Bogenlänge und des ermittelten Radius berechnet werden.

Kritisch für die Anwendbarkeit der Methoden, welche auf Scans von realen Schweißnähten aufbauen, ist die Auflösung des Messsystems. Jung et. al zeigten auf, dass die Ermittelbarkeit des Radius der untersuchten Methoden durch die Auflösung des Messsystems limitiert ist [49]. Es wurde festgestellt, dass der Radius hierbei mit mindestens 3 Punkten abgebildet werden muss. Der Nahtanstiegswinkel hingegen konnte von allen untersuchten Methoden relativ genau ermittelt werden mit einer Streuung von 8 % oder weniger zwischen den verschiedenen untersuchten Methoden. Dies lässt sich auf die Definition als Winkel zwischen zwei Geraden zurückführen, welche bei jeder Methode mit einer vergleichbar hohen Anzahl von Punkten vernetzt wurden.

3.2 Bestimmung der Kerbformzahl von realen Schweißnähten

Gemäß den Ausführungen in 2.1 ist die Ermittlung der Kerbformzahl von großer Bedeutung um das Verhalten einer Schweißnaht unter Belastung besser zu verstehen. Zur Bestimmung dieser sind verschiedene Methoden vorhanden, welche unterschiedliche Ansätze verfolgen. Hierbei hängt die Kerbformzahl von der Beanspruchungsart und vom Typ der Schweiß-

verbindung ab. In diesem Kapitel werden die Schweißnahtvarianten des Kreuzstoßes und Stumpfstoßes auf Zug- und Biegebeanspruchung behandelt.

3.2.1 Näherungsformeln

Eine Möglichkeit die Kerbformzahlen von Schweißverbindungen zu ermitteln ist die Nutzung von Finite-Elemente (FE)-Simulationen. Hierbei werden entweder 2D-Schnitte oder dreidimensionale Daten der Schweißnaht mit einer Software modelliert und anschließend die simulierten Spannungskonzentrationen ausgewertet. Beispiele hierfür finden sich in den Arbeiten von Jung [26], Hou [25], Chanh [31] und Schubnell et al. [49]. Die FE-Modellierung und die Simulation ist jedoch mit hohem Rechen- und Modellierungsaufwand verbunden. Aufgrund dessen werden häufig Näherungsformeln verwendet, welche eine einfache und schnelle Möglichkeit bieten um die Kerbformzahl für bestimmte Schweißnahttypen und Belastungsfälle zu berechnen.

Anthes et. al [12] haben basierend auf einer idealisierten Nahtgeometrie, bestehend aus wenigen Parametern, verschiedene Nahtgeometrievarianten erstellt und mit Hilfe der Randelemente-Methode die Kerbformzahl für Normalkraft- und Biegebeanspruchung berechnet. Hierbei wurden Stumpfstoße und Kreuzstoße betrachtet. Auf dieser Basis wurde eine nichtlineare multiple Regressionsanalyse durchgeführt und eine Näherungsformel erstellt, welche die Kerbformzahl in Abhängigkeit der definierten Geometrieparameter berechnet. Die Formeln für die Biegebeanspruchung und deren Anwendungsbereiche sind in A.1.1 zu finden. Ähnlich dazu hat Rainer mit Hilfe von FE-Simulationen basierend auf Elemente höherer Ordnung und feiner Vernetzung Kerbformzahlen für Stumpf- und Kreuzstoße mit hoher Genauigkeit berechnet [41]. Die Formeln und Anwendungsbereiche für den Biegefall sind in A.1.2 vermerkt. Kiyak et al. haben ebenfalls mit Hilfe von 2D-FE-Analysen Näherungsformeln bestimmt, welche den Anwendungsbereich bereits vorhandener Ansätze erweitert, insbesondere was den Nahtübergangsradius und Winkel angeht [28]. Abgesehen von der Erweiterung des Anwendungsbereichs zeigte ihre Methode außerdem eine verbesserte Genauigkeit im Vergleich zu vorhandenen Methoden auf. Die Formel und der Anwendungsbereich dieser Methode ist in A.1.3 aufgelistet.

3.2.2 Methoden basierend auf künstlichen neuronalen Netzen

Alternative Ansätze nutzen KNN, die mittels Geometrieparameter trainiert wurden und die Kerbformzahl als Ausgabe liefern. Oswald et al. haben einen Ansatz basierend auf KNN sowohl für Kreuzstöße, als auch für Stumpfstöße für Spannungs- und Biegebeanspruchung hervorgebracht [35, 36]. Ihre Modelle wurden auf einem großen Datensatz bestehend aus Geometrieparametern und Kerbformzahlen, welche aus FE-Simulationen stammen, trainiert. Im Vergleich zu anderen Methoden konnte ihr Ansatz geringere Fehler in der Bestimmung der Kerbformzahl aufweisen und gleichzeitig den zulässigen Anwendungsbereich erweitern. Dabiri et al. untersuchten die Kerbformzahl in Schweißverbindungen von Stumpfstößen unter axialer Zug- und Biegebeanspruchung mit Hilfe von KNN [17]. Sie unterschieden zwischen V-Naht und DV-Naht an Stumpfstößen, da die symmetrischen und unsymmetrischen Formen dieser Schweißnahtart die Kerbformzahl beeinflussen, was die Verwendung einer ähnlichen Gleichung für beide Formen ungeeignet macht. Außerdem berücksichtigt ihre Methode axiale Fehlausrichtungen und liefert so genauere Ergebnisse als herkömmliche Methoden.

3.3 Machine Learning auf Punktfolgen

Im Bereich des Machine Learnings ist die Verarbeitung visueller Informationen anhand von Bildern sehr verbreitet. Bei der direkten Erfassung von Daten in 3D ist es jedoch weniger üblich, eine dreidimensionale Pixelanordnung zu haben, die ein ganzes Volumen repräsentiert. Die Darstellung von räumlichen Daten geschieht stattdessen oft in Form einer Punktfolge. Punktfolgen stellen eine Menge von Datenpunkten im dreidimensionalen Raum dar. Der Raum muss sich jedoch nicht auf drei Dimensionen beschränken und kann auch aus mehr oder weniger Dimensionen bestehen. Ein gängiger Ansatz zur Erfassung von Daten als Punktfolge ist die Verwendung von Scannern, welche Objektoberflächen als Koordinaten aufzeichnen. Ein Beispiel hierfür sind LiDAR-Sensoren.

Während die Verarbeitung von Bildern aufgrund der Gitterstruktur für ML-Methoden weit verbreitet ist, ist dies bei Punktfolgen weniger der Fall. Punktfolgen können beliebig viele Punkte ohne feste Struktur oder Gitteranordnung haben, weswegen speziell dafür konstruierte ML-Methoden verwendet werden müssen.

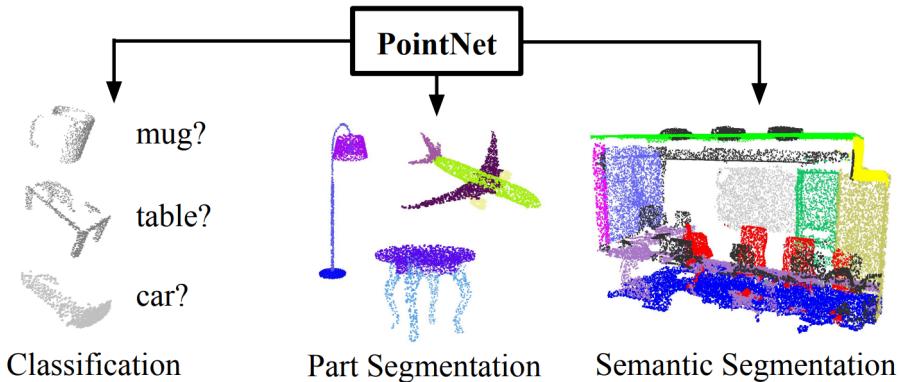


Abbildung 3.4: Die drei Anwendungsarten von PointNet: Klassifizierung, Teilsegmentierung, semantische Segmentierung [40]

3.3.1 PointNet

Eine der Pionierarbeiten in diesem Bereich wurde von Qi et al. entwickelt und nennt sich *PointNet* [40]. Im Vergleich zu anderen Arbeiten, welche räumliche Daten in 3D-Voxel-Gitter oder andere Datenformate umwandeln um sie für CNNs nutzbar zu machen, ist PointNet in der Lage räumliche Daten direkt als Punktwolken zu verarbeiten. Hierdurch wird der Umwandlungsprozess der 3D-Daten umgangen, welcher ineffizient (mehr Speicherverbrauch) oder zu Quantisierungsartefakten führen kann, d.h. Informationen gehen verloren oder werden verzerrt. PointNet besitzt drei verschiedene Anwendungsarten: Klassifizierung, Teilsegmentierung, semantische Segmentierung (siehe Abbildung 3.4). Diese stellen typische Anwendungsarten im Bereich von visuellen Daten dar. Bei der Klassifizierung gibt PointNet eine Klasse für den gesamten Input aus, wohingegen bei der semantischen Segmentierung jedem Pixel eine Klasse zugeordnet wird. Im Fall der Teilsegmentierung wird die Punktwolke in einzelne Teile oder Segmente gelabelt, welche bestimmte Teile oder Regionen eines 3D-Objekts kennzeichnen sollen.

Die PointNet Architektur zur Klassifizierung und Segmentierung ist in Abbildung 3.5 dargestellt. Der Klassifizierungsteil nimmt n Punkte mit drei Dimensionen als Eingabe und wendet Eingabe- und Featuretransformationen an und fasst daraufhin die Features der Punkte durch max-pooling zusammen. Die Ausgabe ist schließlich ein Vektor mit k -Klassen, wobei jeder Klasse ein Score bzw. eine Wahrscheinlichkeit zugewiesen wird. Das Netzwerk zur Segmentierung stellt eine Erweiterung des Klassifizierungsnetzwerks dar und

3 Stand der Technik

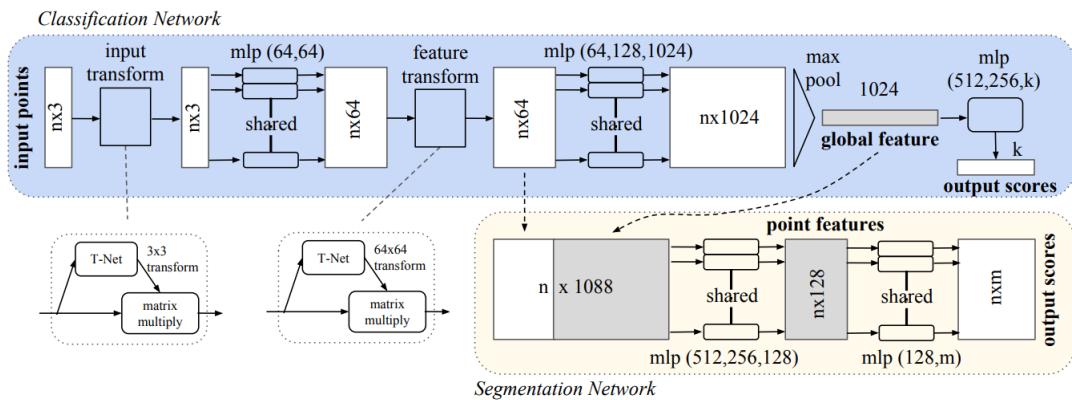


Abbildung 3.5: PointNet Architektur [40]

verbindet globale und lokale Features. Die max-pooling Schicht spielt bei PointNet eine Schlüsselrolle, da sie sich aufgrund ihrer symmetrischen Eigenschaft (Ausgabe ist invariant zur Anordnung der Eingabe) gut zur Verarbeitung von Punktwolken eignet, bei denen die Punkte in ungeordneter Form vorliegen. Zur Evaluierung der Methode wurde PointNet auf den Benchmark Datensätzen ModelNet40, ShapeNet, Stanford 3D zur Klassifizierung, Teilsegmentierung und semantischen Segmentierung angewandt. Im Vergleich zu anderen Methoden, welche auf dreidimensionalen Eingangsdaten basieren, konnten sie in allen drei Anwendungsbereichen gleich gute oder besser Ergebnisse erzielen.

3.3.2 PointNet++

Aufgrund der Konzipierung von PointNet, ist es nicht in der Lage feinere lokale Strukturen auf verschiedenen Größenordnungen zu erfassen. Um diese Schwachstelle zu verbessern veröffentlichten Qi et al. im Jahre 2017 ein Nachfolgemodell namens *PointNet++*, welches auf PointNet aufbaut. Die Funktionsweise von PointNet++ lässt sich folgendermaßen erklären: zunächst wird die Punktwolke in kleinere, sich überschneidende Regionen eingeteilt, welche Nachbarschaften darstellen. Anschließend werden lokale Features aus jeder Region ermittelt, die feinere geometrische Strukturen zusammenfassen. Um die lokalen Features zu generieren wird hierbei PointNet verwendet, da es in der Lage ist ungeordnete Punktmengen in semantische Features zu extrahieren. Die lokalen Features werden schließlich zu größeren zusammengefasst und verarbeitet um höher dimensionale Features zu generieren.

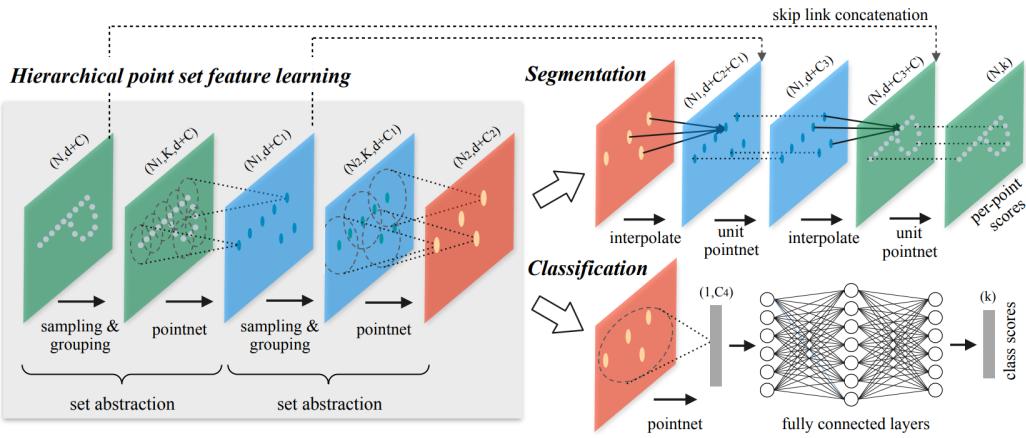


Abbildung 3.6: PointNet++ Architektur [39]

Diese größeren Einheiten erfassen komplexere Muster, die über einzelne Punkte hinausgehen. Der Prozess wird wiederholt bis Features der ganzen Punktwolke erzeugt wurden. Die endgültigen Features stellen hochdimensionale Informationen dar, welche Auskunft über die Form, Struktur oder das Objekt geben, die in der gesamten 3D-Punktwolke enthalten sind. Die Architektur des PointNet+ Modells ist in Abbildung 3.6 zu sehen. Ihr Ansatz wurde auf vier unterschiedliche Datensätze evaluiert. Diese Datensätze enthielten 2D-Objekte (MNIST), 3D-Objekte (ModelNet40 für starre Objekte und SHREC15 für nicht-starre Objekte) sowie echte 3D-Szenen (ScanNet). In jedem der genannten Anwendungsfälle erreichte ihr Ansatz herausragende Ergebnisse und übertraf dabei andere moderne Methoden zur Verarbeitung von visuellen Daten.

3.3.3 2DLaserNet

Eine weitere Arbeit im Bereich des Machine Learning auf Punktwolken wurde von Kaleci et al. hervorgebracht [27]. Ihr Ziel war die erfolgreiche Klassifizierung der Standorte eines mobilen Roboters in verschiedene semantische Kategorien wie *Raum*, *Korridor* und *Tür* anhand von 2D-Laserdaten. Diese 2D-Laserdaten lagen hierbei in Form von 2D-Punktwolken vor. Aufgenommen wurden die Daten von einem Roboter, der in der Mitte einer Umgebung steht und im Gegenuhrzeigersinn die Koordinaten des Treffpunkts zwischen dem Laser und der Umgebung ermittelt. In Abbildung 3.7 ist ein beispielhafter Laserscan und die ermittelten Punkte zu sehen. Da die aufeinanderfolgenden Messungen räumlich aufeinander bezogen

3 Stand der Technik

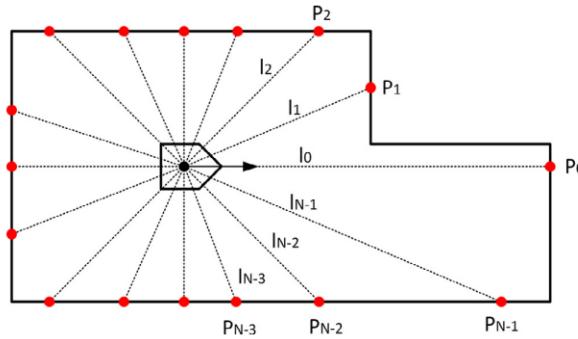


Abbildung 3.7: Ein Beispiel für einen Laserscan und den dazugehörigen Punkten. Der schwarze Punkt auf dem Roboter stellt den Ursprung des Entfernungsmessers und der Pfeil die Ausrichtung des Roboters dar. [27]

und in kanonischer Reihenfolge angeordnet sind, ist die erzeugte Punktwolke ebenfalls von Natur aus geordnet. Im Gegensatz zu bestehenden punktbasierten Deep-Learning-Verfahren nutzt 2DLaserNet die sequentielle Anordnung der Punkte aus um effektiver zu lernen.

Die 2DLaserNet Architektur ist in Abbildung 3.8 dargestellt. Als Eingabe erhält das Modell eine geordnete 2D-Punktmenge, welche in der ersten Schicht durch eine Faltungsschicht aus 32 Filtern der Größe 11 verarbeitet werden. In der darauf folgenden Faltungsschicht werden 64 Features mit Filtern der Größe 5 erzeugt, welche anschließend durch eine max-pooling Schicht halbiert werden. Die dritte Faltungsschicht besteht aus 256 Features mit Filtern der Größe 5, welche von einer weiteren max-pooling Schicht gefolgt wird. Anschließend werden die erzeugten globalen Features durch drei Schichten an voll verbundenen Multi-Layer-Perzeptronen (MLP) mit 1024, 256 und 3 Knoten geleitet. Die Ausgabe der letzten MLP-Schicht gibt die Wahrscheinlichkeit der Klassen *Raum*, *Korridor* und *Tür* wieder. In allen Schichten, außer der letzten, werden ReLU und Batch-Normalisierung verwendet. Zur Evaluierung wurde 2DLaserNet auf dem Datensatz *Freiburg 79* evaluiert und mit PointNet und Pointnet++ verglichen [27]. Ihre Methode erzielte bessere Recall-Werte für die Türklasse, sowie einen überdurchschnittlichen Recall über alle Klassen.

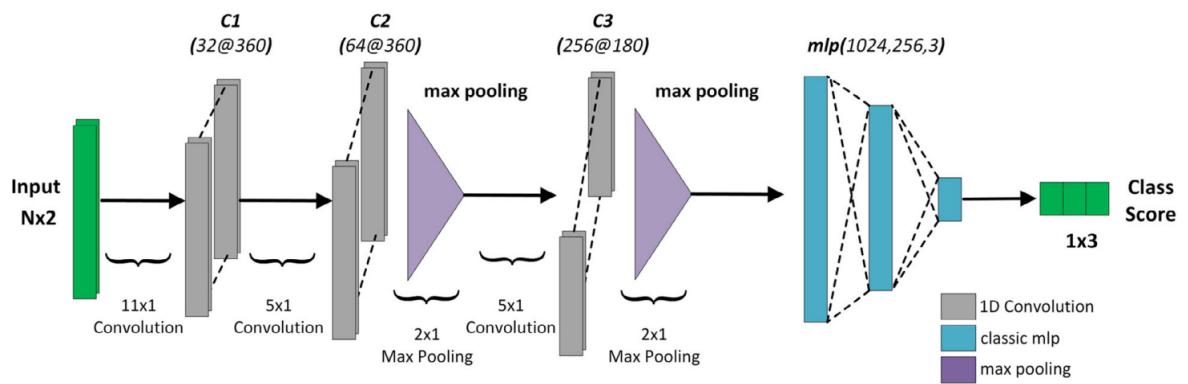


Abbildung 3.8: 2DLaserNet Architektur [27]

4 Forschungslücken & eigener Ansatz

Das Ziel der vorliegenden Arbeit ist es, neue Ansätze zur direkten Bestimmung der Kerbformzahlen von Schweißverbindungen aus Oberflächenscans mit Hilfe von KNN zu entwickeln. In diesem Kapitel werden die Forschungslücken der vorhandenen Methoden zur Bestimmung der Kerbformzahl von Schweißverbindungen bewertet und der eigene Ansatz erläutert, der sich auf die direkte Nutzung von Oberflächenscans und die Verwendung von KNN konzentriert.

4.1 Diskussion vorhandener Ansätze

Wie in Abschnitt 3.2 erläutert, existieren verschiedene Ansätze um die Kerbformzahl von realen Schweißverbindungen zu bestimmen. Hierbei stellt das Nutzen von FE-Simulationen eine Möglichkeit zur Ermittlung dar. Dabei gibt es im Allgemeinen zwei Möglichkeiten: zum einen kann ein vereinfachtes FE-Modell aus einer idealisierten Darstellung der Schweißnaht bestehend aus verschiedenen Geometrieparametern erstellt werden (siehe 2.2). Zum anderen kann basierend auf den Scandaten der realen Schweißnaht ein FE-Modell generiert werden, welches die tatsächliche Geometrie der Schweißnaht darstellt. Diese zwei Varianten sind beispielhaft in Abbildung 4.1 dargestellt. Anhand dieser Modelle kann schließlich ein Belastungsfall simuliert und die Kerbformzahl bestimmt werden. Ein Vorteil der ersten Variante ist der verringerte Modellierungs- und Rechenaufwand, welcher durch die idealisierte Geometrie gegeben ist. Problematisch kann jedoch die Verwendung einer idealisierten Geometrie der Schweißnaht sein. Diese lässt die komplexe reale Geometrie der Schweißnaht außer Acht und somit werden auch potenzielle Einflussfaktoren auf die Spannungskonzentrationsverteilung und die Kerbformzahl nicht berücksichtigt.

Im Gegensatz dazu lässt sich mit FE-Modellen, welche auf den Scandaten der realen Schweißnaht basieren, die komplexe Geometrie berücksichtigen. Untersuchungen ergaben,

4 Forschungslücken & eigener Ansatz

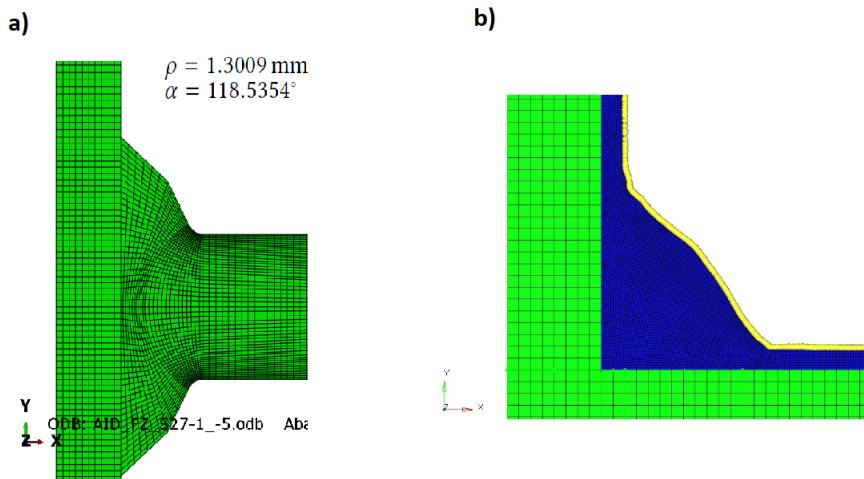


Abbildung 4.1: a) idealisiertes FE-Modell einer Schweißnaht b) FE-Modell basierend auf der realen Schweißnaht [26]

dass zwischen den Kerbformzahlen aus FE-Modellen mit idealisierter und realer Nahtgeometrie größere Abweichungen vorhanden sind [38, 49]. Die geometrienahe Modellierung ist jedoch mit hohem Modellierungs- und Rechenaufwand verbunden und somit in großem Maße nicht durchführbar [15]. Hinzu kommt, dass für große Strukturen die Überführung der realen Geometrie in FE-Modelle sehr komplex und herausfordernd sein kann [30, 51].

Kerbformzahlen wurden bisher durch analytische Näherungsformeln bestimmt (siehe 3.2). Genau wie die erwähnten vereinfachten FE-Modelle basieren auch diese Methoden auf einer idealisierten Nahtgeometrie und Nahtgeometrieparametern. Neuere Lösungen nutzen KNN, basieren aber ebenfalls auf idealisierter Nahtgeometrie und Nahtgeometrieparametern [35, 36, 17]. Wie in 3.1 erläutert, ist die Bestimmung der Geometrieparameter an einer realen Naht vom Messsystem und Algorithmus abhängig [49] und mit hohen Streuungen verbunden [16]. Abgesehen davon sind die geometrischen Parameter nicht immer eindeutig zuzuordnen und abhängig von der Scanauflösung [46]. Hinzu kommt, dass Näherungsformeln den Einfluss von komplexen Geometriestrukturen, wie zum Beispiel von Hinterschneidungen, auf die Kerbformzahl nicht erfassen können [38].

4.2 Eigener Ansatz

In dieser Arbeit wurde ein neuer Ansatz zur Bestimmung von Kerbformzahlen an Schweißverbindungen mit künstlichen neuronalen Netzen entwickelt, welcher die direkte Nutzung von Oberflächenscans ermöglichen soll, wodurch die reale Nahtgeometrie verwendet werden kann. Als Eingangsdaten werden Punktwolken verwendet, welche 2D-Schnitte der Schweißnaht darstellen. Das KNN ermöglicht durch das selbstständige Erlernen von Merkmalen die hochdimensionalen Eingangsdaten auf einen Ausgabewert, die Kerbformzahl, abzubilden. Zwei faltungsbasierter Netzwerkarchitekturen PointNet++ (siehe 3.3.2) und 2DLaserNet (siehe 3.3.3), die bisher zur Klassifizierung verwendet wurden, wurden passend zur Problemstellung für die Regression angepasst. Da PointNet++ und 2DLaserNet vielversprechende Ergebnisse in der Verarbeitung von Punktwolken aufweisen, wurden sie für diesen Anwendungszweck ausgewählt.

Gegenüber den erwähnten klassischen Methoden kann auf eine idealisierte Nahtgeometrie verzichtet werden und die Nutzung von Geometrieparametern wird obsolet. Dies bringt den Vorteil mit sich, dass die nicht standardisierte Bestimmung von Geometrieparametern vermieden wird (siehe 3.1). Außerdem soll diese Methode eine Alternative zur Bestimmung der Kerbforzahl mit FE-Simulationen ermöglichen, da diese entweder ebenfalls auf vereinfachten Geometrien basieren oder bei scannahen Modellierungen mit hohem Rechen- und Modellierungsaufwand verbunden sind.

Die Vorgehensweise dieser Arbeit ist in Abbildung 4.2 geschildert. Diese lässt sich in zwei Ansätze gliedern:

- a) Bestimmung von Kerbformzahlen an realen Schweißverbindungen auf Basis gescannter 2D-Nahtprofile (reale Nahtgeometrie)
- b) Bestimmung von Kerbformzahlen auf Basis künstlicher 2D-Nahtprofile, welche im Rahmen dieser Arbeit erstellt wurden (idealisierte Nahtgeometrie)

Beim ersten Ansatz werden 2D-Schnitte von Kehlnähten aus der Arbeit von Jung verwendet [26], während beim zweiten Ansatz künstliche Punktwolken von Stumpf- und Kehlnähten generiert werden, welche auf vorher definierten Nahtgeometrien basieren. Der zweite Ansatz bietet den Vorteil, dass ohne großen Aufwand verschiedene Formen von Schweißnahtgeometrien erzeugt werden können. Zum Training der künstlichen neuronalen Netze werden für

4 Forschungslücken & eigener Ansatz

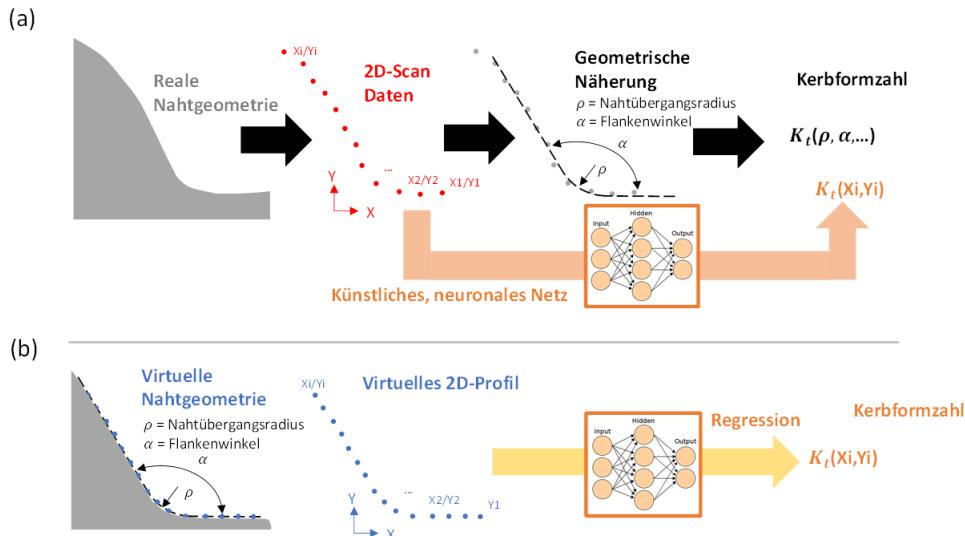


Abbildung 4.2: (a) Vorgehensweise zur Bestimmung von Kerbformzahlen an Schweißverbindungen auf Basis gescannter 2D-Nahtprofile (b) Vorgehensweise zur Bestimmung von Kerbformzahlen auf Basis künstlicher 2D-Nahtprofile [48]

Ansatz a) die Kerbformzahlen der FE-Simulationen von Jung verwendet [26]. Für Ansatz b) wurden die Kerbformzahlen durch automatisierte linear-elastische 2D-FE-Simulation generiert, welche auf den Geometrieparametern der idealisierten Schweißnähte basieren. In beiden Fällen wurde der Fall der Biegebeanspruchung ausgewertet und bei der Betrachtung der Kerbformzahl wurde nur der Nahtübergangsbereich betrachtet, da dieser die häufige Versagensstelle bei Schweißverbindungen darstellt [24].

5 Methodik

In diesem Kapitel werden die für diese Arbeit verwendeten Datensätze und deren Generierung sowie die angepassten ML-Modelle und deren Trainings- und Evaluierungsprozesse näher beschrieben. Die genutzten Datensätze lassen sich in zwei Kategorien einteilen: zum einen handelt es sich um 3D-Scandaten von realen Schweißverbindungen (Kreuzstöße) und zum anderen um künstlich erzeugte Daten, welche idealisierte Stumpf- und Kreuzstöße darstellen. Sowohl bei den realen Daten als auch bei den künstlich erzeugten Daten wurden 2D-Querschnitte der Schweißverbindung verwendet bzw. erzeugt. Bei den realen Daten wurden basierend auf den 3D-Oberflächenscans der Schweißverbindungen im festen Abstand 2D-Querschnitte extrahiert und als Daten verwendet. Die Datensätze sind folgendermaßen benannt:

Name des Datensatzes	Beschreibung	Anzahl Samples
weld_scans	reale Daten von Kreuzstößen	202
apc_kreuzstöße	künstlich erzeugte Daten von Kreuzstößen	4352
apc_stumpfstöße	künstlich erzeugte Daten von Stumpfstößen	2540

Tabelle 5.1: Die verwendeten Datensätze mit deren Beschreibung und Anzahl an Samples

5.1 Reale Daten

Der *weld_scans* Datensatz besteht aus Oberflächenscans von realen Kreuzstößen, welche aus der Arbeit von Jung [26] stammen. Bei den Proben handelt es sich um beidseitig nicht-lasttragende Quersteifen mit einer Blechdicke von 10 mm. Insgesamt standen Daten von zwei *Specimen* zur Verfügung:

- 4 Schweißnähte eines Kreuzstoß der Materialvariante S355J2+N
- 3 Schweißnähte eines Kreuzstoß der Materialvariante S960Q



Abbildung 5.1: Beispiel eines 2D-Querschnitts einer realen Schweißnaht aus dem *weld_scans* Datensatz

Für jede der Schweißnähte standen Querschnitte entlang der Naht zur Verfügung, welche mit einem Abstand von 1 mm gesetzt wurden. In Abbildung 5.1 ist beispielhaft ein 2D-Querschnitt aus dem Datensatz zu sehen. Insgesamt enthielt der Datensatz 202 verschiedene 2D-Querschnitte. Zusätzlich zu diesen Punktfolgen lagen die Kerbformzahlen der entsprechenden 2D-Nahtverläufe vor, welche anhand von geometrienahen FE-Simulationen von Jung erstellt wurden. Außerdem enthielt der Datensatz die Geometrieparameter Nahtanstiegswinkel α , Nahtübergangsradius r und die Steglänge L , welche durch die Curvature-Methode bestimmt wurden. In Abbildung 5.2 ist das Histogramm der Kerbformzahlen zu erkennen. Die Werte befinden sich zwischen 1.5 und 2.7, wobei die meisten Kerbformzahlen zwischen 1.6 und 2.3 liegen mit einem Maximum bei 1.7.

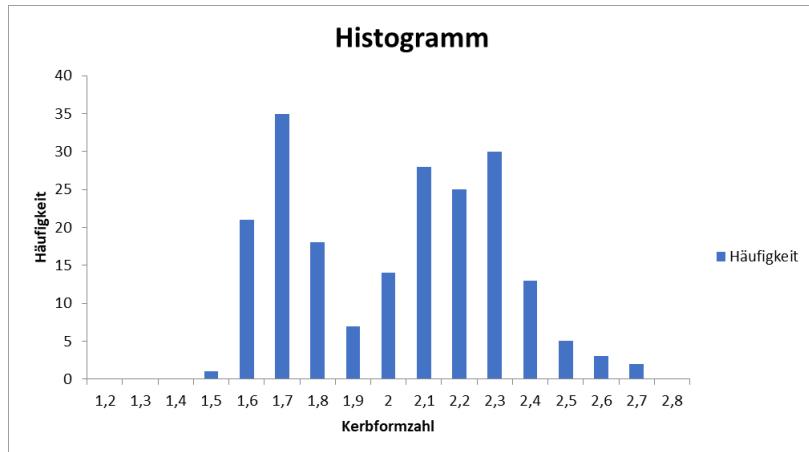


Abbildung 5.2: Histogramm der Kerbformzahlen aus dem *weld_scans* Datensatz

5.2 Künstlich erzeugte Daten

Das Schweißen von verschiedenen Schweißnähten, das Scannen dieser und die Berechnung der Kerbformzahlen ist der Realitätist mit einem großem Aufwand verbunden. Aufgrund dieser Tatsache und der geringen Anzahl an Proben von realen Schweißnähten und deren Kerbformzahlen, die für diese Arbeit zur Verfügung standen, wurden künstlich erzeugte Daten generiert. Diese ermöglichen nicht nur die ML-Modelle mit mehr Daten zu trainieren, sondern auch eine große Variation an verschiedenen Ausprägungen von Schweißverbindungen zu modellieren. Ein weiterer großer Vorteil besteht darin, dass die tatsächlichen Geometrieparameter der jeweiligen Schweißnähte bekannt sind, da diese im Vorhinein definiert und zur Erzeugung der Daten benutzt werden. Hierdurch lässt sich die manuelle Bestimmung der Geometrieparameter vermeiden. Außerdem ermöglicht dies einen besseren Vergleich mit anderen Methoden zur Berechnung der Kerbformzahl, welche auf Geometrieparametern basieren (siehe 6.6).

Zunächst wird ein geometrisches Modell der Schweißnaht festgelegt, bestehend aus verschiedenen Parametern, welche die Schweißnaht beschreiben. Diese Parameter dienen zur Erzeugung einer 2D-Punktfolge, die den Querschnitt der Schweißnaht darstellt. Gleichzeitig werden dieselben Parameter verwendet, um ein Abaqus FE-Modell zu erstellen, mit dem die Kerbformzahl berechnet werden kann. Im Anschluss daran kann das Datenpaar, bestehend aus Punktfolge und Kerbformzahl der künstlich erzeugten Schweißnaht, zum

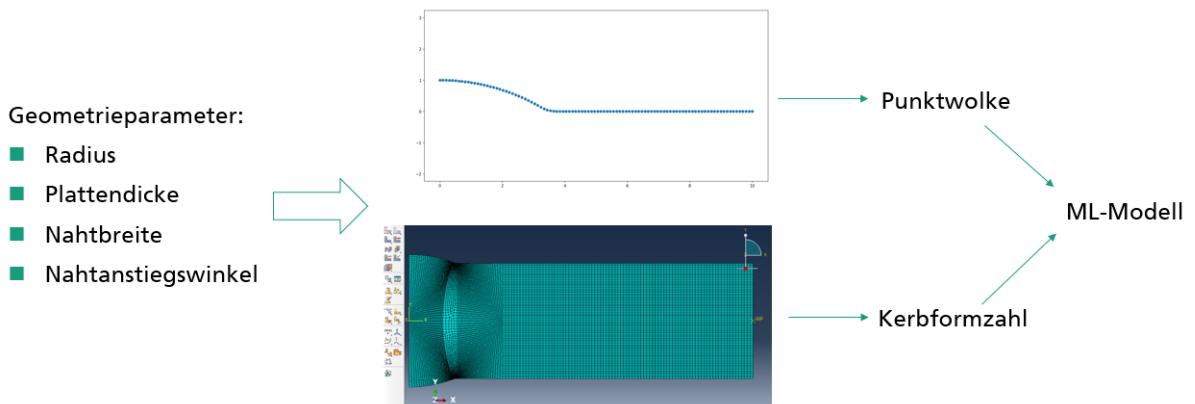


Abbildung 5.3: Beispielhafter Ablauf zur Erzeugung von künstlichen Daten eines Stumpfstoßes

Training und Evaluation des ML-Modells verwendet werden. Dieser Ablauf ist beispielhaft für einen Stumpfstoß in Abbildung 5.3 veranschaulicht. Für diese Arbeit wurden Datensätze von zwei verschiedenen Arten von Schweißnähten für den Belastungsfall der Biegung generiert: Stumpfstoße und Kreuzstoße. In den folgenden Abschnitten wird dies näher erläutert.

5.2.1 Stumpfstoß

Definition Geometrieparameter

Bei dem modellierten Stumpfstoß handelt es sich um einen voll durchgeschweißten DV-Stumpfstoß. In Abbildung 5.4 ist die schematische Darstellung des Stumpfstoßes zu sehen. Ersichtlich sind die 4 Geometrieparameter, welche die Schweißnaht beschreiben:

- Nahtübergangsradius r
- Nahtanstiegswinkel α
- Nahtbreite w
- Bleckdicke t

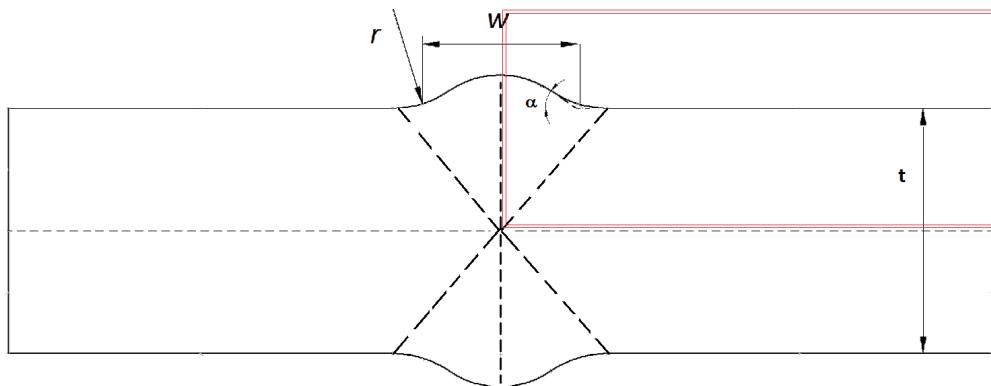


Abbildung 5.4: Schematische Darstellung des DV-Stumpfstoßes (In Anlehnung an: [37])

Die Nahtbreite ist hierbei definiert durch den Schnittpunkt der Blechoberfläche und der Tangente, durch die der Nahtanstiegswinkel bestimmt wird.

Generierung der Punktfolge

Um die Punktfolge zu erzeugen, wird die Oberfläche der Schweißnaht modelliert. Hierbei wird nur der Teil, welcher in Abbildung 5.4 rot umrahmt ist, modelliert. Diese Vorgehensweise wurde gewählt, da sie den Oberflächenscans realer Schweißnähte am nächsten kommt, die für diese Arbeit als Grundlage dienten. Außerdem ist es möglich Punktfolgen mit verschiedenen Punktabständen zu generieren. Dies ermöglicht eine Darstellung verschiedener Scanauflösungen, die in realen Schweißnähten beim Scannen mit unterschiedlichen Messsystemen auftreten können. Für die Stumpfstoße wurden Punktfolgen mit Punktabstand von 0.025 mm, 0.035 mm, 0.045 mm, 0.055 mm, 0.065 mm und 0.075 mm generiert. Die Gesamtlänge der Punktfolge ist auf 60 mm beschränkt. Dies entspricht dem halben Aufnahmebereich des Scanners SC3500-120 (120mm x 75mm), welcher für zukünftige Projekte verwendet werden soll [1]. Eine beispielhafte Punktfolge ist in Abbildung 5.5 dargestellt.

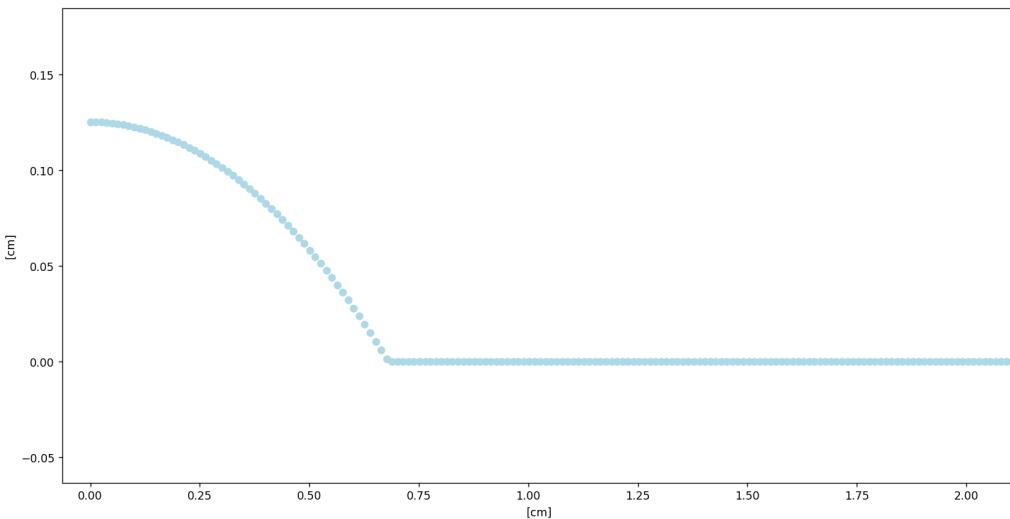


Abbildung 5.5: Ausschnitt einer künstlich erzeugten Punktwolke eines Stumpfstoßes

Parametrischer Raum

Für die Erzeugung künstlicher Daten ist es notwendig, den Wertebereich der einzelnen geometrischen Parameter zu definieren und dabei bestimmte Einschränkungen zu beachten, die auch in der realen Schweißpraxis bestehen. Damit soll erreicht werden, dass die künstlich erzeugten Daten eine bessere Darstellung realer Schweißnähte liefern können. Der Raum der Geometrieparameter ist folgendermaßen definiert:

- Nahtübergangsradius $r : \log[0.1 \text{ mm} - 5 \text{ mm}] \rightarrow 25$ Ausprägungen
- Nahtanstiegswinkel $\alpha : [5^\circ - 80^\circ] \rightarrow 25$ Ausprägungen
- Nahtbreite $w : [7 \text{ mm} - 18 \text{ mm}] \rightarrow 11$ Ausprägungen
- Blechdicke $t : 10 \text{ mm} \rightarrow 1$ Ausprägung

Der Wertebereich des Nahtübergangsradius wurde gemäß den Definitionen von Kiyak et al. [28] und der Bereich des Nahtanstiegswinkels gemäß Oswald et al. [36] festgelegt. Es besteht eine Korrelation, bei der kleine Nahtübergangsradien mit höheren Kerbformzahlen einhergehen [44]. Um diesen Zusammenhang genauer abzubilden, wird der Bereich der Nahtübergangsradien auf einer logarithmischen Skala definiert, um eine feinere Auflösung

5 Methodik

im kleinen Bereich zu ermöglichen. Der Bereich der Nahtbreite ist bestimmt anhand der Begrenzungen der Nahtvorbereitung für das Schutzgas-, Gas- und Lichtbogenschweißen aus der DIN EN ISO 9692-1 [2]. Die Blechdicke ist fix festgelegt auf 10 mm, da dieser den realen Blechdicken der vorliegenden Schweißnähte entspricht. Außerdem wurde der Einfluss des Faktors als untergeordnet bewertet.

Aus dem Intervall für den Nahtübergangsradius und den Nahtanstiegswinkel wurden jeweils 25 Werte gesampled. Aus dem Intervall der Nahtbreite wurden 11 Datenpunkte gesampled. Hierdurch ergeben sich insgesamt $25 * 25 * 11 * 1 = 6875$ Kombinationen an einzigartigen Geometrieparametern.

Zusätzlich wurde die Nahtüberhöhung, welche wie in Gleichung A.1 berechnet wird, durch die DIN EN ISO 5817 Bewertungsgruppe C beschränkt [3]. Diese gibt folgende Beschränkung vor:

$$h \leq 1 \text{ mm} + 0,15w \leq 7 \text{ mm} \quad (5.1)$$

Da bei der Kombination von kleinen Radien mit kleinen Winkeln sehr kleine Bogenlängen des Nahtübergangsradius entstehen und diese für die FE-Modelle nicht praktisch sind, wurde die Bogenlänge auf das 5-fache einer Elementkantenlänge von 0.025 mm begrenzt. Hieraus resultiert folgende Beschränkung für den Nahtübergangsradius:

$$r \geq \frac{0.025 * 5}{\alpha} * \frac{180}{\pi} \quad (5.2)$$

Unter Berücksichtigung der festgelegten Beschränkungen reduziert sich die ursprüngliche Anzahl von 6875 Kombinationen auf insgesamt 2590 Kombinationen an Geometrieparametern, welche als Datensatz genutzt werden können.

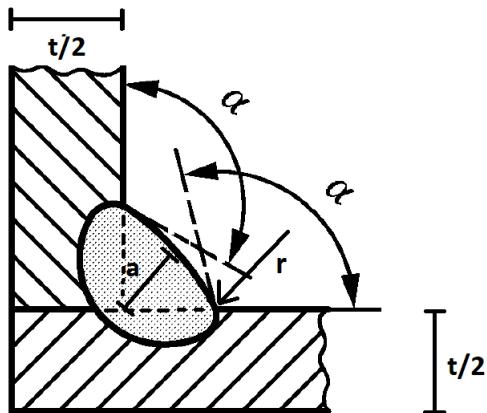


Abbildung 5.6: Schematische Darstellung der Kehlnaht (In Anlehnung an: [3])

5.2.2 Kreuzstoß

Definition Geometrieparameter

Bei dem modellierten Kreuzstoß handelt es sich um einen voll durchgeschweißten Kreuzstoß mit einer symmetrischen Kehlnaht. In Abbildung 5.6 ist die schematische Darstellung der Kehlnaht zu sehen. Ersichtlich sind die 4 Geometrieparameter, welche die Schweißnaht beschreiben:

- Nahtübergangsradius r
- Nahtanstiegswinkel α
- Nahtdicke a
- Blechdicke t

Generierung der Punktwolke

Bei dieser spezifischen Schweißverbindung wurde die Punktfolge aus dem Abaqus-Modell exportiert, was in 5.2.3 beschrieben wird. Bei der Erstellung des Abaqus Modells wird ein Set aus Kanten definiert, welches die Oberfläche der Schweißverbindung darstellt (siehe

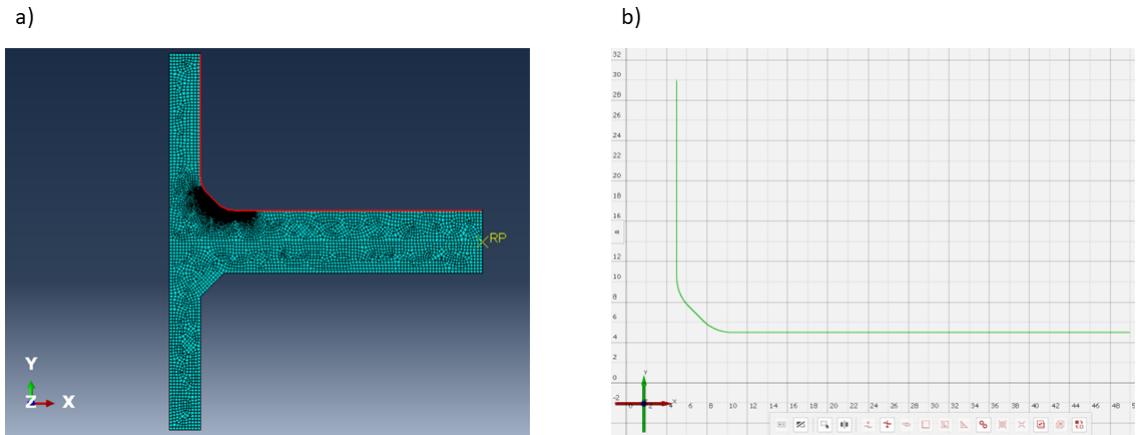


Abbildung 5.7: a) Abaqus Modell des Kreuzstoßes mit dem vordefinierten Set (in rot) zur Erzeugung der Punktfolge b) die entsprechende erzeugte Punktfolge

Abbildung 5.7). Hierbei wird nur die Oberfläche rechts der vertikalen Platte betrachtet. Aus diesem Set werden schließlich mit Hilfe des Abaqus Scripting Interface die Koordinaten der *Nodes* im Abstand von 0.025 mm extrahiert und als Punktfolge gespeichert. Die Gesamtlänge der Punktfolge wird genau wie bei den Stumpfstößen auf 60 mm beschränkt bzw. erweitert.

Parametrischer Raum

Der Raum der Geometrieparameter für die Kreuzstöße ist folgendermaßen definiert:

- Nahtübergangsradius $r: 0.1 \text{ mm} - 4 \text{ mm} \rightarrow 25$ Ausprägungen
- Nahtanstiegswinkel $\alpha: 115^\circ - 155^\circ \rightarrow 25$ Ausprägungen
- Nahtdicke $a: 2.66 \text{ mm} - 7 \text{ mm} \rightarrow 10$ Ausprägungen
- Blechdicke $t: 10 \text{ mm} \rightarrow 1$ Ausprägung

Der Wertebereich des Nahtübergangsradius wurde gemäß den Definitionen von Kiyak et. al [28] ausgewählt. Genau wie bei den Stumpfstößen wird der Bereich durch ein logarithmisches Intervall definiert, um eine feinere Auflösung im kleinen Radienbereich abzubilden.

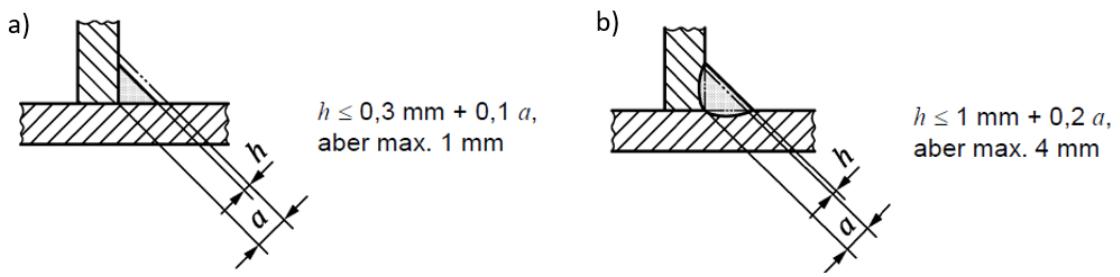


Abbildung 5.8: a) Beschränkung der Nahtunterwölbung für Kreuzstöße [2]
b) Beschränkung der Nahtüberhöhung für Kreuzstöße [2]

Der Bereich des Nahtanstiegswinkels wurde so gewählt, dass die Beschränkungen für die Nahtunterwölbung bzw Nahtüberhöhung eingehalten werden. Die Blechdicke ist wie beim Stumpfstoß festgelegt auf 10 mm, da dieser den realen Blechdicken der vorliegenden Schweißnähte entspricht. Der Bereich der Nahtdicke ist bestimmt durch die Empfehlungen der DIN 18800-1 [19] und wird ausgedrückt durch folgende Gleichungen:

$$2 \text{ mm} \leq a \leq 0.7 * t = 7 \text{ mm} \quad (5.3)$$

$$a \geq \sqrt{t} - 0.5 = 2.66 \text{ mm} \quad (5.4)$$

Aus dem Intervall des Nahtübergangsradius und des Nahtanstiegswinkels wurden jeweils 25 Werte gesampled. Aus dem Intervall der Nahtdicke wurden 10 Datenpunkte gesampled. Hierdurch ergeben sich insgesamt $25 * 25 * 10 * 1 = 6250$ Kombinationen an einzigartigen Geometrieparametern. Außerdem wurde der Nahtübergangsradius äquivalent zum Stumpfstoß beschränkt, siehe Gleichung 5.2. Die Nahtüberhöhung bzw. Nahtunterwölbung wurde gemäß der DIN EN ISO 5817 Bewertungsgruppe C [3] beschränkt (siehe Abbildung 5.8).

Unter Berücksichtigung dieser Beschränkungen verringert sich die ursprüngliche Anzahl von 6250 Parameterkombinationen auf insgesamt 4360 Kombinationen.

5.2.3 FE-Modelle

Wie bereits in Abschnitt 5.2 erläutert und in Abbildung 5.3 veranschaulicht, wurden für die Verwendung von künstlich erzeugten Daten Finite-Elemente-Modelle erstellt. Diese Modelle basieren genau wie die Punktwolken auf den vordefinierten Geometrieparametern und Modellen der beiden Schweißnahttypen. Mit Hilfe der in Abaqus erstellten FE-Modelle wurde die Kerbformzahl am Nahtübergang berechnet, in dem die maximale Hauptspannung entlang des Nahtübergangsradius evaluiert wurde. Um die Größe des Modells zu reduzieren wurde die Symmetrie der beiden Verbindungsvarianten ausgenutzt. Die Belastung wurde als Biegebelastung am äußeren Ende des Modells angesetzt. Das Material wurde rein elastisch mit einem Elastizitätsmodul von $E = 210 \text{ GPa}$ und $\nu = 0.3$ modelliert. In der Nähe des Nahtübergangs wurden quadratische *8-node plane-stress*- Elemente mit einer Elementgröße von 0.025 mm verwendet. Beispielhafte FE-Modelle des Stumpf- und Kreuzstoßes sind in Abbildung 5.9 zu sehen.

In den Abbildungen 5.10 und 5.11 sind die berechneten Kerbformzahlen als Histogramm dargestellt. Für den Stumpfstoß liegen die Werte hierbei zwischen 1.0 und 3.5 mit den meisten bei einer Kerbformzahl von ca. 1.5. Beim Kreuzstoß liegen die Kerbformzahlen zwischen 1.3 und 4.3, wobei die meisten Werte ebenfalls im Bereich um 1.5 liegen.

Die Abbildungen 5.12 und 5.13 zeigen die berechneten Kerbformzahlen in Abhängigkeit der Geometrieparameter an. Bei beiden Schweißnahttypen ist zu erkennen, dass bei kleiner werdendem Radius r die Kerbformzahl steigt. Beim Stumpfstoß ist außerdem zu erkennen, dass für größere Winkel, das heißt bei einem steilen Nahtverlauf, die Kerbformzahl ebenfalls steigt. Ähnliches lässt sich beim Kreuzstoß beobachten, bei dem ein kleinerer Winkel, also ein steilerer Anstieg, zu größeren Kerbformzahlen führt.

5 Methodik

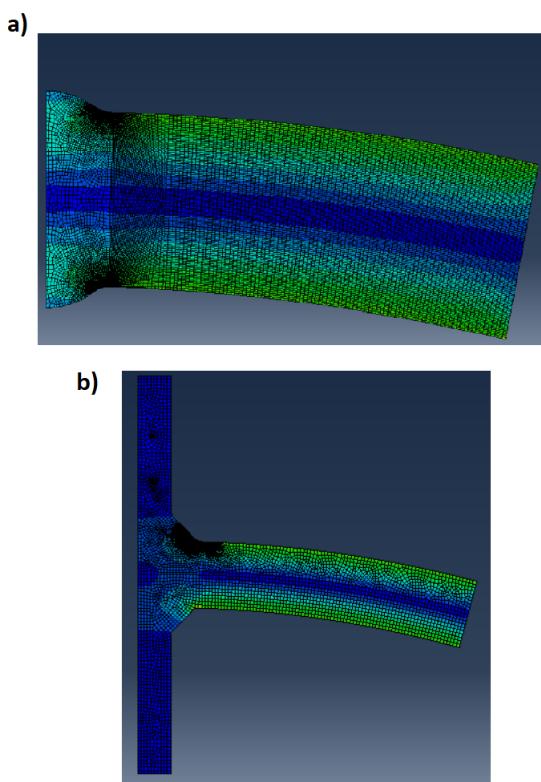


Abbildung 5.9: (a) FE-Modell des Stumpfstoßes unter Biegebelastung (b) FE-Modell des Kreuzstoßes unter Biegebelastung

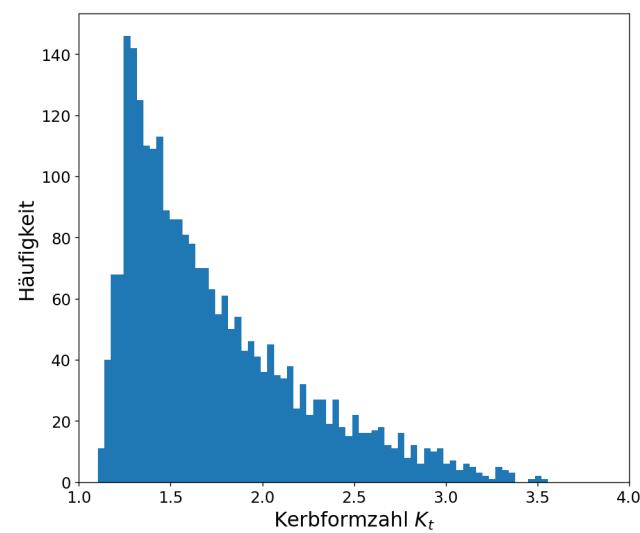


Abbildung 5.10: Histogramm der Kerbformzahlen aus den FE-Simulationen für Stumpfstöße

5 Methodik

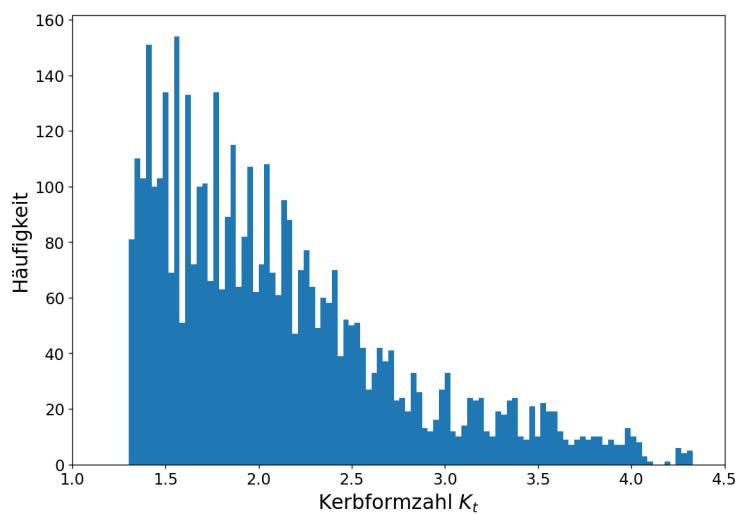


Abbildung 5.11: Histogramm der Kerbformzahlen aus den FE-Simulationen für Kreuzstöße

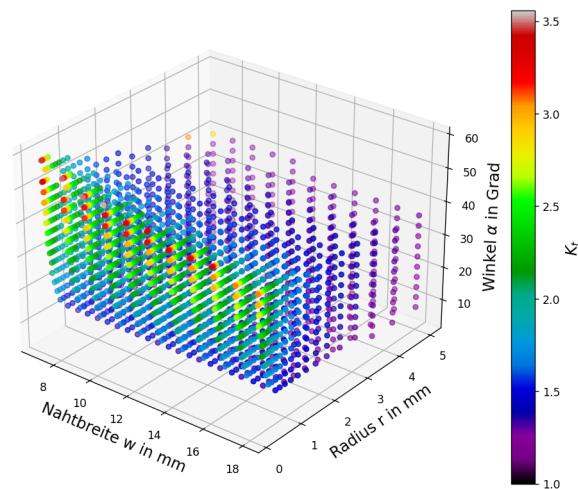


Abbildung 5.12: Die berechneten Kerbformzahlen für die Stumpfstöße in Abhängigkeit der Geometrieparameter

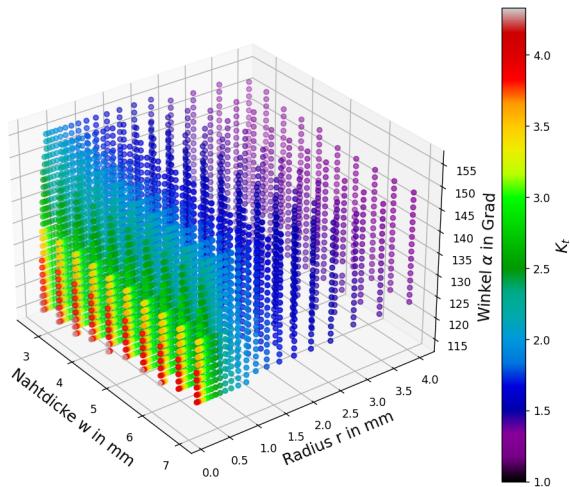


Abbildung 5.13: Die berechneten Kerbformzahlen für die Kreuzstöße in Abhängigkeit der Geometrieparameter

5.3 Datenaugmentierung

Um den Effekt der Überanpassung (siehe Abschnitt 2.2.3) zu reduzieren und die Modelle robuster zu machen, können Augmentierungen auf die Daten angewandt werden [53]. Eine Augmentierung stellt hierbei eine Veränderung der Daten mit verschiedenen Techniken dar. Hierbei kann es sich um Störfaktoren wie zum Beispiel Rauschen, Entfernen von Punkten oder Verschiebungen der Punktfolge handeln. Das Anwenden von Augmentierungen bietet den Vorteil, dass der Umfang der Trainingsdaten vergrößert und eine größere Variation geschaffen wird. Hierdurch begegnet das ML-Modell im Laufe des Trainings mehr unterschiedlichen Daten, generalisiert in der Regel besser und ist robuster [23]. Es wurde untersucht inwiefern Augmentierungen die Generalisierung der Modelle und deren Leistungsfähigkeit beeinflusst, zu finden in Abschnitt 6.4.

Es wurden insgesamt 5 verschiedene Augmentierungsmethoden implementiert, deren Anwendungswahrscheinlichkeit und Parameter vor Start des Trainings festgelegt werden können:

- Verschiebung: alle Punkte der Punktfolge in x-Richtung mit der gegebenen maximalen Bewegung in mm verschoben. Die Punktfolge wird zufällig zwischen (-max_verschiebung, +max_verschiebung) bewegt

- Drehung: die Punktwolke wird mit dem angegebenen maximalen Winkel in Grad gedreht. Die Drehung erfolgt zufällig zwischen (-max_drehung, +max_drehung)
- Rauschen: Gaußsches Rauschen (Mittelwert = 0) wird auf die y-Koordinate der Daten mit der angegebenen Standardabweichung in Mikrometer hinzugefügt
- Entfernen Punkte: unter Angabe eines Prozentsatz werden zufällig Punkte aus der Punktwolke entfernt
- Entfernen Cluster: entfernen von nebeneinander liegenden Punkten (Cluster) aus der Punktwolke mit der angegebenen Anzahl zu entfernder Punkte. Der Ort des Clusters wird zufällig gewählt

5.4 Machine Learning Modelle

Für diese Arbeit wurden die Architekturen von 2DLaserNet und Pointnet++ verwendet [27, 39]. Zur Kerbformzahlbestimmung (Regression statt Klassifikation) wurde die Architektur von PointNet++ und 2DLaserNet verwendet und modifiziert. Statt einen Vektor mit k-Klassen auszugeben, wird am Ende ein einzelner reeller Wert ausgegeben, welcher die Kerbformzahl repräsentiert. Für PointNet++ wurden außerdem Änderungen an den vollverbundenen Schichten des Netzwerks vorgenommen. Hierfür wurde zwischen der letzten linearen Schicht und der Ausgabe eine weitere lineare Schicht hinzugefügt.

5.4.1 Baseline Modelle

Als Grundlage für dies Arbeit wurden Baseline-Modelle definiert, welche im Wesentlichen als Referenz genutzt werden können um die Ergebnisse der trainierten Modelle zu kontextualisieren und zu vergleichen. Für die Baselines der Pointnet++ und 2DLaserNet-Modelle wurde eine *batch_size* von 24 für den *weld_scans* Datensatz und 68 für die *apc_kreuzstöße* und *apc_stumpfstöße* Datensätze verwendet. Die *learning_rate* wurde auf 0.001 festgesetzt und der *Adam-Optimizer* wurde mit einer *decay_rate* von 0.0001 & *beta* von 0.9 gewählt. Als Loss-Funktion wurde der Mean Absolute Error (MAE) gewählt.

5.5 Training und Validierung

Die Eingangsdaten für das Training der ML-Modelle bestehen aus Text-Dateien, welche die Punktwolken der Schweißverbindungen darstellen. Jede Datei enthält hierbei die einzelnen Punkte, welche die Punktwolke formen und aufsteigend nach der x-Koordinate sortiert sind. Jede Zeile, mit Ausnahme der ersten, repräsentiert einen Punkt der Punktwolke und besteht aus den x-,y- und z-Koordinaten, welche mit einem Leerzeichen getrennt sind. Die erste Zeile hat die Form "kt=<Zahlenwert>" und stellt die Kerbformzahl der entsprechenden Schweißnaht dar. Ein Beispiel ist in Abbildung 5.14 zu sehen.

```

1 kt=1.8290972113609314
2 0.0 0.18768202154696545 0.000000
3 0.0025010421008753647 0.18767607881083215 0.000000
4 0.005002084201750729 0.1876582501997932 0.000000
5 0.007503126302626094 0.18762853450579553 0.000000
6 0.010004168403501459 0.1875869297149625 0.000000
7 0.012505210504376824 0.18753343300691083 0.000000
8 0.015006252605252188 0.1874680407537953 0.000000
9 0.017507294706127552 0.18739074851907678 0.000000
10 0.020008336807002917 0.1873015510560161 0.000000

```

Abbildung 5.14: Beispiel: Erste 10 Zeilen der Textdatei einer Punktwolke

Da das Pointnet++-Modell ursprünglich für dreidimensionale Daten konzipiert wurde, benötigt es als Datenformat dreidimensionale Daten. Aus diesem Grund besitzen die Eingangsdaten drei Koordinaten.

Die Aufteilung des Datensatzes in Trainings- und Validierungsdatensatz erfolgt mittels k -facher Kreuzvalidierung. Die Kreuzvalidierung ist eine der am weitesten verbreiteten Methoden zur Bewertung der Generalisierungsfähigkeit eines Modells und zur Verhinderung von *Overfitting* [13]. Bei der k -fachen Kreuzvalidierung wird der ursprüngliche gesamte Datensatz nach dem Zufallsprinzip in k gleichgroße Teile sogenannten *Folds* unterteilt. Für diese Arbeit wurde $k = 5$ gewählt, da dies oft verwendet wird [11]. Von den k Folds wird ein einziger Fold als Validierungsdatensatz zum Testen des Modells genommen und die restlichen $k - 1$ Folds werden als Trainingsdaten verwendet. Der Kreuzvalidierungsprozess wird dann k -mal wiederholt, wobei jede der k Folds genau einmal als Validierungsdaten verwendet wird. Die k Ergebnisse können dann gemittelt werden, um eine einzige Schätzung zu erhalten. Der Ablauf ist beispielhaft in Abbildung 5.15 dargestellt.

5 Methodik

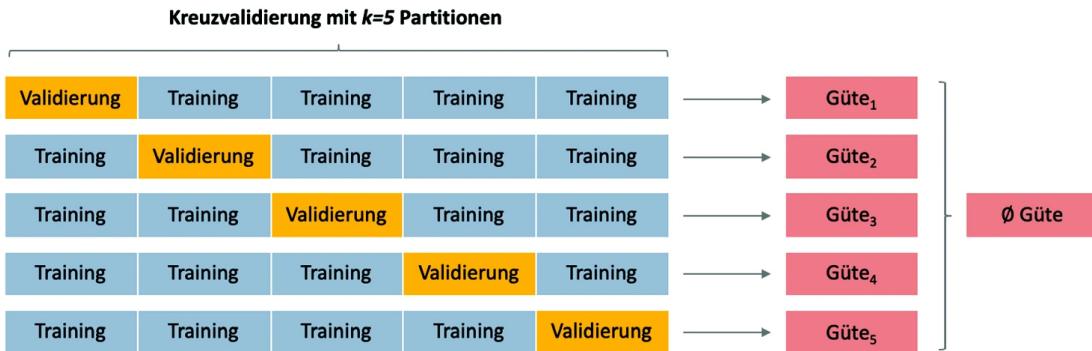


Abbildung 5.15: Beispielhafte Erklärung des Kreuzvalidierungsverfahrens mit $k = 5$ Folds [29]

Zur Erstellung des endgültigen Modells für die Vorhersage von realen zukünftigen Fällen, wird das Modell normalerweise auf dem gesamten Datensatz trainiert. Dieses endgültige Modell kann nicht kreuzvalidiert werden. Der Zweck der Kreuzvalidierung in der Phase der Modellfindung ist eine Schätzung der Leistung dieses Modells auf neuen Daten zu liefern. Eine der Vorteile dieser Methode besteht darin, dass alle Samples sowohl für das Training als auch für die Validierung verwendet werden und dass jeder Sample genau einmal für die Validierung verwendet wird. Dies ist insbesondere bei kleineren Datensätzen vorteilhaft, bei denen die Auswahl des Trainings- und Validierungsdatensatzes große Auswirkungen auf die Performance haben kann und der beobachtete Testfehler keine zuverlässige Schätzung des wahren Fehlers für neue, ungewohnte Fälle darstellt [13].

Der komplette Trainingsdurchlauf einer Iteration ist festgelegt auf 100 Epochen. Beim Laden der Samples werden die in Abschnitt 5.3 beschriebenen Augmentierungen angewandt, falls diese ausgewählt wurden für den Trainingsdurchlauf. Für die Evaluierung der Performance während des Trainings besteht die Option zwischen zwei verschiedenen Loss-Funktionen: zum einen die MAE Loss-Funktion. Diese berechnet die absolute Differenz zwischen der Vorhersage des Modells und dem erwarteten Wert geteilt durch die Anzahl der Samples (siehe Gleichung 5.5). Hierbei soll die absolute Differenz minimiert werden. Bei einer perfekten Vorhersage würde der MAE-Loss einen Wert von 0 ergeben. Alternativ kann der R2-Loss ausgewählt werden, welcher in Gleichung 5.6 definiert ist. Der R2-Loss entspricht dem bekannten negativen R2-Score, da dieser minimiert werden muss. Der R2-Loss kann nur berechnet werden, falls mindestens zwei Samples vorliegen. Für den Fall eines Samples wird der MAE-Loss verwendet. Dies kann bei einer Batchgröße von 1 passieren oder falls

nach Aufteilen des Datensatzes in Batches der letzte Batch nur noch einen Sample enthält. Bei einer perfekten Vorhersage würde der R²-Loss –1 betragen für $n > 1$. Je weiter die Vorhersage von den tatsächlichen Werten weg ist, desto größer wird der R²-Loss.

$$\text{MAE-Loss} = \frac{1}{n} \sum_{i=1}^n |y_i - f(x_i)| \quad (5.5)$$

$$R^2\text{-Loss} = \begin{cases} -(1 - \frac{\sum_{i=1}^n (y_i - f(x_i))^2}{\sum_{i=1}^n (y_i - \bar{y})^2}) & , \text{ für } n > 1, \bar{y} = \text{Mittelwert der Daten} \\ |y_i - f(x_i)| & , \text{ für } n = 1 \end{cases} \quad (5.6)$$

5.6 Implementierung

Sämtliche Skripte dieser Arbeit wurden mit Hilfe der Programmiersprache Python (Version: 3.8.15) erstellt [5]. Zur Implementierung der ML-Modelle und anderer Teile wurden die *PyTorch* [9] und *scikit-learn* [10] Bibliotheken verwendet. Die Implementierung des Pointnet++-Modells basiert hierbei auf dem Repository von Yan [56]. Zum loggen, organisieren und verwalten der Experimente wurde die *neptune-ai* Bibliothek genutzt [7]. Zum Erstellen der FE-Modelle, Simulation der Biegebeanspruchung und Auswerten der Kerbformzahlen wurde die *Abaqus FEA* Software inklusive des *Abaqus Scripting Interface*, welche eine Python-API von Abaqus darstellt, eingesetzt [18].

6 Ergebnisse und Diskussion

6.1 Baseline

Zunächst wurden Experimente durchgeführt um die Performance der Pointnet++ und 2DLaserNet Baselines auf den vorhanden Datensätzen zu evaluieren. Die Baselines wurden mit Hilfe der in Abschnitt 5.4.1 beschriebenen Parameter ausgewählt und anhand der in Abschnitt 5.5 beschriebenen Vorgehensweise trainiert und evaluiert. Hierbei wurde eine 5-fold Kreuzvalidierung angewandt. Es wurden drei verschiedene Datensätze verwendet: reale Daten von Kreuzstößen (202 Samples), künstliche erzeugte Stumpfstöße mit Punktabstand 0.025 mm (2540 Samples) und künstlich erzeugte Kreuzstöße mit Punktabstand 0.025 mm (4352 Samples). Die beiden Modelle wurden separat auf alle drei Datensätze trainiert und evaluiert.

In Abbildung 6.1 sind die Verläufe des Trainings- und Validierungsloss der Pointnet++ und 2DLaserNet Baselines auf den verschiedenen Datensätzen über alle 100 Epochen zu sehen. Hierbei ist der Mittelwert der 5 *Folds* als durchgezogene Linie mit der korrespondierenden Standardabweichung dargestellt.

Alle Kurvenverläufe zeigen auf, dass sich der Trainingsloss im Laufe des Trainings verringert und gegen Ende einem Minimum nähert. Für 2DLaserNet beträgt der finale MAE-Loss 0.13, 0.21 und 0.11 für die Datensätze *weld_scans*, *apc_kreuzstöße* und *apc_stumpfstöße*. Für Pointnet++ betragen die finalen MAE-Loss Werte 0.29, 0.15 und 0.15 für *weld_scans*, *apc_kreuzstöße* und *apc_stumpfstöße*. Die Kurven des Validierungsloss folgen bei beiden Modellen für *weld_scans* ungefähr dem Trainingsloss. Für Pointnet++ sind jedoch größere Schwankungen über alle Folds zu sehen. Für den *apc_kreuzstöße* Datensatz ist bei Pointnet++ zu sehen, dass der Validierungsloss nach kurzem Fallen ab Epoche 15 über 1.0 hinausgeht, danach aber tendenziell bis zum Ende wieder sinkt. Im Vergleich dazu sind bei 2DLaserNet keine größeren Ausreißer nach oben erkennbar. Für den *apc_stumpfstöße*

6 Ergebnisse und Diskussion

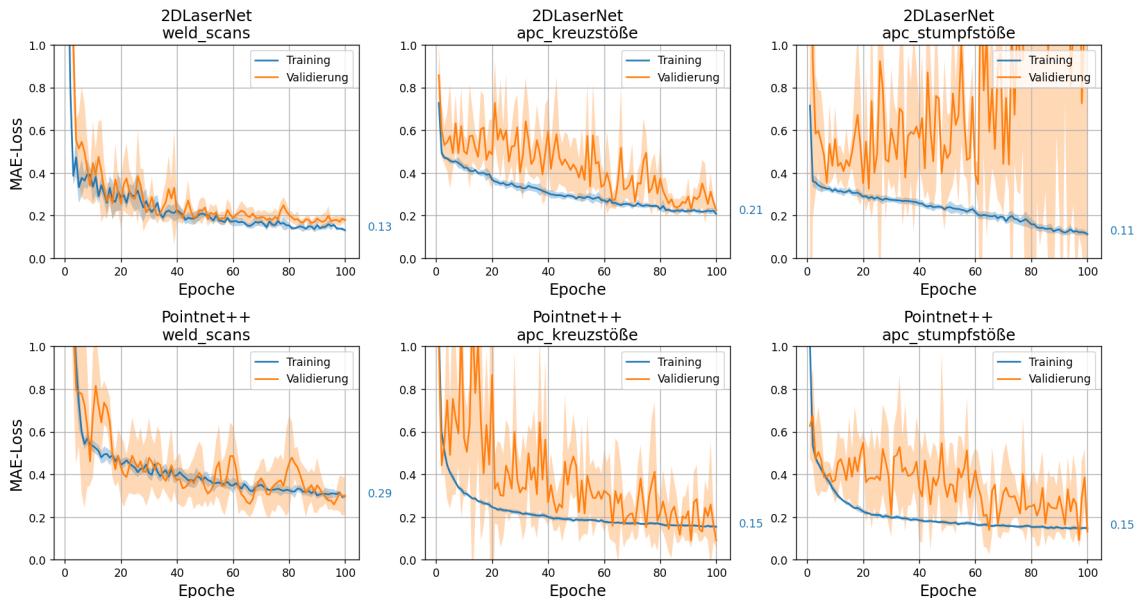


Abbildung 6.1: Kurvenverlauf des MAE-Loss der Pointnet++ und 2DLaserNet Baselines. Die durchgezogene Linie repräsentiert den Durchschnitt über 5 Folds, wobei die Standardabweichung um den Durchschnitt herum dargestellt ist

Datensatz ist bei 2DLaserNet zu sehen, dass der Validierungsloss ab ca. Epoche 10 bis zum Ende des Trainings tendenziell stark ansteigt. Im Vergleich dazu ist bei Pointnet++ eine sinkende Entwicklung des Validierungsloss festzustellen.

In Abbildung 6.2 ist ein Boxplot der R2-Scores für die entsprechenden Baselines und Datensätze dargestellt. Der R2-Score nimmt bei optimaler Vorhersage den Wert 1 an und kann hierbei in den negativen Bereich gehen, was eine schlechte Leistung angibt. Die mittlere Linie innerhalb der Kästen zeigt den Medianwert des R2-Scores über alle Folds an, während der Kasten um diese Linie herum 50% der Daten zwischen dem 25%- und dem 75%-Perzentil umfasst. Die schwarzen Whisker außerhalb der Box reichen bis zum 1,5-fachen des Quartilsabstand. Die übrigen Ausreißer außerhalb der Whisker liegen jenseits dieser Werte.

Für den *weld_scans* Datensatz ist zu erkennen, dass 2DLaserNet im Vergleich zu Pointnet++ bessere Ergebnisse erreicht. Mit einem Median R2-Score von ca. 0.55 bei 2DLasernet liegt dieser höher als 0.4 bei Pointnet++. Das schlechteste Ergebnis von 2DLaserNet liegt hierbei jedoch weit außerhalb des Kastens bei ca 0.25 und ist damit ähnlich zum schlechtesten Ergebnis von Pointnet++. Bei den beiden Datensätzen *apc_kreuzstöße* und

6 Ergebnisse und Diskussion

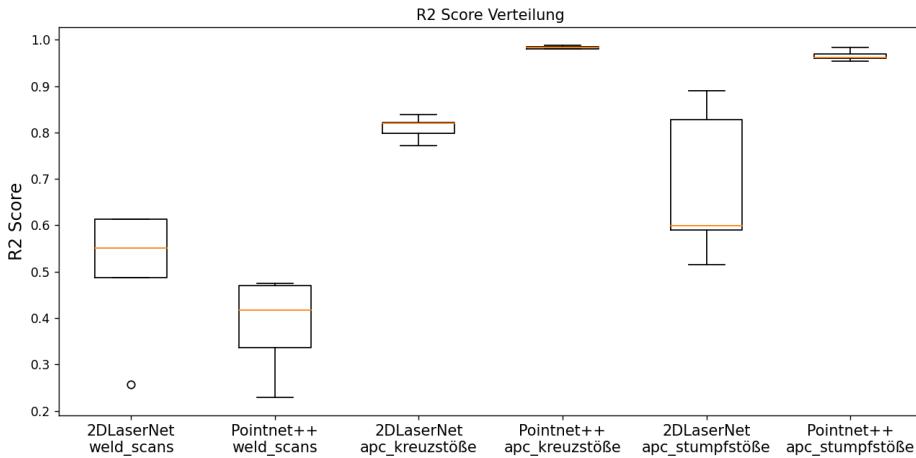


Abbildung 6.2: Boxplot des R2-Scores der Pointnet++ und 2DLaserNet Baselines (mit MAE-Loss trainiert)

apc_stumpfstöße schneidet Pointnet++ im Vergleich zu 2DLaserNet deutlich besser ab mit Medianwerten von 0.98 und 0.95 im Vergleich zu 0.8 und 0.6. Hinzu kommt, dass Pointnet++ für beide Datensätze eine sehr kleine Abweichung der Performance über alle Folds aufweist, während für 2DLaserNet der 50%-Bereich bei *apc_stumpfstöße* einen größeren Bereich abdeckt.

In Abbildung 6.3 sind die prognostizierten Kerbformzahlen der künstlichen neuronalen Netze im Vergleich mit den Kerbformzahlen der FE-Modelle zu sehen. Die gestrichelte Linie stellt hierbei die optimale Vorhersage dar, bei der die Kerbformzahlen der KNNs mit denen der FE-Modelle übereinstimmt. Die dargestellten Datenpunkte stammen aus dem Testdatensatz des besten Folds.

Für den *weld_scans* Datensatz entsprechen die ermittelten Kerbformzahlen von 2DLaserNet von ca. 1.5 denen des FE-Modells. Bei den Kerbformzahlen größer 2.0 sind jedoch größere Abweichungen zu erkennen. Pointnet++ zeigt im größeren Wertebereich der Kerbformzahl ebenfalls Vorhersagen auf, die weiter weg von der Idealgeraden liegen. Beim *apc_kreuzstöße* Datensatz sind zwischen den beiden Modellen deutliche Unterschiede zu erkennen. Während Pointnet++ in der Lage ist für nahezu den kompletten Wertebereich Kerbformzahlen zu ermitteln, die sehr nah an der Idealgeraden liegen, weist 2DLaserNet ungenauere Kerbformzahlen mit größerer Streuung auf. Auch für den *apc_stumpfstöße* Datensatz sind größere Streuungen der Kerbformzahl für 2DLaserNet zu sehen. Die Vorher-

6 Ergebnisse und Diskussion

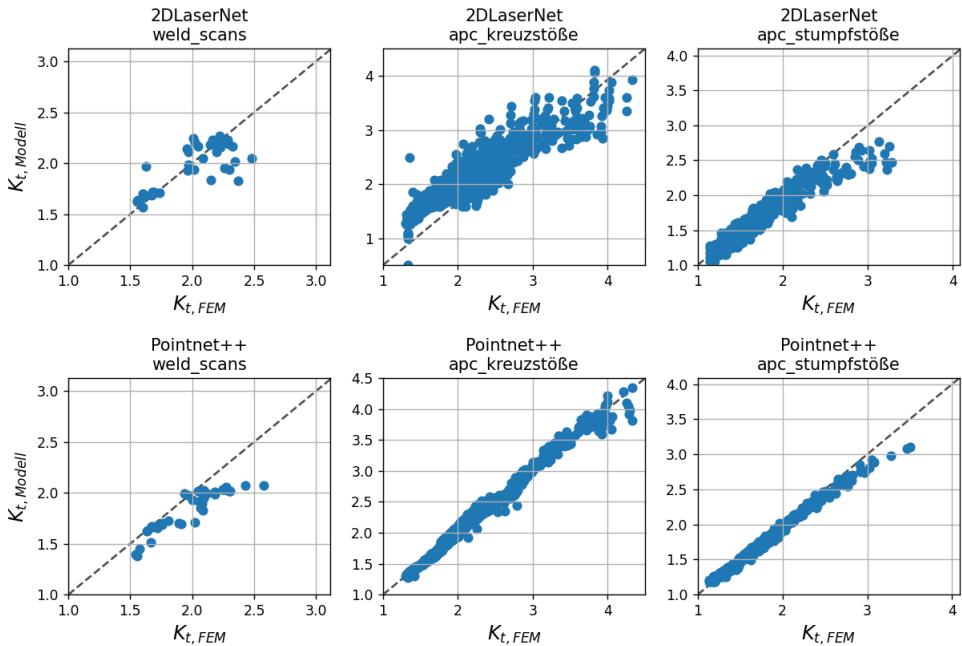


Abbildung 6.3: Die vorhergesagten Kerbformzahlen des besten Folds der Pointnet++ und 2DLaserNet Baselines (mit MAE-Loss trainiert) im Vergleich mit den Kerbformzahlen der entsprechenden FE-Modelle . Datenpunkte stammen aus dem Testset

sagen von Pointnet++ liegen hingegen größtenteils nah an der Idealgeraden. Ausschließlich für Werte größer 3.0 unterschätzt das Modell die Kerbformzahl.

In Abbildung 6.4 ist ein Boxplot des relativen Fehlers des besten Folds zu sehen. Der relative Fehler ist folgendermaßen definiert:

$$\text{relativerFehler} = \frac{K_{t,\text{Modell}} - K_{t,\text{FEM}}}{K_{t,\text{FEM}}} * 100[\%] \quad (6.1)$$

Ein relativer Fehler größer 0 bedeutet, dass das Modell die Kerbformzahl überschätzt, während ein Fehler kleiner 0 einer Unterschätzung entspricht.

Für den *weld_scans* Datensatz weist 2DLaserNet einen Median von ca. 0% auf mit vereinzelten Ausreißern bei ca. 20% und -20%. Im Vergleich dazu liegt der relative Fehler von Pointnet++ bei ca. -10% und zeigt keine größeren Ausreißer auf. Für den *apc_kreuzstöße* Datensatz zeigen beide Modelle einen ähnlichen mittleren Fehler von 0%. Bei 2DLaserNet ist jedoch der Fehlerbereich deutlich größer und die minimalen und maximalen Fehler liegen jeweils bei ca. -30% und 30%. Für PointNet++ sind ebenfalls einige Punkte außerhalb der

6 Ergebnisse und Diskussion

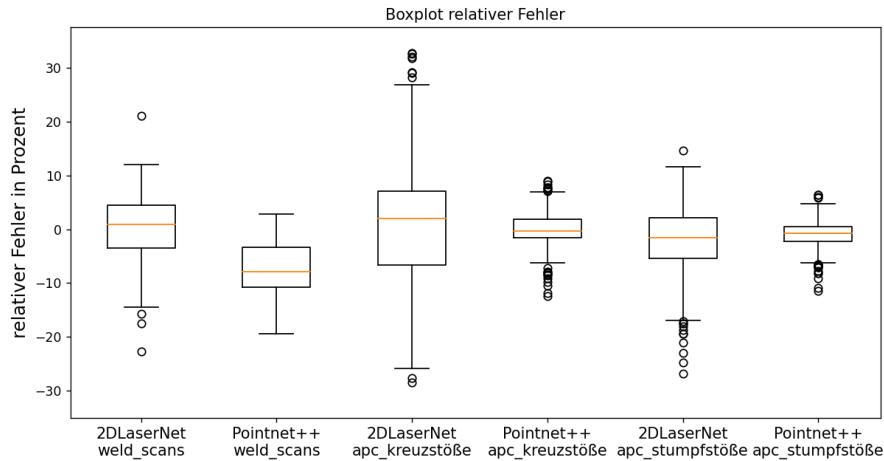


Abbildung 6.4: Boxplot des relativen Fehlers des besten Folds der Pointnet++ und 2DLaserNet Baselines (mit MAE-Loss trainiert)

Whisker-Grenzen zu sehen. Diese zeigen einen minimalen bzw. maximalen relativen Fehler von ca. 10% auf. Beim *apc_stumpfstöße* Datensatz besitzen beide Modelle ebenfalls einen ähnlichen mittleren Fehler von ca. 0%. Ähnlich wie beim *apc_kreuzstöße* Datensatz sind bei 2DLaserNet jedoch Ausreißer im größeren Fehlerbereich zu erkennen.

In Tabelle 6.1 sind die Pearson Korrelationskoeffizienten zwischen den einzelnen Geometrieparametern und dem absoluten Fehler des besten Folds dargestellt. Die Geometrieparameter für die *weld_scans* Datenpunkte entstammen den Auswertungen der Curvature-Methode aus der Arbeit von Jung [26]. Eine negative Korrelation bedeutet, dass bei steigendem Wert des Parameters der Fehler sinkt und bei sinkendem Wert der Fehler steigt. Eine positive Korrelation gibt an, dass bei steigendem oder sinkendem Wert des Parameters der Fehler ebenfalls steigt oder sinkt. Bei einer Korrelation von ungefähr Null sind die Variablen unkorreliert, das heißt, zwischen ihnen besteht kein linearer Zusammenhang.

Für keine der Einträge ist eine signifikante Korrelation zu erkennen. Die größte Korrelation ist für den Parameter *Radius* bei Pointnet++ auf dem *apc_kreuzstöße* Datensatz zu erkennen mit -0.435. Dies bedeutet, dass bei Schweißnähten mit kleinerem Radius das Modell sich schwerer tut und größere Fehler in der Vorhersage der Kerbformzahl aufweist. Außerdem besagt dies, dass bei größer werdendem Radius das Modell bessere Vorhersagen gibt. Ähnliches lässt sich bei 2DLaserNet auf dem *apc_kreuzstöße* Datensatz feststellen. Hier liegt die größte Korrelation ebenfalls beim Radius mit einem Wert von -0.354. Für die restlichen

6 Ergebnisse und Diskussion

Datensatz	Modell	Geometrieparameter			
		Nahtdicke α	Winkel α	Radius r	Nahtbreite w
weld_scans	2DLaserNet	-0.206	-0.034	-0.169	-
	Pointnet++	-0.106	-0.203	-0.259	-
apc_kreuzstöße	2DLaserNet	0.11	-0.125	-0.354	-
	Pointnet++	0.013	-0.124	-0.435	-
apc_stumpfstöße	2DLaserNet	-	0.193	-0.084	-0.162
	Pointnet++	-	0.226	-0.141	-0.138

Tabelle 6.1: Pearson Korrelationskoeffizient zwischen den Geometrieparametern und dem absoluten Fehler

Modell und Datensatz Kombinationen liegen keine nennenswerten Korrelationen größer 0.3 oder kleiner -0.3 vor.

Diskussion

Die Ergebnisse der beiden Modelle, Pointnet++ und 2DLaserNet, wurden auf den verschiedenen Datensätzen anhand des MAE-Loss und des R2-Scores bewertet. Anhand der R2-Scores ist ersichtlich, dass Pointnet++ für die Datensätze *apc_kreuzstöße* und *apc_stumpfstöße* besser abschneidet als 2DLaserNet, mit Medianwerten von 0.98 und 0.95 im Vergleich zu 0.8 und 0.6. Die in diesem Fall sehr guten R2-Scores für Pointnet++ zeigen, dass das Modell in der Lage ist Muster und Merkmale der künstlich generierten Daten zu erlernen und auf neue ungewohnte Daten zu generalisieren. Im Fall von *apc_stumpfstöße* zeigt 2DLaserNet eine große Varianz in der Performance (R2-Score) über alle Folds auf. Dies deutet auf eine hohe Sensitivität des Modells gegenüber der Auswahl des Testsets hin. An dem Kurvenverlauf des MAE-Loss lässt sich außerdem erkennen, dass das Modell sich zwar gut an die Trainingsdaten anpasst (mit einem MAE-Loss von 0.11), jedoch nicht in der Lage ist auf die Testdaten zu verallgemeinern. Anhang des Kurvenverlaufs wird eine Überanpassung des Modells ersichtlich. Dies lässt sich erklären anhand des sinkenden Trainingsloss während der Validierungsloss relativ früh ab Epoche 10 ansteigt. Insgesamt lässt dies also auf eine mangelnde Generalisierung von 2DLaserNet für *apc_stumpfstöße* führen.

Für den *weld_scans* Datensatz zeigt sich, dass 2DLaserNet bessere Ergebnisse erzielt als Pointnet++. Der Median R2-Score von 2DLaserNet liegt bei ca. 0.55, während Pointnet++

6 Ergebnisse und Diskussion

einen Wert von etwa 0.4 aufweist. Der niedrige bis mittlere R2-Score in beiden Fällen lässt darauf hindeuten, dass beide Modelle sich nicht angemessen an die Daten anpassen. Es ist wichtig zu berücksichtigen, dass die Leistung eines KNN oft stark von der Größe und Qualität des Trainingsdatensatzes abhängt. In diesem Fall liegt nur eine begrenzten Anzahl von 202 Trainingssamples vor, welche durch die Aufteilung auf Trainings- und Validierungsdatensatz sogar noch weiter auf 161 reduziert wird. Es besteht die Möglichkeit, dass das Modell aufgrund der geringen Datenmenge nicht die Fähigkeit entwickelt, komplexe Zusammenhänge zu erfassen, die für eine zuverlässige Vorhersage notwendig sind.

Im Fall von Pointnet++ lässt sich anhand des Boxplots des relativen Fehlers erkennen, dass das Modell für *weld_scans* dazu tendiert die Kerbformzahlen zu unterschätzen. Diese Beobachtung ist von besonderer Bedeutung, da niedrig geschätzte Kerbformzahlen den Eindruck vermitteln können, dass die betrachtete Schweißverbindung einer höheren Belastbarkeit standhält als tatsächlich der Fall ist. Dies kann insofern problematisch werden, falls es sich um sicherheitskritische Anwendungen handelt, bei denen Sicherheit und Integrität von hoher Relevanz sind.

Anhand des Korrelationskoeffizienten wird ersichtlich, dass beide Modelle für den Datensatz *apc_kreuzstöße* einen Zusammenhang mit dem Geometrieparameter Nahtübergangsradius r aufweisen. Wie in 2.1 erläutert, stellt der Nahtübergangsradius einen maßgeblichen Parameter dar, welcher die Belastung einer Schweißnaht beeinflusst, was diese Korrelation erklären könnte.

6.2 R2-Loss

In diesem Teil wurden Experimente durchgeführt um das Verhalten der Performance mit der *R2-Lossfunktion* zu beobachten, welcher in 5.6 definiert ist. Die Modelle wurden nach dem selben Vorgehen wie in 6.1 trainiert mit dem Unterschied, dass statt dem MAE-Loss die R2-Lossfunktion verwendet wurde.

In der Tabelle 6.2 sind die durchschnittlichen R2-Scores über alle Folds zu sehen. Die Werte links des Pfeils wurden durch Anwendung des MAE-Loss ermittelt, während die Werte rechts des Pfeils unter Verwendung des R2-Loss erreicht wurden. Für 2DLaserNet konnte beim *weld_scans* Datensatz eine Verbesserung der Leistung von 0.50 auf 0.57 erzielt werden. Für *apc_kreuzstöße* blieb die Leistung auf einem ähnlichen Level mit einem

6 Ergebnisse und Diskussion

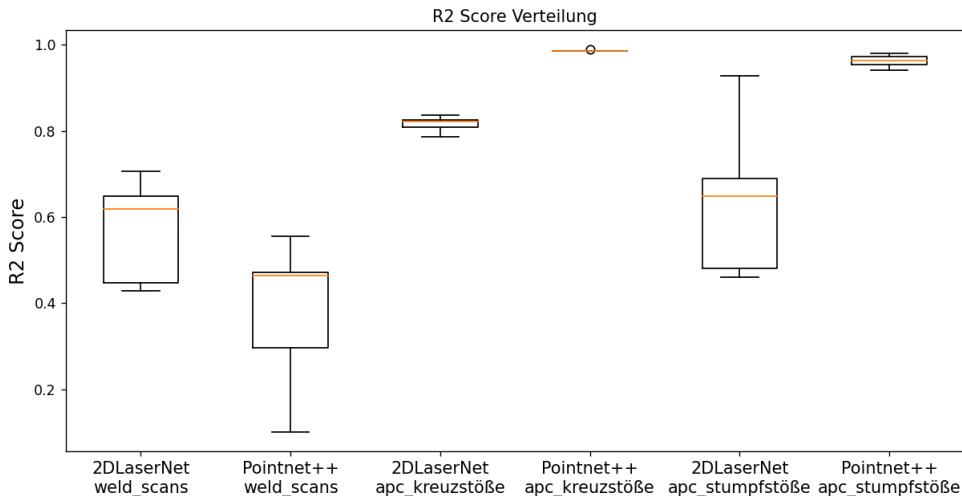


Abbildung 6.5: Boxplot des R2-Scores von Pointnet++ und 2DLasernet (mit R2-Loss trainiert)

Anstieg von 0.81 auf 0.82. Auf dem *apc_stumpfstöße* Datensatz verringerte sich der R2-Score von 0.68 auf 0.64. Die Leistung des Pointnet++ Modells veränderte sich über alle Datensätze hinweg kaum durch den Einsatz des R2-Loss. Beim Vergleich der Boxplots von 6.2 und 6.5 lässt sich ebenfalls erkennen, dass der Median R2-Score für 2DLaserNet beim *weld_scans* Datensatz von 0.55 auf ca. 0.62 anstieg. Hinzu kommt, dass kein Ausreißer mit dem R2-Score ersichtlich ist.

Modell	Datensatz		
	weld_scans	apc_kreuzstöße	apc_stumpfstöße
2DLaserNet	0.50 → 0.57	0.81 → 0.82	0.68 → 0.64
Pointnet++	0.39 → 0.38	0.98 → 0.99	0.96 → 0.96

Tabelle 6.2: Vergleich durchschnittliche R2-Scores über alle Folds. R2-Score mit MAE-Loss → gefolgt von R2-Score mit R2-Loss

In der Tabelle 6.3 sind die Pearson Korrelationskoeffizienten zwischen dem absoluten Fehler und den Geometrieparametern aufgelistet. Eine starke negative Korrelation ist für den Geometrieparameter Radius bei Pointnet++ auf dem *weld_scans* Datensatz mit -0.445 und bei 2DLaserNet auf dem *apc_kreuzstöße* Datensatz mit -0.471 zu erkennen. Dies bedeutet, dass in beiden Fällen bei größer bzw. kleiner werdendem Radius der Fehler sich verringert bzw. vergrößert. Für 2DLaserNet ist außerdem für den Winkel eine starke Korrelation von 0.502 ersichtlich. Dies bedeutet, dass das Modell sich bei größeren Winkel, das heißt einem

6 Ergebnisse und Diskussion

Datensatz	Modell	Geometrieparameter			
		Nahtdicke a	Winkel α	Radius r	Nahtbreite w
weld_scans	2DLaserNet	-0.243	-0.171	-0.132	-
	Pointnet++	0.042	-0.154	-0.445	-
apc_kreuzstöße	2DLaserNet	0.035	-0.123	-0.471	-
	Pointnet++	-0.021	-0.038	-0.207	-
apc_stumpfstöße	2DLaserNet	-	0.502	-0.168	-0.16
	Pointnet++	-	0.022	0.036	-0.095

Tabelle 6.3: Pearson Korrelationskoeffizient zwischen den Geometrieparametern und dem absoluten Fehler

steilen Nahtanstieg, schwerer tut die Kerbformzahl korrekt zu bestimmen. Alle weiteren Werte zeigen keine nennenswerte Korrelation auf.

Der Vollständigkeit halber sind die Kurvenverläufe des Trainings (A.2), die bestimmten Kerbformzahlen im Vergleich mit den FE-Modellen (A.3) und der Boxplot des relativen Fehlers (A.4) im Anhang zu finden.

Diskussion

Die Anwendung der R2-Lossfunktion wurde durchgeführt um zu untersuchen, wie sich die Leistung von Pointnet++ und 2DLaserNet unter Verwendung einer anderen Lossfunktion verändert. Die Ergebnisse zeigen, dass 2DLaserNet beim *weld_scans* Datensatz durch den Einsatz des R2-Losses eine Verbesserung der Leistung von 0.50 auf 0.57 erzielte. Eventuell könnte dies darauf hindeuten, dass der R2-Loss für diesen Datensatz besser geeignet ist. Es muss jedoch beachtet werden, dass dieser Datensatz nur eine sehr geringe Anzahl an Datenpunkten aufweist. Somit sind Erkenntnisse, die aus den vorliegenden Ergebnissen gezogen werden, stets mit Vorsicht zu betrachten. Beim *apc_stumpfstöße* Datensatz verringerte sich der R2-Score des 2DLaserNet-Modells von 0.68 auf 0.64, was auf eine mögliche Schwäche des R2-Losses bei der Handhabung bestimmter Eigenschaften dieses Datensatzes hindeuten könnte. Interessanterweise zeigten die Ergebnisse für das Pointnet++-Modell im Vergleich zum 2DLaserNet-Modell nur geringfügige Veränderungen in der Leistung, durch den Einsatz des R2-Losses. Dies legt nahe, dass das Pointnet++-Modell möglicherweise robust gegenüber der Wahl der Lossfunktion ist und sich durch den

6 Ergebnisse und Diskussion

Einsatz des R2-Losses kaum beeinflussen lässt. In fast allen Fällen zeigte der Einsatz des R2-Losses jedoch nur marginale Veränderungen in der Leistung der Modelle auf, was darauf hin deutet, dass beide Lossfunktionen zur Anwendung geeignet wären.

6.3 Vergleich verschiedener Punktabstände

In der Realität verfügen Scanner stets über unterschiedliche Scanauflösungen, die je nach verwendetem Messsystem variieren können. Um dies zu berücksichtigen und die Auswirkungen des Punktabstands auf die Pointnet++ und 2DLaserNet Modelle zu untersuchen, wurde der *apc_stumpfstöbe* Datensatz mit verschiedenen Punktabständen $\in \{0.025 \text{ mm}, 0.035 \text{ mm}, 0.045 \text{ mm}, 0.055 \text{ mm}, 0.065 \text{ mm}, 0.075 \text{ mm}\}$ erstellt. Die Modelle wurden mit Hilfe der in Abschnitt 5.4.1 beschriebenen Parameter ausgewählt und anhand der in Abschnitt 5.5 beschriebenen Vorgehensweise mit 5-fold Kreuzvalidierung und R2-Lossfunktion trainiert und validiert.

In Abbildung 6.6 sind die R2-Scores der beiden Modelle in Abhängigkeit des Punktabstands für den *apc_stumpfstöbe* Datensatz dargestellt. Bei dem kleinsten Punktabstand von 0.025 mm erzielte 2DLaserNet einen durchschnittlichen R2-Score von 0.683 mit einer Standardabweichung von 0.11. Für die größeren Punktabstände von 0.035 mm und 0.045 mm erreichte das Modell einen durchschnittlichen R2-Score von 0.6 und 0.8 mit Standardabweichungen von 0.1 und 0.07. Der durchschnittliche R2-Score von 0.8 für 0.045 mm stellt hierbei die beste Leistung des Modells dar. Bei den Punktabständen 0.055 mm, 0.065 mm und 0.075 mm zeigt das Modell eine große Varianz der Leistung auf, mit Standardabweichungen von 0.24, 0.21 und 0.27. Im Falle des größten Punktabstands von 0.075 mm liegt der R2-Score am niedrigsten bei 0.35.

Pointnet++ zeigt hingegen deutlich konstantere Leistungen bei größer werdendem Punktabstand auf. Für alle Punktabstände erreicht das Modell einen durchschnittlichen R2-Score von ca. 0.95 mit sehr geringen Standardabweichungen.

6 Ergebnisse und Diskussion

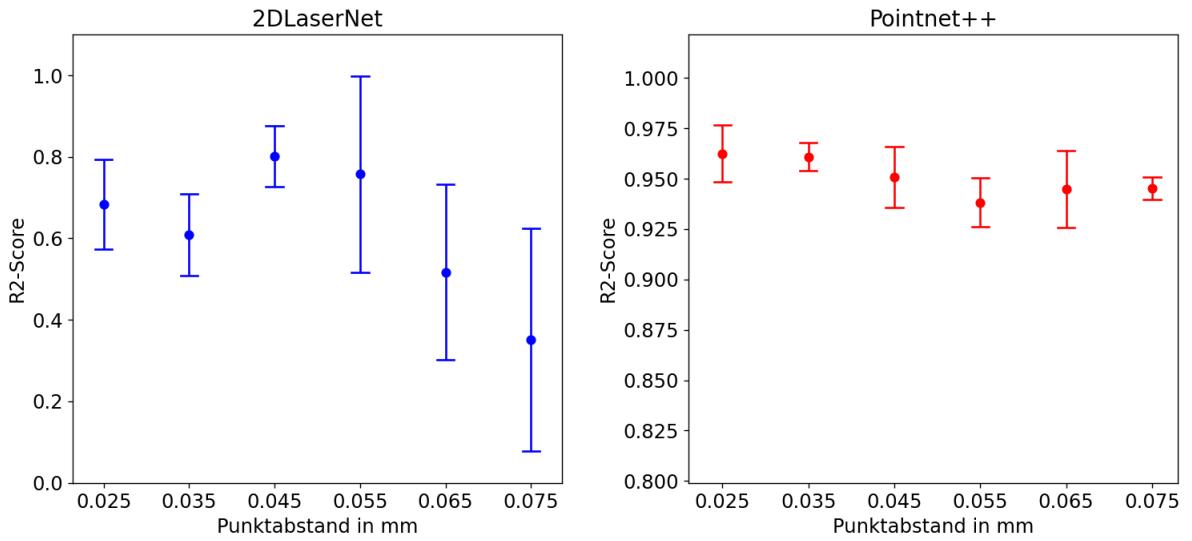


Abbildung 6.6: R2-Scores in Abhängigkeit des Punktabstandes für den *apc_stumpfstöße* Datensatz. Die Punkte stellen den Durchschnitt über alle 5 Folds dar mit entsprechender Standardabweichung.

Diskussion

Zusammenfassend lässt sich sagen, dass 2DLaserNet empfindlich auf den gewählten Punktabstand reagiert. Ein kleinerer Punktabstand führt tendenziell zu besseren Vorhersagen, während ein größerer Punktabstand die Leistung des Modells beeinträchtigen kann. Dies lässt sich vor allem anhand der niedrigeren R2-Scores bei den Punktabständen 0.065 mm und 0.075 mm feststellen. Hinzu kommt die große Standardabweichung für diese Punktabstände und für den Punktabstand 0.055 mm, welche auf eine höhere Sensitivität des Modells gegenüber der Punktauflösung deuten. Bei einer höheren Punktauflösung enthält die Punktwolke mehr Punkte, was dem Modell ermöglicht feinere Details in der Geometrie der Schweißnaht zu erkennen. Dies kann zu einer höheren Genauigkeit bei der Vorhersage der Kerbformzahl führen aufgrund der größeren Informationsmenge. Auf der anderen Seite können Punktwolken mit geringer Punktauflösung weniger Informationen über die Geometrie enthalten, was die Schwierigkeiten des Modells erklären könnte wichtige Merkmale zu erfassen, die für die Vorhersage der Kerbformzahl bedeutend sind. Es ist jedoch zu betonen, dass das 2DLaserNet Modell generell für den *apc_stumpfstöße* Datensatz eine große Sensitivität in der Performance aufweist und schnell zur Überanpassung neigt, wie in

6 Ergebnisse und Diskussion

6.1 diskutiert. Aus diesem Grund wäre es interessant zu sehen, wie sich die Leistung des Modells mit varierendem Punktabstand auf den anderen Datensätzen unterscheidet.

Im Vergleich zu 2DLaserNet fällt bei Pointnet++ auf, dass das Modell über alle Punktabstände einen hohen R2-Score erzielt. Die R2-Scores liegen zwischen etwa 0.94 und 0.96, was nicht nur darauf hindeutet, dass das Modell die Kerbformzahl mit einer guten Genauigkeit vorhersagen kann, sondern auch, dass es robust gegenüber den verschiedenen Punktabständen ist. Die Standardabweichung liegt etwa zwischen 0.005 und 0.02 bei allen Punktabständen, was darauf hindeutet, dass das Modell konsistente Vorhersagen für die Kerbformzahl liefert, unabhängig von den gewählten Trainings- und Testdaten. Eine mögliche Erklärung für die gute Leistung von Pointnet++ bei Variation des Punktabstands könnte in der Netzwerkarchitektur liegen. Pointnet++ verwendet sogenannte "density adaptive layers", welche es dem Modell ermöglichen, Merkmale aus Regionen mit unterschiedlichen Maßstäben zu kombinieren, je nachdem wie sich die Punktedichte der Eingangsdaten ändert. Dies ist besonders relevant in Situationen, in denen die Punktabstände in den Eingabedaten variieren, wie es bei den Experimenten dieses Abschnitts der Fall ist.

6.4 Datenaugmentierung

In diesem Teil wurde untersucht inwiefern die Anwendung der in 5.3 genannten Augmentierungsmethoden die Leistungen von Pointnet++ und 2DLaserNet beeinflusst. Hierfür wurden die Modelle mit Hilfe der in Abschnitt 5.4.1 beschriebenen Parameter ausgewählt und anhand der in Abschnitt 5.5 beschriebenen Vorgehensweise mit 5-fold Kreuzvalidierung und R2-Lossfunktion trainiert und validiert. Die ausgewählten Datenaugmentierungen wurden während jeder Epoche auf den Trainingsdaten angewendet. Die Anwendung der gewählten Augmentierung erfolgte mit einer bestimmten Wahrscheinlichkeit und die Stärke der Augmentierung wurde zufällig variiert. Insgesamt wurden die in Tabelle 6.4 gelisteten Datenaugmentierungen untersucht. Zusätzlich zu den 5 einzelnen Augmentierungen wurden Trainingsdurchläufe erstellt bei denen alle Augmentierungen angewandt wurden mit den entsprechenden Parametern der jeweiligen Augmentierung.

In der Abbildung 6.7 sind die verschiedenen Kurvenverläufe des R2-Scores für den Trainingsdatensatz zu sehen. Im Anhang A.5 befinden sich außerdem die Verläufe für den

6 Ergebnisse und Diskussion

Augmentierung	Anwendungs-wahrscheinlichkeit	Parameter
Verschiebung	75 %	max. Verschiebung in x-Richtung: $\pm 5 \text{ mm}$
Drehung	60 %	maximale Rotation: $\pm 14^\circ$
Rauschen	100 %	Standardabweichung: $1.1 \mu\text{m}$
Entfernen Punkte	50 %	Anteil entfernt: 5 %
Entfernen Cluster	20 %	Anzahl Punkte entfernt: 35
Alle zusammen	-	-

Tabelle 6.4: Die angewandten Datenaugmentierungen und ihre Parameter

Validierungsdatensatz. Bei den Kurven auf den Trainingsdatensatz lässt sich feststellen, dass bei allen Trainings die Leistung unter Anwendung aller Augmentierungen am schlechtesten ist. Für 2DLaserNet betragen diese zwischen 0.2 und 0.4. Außerdem ist erkennbar, dass 2DLaserNet das beste Ergebnis erzielt unter Anwendung der Augmentierung *Rauschen* für alle drei Datensätze. Für Pointnet++ auf dem *weld_scans* Datensatz ist eine große Varianz in der Leistung des Modells für alle Augmentierungen zu erkennen und die R2-Scores liegen in diesem Fall alle im schlechten negativen Bereich. Im Vergleich dazu zeigt 2DLaserNet auf *weld_scans* deutlich bessere Ergebnisse mit R2-Scores bis zu 0.65 für die Augmentierung *Rauschen*. Der R2-Score für Pointnet++ auf *apc_kreuzstöße* liegt für alle Augmentierungen bei ca. 0.8, während für 2DLaserNet dieser je nach Augmentierung zwischen 0.25 und 0.75 liegt. Für *apc_stumpfstöße* ist bei Pointnet++ ein deutlicher Unterschied zwischen der Leistung mit den Augmentierungen *Drehung* & *Alle zusammen* und den restlichen Augmentierungen zu erkennen.

In der Tabelle 6.5 sind die R2-Scores auf dem Validierungsdatensatz mit deren Standardabweichung über alle 5 Folds zu sehen. Die besten R2-Scores für die jeweilige Kombination von Modell und Datensatz ist hierbei fett markiert. Auf dem *weld_scans* Datensatz konnten die Augmentierungen *Rauschen* und *Entfernen Cluster* die Leistung des Modells von einem R2-Score von 0.56 auf 0.60 heben. Für Pointnet++ hingegen zeigte keine der Augmentierungen eine Leistungsverbesserung auf diesem Datensatz auf. Für die Augmentierung *Drehung* ist sogar eine deutliche Verschlechterung erkennbar. Für *apc_kreuzstöße* weist Pointnet++ einen konstanten R2-Score von 0.98 unabhängig der Augmentierung vor. Die Leistung von 2DLaserNet ist hierbei ohne Augmentierungen mit 0.81 genauso gut wie mit den Augmentierungen *Verschiebung* und *Rauschen*. Auch auf dem Datensatz *apc_stumpfstöße* besitzt Pointnet++ über alle Augmentierungsmethoden eine gute Leistung mit einem R2-Score

6 Ergebnisse und Diskussion

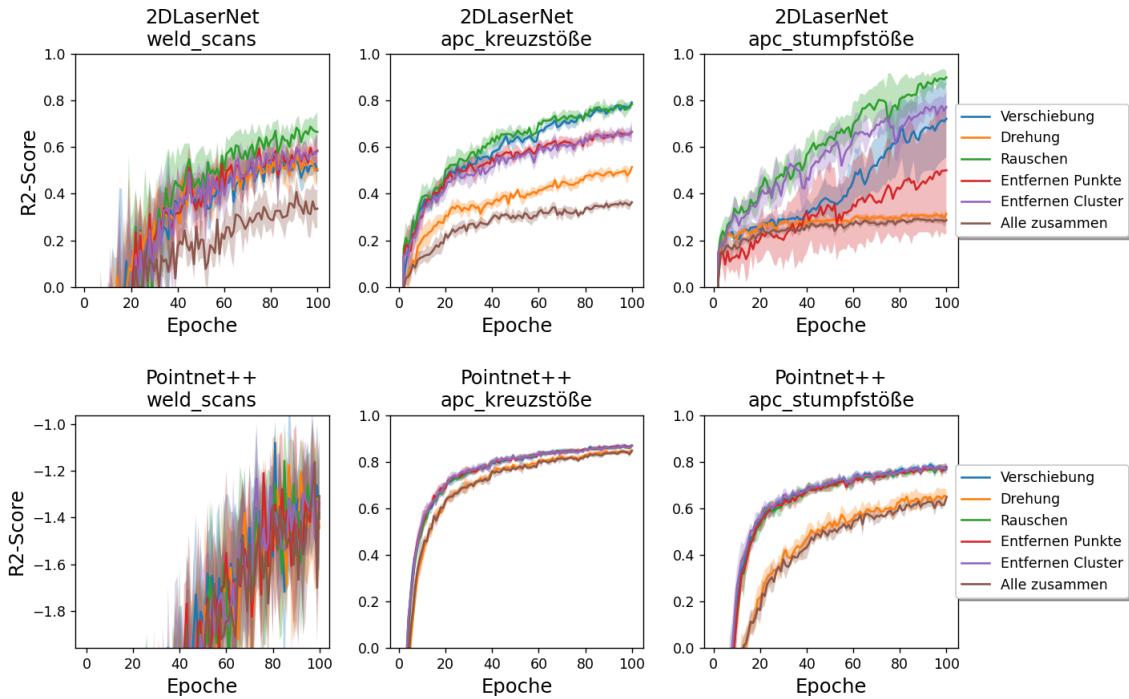


Abbildung 6.7: Kurvenverlauf des R2-Scores für den Trainingsdatensatz

zwischen 0.9 und 0.96. Mit der Augmentierung *Rauschen* konnte 2DLaserNet eine deutliche Leistungssteigerung von 0.64 auf 0.85 auf diesem Datensatz erreichen.

Diskussion

Die Augmentierungen zeigten unterschiedliche Wirkungen auf die Leistung der Modelle auf.

Im Fall des *weld_scans* Datensatz zeigte sich 2DLaserNet sehr robust und konnte durch die

Datensatz		weld_scans		apc_kreuzstöße		apc_stumpfstöße	
Modell		2DLaserNet	Pointnet++	2DLaserNet	Pointnet++	2DLaserNet	Pointnet++
Augmentierungsmethode	Ohne	0.56 +- 0.11	0.37 +- 0.16	0.81 +- 0.01	0.98 +- 0.00	0.64 +- 0.16	0.96 +- 0.01
	Verschiebung	0.57 +- 0.06	-0.02 +- 0.14	0.81 +- 0.01	0.98 +- 0.00	0.75 +- 0.13	0.96 +- 0.00
	Drehung	0.57 +- 0.1	-0.37 +- 0.21	0.57 +- 0.01	0.98 +- 0.00	0.34 +- 0.02	0.90 +- 0.01
	Rauschen	0.60 +- 0.1	-0.1 +- 0.4	0.81 +- 0.01	0.98 +- 0.00	0.85 +- 0.07	0.95 +- 0.01
	Entfernen Punkte	0.58 +- 0.12	-0.17 +- 0.13	0.72 +- 0.00	0.98 +- 0.00	0.52 +- 0.23	0.96 +- 0.01
	Entfernen Cluster	0.60 +- 0.1	-0.19 +- 0.13	0.70 +- 0.03	0.98 +- 0.00	0.58 +- 0.14	0.95 +- 0.00
	Alle zusammen	0.54 +- 0.06	-0.11 +- 0.11	0.44 +- 0.01	0.97 +- 0.00	0.34 +- 0.02	0.89 +- 0.02

Tabelle 6.5: R2-Scores für die angewandten Datenaugmentierungen

6 Ergebnisse und Diskussion

Augmentierungen *Rauschen* und *Entfernen Cluster* sogar die Leistung von einem R2-Score von 0.56 auf 0.60 verbessern. Im Vergleich dazu zeigte PointNet++ auf diesem Datensatz keine Verbesserungen durch die angewendeten Augmentierungen und wies sogar eine Verschlechterung auf, wobei die R2-Scores in den negativen Bereich fielen. Dies könnte auf darauf hindeuten, dass 2DLaserNet für diesen Datensatz besser verallgemeinert und gemacht ist. Aufgrund der begrenzten Datensatzgröße von 202 Datenpunkten sollten die Erkenntnisse an größeren Datensätzen überprüft werden.

Für die beiden künstlich erzeugten Datensätze *apc_kreuzstöße* und *apc_stumpfstöße* zeigte Pointnet++ hingegen eine hohe Robustheit auf mit R2-Scores von 0.89 und höher bei allen Augmentierungen. Die Augmentierung *Rauschen* konnte bei allen drei Datensätzen für 2DLaserNet die Leistung konstant halten oder sogar verbessern wie zum Beispiel beim *apc_stumpfstöße* Datensatz mit einer Steigerung von 0.64 auf 0.85. Dies deutet auf eine hohe Wirksamkeit der Rauschen-Augmentierung für 2DLaserNet hin.

In der Regel sollen Augmentierungen dazu führen, dass die Modelle besser generalisieren. Durch das Erzeugen von künstlichen Variationen und Störungen in die Trainingsdaten soll verhindert werden, dass die Modelle sich zu sehr auf die Trainingsdaten anpassen, sodass sie besser auf ungewohnte Datenpunkte generalisieren können. Die Augmentierungen dieses Experiments, welche zu keiner besseren Generalisierung führen könnte durch verschiedene Faktoren erklärbar sein. Es könnte sein, dass die gewählten Augmentierungen möglicherweise nicht gut zu den Modellen oder zu den Datensätzen passen und die Daten zu sehr verzerrt. In diesem Zusammenhang ist es gut möglich, dass die in Tabelle 6.4 gewählten Augmentierungsparameter zu hoch gewählt wurden und damit wichtige Muster und Strukturen in den Datensätzen verwischen. Die Stärke mit welcher die Augmentierungen angewandt werden und die Anwendungswahrscheinlichkeit sind Hyperparameter, die optimiert werden können.

6.5 Transfer künstliche Daten auf reale Daten

In diesem Teil wurde die Leistungsfähigkeit der Modelle untersucht von künstlich erzeugten Daten auf reale Daten von Kreuzstößen zu übertragen. Hierfür wurden die Modelle auf dem kompletten *apc_kreuzstöße* Datensatz mit R2-Loss und ohne die Anwendung von Kreuzvalidierung trainiert und auf dem *weld_scans* Datensatz evaluiert.

6 Ergebnisse und Diskussion

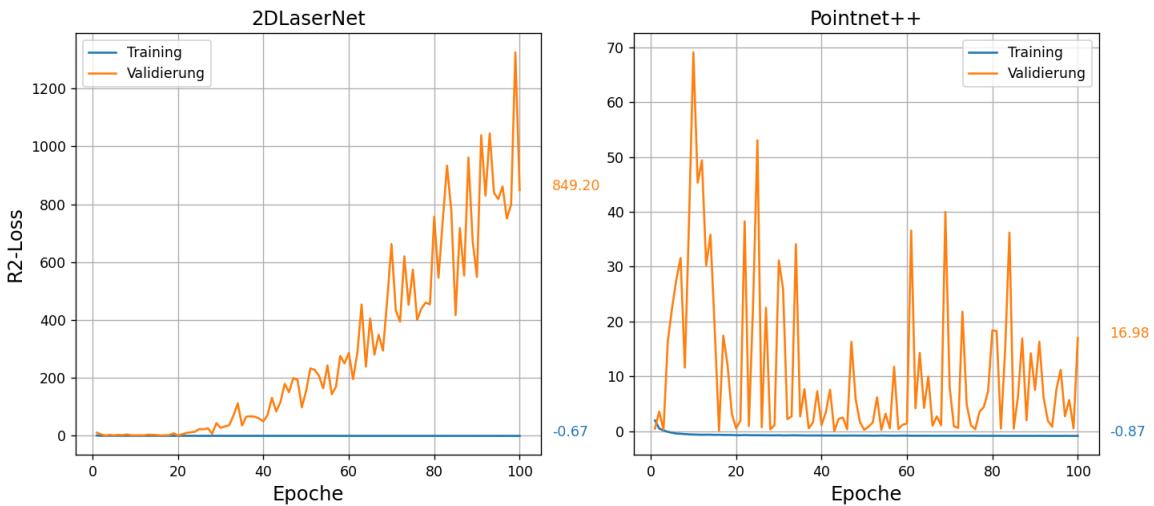


Abbildung 6.8: Kurvenverlauf des R2-Loss von Pointnet++ und 2DLaserNet trainiert mit *apc_kreuzstöße* und evaluiert auf *weld_scans*

In Abbildung 6.8 sind die Kurvenverläufe des R2-Loss dargestellt. Hierbei erreichen 2DLaserNet und Pointnet++ einen finalen R2-Loss von -0.67 und -0.87. Bei beiden Modellen sind große Losses auf dem Validierungsdatensatz zu verzeichnen. Für 2DLaserNet steigt dieser ab Epoche 20 stetig an bis er einen Wert von 849.20 erreicht. Bei Pointnet++ weist der Validierungsloss große Schwankungen auf mit einem finalen Wert von 16.98. Der beste Validierungsloss für die beiden Modelle betragen 0.46 für 2DLaserNet und 0.00 für PointNet++. Die ermittelten Kerbformzahlen der neuronalen Netze im Vergleich mit den Kerbformzahlen der FE-Modelle sind in Abbildung 6.9 zu sehen. Für PointNet++ sind im niedrigen Bereich der Kerbformzahlen von 1.5 bis 1.8 Vorhersagen zu erkennen, die nahe an der Idealgeraden liegen. Für Kerbformzahlen die darüber hinausgehen sind jedoch große Streuungen vorhanden. Bei 2DLaserNet zeigen sich deutliche Streuungen, wobei die Punkte scheinbar zufällig um die Idealgerade verteilt sind.

Der Boxplot des relativen Fehlers in Abbildung 6.10 lässt erkennen, dass Pointnet++ mit einem Median-Fehler von -4% deutlich besser liegt als 2DLaserNet mit einem Median-Fehler von 10%. Zudem zeigt sich, dass die Streuung des Fehlers um den Median rum bei PointNet++ geringer ist als bei 2DLaserNet.

6 Ergebnisse und Diskussion

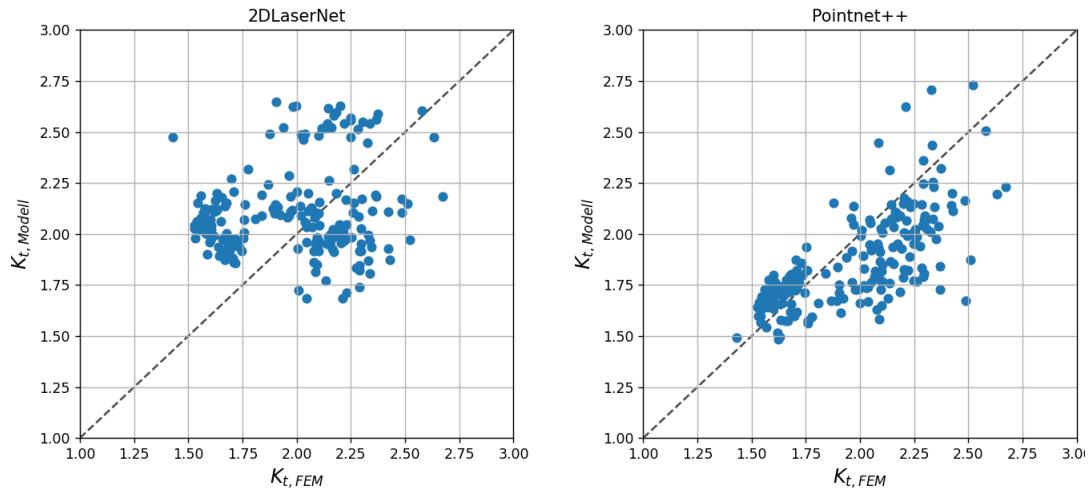


Abbildung 6.9: Die vorhergesagten Kerbformzahlen von Pointnet++ und 2DLaserNet (trainiert auf *apc_kreuzstöße* & evaluiert auf *weld_scans*) im Vergleich mit den Kerbformzahlen der entsprechenden FE-Modelle

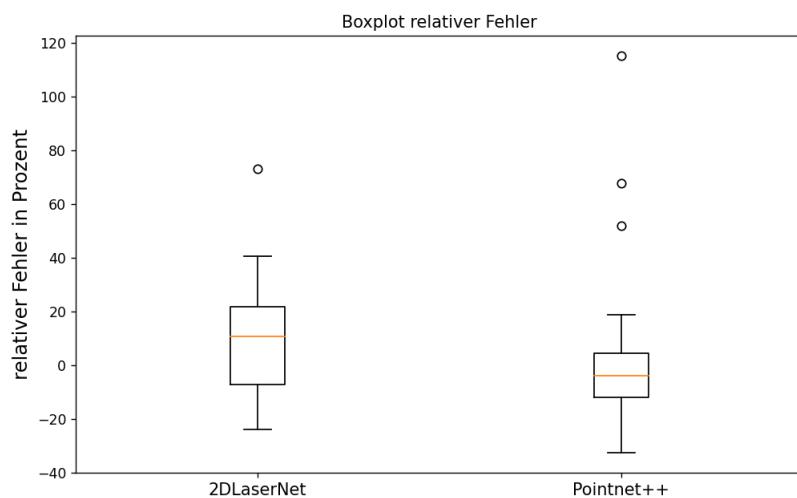


Abbildung 6.10: Boxplot des relativen Fehlers von Pointnet++ und 2DLaserNet (trainiert auf *apc_kreuzstöße* & evaluiert auf *weld_scans*)

Diskussion

Die Ergebnisse lassen darauf hindeuten, dass beiden Modellen keine gute Übertragung von künstlichen auf reale Daten gelingt. Der beste Validierungloss von 0.46 für 2DLaserNet und 0.00 für PointNet++ zeugen von keiner guten Transferleistung vom *apc_kreuzstöße* auf den *weld_scans* Datensatz. Der R2-Loss von 0.0 bedeutet, dass PointNet++ keine bessere Leistung als eine einfache Durchschnittsvorhersage hervorbringt. 2DLaserNet schneidet sogar noch schlechter ab. Bei beiden Modellen scheint eine Überanpassung aufzutreten. Im Falle von 2DLaserNet steigt diese Überanpassung sogar signifikant im Laufe des Trainings wie anhand des Kurvenverlauf des Validierungloss in 6.8 zu erkennen ist. Für Pointnet++ scheinen die Vorhersagen im niedrigen Bereich der Kerbformzahlen recht genau zu sein, da sie nahe an der Idealgeraden liegen. Für höhere Kerbformzahlen liegen jedoch große Streuungen vor. Für 2DLaserNet scheinen die bestimmten Kerbformzahlen keinem eindeutigen Muster zu folgen und zufällig zu sein.

Um eine besseren Transfer von künstlichen auf reale Daten zu erreichen, wäre es sinnvoll mögliche Anpassungen im Prozess der Datenerzeugung durchzuführen, um sicherzustellen, dass die generierten Daten eine größere Ähnlichkeit mit den realen Daten aufweisen. Die reale Geometrie einer Schweißnaht ist sehr komplex mit Oberflächenrauheiten und anderen feinen Strukturen, die durch das Material oder den Schweißvorgang gegeben sind. Diese abzubilden und in den Datengenerierungsprozess mit einzubinden, könnte eine verbesserte Transferfähigkeit mit sich bringen.

6.5.1 Transfer mit Datenaugmentierung

Zur zusätzlichen Untersuchung wurden die Modelle, welche mit Hilfe von Augmentierungen auf *apc_kreuzstöße* trainiert wurden, evaluiert auf den *weld_scans* Datensatz um die Transferleistung der augmentierten Methoden von künstlichen zu realen Daten zu betrachten. Die R2-Scores sind in Tabelle 6.6 aufgelistet. Zu erkennen ist, dass 2DLaserNet bei allen Augmentierungsmethoden sehr schlechte negative R2-Scores aufweist. Das beste Ergebnis mit Augmentierung war für *Alle zusammen* mit einem Score von -164.11. Die beste Leistung insgesamt für 2DLaserNet im Transfer von künstlichen auf reale Daten konnte ohne Augmentierungen erzielt werden mit einem R2-Score von -0.46.

6 Ergebnisse und Diskussion

Datensatz		trainiert mit <i>apc_kreuzstöße</i> & evaluiert auf <i>weld_scans</i>	
Modell		2DLaserNet	Pointnet++
Augmentierungsmehtode	<i>Ohne</i>	-0.46	0.0009
	<i>Verschiebung</i>	-1248.8	-6.86
	<i>Drehung</i>	-246.54	-1
	<i>Rauschen</i>	-835.24	-6.6
	<i>Entfernen Punkte</i>	-312.07	-2.56
	<i>Entfernen Cluser</i>	-1529.27	-8.74
	<i>Alle zusammen</i>	-164.11	-1.97

Tabelle 6.6: R2-Scores für die angewandten Datenaugmentierungen bei Training mit *apc_kreuzstöße* und Evaluierung auf *weld_scans*

PointNet++ konnte ebenfalls keine Verbesserung in der Transferleistung unter Anwendung der Augmentierungen erzielen. Der beste R2-Score blieb hierbei für den Fall ohne Augmentierungen bei 0.

Diskussion

Die vorliegenden Ergebnisse legen nahe, dass sowohl bei 2DLaserNet als auch bei Pointnet++ keine erfolgreiche Übertragung von künstlichen auf reale Daten unter Verwendung der Augmentierungsmethoden erzielt wurde. Der beste R2-Score von -0.46 für 2DLaserNet und 0 für PointNet++ wurden unter Anwendung keiner Augmentierung erzielt. Um einen besseren Transfer von künstlichen auf reale Daten zu ermöglichen, müssten vermutlich Anpassungen an den Augmentierungsmethoden vorgenommen werden, um eine größere Ähnlichkeit zwischen den generierten Augmentierungen und den realen Daten zu erreichen.

6.6 Vergleich mit anderen Methoden

In diesem Teil wurden die Pointnet++ und 2DLaserNet Modelle mit vorhandenen Methoden zur Bestimmung der Kerbformzahl verglichen. Hierfür wurden die Modelle mit dem R2-Loss auf 80% der jeweiligen Datensätze *weld_scans*, *apc_kreuzstöße* und *apc_stumpfstöße* trainiert und auf den restlichen 20% getestet. Zum Vergleich wurden die Näherungsformeln von Anthes et al. [12], Rainer [41] und Kiyak et al. verwendet [28]. Diese sind in A.1.1, A.1.2

6 Ergebnisse und Diskussion

und A.1.3 zu finden. Die Formeln wurden mit Hilfe der vordefinierten Geometrieparameter für die Datensätze *apc_kreuzstöße* und *apc_stumpfstöße* ausgewertet. Für die Auswertung des *weld_scans* Datensatz wurden die Geometrieparameter aus Jungs Arbeit verwendet, welche mittels Curvature-Methode bestimmt wurden [26]. Hierbei wurden die einzelnen Geltungsbereiche der Näherungsformeln beachtet und nur Parameter, welche gültig sind ausgewertet.

Die R2-Scores jeder Methode sind für die unterschiedlichen Datensätze in Tabelle 6.7 aufgelistet. Für den *weld_scans* Datensatz zeigt 2DLaserNet die beste Leistung mit 0.7 gefolgt von PointNet++ mit 0.44. Die Näherungsformeln konnten in diesem Fall keine guten R2-Scores erreichen mit Werten von -10.03, -2.27 und -3.20 für die Methoden von Anthes, Rainer und Kiyak. Beim *apc_kreuzstöße* Datensatz kann PointNet++ die beste Leistung mit einem R2-Score von 0.99 aufweisen. Gefolgt von Kiyak mit 0.87, Anthes mit 0.85, 2DLaserNet mit 0.83 und Rainer mit 0.77. Die Methoden von Anthes und Kiyak erreichten für den *apc_stumpfstöße* Datensatz beide einen R2-Score von 0.99. Darauf folgt mit geringem Abstand PointNet++ mit 0.96, 2DLaserNet mit 0.82 und Rainers Methode mit 0.76.

Die Kerbformzahlen, welche von den einzelnen Methoden bestimmt wurden, sind im Vergleich zu den Kerbformzahlen der FE-Modelle in den Abbildungen 6.11, 6.12 und 6.13 für die Datensätze *weld_scans*, *apc_kreuzstöße* und *apc_stumpfstöße* dargestellt. Für den *weld_scans* Datensatz zeigen die Kerbformzahlen der Näherungsformeln große Abweichungen von der Idealgeraden auf. Bei Anthes und Rainer liegen sämtliche Punkte abseits der Idealgeraden und entsprechen dementsprechend in keinem Fall den Kerbformzahlen des FE-Modells. Die Methode von Kiyak weist in dem Bereich um 1.75 rum Kerbformzahlen auf die nah an der Idealgeraden liegen. Für Kerbformzahlen über diesem Wert überschätzt die Methode jedoch die resultierenden Werte. Bei den neuronalen Netzen ist bei 2DLaserNet ein deutlicher Zusammenhang zwischen den vorhergesagten Kerbformzahlen und denen des FE-Modells zu erkennen. Pointnet++ zeigt ähnliche Resultate auf, mit dem Unterschied, dass die Punkte größer 2.0 größtenteils als konstant 2.0 bestimmt werden.

Bei Betrachtung der bestimmten Kerbformzahlen für den *apc_kreuzstöße* Datensatz ist zu erkennen, dass alle Näherungsformeln einige Datenpunkte aufweisen, welche auf der Idealgeraden liegen. Gleichzeitig ist bei allen drei Methoden jedoch zu sehen, dass bestimmte Datenpunkte überschätzt werden. Im Fall von 2DLaserNet weisen die bestimmten Kerbformzahlen zwar eine Korrelation mit denen des FE-Modells auf, sind jedoch mit deutlich sichtbaren Streuungen um die Idealgerade verbunden. PointNet++ zeigt das beste

6 Ergebnisse und Diskussion

Datensatz	Methode				
	2DLaserNet	Pointnet++	Anthes	Rainer	Kiyak
weld_scans	0.70	0.44	-10.03	-2.27	-3.20
apc_kreuzstöße	0.83	0.99	0.85	0.77	0.87
apc_stumpfstöße	0.82	0.96	0.99	0.76	0.99

Tabelle 6.7: R2 Scores der verschiedenen Methoden im Vergleich

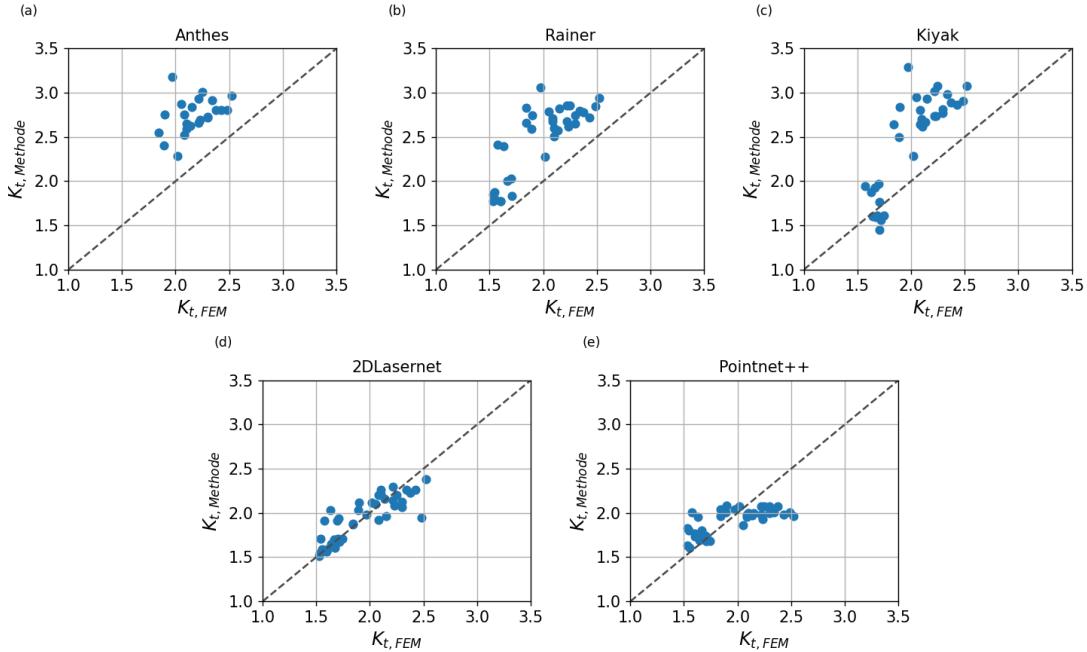


Abbildung 6.11: Die bestimmten Kerbformzahlen der verschiedenen Methoden im Vergleich für den *weld_scans* Datensatz

Ergebnis auf diesem Datensatz und scheint in nahezu allen Fällen eine übereinstimmende Kerbformzahl zu berechnen wie die FE-Modelle.

Für den *apc_stumpfstöße* Datensatz zeigt die Methode von Kiyak die größte Übereinstimmung mit den Kerbformzahlen der FE-Modelle auf. Die Näherungsformel von Anthes ist ebenfalls in der Lage mit hoher Präzision die Kerbformzahlen zu bestimmen. Im Vergleich dazu liegen bei Rainer die Werte zwar nah an der Idealgeraden, besitzen jedoch eine größere Streuung als die anderen beiden Methoden. Bei 2DLaserNet sind mehr Streuungen insbesondere für den Bereich größer 2.5 erkennbar. Die bestimmten Kerbformzahlen von Pointnet++ liegen größtenteils nah an denen des FE-Modells. Für Werte größer als 3.0 weist das Modell eine leichte Unterschätzung der Kerbformzahlen auf.

6 Ergebnisse und Diskussion

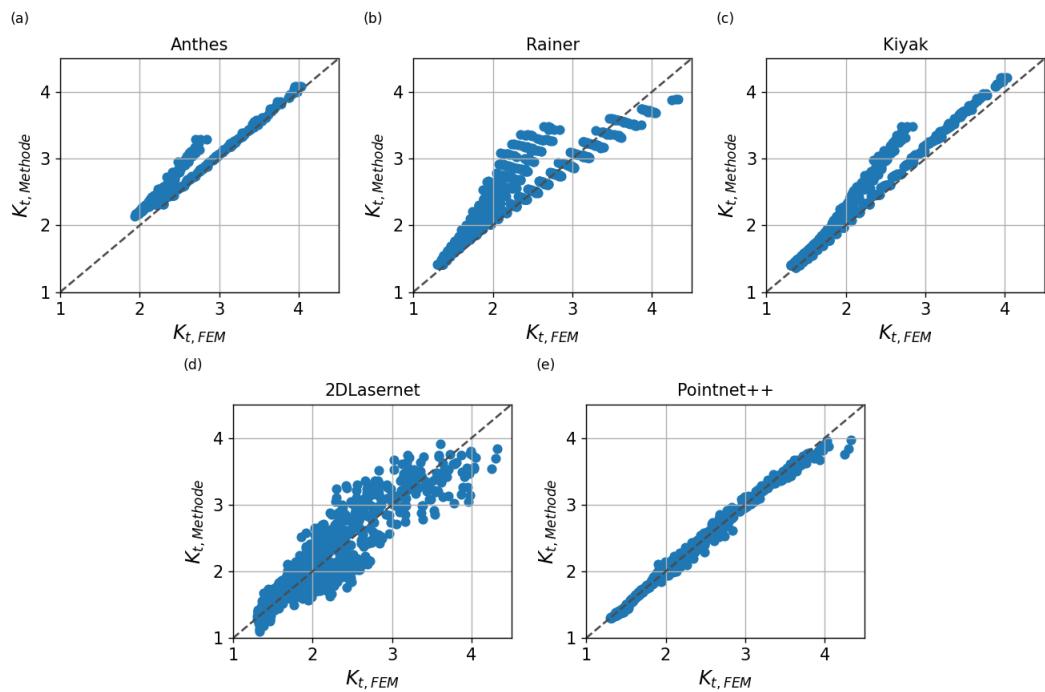


Abbildung 6.12: Die bestimmten Kerbformzahlen der verschiedenen Methoden im Vergleich für den *apc_kreuzstoße* Datensatz

6 Ergebnisse und Diskussion

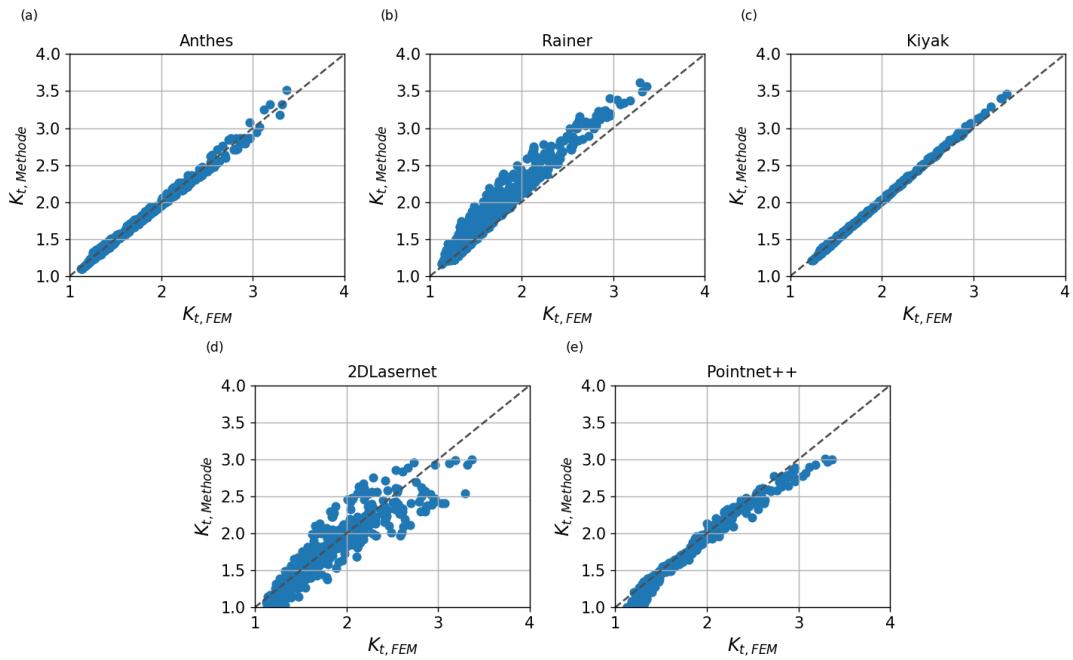


Abbildung 6.13: Die bestimmten Kerbformzahlen der verschiedenen Methoden im Vergleich für den *apc_stumpfstöße* Datensatz

Diskussion

Die Ergebnisse dieses Teils zeigen, dass die verwendeten neuronalen Netze im Vergleich zu den ausgewählten Methoden gut geeignet sind um die Kerbformzahlen der verschiedenen Datensätze zu bestimmen. Hierbei ist jedoch keines der beiden Modelle für alle drei Datensätze die beste Wahl. Für den *apc_kreuzstöße* Datensatz schnitt Pointnet++ mit einem R2-Score von 0.99 am besten ab. Auch für *apc_stumpfstöße* konnte das Modell gute Vorhersagen treffen mit einem R2-Score von 0.96 und liegt damit nur leicht hinter den Methoden von Anthes und Kiyak, welche mit 0.99 das beste Ergebnis erreichten. Im Fall von *weld_scans* konnte 2DLaserNet die beste Leistung mit einem R2-Score von 0.70 erreichen. Wie in den einigen anderen Experimenten dieses Kapitels bereits ersichtlich wurde, scheint 2DLaserNet besser für die Anwendung auf den *weld_scans* Datensatz geeignet zu sein als Pointnet++. Im Vergleich zu den beiden neuronalen Netzen konnten die Näherungsformeln keine sinnvollen Kerbformzahlen bestimmen und zeigten alle negative R2-Scores auf im Bereich von -2.27 bis -10.03.

Eine mögliche Erklärung für die niedrigere Leistung der Näherungsformeln könnte daher

6 Ergebnisse und Diskussion

röhren, dass die Geometrieparameter, welche als Eingabe genutzt wurden, nicht genau bekannt sind und auf den Ergebnissen der Curvature-Methode basieren. Diese unterliegt jedoch großen Streuungen in Abhängigkeit des Messsystems was die Bestimmung der Parameter angeht (siehe 3.1). Hinzu kommt, dass der Nahtübergangsradius nicht immer eindeutig zuordenbar ist, was zu fehlerhaften Bestimmungen führen und damit die geringere Leistung der Näherungsformeln erklären könnte. Für die Methode von Rainer könnte die niedrigere Leistung für *apc_stumpfstöße* damit erklärt werden, dass seine Formel nur für Stumpfstöße mit Nahtanstiegswinkel von 45° entwickelt wurde, was zugunsten des Vergleichs außer Acht gelassen wurde. Kiyaks Methode weist einen hohen R2-Score von 0.99 für den *apc_stumpfstöße* Datensatz auf, während dieser bei *apc_kreuzstöße* geringer ist und 0.87 beträgt. Die Näherungsformel von Kiyak wurde ursprünglich für Stumpfstöße entwickelt. Die Autoren geben jedoch an, dass Experimente gezeigt haben, dass unter Berücksichtigung des Anwendungsbereichs die Näherungsformel ebenfalls für Kreuzstöße verwendet werden kann. Das Ergebnis dieses Experiments könnte Schwächen in dieser Aussage andeuten.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurde das Ziel verfolgt, neue Methoden zur direkten Bestimmung der Kerbformzahlen von Schweißverbindungen aus Oberflächenscans zu entwickeln, indem künstliche Neuronale Netze (KNN) eingesetzt werden. Schweißverbindungen spielen in vielen Industriebereichen eine wichtige Rolle. Dementsprechend ist die Kerbformzahl eine bedeutsame Größe, da sie die Kerbwirkung und die dadurch entstehenden Spannungskonzentrationen quantifiziert und hilft potenzielle Schwachstellen zu erkennen und das Versagensrisiko zu minimieren.

Im Rahmen dieser Arbeit wurden die künstlichen neuronalen Netze 2DLaserNet und Pointnet++ als vielversprechende Methoden an den Regressionsfall angepasst und verwendet, um Oberflächenscans von Schweißnähten direkt zu nutzen und die Kerbformzahl zu berechnen. Im Vergleich zu bestehenden Methoden zur Bestimmung der Kerbformzahl, welche auf Geometrieparametern oder FE-Simulationen basieren, ermöglicht dieser Ansatz eine detailliertere Repräsentation der Schweißnahtgeometrie und spart Ressourcen, indem auf die Bestimmung von Geometrieparametern oder die Modellierung von FE-Modellen verzichtet wird.

Die neuronalen Netze wurden auf drei verschiedenen Datensätzen, bestehend aus Scans von realen Kreuzstößen und künstlich erzeugten Daten von Stumpf- und Kreuzstößen, ausgewertet und verschiedenen Experimenten unterzogen. Die Ergebnisse der Untersuchungen zeigten vielversprechende Leistungen der Modelle für die Vorhersage der Kerbformzahlen. Im Vergleich mit anderen Methoden, welche auf Näherungsformeln basieren, zeigte Pointnet++ äquivalente oder bessere Leistungen auf den künstlich erzeugten Datensätzen *apc_kreuzstöße* und *apc_stumpfstöße* mit R2-Scores von 0.99 und 0.96. Für den Datensatz von realen Schweißnähten konnte 2DLaserNet die beste Leistung vorzeigen mit einem R2-Score von 0.70. Dies deutet das Potenzial der Modelle zur Bestimmung der Kerbformzahl an. Die begrenzte Anzahl von 202 Datenpunkten an realen Scans bietet Potenzial für

7 Zusammenfassung und Ausblick

weitere Verbesserungen, da die Datenmenge eine entscheidende Rolle für die Leistung von künstlichen neuronalen Netzen darstellt.

Weitere Untersuchungen zeigten, dass 2DLaserNet empfindlich gegenüber dem Punktabstand der Daten reagiert, während Pointnet++ robuster bleibt und trotz variierender Auflösung der Punkte gute Leistungen aufweist. Die Wirksamkeit von Datenaugmentierungen wurde ebenfalls untersucht, konnte jedoch nur in wenigen Fällen zu kleinen Verbesserungen der Generalisierungsfähigkeit der Modelle führen. Des Weiteren wurde untersucht, inwiefern die Modelle durch das Training mit dem *apc_kreuzstöße* Datensatz und der anschließenden Evaluation auf *weld_scans* in der Lage sind, von künstlichen auf reale Daten zu übertragen. Die Ergebnisse lassen darauf hindeuten, dass beiden Modellen kein guter Transfer von künstlichen auf reale Daten gelingt, wie durch die R2-Scores von -0.46 für 2DLaserNet und 0.00 für Pointnet++ ersichtlich wird. Zusätzlich konnten auch Datenaugmentierungen die Transferleistung der Modelle nicht verbessern.

7.1 Ausblick

Ein Aspekt, welcher im Rahmen dieser Arbeit beachtet werden muss ist, dass die Bestimmung der Kerbformzahl als Regressionsaufgabe behandelt wurde. Hierbei wurden jedoch ursprünglich zur Klassifizierung gedachte Modelle modifiziert und verwendet, welche nicht optimal auf die Regression angepasst sind. Eine Optimierung der Netzwerkarchitektur zum Beispiel durch Anpassen der verschiedenen Schichten, Aktivierungsfunktionen etc. könnte zu besseren Leistungen führen. Hinzu kommt, dass die Qualität der Vorhersagen von der Genauigkeit der realen Oberflächenscans abhängt. Die verwendeten Daten müssen die realen Schweißnähte gut genug repräsentieren, um wirksame Ergebnisse zu erzielen. Eventuell könnten Verbesserungen in der Datenerfassung die Modellleistung weiter steigern.

Außerdem könnten weiterführende Anpassungen in Bezug auf den Datengenerierungsprozess für die künstlichen Daten erfolgen um die Transferfähigkeit der Modelle auf reale Daten verbessern.

Weiteres Potenzial besteht darin, dass die verwendeten Modelle zwar Vorhersagen für die Kerbformzahlen treffen, aber die Unsicherheit der Vorhersagen nicht betrachtet wird. Die Ermittlung der Verlässlichkeit der Ausgaben könnte wertvolle Informationen liefern, insbesondere für sicherheitskritische Anwendungen.

7 Zusammenfassung und Ausblick

Die vorliegende Arbeit behandelt nur 2D-Oberflächenscans der Schweißnähte. Interessant wäre, wie die Ergebnisse des Pointnet++ Modells, welches für 3D-Punktwolken konzipiert wurde, unter Nutzung von 3D-Scandaten aussehen würden. Die Erweiterung auf 3D-Punktwolken und die Einbeziehung von Nachbarschaftsverhältnissen zwischen den Punkten könnte zu einer besseren Erfassung von räumlichen Strukturen und Mustern führen.

Des Weiteren, wurden in dieser Arbeit ausschließlich Stumpfstöße und Kreuzstöße behandelt. Zukünftige Arbeiten könnten die Anwendung der vorgestellten Methoden auf andere Formen von Schweißverbindungen umfassen um die Allgemeingültigkeit und Anwendbarkeit der Modelle zu untersuchen.

Darüber hinaus könnten die genutzten Modelle nicht nur zur Kerbformzahlbestimmung, sondern auch zur Bestimmung von Geometrieparametern oder anderen Eigenschaften von Schweißnähten verwendet werden.

A Anhang

A.1 Näherungsformeln

Die vorliegenden Formeln stammen aus den Arbeiten von Anthes et al. [12], Rainer [41] und Kiyak et al. [28]. Hierbei sind nur die Formeln und Anwendungsbereiche für den Belastungsfall der Biegung für den Nahtübergang aufgelistet.

A.1.1 Anthes

Kreuzstoß

Formel:

$$K_t = 1.256 + \left[1 + 0.023 * (y * \sin(\alpha))^{-3.09} * z^{2.412} + 2.153 * \left(\frac{t}{r} \right)^{0.154} - 3.738 * \sin(\alpha)^{0.481} \right] * \sin(\alpha)^{1.723} * \left(\frac{t}{r} \right)^{0.172}$$

Anwendungsbereich:

- $r: 4 \leq \frac{t}{r} \leq 200$

- $\alpha: 15 - 85^\circ$

A Anhang

Stumpfstoß

Formel:

$$K_t = (1 - 0.156 * \left(\frac{t}{r}\right)^{0.207}) * (1 + (0.181 + 1.207 * \sin(\alpha) - 1.737 * \sin(\alpha)^2 + 0.689 * \sin(\alpha)^3) * \frac{t^{0.2919+0.3491*\sin(\alpha+3.283)}}{r})$$

Anwendungsbereich:

- $r: 0 \leq \frac{t}{r} \leq 200$

- $\alpha: 0 - 90^\circ$

A.1.2 Rainer

Kreuzstoß

Formel:

$$K_t = 1 + \left\{ \frac{0.4}{(\frac{y*t*\sin(\alpha)}{\sqrt{2}*r})^{0.66}} + 3.8 \left(\sqrt{\left(\frac{2r}{t}\right)^3 + \frac{2r}{t}} \right)^{2.25} + 0.2 * \left(\frac{\frac{t}{2r}}{\left(\frac{t}{2r} + \frac{y*t*\sin(\alpha)}{\sqrt{2}p}\right) * (\frac{y*t*\sin(\alpha)}{\sqrt{2}p})^{1.33}} \right)^{-0.5} \right\}$$

Anwendungsbereich:

- $r: 0 \leq \frac{t}{r} \leq 400$

- $\alpha: 45^\circ$

Stumpfstoß

Formel:

$$K_t = 1 + \left(\frac{0.4}{(\frac{u}{r})^{0.66}} + 3.8 * \left(\frac{1 + \frac{t}{2r}}{\frac{t}{2r} * \sqrt{\frac{t}{2r}}} \right)^{2.25} + 0.2 * \frac{\frac{t}{2r}}{(\frac{t}{2r} + \frac{u}{r}) * (\frac{u}{r})^{1.33}} \right)^{-0.5}$$

Anwendungsbereich:

- $r: 0 \leq \frac{t}{r} \leq 400$
- $\alpha: 45^\circ$

A.1.3 Kiyak

Formel:

$$K_t = 1 + 1.1399 * \left(\frac{h}{t} \right)^{0.2062 * \alpha} * \alpha^{1.067} * e^{-1.6775 * \alpha} * \left(\frac{r}{t} \right)^{-0.295 * \alpha} * \left(0.021 + \frac{r}{t} \right)^{-0.4711}$$

Anwendungsbereich:

- $r: 0.1 \leq r \leq 4$
- $\alpha: 10 \leq \alpha \leq 60$
- $L = 14.6 \text{ mm}$

A.2 Generierung Punktfolge Stumpfstoß

Die Erzeugung des Nahtverlaufs geschieht in drei Abschnitten (siehe Abbildung A.1). Abschnitt 1 beinhaltet alle Punkte von der Symmetriearchse bis zum Wendepunkt des Kurvenverlaufs, welcher dem Punkt P2 entspricht. Abschnitt 2 enthält die Punkte vom Wendepunkt bis

A Anhang

zum Beginn des Nahtübergangs, also den Abschnitt zwischen Punkt P2 und P1. Abschnitt 3 repräsentiert alle Punkte auf der Blechoberfläche mit einer y-Koordinate von 0.

Die Nahtüberhöhung h wird anhand der anderen Parameter folgendermaßen ermittelt:

$$h = \left(\frac{w}{2\sin(\alpha)} + \frac{r * \tan(\frac{\alpha}{2})}{\sin(\alpha)} \right) (1 - \cos(\alpha)) \quad (\text{A.1})$$

Die Punkte $P1, P2, Pr, Pr2$ sind wie folgt definiert:

$$P1 = \left[\frac{h * \sin(\alpha)}{1 - \cos(\alpha)}, 0 \right] \quad (\text{A.2})$$

$$P2 = \left[\left(\frac{h}{1 - \cos(\alpha)} - r \right) * \sin(\alpha), r * (1 - \cos(\alpha)) \right] \quad (\text{A.3})$$

$$Pr = \left[\frac{h * \sin(\alpha)}{1 - \cos(\alpha)}, r \right] \quad (\text{A.4})$$

$$Pr2 = \left[0, \frac{-h}{1 - \cos(\alpha)} + r + h \right] \quad (\text{A.5})$$

Die Berechnung der y-Koordinate für einen Punkt i der Punktwolke erfolgt folgendermaßen:

$$y_i(x_i, \alpha, h, r) = \begin{cases} \sqrt{\left(\frac{h}{1 - \cos(\alpha)} - r \right)^2 - (x_i - Pr2_x)^2} + Pr2_y & , \text{ für } 0 \leq x_i < P2_x \\ -\sqrt{r^2 - (x_i - Pr_x)^2} + Pr_y & , \text{ für } P2_x \leq x < P1_x \\ 0 & , \text{ für } P1_x \leq x \end{cases} \quad (\text{A.6})$$

wobei α den Nahtanstiegswinkel, h die Nahtüberhöhung und r den Nahtübergangsradius darstellt.

A Anhang

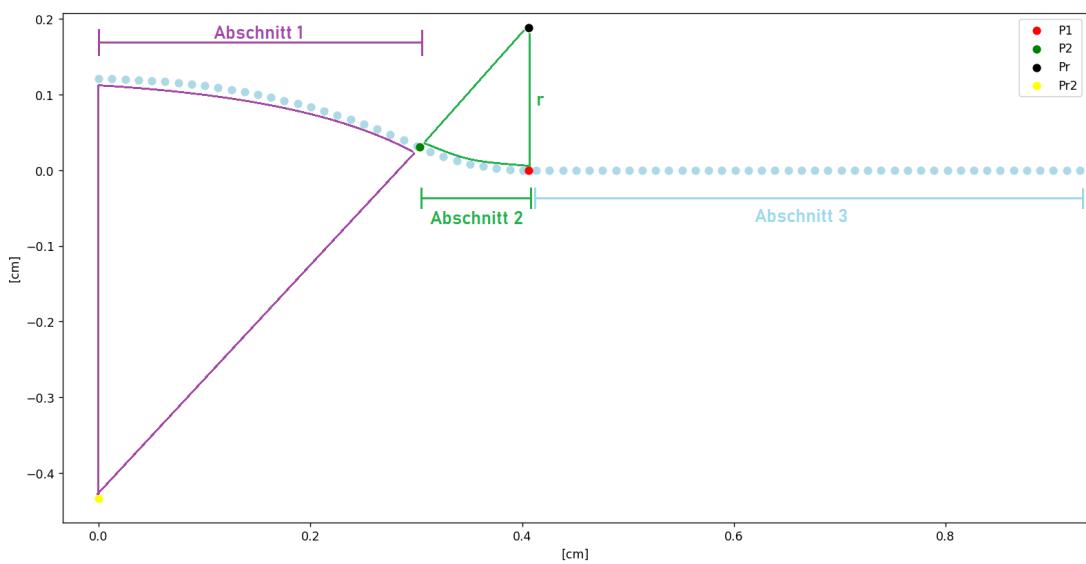


Abbildung A.1: Erzeugung der Punktfolge für einen Stumpfstoß in drei Abschnitten

A.3 Ergebnisse mit R2-Loss

A Anhang

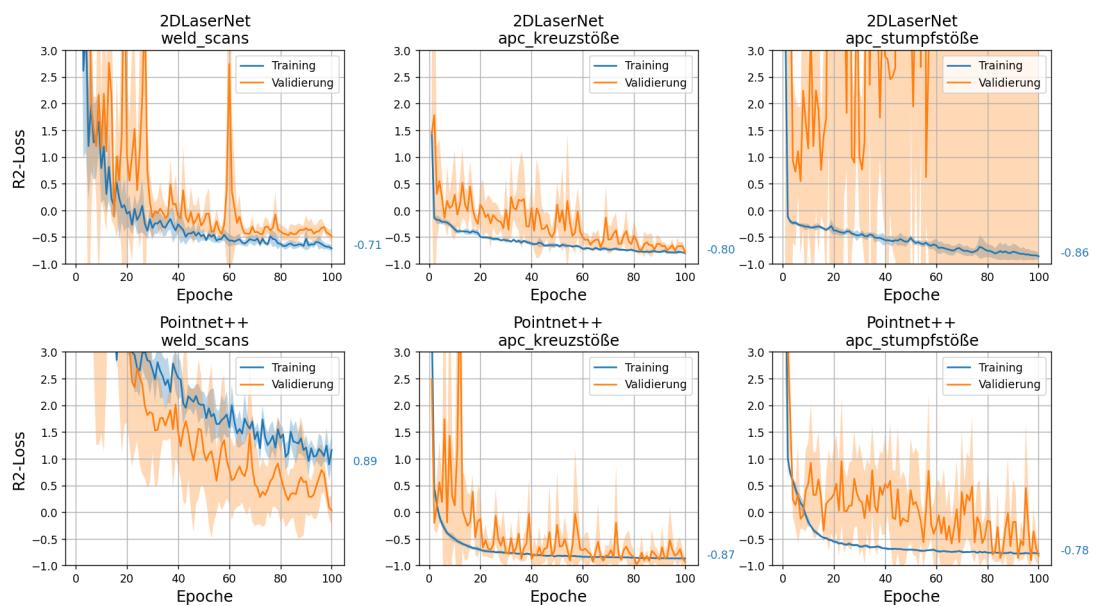


Abbildung A.2: Kurvenverlauf des R2-Loss von Pointnet++ und 2DLaserNet. Die durchgezogene Linie repräsentiert den Durchschnitt über 5 *Folds*, wobei die Standardabweichung um den Durchschnitt herum dargestellt ist

A Anhang

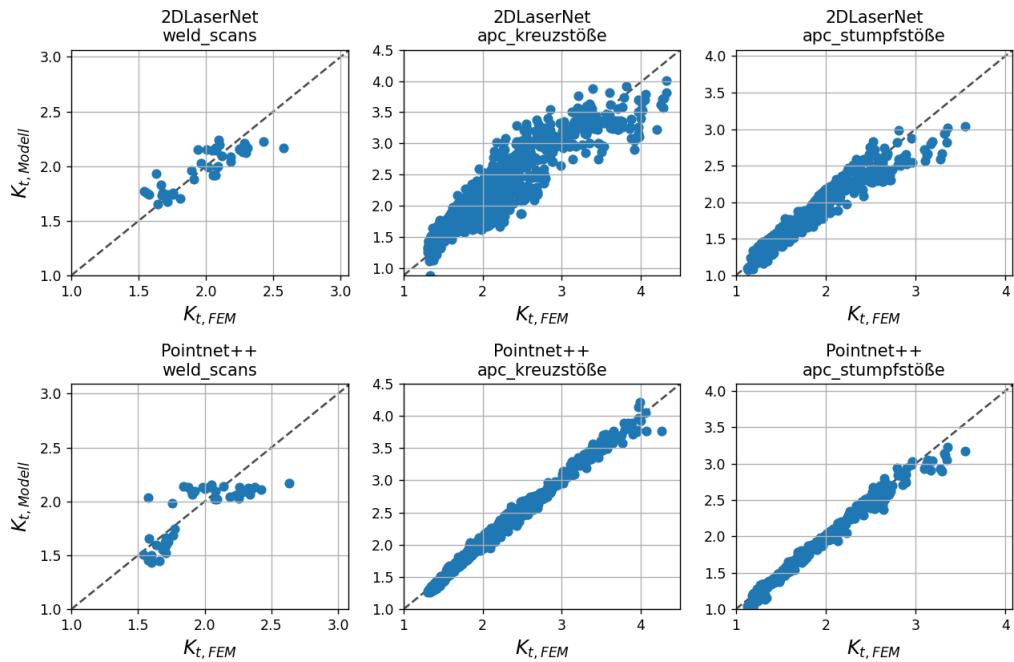


Abbildung A.3: Die vorhergesagten Kerbformzahlen des besten Folds der Pointnet++ und 2DLaserNet Modelle (mit R2-Loss trainiert) im Vergleich mit den Kerbformzahlen der entsprechenden FE-Modelle. Datenpunkte stammen aus dem Testset

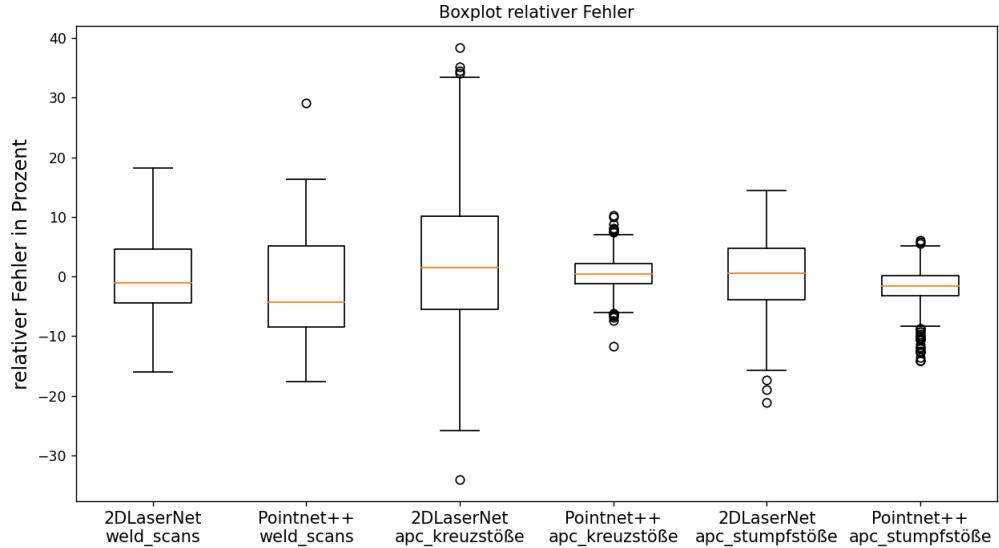


Abbildung A.4: Boxplot des relativen Fehlers des besten Folds der Pointnet++ und 2DLaserNet Modelle (mit R2-Loss trainiert)

A.4 Ergebnisse mit Augmentierungen

A Anhang

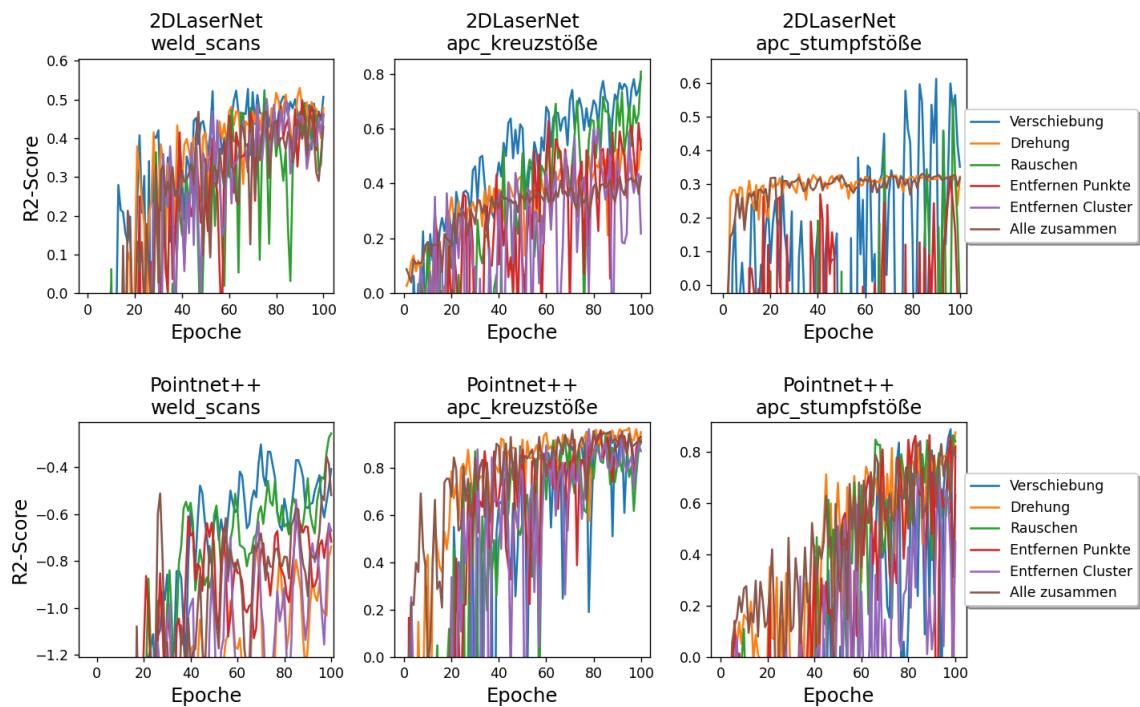


Abbildung A.5: Kurvenverlauf des R2-Scores für den Validierungsdatensatz

B Nomenklatur

FE Finite-Elemente

KNN Künstliches Neuronales Netz

ML Machine Learning

DL Deep Learning

MAE Mean Absolute Error

CNN Convolutional Neural Network

CNNs Convolutional Neural Networks

Literaturverzeichnis

- [1] *surfaceCONTROL 3D.* – <https://www.micro-epsilon.de/download/products/dat--surfaceCONTROL-3D--de.pdf>, aufgerufen am 18.06.2023
- [2] Schweißen und verwandte Prozesse - Arten der Schweißnahtvorbereitung - Teil 1: Lichtbogenhandschweißen, Schutzgasschweißen, Gasschweißen, WIG-Schweißen und Strahlschweißen von Stählen (ISO_9692-1:2013) Deutsche Fassung EN_ISO_9692-1:2013_2012. (2012), Dec. <http://dx.doi.org/10.31030/2067108>. – DOI 10.31030/2067108
- [3] Schweißen- Schmelzschweißverbindungen an Stahl, Nickel, Titan und deren Legierungen (ohne Strahlschweißen) - Bewertungsgruppen von Unregelmäßigkeiten (ISO 5817:2014) Deutsche Fassung EN ISO 5817:2014_06. (2014). ISBN OCLC:935527817
- [4] *Convolutional Neural Networks (CNNs).* 2023. – <https://anhreynolds.com/blogs/cnn.html>, aufgerufen am 22.07.2023
- [5] *Download Python.* 2023. – <https://www.python.org/downloads/>, aufgerufen am 10.07.2023
- [6] *Maschinenelemente 1 Einflussfaktoren einer Schweißnahtverbindung.* 2023. – <https://www.ingenieurkurse.de/maschinenelemente-1/verbindungen-und-verbindungselemente/stoffschluessige-verbindungen/schweissverbindungen/einflussfaktoren-einer-schweissnahtverbindung.html>, aufgerufen am 21.07.2023
- [7] *The MLOps stack component for experiment tracking.* 2023. – <https://neptune.ai/>, aufgerufen am 10.07.2023

Literaturverzeichnis

- [8] Overview of a Neural Network's Learning Process. 2023. – <https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa>, aufgerufen am 21.07.2023
- [9] PyTorch. 2023. – <https://pytorch.org/>, aufgerufen am 10.07.2023
- [10] scikit-learn Machine Learning in Python. 2023. – <https://scikit-learn.org/stable/index.html>, aufgerufen am 10.07.2023
- [11] ANGUITA, Davide ; GHELARDONI, Luca ; GHIO, Alessandro ; ONETO, Luca ; RIDELLA, Sandro: *European Symposium on Artificial Neural Networks*. I6doc.com Publ., 2012. – 441–446 S. – ISBN 978–2–87419–049–0
- [12] ANTHES, RJ ; KÖTTGEN, VB ; SEEGER, T: Kerbformzahlen von Stumpfstößen und Doppel-T-Stößen. In: *Schweißen und Schneiden* 45 (1993), Nr. 12, S. 685–688
- [13] BERRAR, Daniel: *Cross-Validation*. 2018. <http://dx.doi.org/10.1016/B978-0-12-809633-8.20349-X>. – ISBN 9780128096338
- [14] BISHOP, Christopher M.: *Pattern Recognition and Machine Learning*. Springer Verlag, 2006. – ISBN 0387310738
- [15] BRAUN, Moritz ; KELLNER, Leon: Comparison of machine learning and stress concentration factors-based fatigue failure prediction in small-scale butt-welded joints. In: *Fatigue & Fracture of Engineering Materials & Structures* 45 (2022), 08. <http://dx.doi.org/10.1111/ffe.13800>. – DOI 10.1111/ffe.13800
- [16] BRAUN, Moritz ; NEUHÄUSLER, Josef ; DENK, Martin ; RENKEN, Finn ; KELLNER, Leon ; SCHUBNELL, Jan ; JUNG, Matthias ; ROTHER, Klemens ; EHLERS, Sören: Statistical Characterization of Stress Concentrations along Butt Joint Weld Seams Using Deep Neural Networks. In: *Applied Sciences* 12 (2022), Jun, Nr. 12, S. 6089. <http://dx.doi.org/10.3390/app12126089>. – DOI 10.3390/app12126089
- [17] DABIRI, Edris ; GHAFOURI, Mehran ; ROHANI RAFTAR, Hamidreza ; BJÖRK, Timo: Utilizing artificial neural networks for stress concentration factor calculation in butt welds.

Literaturverzeichnis

- In: *Journal of Constructional Steel Research* 138 (2017), 10, S. 488–498. <http://dx.doi.org/10.1016/j.jcsr.2017.08.009> – DOI 10.1016/j.jcsr.2017.08.009
- [18] DASSAULT SYSTÈMES SIMULIA CORP.: *Abaqus/Standard 2021*. Johnston, RI, 2021
- [19] Norm DIN 18800:2008-11 . *Stahlbauten - Teil 1: Bemessung und Konstruktion*
- [20] GHAHREMANI, Kasra ; SAFA, Mahdi ; YEUNG, Jamie ; WALBRIDGE, Scott ; HAAS, Carl ; DUBOIS, Sébastien: Quality assurance for high-frequency mechanical impact (HFMI) treatment of welds using handheld 3D laser scanning technology. In: *Welding in the World* 59 (2014), 12. <http://dx.doi.org/10.1007/s40194-014-0210-3> – DOI 10.1007/s40194-014-0210-3
- [21] GOODFELLOW, Ian ; BENGIO, Yoshua ; COURVILLE, Aaron: *Deep Learning*. MIT Press, 2016. – <http://www.deeplearningbook.org>
- [22] HAMMERSBERG, Peter ; OLSSON, Hasse ; HAMMERSBERG, Hasse: Statistical evaluation of welding quality in production, 2010. – ISBN 978-91-7415-608-9
- [23] HERNÁNDEZ-GARCÍA, Alex ; KÖNIG, Peter: Data augmentation instead of explicit regularization. In: *CoRR* abs/1806.03852 (2018). <http://arxiv.org/abs/1806.03852>
- [24] HOBBACHER, A. F.: *Recommendations for Fatigue Design of Welded Joints and Components*. Springer, 2015. – ISBN 9783319237572
- [25] Hou, C: Fatigue analysis of welded joints with the aid of real three-dimensional weld toe geometry. In: *International Journal of Fatigue* 29 (2007), Apr, Nr. 4, S. 772–785. <http://dx.doi.org/10.1016/j.ijfatigue.2006.06.007> – DOI 10.1016/j.ijfatigue.2006.06.007
- [26] JUNG, Matthias: *Entwicklung und Implementierung eines Algorithmus zur Approximation und Bewertung von Kerbfaktoren an Kehlnähten auf Basis berührungsloser 3D-Vermessung*, Karlsruher Institut für Technologie, Masterarbeit, 2018
- [27] KALECI, Burak ; TURGUT, Kaya ; DUTAGACI, Helin: 2DLaserNet: A deep learning architecture on 2D laser scans for semantic classification of mobile robot locations. In: *Engi-*

Literaturverzeichnis

- neering Science and Technology, an International Journal* 28 (2022), 101027. <http://dx.doi.org/https://doi.org/10.1016/j.jestch.2021.06.007>. – DOI <https://doi.org/10.1016/j.jestch.2021.06.007>. – ISSN 2215–0986
- [28] KIYAK, Yusuf ; MADIA, Mauro ; ZERBST, Uwe: Extended parametric equations for weld toe stress concentration factors and through-thickness stress distributions in butt-welded plates subject to tensile and bending loading. In: *Welding in the World* 60 (2016), Sep, Nr. 6, S. 1247–1259. <http://dx.doi.org/10.1007/s40194-016-0377-x>. – DOI 10.1007/s40194-016-0377-x
- [29] LANQUILLON, Carsten: *Grundzüge des maschinellen Lernens*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2019. – 89–142 S. http://dx.doi.org/10.1007/978-3-662-60408-3_3
- [30] LARSEN, Mikkel L. ; ARORA, Vikas ; LÜTZEN, Marie ; PEDERSEN, Ronnie R. ; PUTNAM, Eric: Fatigue life estimation of the weld joint in K-node of the offshore jacket structure using stochastic finite element analysis. In: *Marine Structures* 78 (2021), 103020. <http://dx.doi.org/https://doi.org/10.1016/j.marstruc.2021.103020>. – DOI <https://doi.org/10.1016/j.marstruc.2021.103020>. – ISSN 0951–8339
- [31] LEE, Chanh H.: *Numerische Untersuchung der Genauigkeit von Approximationsformeln für Kerbformzahlen an der realen Schweißnahtgeometrie von Quersteifen*, Karlsruher Institut für Technologie, Bachelorarbeit, 2018
- [32] LEE, Chin-Hyung ; CHANG, Kyong-Ho ; JANG, Gab-Chul ; LEE, Chan-Young: Effect of weld geometry on the fatigue life of non-load-carrying fillet welded cruciform joints. In: *Engineering Failure Analysis* 16 (2009), Nr. 3, 849-855. <http://dx.doi.org/https://doi.org/10.1016/j.engfailanal.2008.07.004>. – DOI <https://doi.org/10.1016/j.engfailanal.2008.07.004>. – ISSN 1350–6307
- [33] LIEURADE, H. ; HUTHER, Isabel ; LEFEBVRE, Fabien: Effect of Weld Quality and Postweld Improvement Techniques on the Fatigue Resistance of Extra High Strength Steels. In: *Welding in the World* 52 (2008), 07. <http://dx.doi.org/10.1007/BF03266658>. – DOI 10.1007/BF03266658

Literaturverzeichnis

- [34] LINDGREN, Eric ; STENBERG, Thomas: *Quality Inspection and Fatigue Assessment of Welded Structures*, KTH Royal Institute of Technology, Masterarbeit, 2011
- [35] OSWALD, Markus ; MAYR, Christina ; ROTHER, Klemens: Determination of notch factors for welded cruciform joints based on numerical analysis and metamodeling. In: *Welding in the World* 63 (2019), Jun, Nr. 5, S. 1339–1354. <http://dx.doi.org/10.1007/s40194-019-00751-y>. – DOI 10.1007/s40194-019-00751-y
- [36] OSWALD, Markus ; NEUHÄUSLER, Josef ; ROTHER, Klemens: Determination of notch factors for welded butt joints based on numerical analysis and metamodeling. In: *Welding in the World* 64 (2020), Sep, Nr. 12, S. 2053–2074. <http://dx.doi.org/10.1007/s40194-020-00982-4>. – DOI 10.1007/s40194-020-00982-4
- [37] OSWALD, Markus ; NEUHÄUSLER, Josef ; ROTHER, Klemens: *Stress Concentration Factor Calculation DY-Butt Joint - Full Penetration*. 2020. – <http://rother.userweb.mwn.de/Page6/Page6.html>, aufgerufen am 18.06.2023
- [38] OTTERSBÖCK, M.J. ; LEITNER, M. ; STOSCHKA, M.: Characterisation of actual weld geometry and stress concentration of butt welds exhibiting local undercuts. In: *Engineering Structures* 240 (2021), 112266. <http://dx.doi.org/https://doi.org/10.1016/j.engstruct.2021.112266>. – DOI <https://doi.org/10.1016/j.engstruct.2021.112266>. – ISSN 0141–0296
- [39] QI, Charles R. ; YI, Li ; SU, Hao ; GUIBAS, Leonidas J.: *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. 2017
- [40] QI, Charles R. ; SU, Hao ; MO, Kaichun ; GUIBAS, Leonidas J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: *CoRR* abs/1612.00593 (2016). <http://arxiv.org/abs/1612.00593>
- [41] RAINER, G: *Berechnung von Spannungen in Schweißverbindungen mit der Methode der Finiten Elemente*, Technische Universität Darmstadt, Diss., 1978
- [42] RENKEN, Finn ; VON BOCK UND POLACH, Rüdiger Ulrich F. ; SCHUBNELL, Jan ; JUNG, Matthias ; OSWALD, Markus ; ROTHER, Klemens ; EHLERS, Sören ; BRAUN, Moritz: An algorithm for statistical evaluation of weld toe geometries using laser tri-

Literaturverzeichnis

- angulation. In: *International Journal of Fatigue* 149 (2021), 106293. <http://dx.doi.org/https://doi.org/10.1016/j.ijfatigue.2021.106293>. – DOI <https://doi.org/10.1016/j.ijfatigue.2021.106293>. – ISSN 0142–1123
- [43] RENNERT, Roland: *Rechnerischer Festigkeitsnachweis: für Maschinenbauteile aus Stahl, Eisenguss- und Aluminiumwerkstoffen*. 6. Frankfurt am Main : VDMA Verlag, 2012. – ISBN 3816306055
- [44] RENNERT, Roland ; MASCHINENBAU, Frankfurt F.: *Rechnerischer Festigkeitsnachweis: für Maschinenbauteile aus Stahl, Eisenguss- und Aluminiumwerkstoffen*. VDMA Verlag, 2012. – ISBN 3816306055
- [45] RÖSLER, Joachim ; HARDERS, Harald ; BÄKER, Martin: *Mechanisches Verhalten der Werkstoffe*. 4. Springer Vieweg Wiesbaden, 2012. – ISBN 9783834822413
- [46] SCHORK, B. ; KUCHARCZYK, P. ; MADIA, M. ; ZERBST, U. ; HENSEL, J. ; BERNHARD, J. ; TCHUINDJANG, D. ; KAFFENBERGER, M. ; OECHSNER, M.: The effect of the local and global weld geometry as well as material defects on crack initiation and fatigue strength. In: *Engineering Fracture Mechanics* 198 (2018), Jul, S. 103–122. <http://dx.doi.org/10.1016/j.engfracmech.2017.07.001>. – DOI 10.1016/j.engfracmech.2017.07.001
- [47] SCHORK, B. ; ZERBST, Uwe ; KIYAK, Y. ; KAFFENBERGER, M. ; MADIA, M. ; OECHSNER, M.: Effect of the parameters of weld toe geometry on the FAT class as obtained by means of fracture mechanics-based simulations. In: *Welding in the World* 64 (2020), May, Nr. 6, S. 925–936. <http://dx.doi.org/10.1007/s40194-020-00874-7>. – DOI 10.1007/s40194-020-00874-7
- [48] SCHUBNELL, Jan ; AYDOGAN, Öner ; JUNG, Matthias: Bestimmen von Spannungskonzentrationen an Schweißverbindungen aus Oberflächenscans durch künstliche neuronale Netze. In: *DVS – Deutscher Verband für Schweißen und verwandte Verfahren* (2023)
- [49] SCHUBNELL, Jan ; JUNG, Matthias ; LE, Chanh H. ; FARAJIAN, Majid ; BRAUN, Moritz ; EHLERS, Sören ; FRICKE, Wolfgang ; GARCIA, Martin ; NUSSBAUMER, Alain ; BAUMGARTNER, Jörg: Influence of the optical measurement technique and eva-

Literaturverzeichnis

- luation approach on the determination of local weld geometry parameters for different weld types. In: *Welding in the World* 64 (2020), Dec, Nr. 2, S. 301–316. <http://dx.doi.org/10.1007/s40194-019-00830-0>. – DOI 10.1007/s40194-019-00830-0
- [50] SCHÄFER, Jan ; SCHMITT, Philipp ; HLAWITSCHKA, Mark ; BART, Hans-Jörg: Measuring Particle Size Distributions in Multiphase Flows Using a Convolutional Neural Network. In: *Chemie Ingenieur Technik* 91 (2019), 09. <http://dx.doi.org/10.1002/cite.201900099>. – DOI 10.1002/cite.201900099
- [51] SCHÜRMANN, K ; SCHAUMANN, P ; PITTLER, A ; RETHMEIER, M: Experimental investigations on the fatigue resistance of automatically welded tubular X-joints for jacket support structures. In: *Journal of Physics: Conference Series* 1669 (2020), oct, Nr. 1, 012022. <http://dx.doi.org/10.1088/1742-6596/1669/1/012022>. – DOI 10.1088/1742-6596/1669/1/012022
- [52] SESHADRI, Arjun: *Statistical variation of weld profiles and their expected influence on fatigue strength*, Lappeenranta University of Technology, Masterarbeit, 2006. <http://urn.fi/URN:NBN:fi-fe20061351>
- [53] SHORTEN, Connor ; KHOSHGOFTAAR, Taghi M.: A survey on Image Data Augmentation for Deep Learning. In: *Journal of Big Data* 6 (2019), Jul, Nr. 1. <http://dx.doi.org/10.1186/s40537-019-0197-0>. – DOI 10.1186/s40537-019-0197-0
- [54] STENBERG, Thomas ; LINDGREN, Eric ; BARSOUM, Zuheir: Development of an algorithm for quality inspection of welded structures. In: *Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture* 226 (2012), 06, S. 1033–1041. <http://dx.doi.org/10.1177/0954405412439138>. – DOI 10.1177/0954405412439138
- [55] YAMASHITA, Rikiya ; NISHIO, Mizuho ; DO, Richard Kin G. ; TOGASHI, Kaori: Convolutional neural networks: an overview and application in radiology. In: *Insights into Imaging* 9 (2018), Jun, Nr. 4, S. 611–629. <http://dx.doi.org/10.1007/s13244-018-0639-9>. – DOI 10.1007/s13244-018-0639-9

Literaturverzeichnis

- [56] YAN, Xu: *Pointnet/Pointnet++ Pytorch*. 2023. – https://github.com/yanx27/Pointnet_Pointnet2_pytorch, aufgerufen am 10.07.2023
- [57] ZERBST, U. ; AINSWORTH, R.A. ; BEIER, H.Th. ; PISARSKI, H. ; ZHANG, Z.L. ; NIK-BIN, K. ; NITSCHKE-PAGEL, T. ; MÜNSTERMANN, S. ; KUCHARCZYK, P. ; KLINGBEIL, D.: Review on fracture and crack propagation in weldments – A fracture mechanics perspective. In: *Engineering Fracture Mechanics* 132 (2014), 200-276. <http://dx.doi.org/https://doi.org/10.1016/j.engfracmech.2014.05.012>. – DOI <https://doi.org/10.1016/j.engfracmech.2014.05.012>. – ISSN 0013–7944

Abbildungsverzeichnis

2.1 a) Kraftfluss in einem gekerbten Bauteilen bei Zugbelastung. Die Dichte der Kraftflusslinien ist ein Maß für die Höhe der Spannung. Im Kerbgrund ergibt sich eine Spannungskonzentration [45]	
b) Verteilung der Längsspannung bei Zugbelastung [45]	4
2.2 Geometrieparameter einer Kehlnaht	5
2.3 Beispielhaftes Training und Aufbau eines KNN mit Vorwärts- und Rückwärts-propagierung [8]	7
2.4 Faltungsoperation und Max pooling am Beispiel eines 2D-Inputs und einem Kernel der Größe 3×3 [50]	9
2.5 Beispielhafte Darstellung des typischen Verhältnis zwischen Modellkapazität und Fehler [21]	10
3.1 Mehrdeutigkeit von Nahtübergangsgraden einer Schweißverbindung [34] . . .	12
3.2 Einfluss des Punktabstandes auf die Bestimmung der Geometrieparameter [57]	13
3.3 Krümmungsverlauf am Beispiel einer idealisierten Schweißnaht [26]	14
3.4 Die drei Anwendungsarten von PointNet: Klassifizierung, Teilegmentierung, semantische Segmentierung [40]	17
3.5 PointNet Architektur [40]	18
3.6 PointNet++ Architektur [39]	19
3.7 Ein Beispiel für einen Laserscan und den dazugehörigen Punkten. Der schwarze Punkt auf dem Roboter stellt den Ursprung des Entfernungsmessers und der Pfeil die Ausrichtung des Roboters dar. [27]	20
3.8 2DLaserNet Architektur [27]	21
4.1 a) idealisiertes FE-Modell einer Schweißnaht b) FE-Modell basierend auf der realen Schweißnaht [26]	23

Abbildungsverzeichnis

4.2 (a) Vorgehensweise zur Bestimmung von Kerbformzahlen an Schweißverbindungen auf Basis gescannter 2D-Nahtprofile (b) Vorgehensweise zur Bestimmung von Kerbformzahlen auf Basis künstlicher 2D-Nahtprofile [48]	25
5.1 Beispiel eines 2D-Querschnitts einer realen Schweißnaht aus dem <i>weld_scans</i> Datensatz	27
5.2 Histogramm der Kerbformzahlen aus dem <i>weld_scans</i> Datensatz	28
5.3 Beispielhafter Ablauf zur Erzeugung von künstlichen Daten eines Stumpfstoßes	29
5.4 Schematische Darstellung des DV-Stumpfstoßes (In Anlehnung an: [37])	30
5.5 Ausschnitt einer künstlich erzeugten Punktwolke eines Stumpfstoßes	31
5.6 Schematische Darstellung der Kehlnaht (In Anlehnung an: [3])	33
5.7 a) Abaqus Modell des Kreuzstoßes mit dem vordefinierten <i>Set</i> (in rot) zur Erzeugung der Punktwolke b) die entsprechende erzeugte Punktwolke	34
5.8 a) Beschränkung der Nahtunterwölbung für Kreuzstoße [2]	
b) Beschränkung der Nahtüberhöhung für Kreuzstoße [2]	35
5.9 (a) FE-Modell des Stumpfstoßes unter Biegebelastung (b) FE-Modell des Kreuzstoßes unter Biegebelastung	37
5.10 Histogramm der Kerbformzahlen aus den FE-Simulationen für Stumpfstoße	37
5.11 Histogramm der Kerbformzahlen aus den FE-Simulationen für Kreuzstoße	38
5.12 Die berechneten Kerbformzahlen für die Stumpfstoße in Abhängigkeit der Geometrieparameter	38
5.13 Die berechneten Kerbformzahlen für die Kreuzstoße in Abhängigkeit der Geometrieparameter	39
5.14 Beispiel: Erste 10 Zeilen der Textdatei einer Punktwolke	41
5.15 Beispielhafte Erklärung des Kreuzvalidierungsverfahrens mit $k = 5$ Folds [29]	42
6.1 Kurvenverlauf des MAE-Loss der Pointnet++ und 2DLaserNet Baselines. Die durchgezogene Linie repräsentiert den Durchschnitt über 5 <i>Folds</i> , wobei die Standardabweichung um den Durchschnitt herum dargestellt ist	45
6.2 Boxplot des R2-Scores der Pointnet++ und 2DLaserNet Baselines (mit MAE-Loss trainiert)	46
6.3 Die vorhergesagten Kerbformzahlen des besten Folds der Pointnet++ und 2DLaserNet Baselines (mit MAE-Loss trainiert) im Vergleich mit den Kerbformzahlen der entsprechenden FE-Modelle . Datenpunkte stammen aus dem Testset	47

Abbildungsverzeichnis

6.4	Boxplot des relativen Fehlers des besten Folds der Pointnet++ und 2DLaser-Net Baselines (mit MAE-Loss trainiert)	48
6.5	Boxplot des R2-Scores von Pointnet++ und 2DLasernet (mit R2-Loss trainiert)	51
6.6	R2-Scores in Abhängigkeit des Punktabstandes für den <i>apc_stumpfstöße</i> Datensatz. Die Punkte stellen den Durchschnitt über alle 5 Folds dar mit entsprechender Standardabweichung.	54
6.7	Kurvenverlauf des R2-Scores für den Trainingsdatensatz	57
6.8	Kurvenverlauf des R2-Loss von Pointnet++ und 2DLaserNet trainiert mit <i>apc_kreuzstöße</i> und evaluiert auf <i>weld_scans</i>	59
6.9	Die vorhergesagten Kerbformzahlen von Pointnet++ und 2DLaserNet (trainiert auf <i>apc_kreuzstöße</i> & evaluiert auf <i>weld_scans</i>) im Vergleich mit den Kerbformzahlen der entsprechenden FE-Modelle	60
6.10	Boxxplot des relativen Fehlers von Pointnet++ und 2DLaserNet (trainiert auf <i>apc_kreuzstöße</i> & evaluiert auf <i>weld_scans</i>)	60
6.11	Die bestimmten Kerbformzahlen der verschiedenen Methoden im Vergleich für den <i>weld_scans</i> Datensatz	64
6.12	Die bestimmten Kerbformzahlen der verschiedenen Methoden im Vergleich für den <i>apc_kreuzstöße</i> Datensatz	65
6.13	Die bestimmten Kerbformzahlen der verschiedenen Methoden im Vergleich für den <i>apc_stumpfstöße</i> Datensatz	66
A.1	Erzeugung der Punktfolge für einen Stumpfstoß in drei Abschnitten	75
A.2	Kurvenverlauf des R2-Loss von Pointnet++ und 2DLaserNet. Die durchgezogene Linie repräsentiert den Durchschnitt über 5 <i>Folds</i> , wobei die Standardabweichung um den Durchschnitt herum dargestellt ist	76
A.3	Die vorhergesagten Kerbformzahlen des besten Folds der Pointnet++ und 2DLaserNet Modelle (mit R2-Loss trainiert) im Vergleich mit den Kerbformzahlen der entsprechenden FE-Modelle. Datenpunkte stammen aus dem Testset	77
A.4	Boxplot des relativen Fehlers des besten Folds der Pointnet++ und 2DLaser-Net Modelle (mit R2-Loss trainiert)	77
A.5	Kurvenverlauf des R2-Scores für den Validierungsdatensatz	79

Tabellenverzeichnis

5.1	Die verwendeten Datensätze mit deren Beschreibung und Anzahl an Samples	26
6.1	Pearson Korrelationskoeffizient zwischen den Geometrieparametern und dem absoluten Fehler	49
6.2	Vergleich durchschnittliche R2-Scores über alle Folds. R2-Score mit MAE-Loss → gefolgt von R2-Score mit R2-Loss	51
6.3	Pearson Korrelationskoeffizient zwischen den Geometrieparametern und dem absoluten Fehler	52
6.4	Die angewandten Datenaugmentierungen und ihre Parameter	56
6.5	R2-Scores für die angewandten Datenaugmentierungen	57
6.6	R2-Scores für die angewandten Datenaugmentierungen bei Training mit <i>apc_kreuzstöße</i> und Evaluierung auf <i>weld_scans</i>	62
6.7	R2 Scores der verschiedenen Methoden im Vergleich	64