KI-basierte Fahrzeugsegmentierung korrosionskritischer Konstruktionen durch die Analyse konstruktiver Maßnahmen, Elektrolytbeanspruchung und Bauteileigenschaften

AI-Based Vehicle Segmentation of Corrosion-Critical Designs by Analyzing Design Measures, Electrolyte Stress, and Component Properties

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Marcel Gollé-Leidreiter

Berichter/in: Univ.-Prof. Dr. sc. Julia Kowalski

Univ.-Prof. Dr.-Ing.habil. Brita Daniela Zander

Tag der mündlichen Prüfung: 26.03.2026

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Danksagung

An erster Stelle möchte ich Frau Prof. Dr.sc. Kowalski danken, die mir diese Doktorarbeit ermöglicht hat und mich in meiner Promotion unterstützt hat.
Ebenso möchte ich mich bei meinem Betreuer Dr. Andreas Mittelbach für die sehr gute Betreuung bedanken.
Bedanken möchte ich mich weiterhin bei allen, mit denen ich meinen Arbeitsalltag geteilt habe. Dabei gilt dieser Dank im Besonderen Julian Schneider als Mitstreiter, sowie Dominik Erd und Dhusjanth Thavathilakarjah als unsere Nachfolger. Außerdem gilt mein Dank Dr. Konstantin Kapfer, der immer als Sparringpartner bereitstand. Darüber hinaus möchte ich mich bei allen Studierenden bedanken, die ich in dieser Zeit betreuen durfte. Mein Dank gilt auch allen Mitarbeitenden der Abteilung RD/KSO.
Besonders möchte ich mich bei meiner Familie bedanken. Ich bedanke mich bei meiner Partnerin Annika Reinhardt. Danke, dass Du mich immer bedingungslos unterstützt hast. Ferner möchte ich mich bei meinem Bruder Philipp Gollé-Leidreiter bedanken, der mit seiner Erfahrung und seinem Scharfsinn eine große Hilfe war. Ich möchte mich auch bei meinen Eltern bedanken, die mich stets unterstützt und mir so meinen Weg ermöglicht haben.

Inhaltsverzeichnis

Abbildungsverzeichnis **vii**

Tabellenverzeichnis **ix**

1 Einleitung **1**
1.1 Motivation . . . 1
1.2 Forschungslücke und Ziel der Arbeit . . . 2
1.3 Gliederung . . . 2
1.4 Reader‘s Guide . . . 3

2 Theoretische Grundlagen der Korrosionsvorhersage im Automobilbau **6**
2.1 Künstliche Intelligenz . . . 6
2.1.1 Feature Engineering . . . 7
2.1.2 ML-Bewertungskriterien . . . 9
2.1.3 Methoden der Künstlichen Intelligenz . . . 11
2.1.4 Transfer Learning . . . 14
2.1.5 Hyperparameteroptimierung . . . 14
2.1.6 Explainable Artificial Intelligence . . . 15
2.2 Korrosionstheorie . . . 17
2.2.1 Korrosionsprozess . . . 17
2.2.2 Korrosionstest . . . 19
2.2.3 Korrosionssimulation . . . 21
2.2.4 Korrosive Schutzmaßnahmen . . . 21
2.2.5 Einflüsse des Materials auf das Korrosionsverhalten . . . 25
2.3 Elektrolytbeanspruchung . . . 26
2.3.1 Einfluss der Elektrolytbeanspruchung auf das Korrosionsverhalten . . . 26
2.3.2 Lösungsmethodik der numerischen Strömungsmechanik . 27
2.3.3 Ergebnisse der numerischen Strömungsmechanik . . . 27
2.3.4 Kapillarkraft . . . 28
2.4 Beschichtungsverfahren . . . 28
2.4.1 Kathodische Tauchlackierung . . . 29
2.4.2 Decklacksystem . . . 30

3 Forschungsstand der Korrosionsvorhersage im Automobilbau **31**
3.1 Segmentierung korrosionskritischer Konstruktionen . . . 31
3.1.1 Datenaufbereitung . . . 31
3.1.2 Implementierung der Erkennung von Kantenflächen . . . 33
3.1.3 Implementierung der Erkennung von Flanschflächen . . . 35

3.2 Grundlagen der Korrosionsprognose 38

4 Klassifizierung korrosionsgeschützter Bereiche 40
4.1 Implementierung der Geometriedatenaufbereitung 42
4.2 Implementierung der Klebstoffklassifizierung 43
4.2.1 Maßnahmenspezifische Aufbereitung der Geoemtriedaten 43
4.2.2 Transfer Learning zur Implementierung des Klebstoffnetzwerks 44
4.2.3 Implementierung der Eigenschaften zur Klebstoffklassifizierung 47
4.2.4 Hyperparametertuning 51
4.2.5 Klassifizierungsergebnisse 51
4.3 Implementierung der Nahtabdichtungsklassifizierung 54
4.3.1 Transfer Learning zur Implementierung des Nahtabdichtungsnetzwerks 54
4.3.2 Implementierung der Eigenschaften zur Klassifizierung von durch Nahtabdichtung geschützten Bereichen 56
4.3.3 Hyperparametertuning 57
4.3.4 Klassifizierungsergebnisse 58
4.4 Implementierung der Wachsklassifizierung 60
4.4.1 Transfer Learning zur Implementierung des Wachsnetzwerks 60
4.4.2 Implementierung der Eigenschaften zur Wachs-Klassifizierung 62
4.4.3 Hyperparametertuning 63
4.4.4 Klassifizierungsergebnisse 64
4.5 Analyse der Netzwerke mittels Explainable AI 66
4.5.1 Analyse und Anpassung der Flanschklassifizierung mittels Shapley-Werten 67
4.5.2 Analyse der NAD-Klassifizierung mittels Shapley-Werten 69
4.5.3 Analyse der Klebstoffklassifizierung mittels Shapley-Werten 70
4.5.4 Analyse der Wachsklassifizierung mittels Shapley-Werte . 72

5 Auswertung der Elektrolytbeanspruchung 75
5.1 Betrachtung der Elektrolytbeanspruchung mittels Simulationen . 75
5.1.1 Randbedingungen und Berechnungsmodelle der verwendeten Elektrolytsimulationen 76
5.1.2 Beschreibung der Simulationsergebnisse 76
5.2 Übertrag der Simulationswerte 77
5.2.1 Übertrag auf Kantenelemente 77
5.2.2 Übertrag auf Flanschelemente 78
5.3 Limitationen der Elektrolytsimulation 78
5.3.1 Limitationen der Auswertung durch Geometrievernetzung 78
5.3.2 Korrektur der Elektrolytsimulation in Flanschbereichen . 79
5.3.3 Korrelation des Elektrolytverhalten zwischen Simulation und Korrosionstests 80

5.4 Erweiterung des Übertrags unter Berücksichtigung der Simulationslimitationen . . . 80
5.4.1 Geometrieneuvernetzung zur Minimierung der Auswirkungen der Limitationen . . . 80
5.4.2 Erweiterung des Übertrags unter Berücksichtigung von fehlenden Datenpunkten . . . 81
5.4.3 Anpassen der Wetting-Werte aufgrund von Flanschflächen 83
5.4.4 Integration mehrerer Simulationen . . . 87
5.5 Definition eines Korrosionsrisikolabels auf Basis der ausgewerteten Simulationswerte . . . 88
5.6 Übertragene Simulationswerte . . . 88

6 Korrosionsprognose 90
6.1 Analyse der Korrosionsrisikomatrix . . . 91
6.2 Kantenkorrosionsprognose . . . 92
6.2.1 Integration der Wetting-Werte in die Korrosionsprognose 93
6.2.2 Evaluierte Kantenkorrosionsstellen am Gesamtfahrzeug . 96
6.2.3 Ergebnisse der Kantenkorrosionsprognose . . . 97
6.3 Flanschkorrosionprognose . . . 98
6.3.1 Aufbau der KI-basierten Flanschkorrosionsprognose . . . 98
6.3.2 Ergebnisse des Trainings der Flanschkorrosionsprognose . 100
6.3.3 Validierung der Prognose anhand eines Gesamtfahrzeugs 100
6.3.4 Diskussion der Flanschprognoseergebnisse . . . 105

7 Diskussion 109
7.1 Beschreibung des KI-basierten Systems zur Korrosionsprognose . 109
7.1.1 Diskussion des Korrosionsprognosesystems im Kontext der Explainable AI . . . 110
7.2 Diskussion der Laufzeit des Korrosionsprognosesystems . . . 111
7.3 Speicherbedarf des Korrosionsprognosesystems . . . 116
7.4 Diskussion der Trainingsdatensätze und deren Erweiterungspotential . . . 116
7.4.1 Trainingsdatensätze der Korrosionsrisikoprognose . . . 117
7.4.2 Trainingsdatensätze der KI-basierten Korrosionsprognose 118
7.5 Stärken und Limitierungen des Systems . . . 118
7.6 Bewertung des Korrosionsprognosesystems im Kontext der Forschungslücke . . . 120
7.6.1 Bewertung der Korrosionsrisikoprognose . . . 120
7.6.2 Bewertung der Elektrolytsimulationsauswertung . . . 121
7.6.3 Bewertung der KI-basierten Korrosionsprognose . . . 122
7.6.4 Bewertung des Gesamtsystems Korrosionsprognose . . . 122

8 Zusammenfassung und Ausblick 123
8.1 Zusammenfassung . . . 123
8.2 Ausblick . . . 124

A FEM-Datengrundlage 126

B Evaluierte Stellen **128**
B.1 Hiflsberechnungen für Geometrien 145

C Literaturverzeichnis **146**

Abbildungsverzeichnis

1.1 Übersicht der Kapitel und Arbeitsaufbau 3

2.1 Vergleich zwischen manueller Suche und Rastersuche 15
2.2 Schliffbild an der Oberfläche eines Bleches 18
2.3 Realaufbau eines NAD-Bauteils 22
2.4 Visualisierung eines digitalen NAD-Bauteils 23
2.5 Realapplikation von Wachs auf einem Bauteil 23
2.6 Visualisierung eines digitalen Wachs-Bauteils 24
2.7 Realapplikation des Klebstoffs am Bauteil 25
2.8 Digitale Dokumentation des Klebstoffes 25
2.9 Schichten eines Lackierungssystems nach Streitberger 29

3.1 Beispielgeometrie mit Bauteilkanten 33
3.2 Visualisierung des Strahlenansatzes 34
3.3 Beispielgeometrien mit Bauteilflansche 35

4.1 Ablaufdiagramm eines KI-Systems zu Klassifizierung 41
4.2 Schema der Geometrieneuvernetzung 42
4.3 Verteilung des gelabelten Datensatzes 46
4.4 Visualisierung eines Falzes 49
4.5 Ergebnis der Klassifizierung eines Gesamtfahrzeugs 53
4.6 Ausschnitte der Ergebnisse der Klebstoffklassifizierung 54
4.7 Verteilung des NAD-Trainingsdatensatzes 55
4.8 Ergebnis der NAD-Gesamtfahrzeugsklassifizierung 59
4.9 Ausschnitte der Ergebnisse der Nahtabdichtungsklassifizierung . 60
4.10 Verteilung des gelabelten Wachs-Datensatzes 62
4.11 Ergebnis der Wachs-Klassifizierung eines Gesamtfahrzeugs . . . 65
4.12 Ausschnitte der Ergebnisse der Wachsklassifizierung 66
4.13 Shapley-Werte der vier wichtigsten Flanscheigenschaften 67
4.14 Niedrigste Shapley-Werte der Flansch-Klassifizierung 67
4.15 Vergleich der Trainings- und Validierungsgenauigkeit 68
4.16 Differenz zwischen Trainings- und Validierungsfehler 69
4.17 Höchste Shapley-Werte eines Netzwerks zur NAD-Klassifizierung 69
4.18 Niedrigste Shapley-Werte der NAD-Klassifizierung 70
4.19 Höchste Shapley-Werte der Klebstoff-Klassifizierung 71
4.20 Niedrigste Shapley-Werte der Klebstoff-Klassifizierung 72
4.21 Shapley-Werte der geometrischen Eigenschaften 73
4.22 Höchste Shapley-Werte des Netzwerks zur Wachs-Klassifizierung 73

5.1 Im Rahmen der Arbeit analysiertes Bauteil 75

5.2 Visualisierung der Simulationsergebnisse der Time of Wetness . 77
5.3 Visualisierung einer Object Bounding Box 78
5.4 Unterschiede durch die Geometrievernetzung 79
5.5 Maximallänge l_{max} in Abhängigkeit der Wetting-Werte 81
5.6 Ergebnisse des Ansatz zur Korrektur von NaN-Werten 83
5.7 Ergebnisse des Ansatzes zur Korrektur von Null-Werten 84
5.8 Betrachtung von T_{max} in Abhängigkeit der Berechnungszeit . . 85
5.9 Ablaufdiagramm der Auswertungsmethodik 87
5.10 Ergebnisse des Minikomponenten- und des Kapillarkraftansatzes 87
5.11 Ausgewertetes Gesamtfahrzeug 89

6.1 Ablaufdiagramm der Implementierung der Korrosionsprognose . 91
6.2 Wetting-Werte und Winkel auf einem Probeblech 93
6.3 Verteilung von Wetting-Werten und Winkeln von Bauteilen . . . 94
6.4 Interpolationsansatzes und Beispiel eines Kantenelements 95
6.5 Simulations- und Realtestergebnisse nach Woche 8 96
6.6 Auswertung der Kanten eines Gesamtfahrzeugs 97
6.7 Trainingsdaten des Flanschprognosenetzwerks 99
6.8 Prognostiziertes Gesamtfahrzeug 100
6.9 Flanschstellen 1 und 2 im Realtestverhalten und in der Prognose 101
6.10 Flanschstelle 3 im Realtestverhalten und in der Prognose 102
6.11 Flanschstelle 12 im Realtestverhalten und in der Prognose . . . 103
6.12 Flanschstelle 13 im Realtestverhalten und in der Prognose . . . 104
6.13 Vergleich von Realbauteilen 105
6.14 Flanschstelle 5 im Realtestverhalten und in der Prognose 106

7.1 Übersichtsworkflow der Anwendung nach Implementierung . . . 110
7.2 Interface des Datenmodells 119

A.1 Die Daten, die zum Training des Netzwerks benutzt werden . . 127

B.1 Flanschstellen 1 und 2 im Realtest und Simulation 128
B.2 Flanschstelle 3 im Realtest und Simulation 129
B.3 Flanschstelle 4 im Realverhalten und Prognose 130
B.4 Flanschstelle 5 im Realverhalten und Prognose 131
B.5 Flanschstelle 6 im Realverhalten und Prognose 132
B.6 Flanschstelle 8 im Realverhalten und Prognose 133
B.7 Flanschstelle 9 im Realverhalten und Prognose 134
B.8 Flanschstelle 10 im Realverhalten und Prognose 135
B.9 Flanschstelle 11 im Realverhalten und Prognose 136
B.10 Flanschstelle 12 im Realverhalten und Prognose 137
B.11 Flanschstelle 13 im Realverhalten und Prognose 138
B.12 Flanschstelle 14 im Realverhalten und Prognose 139
B.13 Flanschstelle 15 im Realverhalten und Prognose 140
B.14 Flanschstelle 16 im Realverhalten und Prognose 141
B.15 Flanschstelle 17 im Realverhalten und Prognose 142
B.16 Flanschstellen 18 und 19 im Realverhalten und Prognose 144

Tabellenverzeichnis

2.1 Aufbau einer Fehlermatrix im Fall einer Binärklassifikation . . . 9
2.2 Randbedingungen der Korrosionssimulationen 21

4.1 Transfer Learning Parameter in der Klebstoffklassifizierung . . . 45
4.2 Aufstellung des Klebstoff-Trainingsdatensatzes 46
4.3 Werteraum und Hyperparameter des Klebstoffmodells 51
4.4 Klassifikationsmetriken des trainierten Klebstoff-Netzwerks . . . 52
4.5 Konfusionsmatrix für die zwei Klassen 0 und 1 52
4.6 Mittels Dreiecksfläche gewichtete Konfusionsmatrix 52
4.7 Klassifikationsgenauigkeiten des Gesamtfahrzeugs 53
4.8 Parameter des Flanschnetzwerks im Transfer Learning 55
4.9 Aufbau des NAD-Trainingsdatensatz 56
4.10 Parameterraum und ermittelte Hyperparameterwerte 57
4.11 Validierungsgenauigkeiten des NAD-Netzwerks 58
4.12 Mittels Dreiecksfläche gewichtete Konfusionsmatrix 58
4.13 Über Dreiecksflächen gewichtete Konfusionsmatrix 59
4.14 Genauigkeiten der NAD-Klassifizierung eines Gesamtfahrzeugs . 59
4.15 Parameter zur Klassifizierung von wachsgeschützen Bereichen . 61
4.16 Zusammensetzung des Wachs-Trainingsdatensatzes 61
4.17 Parameterraum und Hyperparameter der Rastersuche 64
4.18 Trainingsgenauigkeiten des Wachsnetzwerks 64
4.19 Konfusionsmatrix für die zwei Klassen 0 und 1 64
4.20 Gewichtete Konfusionsmatrix der Wachsklassifizierung 65
4.21 Genauigkeiten der Wachsklassifizierung des Gesamtfahrzeugs . . 65
4.22 Vergleich der beiden Netzwerke 68

6.1 Flanschstellen 1 und 2, Vergleich von Realtest und Prognose . . 101
6.2 Flanschstelle 3, Vergleich von Realtest und Prognose 102
6.3 Flanschstelle 12, Vergleich von Realtest und Prognose 103
6.4 Flanschstelle 13, Vergleich von Realtest und Prognose 104
6.5 Flanschstelle 5, Vergleich von Realtest mit Prognose 105
6.6 Ergebnisse der ausgewerteten Flanschstellen 107
6.7 Auswertung der Ergebnisse aus Tabelle 6.6 107

7.1 Genutzte Hardware 112
7.2 Übersicht der Datensätze 112
7.3 Klassifizierungszeiten für jedes Label 113
7.4 Vergleich der Klassifikationsleistung zweier Klebstoff-Netzwerke 114
7.5 Vergleich der Laufzeit mit und ohne Octree 115

7.6 Vergleich der Laufzeit unterschiedlicher Vorhersagetools 115
7.7 Speicherbedarf des Korrosionsprognosesystem 116
7.8 Reduktion der erkannten korrosionskritischen Bereiche 122

B.1 Prognose und Realergebnisse der Flanschstellen 1 und 2 128
B.2 Prognose und Realergebnisse der Flanschstellen 3 129
B.3 Prognose und Realergebnisse der Flanschstelle 4 130
B.4 Prognose und Realergebnisse der Flanschstelle 5 131
B.5 Prognose und Realergebnisse der Flanschstelle 6 132
B.6 Prognose und Realergebnisse der Flanschstelle 8 133
B.7 Prognose und Realergebnisse der Flanschstelle 9 134
B.8 Prognose und Realergebnisse der Flanschstelle 10 135
B.9 Prognose und Realergebnisse der Flanschstelle 11 136
B.10 Prognose und Realergebnisse der Flanschstelle 12 137
B.11 Prognose und Realergebnisse der Flanschstelle 13 138
B.12 Prognose und Realergebnisse der Flanschstelle 14 139
B.13 Prognose und Realergebnisse der Flanschstelle 15 140
B.14 Prognose und Realergebnisse der Flanschstelle 16 141
B.15 Prognose und Realergebnisse der Flanschstelle 17 142
B.16 Prognose und Realergebnisse der Flanschstellen 18 und 19 . . . 143

Abkürzungsverzeichnis

SDF Shape Diameter Function

NAD Nahtabdichtung

HRK Hohlraumkonservierung, Wachs

OBB Objekt Bounding Box

AABB Achsenorientierte Bounding Box, *engl.* Axis-aligned Bounding Box

ML Maschninelles Lernen

KI Künstliche Intelligenz

NN Neuronales Netz

NARX Nonlinear Autoregressive Exogenous Model

TP Wahr-positiv *engl.* True positive

FP Falsch-positiv *engl.* False positive

TN Wahr-negativ *engl.* True negative

FN Falsch-negativ *engl.* False negativ

TPA Präzision *engl.* True positiv accuracy

TPR Sensitivität *engl.* True positive rate

MCC Matthew Correlation Coefficient

MaxAE Maximaler absoluter Fehler

MAE Mittlerer durchschnittlicher Fehler

MSE Mittlerer quadrierter Fehler

RMSE Standardfehler

MAPE Mittlerer absoluter prozentualer Fehler

KTL Kathodische Tauchlackierung

CR300 Stahlsorte

PMMA Polymethylmethacrylat

SB Stahl blank

GI Feuerverzinkt

FEM Finite-Elemente-Methode

STL Stereolithografie, (Datenformat)

PVC Polyvinylchlorid

ToW Time of Wettnessg

PLM Product Lifecycle Managment

CAD Computer Aided Design

xAI Erklärbare künstliche Intelligenz *engl.* Explainable Artificial Intelligence

CFD Numerische Strömungsmechanik *engl.* Computational Fluid Dynamics

SPH Solid particle hydrodynamics

1 Einleitung

1.1 Motivation

Die Automobilindustrie ist mit vielfältigen Herausforderungen konfrontiert, die den Entwicklungsprozess neuer Produkte erheblich beeinflussen. Die zunehmende Marktsättigung und das stetig wachsende Angebot führen dazu, dass Hersteller gezwungen sind, ihre Produktentwicklungszeiten drastisch zu verkürzen, um wettbewerbsfähig zu bleiben [1]. Gleichzeitig stehen sie unter enormem Kostendruck, der es erforderlich macht, effiziente und ressourcenschonende Entwicklungsmethoden zu etablieren [2]. Diese Herausforderungen verlangen nach innovativen Ansätzen. Diese sollen in der Lage sein, bereits früh im Entwicklungsprozess Aussagen über das Produkt treffen zu können. Dies ermöglicht es, früh Änderungen einzubringen und die Entwicklungszeiten und Kosten durch die Abhängigkeit von physischen Prototypen zu verringern [3]. Ein wichtiger Aspekt, welcher im Rahmen der Entwicklung eines Fahrzeugs betrachtet werden muss, ist die Vermeidung von korrosionskritischen Bereichen. Unter metallischer Korrosion versteht man die physikochemische Wechselwirkung zwischen einem metallischen Werkstoff und seiner Umgebung, die zu einer messbaren Veränderung des Materials und damit zu einer Beeinträchtigung der Bauteilfunktion führt [4]. Korrosion ist insbesondere in der Automobilindustrie eine besondere Herausforderung, da korrosionsbedingte Schäden eine große wirtschaftliche Bedeutung haben [5]. Dies ist in der Beeinträchtigung der Funktion und damit auch der Sicherheit eines Fahrzeugs begründet [6].

In der Literatur existieren verschiedene Ansätze, Korrosion mithilfe von FEM-Simulationen zu berechnen [7–10]. Eine Übertragung dieser Methoden auf ein komplettes Fahrzeug ist jedoch aufgrund des enormen Speicher- und Rechenbedarfs nicht praktikabel [11].
Die gegenwärtig genutzten Ansätze zur Vorhersage des Korrosionsfortschrittes teilen sich in zwei Bereiche. Es existiert die Erkennung korrosionskritischer Bereiche, die Korrosionsrisikoerkennung [12], sowie die Vorhersage des Korrosionsfortschritts, die Korrosionsprognose [13]. Die bisherige Vorhersage des Korrosionsfortschritts lässt sich nur eingeschränkt auf ein Gesamtfahrzeug übertragen. Zum einen werden Korrosionsrisikobereiche nicht vollständig abgebildet, da eine Berücksichtigung korrosionsvermeidender Maßnahmen nicht gegeben ist. Auch erfolgt eine Abschätzung der Benässung von korrosionskritischen Bereichen nur indirekt und damit sehr ungenau.
Um diese Limitierungen zu überwinden, werden in der vorliegenden Arbeit drei Ansätze entwickelt, die die Korrosionsrisikoerkennung auf Gesamtfahrzeug-

ebene erweitern. Die Integration von korrosionsvermeidenden Maßnahmen, die Vorhersage des Korrosionsverhaltens von Flanschflächen sowie die Berücksichtigung der Elektrolytbeanspruchung mittels Simulation.

1.2 Forschungslücke und Ziel der Arbeit

Die Forschungslücke, die diese Arbeit betrachtet, kann in mehrere Bereiche unterteilt werden, die aus Kapitel 1.1 resultieren. Die Vorhersage des Korrosionsverhaltens einer Fahrzeugkarosserie benötigt eine Betrachtung von Bereichen, die durch Korrosion gefährdet sind: die Korrosionsrisikoerkennung. Eine Betrachtung von Bereichen, die durch korrosionsvermeidende Maßnahmen geschützt sind, existiert nicht. Diese wird jedoch für eine Vorhersage benötigt. Darüber hinaus existiert keine Methodik, um eine Vorhersage der Korrosion an komplexen Geometrien unter Berücksichtigung von Elektrolytbeanspruchung und geometrischer Lage im Kontext der Automobilentwicklung durchzuführen.

Das Ziel der vorliegenden Dissertation lässt sich wie folgt beschreiben: Die Entwicklung eines KI-basierten Systems zur automatischen, simulationsgestützten Vorhersage des Korrosionsverhaltens an Gesamtfahrzeugen unter Berücksichtigung von Korrosionsschutzmaßnahmen, Elektrolytbeanspruchung und spezifischer Bauteileigenschaften. Das System kann korrosionskritische Bereiche automatisiert erkennen und potenzielle Korrosionsstellen frühzeitig identifizieren. Konkret ergeben sich folgende Anforderungen an das System:

1. Automatisierte Identifikation korrosionsgeschützter Bereiche
2. Berücksichtigung konstruktiver Maßnahmen sowie Materialeigenschaften
3. Definieren und Verarbeiten der Interaktionen zwischen korrosionsgeschützter und korrosionskritischer Bereiche
4. Integration simulativer Methoden zur Berechnung der Elektrolytbeanspruchung
5. Prognose von Flanschverbindungen auf Basis der ausgewerteten Elektrolyteinflüsse
6. Validierung der Ergebnisse durch Abgleich mit experimentellen Testdaten

Sind diese Anforderungen erfüllt, kann das oben definierte Ziel erreicht werden.

1.3 Gliederung

Die Struktur der vorliegenden Arbeit ist in Abbildung 1.1 visualisiert. Kapitel 2 und 3 beschreiben die Grundlagen dieser Arbeit. Die entwickelte Methodik ist in Kapitel 4, 5 und 6 erläutert. Wobei in Kapitel 4 die Erkennung von korrosionsgeschützten Bereichen und in Kapitel 5 die Methodik der Integration

einer Partikelsimulation beschrieben werden. Die Methodik der Vorhersage des Korrosionsverhaltens wird in Kapitel 6 erklärt. Damit bilden die Kapitel 4, 5 und 6 den thematischen Schwerpunkt der vorliegenden Arbeit. Kapitel 7 und 8 enthalten die Diskussion der Ergebnisse sowie eine Zusammenfassung mit Ausblick.

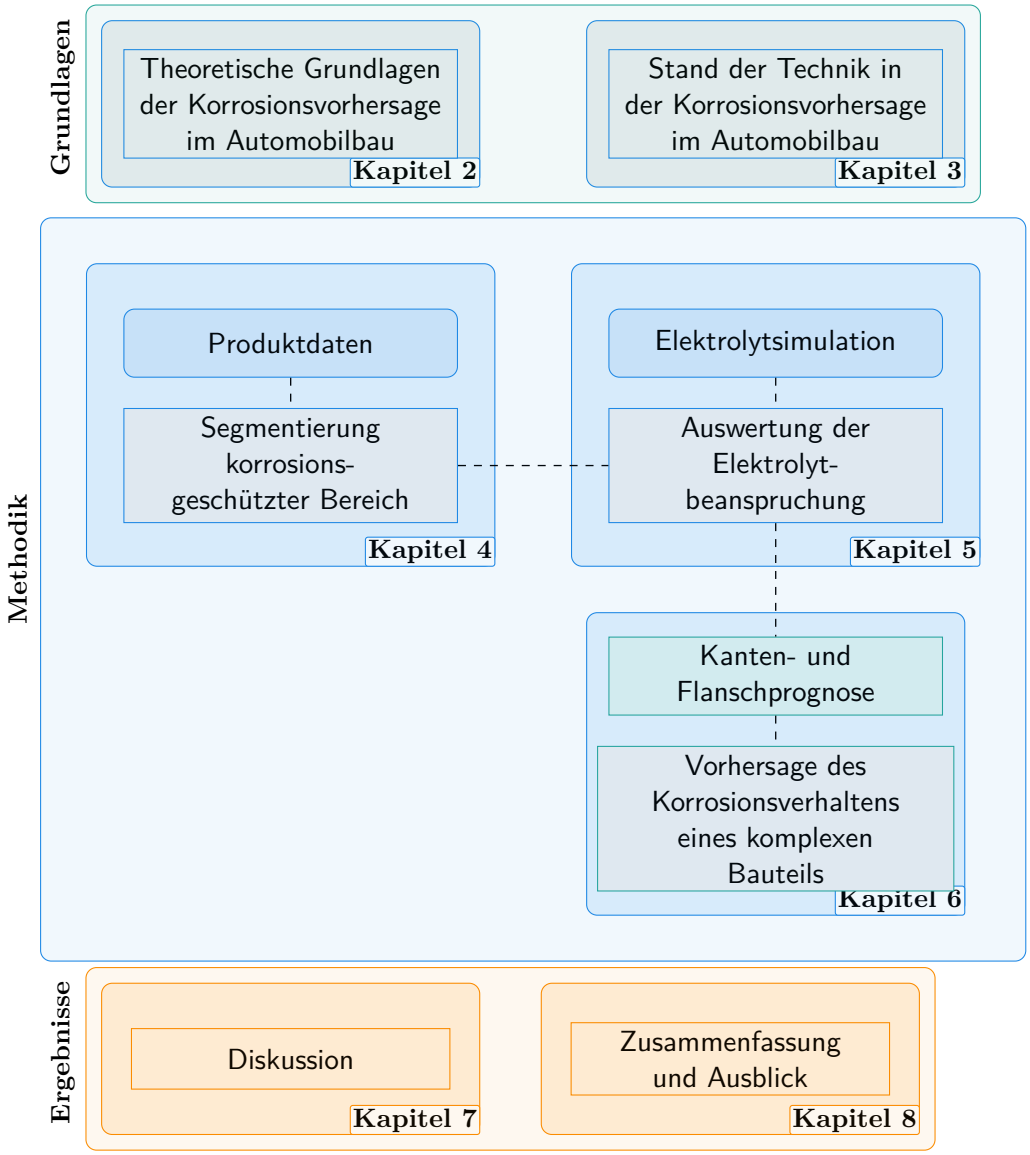


Abb. 1.1: Übersicht der Kapitel und Arbeitsaufbau

1.4 Reader‘s Guide

Für das Verständnis dieser Arbeit ist es wichtig die Semantik zu erklären. Dazu werden im Folgenden einige Begriffe definiert:

Gesamtfahrzeug: Gesamtfahrzeug bezieht sich in diesem Rahmen auf die Karosserie eines Fahrzeugs, ohne Anbauteile.

Segmentierung: Die Segmentierung beschreibt die Unterteilung der Fahrzeugoberfläche in betroffene und nicht betroffene Bereiche eines korrosionsrelevanten Einflusses. Eine Segmentierung kann mehrere Einflüsse berücksichtigen. Diese Segmentierung wird im Rahmen der Arbeit durch eine Klassifizierung erreicht.

Klassifizierung: Die Klassifizierung beschreibt die Methodik, mit deren Hilfe eine Oberfläche segmentiert wird. Eine Klassifizierung ordnet jedes Element einer Klasse zu.

Kante: Kanten werden als Bauteilkanten bezeichnet, die durch das Vereinzeln von Blechen entstehen. Diese Kanten werden als korrosionskritische Bereiche definiert.

Kantenelement: Ein Kantenelement beschreibt ein diskretes Element der Geometrie. Kantenelemente sind Verbindungen zwischen Punkten. Mehrere Kantenelemente bilden eine Kante.

Flansche: Ebenso wie Kanten bezeichnen Flansche korrosionskritische Bereiche. Dabei bezeichnen Flansche Bereiche, in denen zwei Bauteile miteinander verbunden sind.

Oberflächenelemente: Ein Oberflächenelement beschreibt im Kontext dieser Arbeit ein Dreieck. Diese Oberflächenelemente werden der Klassifizierung übergeben und bekommen ein Korrosionsrisikolabel zugewiesen.

Korrosionsrisikolabel: Bezeichnet einen binären Operator, der das Korrosionsrisiko eines einzelnen Oberflächenelements beschreibt.

Korrosionsrisiko: Das Korrosionsrisiko ergibt sich aus der Segmentierung einer Oberfläche. Das Korrosionsrisiko beschreibt dabei, ob an einem Oberflächenelement Korrosion entstehen kann oder nicht.

Korrosionsrisikostellen: Korrosionsrisikostellen beschreiben mindestens ein zusammenhängendes Dreieck, an dem ein Korrosionsrisiko entstehen kann. In der Elektrolytauswertung wird eine zusammenhängende Fläche auch Komponente genannt.

Korrosionsrisikoprognose: Die Korrosionsrisikoprognose beschreibt die Vorhersage der Korrosionsrisikostellen mittels eines Ensembles von KI-Netzwerken sowie definierten Beziehungen zwischen den einzelnen Klassifizierungen.

Time of Wetness: Time of Wetness beschreibt, wie lange im Rahmen einer Elektrolytsimulation ein Punkt von Fluidpartikeln umgeben ist. Diese Punkte werden auf der Oberfläche eines Bauteils platziert.

Wetting-Wert: Der Wetting-Wert entsteht aus der Time of Wetness und beschreibt, wie sehr ein Oberflächenelement benetzt ist.

Benässungszeit: Die Benässungszeit beschreibt die Zeit, in der eine reale Oberfläche durch Elektrolyt bedeckt ist.

Korrosionssimulation: Die Korrosionssimulation beschreibt eine FEM-Simulation, welche in 2D das Korrosionsverhalten von Korrosionsrisikostellen simuliert.

Korrosionsprognose: Die Korrosionsprognose nutzt die Korrosionssimulation als Grundlage für das Training. Die Korrosionsprognose ist eine 3D-Vorhersage des Korrosionsverhaltens von komplexen Geometrien. Umgangssprachlich wird die Korrosionsprognose als Korrosionssimulation bezeichnet.

KI-basierte Korrosionsprognose: Siehe Korrosionsprognose

Korrosionsprognosesystem: Das Korrosionsprognosesystem beschreibt das komplette System aus Korrosionsrisikobestimmung, Elektrolytauswertung und KI-basierten Korrosionsprognosen, welche auf Basis von Geometriedaten das Korrosionsverhalten vorhersagen können.

2 Theoretische Grundlagen der Korrosionsvorhersage im Automobilbau

In diesem Kapitel werden die Grundlagen behandelt, die zum Verständnis der Arbeit benötigt werden.

1. Künstlichen Intelligenz (KI): Hier wird die in der Arbeit benutzten Evaluationsmetriken beschrieben. Darüber hinaus werden die Grundlagen zur in der Arbeit angewendeten Hyperparametertuning und Explainable AI erläutert.

2. Korrosionstheorie: Der Korrosionsprozess sowie die daraus für eine Rohbaukonstruktion abgeleiteten korrosionskritischen Stellen werden definiert.

3. Elektrolytbeanspruchung: Dies beinhaltet neben dem Einfluss des Elektrolytverhaltens auf den Korrosionsprozess auch Simulationsansätze.

4. Beschichtungsverfahren: Unterschiedliche Beschichtungsverfahren, die das Korrosionsverhalten beeinflussen, werden erläutert.

2.1 Künstliche Intelligenz

Zum Verständnis dieser Arbeit sind Grundlagen im Bereich der KI erforderlich. Nach Bishop [14] bildet das Maschinelle Lernen (ML) ein Konzept der KI. In seinem einfachsten Fall beschreibt KI lediglich automatisiertes, intelligentes Verhalten von Systemen, z.B. automatisierte Laborroutinen [14]. Beim Maschinellen Lernen handelt es sich um einen Ansatz, bei dem Daten genutzt werden, um ein Modell zu trainieren. Dieses Modell kann mathematisch als Funktion $y(x)$ beschrieben werden. Nach Bishop [14] lernt ein Computerprogramm von der Erfahrung E, bezogen auf eine bestimmte Aufgabe T und ein Leistungsmerkmal P, wenn seine Leistung hinsichtlich der Aufgabe T unter Berücksichtigung von P durch Erfahrung R verbessert wird. Die Aufgabe T beschreibt dabei, wie der lernende Algorithmus das Ziel erreicht. Dies geschieht durch die Anwendung von mathematischen Modellen wie der Klassifizierung oder Regression. Trainiert wird das Modell anhand von Trainingsdatensätzen, die sowohl Input als auch das bekannte Ergebnis y enthalten. Das Leistungsmerkmal P steht für die Genauigkeit des Modells, die oft anhand von Testdatensätzen gemessen wird. Dabei werden vorhergesagte Ergebnisse mit den tatsächlichen

Ergebnissen verglichen, um eine Genauigkeitsbewertung zu erhalten. Die Erfahrung E bezieht sich auf die Art der Trainingsdatensätze, die in Form von beschrifteten oder nicht beschrifteten Daten verwendet werden können. Im Falle des überwachten Lernens sind die Daten bereits gekennzeichnet, während beim unüberwachten Lernen diese Kennzeichnung nicht vorhanden ist.

Neuronale Netze

Neuronale Netze lassen sich durch drei Bereiche unterscheiden. Den Bereich der Architektur, des Lernens und der Aktivierungsfunktion [15]. Die Architektur umfasst den Aufbau des Netzes, einschließlich der Anzahl der Schichten, der Neuronen und der Verbindungen zwischen ihnen. Aufgrund der Ähnlichkeiten zwischen Neuronalen Netzen und biologischen Systemen werden Teile des Netzes als Neuronen und Schichten bezeichnet. Das Lernen beschreibt den Prozess, bei dem das Netzwerk die Gewichte der einzelnen Neuronen anpasst, um einen Fehler zu minimieren. Die Aktivierungsfunktion bestimmt, mit welchem Wert die Informationen innerhalb des Netzes weitergegeben werden. Ein klassisches Neuronales Netz besteht aus einer Eingangs- und einer Ausgangsschicht. Deep Learning Netze, eine Unterkategorie der Neuronalen Netze, unterscheiden sich von klassischen Neuronalen Netzen in der Komplexität des Systems. Deep Learning zeichnet sich durch die Richtung der Neuronen und die Anzahl der Schichten aus, wobei einfache Neuronale Netze vorwärts gerichtet sind, während bei höherer Komplexität die Richtung variieren kann und die Anzahl der Schichten zunimmt. Deep Learning ist robuster gegenüber volatilen Daten und ermöglicht es, Repräsentationen besser zu lernen und zu erkennen. Allerdings muss für jedes System eine passende Architektur gewählt werden. Beispiele für Deep Learning-Algorithmen sind mehrschichtige Perzeptronen, Autoencoder/Autodecoder und Convolutional Neural Networks (CNN). Diese finden besonders in der Bildverarbeitung Anwendung [16].

2.1.1 Feature Engineering

Netzwerken werden strukturierte Daten übergeben. Diese können Bilder, Texte oder anderweitig strukturierte Daten sein. Die Anpassung, um strukturierte Daten zu erhalten, sowie die Aufbereitung der Daten wird im Rahmen dieser Arbeit „Feature Engineering“ genannt.

Shape Diameter Function nach Shapira [17]

Die Shape Diameter Function (SDF) ist eine Vorgehensweise, um eine Gruppe an 3D-Objekten zu charakterisieren. Dabei basiert dieser Ansatz auf der Idee, die Distanz zwischen einem Punktepaar zu bestimmen, das auf der Oberfläche des Objekts liegt. Die maximale Entfernung aus allen Punktepaaren wird als der Durchmesser des Objekts bestimmt. Auf das gesamte Objekt angewendet, ergibt dieser Ansatz Formkompaktheit. Dabei wird zu jedem Punkt in dem Volumen der Volumenoberflächenpunkt gesucht, welcher die kürzeste Distanz

zu diesem Punkt besitzt. Diese Funktion ist durch ihre Funktionsweise unabhängig von Translation und Rotation des jeweiligen Objekts und ist vor allem gegenüber Rauschen und Ungenauigkeiten in der Oberfläche sehr robust.

Feature Normalisierung nach Gereon [16]

Eine verbreitete Ursache von Fehlern beim Trainieren von Netzwerken sind wachsende oder verschwindende Gewichte durch das Trainieren von Daten, welche starke Unterschiede in ihrer Dimension haben. Um dies zu verhindern, existiert das Prinzip der Feature-Normalisierung.
Durch das Normalisieren wird sichergestellt, dass die Eingabemerkmale in einem ähnlichen Maßstab vorliegen. Liegen mehrere unterschiedliche Merkmale auf deutlich unterschiedlichen Bereichen der Skala, so kann das beim Trainieren zu Problemen führen, da durch den größeren Skalenbereich Merkmale in kleineren Bereichen unterbewertet werden. Durch die Normalisierung der Merkmale werden die Skalen so geändert, dass die Merkmale in ähnlichen Bereichen liegen. Eine Möglichkeit der Normalisierung ist die Min-Max-Normalisierung. Dabei wird jeder Merkmalswert zwischen 0 und 1 skaliert und wie folgt berechnet:

$$x_{norm} = \frac{(x - min(x))}{[max(x) - min(x)]} \tag{2.1}$$

Dabei ist x der ursprüngliche Wert, x_{norm} ist der normalisierte Wert des Merkmals, $min(x)$ ist der kleinste, $max(x)$ der größte Wert des Merkmals.
Eine weitere Methode der Normalisierung ist die Z-Score-Normalisierung. Bei dieser Methode wird jeder Merkmalswert zu einem Wert skaliert, der von der Standardabweichung dieses Merkmals entfernt ist. Vorteil ist dabei, dass diese Methode besonders mit Ausreißern gut umgehen kann:

$$x_{norm} = \frac{x - mean(x)}{std(s)} \tag{2.2}$$

Datensatzaugmentierung

Bei der Datensatzaugmentierung wird durch Anwendung von Transformationen der Daten eine erweiterte Datengrundlage erzeugt. Damit wird die Leistungsfähigkeit eines Machine-Learning-Modells verbessert. Dabei existieren unterschiedliche Möglichkeiten, dieses Ziel zu erreichen [18].

1. Interpolation: Bei der Interpolation werden vorhandene Datenpunkte benutzt, um daraus im Bereich der bestehenden Datenpunkte neue Datenpunkte zu generieren.

2. Extrapolation: Bei der Extrapolation werden im Gegensatz zur Interpolation außerhalb des bereits bekannten Bereich Punkte generiert und zum Datensatz hinzugefügt.

3. Noise: Durch das Hinzufügen von zufälligen Transformationen wird die Anzahl der Datenpunkte erhöht.

2.1.2 ML-Bewertungskriterien

Im Folgenden werden die Kriterien beschrieben, die im Rahmen dieser Arbeit genutzt werden, um Netzwerkarchitekturen zu evaluieren.

Bewertungsmetriken der Klassifikation

Im Falle einer binären Klassifikation beschreibt eine Fehlermatrix (engl. Confusion Matrix) die Fälle der Klassifikation [19].

	Positive ungewichtete Vorhersage (1)	**Negative ungewichtete Vorhersage (0)**
Positives ungewichtete Label (1)	Wahr-positiv	Falsch-negativ
Negative ungewichtetes Label (0)	Falsch-positiv	Wahr-negativ

Tabelle 2.1: Aufbau einer Fehlermatrix im Fall einer Binärklassifikation

Die Fehlermatrix ist in Tabelle 2.1 dargestellt. Dabei werden die Ergebnisse der Vorhersage mit den wahren Werten verglichen; vier unterschiedliche Fälle sind möglich.

1. Wahr-positiv (TP): Durch das Neuronale Netz korrekt als positiv klassifizierte Werte.
2. Wahr-negativ (TN): Durch das Neuronale Netz korrekt als negativ klassifizierte Werte.
3. Falsch-positiv (FP): Durch das Neuronale Netz falsch als positive klassifizierte Werte.
4. Falsch-negativ (FN): Durch das Neuronale Netz falsch als negativ klassifizierte Werte.

Werden diese Werte kombiniert, ergibt sich die totale Menge an positiven Werten P durch:

$$P = TP + FN \tag{2.3}$$

Es ergibt sich die vorhergesagte Menge an positiven Werten durch:

$$P' = TP + FP \tag{2.4}$$

Parallel dazu ergibt sich die Menge der wahren negativen Werte durch:

$$N = TN + FP \tag{2.5}$$

sowie die Menge der vorhergesagten negativen Werte durch:

$$N' = TN + FN \tag{2.6}$$

Diese Parameter können auch gewichtet werden. Dazu wird der Gewichtungsparameter w jeweils mit Gleichung 2.3, 2.4, 2.5 und 2.6 multipliziert. Auf Basis dieser Parameter wird die Genauigkeit eines Neuronalen Netzes bestimmt. Durch das Auswerten dieser Werte ergeben sich weitere Möglichkeiten der Analyse:

1. Präzision TPA: Die Präzision beschreibt die Eigenschaft des Neuronalen Netzes mit welcher Genauigkeit es positive Werte korrekt klassifizieren kann:
$$TPA = \frac{TP}{TP + FP} = \frac{TP}{P'} \tag{2.7}$$

2. Trefferquote TPR: Die Trefferquote beschreiben die Wahrscheinlichkeit, dass ein wahrer Wert als positiv klassifiziert wird:
$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{2.8}$$

3. F-Maß: Das F-Maß [20] beschreibt eine Kombination aus Recall und Präzision. Damit ist dies ein Maß für ein Netz, welches beide Faktoren einbezieht:
$$F = 2\frac{TPR * TPA}{TPR + TPA} \tag{2.9}$$

4. Genauigkeit ACC: Die Genauigkeit ([21]) beschreibt die Wahrscheinlichkeit, mit der ein richtiger Wert zugewiesen wird.
$$ACC = \frac{TP + TN}{P + N} \tag{2.10}$$

5. MCC: Der Matthews Correlation Coefficient (MCC) [22], kombiniert alle der oben gennanten Parameter:
$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) + (TP + FN) + (TN + FP) + (TN + FN)}} \tag{2.11}$$

Der Matthews-Korrelationskoeffizient (MCC) ermöglicht eine robuste Bewertung von Klassifikationsmodellen, insbesondere bei stark unausgeglichenen Datensätzen. Sein Wertebereich liegt zwischen -1 und 1, wobei 1 eine perfekte, 0 eine zufällige und -1 eine vollständig entgegengesetzte Klassifikation beschreibt. Zur Berücksichtigung unterschiedlicher Klassenverteilungen kann zudem ein gewichteter MCC berechnet werden.

Bewertungsmetriken der Regression

Im Falle eines Netzwerkes, welches als Output keine Klassifikationen, sondern Vorhersagen eines numerischen Wertes liefert, werden andere Werte zur Evaluierung des Ergebnisses benötigt [23]. Dabei werden die geschätzten Werte mit den wahren Werten verglichen und daraus Metriken entwickelt, die im Rahmen dieser Arbeit verwendet werden, um die Vorhersagen zu evaluieren:

1. Maximum Absolute Error (MaxAE):

$$MaxAE(r_i,r) = \max_{i=1,\dots,n_A} |r_i - r| \tag{2.12}$$

2. Mean Average Error (MAE):

$$MAE(r_i,r) = \frac{1}{n_A} \sum_{i=1}^{n_A} |r_i - r| \tag{2.13}$$

3. Mean Squared Error (MSE):

$$MSE(r_i,r) = \frac{1}{n_A} \sum_{i=1}^{n_A} (r_i - r)^2 \tag{2.14}$$

4. Root Mean Squared Error (RMSE):

$$RMSE(r_i,r) = \sqrt{MSE} \tag{2.15}$$

5. Mean Absolute Percentage Error (MAPE):

$$MAPE(r_i,r) = \frac{100\%}{n_A} \sum_{i=1}^{n_A} \frac{|r_i - r|}{|r_i|} \tag{2.16}$$

2.1.3 Methoden der Künstlichen Intelligenz

Folgende Methoden der Künstlichen Intelligenz werden in der vorliegenden Arbeit genutzt.

Charakteristika von Neuronalen Netzen

Neuronale Netze lassen sich als mathematisches Modell betrachten, welches Informationen verarbeiten kann. Basierend auf diesem Verständnis von Neuronalen Netzen können diese anhand von drei Charakteristika differenziert werden: Der Architektur der Netze, der Methodik wie Informationen verarbeitet werden und den Aktivierungsfunktionen.
Die Architektur eines Neuronalen Netzes ist durch die Anzahl der Schichten beschrieben, die das Netz aufbauen. Eingangs- und Ausgangsdimension sind durch die Form des Inputs bzw. des gewünschten Outputs definiert.
Das Lernen eines Neuronalen Netzes beschreibt, auf welche Art und Weise

das Neuronale Netz Informationen aus Trainingsdatensätzen extrahiert. Zentrale Bedeutung hat dabei die Gradientenfunktion. Sie bestimmt, wie das Neuronale Netz Gewichtungen im Laufe des Lernens anpasst und welche Kriterien zur Bewertung der Ergebnisse herangezogen werden.
Aktivierungsfunktionen sind Funktionen, die beschreiben, mit welchem Wert die aufgenommenen Informationen von Neuron zu Neuron gereicht werden. Beispielhaft hierbei sind:

1. Die ReLu-Funktion:
$$f(a) = \begin{cases} a & \text{if } a > 0, \\ 0 & \text{if } a \leqslant 0, \end{cases} \tag{2.17}$$

2. Die logarithmsiche Sigmoidkurve:
$$f(a) = (\frac{1}{1 + exp(a)}) - 1 \tag{2.18}$$

3. Die Identität:
$$f(a) = x \tag{2.19}$$

4. Der Tangens hyperbolicus:
$$f(a) = tanh(a) = \frac{2}{1 + exp(-2x)} - 1 \tag{2.20}$$

5. Die Softmax-Funktion:
$$f(a)_j = \frac{exp(x_j)}{\sum_{k=1}^{d} exp(x_k)} \tag{2.21}$$

Die Architektur eines Neuronalen Netzwerks wird über den Aufbau eines Neurons definiert. Ein Neuron besteht aus einer Aktivierungsfunktion sowie den Gewichtungen. Eine Schicht eines Neuronalen Netzes besteht aus mehreren Neuronen [24, 25].

Training

Die zum Training genutzte Funktion wird als Loss-Funktion bezeichnet und im Folgenden als E_{loss} definiert. Die in der Arbeit genutzte Loss-Funktion der Neuronalen Netze ist die Kreuzentropie:

$$E_{loss}(p,l) = -\frac{1}{n_T} \sum_{j=1}^{n_T} l_j log(p_j) + (1 - l_j) log(1 - p_j) \tag{2.22}$$

n_T beschreibt die Anzahl der Datenpunkte, p_j die Klassifizierungsergebnisse und l_j die wahren Werte des Datenpunkts j. Im Rahmen des Trainings wird

versucht die Gleichung 2.22 zu minimieren. Damit spiegelt diese Funktion den Fehler zwischen Eingangs- und Ausgangsgröße wider. Im Falle einer Regression wird der MSE in abgewandelter Form als Teil der Loss-Funktion wie folgt definiert:

$$E_{loss}(r,r) = \frac{1}{n_T} \sum_{j=1}^{n_T} (r_j - r_j)^2 \tag{2.23}$$

Hierbei wird die Differenz zwischen den geschätzten und den wahren Werten bestimmt.

Zeitreihennetzwerke

Um Zeitreihen darstellen zu können, existieren unterschiedliche Architekturen von Neuronalen Netzwerken. In der Literatur werden unterschiedliche Arten von Rekurrenten Neuronalen Netzen (RNN) beschrieben. RNNs sind dabei Netze, welche bestimmte Rückkopplungen zulassen, um Informationen abhängig von Zeitpunkten darstellen zu können. Neben Time Dependent Neural Network (TDNN) und Long Short Term Memory (LSTM) ist das Nonlinear Autoregressive Exogenous Model (NARX) eine Netzarchitektur, um Zeitreihen darstellen zu können. Diese Netzarchitektur wird im Rahmen der Arbeit weiter betrachtet. Leontaritis und Billings [26] beschreiben dieses NARX als ein Netzwerk, das zeitverzögert Inputs und auch Outputs verarbeitet.

Backpropagation

Backpropagation beschreibt eine Methode, bei der die Gewichte abhängig von der vorangegangenen Iteration τ berechnet werden [27, 28]:

$$\omega^{\tau+1} = \omega^{\tau} + \Delta\omega^{\tau} \tag{2.24}$$

Dabei ist das Ergebnis der Gleichung die neue Gewichtung der Iteration τ, ω. Δ beschreibt dabei die Anpassung des Gewichtes aufgrund der verfügbaren Daten, um die Loss-Funktion zu minimieren.

Gradientenwahl

Um sicherzustellen, dass die Gewichtungen gegen ihren optimalen Wert konvergieren, werden die klassischen Trainingskonzepte um einen Gradienten erweitert, welcher die Gewichtungsanpassungen ergänzt. Es existieren unterschiedliche Verfahren zur Bestimmung des Gradienten ([29–33]). Nach [27] sieht die Gewichtungsänderung wie folgt aus:

$$\Delta\omega_j^{\tau} = -\nu \frac{\sigma E_{loss}(NN_{\omega}(f_j), l_j)}{\sigma\omega^{\tau}} + \eta\Delta\omega^{\tau-1} \tag{2.25}$$

Dabei beschreibt η einen Parameter, welcher sich über die Iterationsstufen ändern kann und so die Konvergenz der Gewichtungen verbessert.

2.1.4 Transfer Learning

Das Konzept des Transfer Learning [16] beschreibt die Idee, ein bereits trainiertes Neuronales Netzwerk für eine neue Aufgabe zu verwenden, um das aufwendige und datenintensive Training eines Netzes von Grund auf zu vermeiden. Dabei wird das bereits erlernte Wissen (wie die trainierten Gewichtungen einer ähnlichen Aufgabe) auf ein ähnliches Problem angewendet und als Startpunkt genutzt. Es gibt jedoch bestimmte Voraussetzungen, die erfüllt sein müssen, damit Transfer Learning erfolgreich angewendet werden kann. Ziel dieses Ansatzes ist es, effizienter neue Aufgaben zu erfüllen. Die Ausführung von Transfer Learning kann je nach Anwendungsfall unterschiedlich erfolgen. Einerseits lässt sich ein ursprünglich trainiertes Netzwerk mit einem neuen Datensatz weiter trainieren. Andererseits kann auch ein bestehendes Netzwerk genutzt werden, um eine ähnliche Aufgabe zu erfüllen, für die es ursprünglich nicht trainiert wurde. Auch die Initialisierung der Gewichtungen kann variieren. Ein neu zu trainierendes Netzwerk kann mit den Parametern des ursprünglichen Netzes initialisiert und anschließend weiter trainiert werden. Alternativ können die Gewichtungen eines bestehenden Netzes festgesetzt werden und nur einzelne Teile des Netzwerks mit zufälligen Werten initialisiert und neu trainiert werden. In diesem Fall wird also lediglich ein Teil des Netzwerks angepasst.

2.1.5 Hyperparameteroptimierung

Bei der Implementierung von Neuronalen Netzen spielt die Wahl der richtigen Hyperparameter eine entscheidende Rolle für die Qualität der Ergebnisse, die von der KI erzielt werden können. Hyperparameter sind Parameter, die nicht vom Modell selbst gelernt werden, sondern bei der Implementierung des Netzwerkes manuell gesetzt werden. Beispiele für Hyperparameter bei einem Neuronalen Netzwerk sind die Anzahl der Neuronen pro Schicht oder die Anzahl der Schichten, sowie Trainingsparameter wie die Trainingsrate. Die Herausforderung ist dabei, dass die optimale Zusammenstellung der Parameter nicht im Vorhinein bekannt ist, sondern durch eine Versuchsstrategie bestimmt werden muss. Eine Möglichkeit, die Hyperparameter zu optimieren, ist die Rastersuche. Dabei werden alle möglichen Kombinationen der Parameterwerte im Hyperparameterraum ausprobiert und das Modell anhand einer Bewertungsmetrik (z.B. Genauigkeit, F1-Score) beurteilt. Für kontinuierliche Parameter wird eine feste Anzahl von Auswertungspunkten festgelegt, um den Parameterraum abzudecken. Eine andere Möglichkeit ist die manuelle Suche, bei der die Parameterwerte iterativ angepasst und das Modell jeweils neu trainiert und ausgewertet werden. Hierbei geht es darum, vielversprechende Bereiche im Parameterraum zu identifizieren und schrittweise zu optimieren [34]. In Abbildung 2.1 wird der Unterschied zwischen manueller Suche und der Rastersuche visualisiert. Hier wird gezeigt, dass mit der gleichen Anzahl von Auswertungen der sensitive Parameter durch die Zufallssuche häufiger abgetastet wird. Das bedeutet, dass bei der manuellen Suche der sensitive Parameter im Vergleich zur Rastersuche öfter getestet wird, was zu besseren Ergebnissen führen kann [34]. Allerdings besteht bei der manuellen Suche auch die Gefahr, dass

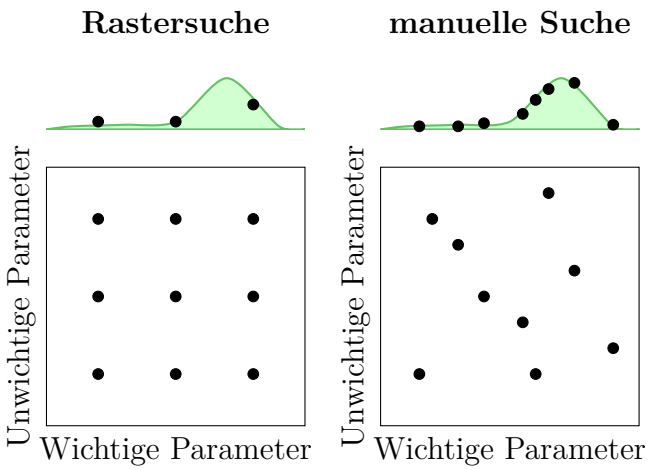


Abb. 2.1: Vergleich zwischen manueller Suche und Rastersuche nach Trouvain et al. [35]

die Suche in Bereichen des Parameterraums stattfindet, die keine vielversprechenden Lösungen enthalten. In diesem Fall kann es sinnvoll sein, die Suche durch einen anderen Algorithmus zu ergänzen oder zu ersetzen. Neben diesen einfachen Optimierungsmethoden sind auch andere Algorithmen zur Wahl der Hyperparameter in der Literatur bekannt ([36, 37]). Diese Methoden sind jedoch oft komplexer und erfordern mehr Rechenressourcen als die zuvor genannten Ansätze. Die Wahl der Optimierungsmethode hängt somit von der Komplexität des Problems, aber auch von den verfügbaren Ressourcen ab. Im Rahmen dieser Arbeit werden dabei zwei Ansätze kombiniert, die manuelle Suche sowie die Rastersuche.

2.1.6 Explainable Artificial Intelligence

Künstliche Intelligenz bringt das Problem mit sich, dass man den Entscheidungsweg, den ein Neuronales Netz wählt, nur sehr schwer nachvollziehen kann. Den Ansatz, diesen Entscheidungsweg nachzuvollziehen, beschreibt Holzinger [38]. Dabei wird das Ziel verfolgt, die Ergebnisse eines Neuronalen Netzes erklärbar zu machen. Beispielsweise existieren Ansätze, in Bildverarbeitungsnetzen Pixeln Gewichtungen zuzuordnen, die beschreiben, wie stark ein Pixel zu einem Klassifizierungsergebnis beiträgt [39]. Dabei werden unterschiedliche Arten der Erklärbarkeit von Systemen unterschieden.

1. Die quantitative Analyse von Netzwerken kann betrachtet werden.
2. Die Grundsätzliche Architektur eines ML-Systems kann analysiert werden.

Quantitative Ansätze der Erklärbarkeit von ML-Systemen

Im Allgemeinen beinhalten diese Ansätze eine Reduktion des ursprünglichen Modells zu einem leichter verständlichen Modell [40]. Dabei wird das komple-

xe Modell f durch ein approximierendes Modell g erklärt. Lokale Methoden versuchen, Vorhersagen des ursprünglichen Modells in einem lokalen Punkt $f(x)$ vorherzusagen. Der Input ist dabei nur ein einziger Datenpunkt. Neben dem vereinfachten Modell g werden auch vereinfachte Inputs x' benutzt. Der Zusammenhang zwischen x und x' wird dabei durch die Funktion $x = h(x')$ beschrieben. Damit erreichen lokale Methoden eine annähernd gute Approximation $y(z') = f(h(z'))$, solange $z' = x'$ gilt. Existierende Frameworks dieses Ansatzes sind LIME [41], Shapley [42] und SHAP [43]. Im Rahmen dieser Arbeit wird SHAP genauer betrachtet und verwendet.
SHAP [42, 43] ist ein Framework, welches eine Annäherung eines ursprünglichen Modells f findet, sodass gilt $g(x') \cong f(x')$:

$$f_x(z') = f(h(z')) = E[f(z) \mid z_S] \tag{2.26}$$

wobei S die Menge der Indizes in der Menge z' ist, welche ungleich null sind. SHAP-Werte bieten die Möglichkeit, den Erwartungswert eines vereinfachten Modells g in Abhängigkeit von vereinfachten Inputs x' zu bestimmen. Die Reduktion des Inputs wird über eine Mappingfunktion h erreicht. Diese Mappingfunktion löst ebenso die Abhängigkeit von Modellen, mit fehlenden Inputs umzugehen, indem $f(z)$ mit $E[f(z) \mid z_s]$ gilt, wobei $z_s = h_x(z')$, möglich ist. Dadurch wird das Problem fehlender Inputs gelöst. Die exakte Errechnung der SHAP-Werte wird durch Kombination unterschiedlicher Methoden erreicht. Diese Zusammenhänge sind in folgenden Gleichungen definiert:

$$\begin{aligned} h_x(z') &= E[f(z) \mid z_S] && (2.27)\\ &= E_{z_s|z_s}[f(z)] && (2.28)\\ &\approx E_{z_s}[f(z)] && (2.29)\\ &\approx f([Z_s, E[z_S]]) && (2.30) \end{aligned}$$

Um eine genauere Modellabschätzung zu erreichen, wird generell ein höherer Rechenaufwand erwartet.

Qualitative Konzepte der Erklärbarkeit von ML-Systemen

Hybride ML-Architekturen Hybride Architekturen kombinieren datengestützte Lernverfahren mit symbolischen, regelbasierten oder modellbasierten Bausteinen, um sowohl Leistungsfähigkeit als auch Nachvollziehbarkeit zu erreichen. Das Ziel ist es, die letztgültige Entscheidung von Komponenten zu treffen, deren Funktionsweise explizit und menschenverständlich ist [44]. Dies bedeutet, dass ein lernendes Netzwerk strukturierte, semantische Zwischenrepräsentationen erzeugt, über die eine nachgelagerte, explizit begründbare Logik entscheidet. Dadurch wird der Erklärpfad zweistufig: Zum einen kann die regelbasierte Ebene offengelegt werden: welche Bedingungen, Ziele oder Nebenbedingungen geben den Ausschlag. Zum anderen können die Ergebnisse der Neuronalen Netze beschrieben werden.

Ein verbreitetes Muster in der Robotik oder dem autonomen Fahren ist die Kopplung von *ML* und *Reasoning* ([45–47]).

Systemische Designprinzipien für Erklärbarkeit von ML-Systemen

Die Erklärbarkeit oder Transparenz eines ML-Systems besitzt unterschiedliche Einflüsse: Erstens ist *Modularität und Separation of Concerns* zentral: Kleine und unterteilte Komponenten mit definierten Schnittstellen begrenzen die Komplexität und erlauben Erklärungen. Zweitens sollte der *Nachvollziehbare Informationsfluss* als Architekturziel behandelt werden. Jede Inferenz ist trennbar und nachvollziehbar [48]. Drittens braucht es *menscheninterpretierbare Zwischenrepräsentationen*: Zwischenergebnisse von ML-Systemen müssen semantisch eine interpretierbare Form besitzen. Vierter Einfluss ist *Human-in-the-Loop*. In ML-Systemen, in denen Risiken hoch sind, kann die Erklärbarkeit von Systemen durch Zwischenentscheidungen von Menschen integriert werden [49].
Modularität und erklärende Zwischenschichten können Effizienz kosten, erhöhen jedoch Robustheit und Wartbarkeit.

2.2 Korrosionstheorie

Korrosion ist ein Prozess der Umwandlung von metallischen Bauteilen aufgrund einer Reaktion mit ihrer Umgebung [4]. Diese Reaktion kann als oxidative oder reduktive Reaktion ablaufen und zu Veränderungen der chemischen und physikalischen Eigenschaften des Materials führen.
Inhalt dieses Kapitels sind die elektrochemischen Grundlagen der Korrosion. Ferner wird die verwendete Methodik beschrieben, um Validierungsdaten zu Korrosionsverläufen zu erzeugen. Abschließend werden korrosionsvermeidende Maßnahmen, die im Rahmen dieser Arbeit in die Bestimmung des Korrosionsrisikos integriert werden, erläutert.

2.2.1 Korrosionsprozess

Elektrochemisches Korrosionsverhalten

Nach DIN EN ISO 8044 [50] wird Korrosion als eine physikochemische Wechselwirkung zwischen einem Metall und seinem Umfeld betrachtet. Diese Wechselwirkung führt zu einer Veränderung der Eigenschaften eines Bauteils und kann die Funktion des Metalls, der Umgebung oder des technischen Systems erheblich beeinträchtigen. Korrosion kann jedoch durch geeignete Maßnahmen verlangsamt werden.
Im Rahmen dieser Arbeit wird ausschließlich die galvanische Korrosion betrachtet. Sie ist eine Form der Korrosion, die auftritt, wenn sich ein unedleres Metall in direktem Kontakt mit einem edleren Metall befindet und in Anwesenheit eines Elektrolyten ein Elektronenfluss stattfindet. In diesem Fall fungiert das edlere Metall als Kathode, während das unedlere Metall als Anode dient. Die Korrosionsrate hängt von verschiedenen Faktoren ab, wie beispielsweise

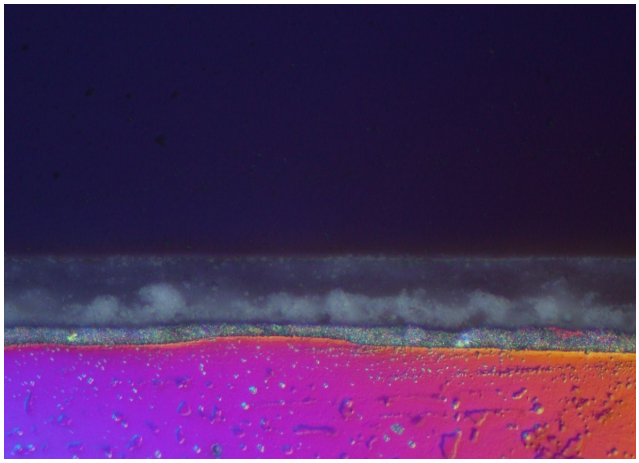


Abb. 2.2: Schliffbild an der Oberfläche eines Bleches [7]. Die KTL-Schicht ist als oberste Schicht in grau zu erkennen. Die Schichtdicke beträgt 30 µm, die Zinkschicht 7 µm.

der Differenz der Ruhepotentiale, dem Polarisationsverhalten oder dem Verhältnis der Anoden- zur Kathodenfläche [50, 51]. Im Automobilbau wird die galvanische Korrosion gezielt ausgenutzt, indem eine Zinkschicht auf die Stahlbauteile aufgebracht wird. Diese Zinkschicht dient als Opferanode und schützt somit das Grundsubstrat. Nach dem Zusammenbau der Rohkarosserie wird die Karosserie durch ein Tauchbad hindurchgeführt, um eine nicht-metallische Beschichtung namens Kathodische Tauchlackierung (KTL) auf den Bauteilen abzuscheiden. Diese Beschichtung bietet zusätzlichen Schutz vor Korrosion. Es ist jedoch wichtig zu beachten, dass z.B. Flansche damit weiterhin ungeschützt bleiben, was zu einem erheblichen Korrosionsrisiko führen kann [9, 50]. Im Folgenden werden die korrosionskritischen Konstruktionen genauer beschrieben.

Korrosionskritische Konstruktionen

Oberfläche Die Oberfläche von Bauteilen ist im Kontext der Korrosionsentstehung der risikoärmste Bauteilbereich. Durch die KTL wird die Oberfläche ausreichend geschützt.

In Abbildung 2.2 ist ein Schliffbild der Oberfläche eines Bauteils gezeigt. In diesem Beispiel wird auf ein mit zinkbeschichtetem Stahlblech (Dicke $\geq$7 µm) eine Tauchlackierung aufgetragen, welche eine ungefähre Schichtdicke von 25 µm aufweist. Dieser Aufbau ist jedoch nur an Oberflächen ohne Störfaktoren realisierbar. Sowohl an Bauteilkanten als auch an Flanschen sind diese Schichtdicken der KTL nicht möglich [8].

Bauteilkanten Die Herstellung der meisten Bauteile der Rohkarosserie erfolgt durch Schneid-Stanzen von verzinkten Blechen. Hierbei wird der Werkstoff durch das Eindringen des Schneidwerkzeugs plastisch verformt, was auf

der Oberseite zu einer abgerundeten Kante führt. An der Unterseite entsteht dabei eine Rissbildung, die das Material ab einer bestimmten Schnitttiefe bricht und zu einem Schnittgrat an der Kante führt. [52]. Die Seite des ausgestanzten Bleches, die auf der Seite des angesetzten Schnittes liegt, ist eine korrosionskritische Schwachstelle. Dies liegt an der fehlenden Zinkschicht an der Schnittkante. Außerdem ist die KTL-Schicht an Bauteilkanten im Vergleich zu der Bauteiloberfläche geringer, da die Oberflächenspannung der KTL an Kantenecken eine dickere Schicht verhindert. Dadurch verringert sich die KTL-Schichtdicke in diesen Bereichen.
Kantentypen werden in drei unterschiedliche Gruppen unterteilt. Diese Kantentypen kommen in Prüfkörpern und Bauteilen vor:

1. Abgerundete Kanten:
   Diese Kante liegt auf der Seite, auf welcher das Stanzmesser eindringt. Abgerundete Kanten weisen das geringste Korrosionsrisiko auf, da durch die Form der Kante eine dickere KTL-Schicht ausgebildet wird. Diese Art der Kanten wird nicht weiter betrachtet.

2. Rechtwinklige Kanten:
   Durch manuelles Schleifen von abgerundeten oder spitzen Kanten werden rechtwinklige Kanten hergestellt. Diese Bauteilkanten werden in den Prüfkörpern betrachtet. Des Weiteren sind die Ergebnisse vergleichbar mit denen von gestanzten Kanten [7] und werden im Rahmen dieser Arbeit als äquivalent betrachtet.

3. Spitze Kanten:
   Diese Kante liegt auf der Austrittsseite des Stanzmessers. Das Kantenbild entsteht durch das Überstehen eines Grats. Diese Art der Kante weist eine besonders geringe Dicke der KTL-Schicht auf und zeigt damit ein deutlich erhöhtes Korrosionsrisiko. Diese Kante kommt in Realbauteilen vor.

Flansch Gemäß Wünsche [8] bezeichnet der Begriff Flansch eine Verbindung zwischen zwei Blechbauteilen. Eine Bauteilverbindung kann mit unterschiedlichen Fügeverfahren gebildet werden.
Bei Flanschen kann die KTL nicht in den Spalt zwischen den Bauteilen eindringen. Als Folge sind die Bereiche der verbundenen Bleche nur durch das Zink und nicht durch eine KTL-Beschichtung geschützt, was zu besonderer Anfälligkeit gegenüber Korrosion führt [11].

2.2.2 Korrosionstest

Experimentelle Korrosionstests

Der Korrosionsprozess kann durch einen Korrosionstest gerafft werden. Der in dieser Arbeit verwendete Korrosionstest ist in Kapfer [7] und Schneider [53]

beschrieben. Korrosionstests sind in der Entwicklung von Automobilen wichtige Werkzeuge bei der Absicherung von Bauteilen und Gesamtfahrzeugen hinsichtlich der Korrosionsvermeidung. Die Beständigkeit gegen Korrosion wird in zeitlich gerafften standardisierten Verfahren geprüft. Diese standardisierten Tests ermöglichen es, das Verhalten von Bauteilen und Werkstoffen unter realistischen Bedingungen reproduzierbar zu simulieren und dadurch mögliche Schwachstellen in der Materialauswahl oder in der Geometrie zu identifizieren. Die Ergebnisse dieser Tests tragen dazu bei, Korrosionsschutzmaßnahmen zu entwickeln und zu verbessern.

Korrosionstestauswertung

Die bisherige Auswertung der Ergebnisse der Kantenkorrosionstests wird ausführlich von Kapfer [7] beschrieben. Schneider [53] erklärt detailliert die Flanschkorrosionsauswertung.

Kantenkorrosion Die Auswertung der Kantenkorrosion erfolgt über standardisierte Prüfbleche [7], die in den Wochen 1-8 sowie nach Woche 12 und 16 ausgewertet werden. Als beschreibende Variable für die Kantenkorrosion wird die Delaminationsweite gewählt. Sie beschreibt die Unterwanderung der KTL-Beschichtung, ausgehend von der Schnittkante. Dabei werden als Einflussfaktoren auf diese Delaminationsweite $d_B(e;\alpha;t)$ die Prüfzeit t, die Kantenklassen e und der Neigungswinkel α berücksichtigt. Nach Kapfer [7] werden die Prüfkörper zur Reproduzierbarkeit mit rechtwinkligen Kanten geprüft. Die Evaluierung der Ergebnisse in Bezug auf Blechkanten wird über sogenannte Kantenklassen realisiert. Dabei wird der Neigungswinkel α der beschichteten Probebleche zur xy-Ebene bestimmt. Des Weiteren werden die Orientierungen der Kanten ermittelt und basierend darauf in Klassen unterteilt.
Nach jeder Woche werden mehrere Probebleche aus dem Korrosionstest entnommen, um die experimentellen Ergebnisse strukturiert auszuwerten. Kapfer bestimmt die Delaminationsweite an Bauteilkanten der Probebleche d_B, indem die Bleche nach dem Scannen durch einen deterministischen Ansatz binär segmentiert werden, um die delaminierten Bereiche zu identifizieren. Die Delaminationsweite d_B wird aus dem Binärbild als Verhältnis zwischen der delaminierten Fläche und der korrodierten Kantenlänge des Blechs berechnet.

Flanschkorrosion In seiner Studie untersucht Wünsche [8] die Auswirkungen verschiedener Faktoren auf die Korrosion von idealisierten Flanschen. Dabei werden die Spalthöhen, die Flanschbreiten und die Grundsubstrate variiert, während die Flansche stets unter einem konstanten Winkel zur xy-Ebene positioniert sind und durch ein Glasbauteil gebildet werden. Um die Flansche herzustellen, werden Platten aus unterschiedlichen Materialien in definierten Spalthöhen von $50, 100, 150, 200$ µm auf Probebleche positioniert. Als Grundsubstrate kommen sowohl blanker Stahl (SB) als auch feuerverzinkter Stahl (GI) zum Einsatz, während die Flanschbreiten $20, 30$ und 50 mm betragen, was der Breite des aufgetragenen Spaltbildners entspricht. Schneider [53] verändert

dieses Vorgehen durch die Variation weiterer Parameter. Einerseits wird neben der Flanschhöhe die Flanschbreite variiert. Andererseits wird der Einfluss des Spaltbildners durch Variation von Materialien berücksichtigt. Diese Materialien sind unverzinkter Stahl, Aluminium, feuerverzinkter Stahl sowie PMMA. Im Rahmen der Untersuchung werden unterschiedliche Stähle betrachtet. [53] erkennt keinen relevanten Unterschied in dem Korrosionsverhalten von unterschiedlichen höherfesten Stählen wie beispielsweise CR300-LA und CR-5.
Die aus diesen Versuchen resultierenden Zeitreihen dienen als Grundlage zur Validierung von Simulationsverfahren von Wünsche [8] und Schneider [53].

2.2.3 Korrosionssimulation

Der Kantenkorrosionssimulation nach Kapfer [7] und der Korrosionssimulation nach Schneider [53] liegen eine FEM-Simulation zugrunde. Sie wird an in Experimentalreihen erzeugten Datenpunkten validiert. Diese Experimentalreihe wurde in Kapitel 2.2.2 vorgestellt. Die Simulationsmodelle für Kanten und Flansche unterscheiden sich durch die gewählten Randbedingungen und konstruktiven Zusammenhänge (Tabelle 2.2). Neben der Definition der geometrischen Form, die Kante und Flansch voneinander abheben, unterscheiden sich die Rahmenbedingungen des simulierten Bereichs. Einerseits ist die Sauerstoffzufuhr in einem Flansch eingeschränkt, andererseits spielt die Kathodische Tauchlackierung durch die fehlende Abscheidung im Flansch keine Rolle. Einen weiteren Unterschied stellt der Einfluss der räumlichen Orientierung der Kanten dar. Durch Wassereinflüsse unterscheidet sich das Korrosionsverhalten von horizontalen und vertikalen Kanten. Aufgrund dieses Umstandes wird für jede Kantenklasse eine eigene Kantenkorrosionssimulation umgesetzt.

Einflussfaktor	**Kante**	**Flansch**
Simulationsmodell	FEM-basierte Simulation	FEM-basierte Simulation
Geometrische Unterschiede	Rechtwinklige Kante	Bauteilverbindung mit spezifischer Flanschbreite
Sauerstoffzufuhr	Frei verfügbar	Eingeschränkt verfügbar
Kathodische Tauchlackierung	Einfluss über Phasengrenzbedingung vorhanden	Kein Einfluss aufgrund von fehlenden Abscheidungen im Flansch
Räumliche Orientierung	Signifikanter Einfluss durch Kantenklasse, sowie Winkel α	Keine Unterscheidung
Klimatische Einflüsse	Berücksichtigung, ermittelt aus zyklischem Wechselklimatest	Berücksichtigung, ermittelt aus zyklischem Wechselklimatest

Tabelle 2.2: Vergleich der Randbedingungen der Korrosionssimulation von Kanten und Flanschen

2.2.4 Korrosive Schutzmaßnahmen

Im Rahmen der Korrosionsvermeidung existieren unterschiedliche Methoden. Diese lassen sich in konstruktive und nicht-konstruktive Maßnahmen unterteil-

len. Nicht-konstruktive Maßnahmen bezeichnen im Kontext der Korrosion die bereits beschriebene Verzinkung und die Lackierung des Rohbaus. Im Rahmen dieser Arbeit werden drei konstruktive Maßnahmen zur Korrosionsvermeidung betrachtet.

1. Nahtabdichtung (NAD)
2. Hohlraumkonservierung (HRK)
3. Klebstoffe

Jeder Datensatz liegt im Rahmen dieser Arbeit in einer tessellierten Form vor. Die Dokumentation der Maßnahmen geschieht dabei über Geometrien. Diese Geometrien werden über tessellierte Oberflächen definiert, das heißt, die Oberflächen werden über einzelne 2D-Elemente beschrieben. Im Rahmen dieser Arbeit sind diese 2D-Elemente Dreiecke.

Nahtabdichtungen

Die Nahtabdichtung im Automobilbau bezieht sich auf die Versiegelung von Fugen und Nähten in der Karosserie, um Wasserzutritt zu verhindern. Diese Versiegelung wird durch das Aufbringen spezieller Dichtungsmaterialien, wie zum Beispiel Polyvinylchlorid (PVC), erreicht. Sie schützt die metallischen Komponenten vor oxidativen Umwelteinflüssen. In der industriellen Produktion wird die Nahtabdichtung meist automatisiert durch Roboter aufgetragen. An einzelnen Stellen kann auch manuell aufgetragen und verteilt werden [54]. Die Realapplikation am Fahrzeug ist in Abbildung 2.3 dargestellt.



Abb. 2.3: Eine aufgetrennte komplexe Geometrie eines Realfahrzeugs. In Grau sieht man die nicht korrosionskritische Oberfläche. Die Realapplikation der NAD am Bauteil ist in hellerer Farbe sichtbar.

Die Informationen zu den Nahtabdichtungskonstruktionen liegen ebenso in tesselierten Datensätzen vor. Daraus kann ein Datensatz generiert werden.

Eigenschaften dieses Datensatzes sind die Form der tesselierten Daten (Abbildung 2.4). Dabei beschreiben die tesselierten Geometrien die Bauräume, welche durch das PVC ausgefüllt werden.

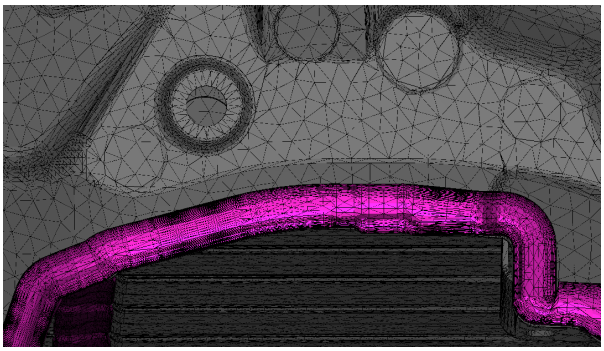


Abb. 2.4: Visualisierung eines tesselierten NAD-Datensatzes in pink, zusammen mit einem Rohbaudatensatz in grau. Die Tesselierung des Datensatzes ist in schwarz sichtbar.

Hohlraumkonservierung

Diese konstruktive Maßnahme beschreibt den Prozess, bei dem ein viskoses Wachs in Hohlräume eingebracht wird und in Flansche penetriert. Das Wachs wird durch Düsen auf das jeweilige Bauteil appliziert. Alternativ können Bauteile durch sogenanntes Wachsfluten geschützt werden. Dabei werden Bauteilverbindungen einseitig geschlossen, mit Wachs geflutet und abschließend wieder entleert. Die Realapplikation am Fahrzeug ist in Abbildung 2.5 dargestellt.



Abb. 2.5: Realapplikation von Wachs auf einem Bauteil. In grau die Bauteiloberfläche, in Gelb das applizierte Wachs. Die Abbildung zeigt im unteren Teil einen geöffneten Flansch, in den das Wachs penetriert ist.

Wie bei NAD liegt die Hohlraumkonservierung in tesselierten Daten als einzelner Datensatz vor. Abbildung 2.6 zeigt den tessellierten Datensatz in Blau

gemeinsam mit einem Rohbaudatensatz. Dabei existieren zwei unterschiedliche Arten, Wachs zu dokumentieren. Die erste Art ist über Röhren. Die zweite Art, die Wachsapplikation zu dokumentieren, ist über Flächen. Dabei werden die Bereiche, die wie in Abbildung 2.5 durch Wachs bedeckt sind, markiert. Dazu werden die bedeckten Flächen durch einen Quader markiert, der die geschützte Oberfläche einfasst. Beide Arten sind in der Visualisierung des Datensatzes zu sehen. Die rechts Blau markierten Bereiche sind Flächen, um die ebenfalls Röhren gelegt wurden.

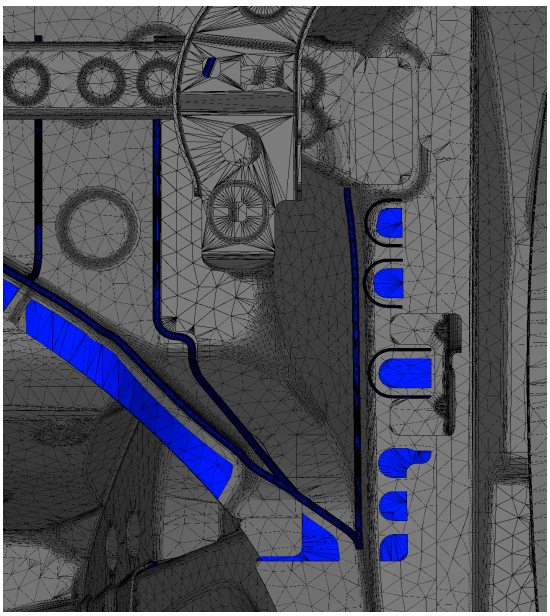


Abb. 2.6: Visualisierung eines tesselierten Wachs-Datensatzes in blau, zusammen mit einem Rohbaudatensatz in grau. Im Bild sichtbar sind Wachsbauteile als Röhrchen und als Fläche.

Klebstoff

Klebstoffe werden in der Rohbaukonstruktion in unterschiedlichen Funktionen eingesetzt. Klebstoffe können als Klebung in der Struktur bei tragenden Verbindungen eingesetzt werden. Unterfütterungsklebungen können zum Anbinden von Versteifungsprofilen genutzt werden. Außerdem wird Klebstoff bei Falzen eingesetzt. Dies sind Verbindungen von Dünnblechen, bei denen das Außenblech um 180° umgefalzt wird [55]. Im Kontext des Korrosionsschutzes wird Klebstoff in Flansche eingebracht, füllt es Flanschräume und wirkt korrosionsvermeidend. Klebstoffe, die strukturelle Funktionen besitzen, werden als für den Korrosionsschutz nicht relevant betrachtet. Ebenso wie bei den beiden vorangegangenen Maßnahmen kann ein tessellierter Datensatz generiert werden. Die Realapplikation am Fahrzeug sowie das digitale Format sind in Abbildung 2.7 und Abbildung 2.8 dargestellt.



Abb. 2.7: Realapplikation des Klebstoff am Bauteil. In Rot ist der Klebstoff nach der Verpressung zu sehen. Appliziert wird er in kleinen Röhrchen.

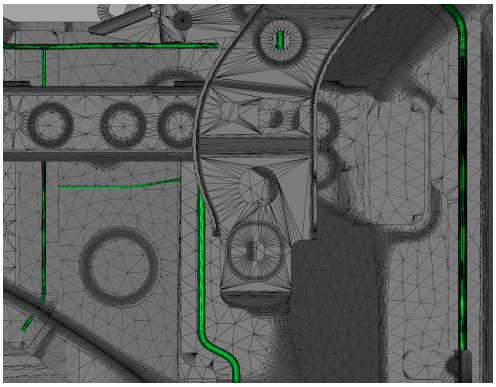


Abb. 2.8: Digitale Dokumentation des Klebstoffes. In Grün sind die Klebstoffbauteile sichtbar, in Grau der digitale Datensatz eines Fahrzeugs. Die schwarzen Linien zeigen die Tesselierung beider Datensätze.

2.2.5 Einflüsse des Materials auf das Korrosionsverhalten

Revie [56] erörtert die verschiedenen Korrosionsverhaltensweisen von Metallen. Der Materialeinfluss ist ein wesentlicher Faktor für das Korrosionsverhalten. Außerdem spielen auch galvanische Effekte durch das Verbinden von unterschiedlichen Stählen eine Rolle [57].

Passivierungseigenschaften

Diese Reaktionen führen dazu, dass sich stabile Verbindungen oder Oxidschichten bilden, die das darunterliegende Metall vor weiterer Korrosion schützen. Die Bildung von weniger reaktiven Schichten, sogenannten Passivierungsschichten, erfolgt in der Regel durch chemische Reaktionen zwischen der Metalloberfläche und der umgebenden Atmosphäre oder dem Korrosionsmedium. Die Art der entstehenden Schicht hängt stark von der chemischen Zusammensetzung des Metalls sowie von den spezifischen Bedingungen der Umgebung ab, wie z.B. des pH-Werts, der Temperatur und des Vorhandenseins von ag-

gressiven Chemikalien [58].
Der pH-Wert des Korrosionsmediums hat einen signifikanten Einfluss auf die Bildung und Stabilität von Passivierungsschichten. In sauren oder alkalischen Umgebungen können sich die chemischen Reaktionen an der Metalloberfläche verändern, was zu unterschiedlichen Arten und Eigenschaften von Passivierungsschichten führen kann. In sauren Medien können beispielsweise saure Reaktionsprodukte die Passivierungsschicht destabilisieren und zu einer beschleunigten Korrosion führen. Die Kenntnis des pH-Werts der Umgebung ist daher entscheidend für die Vorhersage des Korrosionsverhaltens von Metallen und die Auswahl geeigneter Korrosionsschutzmaßnahmen [59, 60].

Einfluss von Potentialdifferenzen bei der Galvanischen Korrosion

Die Stärke des galvanischen Effekts [4] wird durch die Potentialdifferenz zwischen den beiden beteiligten Metallen bestimmt. Je größer die Potentialdifferenz ist, desto größer ist der Elektronenfluss und desto schneller verläuft die Korrosion des anodischen Metalls. Diese Potentialdifferenz wird durch die elektrochemischen Eigenschaften der Metalle bestimmt, einschließlich ihrer elektronischen Struktur und ihrer Reaktivität in der jeweiligen Umgebung. Die Auswahl von Metallpaarungen mit ähnlichen Potentialen kann dazu beitragen, galvanische Effekte zu minimieren und Korrosion zu reduzieren [61]. Alternativ können Trennschichten oder Distanzierungen das Korrosionsverhalten beeinflussen.

2.3 Elektrolytbeanspruchung

In dem folgenden Kapitel werden die Grundlagen der Elektrolytbeanspruchung behandelt. Dies wird im speziellen Kontext der in dieser Arbeit betrachteten Korrosionsart, der galvanischen Korrosion, durchgeführt. Außerdem werden digitale Methoden beschrieben, um das Elektrolytverhalten zu simulieren.

2.3.1 Einfluss der Elektrolytbeanspruchung auf das Korrosionsverhalten

Die Elektrolytbeanspruchung spielt eine besondere Rolle in der Betrachtung der Korrosion [62].
In der Literatur wird der Begriff Benässung definiert. Nach DIN EN ISO 9223 [63] beschreibt die Benässung den Anteil der Zeit τ_i, in dem der betrachtete Bereich befeuchtet bleibt. Basierend auf diesem Anteil werden unterschiedliche Korrosionsrisikokategorien definiert.
Nach Pedefern [64] hat diese Benässung einen entscheidenden Einfluss auf das Korrosionsverhalten, da bei fehlender Benetzung Korrosion nur deutlich verringert stattfinden kann.

2.3.2 Lösungsmethodik der numerischen Strömungsmechanik

Ziel dieser Arbeit ist die Bestimmung des Korrosionsverhaltens eines Gesamtfahrzeugs auf Basis von digitalen Datensätzen. Soll der Einfluss von Elektrolyt integriert werden, so ergeben sich Anforderungen an die Datenquelle. Die Datenquelle muss strukturierte quantitative Daten beinhalten. Darüber hinaus muss sie ebenfalls auf Basis von digitalen Daten aufgebaut werden. Daher ergeben sich als Möglichkeit numerische Strömungssimulationen, welche im Kontext der Automobilentwicklung bereits existieren [65]. Dieser Ansatz ermöglicht das Beschreiben des Fluidverhaltens. Diese Methoden können genutzt werden, um die Elektrolytbeanspruchung einer komplexen Geometrie digital zu bestimmen. Die Eigenschaften dieser Stoffe werden durch Gleichungen basierend auf dem Massen-, Impuls- und Energiegehalt definiert. Dabei wird die Eigenschaft des Massenerhalts durch die Kontinuitätsgleichung beschrieben [66]. Die nach Batchelor [67] abgeleitete Kontinuitätsgleichung ergibt sich dadurch wie folgt:

$$\frac{\delta}{\delta t} + \frac{\delta}{\delta x}\rho * u + \frac{\delta}{\delta y}\rho * v + \frac{\delta}{\delta z}\rho * w = 0 \tag{2.31}$$

Die Impulserhaltung wird durch die Navier-Stokes-Gleichung [68] beschrieben. Innerhalb eines Systems werden einzelne Impulse zu einem Impulsstrom zusammengefasst und betrachtet. Analog zur Kontinuitätsgleichung kann damit auch die Navier-Stokes-Gleichung hergeleitet werden:

$$\rho(\frac{\delta v}{\delta t} + v(\nabla * v)) = -\nabla p + \mu * \nabla * v + f \tag{2.32}$$

Dies gilt unter der Annahme eines Kontinuums für das betrachtete Fluid. Dabei beschreibt die linke Seite die Fluidträgheit, während die andere Seite in Reihe die Druckveränderung, die Reibung unter Annahme von inkompressiblen Fluiden und externe Kräfte aufaddiert. Energieerhaltung beschreibt die letzte der Grundgleichungen. Unter der Annahme, dass sich die Gesamtenergie des Systems durch Bewegungs- und innere Energie beschreiben lässt, kann die Gesamtenergie wie folgt beschrieben werden:

$$\frac{\delta u}{\delta t} = -\frac{p}{\rho} * \nabla \vec{v} + \frac{\nabla * (k \nabla T)}{\rho} + \frac{1}{\rho}\phi \tag{2.33}$$

Dabei sind die drei Summanden der Gleichung die Druckänderung, die Wärmeleitung innerhalb des Fluids und die Dissipation, also die Energieänderung durch Umwandlung von mechanischer in innere Energie des Fluids.

2.3.3 Ergebnisse der numerischen Strömungsmechanik

Um den Einfluss der Elektrolytbeanspruchung in eine Korrosionsprognose integrieren zu können, wird eine Strömungssimulation nach Kapitel 2.3.2 herangezogen. Ergebnisse der Strömungssimulation lassen sich in unterschiedliche

Gruppen aufteilen. Dabei wird zwischen zeitabhängigen und akkumulierten Informationen unterschieden. Bei den zeitabhängigen Simulationsergebnissen wird betrachtet, ob in der Nähe eines Bereichs ein Partikel liegt. Bei den zeitunabhängigen Simulationsergebnissen werden die einzelnen Zeitpunkte der Simulation kumuliert und berechnet, in wie vielen der Zeitpunkte der Bereich benetzt war. Damit ergibt sich ein Wert, der wie die ToW aus Kapitel 2.3.1 eine Beschreibung der Elektrolytbeanspruchung darstellt.

2.3.4 Kapillarkraft

Die Kapillarkraft ist ein entscheidender Faktor für das Verständnis der Elektrolytbeanspruchung auf Oberflächen. Der Anstieg wird von der Oberflächenspannung, dem Kontaktwinkel, der Flüssigkeitsdichte, der Schwerkraft und dem Radius der Kapillare beeinflusst. Die Berechnung des Kapillaraufstiegs hilft abzuschätzen, wie hoch Feuchtigkeit aufgrund der Kapillarwirkung entgegen der Schwerkraft aufsteigen kann. Der Effekt tritt dabei auf, wenn sich die Grenzfläche einer Flüssigkeitsphase und einer Gasphase oder nicht mischbaren Flüssigkeitsphase in einer Kapillare oder einem Spalt befindet. Nach Kapfer [7] bilden Flansche ein Kapillar aus. In einer Wechselwirkung aus Adhäsion (Grenzflächenspannung zwischen den Phasen) und Kohäsion (Oberflächenspannung der Flüssigkeitsphase) lässt der Kapillareffekt die Flüssigkeitsphase um die Kapillarhöhe h aufsteigen. Der Kapillaraufstieg h wird mit der folgenden Formel berechnet:

$$h = \frac{2\gamma \cos(\theta)}{\rho g r} \tag{2.34}$$

wobei γ die Oberflächenspannung der Flüssigkeit, θ der Kontaktwinkel, ρ die Dichte der Flüssigkeit, g die Erdbeschleunigung und r der Radius der Kapillare sind.

2.4 Beschichtungsverfahren

Im Folgenden werden die Einflüsse, der in dieser Arbeit betrachteten Beschichtungsverfahren, im Hinblick auf das Korrosionsverhalten beschrieben. Dabei orientiert sich dieses Kapitel am Aufbau eines Lackierungssystems, welches in der Automobilindustrie standardmäßig verwendet wird (Abbildung 2.9). Das Decklacksystem kann aus einer unterschiedlichen Anzahl an Komponenten bestehen. Neben Klarlack und Basislack kann ein Füller ebenfalls Teil des Decklacksystems sein.
Es existieren unterschiedliche Vorbehandlungsverfahren für Stahloberflächen. Phosphatierung ist ein Verfahren zur Oberflächenbehandlung von Metallen, das darauf abzielt, eine dünne Phosphatschicht auf der Oberfläche zu erzeugen. Diese Schicht verbessert die Haftung und Haltbarkeit von nachfolgenden Beschichtungen wie Lacken oder Polymeren und dient gleichzeitig als Korrosionsschutz [70]. Schneider [53] betrachtet den Einfluss von unterschiedlichen

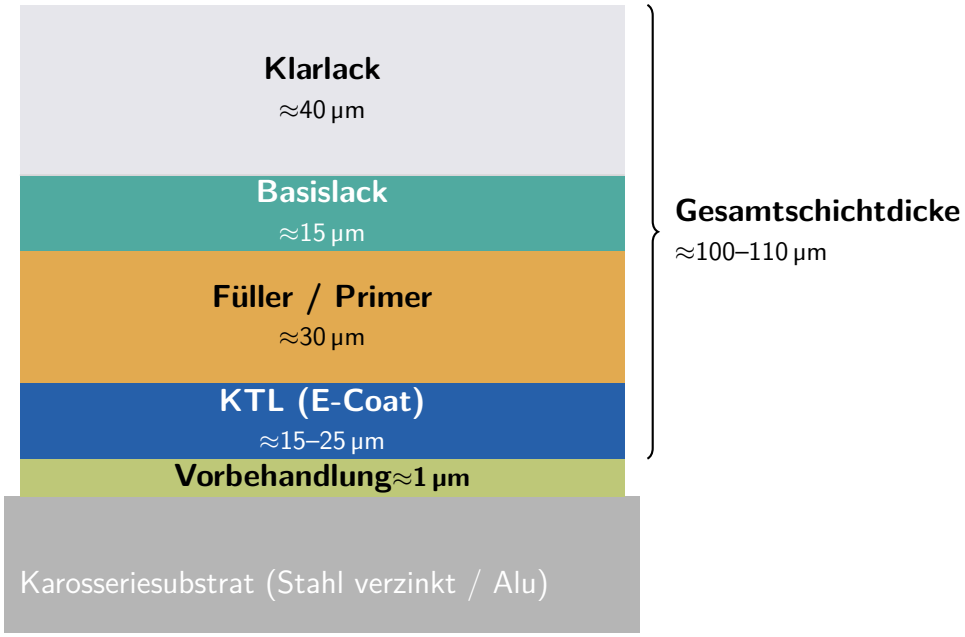


Abb. 2.9: Überblick eines Lackierungssystems nach Streitberger [69]

Beschichtungsverfahren und kommt zu dem Ergebnis, dass diese keinen Einfluss auf das Korrosionsverhalten haben.

2.4.1 Kathodische Tauchlackierung

Das im Hinblick auf das Korrosionsverhalten relevante Beschichtungsverfahren ist die kathodische Tauchlackierung. Dabei werden Konstruktionen aus mehreren Einzelbauteilen vollständig in ein Bad eingetaucht, welches neben Wasser mit Bindemittel und Pigmenten gefüllt ist, die zur Beschichtung auf die Karosserie mittels kathodischer Polarisierung aufgetragen werden. Diese korrosionsvermeidende Schicht wird durch einen Einbrennvorgang im Ofen abgeschlossen, bei dem sich die Pigmente in ein vernetztes Polymer wandeln [71].

Einflüsse der Kathodischen Tauchlackierung

Die Kathodische Tauchlackierung beeinflusst maßgeblich das Korrosionsverhalten von beschichteten Substraten. Durch die Bildung einer dichten und gleichmäßigen Beschichtung wird das Eindringen von Feuchtigkeit und aggressiven Medien auf die Metalloberfläche verhindert, was zu einer verbesserten Korrosionsbeständigkeit führt. Zudem bietet die Beschichtung einen mechanischen Schutz vor Kratzern und Abrieb, die Korrosion begünstigen [72].
Wünsche [8] betrachtet das KTL-Verhalten bei Flanschen. Flansche mit verschiedenen Spalthöhen werden im Hinblick auf die KTL-Benetzung und damit ihren Korrosionswiderstand bewertet. Ab einer Spalthöhe von 150 µm lässt sich zwar eine bessere und eine schlechter beschichtete Seite identifizieren, was auf die Geometrie des Prüfkörpers zurückzuführen ist. Jedoch ist der Großteil des Flanschbereichs beschichtet. Bei Spalthöhen kleiner 150 µm nimmt die

Beschichtung deutlich ab. Bei 50 µm liegt der Anteil der beschichteten Fläche zwischen 10 % und 30 %. Die Bereiche dieser nicht benetzten Fläche besitzen damit nicht mehr den oben beschriebenen Korrosionsschutz und bilden Korrosionsrisikostellen.

2.4.2 Decklacksystem

Lack ist ein flüssiges oder auch pulverförmiges Beschichtungsmaterial, das auf Oberflächen aufgetragen wird, um diese gegen Umwelteinflüsse zu schützen, das Aussehen zu verbessern oder die Oberflächenbeschaffenheit zu verändern. Lackschichten bestehen aus Bindemitteln, Pigmenten, Lösemitteln und verschiedenen Zusatzstoffen. Die Bindemittel sind für die Bildung des Lackfilms verantwortlich, während Pigmente Farbe und Deckkraft liefern. Lösemittel dienen der Einstellung der Viskosität für die Applikation und verdunsten nach dem Auftragen. Zusatzstoffe verbessern spezifische Eigenschaften wie die UV-Beständigkeit, Kratzfestigkeit oder die Trocknungsgeschwindigkeit. Das Decklacksystem besteht aus einer Klarlackschicht und einer Basislackschicht. Eine optionale Füllerlackschicht wird hier nicht weiter betrachtet [73].

Einflüsse des Lacksystems auf das Korrosionsverhalten

Basislacke und Decklacke sind Schichten in Mehrschichtlackierungen, insbesondere im Fahrzeugbau. Sie tragen ebenfalls zum Korrosionsschutz bei, da sie die primäre Barriere gegen Umwelteinflüsse wie Wasser, Salz, mechanische Einwirkungen und Chemikalien bilden. Die Zusammensetzung des Lacks, insbesondere die Art und Menge des Bindemittels sowie der Korrosionsinhibitoren, bestimmen seine Schutzwirkung. Studien haben gezeigt, dass epoxidharzbasierte Basislacke aufgrund ihrer chemischen Struktur und Adhäsionseigenschaften besonders wirksam im Schutz gegen Korrosion sind [74].

3 Forschungsstand der Korrosionsvorhersage im Bereich des Automobilbaus

Ziel dieses Kapitels ist es, den aktuellen Stand der automobilen Forschung im Bereich der Korrosionsvorhersage zu beschreiben.
Die Segmentierung eines Gesamtfahrzeugs mithilfe von Klassifizierungsmethodiken ist dabei das Hauptaugenmerk dieses Kapitels. Darüber hinaus wird die KI-basierte Kantenkorrosionsprognose von Waibel et al. [13] beschrieben.

3.1 Segmentierung korrosionskritischer Konstruktionen

Wie in Kapitel 2.2 beschrieben, werden im Rahmen der Korrosionsvermeidung strukturelle Schwachstellen identifiziert. Dies sind vor allem in der Rohbaukonstruktion existierende Kanten und Flansche.

Um eine Gesamtfahrzeugsegmentierung korrosionskritischer Konstruktionen durchzuführen, werden die geometrischen Daten dieser Konstruktion herangezogen. Der von Waibel [11] dazu implementierte Algorithmus wird nachfolgend erklärt.

3.1.1 Datenaufbereitung

Um geometrische Daten analysieren zu können, muss die Information über die Bauteilgeometrie transportiert werden. Dazu existieren unterschiedliche Formate, mithilfe derer diese Information gespeichert wird. Außerhalb der Mercedes-Benz AG Entwicklungswelt sind dies Formate wie *Stereo Lithography* (STL) [75], *Initial Graphics Exchange Specification* (IGES) [76] oder *Standard for the Exchange of Product Model Data* (STEP) [77].
Gespeicherte Geometrien unterscheiden sich dabei, in welcher Form sie abgespeichert werden. Bei einer diskreten, oder auch tesselierten Geometrie, wird die Oberfläche des Bauteils durch kleinere Flächenelemente beschrieben. Dies können Polygone, Dreiecke oder andere 2D-Formen sein. Kontinuierliche Volumenmodelle werden durch direkte oder indirekte Modellierung aufgebaut. Der Aufbau wird bei der direkten Modellierung durch die Zusammensetzung von einfachen Volumen und booleschen Operatoren durchgeführt. Bei der indi-

rekten Modellierung werden Körper durch die umgrenzende Fläche über Eckpunkte, Kanten und Flächen definiert.
Waibel [11] beschreibt die Segmentierung einer Oberfläche durch Klassifizierung jedes Dreiecks hinsichtlich seines Korrosionsrisikos. Dreiecke werden zweimal klassifiziert. Die erste Klassifizierung erkennt Oberflächenelemente, die Teil einer Kante sind. Die zweite Klassifizierung erkennt die Menge aller Oberflächenelemente, die Teil eines Flansches sind.
Die Product Lifecycle Management (PLM) Software der Mercedes-Benz AG ermöglicht einen Export von Geometriedaten in diskreter Form. Die Datenbereitstellung wird dabei ausgehend von einer Produktdatenbank und einem CAD-Programm gewährleistet.

Stereo Lithography Datenformat

Das im Rahmen der Segmentierung gewählte Datenformat STL beschreibt 3D-Geometrien über eine Aneinanderreihung von einzelnen Elementen. Das ASCII-Datenformat ist dabei wie folgt standardisiert:

solid *name*
$\left\{\right.$ **facet normal** $n_i n_j n_k$
outer loop
vertex $v1_x\ v1_y\ v1_z$
vertex $v2_x\ v2_y\ v2_z$
vertex $v3_x\ v3_y\ v3_z$
endloop
endfacet $\left.\right\}^i$
endsolid *name*

Das Textfeld nach ‘solid‘ und ‘endsolid‘ ist ein optionales Feld und kann bei Bedarf mit einem String gefüllt werden.

Geometriedatenkorrektur

Der oben beschriebene Export gewährleistet nicht, dass die Datensätze, die exportiert werden, korrekt sind. Daher wird der Datensatz aufbereitet. Die korrigierten Fehler sind:

1. Mehrere Bauteile können in einem STL-Datei-Block gespeichert werden. Zur Identifizierung von Flanschbauteilen werden einzelne Bauteile benötigt.

2. Die Dreiecksnetze werden bereinigt, indem fehlerhafte Elemente entfernt werden. Damit wird sichergestellt, dass jede Kante genau zwei Nachbardreiecke besitzt.

3. Die Normalenvektoren von Dreiecken eines einzelnen Bauteils müssen nach außen gerichtet sein. Dazu wird eine konsistente Normalenorientierung durch Anpassung der Knotenreihenfolge erreicht. Ein Dualgraph hilft bei den Orientierungsinformation. Die korrekte Ausrichtung wird durch Berechnung des vorzeichenbehafteten Volumens eines Bauteils überprüft.

3.1.2 Implementierung der Erkennung von Kantenflächen

Um eine Klassifizierung zur Kantenerkennung durchführen zu können, werden für jedes Dreieck Eigenschaften extrahiert. Auf Basis dieser Eigenschaften wird anschließend bewertet, ob Dreiecke Teil einer Kante sind. Die zu erkennenden Kanten sind in Abbildung 3.1 gezeigt. Hier ist eine schematische Beispielgeometrie im Schnitt gezeigt. In Grün sind die Kanten visualisiert.

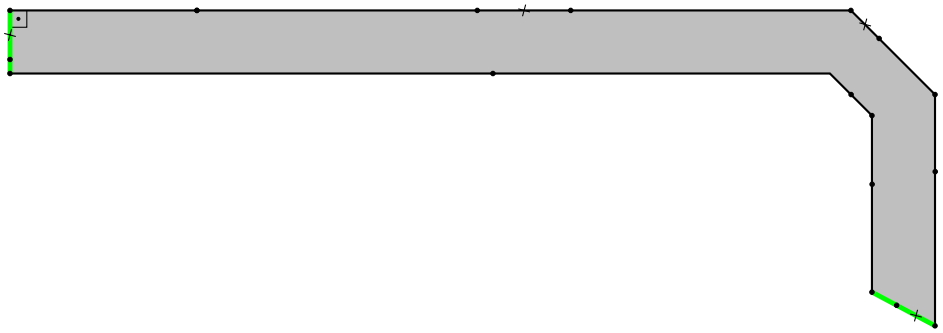


Abb. 3.1: Visualisierung einer Beispielblechgeometrie, übernommen von Waibel [11]. In Grün sind die Dreiecke markiert, die als Teil einer Kante klassifiziert werden sollen. Die schwarzen Puntke symbolisieren die Eckpuntke der Dreiecke, die Kreuze die Mittelpunkte.

Berechnung der Eigenschaften zur Kantenklassifizierung

Zur Ermittlung relevanter Dreieckselemente in der Umgebung eines Dreiecks t_i werden dessen Eckpunkte zunächst entlang der entgegengesetzten Normalenrichtung um einen bestimmten Wert verschoben, um eine erweiterte Axis-Aligned Bounding Box (AABB) zu erzeugen. Diese Box wird zusätzlich in jede Raumrichtung um einen berechneten Wert erweitert. Die relevante Umgebung von t_i wird dann bestimmt, indem die AABB mit den AABBs anderer Dreiecke t_j auf Überschneidungen überprüft wird. Überschneidungen identifizieren die Dreiecke t_j, die für die Klassifizierung von t_i als Kante relevant sind. Dieser Prozess nutzt einen AABB-Baum, um effizient Überschneidungen zu finden und zu analysieren. Die erweiterte AABB repräsentiert das lokale Volumen, das benötigt wird, um auf der Basis der nahegelegenen Dreiecke eine Klassifikation durchzuführen.

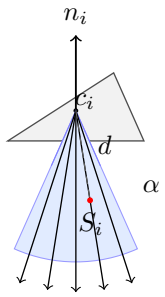


Abb. 3.2: Visualisierung der SDF. In Blau ist der Kegel hinterlegt, von dem die Strahlen aus dem Mittelpunkt c_i ausgehen.

Shape Diameter Function

Die durch die erweiterte AABB bestimmte Umgebung wird mit Hilfe eines Strahlenscans, der Shape Diameter Function (SDF) aus dem Schwerpunkt c_i des Dreiecks t_i analysiert. Die Strahlen S_i werden innerhalb eines Kegels mit einem Öffnungswinkel zwischen 0 und einem Winkel α in die entgegengesetzte Richtung der Normalen n_i emittiert (Abbildung 3.2). Es werden $p * q + 1$ Strahlen generiert, indem ein Vektor um die Achse aus $-n_i$ gedreht wird. Die Strahlen entstehen durch Rotationen in äquidistanten Winkeln aus einem Intervall und werden zusätzlich um den Winkel θ gedreht. Dies wird implementiert, um eine dichtere Verteilung im Zentrum des Kegels zu erreichen. Jeder dieser Strahlen wird weiter durch eine komplexe Rotation bestimmt (Rodrigues-Formel [78]), die vom Schwerpunkt c_i ausgeht. Das intensive Abtasten im Zentrum ermöglicht eine genauere Analyse der relevanten geometrischen Strukturen.
In der beschriebenen Methode werden die Strahlen S_i von einem Zentrumspunkt c_i innerhalb eines Kegels emittiert und auf Kollisionen mit den umliegenden Dreiecken t_j geprüft. Dies geschieht unter Nutzung des Möller-Trumbore-Algorithmus [79] auf einer GPU, was eine effiziente Berechnung potenzieller Schnittpunkte ermöglicht. Bei einem Schnittpunkt wird die euklidische Distanz zwischen dem Schnittpunkt und dem Schwerpunkt c_i des Dreiecks t_i als Strahlenlänge gemessen. Falls kein Schnittpunkt existiert, wird ein vordefinierter Defaultwert verwendet. Diese Distanzen werden pro Strahl aggregiert, wobei das Minimum, der Durchschnitt und das Maximum der Distanzen über alle Strahlen berechnet werden.

Flächenwinkel Ein weiteres Feature für die Kantensegmentierung sind die Flächenwinkel. Um die Flächenwinkel zwischen einem Dreieck t_i und seinen anliegenden Nachbardreiecken t_j zu berechnen, wird das Skalarprodukt der Normalenvektoren dieser Dreiecke benutzt. Die drei Winkel werden absteigend sortiert.

Dreieckshöhen Außerdem werden die Höhen der Dreiecke als Eigenschaften extrahiert. Die Berechnung der Höhen für ein Dreieck t_i wird in Abhängigkeit der Kantenlängen $l_{t_{ik}} k \in 1{,}2{,}3$ berechnet. Die Höhen für jede der drei Seiten

des Dreiecks lassen sich nach Gleichung 3.1 bestimmen. Die Dreiecksfläche A_i wird nach Köcher [80] berechnet:

$$h_{ik} = \frac{2A_i}{de_{ipq}}, \quad p = (k \mod 3) + 1, \; q = (k+1 \mod 3) + 1 \tag{3.1}$$

Machine Learning Klassifizierung

Die Architektur, die genutzt wird, um Dreiecke zu klassifizieren, ist das durch MATLAB implementierte PatternNet. PatternNet besteht aus mehreren vollständig verbundenen Schichten. Die Anzahl der Schichten und die Anzahl der Neuronen pro Schicht sind variabel und voll miteinander verbunden. In den verborgenen Schichten wird eine Sigmoid-Aktivierungsfunktion verwendet, während in der Ausgabeschicht eine Softmax-Funktion zum Einsatz kommt, die das Netzwerk befähigt, Wahrscheinlichkeiten für die Klassenzugehörigkeit zu generieren. Für das Training des Netzwerks wird der Backpropagation-Algorithmus verwendet [28]. Die Trainingsinputdaten werden normalisiert.

3.1.3 Implementierung der Erkennung von Flanschflächen

Die in Kapitel 2.2 beschriebenen Flansche werden durch Waibel [11] erkannt. Dabei implementiert er eine binäre Klassifikation, die Flansche erkennt. Diese Flansche sind in Abbildung 3.3 grün markiert. Waibel implementiert einen Ansatz, der Falze, Distanzierungen und Radien erkennt.

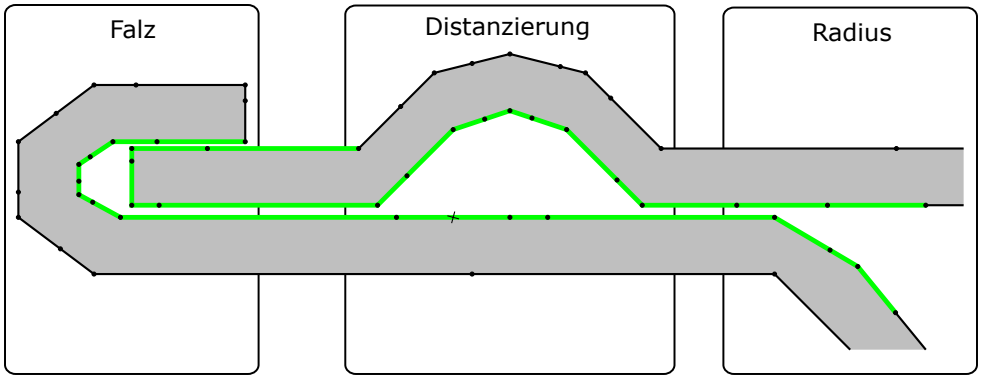


Abb. 3.3: Visualisierung der drei Arten der Flansche, welche nach Waibel [11] betrachtet werden. In Grün sind die als Teil eines Flansches zu klassifizierenden Dreiecke markiert. Schwarz sind Dreiecke, die nicht Teil eines Flansches sind.

Die Eigenschaftsberechnung für die Flanschklassifizierung von Bauteilen erfolgt unabhängig von spezifischen Baureihe und basiert auf der numerischen Darstellung problemspezifischer Eigenschaften, die die Unterscheidung der Dreieckselemente ermöglichen. Dies umfasst die Ausweitung der AABBs der Blattknoten um 5 mm in allen räumlichen Achsen, um relevante Bauteilpaare

zu identifizieren, die Überschneidungen aufweisen. Bauteilpaare ohne ausreichende Nähe (mindestens 5 mm Abstand [12]) werden von der Analyse ausgeschlossen. Die relevante Untersuchung erfolgt durch systematisches Überprüfen der erweiterten AABBs auf Überschneidungen mit denen anderer Dreieckselemente, um die Indizes überschneidender AABBs zu bestimmen.
Die Analyse konzentriert sich weiterhin auf die Segmentierung der relevanten Bereiche, indem die erweiterten AABBs auf Überschneidungen geprüft und nur diejenigen Dreieckselemente berücksichtigt werden, die sich in unmittelbarer Nähe befinden. Diese Vorgehensweise reduziert die Menge der zu testenden Dreieckselemente und die theoretische Laufzeit.

Berechnung der Eigenschaften der Flanschklassifizierung

Wenn für ein Dreieckselement t_{ai} keine Überschneidung mit irgendeinem Dreieckselement t_{bj} festgestellt wird, kann davon ausgegangen werden, dass kein Flansch vorhanden ist. In solchen Fällen werden für t_{ai} keine Eigenschaften extrahiert und es wird als $l_i = 0$ klassifiziert.
Für relevante Dreieckselemente t_{ai}, für die mindestens ein wie oben beschriebenes Dreieck t_{bj} gefunden werden kann, werden spezifische Eigenschaften für alle Dreieckselemente, die eine Überschneidung mit einem anderen Dreieck aufweisen, extrahiert. Diese Eigenschaften helfen dabei, die Struktur und Position von einem Dreieck t_{ai} relativ zu t_{bj} genauer zu bestimmen und sind entscheidend für die Klassifizierung und Analyse von potenziellen Flanschverbindungen. Diese Eigenschaften sind folgende:

1. Minimale euklidische Distanz: Diese Distanz wird zwischen den Dreieckselementen t_{ai} und t_{bj} bestimmt. Die Distanz wird als der kleinste räumliche Abstand zwischen beliebigen Punkten x auf t_{ai} und y auf t_{bj} definiert, und mathematisch durch folgende Formel beschrieben:

$$d_{ab_{ij}} = \min_{x \in t_{ai},\, y \in t_{bj}} \|x - y\|^2 \tag{3.2}$$

   Dabei folgt die Implementierung dem Algorithmus von Shellshear und Ytterlid [81].

2. Vektorprojektion: Mit diesem Feature wird die Projektion des Differenzvektors $(y - x)$ der Punkte x und y auf die Normale n_{ai} beschrieben. Die Punkte x und y sind diejenigen, die den minimalen euklidischen Abstand zwischen den Dreiecken t_{ai} und t_{bj} aufweisen, dargestellt durch die Formel:

$$d_{nab_{ij}} = (n_{ai})^T (y - x) \tag{3.3}$$

3. Normalenprodukt: Das Produkt der Normalen der zwei zu untersuchenden Dreiecken wird mit folgender Formel berechnet:

$$n_{nab_{ij}} = (n_{ai})^T n_{bj} \tag{3.4}$$

4. Projizierte Überlappung Dreiecksflächen: Zuerst wird das Dreieck t_{bj} auf

eine zweidimensionalen Ebene projiziert, die durch t_{ai} aufgespannt wird. Diese Projektion ergibt die Projektion $\hat{t}_{bj}$. Die Eckpunkte von $\hat{t}_{bj}$ werden durch die Formel:

$$\hat{v}_b(idjk) = v_b(idjk) - \hat{d}_{ab_{ijk}} n_{ai} \tag{3.5}$$

berechnet, wobei $\hat{d}_{ab_{ijk}}$ die Distanz des Eckpunktes des Dreiecks t_{bj} zu der Ebene ist, die durch t_{ai} definiert wird. Anschließend wird das Schnittpolygon, das die Schnittfläche von t_{ai} und t_{bj} repräsentiert, mittels des Sutherland-Hodgeman Clipping Algorithmus bestimmt [82]. Die Fläche $\hat{A}_{ij}^{ab}$ des Schnittpolygons wird mit der Shoelace-Formel berechnet [83]. Die tatsächliche prozentuale Schnittfläche A_{ij}^{ab} wird dann als Verhältnis der Fläche des Schnittpolygons zur Fläche des Dreiecks t_{ai} nach Gleichung 3.6 ermittelt.

$$A_{ij}^{ab} = \frac{\hat{A}_{ij}^{ab}}{A_i^a} \tag{3.6}$$

5. Strahlenlänge: Die SDF [17] wird angepasst, um Eigenschaften für die Klassifizierung zu extrahieren. Dabei wird für jedes Dreieck ein Satz von Strahlen aus dem Zentrum jedes relevanten Dreiecks in einem Kegel ausgesendet. α_{cone} beschreibt den Kegel, in dem die Strahlen ausgesendet werden, orientiert an der Richtung der Dreiecksnormalen. Auch die Anzahl der ausgesandten Strahlen wird definiert (p_4, p_5), die innerhalb des durch α_{cone} definierten Kegels in jede Richtung verteilt werden. Die Menge der ausgesandten Strahlen r_i^a wird durch $1 + p_4 * p_5$ berechnet, was zu einer Menge von Strahlen r_i^{al} mit $l = 1,...,1 + p_4 * p_5$ führt. Für jeden dieser Strahlen werden die folgenden Merkmale extrahiert:

   a) Der Abstand $dr_{i_{i,j}}^a$, den ein Strahl zwischen dem Ursprung des Kegels und dem Schnittpunkt $p_j^{bl} \in t_j^b$ zurücklegt, wird nach Gleichung 3.7 berechnet.

$$dr_{i_{i,j}}^a = \begin{cases} \parallel cp_i - p_j^l \parallel_2 & \text{if } r_i^l \cap t_j \neq \emptyset \\ P_3 & \text{sonst} \end{cases} \tag{3.7}$$

   cp_i beschreibt den Mittelpunkt des Dreiecks t_i. Damit dies für die Segmentierung hilfreich ist, wird dieser Wert auf einen Minimalwert nach Gleichung 3.8 reduziert.

$$dr_{i_j}^a = \min_l dr_{i_{i,j}}^a \tag{3.8}$$

   Dies beschreibt die kürzeste Entfernung eines Strahls zu einem Schnittpunkt.

   b) Ein weiteres Merkmal nh_{ij}^{ab}, das die Anzahl der Schnittpunkte eines Strahls mit anderen Dreiecken darstellt, wird durch die Anzahl der Abstände pro Dreiecksinteraktion gemessen, wobei $dr_{i_j}^a \neq P_3$.

   c) Der maximale Abstand über alle Strahlen für das Dreieck wird

durch Gleichung 3.9 berechnet.

$$Dr_i = \max_l dr_{i_l} \tag{3.9}$$

d) Der minimale Abstand über alle Strahlen eines Dreiecks wird nach Gleichung 3.10 berechnet.

$$dr_i = \min_l dr_{i_l} \tag{3.10}$$

e) Der mittlere Abstand über alle Strahlen eines Dreiecks wird nach Gleichung 3.11 berechnet.

$$\overline{d}r_i = \frac{1}{P_4 P_5 + 1} \sum_l dr_{i_j} \tag{3.11}$$

Aus den aggregierten Distanzen werden die minimale, mittlere und maximale Strahlenlänge ermittelt. Die Anzahl der Treffer nh_{ij} pro Dreieckselement wird durch Summierung der gültigen Treffer berechnet.

Damit ergeben sich alle Parameter, die zur Klassifikation eines Dreiecks benötigt werden.

Machine Learning Klassifikation

Ebenso wie bei der Kantensegmentierung kommt das PatternNet [84] zum Einsatz. Die Normalisierung der Inputdaten ist wie bei der Kantensegmentierung implementiert.

3.2 Grundlagen der Korrosionsprognose

Eine Vorhersage über das Verhalten von Bauteilen unter korrosiver Belastung zu treffen, ist Gegenstand vieler wissenschaftlicher Arbeiten [11, 85, 86]. In den folgenden Kapiteln wird eine Korrosionsprognose für Kanten und Flansche gezeigt. Dabei basiert diese Arbeit auf der Arbeit von Waibel et al. [13]. Hierbei wird auf Grundlage einer FEM-Simulation eine Korrosionsprognose implementiert. Die variierenden Prädiktoren dieser Arbeit sind folgende:

1. Räumliche Prädiktoren: Einflüsse für die Vorhersage des Korrosionsverhaltens sind die räumlichen Prädiktoren. Dabei werden experimentell vier verschiedene Winkel α betrachtet ($\alpha \in 5°, 25°, 65°, 86°$). Der Winkel beschreibt dabei die Neigung des jeweiligen Oberflächenelements bezüglich der horizontalen xy-Ebene.

2. Temporale Prädiktoren: Die Zeit eines Prüfzyklus wird in der Vorhersage berücksichtigt. Dabei wird ein Testzeitraum betrachtet, der sich an den Prüfzeiten des Korrosionstests orientiert. Der Prüfzeitraum beträgt dabei 16 Wochen. Die Auswertung findet in einem festen Zyklus jede Woche statt.

Die Vorhersage berücksichtigt weitere Prädiktoren wie metallurgische, klimatische und elektrochemische Eigenschaften. Diese werden jedoch im Rahmen der Vorhersage nicht variiert. Die in Waibel benutzten Daten [11] betrachten ein Stahlsubstrat und ein Klima. Im Rahmen der Korrosionsprognose werden synthetische Daten für verschiedene Geometrien und Zeitpunkte generiert. Diese erzeugten Daten dienen als Datengrundlage zur Korrosionsprognose. Dabei wurden unterschiedliche Zeitreihennetzwerke implementiert, welche auf Basis der beschriebenen Daten eine Prognose durchführen. Als für die Korrosionsprognose am geeignetsten wurde die NARX-Architektur bestimmt [11].

4 Klassifizierung korrosionsgeschützter Bereiche

In diesem Abschnitt wird eine Vorgehensweise beschrieben, um Bereiche, die durch Klebstoff, Hohlraumkonservierung und Nahtabdichtung geschützt sind, zu erkennen. Dazu wird

1. das Refinement der Geometriedaten erläutert (Kapitel 4.1),
2. die Implementierung der drei Korrosionsschutzmaßnahmen beschrieben (Kapitel 4.2, 4.3 und 4.4), und
3. die implementierten Netzwerken mithilfe von Konzepten der explainable AI analysiert (Kapitel 4.5).

Das Vorgehen ist in einem Ablaufdiagramm dargestellt (Abbildung 4.1). Es beschreibt den Ablauf einer Klassifizierung. Zu Beginn wird die Datenaufbereitung erläutert. Danach lässt sich der Ablauf in die Implementierung des Netzes und die Anwendung der Methodik unterteilen. Soll die Oberfläche eines Fahrzeugs klassifiziert werden, so wird dieser Ablauf für jeden Datensatz durchlaufen. Dabei wird für jede Korrosionsschutzmaßnahme aus Kapitel 2.2 ein eigenes KI-Modell benötigt. Diese Maßnahmen sind Klebstoff, Nahtabdichtung und Hohlraumkonservierung. Existiert ein passendes KI-Modell, so kann dieses Modell ohne die Implementierung auf die Oberfläche angewandt werden. Existiert kein passendes Modell, so muss dieses trainiert werden, bevor es auf die Oberfläche angewandt werden kann.

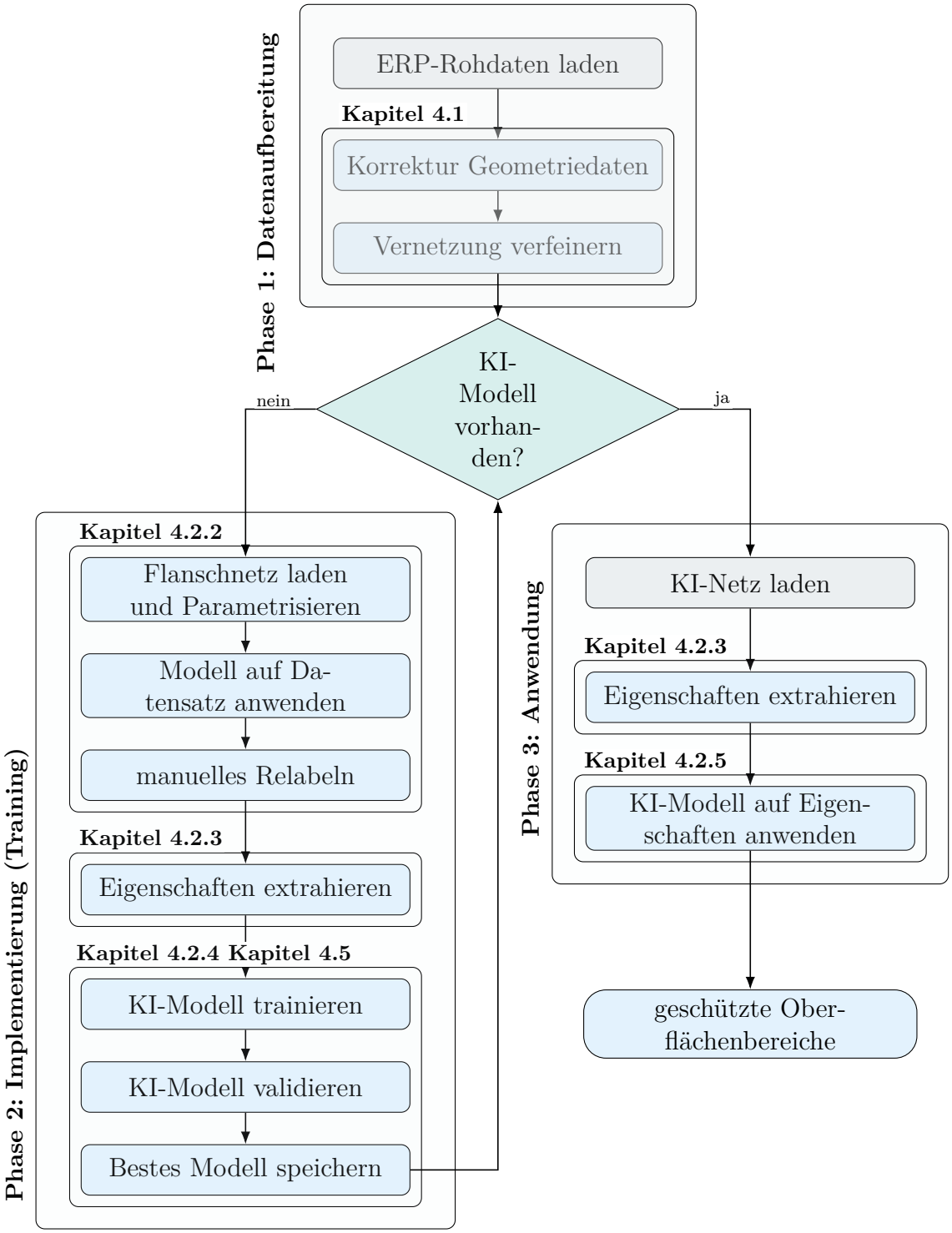
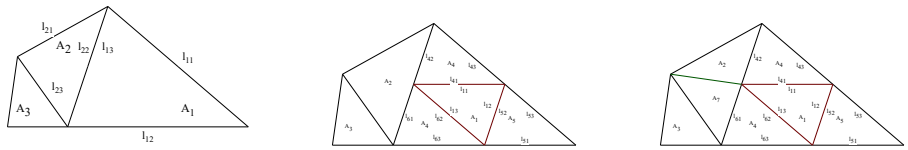


Abb. 4.1: Ablaufdiagramm der Klassifizierungsmethodik

4.1 Implementierung der Geometriedatenaufbereitung

Um gute Klassifizierungsergebnisse zu erzielen, wird eine qualitativ hochwertige Vernetzung der Geometrien benötigt, die über eine Korrektur der Geometriedaten aus Kapitel 2 hinausgeht. Eine hochwertige Vernetzung zeichnet sich dabei durch gleichmäßige Dreiecke aus. Dies ist durch die Berechnung der Eigenschaften in Kapitel 3 begründet. Die Eigenschaften sind, besonders bei den Strahleneigenschaften, abhängig von der Position des Zentrums eines Dreiecks. Bei der Implementierung einer Vernetzungsanpassung ist es wichtig, dass die Oberflächeninformationen erhalten bleiben. Einerseits benötigen unterschiedliche Klassifizierungen das selbe Geometriemodell, andererseits sollen Quellen wie die Simulationen die gleiche Oberfläche als Grundlage nutzen. Ziel einer Geometrieaufbereitung ist zu große oder zu spitze Dreiecke zu unterteilen. Dabei darf die Oberfläche des Gesamtkörpers nicht verändert werden. Daher werden Dreiecke, die in ihrer Fläche zu groß sind, oder zu lange Kanten besitzen, unterteilt. Dabei wird die Dreiecksfläche eines Dreiecks i wie in Gleichung B.1 berechnet. Die Verfeinerung wird mit zwei Schwellenwerten durchgeführt, der Kantenlänge l_i und der Dreiecksfläche A_i. Diese Schwellwerte werden in Kapitel 5 global für die Oberfläche des Rohbaus definiert.



(a) Ausgangsvernetzung einer Beispielgeometrie mit Dreiecken A_1, A_2, A_3. Die Kantenlänge des Dreiecks A_1 liegt oberhalb des Schwellenwerts und wird neu vernetzt.

(b) Das identifizierte Dreieck wird in vier neue Dreiecke unterteilt. Die neu eingefügten Kanten sind rot gefärbt. Die neuen Dreiecke werden der Menge der Dreiecke k hinzugefügt.

(c) Teilung der benachbarten Dreiecke, um die STL-Konventionen einzuhalten. Die neu eingefügte Kante ist grün markiert.

Abb. 4.2: Schematischer Ablauf einer Iteration der Unterteilung der Vernetzung einer Geometrie

Da eine bessere Qualität der Vernetzung benötigt wird, die Geometrie jedoch nicht verändert werden darf, wird ein Ansatz gewählt, der die Kanten von Dreiecken unterteilt. Die Dreiecke, die einen der Schwellenwerte überschreiten, werden durch das Teilen der Kanten in 4 neue Dreiecke unterteilt. Um die Konvention der STL-Dateien zu behalten, wird daraufhin eine Korrektur der Nachbarn durchgeführt, deren Kanten halbiert wurden (Abbildung 4.2). Dies

kann iterativ wiederholt werden, bis keine Dreiecke mehr aufgeteilt werden können. Damit stellt sich der Algorithmus 1 zur Neuvernetzung wie folgt dar:

Algorithmus 1 STL-Neuvernetzungsalgorithmus

Definieren der Schwellwerte, um zu verfeinernde Dreiecke zu identifizieren:
A_{max}; l_{max}; $\mathcal{P}$: Einzelbauteil; $\mathcal{A}$: Datensatz
for parts $\mathcal{P}$ in $\mathcal{A}$ **do**
  $k \in \mathcal{P}; A_k \geq A_{max}$
  while $k \neq \emptyset$ **do**
    if $l_{kj} \geq l_{max} \vee A_k \geq A_{max}$ **then**
      Dreiecke werden in vier neue Dreieecke geteilt
    end if
    Finde Nachbarn des Dreiecks k
    Diese Dreiecke werden in zwei Dreiecke geteilt
    $k \leftarrow i \forall l_{kj} > l_{max} \vee A_k > A_{max}$
  end while
end for
return $\mathcal{P}_{\text{refined}}$

4.2 Implementierung der Klebstoffklassifizierung

Zusätzlich zu den in Kapitel 3.1 vorgestellten Eigenschaften werden für die Klebstoffklassifizierung weitere Informationen benötigt. Um die durch Klebstoff geschützten Bereiche zu identifizieren, müssen aus den digitalen Datensätzen (Abbildung 2.8) weitere Informationen für die Klebstoffklassifizierung gewonnen werden. In einem ersten Schritt werden die datensatzspezifischen Geometrieaufbereitungsschritte behandelt, anschließend die Implementierung des Transfer Learnings, sowie die klebstoffspezifische Eigenschaftsextraktion und abschließend die Architektur des Klassifizierungsnetzwerks für die Klebstoffanwendung.

4.2.1 Maßnahmenspezifische Aufbereitung der Geoemtriedaten

Bauteildurchmesser Für jedes Klebstoffbauteil $M_{Klebstoff}$ (gezeigt in Abbildung 2.8), welches eine Maßnahme dokumentiert, werden die Bauteildurchmesser berechnet.
Um die Dicke von diesen Klebstoffbauteilen zu extrahieren, wird auf den bereits verwendeten Ansatz der SDF [17] zurückgegriffen. Dabei wird jedoch nur ein Strahl ausgesandt, da nur der senkrecht zum Dreieck verlaufende Strahl benötigt wird, um den Durchmesser zu berechnen. Die Richtung wird entgegen der Normalenorientierung gewählt, indem man den Vektor für seine Richtung durch $\vec{r_i} = -n_i$ definiert.

Für diesen Strahl wird der Schnittpunkt mit allen anderen Dreiecken $t_j; j \neq i; j \in M_{Klebstoff}$ geprüft, die in den AABBs der gruppierten Elemente einen Schnittpunkt haben. Wenn es keinen Schnittpunkt gibt, wird der Wert für die Strahlenlänge auf den zuvor definierten Maximalwert d_{max} des Strahls gesetzt. Gibt es einen Schnittpunkt zwischen dem Strahl und einem Dreieck, so wird der Schnittpunkt p_j berechnet und anhand dieser Verbindung der Abstand $dr_{i,j}$ des Strahls ermittelt:

$$dr_{i,j} = \begin{cases} \|c_i - p_{i,j}\|_2 & \text{if } r_{i,j} \cap t_j \neq \emptyset \\ d_{max} & \text{sonst} \end{cases} \tag{4.1}$$

Der minimale Wert jedes Strahls definiert sich als:

$$dr_i = min(dr_{i,j}) \tag{4.2}$$

Um den Durchmesser $d_{Klebstoff,k}$ des Bauteils zu erhalten, wird Gleichung 4.3 verwendet.

$$d_{Klebstoff,k} = \frac{\sum_1^i dr_i}{i} \tag{4.3}$$

Dies wird für jedes $M \in P_{Klebstoff}$ im Datensatz vorhandene Klebstoffteil durchgeführt.
Bei den Datensätzen, die Informationen über den Klebstoff enthalten, muss eine Filterung vorgenommen werden. In diesem Datensatz sind nicht nur Klebstoffe dokumentiert, sondern auch andere Informationen zu Bauteilen enthalten. Die Filterung erfolgt auf der Grundlage der in der Datenvorverarbeitung extrahierten Informationen. Dies sind das Volumen der Bounding Box und der in Gleichung 4.3 berechneten Dicke der Klebstoffbauteile. Ziel ist es, größere und kleine Teile zu filtern, die abweichende Informationen beinhalten. Wird dieser Filter und die Geometrieneuvernetzung auf einen Datensatz angewandt, so ist dieser Datensatz bereit, um klassifiziert zu werden. Diese Vorbereitung wird für jeden Datensatz durchgeführt, unabhängig davon, ob er zum Training oder zur Anwendung genutzt wird.

4.2.2 Transfer Learning zur Implementierung des Klebstoffnetzwerks

Zur effizienten Implementierung des Klebstoffnetzwerks wird in einem ersten Schritt ein Neuronales Netzwerk eingesetzt, das zur Flanschklassifizierung nach Waibel [12] genutzt wurde. Dieses Netzwerk wird genutzt, um ein Klassifizierungslabel auf Basis des Klebstoffdatensatzes, sowie des Rohbaudatensatzes zu erzeugen. In einem letzten Schritt werden diese Daten zum Training des Netzwerks mit neu extrahierten Eigenschaften verwendet.
Der Definition aus Kapitel 2.1 folgend, beschreibt Transfer Learning das Lösen von Aufgaben, für welche das Netzwerk nicht trainiert wurde, die der ursprünglichen Aufgabe jedoch ähnlich sind.
Für den Fall des Klebstoffs ist dies die Flanschklassifizierung aus Kapitel 2.

Das Problem kann in beiden Fällen als Klassifizierungsproblem in Abhängigkeit von mehreren Geometrien beschrieben werden. Um Transfer Learning zu ermöglichen, werden Rohbau-Klebstoffverbindungen betrachtet. Um die Vorteile der Klassifizierung aus Kapitel 3.1 nutzen zu können, muss eine Reduktion der Klebstoffverbindungen vorgenommen werden. Während Flansche aus zwei Rohbauteilen bestehen, setzen sich Klebstoffverbindungen aus zwei Rohbauteilen und einem Klebstoffbauteil zusammen. Diese Bauteilverbindung wird auf ein Rohbauteil-Klebstoffbauteil reduziert, also jedes Rohbauteil in seiner Interaktion mit dem Klebstoffbauteil getrennt betrachtet. Die Eigenschaften zum Aufbau der angepassten Flanschklassifizierung sind wie in Tabelle 4.1 aufgezählt definiert.

Parameter	Wert
Maximaler Strahlenwinkel α_{cone}	2π
Strahleneigenschaft p	10
Strahleneigenschaft q	10
Erweiterung AABB p_1, p_2	10,10
Regularisierung γ	0
Nachbearbeitung δ_A^0, δ_A^1	100
Nachbearbeitung δ_n^0, δ_n^1	1
Anzahl Topeigenschaften k	10

Tabelle 4.1: Parameter zur Klassifizierung von Bauteiloberflächen im Hinblick auf Klebstoffe mittels einem Flanschnetzwerk

Dabei beschreibt α_{cone} den Winkel, in dem Strahlen ausgesendet werden, k die Anzahl der in der Klassifizierung betrachteten Strahlen, p und q die Anzahl der Strahlen. γ beschreibt die Parameter zur Normalisierung der Eigenschaften. δ definiert die Parameter des Nachbereitungsalgorithmus nach Waibel [11].
Um eine korrekte Trainingsgrundlage zu garantieren, werden die aus den angepassten Klassifizierungsergebnissen produzierten Daten händisch korrigiert. Dazu wurde für diese Arbeit ein Tool erstellt, um einzelne Dreiecke zu markieren. Dieses Tool ermöglicht die Korrektur der Datensätze, um die Trainingsdaten zu generieren. Diese Trainingsdaten werden exportiert und zum Training benutzt. Mithilfe dieses Ansatzes ergibt sich ein Trainingsdatensatz, der wie in Tabelle 4.2 aufgebaut ist. In der Erstellung des Datensatzes wurde vor allem auf Fahrzeugtüren zurückgegriffen. Türen besitzen eine hohe Komplexität und eine hohe Dichte an Maßnahmen. Dies kann im Vergleich der einzelnen Geometrien beobachtet werden. Während bei dem Längsträger nur 0,1% der Oberfläche durch Klebstoff geschützt ist, sind dies bei den Türen mit 3-5% deutlich mehr. Daher bieten sich diese Geometrien gut als Datenbasis an. Darüber hinaus wird eine Geometrie gewählt, die aus klassischen Rohbauteilen besteht. Damit lässt sich die Verallgemeinerung zwischen mehreren Baureihen sicherstellen. Darüber hinaus liefern diese Datensätze eine große Datenbasis.
Abbildung 4.3 zeigt die Verteilung der beiden Klassen 0 (nicht von Klebstoff geschützt) und 1 (von Klebstoff geschützt).

Fahrzeug	Bauteil	Kurze Beschreibung	Anzahl Dreiecke	Anzahl positiver Dreiecke	Anteil positiver Dreiecke
A	Rohbau-teil	Einzelner Längsträger, Teil eines Rohbaus	2388890	2507	0.1%
A	Tür	Vordere und hintere linke Tür mit Scharnieren	3544182	128523	3.6%
B	Tür	Vordere und hintere linke Tür mit Scharnieren	3262408	170963	5.2%
C	Tür	Vordere und hintere linke Tür ohne Scharnieren	2986186	133340	4.5%
D	Tür	Vordere und hintere linke Tür mit Scharnieren	2939380	124005	4.2%

Tabelle 4.2: Datensätze die zum Training des Neuronalen Netzes genutzt werden. Die Buchstaben der Fahrzeuge beschreiben keine Namen, sondern lediglich die Zugehörigkeit der Geometrien zu gemeinsamen Baureihen.

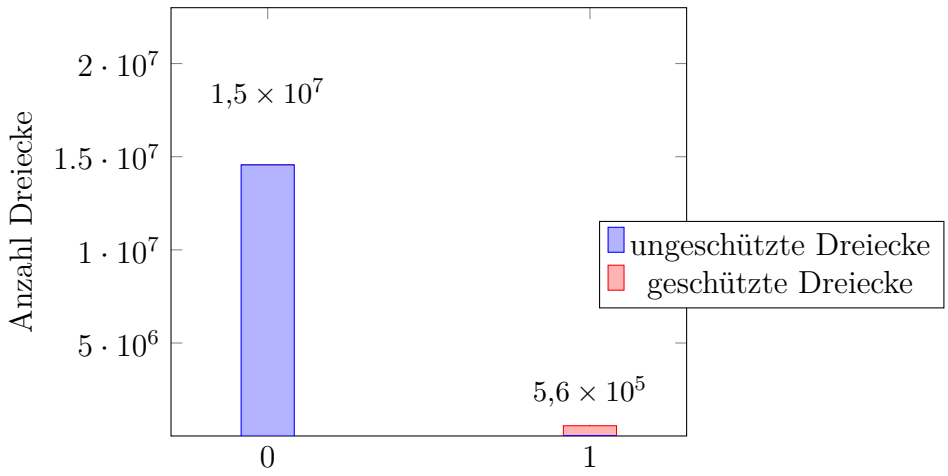


Abb. 4.3: Verteilung des gelabelten Datensatzes. Es existieren deutlich mehr nicht von Klebstoff geschützte Bereiche als Bereiche, die geschützt sind.

Dabei ist die Anzahl der Klasse 0 um den Faktor 100 höher als die Anzahl der Klasse 1. Dies bestätigt den Bedarf nach einem Evaluierungsmaßstab für das Netzwerk, das es ermöglicht, besonders ungleiche Datensätze zu evaluieren. Wie in Kapitel 2.1 beschrieben, bietet sich hier als Evaluierungsmetrik der

MCC an.

4.2.3 Implementierung der Eigenschaften zur Klebstoffklassifizierung

Um eine verbesserte Klassifizierung für Klebstoffe zu implementieren, werden weitere Eigenschaften berechnet.

Übernommene Eigenschaften aus der Flanschklassifizierung

Einige Eigenschaften werden aus den Eigenschaften der Klassifizierung der Flansche übernommen:

1. Wie in der Flanschklassifizierung wird zunächst geprüft, welches der Dreiecke eines Klebstoffbauteils in der Nähe des zu klassifizierenden Dreiecks liegt. Dabei beschreiben die Parameter p_1 und p_2 den Wert, um den die Bounding Box verschoben wird. Für Dreiecke, in deren Nähe keine Klebstoffdreiecke zu finden sind, werden keine Eigenschaften extrahiert. Sie werden auf null gesetzt.

2. Die bekannten Eigenschaften der Flanschklassifizierung werden benutzt:
   a) Intrinsische Eigenschaften: Die Winkel zu den Nachbardreiecken j sowie die mittlere Strahlenlänge d_m.
   b) Extrinsische Eigenschaften:
      i. Die minimale euklidische Distanz wird gemäß Gleichung 3.2 berechnet.
      ii. Die Vektorprojektion wird gemäß Gleichung 3.3 berechnet.
      iii. Das Normalenprodukt wird gemäß Gleichung 3.4 berechnet.
      iv. Die projizierte Überlappung der Dreiecksflächen wird gemäß Gleichung 3.6 berechnet.
      v. Die Anzahl der Schnittpunkte, gemäß Kapitel 3.1, wird als Eigenschaft genutzt und als nh_{ij} bezeichnet.
      vi. Die minimale Strahlenlänge dr_{ij}^{a} gemäß Gleichung 3.8.

Erweiterung der bestehenden Eigenschaften

Es werden weitere Eigenschaften implementiert, um die Klassifizierung von klebstoffgeschützten Dreieckselementen zu verbessern.

1. Die Normalenrichtung wird in den bisher beschriebenen Eigenschaften nicht berücksichtigt. Der Wert des Normalenprodukts berücksichtigt zwar den Winkel, jedoch nicht die Orientierung der beiden Normalen zueinander. Daher berechnet sich diese Eigenschaft wie folgt:

$$np_{ij} = \|n_i + nj\| \tag{4.4}$$

Dieses Feature berücksichtigt damit die Orientierung der beiden Dreiecke zueinander.

2. Bei der Klebstoffsegmentierung spielt der Klebstoffdurchmesser eine wichtige Rolle. Daher wird dieser Durchmesser entsprechend Gleichung 4.3 berechnet und mit $d_{m,Klebstoff}$ bezeichnet. Dabei wird der Durchmesser für ein gesamtes Klebstoffbauteil berechnet und jedem Dreieck des Bauteils m zugeordnet.

3. Klebstoffröhren werden auf Bauteilen konstruiert. Diese Bauteile werden Trägerbauteil genannt. Ob ein Bauteil ein Trägerbauteil ist, wird in eine binäre Eigenschaft gefasst. Die Trägerbauteile sind die Bauteile, deren Oberfläche am nächsten an der Mittellinie des Klebstoffbauteils liegt. Um dies numerisch zu bestimmen, wird folgender Ablauf implementiert.
   a) Mittels des bestimmten Klebstoffdurchmessers $d_{m,\,Klebstoff}$ werden die Zentrumspunkte $cp_{Klebstoff}$ der Klebstoffröhre konstruiert.
   b) Um jeden dieser Punkte wird eine Bounding Box entsprechend der Implementierung für Dreiecke konstruiert. Jedes Dreieck eines Rohbauteils wird in den weiteren Schritten betrachtet.
   c) Für jedes dieser Dreiecke wird die kürzeste Distanz zu einem Klebstoffdreieck berechnet.
   d) Die Bauteildreiecke t_l mit der kürzesten Distanz werden gesammelt und gezählt. Damit ergibt sich der numerische Wert je Rohbauteil M wie folgt:
   $$nk_{M_{Klebstoff},M} = \sum_{1}^{l_M} 1 \tag{4.5}$$
   Da Klebstoffe mit mehreren Bauteilen interagieren können, wird das Bauteil als Trägerbauteil gewählt, welches die meisten Dreiecke mit der kürzesten Distanz besitzen:
   $$nk_M = \max nk_{M_{Klebstoff},M} \tag{4.6}$$
   e) Damit ergibt sich die Eigenschaft für jede Interaktion eines Rohbaudreiecks $t_i i \in M$ mit einem Klebstoffbauteil $M_{Klebstoff}$ wie folgt:
   $$nc_i = \begin{cases} 1, & \text{falls } M \in nk_M \\ 0, & \text{sonst} \end{cases} \tag{4.7}$$

4. Wie in Kapitel 2.2.4 beschrieben, existieren Klebstoffe, die in Falzen liegen. Um diese von Zweiblechverbindungen unterscheiden zu können, bei denen keine Falzung vorgenommen wird, wird jedem Klebstoffbauteil $M_{Klebstoff}$ eine Eigenschaft zugeordnet. Diese beschreibt, ob das Bauteil in einer Falz liegt. Dieses Feature ist in Abbildung 4.4 gezeigt und wird wie folgt bestimmt:

a) Mittels der Zentrumspunkte $cp_{Klebstoff}$ und den am nächsten liegenden Dreiecken t_l wird die senkrecht zu der Klebstoffröhre liegende Richtung definiert. Dazu werden die Normalen der Dreiecke t_l genutzt.

b) In einem weiteren Schritt werden in und gegen die Richtung der Normalen Schnittpunkte mit Rohbaudreiecken gesucht. Diese Schnittpunkte werden gezählt und als nsp bezeichnet.

c) Jedem Klebstoffbauteil wird die Eigenschaft nsp_{Falz} zugeordnet:

$$nsp_{Klebstoff} \begin{cases} 1 & \text{, falls } nsp = 4 \\ 0 & \text{, ansonsten} \end{cases} \tag{4.8}$$

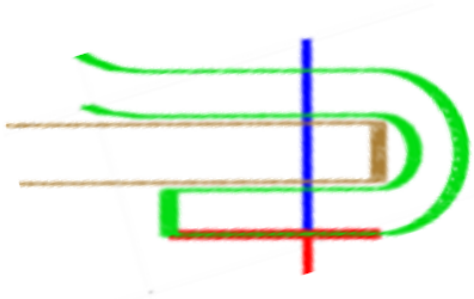


Abb. 4.4: Visualisierung eines Falzes mit zwei Bauteilen (grün,braun), sowie den in der Methodik beschriebenen Normalen (rot,blau). Das grüne Bauteil wird als Trägerbauteil identifiziert. Dies wird über die rote Normale und die rot markierte Fläche bestimmt.

Erweiterung der Eigenschaften um geometrische Einflüsse

Die Eigenschaftsextraktion basiert auf der Interaktion eines Dreiecks mit einem anderen Dreieck. Dies kann durch das Berücksichtigen der geometrischen Umgebung des zu klassifizierenden Bauteils erweitert werden. Dazu werden die berechneten Normalen n_i und die Schwerpunkte cp_i für jedes Dreieck verwendet. Die Implementierung der geometrischen Merkmale folgt der Implementierung von Weinmann [87] und Blomley [88]. Für jedes Dreieck werden dessen Nachbarn und die Normalen im Schwerpunkt des entsprechenden Dreiecks extrahiert. Aus allen Normalen wird dabei eine Eigenwertmatrix berechnet. λ_i beschreibt dabei die Eigenwerte dieser Matrix. Die folgenden Merkmale werden berechnet:

1. Linearität: Misst das Ausmaß, in dem Punkte in einer Linie ausgerichtet sind.

$$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{4.9}$$

2. Planarität: Misst, ob die Punkte auf einer Ebene liegen.

$$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{4.10}$$

3. Sphärizität : Gibt die Gleichmäßigkeit der Punkteverteilung in alle Richtungen an.

$$S_\lambda = \frac{\lambda_3}{\lambda_1} \tag{4.11}$$

4. Omnivarianz: Ein Maß, das alle Eigenwerte kombiniert, um die Gesamtstreuung der Punkte darzustellen.

$$O_\lambda = \sqrt[3]{\prod_{i=1}^{3} \lambda_i} \tag{4.12}$$

5. Anisotropie: Stellt die Varianz der Punkte in der am stärksten variierenden Richtung dar.

$$A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} \tag{4.13}$$

6. Eigenentropie: Quantifiziert die Zufälligkeit in der Verteilung der Punkte.

$$E_\lambda = -\sum_{i=1}^{3} \lambda_i \ln \lambda_i \tag{4.14}$$

7. Summe der Eigenwerte: Spiegelt die Gesamtstreuung der Punkte wieder.

$$\Sigma_\lambda = \sum_{i=1}^{3} \lambda_i \tag{4.15}$$

8. Lokale Oberflächenvariation: Gibt die Glätte oder Rauheit der Oberfläche an.

$$C_\lambda = \frac{\lambda_3}{\sum_{i=1}^{3} \lambda_i} \tag{4.16}$$

Entgegen dem Ansatz von West [89], werden diese Merkmale nicht auf der Grundlage eines ganzen Teils implementiert. Stattdessen werden sie auf der Grundlage eines lokalen Ansatzes berechnet, um die lokale Geometrieumgebung in die Bewertung einfließen zu lassen. Die gesamte Punktwolke wird auf Basis der Entfernung zum Dreieck t_i gefiltert. Es wird ein Radius definiert, in dem alle Punkte berücksichtigt werden, die im Bereich der Klebstoffwirkung liegen. Die Klebstoffwirkung wird in ihrer maximalen radialen Wirkung als 8 mm angenommen. Dieser Wert wird als Radius für den Ansatz nach West [89] gewählt.

4.2.4 Hyperparametertuning

Um ein optimiertes Netzwerk definieren zu können, wurde ein Hyperparametertuning durchgeführt, das eine Kombination der Rastersuche und der manuellen Suche ist. Es wird ein Bereich gewählt, in dem sich die Parameter bewegen. Innerhalb dieses Rasters wird dann der Wert bestimmt. Dies liegt in der Laufzeit des Systems begründet. Dabei wurde der Werteraum, in dem die Parameter betrachtet werden, wie in Tabelle 4.3 definiert.

Parameter	Wertraum	Bestimmter Wert
Strahleneigenschaft p	[4;16]	4
Strahleneigenschaft q	[4;16]	4
Maximaler Strahlenwinkel α_{cone}	$[\pi/8; 2\pi]$	2π
Maximale Strahlenlänge	[10; 40]	10
Anzahl Eigenschaften k	[5;15]	5
Anzahl der Netzwerkschichten	[3;5]	5
r,u,w,e,f,g	[0;1]	1

Tabelle 4.3: Parameterraum und durch Rastersuche ermittelte Hyperparameterwerte für das Klebermodell.

Die Parameter r,u,w,e,f,g haben folgende Bedeutung:

1. r: Falls dieser Parameter null ist, wird eine Klassifizierung implementiert, welche nicht den Parameter der Überlappung berücksichtigt.
2. u: Falls dieser Parameter null ist, werden die Eigenschaften aus Gleichung 4.7 und Gleichung 4.4 nicht weiter genutzt.
3. w: Falls dieser Parameter null ist werden die Eigenschaften aus Kapitel 4.2.3 nicht berücksichtigt.
4. e: Falls dieser Parameter null ist, werden die Eigenschaften aus Gleichung 4.8 nicht berücksichtigt.
5. f: Falls dieser Parameter null ist, wird der Durchmesser einer Klebstoffröhre nicht berücksichtigt.
6. g: Falls dieser Parameter null ist, wird der Durchmesser des Rohbauteils nicht berücksichtigt.

Ergebnis dieser Parameteranalyse sind die bestimmten Werte aus Tabelle 4.3. Die Genauigkeit des trainierten Netzwerks ergibt sich wie in Tabelle 4.4.

4.2.5 Klassifizierungsergebnisse

Mit den im vorangegangenen Kapitel beschriebenen Parametern und dem trainierten Netzwerk wird eine Klassifizierung eines Gesamtfahrzeugs durchgeführt. Diese kann sowohl quantitativ als auch qualitativ evaluiert werden. Die

Metrik	TPR	TPA	ACC	MCC
Wert	0.9774	0.9792	0.9786	0.9546

Tabelle 4.4: Validierungsgenauigkeit des trainierten Klebstoff-Netzwerks

quantitative Evaluierung ist in Kapitel 2.1 beschrieben. Die Konfusionsmatrix ergibt sich wie in Tabelle 4.5 Die in Abbildung 4.3 dargestellte Verteilung lässt erkennen, dass ein sehr ungleich verteilter Datensatz vorliegt. Dies bedeutet, dass es um den Faktor 100 mehr 0 bewertete Dreiecke als mit 1 bewertete Dreiecke gibt. Für diesen Fall bietet sich der MCC (Gleichung 2.11) als Genauigkeitsmaß an.

	Positive ungewichtete Vorhersage (1)	**Negative ungewichtete Vorhersage (0)**
Positives ungewichtete Label (1)	173082 (1,0%)	7208 (0,00%)
Negative ungewichtetes Label (0)	5900 (0,00%)	18852296 (99,00%)

Tabelle 4.5: Konfusionsmatrix für die zwei Klassen 0 und 1

Jede Vorhersage kann parallel zu dem Ansatz der Auswertung der Klebstoffklassifizierung gewichtet werden. Dabei wird die Gewichtung auf Basis der Fläche jedes vorhergesagten Dreiecks implementiert.

	Positive gewichtete Vorhersage (1)	**Negative gewichtete Vorhersage (0)**
Positive gewichtete Label (1)	3831721 (3,7%)	100825 (0,10%)
Negative gewichtete Label (0)	227807 (0,2%)	96810356 (96,0%)

Tabelle 4.6: Über die Fläche der einzelnen Dreiecke gewichtete Konfusionsmatrix für die zwei Klassen 0 und 1.

Die Bereiche, die in den Konfusionsmatrizen mit Grün markiert sind, sind solche Bereiche, in denen das Netzwerk eine korrekte Vorhersage trifft. Damit wird

99,7% (Tabelle 4.6) der Fläche, oder fast 100% der Dreiecke (Tabelle 4.5), korrekt erkannt. Die prozentualen Anteile von klebstoffgeschützten Bereichen an einem Gesamtfahrzeug lassen sich damit weiterhin genauer berechnen. Ungefähr 1% der Oberflächendreiecke eines Gesamtfahrzeugs sind durch Klebstoff geschützt. Damit deckt sich dies mit dem im Trainingsdatensatz genutzten Rohbauteil. Aus den Werten der Tabelle 4.5 und Tabelle 4.6 kann der MCC berechnet werden. Die in Kapitel 2.1 beschriebenen Genauigkeiten ergeben sich aus Tabelle 4.7.

Genauigkeit	TPA	TPR	F-Maß	ACC	MCC
Gewichtet	0.9439	0.9744	0.9589	0.9967	0.9573
Ungewichtet	0.9997	0.9996	0.9997	0.9993	0.9632

Tabelle 4.7: Die Genauigkeiten der Klebstoffklassifizierung, basierend auf der gewichteten und der nicht gewichteten Konfusionsmatrix

Generell liefert die vorgestellte Methodik quantitativ sehr gute Ergebnisse für das Gesamtfahrzeug. Das sehr gute Ergebnis ist in Abbildung 4.5 visualisiert. Das Ergebnis dieser Klassifizierung ist die Visualisierung der Genauigkeiten aus den Konfusionsmatrizen.

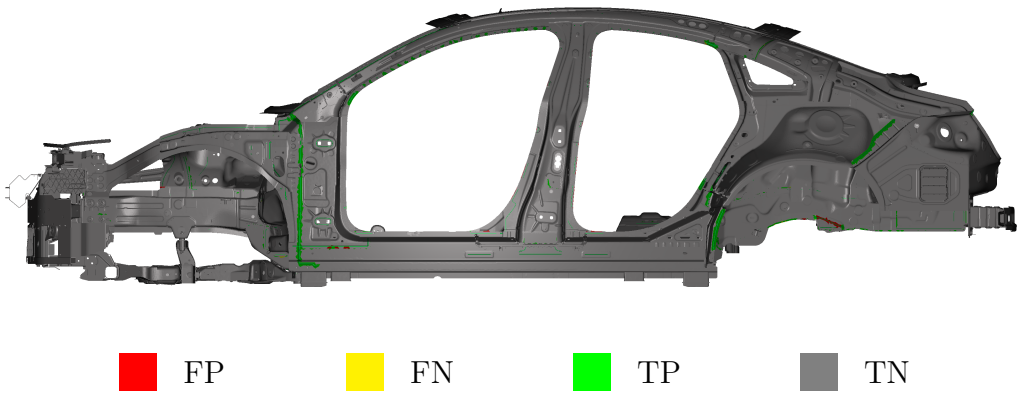
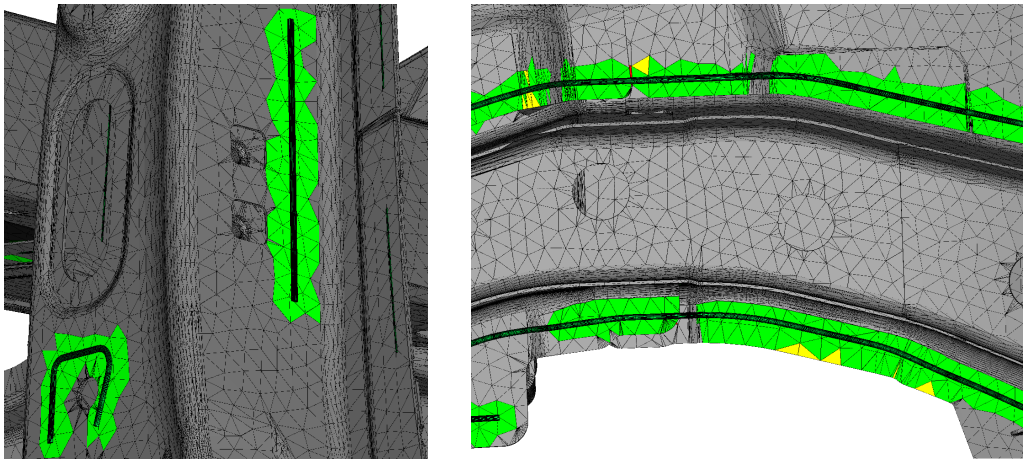


Abb. 4.5: Ergebnis der Klassifizierung eines Gesamtfahrzeugs

Abbildung 4.6 zeigt Ausschnitte aus der Klassifizierung des Gesamtfahrzeugs. Dabei zeigt Abbildung 4.6a ein Beispiel für eine sehr gut funktionierende Klassifikation von klebstoffgeschützten Oberflächen. Abbildung 4.6b zeigt Bereiche mit falsch-negativ klassifizierten Dreiecken. Wichtig für diese Anwendung ist es, falsch-positive Klassifizierungen zu vermeiden, denn diese Klassifizierung dient als Grundlage zur Bestimmung des Korrosionsrisikos. Damit kein Korrosionsrisiko unerkannt bleibt, ist es wichtig, eine geringe Anzahl an falsch-positiven Dreiecken zu erreichen. Abbildung 4.6b zeigt zwar einige falsche Klassifizierungen, diese sind jedoch lediglich falsch-negative Dreiecke.



(a) Beispiel für einen gut klassifizierten Bauteilbereich

(b) Beispiel für einen Bauteilbereich mit Fehlklassifizierung (FN)

FP FN TP TN

Abb. 4.6: Ausschnitte der Ergebnisse der Klebstoffklassifizierung

4.3 Implementierung der Nahtabdichtungsklassifizierung

In diesem Abschnitt wird die Implementierung der NAD-Klassifizierung beschrieben. Dabei werden die durch NAD korrosionsgeschützten Bereiche identifiziert. Dies geschieht durch die Kombination von Nahtabdichtungsdatensätzen (Abbildung 2.4) zu den zugehörigen Rohbaudatensätzen. Dieser methodische Ansatz orientiert sich eng an dem Ansatz der Klebstoffklassifizierung.

4.3.1 Transfer Learning zur Implementierung des Nahtabdichtungsnetzwerks

Die Parameter in Tabelle 4.8 wurden im Vergleich zu dem Ansatz des Transfer Learning bei der Implementierung des Klebstoffs näher an den Parametern der Flanschklassifizierung belassen. Dies liegt in der geometrischen Interaktion zwischen Rohbauteil und NAD-Bauteil begründet. NAD-Bauteil und Rohbauteil bilden einen Flansch, es kann also eine Flanschklassifizierung genutzt werden, um die Trainingsdatensätze zu erstellen. Die Parameter p und q sind deutlich kleiner, da auch die Wirkweite deutlich geringer ist. Die Anzahl der Strahlen und die Berücksichtigung der besten k Strahlen sind von der Flanschklassifizierung übernommen. Besonders anzumerken ist die Erweiterung des Winkels α_{cone}. Diese Abweichung begründet sich in der Geometrie der NAD, die sich in einzelnen Fällen mit dem Rohbauteil überschneidet.
Mittels Transfer Learning wurde für das Training ein Datensatz generiert. Dieser ist wie in Tabelle 4.9 zusammengesetzt.

Parameter	Wert
Maximaler Strahlenwinkel α_{cone}	2π
Strahleneigenschaft p	5
Strahleneigenschaft q	5
Erweiterung AABB p_1, p_2	5,5
Regularisierung γ	0
Nachbearbeitung δ_A^0, δ_A^1	100
Nachbearbeitung δ_n^0, δ_n^1	1
Anzahl Topeigenschaften k	5

Tabelle 4.8: Genutzte Parameter des Flanschnetzwerks zur Klassifizierung von NAD-Bauteilen mittels Transfer Learning.

Abbildung 4.7 zeigt eine deutliche Neigung zu nicht von NAD geschützten Dreiecken. Wie auch bei der Klebstoffklassifizierung, beeinflusst dies, welche die geeignete Evaluierungsmethodik ist. Der MCC (Gleichung 2.11) wird zur Bewertung der Qualität der Prognose gewählt.

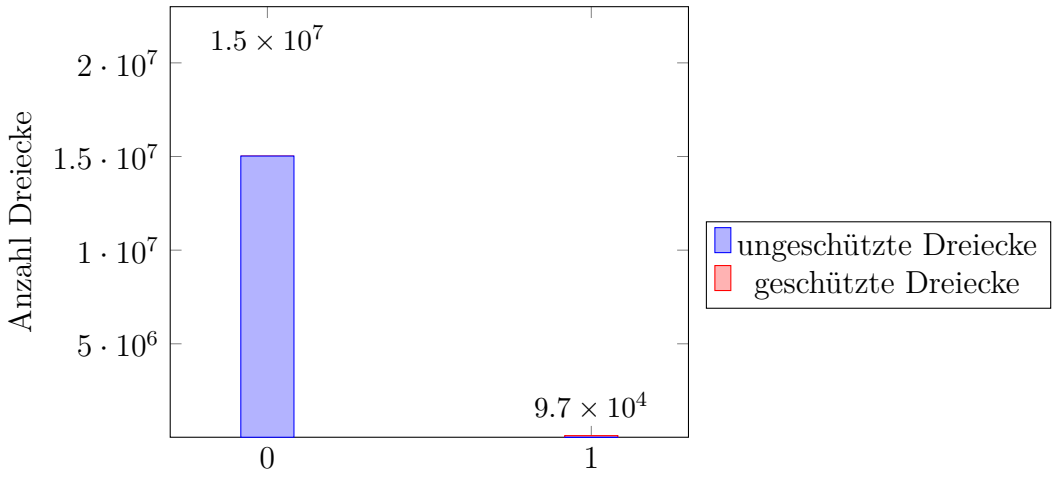


Abb. 4.7: Verteilung des gelabelten Datensatzes zur NAD-Klassifizierung. Der Unterschied in der Größe der Klassen liegt bei Faktor 100.

Fahrzeug	Bauteil	Kurze Beschreibung	Anzahl der Dreiecke	Anzahl positiver Dreiecke	Anteil positiver Dreiecke
A	Längs-träger	Einzelner Längsträger als Teil des Rohbauaufbaus	2388890	11782	0.4%
A	Tür	Vordere und hintere linke Tür mit Scharnieren	3544182	22949	0.6%
B	Tür	Vordere und hintere linke Tür mit Scharnieren	3262408	19617	0.6%
C	Tür	Vordere und hintere linke Tür ohne Scharnieren	2986186	14807	0.5%
D	Tür	Vordere und hintere linke Tür mit Scharnieren	2939380	27627	0.9%

Tabelle 4.9: Datensatz, der zum Training des Neuronalen Netzes genutzt wird. Die Buchstaben der Fahrzeuge beschreiben keine Namen, sondern lediglich die Zugehörigkeit der Geometrien zu gemeinsamen Baureihen.

4.3.2 Implementierung der Eigenschaften zur Klassifizierung von durch Nahtabdichtung geschützten Bereichen

Wie auch das Klebstoffnetzwerk benötigt das NAD-Netzwerk Eigenschaften als Input. Diese Eigenschaften können aus der Klebstoffklassifizierung und aus der Flanschklassifizierung übernommen werden.

Übernommene Eigenschaften aus der Flanschssegmentierung

Für die NAD-Klassifizierungen können Eigenschaften aus der Flanschklassifizierung übernommen werden:

1. Wie in der Klebstoffklassifizierung wird zunächst geprüft, welches der Dreiecke eines NAD-Bauteils in der Nähe des zu klassifizierenden Dreiecks liegt. Dabei beschreiben die Parameter p_1 und p_2 den Wert, um den die Bounding Box verschoben wird. Für Dreiecke, in deren Nähe keine NAD-Bauteile zu finden sind, werden keine Eigenschaften extrahiert. Sie werden auf null gesetzt.

2. Die bekannten Eigenschaften der Flanschklassifizierung werden benutzt:

a) Intrinsische Eigenschaften: Die Winkel zu den Nachbardreiecken j sowie die mittlere Strahlenlänge d_m

b) Extrinsische Eigenschaften:

   i. Die minimale euklidische Distanz wird gemäß Gleichung 3.2 berechnet.

   ii. Die Vektorprojektion wird gemäß Gleichung 3.3 berechnet.

   iii. Das Normalenprodukt wird gemäß Gleichung 3.4 berechnet.

   iv. Die projizierte Überlappung der Dreiecksflächen wird gemäß Gleichung 3.6 berechnet.

   v. Die Anzahl der Schnittpunkte wird als Eigenschaft genutzt.

   vi. Die minimale Strahlenlänge d_{ij} wird verwendet.

Übernommene Eigenschaften aus der Klebstoffklassifizierung

Zur Klassifizierung von NAD-geschützten Bereichen können Eigenschaften aus der Klebstoffklassifizierung übernommen werden.

1. Die Normalenrichtung (Gleichung 4.4) kann übernommen werden.
2. Die Durchmesserberechnung (Gleichung 4.3) kann auf NAD-Bauteile angewendet werden.

4.3.3 Hyperparametertuning

Um ein optimiertes Netzwerk definieren zu können, wird ein Hyperparametertuning durchgeführt. Dabei wird der Werteraum, in dem die Parameter betrachtet werden, analog der Klebstoffimplementierung, wie in Tabelle 4.10 definiert. Dabei beschreibt r, ob die Normalenrichtung als Parameter übernommen wird. e beschreibt, ob der Durchmesser der NAD-Bauteile übernommen wird.

Parameter	Wertraum	Bestimmter Wert
Strahlenanzahl p	[4; 16]	4
Strahlenanzahl q	[4; 16]	4
Maximaler Strahlenwinkel α_{cone}	$[\pi/8; 2\pi]$	2π
Maximale Strahlenlänge	[10; 40]	10
Anzahl Eigenschaften k	[5; 15]	5
Anzahl der Netzwerkschichten	[3; 5]	5
r	[0; 1]	1
e	[0; 1]	0

Tabelle 4.10: Parameterraum und ermittelte Hyperparameterwerte für das NAD-Modell

Über dieses Hyperparametertuning werden die optimierten Werte bestimmt. Diese sind in Tabelle 4.10 gezeigt. Mit diesen Werten ergibt sich ein trainiertes Netzwerk. Die Genauigkeiten dieses Netzwerks im Training sind in Tabelle 4.11 gezeigt.

Metrik	**TPR**	**TPA**	**ACC**	**MCC**
Wert	0.9075	0.9245	0.9782	0.9034

Tabelle 4.11: Validierungsgenauigkeiten des trainierten NAD-Netzwerks

4.3.4 Klassifizierungsergebnisse

Im Folgenden wird das ermittelte Netz auf einen dem Netz unbekannten Datensatz angewendet und validiert. Dies ist der gleiche Datensatz eines Gesamtfahrzeugs wie aus der Klebstoffklassifizierung. Dabei wird dieses Gesamtfahrzeug klassifiziert. Die Ergebnisse werden manuell bewertet. Dadurch ergibt sich eine Konfusionsmatrix (Tabelle 4.12).

	Positive gewichtete Vorhersage (1)	**Negative gewichtete Vorhersage (0)**
Positive gewichtete Label (1)	293562 (1,5%)	887 (0,00%)
Negative gewichtete Label (0)	19631 (0,10%)	18724406 (98,4%)

Tabelle 4.12: Über die Fläche der einzelnen Dreiecke gewichtete Konfusionsmatrix für die zwei Klassen 0 und 1.

Jede Vorhersage kann analog zu der Auswertung der Klebstoffklassifizierung gewichtet werden. Dabei wird die Gewichtung auf Basis der Fläche jedes vorhergesagten Dreiecks implementiert. Daraus ergibt sich eine gewichtete Konfusionsmatrix (Tabelle 4.13).
Aus den Konfusionsmatrizen (Tabelle 4.12 und 4.13) ergeben sich über die grünen Flächen die korrekt bestimmten Bereiche. Hier ist zu erkennen, dass 99,9% der Dreiecke korrekt klassifiziert werden, die Fläche ebenfalls zu 99,93% korrekt bestimmt wird. Die Genauigkeit der NAD-Klassifizierung eines Gesamtfahrzeugs ergibt sich aus Tabelle 4.14. Hier zeigen sich Unterschiede vor allem in der prozentualen Verteilung der gewichteten und ungewichteten Konfusionsmatrix. Dies lässt zuallererst schließen, dass die positiv gelabelten Dreiecke eher größere Flächen abdecken. Dadurch wird der Anteil der TP-Werte in der gewichteten Matrix höher. Generell lässt sich aber unabhängig von der

	Positive gewichtete Vorhersage (1)	**Negative gewichtete Vorhersage (0)**
Positive gewichtete Label (1)	3232907 (3,2%)	48386 (0,05%)
Negative gewichtete Label (0)	18089 (0,02%)	97671326 (96,73%)

Tabelle 4.13: Über die Fläche der einzelnen Dreiecke gewichtete Konfusionsmatrix für die zwei Klassen 0 und 1

prozentualen Verteilung eine hohe Genauigkeit der NAD-Klassifikation zeigen. Besonders der Wert der FP-Werte ist sehr hoch. In der Gewichtung verliert dieser Wert jedoch wieder an Bedeutung. Dies kann daran liegen, dass die FN-Werte alles sehr kleine Dreiecke sind.

Genauigkeit	TPA	TPR	F-Maß	ACC	MCC
Gewichtet	0.9944	0.9853	0.9898	0.9993	0.9895
Ungewichtet	0.9990	1.0000	0.9995	0.9989	0.9662

Tabelle 4.14: Genauigkeiten der NAD-Klassifizierung, basierend auf der gewichteten und der ungewichteten Konfusionsmatrix.

Diese Genauigkeiten resultieren aus der Klassifikation des Gesamtfahrzeugs, das in Abbildung 4.8 gezeigt ist.

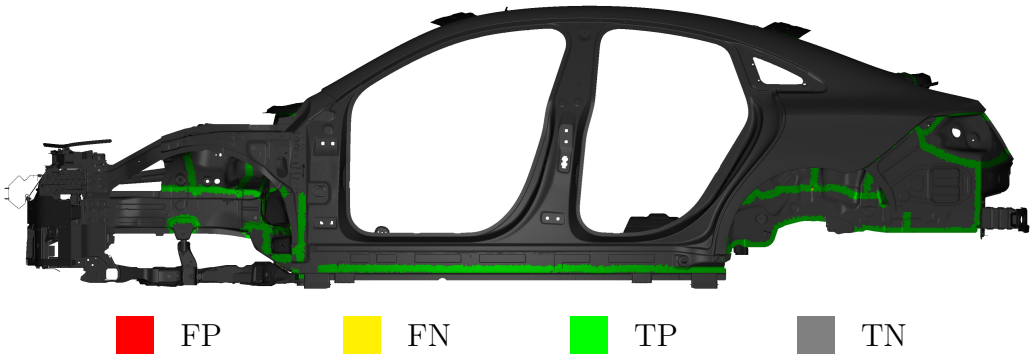
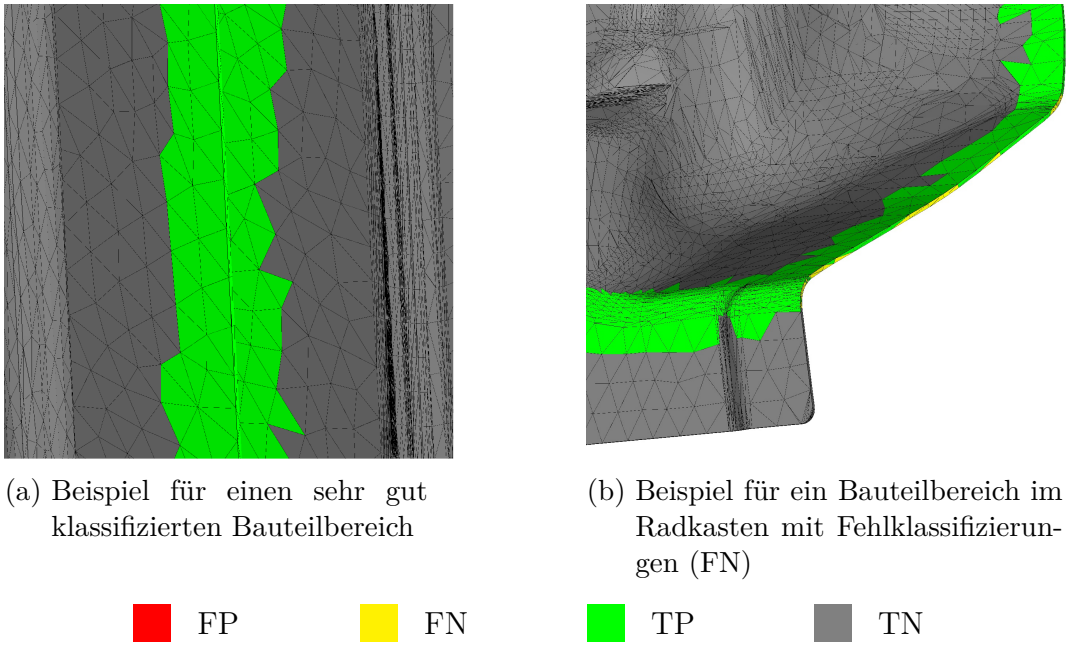


Abb. 4.8: Ergebnis der Nahtabdichtungsklassifizierung eines Gesamtfahrzeugs

Abbildung 4.9 zeigt Ausschnitte dieser Segmentierung. In Abbildung 4.9a wird ein sehr gutes Ergebnis der Klassifizierungsmethodik gezeigt. In Abbildung 4.9b ist eine Klassifizierung zu sehen, bei der der Ansatz an der Kante schlechter arbeitet. Ein Grund dafür können Ungenauigkeiten in der Konstruk-

tion der NAD-Bauteile sein, beispielsweise Überschneidungen zwischen Rohbauteil und NAD-Bauteil. Durch die Methodik der Klassifizierung dominiert bei den falsch segmentierten Dreiecken jedoch die falsch-negativen Dreiecke. Damit ist sichergestellt, dass eine Prognose zur sicheren Seite abschätzt.



(a) Beispiel für einen sehr gut klassifizierten Bauteilbereich

(b) Beispiel für ein Bauteilbereich im Radkasten mit Fehlklassifizierungen (FN)

Abb. 4.9: Ausschnitte der Ergebnisse der Nahtabdichtungsklassifizierung

Generell liefert die vorgestellte Methodik sehr gute Ergebnisse für das Gesamtfahrzeug. Das sehr gute Ergebnis ist in Abbildung 4.8 visualisiert.

4.4 Implementierung der Wachsklassifizierung

Auf Basis von Wachsdatensätzen (Abbildung 2.6) werden die durch Wachs korrosionsgeschützten Bereiche identifiziert. Die Methodik der Eigenschaftsextraktion wird erläutert und zum Training eines neuen Neuronalen Netzes benutzt.

4.4.1 Transfer Learning zur Implementierung des Wachsnetzwerks

Um Effizienz sicherzustellen, wird demselben Konzept gefolgt, wie bei den vorherigen Klassifizierungsnetzwerken. Es wird ebenfalls auf ein Flanschnetzwerk zurückgegriffen. Nach dieser Klassifizierung werden die Daten überprüft und korrigiert. In einem weiteren Schritt werden diese Daten zum Training des Netzwerks mit der beschriebenen Methodik aus diesem Kapitel benutzt. Die Eigenschaften zur Implementierung eines angepassten Flanschnetzwerks nach Waibel [11] wurden für diese Klassifizierung wie in Tabelle 4.15 gewählt.

Parameter	Wert
Maximaler Strahlenwinkel α_{cone}	2π
Strahleneigenschaft p	5
Strahleneigenschaft q	5
Erweiterung AABB p_1, p_2	10,10
Regularisierung γ	0
Nachbearbeitungsparameter δ_A^0, δ_A^1	100
Nachbearbeitungsparameter δ_n^0, δ_n^1	1
Anzahl Topeigenschaften k	5

Tabelle 4.15: Gewählte Parameter des Flanschnetzwerks zur Klassifizierung von wachsgeschützen Bereichen mittels Transfer Learning

Diesen Ansatz nutzend, werden für das Training Datensätze generiert. Diese sind wie in Tabelle 4.16 zusammengesetzt. Dazu wurde wie in den vorangegangenen Implementierungen die durch das Flanschnetzwerk erstellte Klassifizierung korrigiert und als Trainingsdatensatz bereitgestellt.

Name	Bauteil	Kurze Beschreibung	Anzahl Dreiecke	Anzahl positiver Dreiecke	Anteil positiver Dreiecke
A	Tür	Vordere und hintere linke Tür mit Scharnieren.	3262407	127553	3.9%
D	Tür	Vordere und hintere linke Tür mit Scharnieren.	2939380	29884	1.0%

Tabelle 4.16: Datensatz, der zum Training des Neuronalen Netzes genutzt wird. Die Buchstaben der Fahrzeuge beschreiben keine Namen, sondern lediglich die Zugehörigkeit der Geometrien zu gemeinsamen Baureihen.

Abbildung 4.10 zeigt eine Verteilung mit deutlich mehr, nicht von Wachs geschützten Dreiecken. Wie auch bei den Klebstoff- und NAD-Klassifizierungen werden daraus Rückschlüsse auf eine geeignete Evaluierungsmethodik gezogen. Der MCC (Gleichung 2.11) wird zur Bewertung der Qualität der Prognose gewählt, da sich dieser am besten für ungleich verteilte Datensätze eignet.

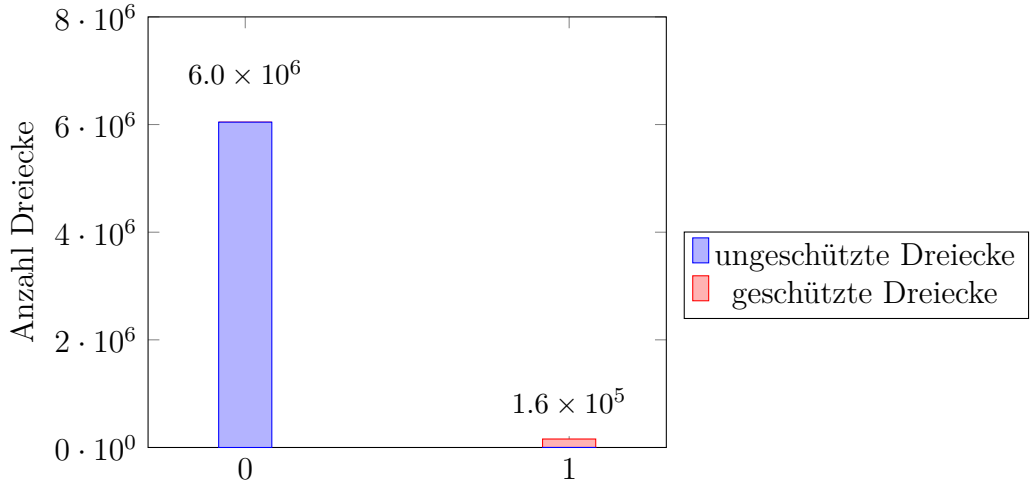


Abb. 4.10: Verteilung des gelabelten Datensatzes zur Wachs-Klassifizierung. Der Unterschied in der Größe der Klassen liegt bei Faktor 50

4.4.2 Implementierung der Eigenschaften zur Wachs-Klassifizierung

Auch für das Wachs-Klassifizierungsnetzwerk können die gleichen Eigenschaften berechnet werden wie für das Klebstoff- und NAD-Netzwerk.

Übernommene Eigenschaften aus der Flanschklassifizierung

Eigenschaften werden aus der Methodik der Flanschklassifizierung übernommen:

1. Wie bei der Klebstoffklassifizierung wird zunächst geprüft, welches der Dreiecke eines Wachs-Bauteils in der Nähe des zu klassifizierenden Dreiecks liegt. Dabei beschreiben die Parameter p_1 und p_2 den Wert, um den die Bounding Box verschoben wird. Für Dreiecke, in deren Nähe keine Wachs-Bauteile zu finden sind, werden keine Eigenschaften extrahiert. Sie werden auf null gesetzt.

2. Die bekannten Eigenschaften der Flanschsegmentierung werden benutzt:
   a) Intrinsische Eigenschaften: Die Winkel zu den Nachbardreiecken j sowie die mittlere Strahlenlänge d_m gemäß Gleichung 3.11
   b) Extrinsische Eigenschaften:
      i. Die minimale euklidische Distanz wird gemäß Gleichung 3.2 berechnet.
      ii. Die Vektorprojektion wird gemäß Gleichung 3.3 berechnet.
      iii. Das Normalenprodukt wird gemäß Gleichung 3.4 berechnet.
      iv. Die projizierte Überlappung der Dreiecksflächen wird gemäß Gleichung 3.6 berechnet.

v. Die Anzahl der Schnittpunkte wird als Eigenschaft genutzt.

vi. Die minimale Strahlenlänge d_{ij} findet Verwendung.

Übernommene Eigenschaften aus der Klebstoffsegmentierung

Folgende Eigenschaften werden genauso wie in der Klebstoffsegmentierung berechnet

1. Die Normalenaddition wird berechnet, Gleichung 4.4 wird genutzt.
2. Die Wachsbauteildicke wird berechnet, Gleichung 4.3 wird dazu genutzt.

Erweiterung der bestehenden Eigenschaften

Es werden weitere Eigenschaften eingeführt, um die Klassifizierung von wachsgeschützten Flächen zu verbessern.

1. Es existieren zwei unterschiedliche Möglichkeiten, Wachs zu dokumentieren. Dabei wird zwischen Röhren und Flächen unterschieden. Um dies in die Prognose einfließen zu lassen, wird eine Eigenschaft implementiert, die diese beiden Geometrien unterscheiden kann. Dafür werden, wie für den Klebstoff, die Durchmesser $d_{m,Wachs}$ jedes Wachsbauteils berechnet. Für die Eigenschaft np_i gilt:

$$np_i = \begin{cases} 1, & \text{if } d_{m,Wachs} > 1 \\ 0, & \text{else} \end{cases} \tag{4.17}$$

   Damit beschreibt np_i den Unterschied zwischen Wachsröhren und Wachsflächen. Dabei bedeutet 1, ein Bauteil ist ein Röhrchen, und 0, ein Bauteil ist eine Fläche.

2. Wachs beschreibt ein niedrigviskoses Medium. Damit einhergehend ist das Verhalten, dass durch Verlaufen auf Oberflächen, die bedeckte Fläche vergrößert wird. Ob ein Rohbauteil über oder unterhalb eines Wachsbauteils liegt, kann ebenfalls über die Z-Achsenwerte $z_{c_{j,\text{Wachs}}}$ der Zentrumspuntke des Wachsbauteils $c_{j,Wachs}$ und des Z-Achsenwert z_{c_i} des Rohbauzentrumspunkt c_i berücksichtigt werden:

$$\Delta z_{ij} = z_{c_{j,\text{Wachs}}} - z_{c_i} \tag{4.18}$$

4.4.3 Hyperparametertuning

Um eine optimale Parameterkombination zu finden, wird eine Rastersuche durchgeführt. Dabei wird der Parameterraum, wie in Tabelle 4.17 definiert. r beschreibt, ob die beiden neu eingeführten Parameter genutzt werden. 0 bedeutet, sie nicht zu nutzen, 1 bedeutet, sie zu nutzen. Die bestimmten Werte sind in Tabelle 4.17 der zweiten Spalte gezeigt.

Parameter	Wertraum	Bestimmter Wert
Strahlenanzahl p	[4; 16]	4
Strahlenanzahl q	[4; 16]	4
Maximaler Strahlenwinkel α_{cone}	$[\pi/8; 2\pi]$	2π
Maximale Strahlenlänge	[10; 40]	10
Anzahl Top-Eigenschaften k	[5; 10]	5
Anzahl der Netzwerkschichten	[3; 5]	5
r	[0; 1]	1

Tabelle 4.17: Parameterraum und durch die Rastersuche ermittelte Hyperparameterwerte für das Wachsklassifizierungsnetzwerk

Das Netzwerk mit den bestimmten Werten wird mit dem Trainingsdatensatz aus Tabelle 4.16 trainiert. Dieses Training ergibt eine Validierungsgenauigkeit, die in Tabelle 4.18 gezeigt sind.

Metrik	**TPR**	**TPA**	**ACC**	**MCC**
Wert	0.9680	0.9700	0.9630	0.9230

Tabelle 4.18: Validierungsgenauigkeit des trainierten Wachs-Netzwerks

4.4.4 Klassifizierungsergebnisse

Im Folgenden wird das ermittelte Netz auf einen dem Netz unbekannten Datensatz angewendet und validiert. Dazu wird der gleiche Gesamtfahrzeugsdatensatz genutzt, der auch in den vorherigen Kapiteln genutzt wird. Wie auch bei den anderen Klassifizierungen wird der Datensatz korrigiert und ausgewertet. Daraus ergibt sich eine Konfusionsmatrix (Tabelle 4.19).

	Positive ungewichtete Vorhersage (1)	**Negative ungewichtete Vorhersage (0)**
Positives ungewichtetes Label (1)	367375 (1,93%)	96 (0,00%)
Negatives ungewichtetes Label (0)	12006 (0,06%)	18659009 (98,01%)

Tabelle 4.19: Konfusionsmatrix für die zwei Klassen 0 und 1

Jede Vorhersage kann parallel zu dem Ansatz der Auswertung der Klebstoffklassifizierung gewichtet werden. Dabei wird die Gewichtung auf Basis der

Fläche jedes vorhergesagten Dreiecks implementiert. Daraus ergibt sich eine gewichtete Konfusionsmatrix (Tabelle 4.20).

	Positive gewichtete Vorhersage (1)	**Negative gewichtete Vorhersage (0)**
Positive gewichtete Label (1)	3417257 (3,38%)	844 (0,00%)
Negative gewichtete Label (0)	120058 (0,12%)	97432549 (96,50%)

Tabelle 4.20: Über die Fläche der einzelnen Dreiecke gewichtete Konfusionsmatrix für die zwei Klassen 0 und 1

Aus Tabelle 4.19 und Tabelle 4.20 kann der MCC berechnet werden. Die in Kapitel 2.1 beschriebenen Genauigkeiten ergeben sich aus Tabelle 4.21.

Genauigkeit	TPA	TPR	F-Maß	ACC	MCC
Gewichtet	0.9661	0.9998	0.9826	0.9988	0.9822
Ungewichtet	1.0000	0.9994	0.9997	0.9994	0.9836

Tabelle 4.21: Genauigkeiten der Wachsklassifizierung des Gesamtfahrzeugs, basierend auf der gewichteten und der nicht gewichteten Konfusionsmatrix.

Die Klassifizierung des Gesamtfahrzeugs, auf dem die Genauigkeitsmaße basieren, ist in Abbildung 4.11 visualisiert. Auch im Bereich der Wachsklassifizierung liefert die Methodik sehr gute Genauigkeiten.

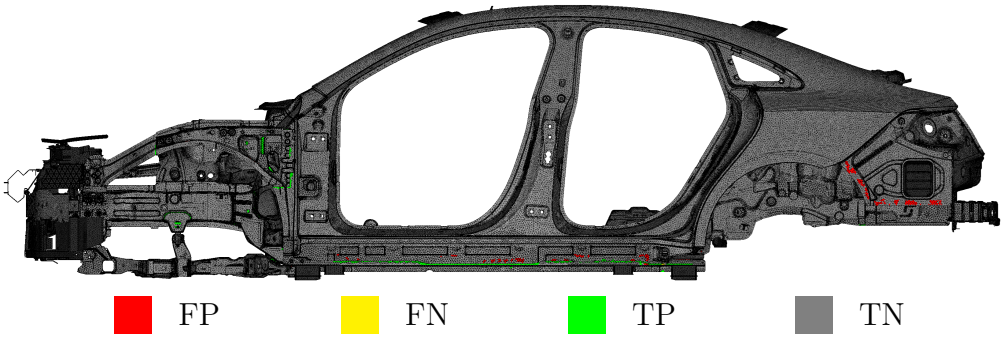
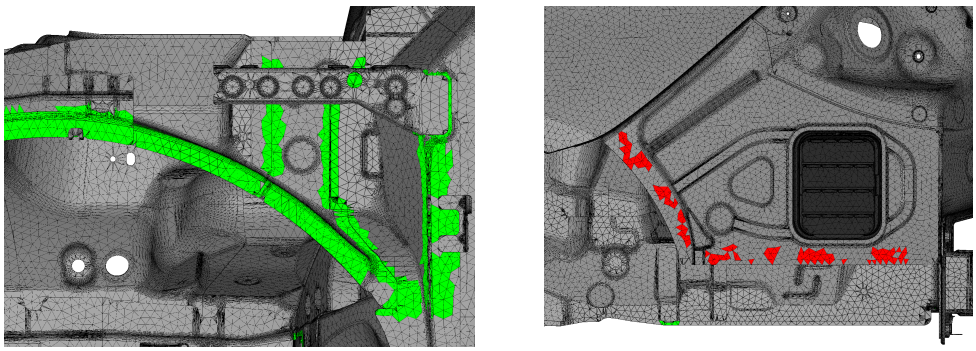


Abb. 4.11: Ergebnis der Wachs-Klassifizierung eines Gesamtfahrzeugs

Abbildung 4.12 zeigt Ausschnitte aus der Klassifizierung des Gesamtfahrzeugs. Dabei ist Abbildung 4.12a ein Beispiel für einen sehr großen Bereich, in dem

die Wachsklassifikation sehr gute Ergebnisse liefert. Abbildung 4.12b zeigt dagegen Bereiche des Gesamtfahrzeugs, die falsch als wachsgeschützt klassifiziert wurden. Eine genauere Analyse dieses Fehlerbilds wird in dem Kapitel 4.5 durchgeführt. In diesem Bereich wird das Wachs jedoch hauptsächlich durch Flächen und nicht durch Röhren dokumentiert.



(a) Beispiel für ein sehr gut klassifizierten Bauteilbereich

(b) Beispiel für einen Bauteilbereich mit Fehlklassifizierungen

Abb. 4.12: Ausschnitte der Ergebnisse der Wachsklassifizierung

Zusammenfassend lässt sich auch für die Wachsklassifizierung bestätigen, dass die vorgestellte Methodik sehr gute Ergebnisse in der Klassifizierung von Gesamtfahrzeugen liefert.

4.5 Analyse der Netzwerke mittels Explainable AI

Im Folgenden werden die vorgestellten Netzwerke mittels xAI analysiert. Dazu werden die Shapley-Werte der einzelnen Eigenschaften jedes Netzwerks berechnet. Dabei werden am Beispiel der Flanschklassifizierung Änderungsvorschläge eingeführt und validiert. Für die Netzwerke der Korrosionsschutzmaßnahmen werden Handlungsempfehlungen formuliert.
Es werden zwei Arten von Auswertungen der Shapley-Werte berücksichtigt. Die mittleren absoluten Werte jedes einzelnen Inputs beschreiben die grundsätzliche Relevanz einer Eigenschaft. Je höher der durchschnittliche Shapley-Wert, desto höher die Relevanz der Eigenschaft für die Klassifizierung eines Dreiecks. Je niedriger dieser Wert, desto geringer der Beitrag dieser Eigenschaft zum Klassifizierungsergebnis. In diesem Kapitel werden Eigenschaften mit einer Zahl markiert. Durch die Implementierung ergeben sich mehrere der gleichen Eigenschaften. Diese Eigenschaften werden dann mit einer Zahl in Klammern unterschieden. Vektorprojektion A (2) beschreibt also die Vektorprojektion A der 2. Eigenschaftssets.

4.5.1 Analyse und Anpassung der Flanschklassifizierung mittels Shapley-Werten

In einem ersten Schritt werden die Shapley-Werte für das Flanschnetzwerk berechnet. Die höchsten (Abbildung 4.13) und geringsten (Abbildung 4.14) Werte sind abgebildet. Aus Abbildung 4.14 lässt sich erkennen, dass die Eigenschaft der Vektorprojektion (Gleichung 3.3) geringen Einfluss auf die Klassifizierungsergebnisse haben.

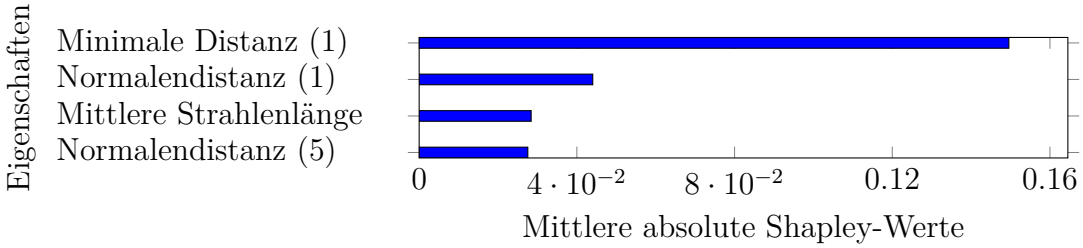


Abb. 4.13: Mittlere absolute Shapley-Werte der vier wichtigsten Flanscheigenschaften

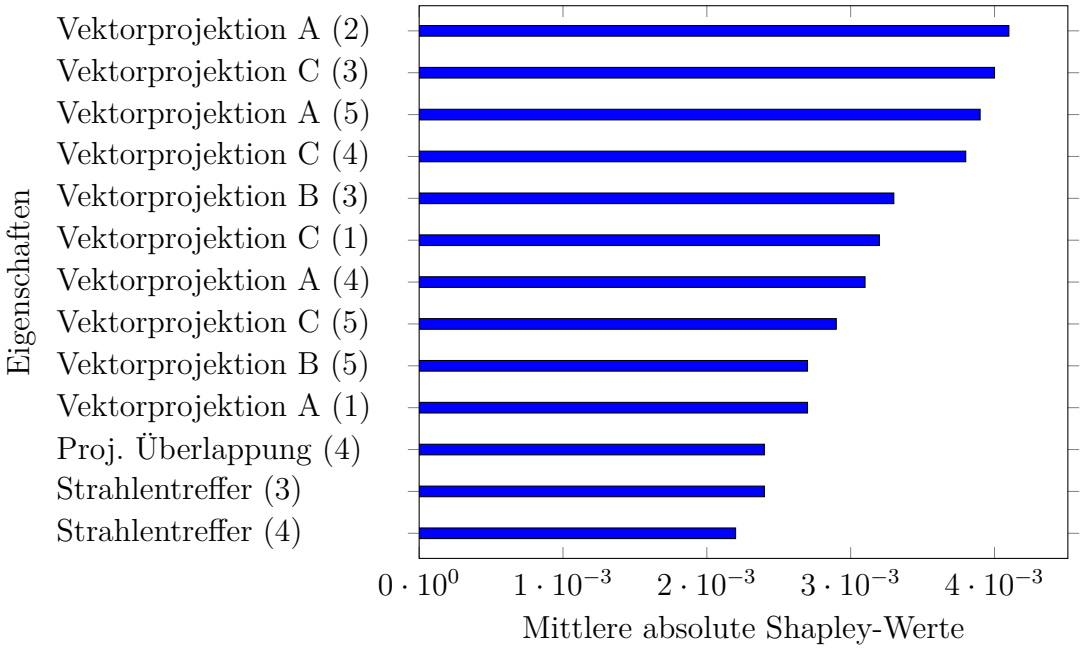


Abb. 4.14: Mittlere absolute Shapley-Werte der am wenigsten relevanten Flanscheigenschaften

Um Ressourceneffizienz sicherzustellen, wird das Netzwerk ohne diese Eigenschaften neu trainiert.
Das neu trainierte Netzwerk wird mit dem ursprünglichen Netzwerk verglichen (Tabelle 4.22). Die reduzierte Anzahl der Eigenschaften entsteht durch das Wegfallen der Vektorprojektion. Dies bedeutet ebenfalls eine reduzierte

Anzahl an Parametern. Wie erwartet reduziert sich die Genauigkeit des Netzwerkes um 0.0007 Punkte im MCC, da dem Netzwerk weniger Informationen zur Verfügung gestellt werden. Dadurch finden zwei Reduktionen statt: die Trainingszeit wird auf 82% reduziert. Das Netzwerk wird auf 81% der ursprünglichen Größe reduziert. Beides ist ohne nennenswerte Einbrüche in der Genauigkeit möglich.

Modell	**MCC**	**Trainingszeit**	**Eigenschaften**	**Trainierbare Parameter**
Alt	0,9537	460,60s	49	4861
Neu	0,9530	379,30s	34	3961

Tabelle 4.22: Vergleich der beiden Netzwerke

Die verkürzte Trainingszeit spiegelt sich auch im Verlauf der Trainingsgenauigkeit wieder.

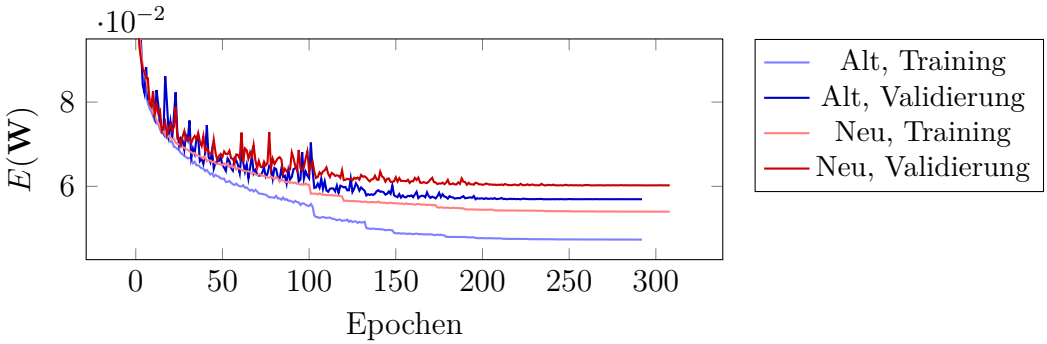


Abb. 4.15: Trainings- und Validierungsgenauigkeit des neuen und alten Modells

Abbildung 4.15 zeigt die Trainingsgenauigkeit und Validierungsgenauigkeit des alten und des angepassten, verkleinerten Modells. $E(W)$ beschreibt dabei den Loss. Aus dieser Abbildung wird auch die verkürzte Trainingszeit deutlich. Das Training bricht aufgrund von Abbruchkriterien einige Epochen früher ab. Darüber hinaus ist die Zeit, die für eine Epoche benötigt wird, aufgrund der zu trainierenden Parameter ebenfalls verringert. Daraus resultiert die oben genannte Zeitreduktion.
Aus Abbildung 4.16 lässt sich ein weiterer Vorteil der Reduktion ablesen. Beim Training von Neuronalen Netzen besteht die Gefahr, lediglich auf den Trainingsdatensatz, nicht aber auf das eigentliche, verallgemeinerte Problem zu trainieren. Dies wird in dieser Arbeit mit der großen Menge an verfügbaren Daten adressiert. Für das Flanschnetzwerk kann diese Gefahr ebenfalls reduziert werden. Abbildung 4.16 zeigt die Differenz aus Validierungs- und Trainingsgenauigkeit. Je höher diese Differenz, desto höher ist die Gefahr des Overfittings. Die geringere Differenz zwischen Trainings- und Validierungsgenauigkeit zeigt damit den großen Vorteil der Netzreduktion, eine robustere, besser verallgemeinernde Prognose.

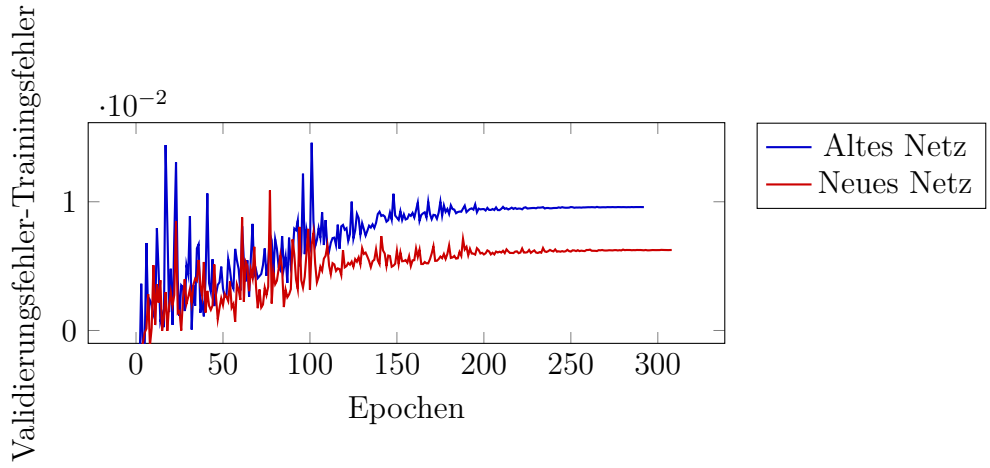


Abb. 4.16: Differenz zwischen Trainings- und Validierungsfehler des neuen und alten Modells

4.5.2 Analyse der NAD-Klassifizierung mittels Shapley-Werten

Wie auch für das Flanschnetzwerk wird das NAD-Netzwerk über Konzepte des xAI analysiert. Dabei wird das Konzept der Shapley-Werte genutzt. In Abbildung 4.17 sind die Shapley-Werte der wichtigsten Parameter aufgeführt. VA steht dabei für Vertexabstand, NVProdukt für das Normalenvektorprodukt. NRichtung steht für Normalenrichtung.

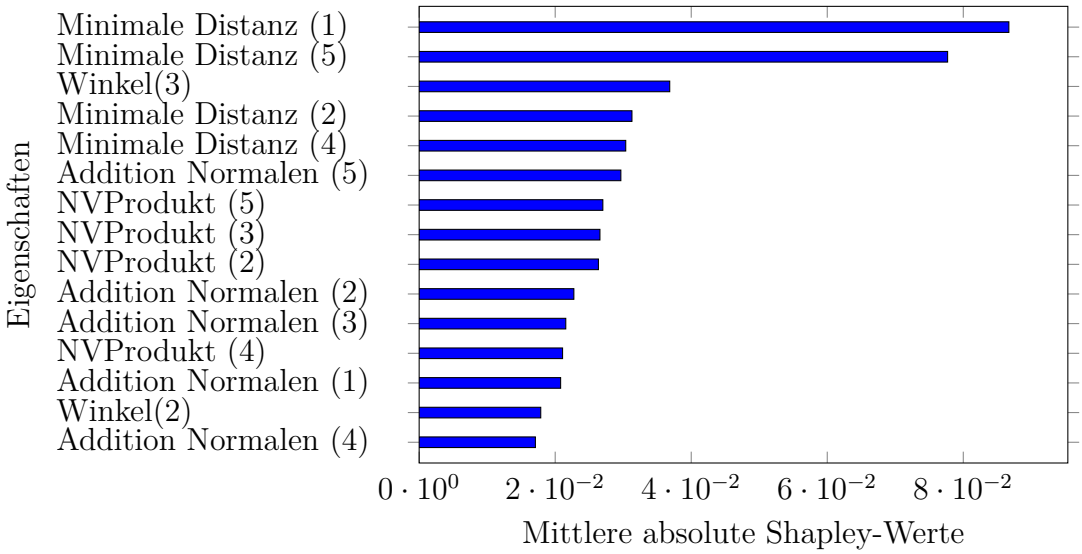


Abb. 4.17: Die 15 mittleren, absoluten Shapley-Werte eines Netzwerks zur NAD-Klassifizierung von Dreiecken mit der höchsten Rekevanz

In Abbildung 4.18 sind die Shapley-Werte eines jeden Parameters gezeigt,

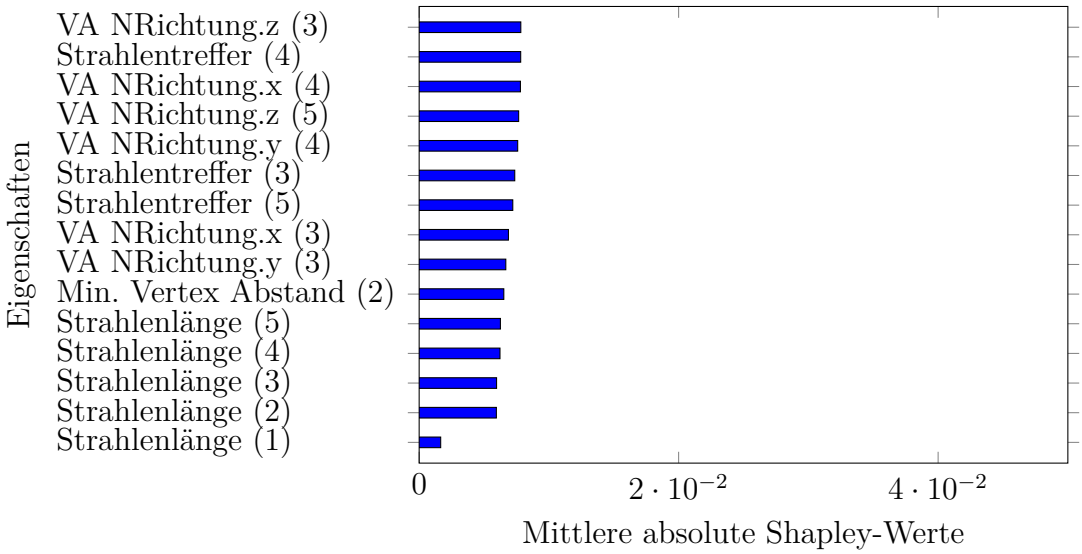


Abb. 4.18: Die 15 mittleren absoluten Shapley-Werte eines Netzwerks zur NAD-Klassifizierung von Dreiecken mit der niedrigsten Relevanz

Diese Werte zeigen eine hohe Bedeutung der Strahlenfeatures für die Prognose von NAD-geschützten Flächen. Die relevanteste Eigenschaft ist jedoch die minimale Distanz zwischen dem am nächsten liegenden NAD-Dreieck und dem Rohbauteil. Die Dokumentation des NAD im digitalen Datensatz beschreibt die Oberfläche des aufgetragenen Materials. Liegt diese Oberfläche nah an einem Bauteil, so ist es von NAD bedeckt und geschützt. Darüber hinaus ist die neu eingeführte Normalenadditionseigenschaft eine relevante Eigenschaft, da es mit einem Shapley-Wert von 0.03 der sechstgrößte Shapley-Wert ist. Da die NAD-Dokumentation eine Bauraumdokumentation ist und es dadurch zu Durchdringungen zwischen dem NAD- und Rohbauteil kommen kann, ist zu vermuten, dass diese Normalenaddition die Identifizierung dieser Durchdringungen unterstützt und die Prognose verbessert.
Aus den niedrigsten Shapley-Werten (Abbildung 4.18) wurden im Besonderen Maßnahmen zur Optimierung des Netzwerks abgeleitet. Die Eigenschaft *Strahlenlänge* aller fünf Strahlen trägt am wenigsten zu der Klassifikationsgenauigkeit bei. In einem laufzeitoptimierten Netzwerk ist die Strahlenlänge als Eigenschaft wegzulassen, ohne relevante Einbußen in der Genauigkeit hinnehmen zu müssen.

4.5.3 Analyse der Klebstoffklassifizierung mittels Shapley-Werten

In Abbildung 4.19 sind die 25 relevantesten Eigenschaften und deren Shapley-Werte dargestellt. NDistanz steht dabei für Normalendistanz. NVProdukt steht für das Normalenvektorprodukt.

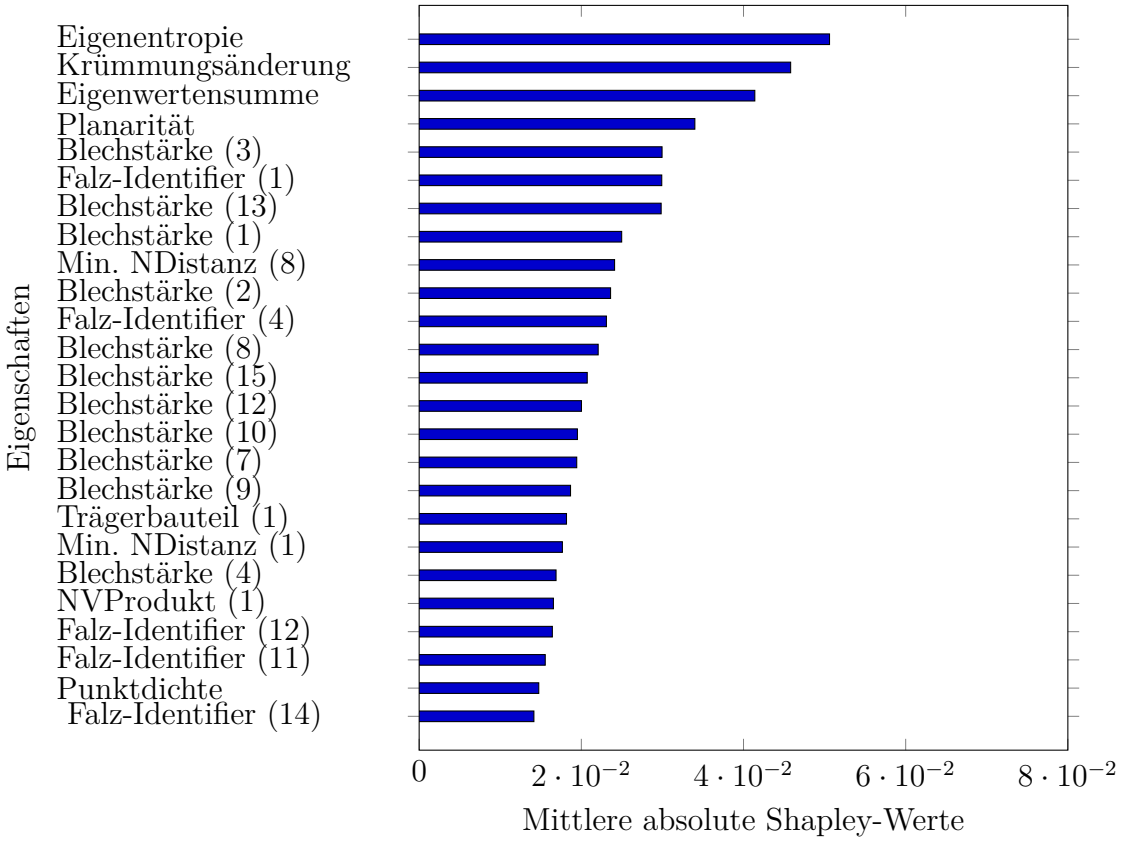


Abb. 4.19: Die 25 mittleren, absoluten Shapley-Werte eines Netzwerks zur Kleber-Klassifizierung von Dreiecken mit der höchsten Relevanz

Diese bestätigen die Bedeutung der hinzugefügten Eigenschaften. Die ersten vier Parameter sind Eigenschaften, die die Geometrie durch die Eigenvektoren beschreiben. Darüber hinaus besitzen die Durchmesser von Klebstoffbauteilen und Rohbauteilen einen großen Einfluss, genauso wie die aus der Flanschklassifizierung übernommenen Eigenschaften. Eine weitere Information, die die Shapley-Werte liefern, ist die Bestätigung für ein hohes k, also eine hohe Zahl an zu betrachtenden Eigenschaftssets. Auf Basis der am wenigsten relevanten 25 Eigenschaften kann keine eindeutige Tendenz ausgemacht werden (Abbildung 4.20). Dabei bedeutet NDistanz Normalendistanz und Normalenvektorprod. Normalenvektorprodukt. Die Eigenschaft Trägerbauteil für $k = 8$ ist zwar eine sehr schwache Eigenschaft, da die Eigenschaft für $k = 1$ jedoch unter den 25 relevantesten Eigenschaften ist, bestätigt dies die Relevanz der Eigenschaft und des bestimmten Werts k. In Kapitel 4.2 werden Eigenschaften implementiert. Diese Eigenschaften sind jedoch nicht alle gleich relevant (Abbildung 4.21). Grundsätzlich gilt , dass alle Eigenschaften das Ergebnis der Prognose verbessern. Die Eigenschaften, größte Entfernung und minimale Höhe sowie maximale Höhe der Punkte beschreiben Eigenschaften, die für die Prognose nur wenig Relevanz haben. Eine Reduktion aus Laufzeitgründen bietet sich nicht an. Die Reduktion dieser Eigenschaften reduziert das Netzwerk lediglich um 1. In der Berechnung der Eigenschaft ist der ressourcenintensive

Teil die Berechnung der Kovarianzmatrix. Eine Reduktion dieser Eigenschaften wird daher nicht vorgenommen.

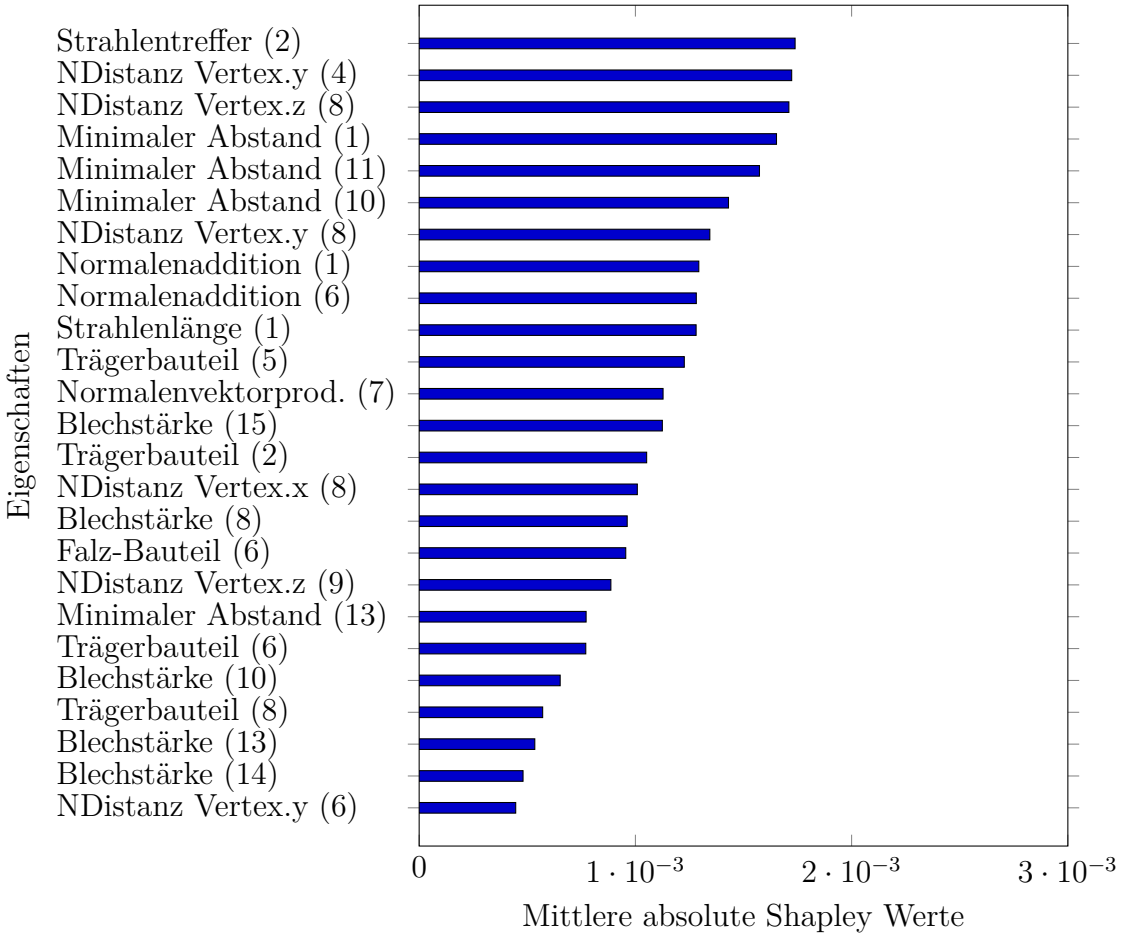


Abb. 4.20: Die 25 absoluten mittleren Shapley-Werte eines Netzwerks zur Klebstoffklassifizierung von Dreiecken mit der niedrigsten Relevanz

4.5.4 Analyse der Wachsklassifizierung mittels Shapley-Werte

Abbildung 4.22 zeigt die Relevanz der einzelnen Eigenschaften, die in der Wachsklassifizierung genutzt werden. Dabei beschreibt NDistanz die Normalendistanz, NAbstand den Normalenabstand.
Bei der Betrachtung der Shapley-Werte liegt der Fokus auf den neu implementierten Eigenschaften aus Kapitel 4.4. Diese Eigenschaften sind einerseits die Dicke, die Z-Differenz und der Identifier, der bestimmt, ob ein Wachsbauteil eine Röhre, oder eine Fläche ist.
Besonders auffallend ist die hohe Bedeutung der Eigenschaft Dicke. Dicke (1) ist dabei die am 4. relevantesten Eigenschaft. Dabei ist auffällig, dass die Dicke von 4 Eigenschaftssets dabei unter den höchsten 15 Eigenschaften ist. Dies kann darin begründet sein, dass die Dicke der Bauteile einen starken Einfluss

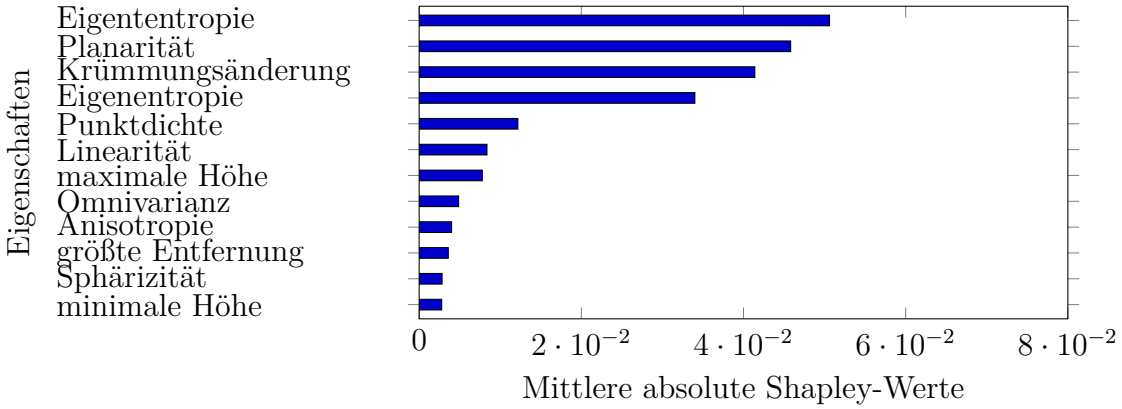


Abb. 4.21: Die Shapley-Werte der Eigenschaften aus der Klebstoffklassifizierung die geometrische Einflüsse berücksichtigen

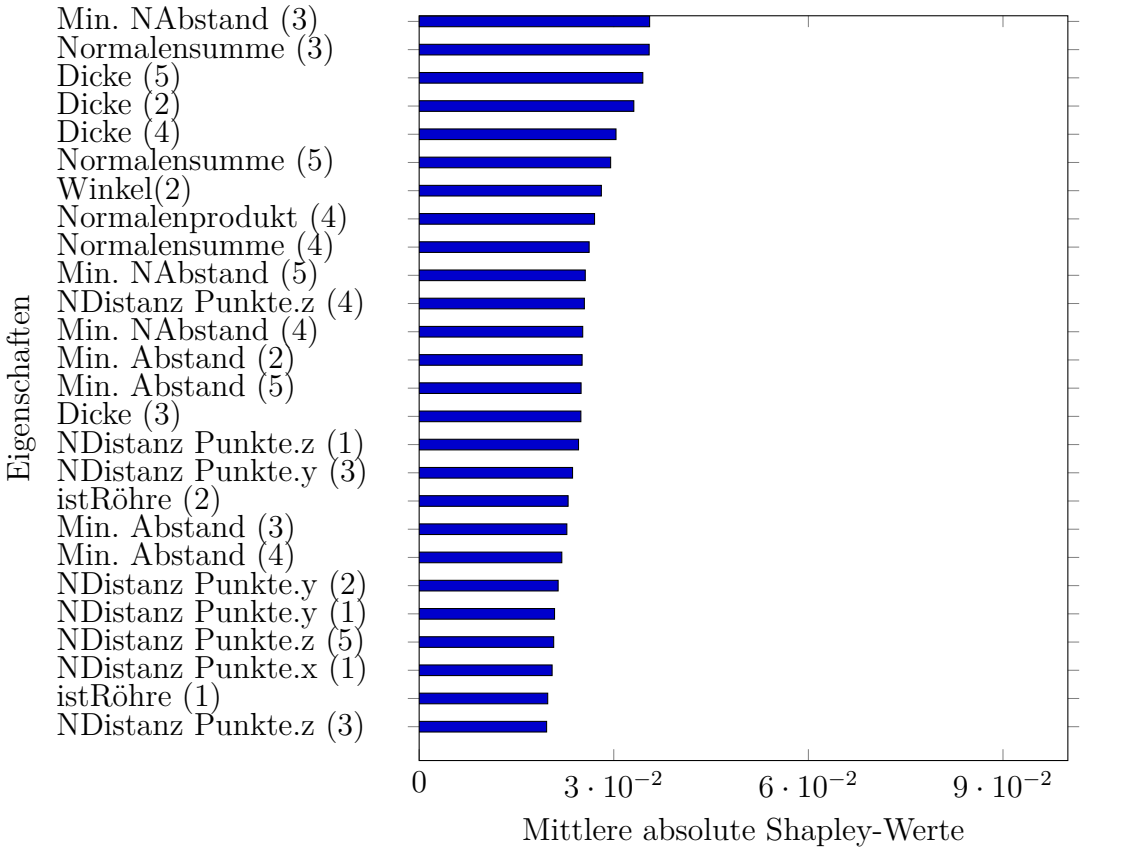


Abb. 4.22: Die 25 absoluten mittleren Shapley-Werte des Netzwerks zur Wachs-Klassifizierung von Dreiecken mit der höchsten Relevanz

darauf hat, wie die anderen Bauteile aussehen. Das Feature *istRöhre* hat ebenfalls einen relevanten Einfluss auf die Wachs-Klassifizierung. Auch hier gilt die gleiche Herleitung der Relevanz. Die berechneten Eigenschaften, beispielsweise das Normalenprodukt, verändern sich, wenn ein Bauteil eine Röhre oder eine Fläche ist. Die Relevanz ist jedoch nicht so hoch, da beispielsweise der Abstand nicht grundsätzlich von der Form des Wachsbauteils abhängig ist. Die Differenz in Z-Richtung zwischen Wachs- und Rohbauteil findet sich nicht unter den 25 höchsten Eigenschaften. Die zwei höchsten Shapley-Werte der Z-Differenz-Eigenschaft liegen mit 0,01- 0,02 im Mittel der Relevanz. Damit bestätigen die Shapley-Werte die Relevanz der implementierten Features der Eigenschaften *istRöhre*, der Dicke des Bauteils und der Z-Differenz.

5 Auswertung der Elektrolytbeanspruchung

Im Folgenden wird der Ansatz beschrieben, wie die Einflüsse der Elektrolytbeanspruchung in das Informationsmodell integriert werden. Dazu wird eine Simulation der Elektrolytbeanspruchung behandelt. Der Aufbau sowie der daraus resultierende Datensatz werden erläutert. Dabei wird die Elektrolytsimulation auf die Kanten- und Flanschelemente bezogen. Anschließend wird die Simulation auf die Oberflächenelemente übertragen. Die Definition eines Korrosionsrisikolabels auf Basis der Ergebnisse wird beschrieben. Final werden die Simulationswerte im Kontext der Prognose interpretiert und in Zusammenhang gebracht. Die in diesem Kapitel vorgestellte Methodik wird am Beispiel des Längsträgers in Abbildung 5.1 vorgestellt.

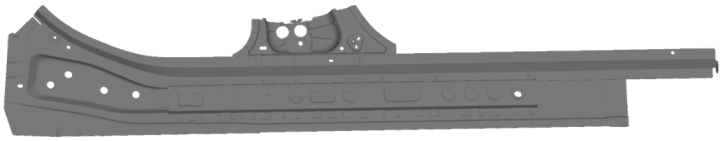


Abb. 5.1: Im Rahmen der Arbeit analysiertes Bauteil

5.1 Betrachtung der Elektrolytbeanspruchung mittels Simulationen

Wie in Kapitel 2.3 dargestellt, hat die Elektrolytbeanspruchung einen großen Einfluss auf das Korrosionsverhalten. Daher ist dieser Einfluss in der Vorhersage zu berücksichtigen. Dies geschieht durch die Integration einer bestehenden Smoothed Particle Hydrodynamics Simulation (SPH). Dazu werden in diesem Kapitel in einem ersten Schritt die Randbedingungen und Modelle von Wickert [65] beschrieben, die im Rahmen dieser Arbeit genutzt werden. Darauf aufbauend wird ein Ansatz dargestellt, der es ermöglicht, die Simulationswerte auf die in Kapitel 3.1 beschriebenen Korrosionsrisikostellen zu beziehen.

5.1.1 Randbedingungen und Berechnungsmodelle der verwendeten Elektrolytsimulationen

Die vorliegende Arbeit verwendet Ergebnisse aus der Simulationssoftware PreonLab der Fifty2 Technology GmbH. Diese ist speziell für die Automobilindustrie entwickelt worden, um numerische Simulationen zu erstellen.

Physikalische Modellierung und Simulationsparameter

Wickert [90] beschäftigt sich mit der Erstellung und Validierung eines Strömungsmodells mithilfe der SPH-Methode. Im Gegensatz zu klassischen gitterbasierten CFD-Verfahren verwendet SPH eine partikelbasierte, gitterfreie Methode, die insbesondere bei komplexen Geometrien Vorteile bietet. Zur Parametrisierung des Modells wurden verschiedene Versuche durchgeführt:

1. Die Partikelgröße wurde über einen Trichterversuch ermittelt.
2. Die Adhäsion und Kohäsion als Parameter wurden mithilfe eines Tropfenprüfstands bestimmt.
3. Die Fluidreibung und die dynamische Viskosität wurden mit einem Marshtrichter ermittelt.

Wickert [65] validiert die Simulationseinstellungen durch experimentelle Daten und Literaturwerte und stellt sicher, dass die physikalischen Effekte realitätsnah abgebildet werden.

5.1.2 Beschreibung der Simulationsergebnisse

Die Ergebnisse der SPH-Simulation werden in Form einer Punktewolke $PC_{Elektrolyt}$ mit $x_i; y_i; z_i \in PC_{Elektrolyt}$ und $p_{i,\,Elektrolyt} = \{x_i; y_i; z_i\}$ mit zugeordneten Werten τ_i ausgegeben. Um eine Prognose unter Berücksichtigung der Elektrolytbeanspruchung durchführen zu können, müssen diese Simulationswerte auf die Korrosionsrisikostellen bezogen werden. Ferner sind Einschränkungen aus der Elektrolytsimulation zu berücksichtigen. Die existierenden Daten der beschriebenen Time of Wetness (ToW) können wie in Abbildung 5.2 visualisiert werden. Dabei ist die ToW der Anteil der Simulationszeit, in dem ein beliebiges Simulationspartikel in der Nähe des jeweiligen Oberflächenbereichs ist. Ist ein Wert am Längsträger also rot, so bedeutet dies, dass während beinahe der gesamten Simulationsdauer der betreffende Bereich durch Elektrolyt bedeckt war.
Drei unterschiedliche Simulationen des Gesamtfahrzeugs werden im Rahmen dieser Arbeit genutzt:

1. Wasserdurchfahrt: Dabei wird die Durchfahrt eines Gesamtfahrzeugs simuliert. Ein Fahrzeug fährt durch ein mit Partikeln gefülltes Becken, mit einer Partikelquelle vor dem Fahrzeug.
2. Beregnung: Oberhalb des Fahrzeugs wird eine Partikelquelle platziert und das Fahrzeug wird mit Partikeln beregnet.

3. Türberegnung: Für die Tür wird eine gesonderte Simulation implementiert. Dafür wird oberhalb einer einzelnen Tür eine Regenquelle positioniert.

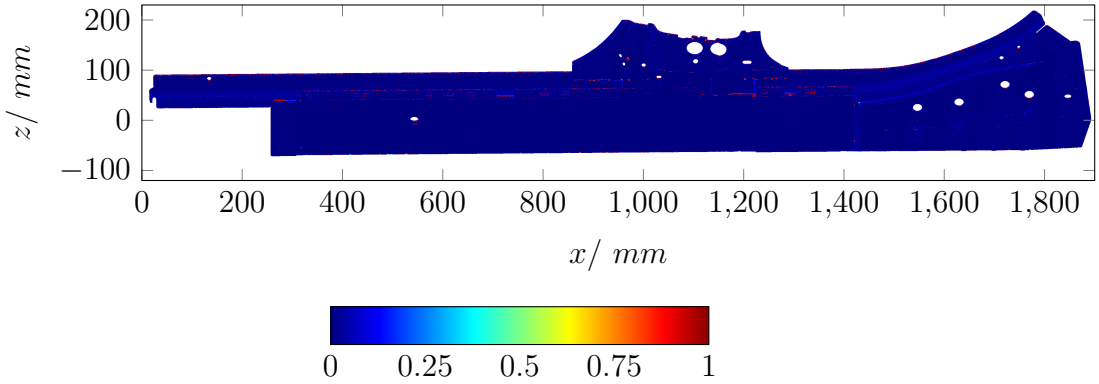


Abb. 5.2: Visualisierung der Simulationsergebnisse der Time of Wetness

5.2 Übertrag der Simulationswerte

Für eine realitätsnahe Korrosionsprognose von Bauteilkonstruktionen müssen die Ergebnisse der zuvor beschriebenen Elektrolytsimulation auf die digitalen Konstruktionsdaten übertragen werden. Dazu wird die ToW der Elektrolytsimulationen aus Beregnung, Wasserdurchfahrt und Türberegnung genutzt.

5.2.1 Übertrag auf Kantenelemente

Um die Elektrolytbeanspruchung eines Kantenelements abschätzen zu können, wird eine Object Bounding Box (OBB) um das jeweilige Element gelegt. Wie in Abbildung 5.3 visualisiert, werden die Punkte der Simulation, die innerhalb dieser OBB liegen, zur Bewertung der Elektrolytbeanspruchung herangezogen. Die Punktemenge einer Simulation ist jedoch für einen einfachen euklidischen Vergleich zu groß. Daher muss vorher für jedes Element eine Reduktion des Rechenaufwands durchgeführt werden. Um dies zu erreichen, wird ein Octree implementiert. Ein Octree ist eine hierarchische Datenstruktur, die den dreidimensionalen Raum rekursiv in gleich große Teile unterteilt. Jeder Knoten repräsentiert dabei einen Raumwürfel, der entweder weitere Unterteilungen enthält oder Datenpunkte speichert [91]. Damit kann der Berechnungsbedarf stark reduziert werden, indem der rote Bereich aus Abbildung 5.3 durch Raumwürfel definiert wird, statt alle darin enthaltenen Punkte zu untersuchen. Auf Basis der OBBs werden die Punkte von $PC_{Elektrolyt}$ gesucht, welche in der Nähe des gesuchten Elements, in diesem Fall der Kante, liegen. Diese Punkte $p_{Elektrolyt_{BB}} \in PC_{Elektrolyt}$, werden benutzt, um den Wetting-Wert τ_i der Kante j zu berechnen. Damit ergibt sich der Wetting-Wert wie folgt:

$$wt_{j,Kante} = \max \tau_i; i \in p_{Elektrolyt_{BB}} \tag{5.1}$$

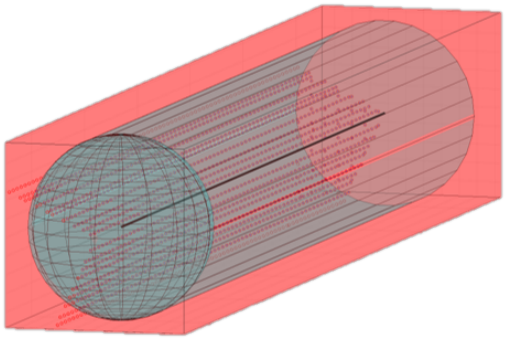


Abb. 5.3: Visualisierung der Object Bounding Box (OBB) am Beispiel einer Kante. Rot beschreibt den durch den Octree definierten Raum. Das graue Volumen beinhaltet alle Punkte, die für den Wetting-Wert berücksichtigt werden.

5.2.2 Übertrag auf Flanschelemente

Parallel zu dem Ansatz für die Kantenelemente wird auch für die Flanschelemente ein Octree implementiert. Dieser Octree wird ebenfalls wie bei der Kantenimplementierung genutzt, um eine OBB um ein Flanschelelement zu bilden, und ermöglicht einen laufzeitoptimierten Ansatz. Damit berechnet sich die Wetting-Time eines Flansches k wie folgt:

$$wt_{k,Flansch} = \max \tau_i; i \in p_{Elektrolyt_{BB}} \tag{5.2}$$

τ_i sind dabei die Wetting-Werte der Sensorpunkte, die in der OBB eines Kantenelements k liegen.

5.3 Limitationen der Elektrolytsimulation

Um die Simulationswerte in die Prognose integrieren zu können, müssen deren Limitationen bestimmt werden, um diese berücksichtigen zu können.

5.3.1 Limitationen der Auswertung durch Geometrievernetzung

Bei der Auswertung der SPH-Simulationen treten drei Herausforderungen auf:

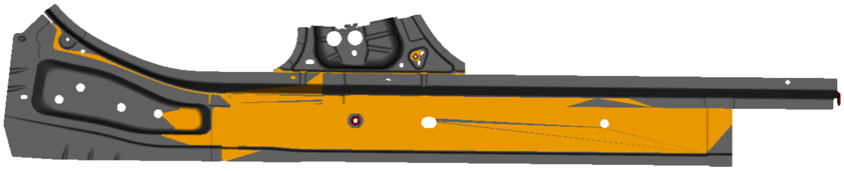
1. In der Realität existieren zwischen Karosserieteilen mikroskopische Spalte, durch die Wasser eindringen kann. Im digitalen CAD-Modell sind diese Spalte jedoch nicht vorhanden – alle Flächen sind ideal dicht. Dadurch können die simulierten Fluidpartikel in PreonLab nicht zwischen die Flansche eindringen.

2. Korrosionsrisikostellen müssen so groß sein, dass sie zuverlässig aus Flanschbereichen herausragen. Bildet eine Korrosionsrisikostelle perfekt

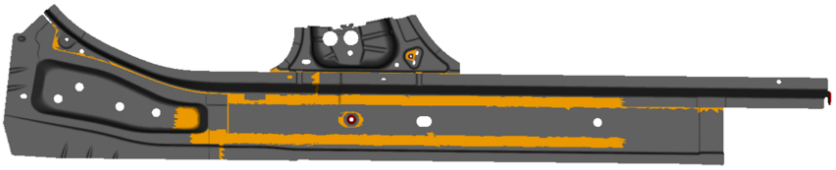
einen Flansch ab, so kann kein Partikel in die Nähe dieser Korrosionsrisikostellen geraten, da Partikel nicht in Flansche penetrieren können. Daher müssen die Dreiecke der Korrosionsrisikostellen außerhalb physisch geschlossener Bereiche liegen, um gemäß Gleichung 5.2 einen Wetting-Wert berechnen zu können.

3. Gleichzeitig darf die Flächen-Diskretisierung nicht zu grob sein. Zu große Dreiecke würden das Korrosionsrisiko unpräzise abbilden. Kleine Dreiecke erlauben eine feinere Auflösung, was für die Güte der Risikoprognose relevant ist.

Abbildung 5.4a ist ein Beispiel für eine Geometrie, welche durch ihre großen Bereiche einen Übertrag der ToW-Werte in den Flansch ermöglicht, aber eine schlechtere Segmentierung beinhaltet.



(a) Klassififizierungsergebnisse einer Flanschsegmentierung ohne Geometrieneuvernetzung



(b) Klassifizierungsergebnisse einer Flanschsegmentierung mit Geometrieneuvernetzung

Abb. 5.4: Visualisierung der Segmentierungsergebnisse. Grau: Dreiecksflächen, die nicht Teil eines Flansches sind. Orange: Dreiecksflächen, die Teil eines Flansches sind.

5.3.2 Korrektur der Elektrolytsimulation in Flanschbereichen

Die Elektrolytsimulation ermöglicht eine quantitative Abschätzung, welche Bereiche elektrolytbeansprucht sind. Dabei treffen zwei Umstände aufeinander. Bei der CFD-Partikelsimulation werden Partikel als Diskretisierung benutzt (Kapitel 2.3.3). Diese Partikel besitzen einen Durchmesser $d_m > 0$. In dem digitalen Datensatz, der den Rohbau abbildet, finden sich Flansche, denen im Rahmen dieser Arbeit ein Korrosionsrisiko zugeordnet wird. Nach Wünsche [8] besitzen gefügte Flansche Blechabstände zwischen 0,1 mm und 0,2 mm. Im digitalen Datensatz werden diese Flansche jedoch als direkt aufeinanderliegend

und damit ohne jeglichen Abstand zueinander konstruiert. Durch die Partikelgröße können daher die CFD-Simulationspartikel nicht in diese Flansche eindringen. Das Ergebnis einer einfachen Auswertung der Simulationsergebnisse würde daher in Flanschen ToW-Werte von null ergeben, obwohl diese in der Realität durchaus von Elektrolyt benetzt sind. Daher benötigt die Auswertung dieser Simulationen einen weiteren Schritt, um diese Ergebnisse anzupassen. Dabei wird ein Algorithmus gewählt, der die Flanschbereiche korrigiert. Dieser wird in den folgenden Kapiteln beschrieben.

5.3.3 Korrelation des Elektrolytverhalten zwischen Simulation und Korrosionstests

Eine Herausforderung der Simulationsinterpretation ist der Zusammenhang zwischen den Simulationswerten, den Wetting-Werten und den Benässungszeiten der Realtests. Der Simulationszeitraum bewegt sich im Bereich von Minuten, während die Realtests über Wochen durchgeführt werden. Um eine Vorhersage des Testverhaltens zu ermöglichen, wird eine Korrelation zwischen dem Elektrolytverhalten der Simulation und dem Korrosionstest gesucht. Ein wesentlicher Grund für Abweichungen zwischen beiden Ansätzen liegt in der Definition der ToW in der Simulation. Bei einer Partikelquelle, die senkrecht von oben einwirkt, sinken die ToW-Werte mit zunehmendem Abstand vom Eintrittspunkt, da die Partikel längere Wege zurücklegen müssen. Dadurch weisen Sensorpunkte in größerer Entfernung von der Quelle geringere ToW- und folglich niedrigere Wetting-Werte auf.
In der Literatur wird dennoch eine grundsätzliche Übereinstimmung zwischen Simulation und Realversuchen bestätigt. Sowohl Wickert [65] als auch Krei [92] bestätigen jeder die Übereinstimmung zwischen Simulation und Realversuchen. Dabei betrachtet Krei die beschriebene Übereinstimmung zwischen Simulation und Realversuch über die Zeit.

5.4 Erweiterung des Übertrags unter Berücksichtigung der Simulationslimitationen

Um die im vorangegangenen Kapitel beschriebenen Limitationen der Simulation zu adressieren, wird der Ansatz für Kante und Flansch aus Kapitel 5.2 erweitert.

5.4.1 Geometrieneuvernetzung zur Minimierung der Auswirkungen der Limitationen

Kapitel 5.3.1 beschreibt eine Abhängigkeit der Prognosequalität von der Geometrievernetzung. Dies bedeutet, dass ein Ansatz benötigt wird, der die Vernetzung optimiert. Um den Übertrag der Elektrolytsimulation auf die Geome-

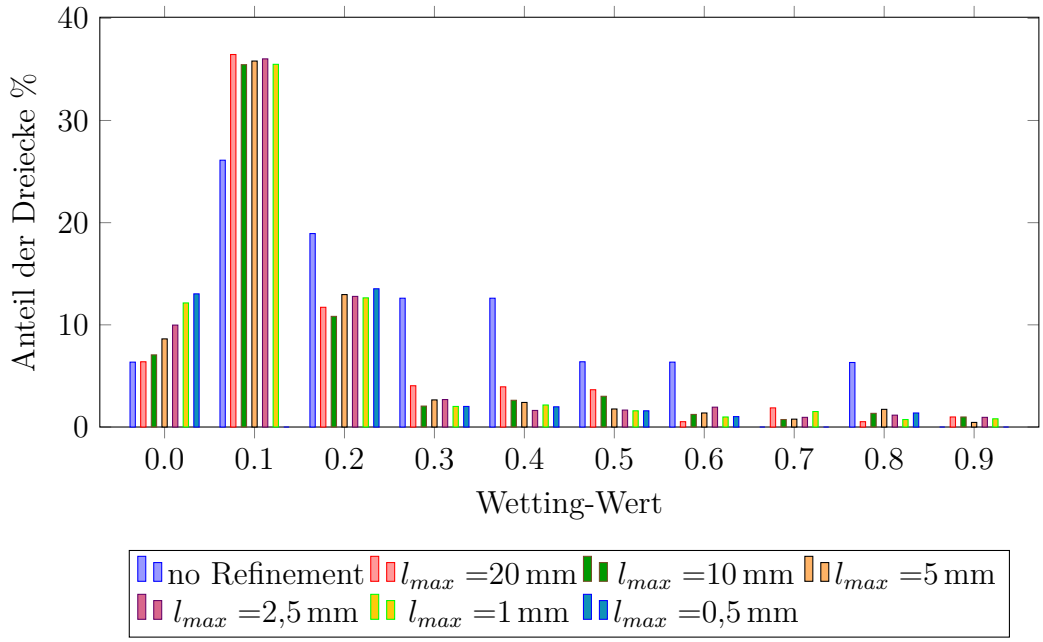


Abb. 5.5: Betrachtung der Geometrieneuvernetzung in Abhängigkeit der Maximallänge l_{max} und den Wetting-Werten

trie weiter zu ermöglichen, darf dabei die Geometrie jedoch nicht verändert werden. Hierbei wird auf die in Kapitel 4.1 vorgestellte Methodik zurückgegriffen. Um eine für die Herausforderungen optimale Geometrieneuvernetzung zu finden, wird ein passendes l_{max} gesucht.
Das Ziel ist es, einen optimalen Wert für die Größe der Geometrievernetzung zu finden. Dies bedeutet, wie in Kapitel 5.3.1 beschrieben, eine Vernetzung, die klein genug ist, um eine verbesserte Segmentierung zu erreichen, und gleichzeitig nicht zu klein, um Flanschflächen außerhalb nicht erreichbarer Bauteilverbindungen sicherzustellen. Dazu wird der oben beschriebene Ansatz auf ein Bauteil mit unterschiedlichen l_{max} angewendet. Die dabei entstehende Verteilung der Wetting-Werte wird für jeden dieser Durchläufe aufgezeichnet (Abbildung 5.5). Dieses Histogramm kann für jede Refinementstufe in einen Verlauf übersetzt werden. Bei sinkenden l_{max}, kann eine Annäherung an einen Zustand beobachtet werden, bei dem sich die Verteilung der Wetting-Werte nicht mehr verändert. Dieser Zustand entspricht einer minimalen Vernetzung, bei der die Wetting-Werte unverändert bleiben. Damit erfüllt dieser Wert die definierten Herausforderungen.

5.4.2 Erweiterung des Übertrags unter Berücksichtigung von fehlenden Datenpunkten

Punkte der Elektrolytsimulation, in denen die ToW-Werte bestimmt werden, werden auf der Oberfläche positioniert. In Ausnahmefällen existieren Kanten oder Flansche, bei denen keine Datenpunkte in den OBBs liegen. Entsprechend der Implementierung für Kanten- und Flanschelemente werden diese Werte als

NaN gewertet. In der Realität besitzen jedoch auch diese Elemente einen Wetting-Wert. Daher muss eine Methodik gefunden werden, um auch diesen Dreiecken einen Wert zuweisen zu können.
Dazu werden die Dreiecke, die keinen Wert besitzen, genauer betrachtet. Es ergibt sich ein definiertes Set an Dreiecken $t_i \in T$, für welches gilt:

$$\tau_{t_i} = \varnothing \forall t_i \tag{5.3}$$

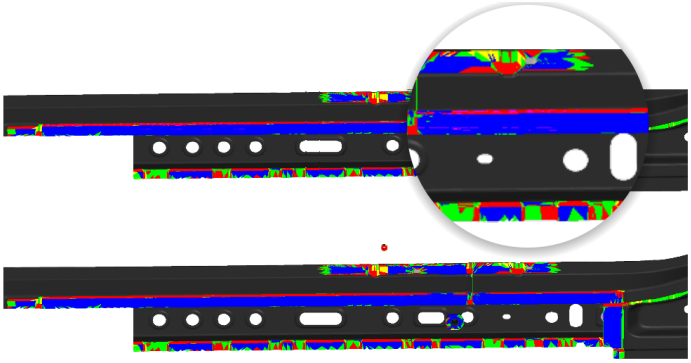
Um diesen Dreiecken t_i einen Wetting-Wert τ_i zuweisen zu können, müssen die Eigenschaften der Dreiecke T um die Informationen zu ihren Nachbarsdreiecken erweitert werden. Diese Information kann in einer Nachbarschaftsmatrix zusammengefasst werden. Diese Nachbarschaftsmatrix beschreibt die Interaktionen und Verbindungen zwischen den einzelnen Dreiecken.
Da die fehlenden Wetting-Werte durch fehlende Datenpunkte und damit nicht durch einen physikalischen, sondern lediglich durch einen Diskretisierungsfehler entstehen, kann angenommen werden, dass sich die Dreiecke t_i vergleichbar zu ihren Nachbarn verhalten. Daher wird initial angenommen, dass der Wetting-Wert des Dreiecks t_i wie folgt berechnet werden kann:

$$\tau_{t_i} = \frac{1}{n_{x_j}} \sum_{j \in \backslash (i)} \tau_j \tag{5.4}$$

Dabei beschreibt τ_j die Nachbarn des Dreiecks t_i. n_{x_i} beschreibt die Menge der Nachbarn. Diese Implementierung ist ausreichend für Vernetzungen, in denen einzelne alleinstehende Dreiecke korrigiert und angepasst werden müssen.

Ergebnisse

Ein Ergebnis dieser Implementierung ist in Abbildung 5.6 visualisiert. Es zeigt, dass mit dem vorgeschlagenen Ansatz alle benötigten Werte korrigiert werden können.



$wt > 0.5$ $0.15 < wt < 0.5$ $0.15 < wt$ Bauteiloberfläche
$wt = \text{NaN}$ $wt = 0$

Abb. 5.6: Ergebnisse der Implementierung des Ansatzes zur Korrektur von NaN-Werten. Oben das ursprüngliche, unten das korrigierte Bauteil.

5.4.3 Anpassen der Wetting-Werte aufgrund von Flanschflächen

Die folgende Ausführung beschreibt, wie der Ansatz erweitert wird, um die in Kapitel 5.3 erwähnten Beschränkungen der Simulation zu bewältigen.

Numerische Anpassung der Wetting-Werte der Flanschflächen

Es werden die Namenskonventionen aus Kapitel 4 übernommen und um V erweitert. V beschreibt die Punkte eines Meshes M. Diese Informationen werden gemeinsam mit $G(M)$ verwendet, um die Wetting-Werte in Flanschen zu korrigieren. Dazu werden in einem ersten Schritt verbundene Flächen identifiziert, welche durch zusammenhängende Dreiecke gebildet werden, die alle Wetting-Werte $\tau_i = 0$ besitzen. Diese sind in Abbildung 5.6 als zusammenhängende blauen Flächen sichtbar. Dazu wird ein Ansatz gewählt, der die oben beschriebenen Bedingungen erfüllt und die Flächen berechnet [93]. Des Weiteren wird ein Breitensuchalgorithmus angewendet [94].
Damit ergeben sich die, im Folgenden als Komponenten C_i bezeichneten, zusammenhängenden Flächen wie folgt:

$$C_i = \{v \in V \mid \exists\ (v, w),\ \forall w \in C_i\} \tag{5.5}$$

Sobald die Komponenten bestimmt sind, zielt der Ansatz darauf ab, τ_i durch angepasste Wetting-Mittelwerte zu ersetzen, die sich an den benachbarten Dreiecken orientieren. Für einen Knoten $v \in C_i$ mit einem Nullwert wird der angepasste Wert bestimmt durch:

$$\tau_j = \begin{cases} \frac{1}{|n_{t_i}|} \sum_{u \in N} \tau_i & \text{if } N(i) \neq \emptyset \\ 0 & \text{if } N(i) = \emptyset \end{cases} \tag{5.6}$$

Dabei beschreibt Gleichung 5.6 die Berechnung der Wetting-Werte in Abhängigkeit der Nachbarn N. Besitzen die Dreiecke $t_i \in C_i$ Nachbarn mit Wetting-Werten, so werden diese Wetting-Werte der Nachbarn auf die gesamten Dreiecke der Komponente C_i übertragen. Hier beschreibt n_{t_i} die Anzahl der Dreiecke mit Wetting-Werten größer als 0. Wie in der vorangegangenen Implementierung wird ein iterativer Ansatz gewählt, um alle Dreiecke abzudecken.
Aufgrund der räumlichen Begrenzung dieses Eintrags wird ein Abbruchkriterium eingeführt:

$$\tau_j = \frac{1}{|n_{d,t_i}|} \sum_{i \in n_{d,t_i}} \tau_{i,d_{max}} \quad \text{for } d(t_i) \leq d_{\max} \tag{5.7}$$

d_{max} beschreibt dabei die maximale Ebene der Breitensuche, des am Anfang von Kapitel 5.4.3 beschriebenen Algorithmus.

Ergebnisse

Die Ergebnisse dieser Implementierung sind in Abbildung 5.7 visualisiert. In der Analyse der Visualisierung fällt auf, dass ein einfacher Übertrag in vielen Bereichen gute Ergebnisse erzielt. In besonders großen Flanschen erscheint der Übertrag jedoch unrealistisch. Da die Breitensuche keine euklidischen Distanzen, sondern lediglich maximale Iterationen berücksichtigt, ist bei unterschiedlicher Vernetzung kein konsistentes Verhalten zu erwarten. Ein konsistenter geometrischer Einfluss lässt sich über den in Kapitel 2.3.4 eingeführten Kapillareffekt integrieren.

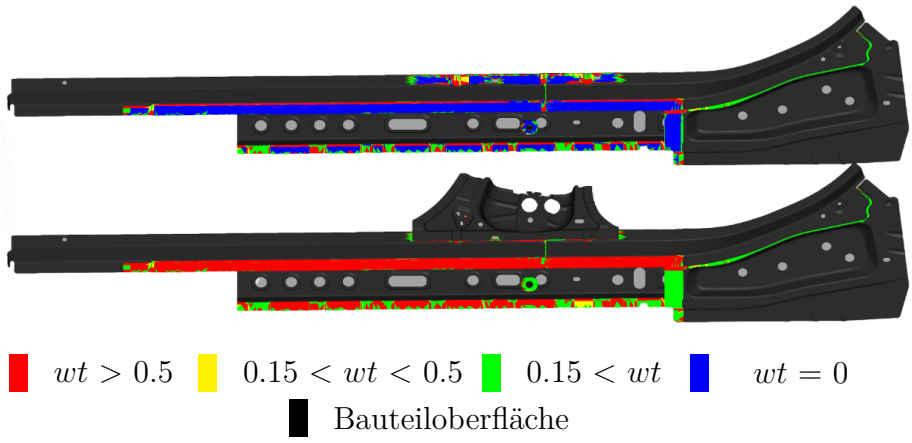


Abb. 5.7: Ergebnisse der Implementierung des Ansatzes zur Korrektur von Null-Werten. Oben das ursprüngliche Bauteil nach der Korrektur der NaN-Werte, unten das korrigierte Bauteil mit dem vorgestellten Ansatz. Um die Ergebnisse in Flanschen zu sehen, wurde das vorne liegende Bauteil ausgeblendet.

Anpassung der Wetting-Werte auf Basis physischer Einflussgrößen

Der gezeigte Ansatz, auf Basis von Nachbardreiecken Wetting-Werte zu berechnen, ist ein möglicher Ansatz. Eine weitere Möglichkeit ist es die Kapillarkraft und damit das physikalische Verhalten des Elektrolyts zu berücksichtigen, um damit eine noch bessere Abschätzung der Elektrolytbeanspruchung zu erreichen. Im Folgenden wird daher eine Erweiterung des Komponentenansatzes erklärt, bevor der darauf aufbauende Algorithmus zur Kapillarkraft-Integration erläutert wird.
Um eine genauere Auswertung der Wetting-Werte zu ermöglichen, werden die Komponenten C_i weiter in kleinere Minikomponenten unterteilt. Jede Hauptkomponente $C_i \subseteq V$ wird in Minikomponenten $\{M_{i1}, M_{i2}, \ldots, M_{ik}\}$ aufgeteilt:

$$M_{ij} \subseteq C_i \quad \text{und} \quad \bigcup_{j=1}^{k} M_{ij} = C_i \tag{5.8}$$

Jede Minikomponente M_{ij} enthält Dreiecke. Dabei wird jeder Knoten $v \in C_i$ auf der Grundlage seines Flansches einem Teil zugeordnet und die Dreiecke in Minikomponenten aufgeteilt. Die Anzahl der Minikomponenten wird wie folgt bestimmt:

$$N = \left\lceil \frac{|C_i|}{T_{\max}} \right\rceil \tag{5.9}$$

T_{max} kann dabei laufzeitoptimiert gewählt werden. In Abbildung 5.8 ist die Laufzeit in Abhängigkeit von T_{max} für eine kleinen Datensatz gezeigt. T_{max} wird mit 30 gewählt.

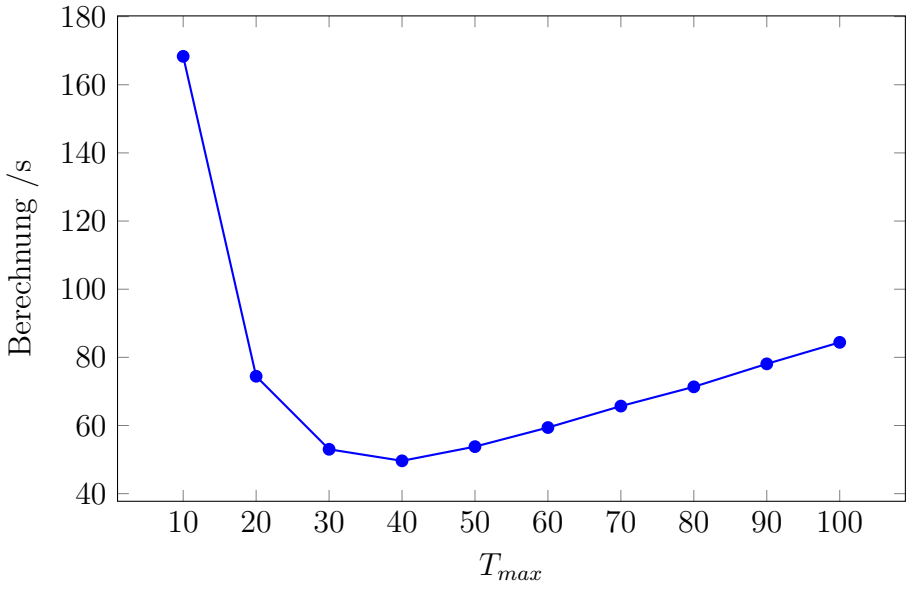


Abb. 5.8: Betrachtung von T_{max} in Abhängigkeit der Berechnungszeit

Zusätzlich wird der Kapillareffekt bei der Übertragung der Wetting-Werte in Flansche berücksichtigt (Abbildung 5.9). Nach Kapfer [7] können Flansche als Kapillaren angenommen werden.

Bei der Erweiterung der Korrektur der Wetting-Werte werden zwei Szenarien unterschieden. Einerseits Flansche, die durch den Kapillareffekt vollständig durch Elektrolyt bedeckt sind und einen Flansch bilden, der vollständig gefüllt ist. Andererseits Flansche, die nur teilweise gefüllt sind.
Bei vollständig bedeckten Bauteilen werden die Flanschflächen nach Gleichung 5.9 unterteilt. Anschließend werden die Wetting-Werte für jede Minikomponente entsprechend Gleichung 2.34 berechnet. Dazu werden im ersten Schritt die Steighöhen h berechnet. Ist die Flanschfläche in der z-Ebene kleiner als die Gesamtsteighöhe, so ist der Flansch als komplett benetzt zu betrachten. Ist die Höhe einer Flanschfläche größer als die maximale Steighöhe h, wird die Auswertung erweitert. Hierbei werden Minikomponenten in zwei Arten unterteilt. Minikomponenten innerhalb der Steighöhe und Minikomponenten außerhalb der Steighöhe. Die Minikomponenten innerhalb der Steighöhe werden mit einem Histogramm-Ansatz angepasst. Dabei wird ein Histogramm der umliegenden ToW-Werte erstellt, wobei die häufigsten Feuchtigkeitswertbereiche auf die Minikomponenten der bedeckten Höhe angewendet werden. Mathematisch kann dies wie folgt ausgedrückt werden:

$$\tau_{M_{ij}} = \arg\max_{n_l} \tau_l(n_l) \tag{5.10}$$

Dabei ist n die Anzahl der Wetting-Werte in den Histogrammkategorien. $\tau_l(n_l)$ sind die Wetting-Werte der Kategorie n, die wie in Gleichung 5.2 errechnet werden. Die Einführung eines Histogramms ermöglicht die Minimierung der Einflüsse, die benachbarte Nullwerte auf die Berechnung haben.
Für die unbedeckten Minikomponenten kommt ein abgewandelter Ansatz zur Anwendung. Der Einfluss der Feuchtigkeit nimmt mit zunehmender Entfernung vom Kapillaraufstieg ab. Dies kann wie folgt modelliert werden:

$$\tau_{M_{ij}} = \tau_i \left(1 - \frac{d}{D}\right) \tag{5.11}$$

Dabei ist d die Entfernung des Kapillaraufstiegs, τ_i der ursprünglich in Gleichung 5.7 berechnete Wetting-Wert und D die maximale Entfernung innerhalb des Bauteils.

Ergebnisse

Abbildung 5.9 gibt einen Überblick über den vorgestellten Ansatz zur Integration der Simulation über Kapillarkräfte.
In Abbildung 5.10 sind die Ergebnisse der Implementierung der Kapillarkraft dargestellt. Im Vergleich zu den vorangegangenen Implementierungen kann nun eine deutlich verbesserte Auswertung der Elektrolytbeanspruchung beobachtet werden. In einem ersten Schritt werden alle NaN-Werte eliminiert. Außerdem sind die Flächen, die durch aufeinanderliegende Flächen als Null erkannt wurden, durch die Kapillarkräfte korrigiert worden. Besonders in großen Flanschflächen wird durch diesen Ansatz eine deutlich bessere Auswertung erreicht. Dies kann im rechten Bereich des Längsträgers beobachtet werden. Der

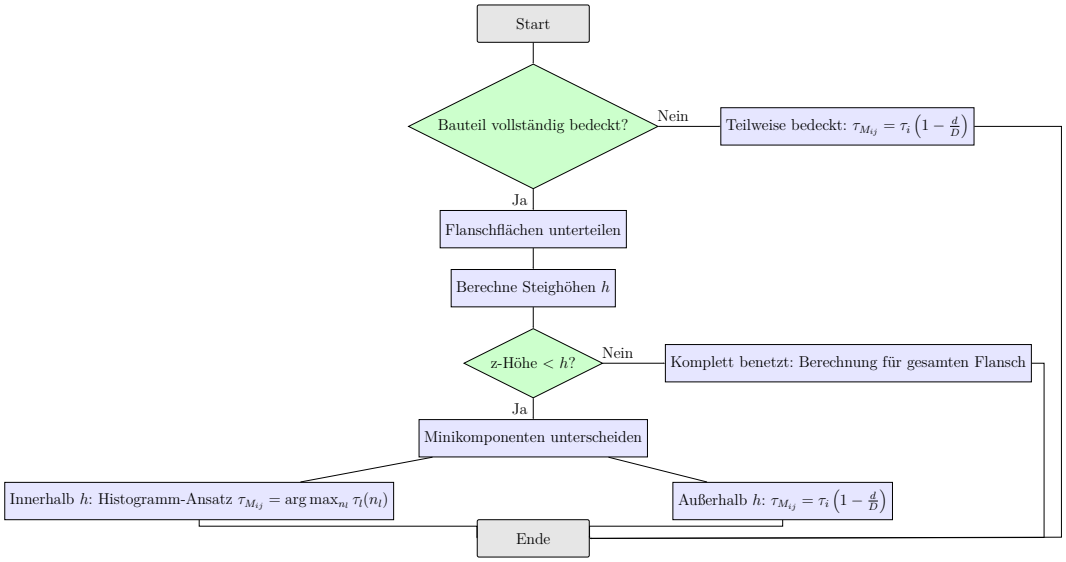


Abb. 5.9: Ablaufdiagramm zur Erweiterung der Korrektur der Wetting-Werte

vertikale Flansch auf der rechten Seite zeigt den gut funktionierenden Ansatz dieses Kapitels. Dieser Bereich ist durch einen roten Kasten markiert.

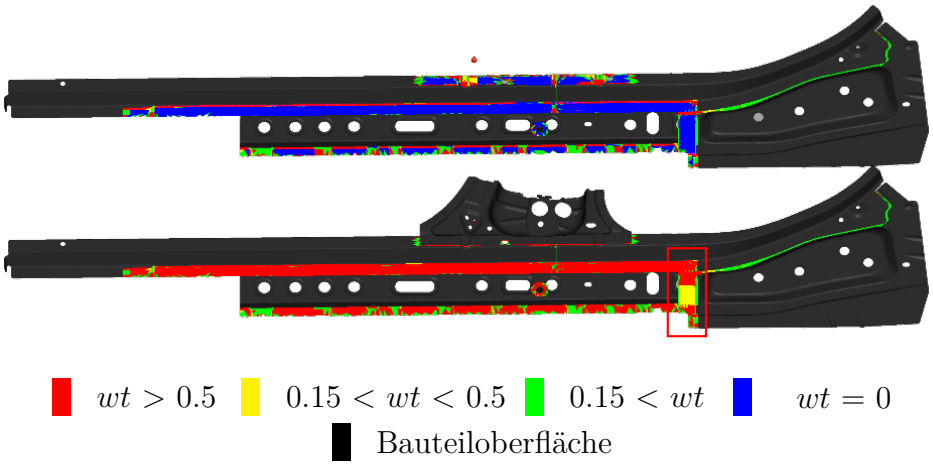


Abb. 5.10: Visualisierung der Implementierung der Minikomponenten und der Kapillarkraft. Oben ist das Bauteil nach Korrektur der NaN-Werte gezeigt, unten das Bauteil mit dem Kapillaransatz.

5.4.4 Integration mehrerer Simulationen

Kapitel 5.3.3 beschreibt den Fehler durch unterschiedlich dauernde Wege, die Partikel zurücklegen müssen, bis sie bei Sensoren ankommen, die auch unterschiedlich weit von der Partikelquelle entfernt liegen. Durch die Kombination mehrerer Simulationen und damit unterschiedlich positionierte Partikelquellen kann diese Auswirkung minimiert werden. Da in Kapitel 5.2 jeweils die Maximalwerte der umgebenden Simulationspunkte gewählt werden, kann eine

einfache Kombination der unterschiedlichen Simulationen vorgenommen werden. Damit definiert sich die neue kombinierte Punktewolke zur Auswertung $PC_{Elektrolyt,kombiniert}$ als Kombination von n Punktewolken $PC_{Elektrolyt,j}$ mit den Punkten $p_{i,Elektrolyt}$ als $PC_{Elektrolyt,kombiniert} = p_{i,Elektrolyt,j}, \tau_{i,j} mit j \in n$.

5.5 Definition eines Korrosionsrisikolabels auf Basis der ausgewerteten Simulationswerte

Mit den Wetting-Werten τ_i aus Kapitel 5.2.1 können nun Aussagen über den Wetting-Wert gemäß DIN 9227 [95] getroffen werden. Um die Korrosionsrisikobewertung aus Kapitel 4 noch zu erweitern, kann die Auswertung der Wetting-Werte auch als Risikolabel verstanden werden. Die bisher vorgestellten Korrosionsrisikolabels werden wie folgt definiert:

$$\mathcal{E}_{i,j} = \begin{pmatrix} \epsilon_{1,Kante} & \epsilon_{1,Flansch} & \cdots & \epsilon_{1,Kleber} \\ \epsilon_{2,Kante} & \epsilon_{2,Flansch} & \cdots & \epsilon_{2,Kleber} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{n_i,Kante} & \epsilon_{i,Flansch} & \cdots & \epsilon_{n_i,Kleber} \end{pmatrix} \tag{5.12}$$

Gleichung 5.12 ergibt sich aus den, in Kapitel 4 definierten, Klassifizierungen. Diese können um das in diesem Kapitel betrachtete Elektrolyt und seinen Einfluss auf das Korrosionsrisiko erweitert werden.
Wie in Kapitel 2.3.1 beschrieben, kann unter Abwesenheit von Elektrolyt keine Korrosion stattfinden. So besteht in Bereichen ohne Elektrolytbeanspruchung ($\tau_i = 0$) kein Korrosionsrisiko. Daher kann, wie mit den konstruktiven Maßnahmen, ein Korrosionslabel implementiert werden, welches die Elektrolytbeanspruchung eines Elements repräsentiert. $\epsilon_{i,Elektrolyt}$ wird null, falls $\tau_i = 0$ und $\epsilon_{i,Elektrolyt}$ wird 1, falls $\tau_i > 0$. Dabei kann Gleichung 5.12 um $\epsilon_{Elektrolyt}$ erweitert werden:

$$\mathcal{E}_{i,j} = \begin{pmatrix} \epsilon_{1,Kante} & \epsilon_{1,Flansch} & \cdots & \epsilon_{1,Kleber}\epsilon_{1,Elektrolyt} \\ \epsilon_{2,Kante} & \epsilon_{2,Flansch} & \cdots & \epsilon_{2,Kleber}\epsilon_{2,Elektrolyt} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{n_i,Kante} & \epsilon_{n_i,Flansch} & \cdots & \epsilon_{n_i,Kleber}\epsilon_{n_i,Eletrolyt} \end{pmatrix} \tag{5.13}$$

Damit beschreibt Gleichung 5.13 das Korrosionsrisiko einer komplexen Geometrie unter Berücksichtigung der Korrosionsschutzmaßnahmen und der Elektrolytbeanpruchung.

5.6 Übertragene Simulationswerte

Abbildung 5.11 zeigt ein Gesamtfahrzeug, für das mit Hilfe der, in diesem Kapitel beschriebenen Methodik Wetting-Werte berechnet werden.
Mit der Analyse der übertragenen Simulationen und unter den Berücksichtigungen der Limitationen der Simulation wird im Folgenden der Arbeit eine



Abb. 5.11: Ergebnis der auf ein Gesamtfahrzeug angewandten Methodik

$wt > 0.5$ $0.15 < wt < 0.5$ $0.15 < wt$ $wt = 0$

Bauteiloberfläche

Äquivalenz der Wetting-Werte und der Benässungszeiten nach DIN 8044 [50] angenommen, das bedeutet, die Wetting-Werte beschreiben die anteilige Zeit, in der das Dreieck mit Elektrolyt bedeckt ist.

6 Korrosionsprognose

In diesem Kapitel wird der Weg vom Korrosionsverhalten an Risikostellen zur Korrosionsprognose an Kanten und Flanschen beschrieben. Dazu werden die in dem vorherigen Kapitel beschriebenen und erzeugten Daten als Startpunkt verwendet. Das Ergebnis des Kapitels ist eine Methodik, die in der Lage ist, das Verhalten eines Gesamtfahrzeugs in einer Testumgebung vorherzusagen.
In Abbildung 6.1 ist ein Ablaufdiagramm der Korrosionsprognose dargestellt. Dabei lässt sich die Korrosionsprognose in vier Bereiche unterteilen:

1. Eingangsdaten
2. Implementierung Kantenprognose
3. Implementierung Flanschprognose
4. Anwendung

Die Eingangsdaten werden aus FEM-Simulationen bezogen und aus einem ERP-System exportiert. Darüber hinaus werden die Ergebnisse der Methodik der vorangegangenen Kapitel verwendet. Diese werden mit einer Matrix, wie in Gleichung 5.13, übergeben. Auch das Wetting wt_i wird als Ergebnis der Elektrolytsimulationsauswertung übergeben. i steht dabei für das jeweilige Oberflächenelement.
Die Implementierung der Flansch- und Kantenkorrosionsprognose bezieht sich hier auf die Integration der unterschiedlichen Einflussfaktoren, und das Training der KI-basierten Korrosionsprognose. Der Teil der Implementierung beschreibt den Fall, dass ein KI-Modell nicht vorhanden ist und trainiert werden muss. In diesem Fall werden die FEM-Daten verarbeitet.
Der letzte Abschnitt der Korrosionsprognose beschreibt die Anwendung. Sind alle KI-Modelle trainiert und bereit zur Anwendung, so können die KI-Modelle auf Daten angewandt werden.
In den folgenden Kapiteln wird die Kantenprognose sowie die Flanschprognose beschrieben.
Dabei benutzt dieses Kapitel die Ergebnisse der Arbeit von Gollé-Leidreiter et al. [Gollé-Leidreiter et al.].

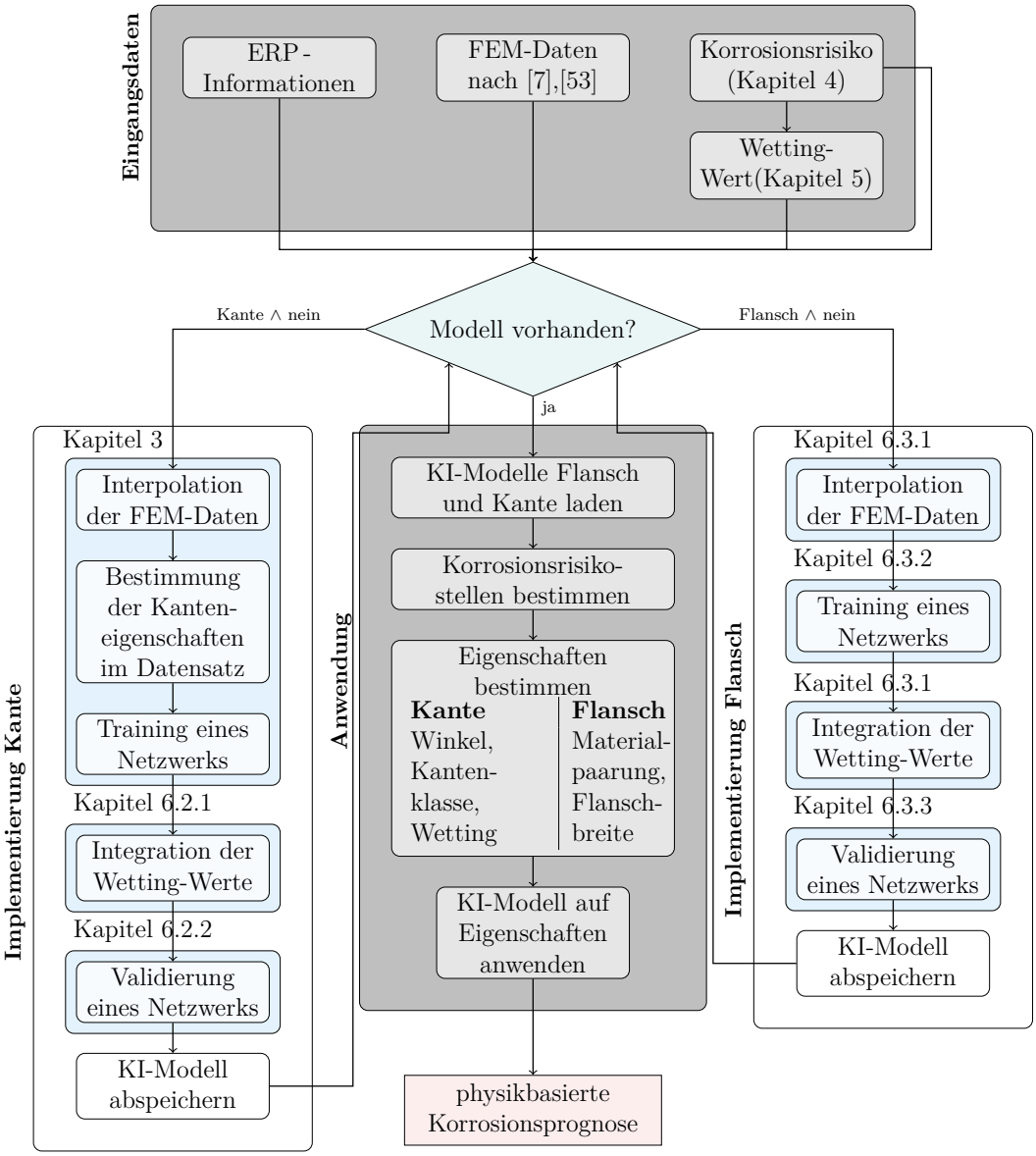


Abb. 6.1: Ablaufdiagramm der Korrosionsprognose

6.1 Analyse der Korrosionsrisikomatrix

In einem ersten Schritt werden auf Basis von Gleichung 5.13 die Dreiecke identifiziert, die ein Korrosionsrisiko tragen und damit für die Korrosionsprognose relevant sind. Bei der Bewertung des Korrosionsrisikos spielt für jedes Dreieck die Interaktion der unterschiedlichen Risikolabels eine Rolle. Diese Interaktionen werden im Nachfolgenden beschrieben.

Kantenrisiko

Das Kantenkorrosionsrisiko wird definiert durch $\epsilon_{i,K,red}$. Dabei beschreibt dieses Label, ob ein Oberflächenelement tatsächlich korrosionsgefährdet ist. $\epsilon_{i,K,red} = 1$ bedeutet eine Korrosionsgefährdung des Oberflächenelements i.

Wie in Kapitel 2.2 beschrieben, existieren Maßnahmen um Korrosionsentstehung an Kanten zu vermeiden. Klebstoff und NAD eliminieren das Korrosionsrisiko an Kanten, während die Präsenz von Wachs keinen positiven Einfluss auf das Korrosionsrisiko hat. Wachs besitzt keine schützende Wirkung gegen Kantenkorrosion. Damit ergibt sich ein neues reduziertes Korrosionsrisikolabel $\epsilon_{i,\,K,\,red}$, welches wie folgt definiert ist:

$$\epsilon_{i,\,K,\,red} = \begin{cases} 1 & \text{if } \epsilon_{i,\,K} = 1 \wedge \epsilon_{i,\,Kl}, \epsilon_{i,NAD} = 0 \wedge \epsilon_{i,\,Eletrolyt} = 1 \\ 0 & \text{if } \epsilon_{i,\,K} = 0 \vee \Big(\epsilon_{i,\,K} = 1 \wedge (\epsilon_{i,\,Kl} \vee \epsilon_{i,NAD} = 1) \vee \epsilon_{i,\,Ele} = 0\Big) \end{cases} \tag{6.1}$$

Die Dreiecke t_i, für die gilt: $\epsilon_{i,\,K,\,red} = 1$, werden in der Kantenkorrosionsprognose weiter betrachtet. $\epsilon_{i,\,Ele}$ beschreibt das Label für das Elektrolyt.

Flanschrisiko

Das Flanschkorrosionsrisiko wird definiert durch $\epsilon_{i,\,Fl,\,red}$. Dabei beschreibt dieses Label, ob ein Oberflächenelement tatsächlich korrosionsgefährdet ist. $\epsilon_{i,\,Fl,red} = 1$ bedeutet eine Korrosionsgefährdung des Oberflächenelements i. Wie in Kapitel 2.2 beschrieben, existieren Maßnahmen, um Korrosionsentstehung an Flanschflächen zu vermeiden. Als Einflüsse auf das Korrosionsrisiko von Flanschen werden dabei NAD, Wachs und Klebstoff berücksichtigt. Damit ergibt sich ein neues reduziertes Korrosionsrisikolabel $\epsilon_{i,\,Fl,\,red}$, welches wie folgt definiert ist:

$$\epsilon_{i,\,Fl,\,red} = \begin{cases} 1 & \text{if } (\epsilon_{i,\,Fl} = 1 \wedge \epsilon_{i,\,Kl}, \epsilon_{i,NAD}, \epsilon_{i,\,Wachs} = 0) \wedge \epsilon_{i,\,Eletrolyt} = \ 0 \\ 0 & \text{if } (\epsilon_{i,\,Fl} = 0) \vee \\ & \quad (\epsilon_{i,\,Fl} = 1 \wedge (\epsilon_{i,\,K} \vee \epsilon_{i,NAD} \vee \epsilon_{i,NAD} = 1)) \vee \epsilon_{i,\,Ele} = 0 \end{cases} \tag{6.2}$$

Für Dreiecke t_i, für die $\epsilon_{i,Fl,red} = 1$ werden in der Flanschkorrosionsprognose weiter betrachtet.

6.2 Kantenkorrosionsprognose

Die in diesem Kapitel vorgestellte Kantenkorrosionsprognose basiert auf der Kantenkorrosionsprognose nach Waibel et al. [13]. Waibel et al. zeigen einen Ansatz zur Prognose von Bauteilen unter Berücksichtigung der geometrischen Lage. Die Kantenkorrosionsprognose wird um den Einfluss der Elektrolytbeanspruchung erweitert. Dabei gilt diese Simulation jedoch nur unter der Einschränkung der freien Beregnung. In komplexen Geometrien wird die abschirmende Wirkung anderer Bauteile nicht berücksichtigt. Die Beschränkung auf Einzelbauteile soll mittels der verfügbaren Wetting-Werte aus Kapitel 5.2 behoben werden. Dabei werden in einem ersten Schritt Kantenbereiche nicht weiter betrachtet, die in der Elektrolytbeanspruchung keine Belastung aufweisen (Kapitel 5.5).

6.2.1 Integration der Wetting-Werte in die Korrosionsprognose

Ziel der Korrosionsprognose ist es, den Einfluss der Wetting-Werte zu implementieren.
Waibel et al. beschreiben die Kantenprognose am Beispiel einer frei beregneten Konsole und frei beregneten Blechen, in Abhängigkeit der Kantenklassen. Diese Korrosionsvorhersage beschreibt sehr gute Ergebnisse für ein frei beregnetes Bauteil. Der Zusammenhang zwischen Winkel α und Benässungszeit wt ist in Abbildung 6.2 visualisiert. Dabei beschreibt jeder Datenpunkt ein Dreieck eines Probeblechs nach Kapfer [7].

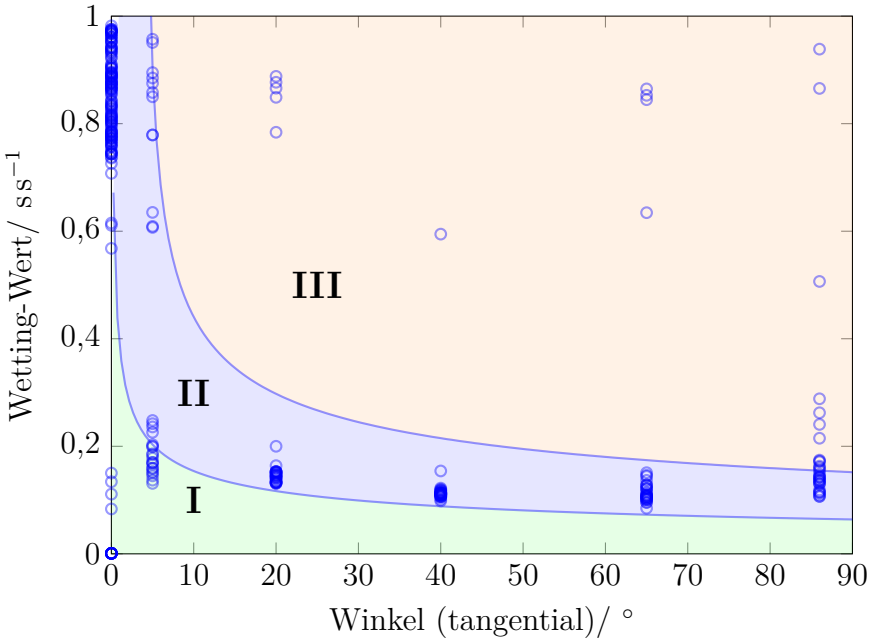


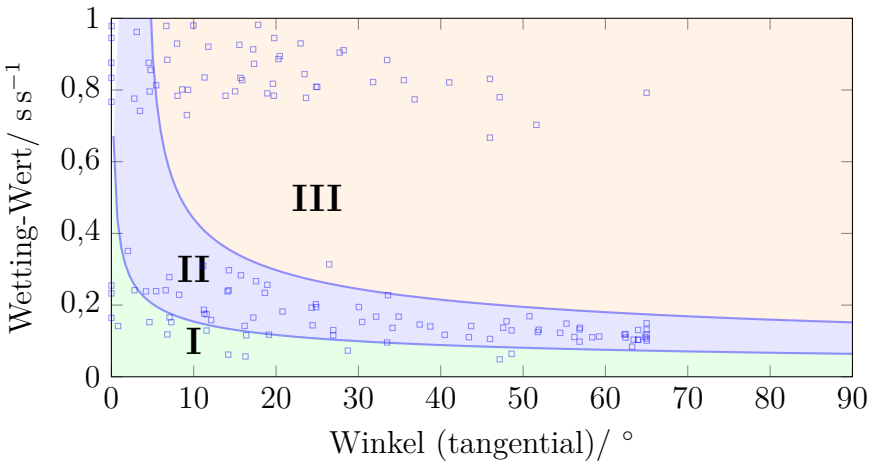
Abb. 6.2: Verteilung der Kombinationen von Wetting-Werten und Winkeln, die auf einem Probeblech gefunden werden.

Die beiden empirisch gewählten Funktionen, die den Bereich beschreiben, werden über Funktionen definiert. Diese sind so gewählt, dass ein Großteil der Punkte eingeschlossen ist. Die obere Grenze definiert sich dabei in Gleichung 6.3 (Grenze **II-III**), die untere Grenze wird in Gleichung 6.4 definiert (Grenze **I-II**). Punkte, die im Bereich **II** liegen, werden als frei beregnet angenommen und sind dadurch als sehr gut, durch bestehende Ansätze, beschreibbar anzusehen.

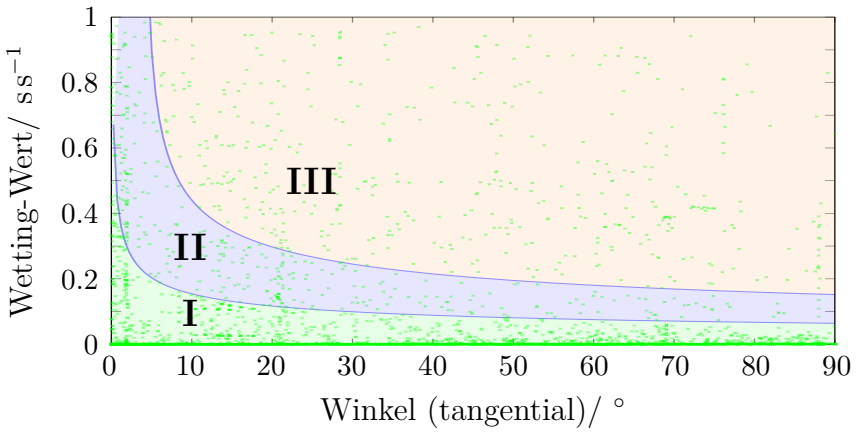
$$wt < 1.4 \cdot (3 \cdot (\alpha - 4))^{(-1/2.5)} \tag{6.3}$$

$$wt > 0.6 \cdot (3 \cdot (\alpha + 0))^{(-1/2.5)} \tag{6.4}$$

Dies kann aus der Probegeometrie abgeleitet werden, die Waibel untersucht. In Abbildung 6.3a wird die Verteilung eines frei beregneten Bauteils gezeigt. Dies bestätigt die empirisch gewählten Bereiche.



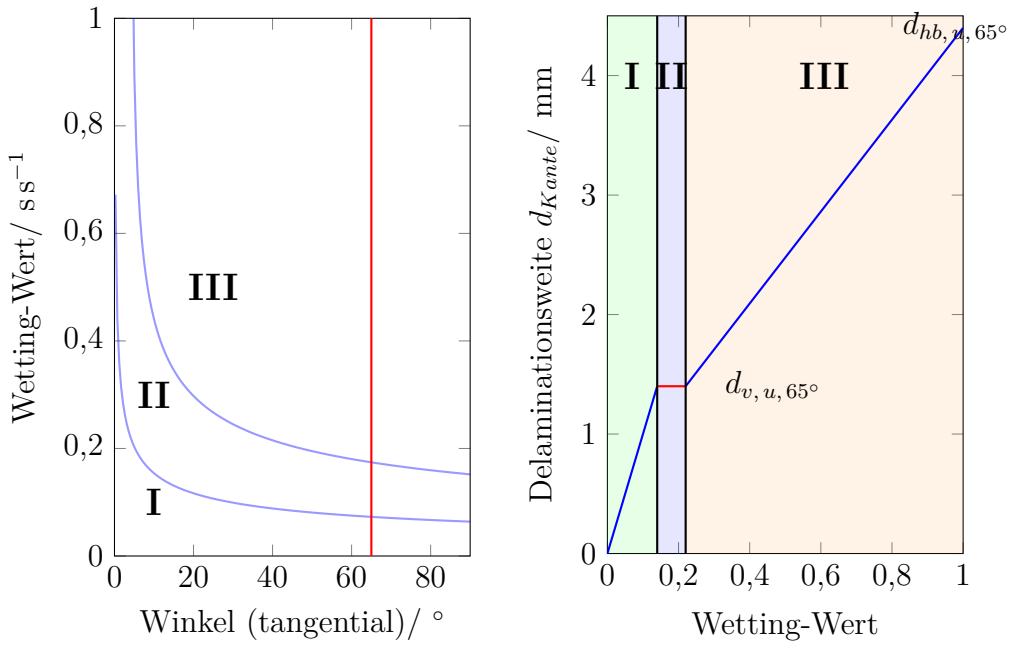
(a) Verteilung der Kombinationen von Wetting-Werten und Winkeln, die auf dem von Waibel vorgestellten Bauteil gefunden werden.



(b) Die Verteilung der Wetting-Werte einer Tür in Abhängigkeit vom tangentialen Winkel der Kantenelemente einer Tür.

Abb. 6.3: Vergleich der Verteilungen der Wetting-Werte in Abhängigkeit vom Winkel für zwei verschiedene Bauteile.

In komplexeren Geometrien, wie beispielsweise einer Autotür, existieren jedoch nicht nur frei beregnete Bereiche, sondern auch Bereiche, die indirekt benetzt werden (Abbildung 6.3b).
Deshalb muss für diese Bereiche die Korrosionsprognose erweitert werden. Um die Ergebnisse der Kantenprognose der Bereiche, welche direkt beregnet sind, zu integrieren, wird ein Interpolationsansatz genutzt, der die Kantenprognose erweitert. Dieser Interpolationsansatz ist in Abbildung 6.4a dargestellt. Dabei ergeben sich für den Winkel $\alpha = 65°$ jeweils ein Schnittpunkt pro Funktion. Diese Schnittpunkte haben die Werte $y_1 = 0.14$ und $y_2 = 0.22$ (Abbildung 6.4b). Damit kann das Kantenelement bei einem Wetting-Wert zwischen y_1 und y_2 sehr gut durch die bestehenden Prognosewerkzeuge vorhergesagt werden.
Dabei wird über die unterschiedlichen Bereiche interpoliert. Die Annahme,



(a) Interpolationsansatz anhand einer beispielhaften Kante mit $\alpha = 65°$.

(b) Beispielhaftes Verhalten eines Kantenelements mit tangentialem Winkel $\alpha = 65°$ in Abhängigkeit von wt und d_{max} bei Woche 16.

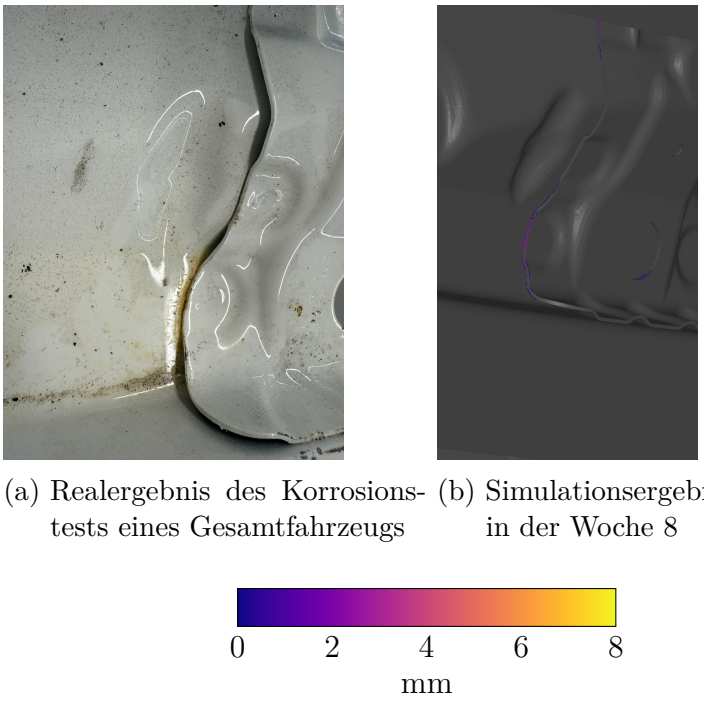
Abb. 6.4: Darstellung des Interpolationsansatzes und beispielhaftes Verhalten eines Kantenelements

dass Bereich **II** bekannt ist, gibt durch die Prognose nach Waibel Informationen darüber, wie sich die Bereiche außerhalb verhalten müssen. Je größer der Wetting-Wert, desto höher muss auch die Delaminationsweite sein, da die galvanische Korrosion länger abläuft. Da keine weiteren Informationen darüber existieren, wie sich die Delaminationsweite in Bereich **I** und **III** verhält, wird ein lineares Verhalten angenommen. Um diese Interpolation durchzuführen, werden die Stützpunkte der Interpolation gesucht.
Für den Bereich **I** lassen sich die Stützpunkte wie folgt definieren. Die Werte in den Grenzbereichen werden aus den Schnittpunkten x_1 und x_2 entnommen. Diese Bereiche sind die Schnittpunkte aus Abbildung 6.4a. Die Delaminationswerte ergeben sich aus der Kantenkorrosionsprognose nach Waibel et al. [13]. Das Verhalten einer Kante mit 65° der Kantenklasse **v,u** ist dadurch als $d_{v,u,65°}$ beschrieben. Es wird darüber hinaus angenommen, dass bei $wt = 0$ keine Korrosion entsteht. Damit sind die Interpolationspunkte des Bereichs **I** definiert.
Für den Bereich **III** sind die Stützpunkte $d_{v,u,65°}$ und $d_{hb,u,65°}$. Der Interpolationspunkt $d_{hb,u,65°}$ ergibt sich durch die maximal mögliche Delaminationsweite. Die Kantenklasse **hb,u** beschreibt Kantenelemente, bei denen Elektrolyt stehen bleibt. Hohe Wetting-Bereiche bedeuten eine starke Benässung. Daher wird die Annahme getroffen, dass die Kante in hohen Wetting-Bereichen eher

dem Verhalten einer Kante der Kantenklasse **hb,u** entspricht. Damit ergibt sich der Verlauf der Delamination zwischen $wt = x_2$ und $wt = 1$.

6.2.2 Evaluierte Kantenkorrosionsstellen am Gesamtfahrzeug

Die vorgestellte Kantenprognose wird an einem Gesamtfahrzeug evaluiert. Dabei werden die Prognoseergebnisse mit dem Testergebnis eines Gesamtfahrzeugs verglichen. Dazu werden 38 Stellen validiert. Da lediglich ein Fahrzeug zur Verfügung steht, wird die letzte Woche der Testzyklen validiert. Ein Beispiel dieses Vergleichs wird in Abbildung 6.5 visualisiert.



(a) Realergebnis des Korrosionstests eines Gesamtfahrzeugs

(b) Simulationsergebnis der Stelle in der Woche 8

Abb. 6.5: Vergleich der Realtestergebnisse mit den Simulationsergebnissen in der Woche 8

Die Proben wurden zerstörungsfrei vermessen. Dies ist bei der Auswertung der Delaminationsweiten zu berücksichtigen, da sich dieses Verfahren von der Vorgehensweise bei der Kantenauswertung der Probebauteile unterscheidet. Daher ist zu erwarten, dass die am Gesamtfahrzeug gemessenen Delaminationsweiten deutlich geringer ausfallen als die Prognosewerte. Zusätzlich werden die Proben vor der Delaminationsanalyse gedampfstrahlt und gebeizt, wodurch die Kanten weiter delaminiert werden. Darüber hinaus sind die Bereiche mit Lack beschichtet, was das Delaminationsverhalten gemäß Kapitel 2.4 zusätzlich beeinflusst.

6.2.3 Ergebnisse der Kantenkorrosionsprognose

Um die Kantenprognose zu evaluieren, werden unterschiedliche Kantenbereiche an einem Gesamtfahrzeug betrachtet. Diese Kantenbereiche finden sich in Abbildung 6.6.

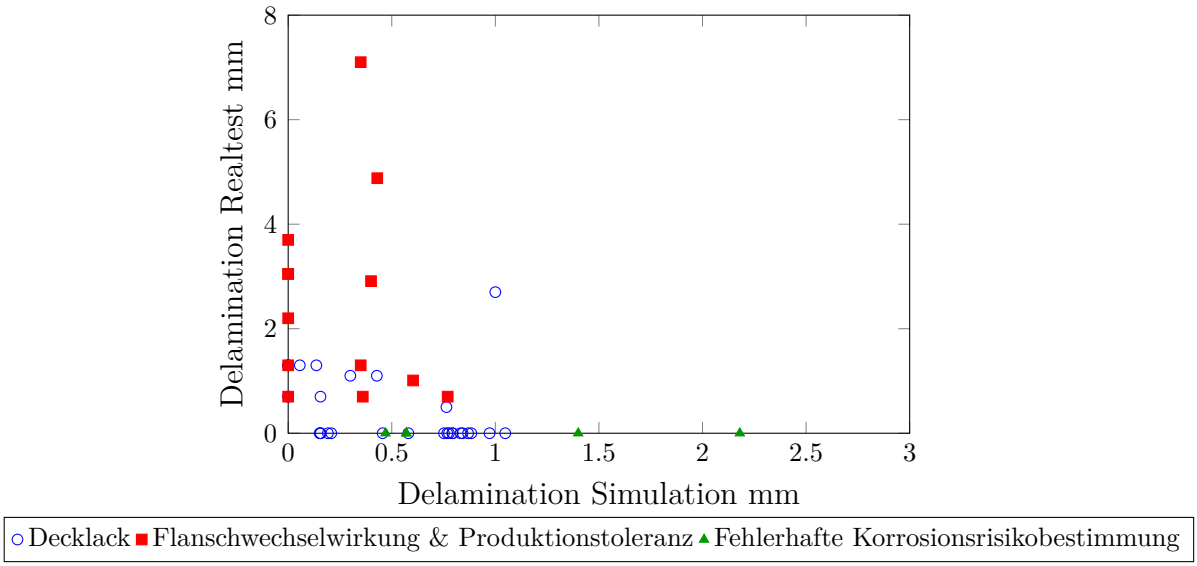


Abb. 6.6: Alle am Gesamtfahrzeug ausgewerteten Kantenpunkte mit simulierter und realer Delamination

Dabei beschreibt die Korrosionsprognose das korrosive Verhalten in einigen Fällen sehr gut. Allerdings existieren unterschiedliche Einflüsse, die eine durchweg gute Prognose verhindern. Diese Stellen können entsprechend ihrer dort wirkenden Einflüsse gruppiert werden.

1. Decklack: Das Korrosionsverhalten wird durch Decklack beeinflusst. Am Gesamtfahrzeug findet sich an den meisten untersuchten Stellen dieser Decklack.

2. Wechselwirkung Korrosionsstellen: Ist die Schutzwirkung durch KTL oder Zink geschwächt, verstärkt dies den Korrosionsprozess, unabhängig davon, ob der ursprüngliche Korrosionsprozess durch einen Kanten- oder einen Flanschkorrosionsmechanismus entstanden ist.

3. Fehlerhafte Korrosionsrisikobestimmung: Durch fehlerhaft dokumentierte oder falsch erkannte Korrosionsrisikomaßen, kann es zu einer fehlerhaften Prognose kommen.

6.3 Flanschkorrosionprognose

Die Flanschprognose ist auf Basis von FEM-Ergebnissen von Schneider [53] implementiert. Integriert sind darüber hinaus die ausgewertete Elektrolytsimulation aus Kapitel 5. Diese Prognose wird auf die in Kapitel 4 identifizierten Oberflächenelemente angewendet. Im Folgenden wird die Implementierung beschrieben.

6.3.1 Aufbau der KI-basierten Flanschkorrosionsprognose

Die Flanschprognose berücksichtigt die aus Kapitel 2.2 abgeleiteten Einflüsse. Als Einflussfaktoren werden auf Basis der Arbeiten von Kapfer [7] und Schneider [53] folgende Parameter definiert:

1. Klimaeinfluss
2. Flanschbreite
3. Materialpaarung

Damit auf Basis dieser Parameter eine Prognose getroffen werden kann, müssen einige Annahmen getroffen werden. Wie in Kapitel 2.2 beschrieben, werden die der Prognose zugrunde liegenden Daten in einem reproduzierbaren und vergleichbaren Aufbau erstellt. Um diese Daten auf eine Realkonstruktion anwenden zu können, wird gesetzt, dass diese Bedingungen auch am Gesamtfahrzeug zu finden sind. Dabei werden mehrere Annahmen getroffen.

1. Beschichtung: Die Flansche der Prüfkörper werden bei der Beschichtung mit KTL maskiert, damit die Flanschfläche frei von jeder Beschichtung ist.
2. Flanschverbindung: Der Flansch wird durch eine Schraubverbindung hergestellt. Es wird angenommen, dass die nach Wünsche [8] beschriebenen Flanschprofile entstehen.
3. Spalthöhe: Es wird von einer konstanten Spalthöhe im Flansch ausgegangen. Diese kann nicht aus dem Datensatz bestimmt werden. Weiterhin wird in der Datengrundlage nicht zwischen unterschiedlichen Spalthöhen unterschieden.
4. Flanschgeometrie: Es wird angenommen, dass Flansche immer in rechteckiger Form vorliegen, die zu der Form des Prüfkörpers passen.

Die Daten, die zum Training genutzt werden, sind wie in Abbildung 6.7 dargestellt. Diese Grafik stellt am Beispiel eines Stahl-Stahl-Flansches (CR300) die Datengrundlage einer Materialkombination dar. Gestrichelte Funktionen beschreiben eine Datenreihe, die durch das Interpolieren einer Simulation entstanden ist. Durchgezogene Funktionen beschreiben Simulationsergebnisse.

Die vier Verläufe werden zur Verbesserung der Datenbasis zwischen 10 mm und 45 mm Flanschbreite kubisch interpoliert. Die interpolierten Daten werden als Zeitreihe zum Training eines Netzwerks genutzt.

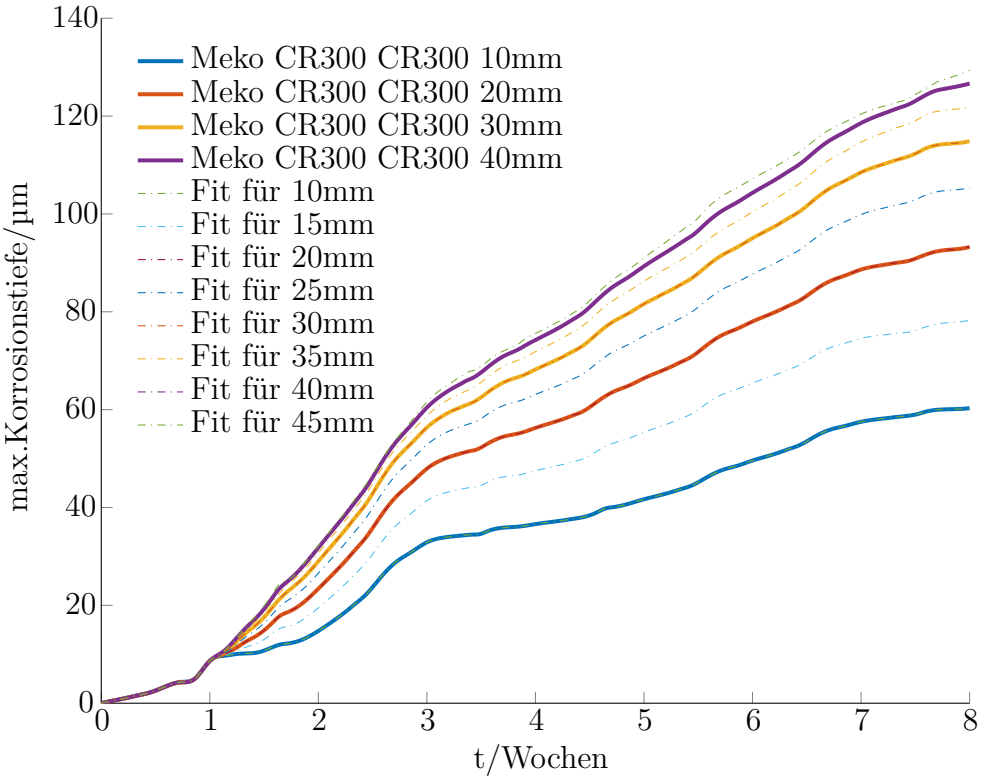


Abb. 6.7: Daten, die zum Training des Netzwerks benutzt werden, am Beispiel der FEM-Simulation von Stahl mit Stahl

Die vollständigen Daten finden sich in Anhang A. Für die Prognose werden als Inputdaten zwei bis vier FEM-Simulationsverläufe übernommen. Jede Simulation kann mit zwei unterschiedlichen Materialien beschrieben werden. Dabei beschreibt die erste Bezeichnung das Material, dessen Korrosionstiefe betrachtet wird. Die zweite Bezeichnung beschreibt den Flanschgegenpart und damit das zweite Material der Materialpaarung. Die Simulationszeitreihe *Meko AL CR300 10mm* beschreibt also eine Simulation, bei der das Substrat, welches simuliert wird, aus dem Material CR300 gemacht ist, das Material des Spaltbildners ist aus Aluminium, und die Flanschbreite ist 10 mm groß.
Als Architektur für die Prognose wird ein NARX gewählt [97]. Als Parameter für die Inputdaten werden die in dem Kapitel Kapitel 6.2.3 gezeigten Einflussfaktoren benutzt.

6.3.2 Ergebnisse des Trainings der Flanschkorrosionsprognose

Das Training des NARX-Netzwerks auf der Grundlage der übergebenen FEM-Daten ergibt eine Genauigkeit von 6,9 µm. Dieses trainierte Netzwerk wird auf den digitalen Datensatz und auf die identifizierten Flanschstellen übertragen. Dazu werden aus den Produktdaten die Materialien ausgelesen und auf die aus der Simulation bekannten Materialien Aluminium, Stahl, PMMA und Luft verallgemeinert. Diese Information geht in Form eines Integers in die Prognose ein. Darüber hinaus wird die Fläche der zusammenhängenden Dreiecke, die den Flansch bilden, bestimmt und durch die Flanschtiefe geteilt. Damit ergibt sich die Flanschbreite des zu prognostizierenden Flanschs, die als Parameter in die Prognose einfließt. Dieser Parameter kann sich zwischen 5 mm und 45 mm bewegen.

6.3.3 Validierung der Prognose anhand eines Gesamtfahrzeugs

Die Validierung der Qualität der Flanschprognose erfolgt anhand von mehreren Flanschstellen an einem Gesamtfahrzeug. Dabei wird das Gesamtfahrzeug in zwei unterschiedlichen Realtestumgebungen getestet. Die Testumgebungen sind äquivalent zu den Testumgebungen, die genutzt werden, um die FEM-Simulation zu validieren. Das prognostizierte Gesamtfahrzeug ist in Abbildung 6.8 gezeigt.



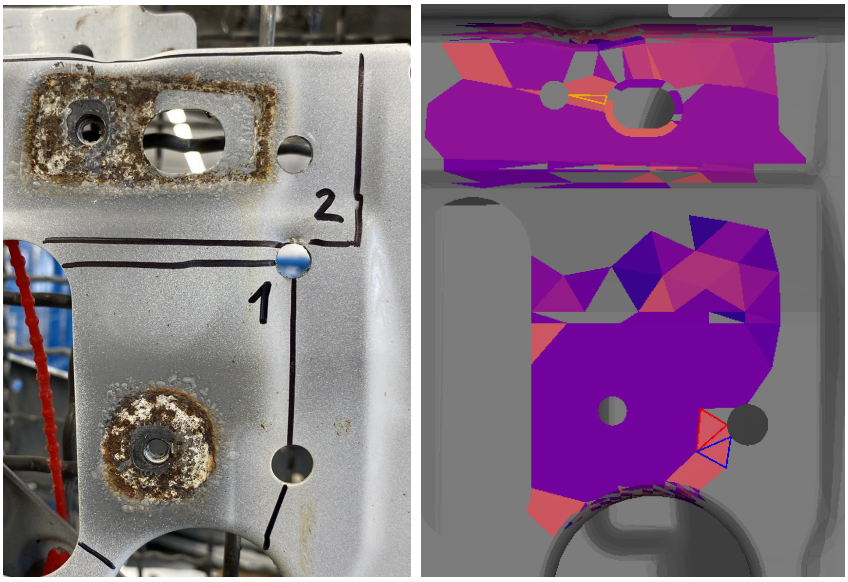
Abb. 6.8: Prognostiziertes Gesamtfahrzeug

Ausgewählte Flanschstellen

Beispielhaft werden einige der ausgewerteten Flanschstellen gezeigt. Die Bilder sind aus dem Realtest 2 entstanden. Die realen Flanschstellen wurden auf die maximalen Korrosionstiefen untersucht.

Flanschstelle 1 und 2 Für Stelle 1 und 2 (Abbildung 6.9)besteht ein Korrosionsrisiko. Es werden in der Entwicklung keine konstruktiven Maßnahmen definiert. Die Elektrolytsimulation dieses Bereichs zeigt eine Benässung, es wird

also eine Prognose durchgeführt (Abbildung 6.9b). Dies deckt sich mit den Ergebnissen der Testumgebung (Abbildung 6.9a). Die Ergebnisse von Stelle 1 und 2 sind in Tabelle 6.1 aufgeführt.



(a) Realtestverhalten der Flanschstellen 1 und 2.

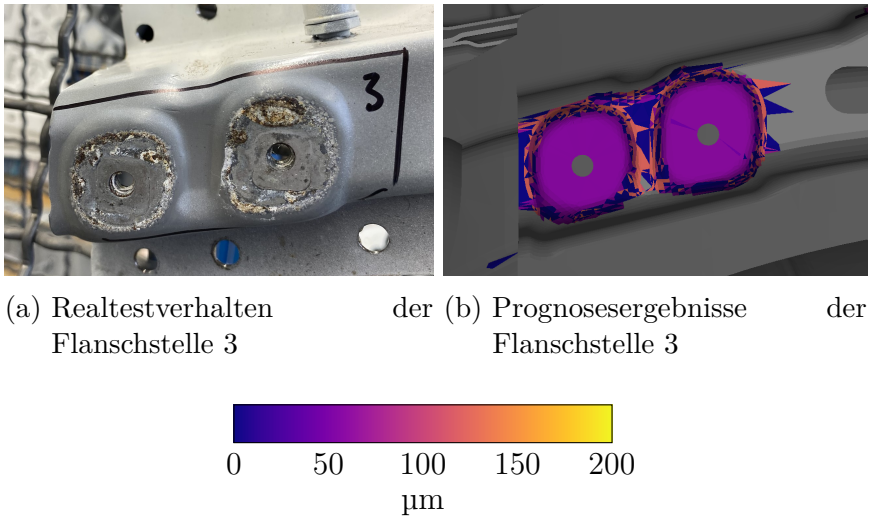
(b) Prognosesergebnisse der Flanschstellen 1 und 2.

Abb. 6.9: Flanschstelle 1 und 2 im Realtestverhalten und in der Prognose

Flanschstelle	Ergebnis Realtest 1	Ergebnis Realtest 2	Ergebnis Prognose
1	154,9 µm	125,0 µm	109 µm
2	110,9 µm	109,3 µm	108 µm

Tabelle 6.1: Flanschstelle 1 und 2, Vergleich von Realtestergebnis und Prognoseergebnis

Flanschstelle 3 Wie auch bei Flanschstelle 1 und 2 wird Flanschstelle 3 (Abbildung 6.10a) auf Basis der Korrosionsrisikolabels als korrosionskritisch erkannt. Sie wird durch Elektrolyt benetzt, und keine der konstruktiven Maßnahmen schützt den Bereich. Daher wird eine Korrosionsprognose durchgeführt (Abbildung 6.10b). Die Ergebnisse von Flanschstelle 3 sind in Tabelle 6.2 aufgeführt.



(a) Realtestverhalten der Flanschstelle 3

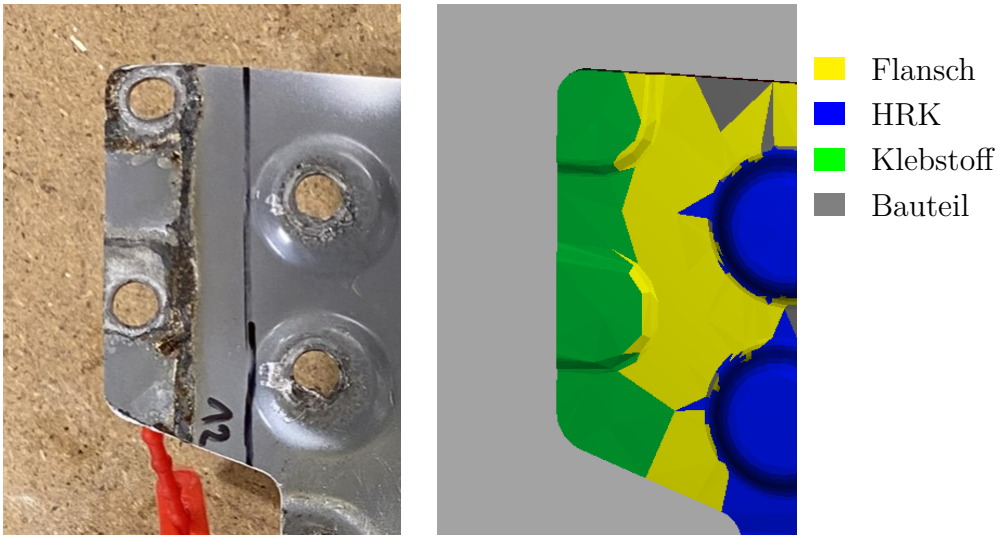
(b) Prognosesergebnisse der Flanschstelle 3

Abb. 6.10: Flanschstelle 3 im Realtestverhalten und in der Prognose

Ergebnis Realtest 1	Ergebnis Realtest 2	Ergebnis Prognose
41,0 µm	50,8 µm	51 µm

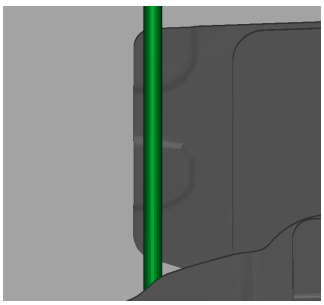
Tabelle 6.2: Flanschstelle 3, Vergleich von Realtestergebnis und Prognoseergebnis

Flanschstelle 12 Flanschstelle 12 beschreibt einen Bereich, in dem beim digitalen Datensatz Klebstoff vorhanden ist (Abbildung 6.11c), in der Realität jedoch nicht. Daher entsteht ein Korrosionsrisiko, obwohl dieser Bereich als nicht korrosionsgefährdet eingestuft wird. Es wird daher keine Prognose durchgeführt, da die Klassifizierung einen klebstoffgeschützten Bereich erwartet (Abbildung 6.11b).



(a) Realtestverhalten der Flanschstelle 12

(b) Klassifizierungsergebnisse der Flanschstelle 12 mit Legende



(c) Klebstoffbauteil (grün) im digitalen Datensatz an Flanschstelle 12

Abb. 6.11: Flanschstelle 12, im Realtestverhalten, mit Klassifizierungsergebnisse und mit Klebstoffdatensatz

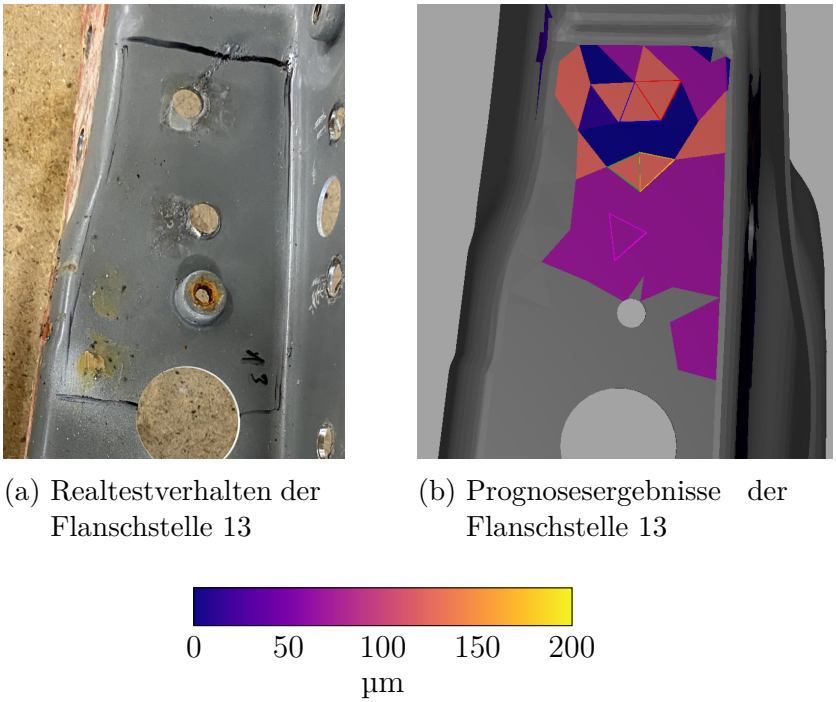
Das Ergebnis ist in Tabelle 6.3 dargestellt. Für das Prognosesergebnis wird 0 µm eingetragen, da keine Prognose durchgeführt wird.

Ergebnis Realtest 1	Ergebnis Realtest 2	Ergebnis Prognose
38,5 µm	30,4 µm	0 µm

Tabelle 6.3: Flanschstelle 12, Vergleich von Realtestergebnissen und Prognoseergebnis

Flanschstelle 13 Flanschstelle 13 (Abbildung 6.12) wird auf Basis der Korrosionsrisikolabels als korrosionskritisch erkannt. Sie wird durch Elektrolyt benetzt und keine der konstruktiven Maßnahmen schützt den Bereich. Daher wird eine Korrosionsprognose durchgeführt (Abbildung 6.10b). Die Ergebnisse von

Flanschstelle 13 sind in Tabelle 6.4 aufgeführt. Realtest 1 der Flanschstelle 13 hat eine Korrosionstiefe von 0 µm, Realtest 2 eine Korrosionstiefe von 85,6 µm. Die Prognosesergebnisse liegen mit einer Differenz von 0,7 µm sehr nah an den Ergebnissen des Realtests 2 (Tabelle 6.4). In diesem Bereich sieht man sehr große Unterschiede in den Korrosionstiefen an der gleichen Stelle. Ein Grund dafür kann die Produktionstoleranz und damit auch die KTL-Beschichtung innerhalb eines Flansches sein. Diese Unterschiede der gleichen Geometrie sind in Abbildung 6.13 visualisiert.



(a) Realtestverhalten der Flanschstelle 13

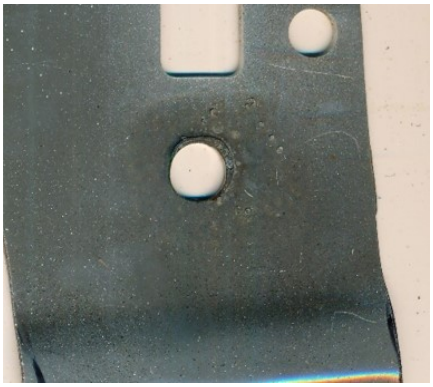
(b) Prognosesergebnisse der Flanschstelle 13

Abb. 6.12: Flanschstelle 13, Vergleich von Realtestergebnissen mit Prognoseergebnissen

Ergebnis Realtest 1	Ergebnis Realtest 2	Ergebnis Prognose
0 µm	85,3 µm	86 µm

Tabelle 6.4: Flanschstelle 13, Vergleich von Realtestergebnisse mit Prognoseergebnis

Abbildung 6.13 vergleicht die gleiche Geometrie in der KTL-Benetzung. Es ist ein großer Unterschied zwischen Realtest 1 (Abbildung 6.13a) und Realtest 2 (Abbildung 6.13b) zu erkennen. Dies kann die starken Unterschiede in den Ergebnissen der Korrosionstiefen erklären.



(a) Ergebnisse im Realtest 1. Eine klare KTL-Schicht innerhalb des Flansches ist sichtbar. Keine sichtbaren Korrosionserscheinungen.



(b) Ergebnis im Realtest 2. Es ist keine klare KTL-Schicht mehr sichtbar. Es sind deutliche Korrosionserscheinungen zu erkennen.

Abb. 6.13: Vergleich von Realtestergebnissen einer Flanschgeoemtrie hinsichtlich KTL-Beschichtung und Korrosionserscheinungen

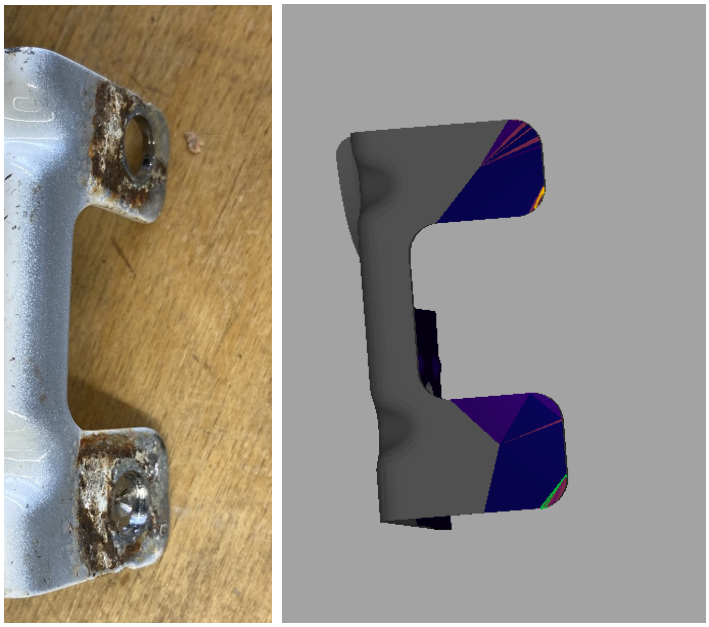
Flanschstelle 5 Flanschstelle 5 beschreibt ein Bauteil, welches einen Flansch mit senkrechter Orientierung beinhaltet (Abbildung 6.14a). In diesem Flansch existieren keine korrosionsvermeidenden Maßnahmen. Es wird die Elektrolytbeanspruchung aus der Simulation ausgewertet und eine Korrosionsprognose durchgeführt (Abbildung 6.14b). Durch die senkrechte Orientierung des Bauteils existieren jedoch lediglich vereinzelte benässte Dreiecke. Entsprechend der Methodik aus Kapitel 5.2 werden keine Werte in die anliegenden Bereiche übertragen. Damit wird die Elektrolytbeanspruchung des Flanschbereichs im Vergleich zu den Realergebnissen aus Realtest 1 deutlich unterbewertet (Tabelle 6.5).

Ergebnis Realtest 1	Ergebnis Realtest 2	Ergebnis Prognose
123,5 µm	0 µm	90 µm

Tabelle 6.5: Flanschstelle 5, Vergleich von Realtestergebnis mit Prognoseergebnis

6.3.4 Diskussion der Flanschprognoseergebnisse

Der Ansatz der Flanschprognose liefert sehr gute Ergebnisse. Dies kann beispielsweise an den Flanschstellen 1 und 2 beobachtet werden (Abbildung 6.9). Die Prognose und Realität liegen hier an beiden Flanschstellen mit 9 µm und 16 µm sehr nah beieinander, wobei diese zwei Werte die Differenz zwischen Prognose und Realität beschreiben. Im Besonderen müssen diese Ergebnisse auch



(a) Realverhalten der Flanschstelle 5

(b) Prognosesergebnisse der Flanschstelle 5

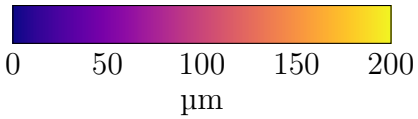


Abb. 6.14: Flanschstelle 5, Vergleich von Realtestergebnissen mit Prognoseergebnissen

in den Kontext der Standardabweichungen der Realergebnisse gesetzt werden, die den FEM-Korrosionssimulationen zugrunde liegen. Unabhängig von der Varianz können die Abweichungen zwischen Korrosionsprognose und Realtestergebnis auf bestimmte Einflussfaktoren zurückgeführt werden. Dies können Differenzen zwischen digitalem Datensatz und den Realbedingungen sein, der Abweichung von Annahmen, die im Rahmen dieser Arbeit getroffen werden oder Ungenauigkeiten in den Simulationen.
Eine Flanschstelle mit einer starken Abweichung zwischen Prognose und Realtestergebnis wird in Abbildung 6.11 gezeigt. Hierbei wird im Klebstoffdatensatz eine Klebstoffnaht dokumentiert und durch die Klassifikationsalgorithmen erkannt. Infolgedessen wird auch keine Prognose durchgeführt. Solche Varianzen können im Rahmen einer Korrosionsprognose jedoch nicht betrachtet werden. Sie führen zu einer Abweichung der Gesamtprognose.
Eine weitere Quelle für Abweichungen zwischen Realtestergebnis und Prognose beschreiben die, in Kapitel 6.3.1 beschriebenen, Annahmen.
Im Besonderen unterliegt die Annahme der beschichtungsfreien Flansche den Produktionstoleranzen. Betrachtet man die zwei Flansche aus dem Realtest

1 und 2 (Abbildung 6.13a, Abbildung 6.13b), fallen starke Unterschiede in der vorhandenen KTL-Schicht auf. Der Flansch ist in einem Test komplett beschichtet, im anderen Flansch ist keine KTL sichtbar. Sobald die Annahmen der zugrunde liegenden Daten wieder eintrifft, nämlich unbeschichtete Flanschbereiche, stimmt die Prognose mit den Realergebnissen überein. Dies zeigt, dass die Prognose in den Bereichen, in denen die Annahmen zutreffen, sehr gute Ergebnisse liefert. Jedoch können Produktionsumgebungen durch Toleranzen immer zu Unschärfen der Ergebnisse führen.
Eine weitere Quelle für Ungenauigkeiten ist der Übertrag der Elektrolytsimulationsergebnisse auf die Flanschflächen. Eine Herausforderung für die Korrosionsprognose sind Flansche, die senkrecht stehen und mit einer Kante enden. Bei dieser Geometrie gibt es keine Möglichkeit, Benässungen in den Flansch zu übertragen. Die Kante wird nicht als Flansch erkannt und damit auch keine angrenzenden Flansche, deren Benässung in den Flansch eingetragen werden kann. Diese Bereiche werden durch den fehlenden Wetting-Wert deutlich unterschätzt (Abbildung 6.14).

Flanschstelle	Realtest 2 \µm	Prognose \µm
1	125	109
2	109	108
3	51	51
13	85	109
14	56	66
15	63	92
16	125	116
18	75	52
19	45	35

Tabelle 6.6: Ergebnisse der ausgewerteten Flanschstellen. Verglichen werden Realtest 2 mit den Prognoseergebnissen. Es werden die Flanschstellen berücksichtigt, bei denen die Annahmen zutreffen.

In Bereichen, in denen die Annahmen zutreffen (Tabelle 6.6), liefert die Prognose sehr gute Ergebnisse.
Ein RMSE von ca. 15,5 µm ist ein sehr guter Wert (Tabelle 6.7).

Metrik	Wert \µm
MAX	23.7
MIN	0.2
MSE	242.24
RMSE	15.564

Tabelle 6.7: Auswertung der Ergebnisse aus Tabelle 6.6

Ebenfalls eine relevante Aussage über die Qualität der Prognose sind die maximalen Abweichungen der Prognose von den Realwerten. Mit 23,7 µm ist diese

maximale Abweichung (MAX) im Rahmen der Schwankungen des Korrosionsprozesses.

7 Diskussion

7.1 Beschreibung des KI-basierten Systems zur Korrosionsprognose

Das in dieser Arbeit entwickelte System kann Geometrie- und Simulationsdaten mit Methoden der KI kombinieren. Die Methodik dieser Arbeit, sobald implementiert, kann dann als Modell verstanden werden. Mit diesem Modell ist es möglich, automatisiert das Korrosionsverhalten von Fahrzeugkarosserien vorherzusagen. Um die Funktionsweise und die Leistungsfähigkeit des Ansatzes einzuordnen, wird im Folgenden das System spezifiziert.
Die Kapitel 4, 5 und 6 beschreiben nicht nur eine Methodik zur Vorhersage des Korrosionsverhaltens eines Fahrzeugs, sondern auch die Schritte, um effizient zu einer Implementierung zu gelangen. Hier ist mit Implementierung beispielsweise das Training von Neuronalen Netzen oder das Vergleichen unterschiedlicher Auswertungsansätze gemeint. Das Ziel des Modells ist jedoch die automatisierte Auswertung des Korrosionsverhaltens eines Gesamtfahrzeugs. In diesem Kontext muss die Implementierung nicht weiter betrachtet werden. Reduziert man die vorgestellte Methodik also auf die Anwendung des Systems, so ergibt sich der Ablauf wie in Abbildung 7.1 dargestellt. Das System gliedert sich in mehrere Teile:

1. Datenimport: Import von CAD-Daten und ERP-Informationen, Import von SPH-Simulationen zur Abbildung der Elektrolytbenetzung, Import von FEM-Simulationen.

2. Korrosionsrisiko: Bestimmung des Korrosionsrisikos einer Fahrzeugoberfläche. Dies ist durch die Unterteilung der Fahrzeugoberfläche mittels Klassifikationsnetzwerken umgesetzt.

3. Elektrolytbeanspruchung: Integration der Fluidsimulation auf Oberflächenelemente.

4. Korrosionsprognose: Interpolation der vorhandenen FEM-Simulationen, Vorhersage jedes einzelnen Oberflächenelements mit Hilfe eines NARX-Netzwerks.

5. Ergebnisdarstellung: Ausgabe des Korrosionsrisikos und der Korrosionsprognose zur Visualisierung.

Diese Struktur ermöglicht eine automatisierte Verarbeitungskette von Rohdaten bis hin zu interpretierbaren Ergebnissen und stellt damit ein aussagekräftiges System dar. Das gesetzte Ziel der Arbeit wird erreicht. Die Vorhersage des

Korrosionsverhaltens eines Gesamtfahrzeugs unter Berücksichtigung konstruktiver Maßnahmen, Elektrolytbeanspruchung und geometrischen Einflüssen ist umgesetzt.

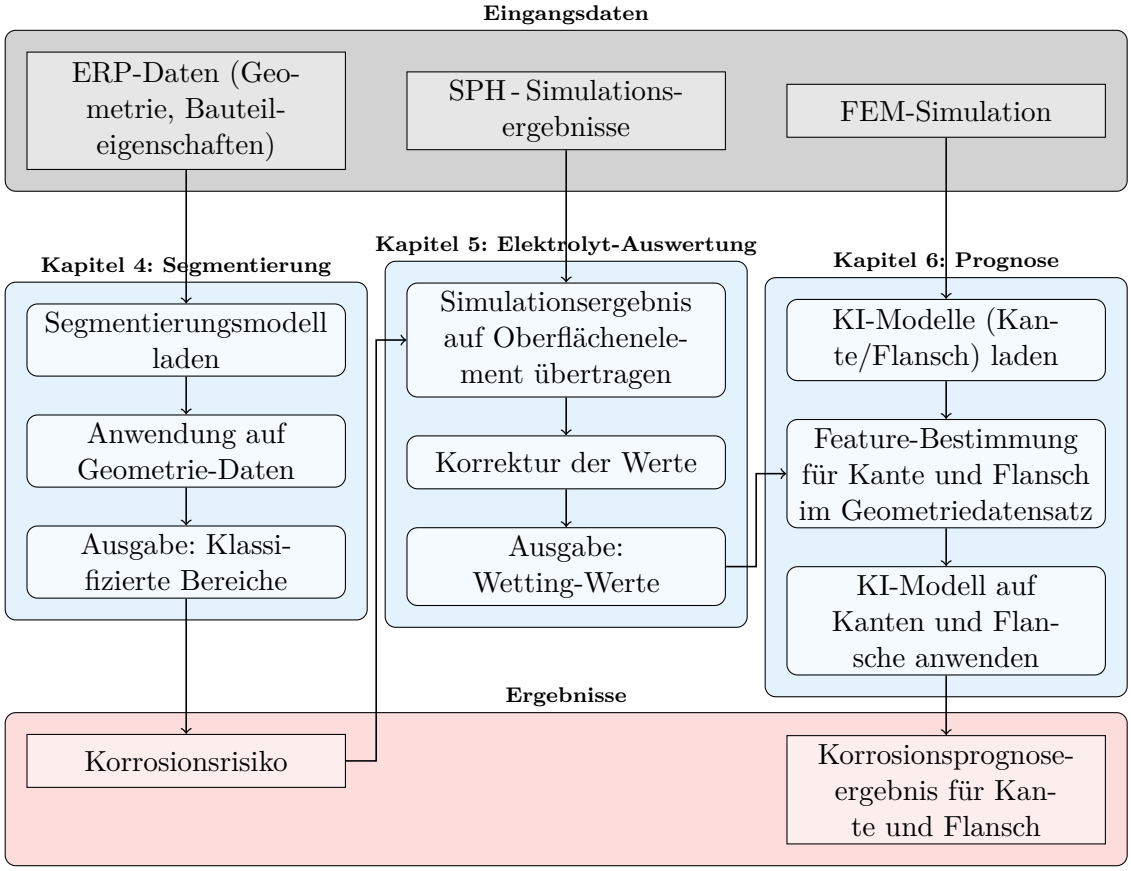


Abb. 7.1: Übersichtsworkflow der in dieser Arbeit vorgestellten Anwendung.

7.1.1 Diskussion des Korrosionsprognosesystems im Kontext der Explainable AI

Hybride ML-Architekturen

Die in dieser Arbeit erzielten Ergebnisse zur Korrosionsprognose am Gesamtfahrzeug sind Produkte eines hybriden Systems: ein ML-gestütztes Modul (Klassifizierung korrosionsrelevanter Bereiche) erzeugt ein auslesbares und menschenverständliches Zwischenergebnis, das anschließend durch regelbasierte Logik (Interaktion zwischen einzelnen Maßnahmen und Korrosionsrisikostellen) verarbeitet wird. Genauso ist die Auswertung der Elektrolytsimulation auf Regeln basiert. Die Ergebnisse dieses Moduls werden in menschenverständlicher, auslesbarer Form bereitgestellt. Die KI-basierte Korrosionsprognose benutzt eindeutig interpretierbare Eigenschaften zur Vorhersage des Korrosionsverhaltens. Die vier in Kapitel 2.1.6 beschriebenen konzeptionellen Einflüsse eines erklärbaren Systems können betrachtet werden.

Bewertung des Korrosionsprognosesystems mittels Desingprinzipien für erklärbare Systeme

Die vier in Kapitel 2.1.6 dargestellten Prinzipien lassen sich unmittelbar auf die erarbeiteten Ergebnisse übertragen:

1. Modularität: Die Trennung in Korrosionsrisikobestimmung, Simulationsauswertung und ML-basierten Korrosionsprognose folgt dem Konzept der Modularität.

2. Nachvollziehbarkeit: Für jede Berechnung und Logik existiert eine definierte Ablaufbeschreibung. Damit ist das Ergebnis nachvollziehbar und rückverfolgbar.

3. Menscheninterpretierbare Zwischenrepräsentationen: Das Ergebnis jedes Moduls (Klassifizierung, Wetting, Korrosionsprognose) ist menschenverständlich und kann interpretiert und eingeordnet werden. Dazu ist ein Interface implementiert, um diese Ergebnisse zu visualisieren.

4. Human-in-the-Loop an kritischen Punkten: Der dargestellte Workflow ermöglicht es, domänenkundigen Experten die Entscheidungen des Systems nachzuvollziehen. Zwischenergebnisse können durch Experten interpretiert und verwendet werden.

Fazit für das Korrosionsprognosesystem

Der konzeptionelle Aufbau des entwickelten Systems ist durch eine transparente erklärbare Architektur des Korrosionsprognosesystems bestimmt. Dies ist einerseits durch seine Modularität gegeben, bei der einzelne Schritte des Ablaufs in erklärbaren und menschenverständlichen Zwischenergebnissen zusammengefasst sind. Diese menschenverständliche Semantik ist in den Gleichung 5.12 und Gleichung 5.11 gezeigt. Beide Formeln ergeben Ergebnisse, die erklärbar und zuordenbar sind. Darüber hinaus besitzt das System einen klar definierten Informationsfluss mit Zwischenergebnissen, die interpretierbar und visualisierbar sind. So wird eine hohe Erklärbarkeit dieser in der Arbeit vorgestellten Korrosionsprognose erreicht.

7.2 Diskussion der Laufzeit des Korrosionsprognosesystems

Für die Anwendung dieses Systems ist die Laufzeit des Gesamtsystems relevant. Ein System, das genauso lang zur Berechnung braucht, wie es dauert, physische Probekörper aufzubauen, widerspricht der Motivation dieser Arbeit. Daher ist eine Betrachtung der benutzten Ressourcen und der Berechnungszeiten notwendig.
Die Durchführung der SPH-Simulationen erforderte erhebliche Rechenressourcen. Je nach Partikelauflösung und Modellgröße variieren die Anforderungen

an Rechenzeit und Speicher deutlich. Die in dieser Arbeit verwendeten Fluidsimulationen eines Gesamtfahrzeugs bewegen sich in der Rechenzeit im Bereich von einer Woche. Diese Rechenzeiten werden hier jedoch nicht berücksichtigt. Es wird sich lediglich auf die in Abbildung 7.1 beschriebenen Schritte bezogen. Die dafür verwendeten Workstations sind in Tabelle 7.1 beschrieben.

	Workstation 1	Workstation 2
Prozessor	Intel Core i7-8700K 3.70GHz	Intel Core i7-8700K 3.70GHz
Grafikkarte	NVIDIA Quadro P2000	NVIDIA Quadro M2000
RAM	64GB	64GB
Speicher	500GB	500GB

Tabelle 7.1: Genutzte Hardware

Das in Kapitel 4 verwendete Gesamtfahrzeug (Abbildung 4.5, 4.8, 4.11) setzt sich aus 1352 Einzelbauteilen zusammen. Diese 1352 Einzelbauteile bestehen insgesamt aus 19,0 Millionen Dreiecken. Dazu kommen die Datensätze für jede einzelne Maßnahme. Damit stellen sich die Datensätze wie in Tabelle 7.2 dar. Eine Kanten- und Flanschklassifizierung erkennt in diesem Datensatz 1,5 Millionen Kantenelemente und 3,12 Millionen Flanschelemente.

Datensatz	**Einzelbauteile**	**Dreiecke** $\setminus \times 10^6$
Rohbau	1352	19,0
Klebstoff	1841	1,7
NAD	475	2,5
Wachs	1134	2,2

Tabelle 7.2: Übersicht der Datensätze mit Anzahl an Einzelbauteilen und Dreiecken

Mithilfe der Workstation 1 aus Tabelle 7.1 werden die Rechenzeiten für das System wie folgt angegeben:

1. Durchführung der Klassifizierung: 37949 Sekunden
   a) Ladezeit Geometrien: 9 - 10 Sekunden je Datensatz
   b) Korrektur der Datensätze und Aufbau des AABB-Tree: 18 - 24 Sekunden
   c) Berechnen der Dreieckszentren: 1,4 - 3 Sekunden
   d) Speichern der Klassifizeriungsergebnisse: 22 Sekunden
2. Integration der Elektrolytauswertung: 20042 Sekunden
   a) Laden der binarisierten Simulationsergebnisse: 105 Sekunden
   b) Berechnung der Wetting-Werte von Kanten: 420 Sekunden
   c) Berechnung der Wetting-Werte von Flanschen: 19517 Sekunden

3. Durchführung der Korrosionsprognose: 161 Sekunden
    a) Berechnung der Kanteneigenschaften: 131 Sekunden
    b) Anwendung des Kantenprognosenetzwerks: 5 Sekunden
    c) Berechnung der Flanschinformationen und Anwendung des Flanschprognosenetzwerks: 25 Sekunden

Label	**Gesamt** \s	**Eigenschaften** \s	**Inferenz**\s	**Postproc.**\s
Kante	10189.93	10131.68	57.94	0.02
Klebstoff	17646.92	17605.00	40.44	0.00
NAD	2724.65	2699.07	25.21	0.00
Wachs	2747.73	2709.02	38.18	0.00
Flansch	4643.44	4626.91	16.28	0.01

Tabelle 7.3: Klassifizierungszeiten für jedes Label

Es ergibt sich eine Gesamtsystemdauer von 58152 Sekunden, also 16,15 Stunden. Tabelle 7.3 zeigt, wo die laufzeitintensiven Teile des Systems liegen und bestätigt damit auch die Methodik dieser Arbeit.

Klassifizierungsdurchführung

Mit 37949 Sekunden beansprucht die Korrosionsrisikobestimmung den größten Teil der Laufzeit. Dabei verteilt sich die Laufzeit auf die unterschiedlichen Klassifizierungen. Die Kantenklassifizierung braucht im Vergleich zu den Klassifizierungen Wachs, NAD und Flansch deutlich länger. Der beschriebene Datensatz besteht aus 21 Millionen Dreiecken. Da jedes dieser Dreiecke theoretisch eine Kante sein kann, muss das System über jede einzelne Kante iterieren. Im Falle des Flansches und der beiden Maßnahmen NAD und Wachs, existiert der Ansatz der AABB, welcher eine extrem hohe Laufzeitreduktion mit sich bringt, obwohl die Berechnung der Eigenschaften deutlich komplexer als im Fall der Kante ist.
Die Maßnahme Klebstoff besitzt mit 17646 Sekunden den größten Anteil an der Rechenzeit der Korrosionsschutzmaßnahmen. Dies liegt in der deutlich erhöhten Komplexität der Eigenschaftsberechnung. Die Berechnung der geometrischen Eigenschaften macht mit ca. 14265 Sekunden den Großteil der Eigenschaftsberechnung aus. Vergleicht man jedoch die Genauigkeit eines Klebstoffklassifizierungsnetzes mit geometrischen Eigenschaften mit einem Klebstoffklassifizierungsnetz ohne geometrische Eigenschaften (Tabelle 7.4), so wird der Einfluss der geometrischen Eigenschaften deutlich. Ein Klebstoffnetzwerk ohne die geometrischen Eigenschaften erzielt einen MCC von 0.9070, während das trainierte Netzwerk aus Kapitel 4.2 einen MCC von 0.9546 erreicht .
Dies bestätigt, neben den Ergebnissen der Shapley-Werte die Effizienz der geometrischen Features. Aufgrund der hohen Laufzeiten der Bestimmung der

Netzwerk	MCC
Netzwerk 1	0.9070
Netzwerk 2	0.9546

Tabelle 7.4: Vergleich der Klassifikationsleistung zweier Klebstoff-Netzwerke, Netzwerk 1 ohne geometrischen Eigenschaften, Netzwerk 2 mit geometrischen Eigenschaften

geometrischen Eigenschaften wird jedoch auf die Implementierung der geometrischen Eigenschaften in den Maßnahmen NAD und Wachs verzichtet, da dies die Laufzeit des Gesamtsystems stark erhöhen würde. Außerdem ist die Validierungsgenauigkeit des Wachsnetzwerks bereits hoch (Tabelle 4.18).
Mit der Berechnung der Eigenschaften ist der große Teil der Laufzeit betrachtet. Die Inferenz jeder Maßnahme fällt mit 0.2% der Laufzeit in der Klebstoffklassifizierung und 1.0% der NAD-Klassifizierung nicht ins Gewicht. Dies liegt an der geringen Größe der Netze.
Das Training der Netze ist nicht Teil der Durchführung einer Prognose. Ein Blick auf die Trainingszeit verdeutlicht, dass diese Trainingszeit im Vergleich zur Eigenschaftenberechnung deutlich geringer ausfällt. Das Training für ein Netzwerk der beschriebenen Größe bewegt sich bei der angegebenen Hardware zwischen 20 und 60 Sekunden.
Diese Ergebnisse zeigen, dass der eigentliche KI-Trainingsprozess – im Gegensatz zur Feature-Generierung – deutlich ressourcenschonender durchgeführt werden kann.

Elektrolytsimulationsauswertung

Die Laufzeit der Auswertung der Elektrolytsimulation spiegelt die Komplexität und Limitationen dieses Modells wider. Die Auswertung der Kantenelemente benötigt durch einen implementierten Octree nur 420 Sekunden. Mithilfe dieses Octrees kann eine Untermenge der Simulationspunkte geladen werden und die Punktwolke ausgewertet werden. Dies beschreibt bereits einen laufzeitoptimierten Ansatz der Auswertung und zeigt sich in der minimierten Laufzeit. Die extreme Auswirkung auf die Laufzeit wird in Tabelle 7.5 gezeigt. Ansatz A beschreibt die Auswertung der Kantenbenässung über einen klassischen Ansatz, in dem 19 Millionen Simulationspunkte ausgewertet werden. Dabei wird für jede Kante eine euklidische Distanzberechnung zwischen Punkten durchgeführt. Ansatz B beschreibt einen Ansatz, bei dem mit den Simulationsdaten ein Octree implementiert wird. Mithilfe dieses Octrees werden im ersten Schritt eine AABB gebaut, ohne jeden einzelnen Simulationspunkt euklidisch zu betrachten. Die Punkte innerhalb dieser AABB werden dann euklidisch verglichen, um eine genaue OBB zu bauen.
Damit wird die Laufzeit mit Hilfe des Octree-Ansatzes um 99.6% reduziert. Dieser Ansatz wird auch für die Auswertung der Flansche benutzt.
Die erhöhte Laufzeit von 19517 Sekunden wird dabei auf die Anpassung der

	Ansatz A	Ansatz B
Laufzeit \s	1045	5.4

Tabelle 7.5: Vergleich der Laufzeit mit und ohne Octree

Wetting-Werte sowie die Berechnung der Materialpaarungen zurückgeführt. Im Rahmen dieser Betrachtung werden die Flanschflächen in kleine Oberflächenelemente unterteilt. Durch diese Unterteilung werden auch die Materialpaarungen bestimmt, die für die ML-basierte Korrosionsprognose nötig sind. Die hohe Laufzeit basiert jedoch im Besonderen auf der Anpassung der Wetting-Werte der einzelnen Dreiecke. Das Laden der Simulationen ist durch Binarisieren der Simulationsdaten auf 105 Sekunden reduziert.

KI-basierte Korrosionsprognose

Die Laufzeit der KI-basierten Korrosionsprognose fällt im Gesamtsystem Korrosionsprognose nicht weiter ins Gewicht.
Um eine Bewertung der Laufzeit des Gesamtsystems Korrosionsprognose durchzuführen, bietet sich der Vergleich mit anderen digitalen Simulationsverfahren wie der Elektrolytsimulation an, aber auch mit der physischen Alternative des Gesamtsystems Korrosionsprognose. Im Zuge dessen wird die Gesamtdurchlaufzeit einer Korrosionsprognose eines Gesamtfahrzeugs auf einen Tag, also 24 Stunden, aufgerundet.

	Gesamtsystem Korrosionsprognose	Elektrolyt simulation	Korrosionstest
Laufzeit \d	1	≈ 10	56

Tabelle 7.6: Vergleich der Laufzeit unterschiedlicher Vorhersagetools

Tabelle 7.6 vergleicht das Gesamtsystem Korrosionsprognose mit der Elektrolytsimulation und dem physischen Korrosionstest. Hier fällt die deutlich geringere Berechnungszeit der Korrosionsprognose auf, im Besonderen zum Korrosionstest. Die 56 Tage des Korrosionstests beziehen sich dabei nur auf die Raffung des Tests und nicht auf die Vorbereitungszeit des physischen Versuchsträgers. Damit zeigt sich das Korrosionsprognosesystem als deutlich schnelleres Vorhersagetool als der Korrosionstest.
Die Simulation des Elektrolyten wird als Vergleichswerkzeug für die Korrosionsprognose genutzt. Dies liegt nicht nur an ihrer Rolle als digitales Simulationstool, sondern auch an der zeitlichen Abfolge. Die Ergebnisse der Simulation werden im Korrosionsprognosesystem benötigt und müssen daher gemeinsam betrachtet werden. Das in dieser Arbeit dargestellte System Korrosionsprognose benötigt diese Elektrolytsimulation als Input. Auf diese Weise kann man den Vergleich im Rahmen eines Arbeitsablaufs von Elektrolytsimulation bis hin zur Gesamtsystem-Korrosionsprognose anstellen. Dies ermöglicht eine Gegenüberstellung innerhalb des spezifischen Workflows. Betrachtet man beide Systeme

im Rahmen der Automobilentwicklung und deren Entwicklungsstufen, so steht nur begrenzt Zeit zur Verfügung, die sich ein solcher Gesamtprozess teilen muss. Daher ist die kurze Laufzeit eines Gesamtsystems Korrosionsprognose als sehr positiv zu bewerten.

7.3 Speicherbedarf des Korrosionsprognosesystems

Der Speicherbedarf des Systems ist maßgeblich durch Simulation und Geometriedaten definiert. Im Rahmen eines Gesamtfahrzeugs stellen sich die Datengrößen wie in Tabelle 7.7 dar.

Datensätze	**Datensatzgröße**
Geometrierohbaudaten	5,9 GB
Geometriedaten Klebstoff	0,09 GB
Geometriedaten NAD	0,08 GB
Geometriedaten Wachs	0,10 GB
ERP-Daten	0,002 GB
Simulationsdaten Elektrolyt	4,56 GB
FEM-Simulationsdaten	2,45 GB

Tabelle 7.7: Datensatzgrößen im Korrosionsprognosesystem für ein Gesamtfahrzeug

Damit ergeben sich als Mindestspeichergröße 12 GB. Diese werden jedoch auf 20 GB erweitert, falls Simulationsdaten nicht binarisiert, sondern als Textdatei vorliegen. Das Zwischenspeichern des Korrosionsrisikos beansprucht weitere 4 GB. Daraus ergibt sich eine Mindestspeichergröße von 24 GB. Im Rahmen der Implementierung mussten Maßnahmen getroffen werden, um den Arbeitsspeicher der Workstation nicht zu überfüllen. Daher empfiehlt sich ein System mit mindestens 64 GB. Zu dem Mindestspeicher des GRAM kann jedoch eine Mindestgröße definiert werden. GRAM darf, für die in dieser Arbeit durchgeführten Klassifizierungen, nicht geringer sein als der der vorgestellten Hardware. Da die Strahlenberechnungen parallelisiert auf der Grafikkarte berechnet werden, kann bei kleineren Grafikkarten der Speicher nicht ausreichen, um große Bauteile zu berechnen.

7.4 Diskussion der Trainingsdatensätze und deren Erweiterungspotential

Ein zentrales Element der Systemdefinition ist die verwendete Datenbasis. Diese umfasst unter anderem

1. Ground-Truth-Daten der Klassifizierungen

2. Numerisch erzeugte Simulationsdaten

7.4.1 Trainingsdatensätze der Korrosionsrisikoprognose

Kritisch für die Leistungsfähigkeit des KI-Systems zur Korrosionsrisikoprognose sind die Ground-Truth-Daten. Die Genauigkeit der Klassifizierungsmethoden ist bereits sehr hoch. Daher kann die Erweiterung der Trainingsdatensätze einerseits im Kontext der Erhöhung der Genauigkeit betrachtet werden, andererseits in der verbesserten Abdeckung von Edge-Cases.
Ein möglicher Grund, um die Trainingsdatensätze mit Ground-Truth Daten aus einem Gesamtfahrzeug zu stützen, ist die Annahme, dass die Trainingsdatensätze ein Gesamtfahrzeug nicht gut abbilden. Dem widerspricht allerdings der Vergleich der Validierungsergebnisses des Netzwerks und der Validierung am Gesamtfahrzeug. In allen Maßnahmen ist die Genauigkeit des Netzwerkes am Gesamtfahrzeug genauso oder höher als bei der Validierung des Netzwerks. Damit bestätigt sich die Annahme, dass die Trainingsdatensätze auch ein Gesamtfahrzeug gut abbilden. Darüber hinaus würde das erweitern der Trainingsdatensätze die Abdeckung von Edge-Cases in der Klassifizierung vergrößern, da die Dichte an komplexen Interaktionen zwischen Maßnahmen und dem Gesamtfahrzeug geringer ist als in den bestehenden Trainingsdatensätzen.Das Korrosionsprognosesystem bietet die Möglichkeit, das ermittelte Korrosionsrisiko zu exportieren und händisch zu korrigieren, um damit die Datenbasis iterativ zu erweitern. Einerseits ermöglicht dies die Implementierung des Transfer Learnings aus Kapitel 4, andererseits kann es dazu dienen, die Datenbasis zu erweitern. Durchgeführte Klassifizierungen können nach einer Korrektur der Daten zurück in das Training gegeben werden. Die Gefahr dabei ist es jedoch, bestimmte Randfälle oder besonderes Verhalten unterzurepräsentieren. Die Größe der Datensätze ist ausreichend für die Größe der Klassifizierungsnetze. Die Anzahl der Parameter der Netze liegt zwischen 2000 und 10000 trainierbaren Parametern. Damit sind die zur Verfügung stehenden Datensätze ausreichend groß für die Anzahl der trainierbaren Parameter. Diese Aussage bestätigt sich auch durch das Trainingsverhalten der Netze, die deutlich vor der maximalen Epochenzahl konvergieren. Dabei ist das Abbruchkriterium die Lernrate. Auch dieses Ergebnis zeigt, dass das Netz bereits mit ausreichend großen Datenmengen trainiert wurde. Damit kann der Datensatz um ausgewählte Edge-Cases erweitert werden, also Fälle, die in den Datensätzen bisher so nicht auftreten. Eine Möglichkeit, dies zu implementieren, ist es, nur flasch klassifizierte Dreiecke als Erweiterung der Trainingsdatensätze zu übergeben.
Beispielhaft für dieses Verhalten kann aus Tabelle 4.16 abgeleitet werden. Bei der Dokumentation des Wachsdatensatzes existieren zwei unterschiedliche Arten, wachsgeschützte Flächen zu dokumentieren. Die Art der Dokumentation in Baureihe A ist hier der limitierende Faktor für den Trainingsdatensatz. Diese neue Art der Dokumentation ist lediglich in diesem Datensatz vorhanden. Zieht man nun ähnlich viele Datensätze wie beispielsweise bei Tabelle 4.2 hinzu, so wird ein deutliches Ungleichgewicht erzeugt. Die Dokumentationsart der Baureihe D wird dominant. Die Gefahr besteht, dass die Qualitätsgüte der

Wachsklassifizierung dadurch abnimmt. Will man die Datengrundlage erhöhen, so bietet es sich an, Datensätze mit der neuen und der alten Dokumentationsart parallel zu erhöhen.

7.4.2 Trainingsdatensätze der KI-basierten Korrosionsprognose

Die KI-basierte Korrosionsprognose basiert auf FEM-Simulationen für Kanten und Flansche, die durch Interpolation erweitert werden. Dabei berücksichtigen diese Prognosen folgende Informationen:

1. Kante:
   a) Winkel
   b) Geometrische Lage
   c) Material
2. Flansch:
   a) Geometrie des Flansches
   b) Materialpaarungen

Diese Informationen können erweitert werden. Dabei spielt im Besonderen die Beschichtung eine Rolle. Bisher wird die KTL-Schicht mit einem festen Schichtdickenwert parametrisiert. Berücksichtigt die Kanten- und Flanschsimulation die KTL-Schichtdicke der umliegenden Oberfläche, so kann dieser Einflussfaktor ebenfalls in die KI-basierte Korrosionsprognose eingehen. Ferner können unterschiedliche Vorbehandlungen eine Rolle im Korrosionsverhalten spielen. Daher sollten diese ebenfalls in den zugrunde liegenden Simulationen variiert werden. Zuletzt kann die Berücksichtigung unterschiedlicher Materialien verfeinert werden. Betrachtet man jedoch beispielsweise die Arbeit von Schneider [53], so kann beobachtet werden, dass unterschiedliche hochfeste Stähle nur geringe Abweichungen nach sich ziehen.
Ein weiterer Aspekt, der eine große Rolle im Korrosionsverhalten eines Gesamtfahrzeugs spielt, sind die Prozessparameter. Diese können über einen Variationsbereich abgebildet werden.

7.5 Stärken und Limitierungen des Systems

Das Gesamtsystem Korrosionsprognose besitzt einige Stärken sowie Schwächen. Die Stärken sind:

1. ein Datenmodell als single-source-of-truth
2. Erweiterbarkeit des Gesamtsystems
3. Nachvollziehbarkeit des Gesamtsystems
4. Laufzeit des Gesamtsystems

Nachteile dieses Systems sind die fehlende Abbildbarkeit der in der Produktion entstehenden Schwankungen. Im Folgenden werden die Vor- und Nachteile genauer beschrieben.

Datenmodell

Das Gesamtsystem ermöglicht es, ein Modell zu konstruieren, in dem alle korrosionsrelevanten Daten aggregiert sind. Dies ermöglicht Korrosionsentwicklern, eine Entscheidung auf der Basis aller verfügbaren Daten zu treffen. Ein Interface zu diesem Datenmodell kann wie in Abbildung 7.2 aussehen.

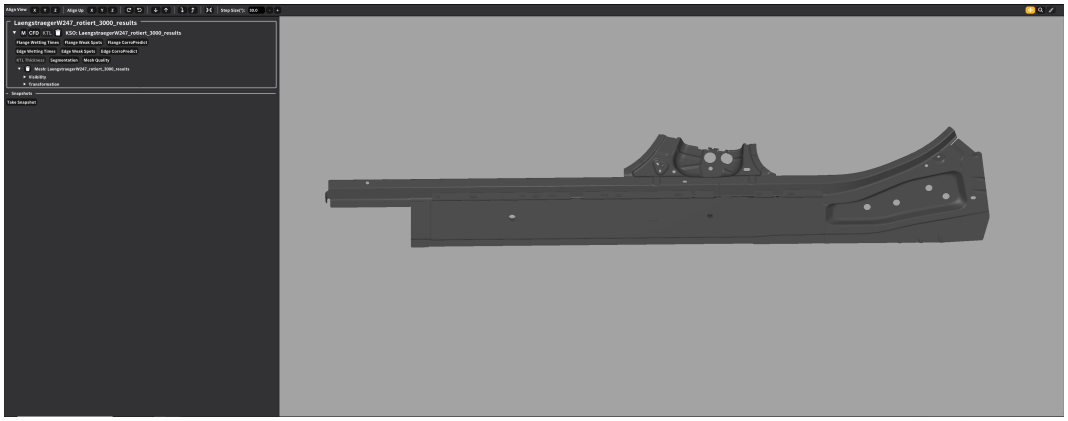


Abb. 7.2: Interface des Datenmodells

Erweiterbarkeit des Systems

Ein weiterer Vorteil des entwickelten Gesamtsystems zur Korrosionsprognose liegt in seiner sehr guten Erweiterbarkeit. Dieses Prinzip zieht sich konsistent durch alle drei Systembausteine Klassifikation, Elektrolytauswertung und KI-basierte Korrosionsprognose. So lässt sich das Korrosionsrisiko mithilfe der zugrunde liegenden Matrix (Gleichung 5.12) beliebig erweitern. Durch eine Ergänzung der Gleichung 6.1 um die Einflüsse zusätzlicher Label auf das Korrosionsrisiko können nicht nur weitere Korrosionsvermeidungsmaßnahmen integriert, sondern auch neue Labelklassen für bislang unberücksichtigte, korrosionskritische Bereiche eingeführt werden.

Nachvollziehbarkeit des Gesamtsystems

Durch den modularen Aufbau des Systems ist eine hohe Nachvollziehbarkeit gewährleistet. Jeder Verarbeitungsschritt kann separat betrachtet und dessen Einfluss visualisiert werden. Auf diese Weise wird das typische Black-Box-Problem klassischer KI-Ansätze vermieden. Statt sämtlicher Oberflächeneigenschaften und deren Bewertung in einem undurchsichtigen Gesamtmodell zu verbergen, entsteht ein „halbtransparentes“ System. Der Entwickler kann klar erkennen, aus welchen Gründen ein bestimmter Bereich als korrosionskritisch eingestuft wird.

Laufzeit des Gesamtsystems

Die in Kapitel 7.2 beschriebene Laufzeit ist eine Stärke des Gesamtsystems. Mit einem Tag Laufzeit ist diese Implementierung ein valides Tool zur Bewertung eines Gesamtfahrzeugs. Diese Laufzeit ermöglicht eine realistische Anwendung im Rahmen der Entwicklung eines Gesamtfahrzeugs. Der modulare Aufbau ermöglicht weiterhin, dass Teilergebnisse des Systems bereits erstellt werden können, bevor das vollständige Datenmodell vorliegt.

Fehlende Abbildbarkeit von Schwankungen im Produktionsprozess

Eine Schwäche des Systems ist die fehlende Abbildung von Produktionsprozessschwankungen. Davon sind sowohl die Korrosionsrisikoprognose als auch die KI-basierte Korrosionsprognose betroffen. Beispiele dafür wurden in der Diskussion der Korrosionsprognose gezeigt. Im Rahmen dieser Arbeit existiert keine Datenbasis, um Informationen dazu in die Korrosionsprognose zu integrieren.

7.6 Bewertung des Korrosionsprognosesystems im Kontext der Forschungslücke

Das zentrale Ziel dieser Arbeit besteht in der Entwicklung eines KI-basierten Systems zur automatischen, simulationsgestützten Vorhersage des Korrosionsverhaltens an Gesamtfahrzeugen unter Berücksichtigung konstruktiver Maßnahmen, der Elektrolytbeanspruchung sowie spezifischer Bauteileigenschaften. Diese Zielerfüllung soll im Folgenden diskutiert werden.

7.6.1 Bewertung der Korrosionsrisikoprognose

Kapitel 4 beschreibt eine Methodik, die in der Lage ist, klassische Konzepte der Korrosionsvermeidung in der Automobilindustrie zu berücksichtigen und in ein Korrosionsrisiko zu übersetzen.

Klebstoffklassifizierung

Diese Arbeit zeigt in Kapitel 4.2 die Implementierung einer Methode zur Erkennung von klebstoffgeschützten Flächen. Mit einem MCC von 0.9573 (Tabelle 4.7) zeigt dieser Ansatz eine sehr hohe Genauigkeit. Diese bestätigt auch den Ansatz, die Trainingsdaten auf kleineren Datensätzen aufzubauen, und sie dann auf ein Gesamtfahrzeug zu übertragen.
Die in Abbildung 4.6a gezeigten Ergebnisse unterstreichen, dass die Methodik sehr gute Ergebnisse für eine Betrachtung eines Gesamtfahrzeugs liefert. Der Übertrag des Trainingsdatensatzes auf ein Gesamtfahrzeug lässt sich durch die gleichbleibend hohe Genauigkeit bestätigen (Tabelle 4.7). Daher kann angenommen werden, dass komplexe Geometrien, die im Gesamtfahrzeug vorkommen, bereits durch das Klebstoffverhalten an der Tür abgedeckt sind.

Nahtabdichtungsklassifizierung

In Kapitel 4.3 wird eine Methodik beschrieben, die durch NAD geschützte Bereiche erkennen kann. Mit einem MCC von 0.9034 zeigt das Netzwerk im Training eine hohe Genauigkeit. Diese bezieht sich, wie bei dem Klebstoff, auf die Validierung im Rahmen des Trainings. Die Genauigkeit am Gesamtfahrzeug ist mit 0.9662 noch höher. Auch hier zeigt sich die sehr gute Übertragbarkeit des Trainingsdatensatzes auf das Gesamtfahrzeug. Die Genauigkeit des Gesamtfahrzeugs liegt sogar über der Genauigkeit des Trainings.

Hohlraumkonservierungsklassifizierung

In Kapitel 4.4 wird eine Methode zur Erkennung von durch HRK geschützte Bereichen gezeigt. Die Trainingsgenauigkeit liegt bei einem MCC von 0.9230 (Tabelle 4.18), die Genauigkeit der Klassifizierung eines Gesamtfahrzeugs bei 0.9822 (Tabelle 4.21). Auch hier zeigt sich, dass das Übertragen von Trainingsdatensatz auf Gesamtfahrzeug funktioniert.
Abbildung 4.12b konnte ein Bereich gezeigt werden, in dem Dreiecke falsch segmentiert werden. Der Wachsdatensatz in diesem Bereich wird auf neue Weise dokumentiert. Anstelle von ausschließlich Röhren erfolgt die Dokumentation über Flächen. Durch Variationen in der Schichtdicke des Rohbaus können Fehlklassifizierungen entstehen. Es muss untersucht werden, ob durch zusätzliche Datensätze die Qualität der Klassifizierung verbessert werden kann.

Allgemeine Bewertung der Korrosionsrisikobewertung

Hinsichtlich der konstruktiven Maßnahmen konnte gezeigt werden, dass das entwickelte System in der Lage ist, typische Korrosionsschutzmaßnahmen wie Klebstoffe, Nahtabdichtungen oder Wachs in eine Korrosionsrisikoprognose einzubeziehen. Durch die explizite Berücksichtigung dieser Maßnahmen in den Eingangsdaten sowie in den resultierenden Features für das KI-Modell wurde eine deutliche Steigerung der Prognosequalität erreicht. Dabei reduziert dieser Ansatz der Erkennung der Schutzmaßnahmen die korrosionskritischen Elemente um ungefähr 50% (Tabelle 7.8).
Gleichwohl ist zu beachten, dass die Vielfalt der im Fahrzeugbau eingesetzten konstruktiven Maßnahmen noch deutlich größer ist als in den hier untersuchten Fällen. Für eine zukünftige Ausweitung des Systems wäre daher eine Erweiterung der Datenbasis auf weitere Konstruktionsprinzipien und Fahrzeugbereiche erforderlich.

7.6.2 Bewertung der Elektrolytsimulationsauswertung

Die Elektrolytbeanspruchung wird durch die Kopplung von SPH-basierten Simulationen mit der Korrosionsrisikoprognose abgebildet. Die Ergebnisse ermöglichen die Integration von bereits bestehenden Elektrolytsimulationen und damit eine Verfeinerung der Korrosionsrisikoprognose sowie einen zusätzlichen Inputfaktor für die KI-basierte Korrosionsprognose. Im Bereich der Korrosionsrisikoprognose implementiert diese Integration einen wichtigen Bestandteil

der Entwicklungsarbeit bei einem Korrosionsschutzkonzept, nämlich die Berücksichtigung, ob ein Bereich im Trockenen liegt. Bei der Korrosionsprognose bieten die Werte der Elektrolytsimulation eine neuartige Möglichkeit der Integration in ein Prognosesystem.

7.6.3 Bewertung der KI-basierten Korrosionsprognose

Es konnte gezeigt werden, dass die Kombination aus geometrischen Merkmalen und FEM-Inputdaten eine differenzierte Bewertung des Korrosionsverhaltens ermöglicht. Damit ist ein Schritt hin zur automatischen Identifikation kritischer Bereiche gelungen, der über die rein expertenbasierte Bewertung hinausgeht. Dennoch bleibt die Herausforderung, auch nicht-geometrische Einflüsse, wie Schwankungen aus dem Produktionsprozess, systematisch in die Vorhersage zu integrieren und damit die Qualität weiter zu erhöhen.

7.6.4 Bewertung des Gesamtsystems Korrosionsprognose

Zusammenfassend lässt sich festhalten, dass das in dieser Arbeit entwickelte System die eingangs formulierte Zielsetzung erfüllt. Es konnte gezeigt werden, dass die Kombination aus KI, Simulation und bauteilspezifischen Merkmalen eine belastbare Vorhersage des Korrosionsverhaltens erlaubt. Darüber hinaus bietet dieses System einen sehr guten Ansatz, das Korrosionsverhalten eines Fahrzeugs bereits im digitalen Zustand zu bewerten. Das System kann im Entwicklungsprozess immer dann Verwendung finden, wenn ein physischer Test im Zeitablauf zu langwierig ist, der Entwicklungsstand der Hardwareproduktion noch nicht weit genug fortgeschritten ist oder iterativ Konzepte verglichen werden sollen. Tabelle 7.8 beschreibt die Reduktion der Korrosionsrisikostellen mit den in dieser Arbeit integrierten Daten (Elektrolyt, NAD, Wachs, Klebstoff).

Risikostellen	ohne Red. \ $\times 10^6$	red. Zahl \ $\times 10^6$	Reduktion \%
Kanten	3.00	2.80	93.48%
Flansch	3.12	1.94	62.80%

Tabelle 7.8: Reduktion der im Fahrzeug erkannten korrosionskritischen Bereiche mit den in dieser Arbeit integrierten Daten

Trotz des Bedarfs, weitere Einflüsse zu integrieren, bietet das System eine Möglichkeit, basierend nur auf digitalen Daten, die Karosserie eines Gesamtfahrzeugs zu analysieren, zu einem Zeitpunkt, in dem noch keine physischen Testobjekte existieren. Damit bietet das entwickelte Gesamtsystem der Korrosionsprognose einen beachtlichen Mehrwert für die Entwicklung von Korrosionsschutzkonzepten.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

In der vorliegenden Arbeit wird erstmals eine KI-basierte Vorhersage des Korrosionsrisikos und des Korrosionsfortschritts eines gesamten Karosserierohbaus auf Basis digitaler Daten und der Elektrolytbeanspruchung beschrieben. Darüber hinaus werden erstmals durch Korrosionsschutzmaßnahmen geschützte Oberflächen erkannt und in der Korrosionsrisikobetrachtung berücksichtigt.
Dazu wird ein erweiterbares, digitales Modell erstellt, mit dessen Hilfe unterschiedliche korrosionsschutzrelevante Einflüsse (Blechkanten, Flansche, Klebstoffe, Hohlraumkonservierung, Nahtabdichtung, Wasserbelastung) integriert werden. Hierfür wird eine Matrix aufgebaut, in der die relevanten Korrosionseinflüsse zu jedem Oberflächenelement hinterlegt sind. Diese in dem Modell zusammengefassten Einflüsse gehen in einen ML-Algorithmus ein und ergeben ein prognostiziertes Korrosionsverhalten.
Für die Klassifizierung der durch die Korrosionsschutzmaßnahmen Klebstoff, Hohlraumkonservierung und Nahtabdichtung geschützten Bereiche, werden drei Klassifizierungsnetzwerke implementiert.

1. Klebstoff-Netzwerk: auf Basis von 226 Eigenschaften wird die durch Klebstoff geschützte Oberfläche des Fahrzeugs mit einem MCC von 0.9573 bestimmt und die Oberflächenelemente mit einem MCC von 0.9632.

2. NAD-Netzwerk: auf Basis von 54 Eigenschaften wird die durch NAD geschützte Oberfläche des Fahrzeugs mit einem MCC von 0.9895 bestimmt und die durch NAD geschützten Oberflächenelemente mit einem MCC von 0.9662.

3. HRK-Netzwerk: auf Basis von 69 Eigenschaften werden durch HRK geschützte Oberflächen des Fahrzeugs mit einem MCC von 0.9822 bestimmt und die durch HRK geschützten Oberflächenelemente mit einem MCC von 0.9836.

Sowohl auf die Fläche als auch auf die Oberflächenelemente bezogen, ergeben die Netzwerke eine sehr hohe Genauigkeit. Die drei Netzwerke sowie ein Flanschnetzwerk werden mit Hilfe von Shapley-Werten analysiert. Dadurch wird die Anzahl der genutzten Eigenschaften in der Flanschklassifizierung von 49 auf 34 reduziert, ohne signifikant an Genauigkeit zu verlieren. Die Reduktion

des MCC von 0.9537 auf 0.9530 kann dabei vernachlässigt werden. Gleichzeitig wird die Trainingszeit von 460,06 s auf 379,3 s reduziert. Bei den restlichen Netzen konnte auf Basis der Shapley-Werten kein Reduktionspotential identifiziert werden. Die Shapley-Werte sowie die hohe Genauigkeit der Netzwerke bestätigen die Relevanz der eingeführten Eigenschaften.
Zur Integration der Elektrolytbeanspruchung in die Korrosionsprognose werden die Ergebnisse von drei bestehenden CFD-Simulationen mittels eines neuartigen Algorithmus auf die relevanten Oberflächenelemente übertragen.
Ein NARX-Zeitreihennetzwerk ermöglicht nun die Vorhersage des Korrosionsverhaltens eines Oberflächenelements in Flanschen, auf Basis von Material und Flanschfläche. Als Trainingsdatensatz werden Korrosionsverläufe von 12 Materialkombinationen aus FEM-Simulationen betrachtet. Das auf diesen Daten trainierte NARX-Netzwerk weist einen RMSE von 6,9 µm, bezogen auf die Korrosionstiefe, auf. Die Prognose wird mit den ausgewerteten Simulationen kombiniert und je Oberflächenelement durchgeführt.
Zur Validierung der Flanschkorrosionsprognose werden die Prognoseergebnisse mit 20 Flanschstellen zweier im Korrosionstest geprüfter Fahrzeuge hinsichtlich der Korrosionstiefe verglichen. Die ausgewählten Flansche der getesteten Fahrzeuge bewegen sich im Bereich 0 µm bis 154 µm Korrosionstiefe. Die mittlere Abweichung der Prognose zur Realität beträgt dabei 57 µm, durch Produktionseinflüsse bereinigt reduziert sich die Abweichung auf 17 µm. Dies entspricht bei reellen Produktionsbedingungen einer sehr guten Übereinstimmung. Diese hohe Genauigkeit bestätigt die beschriebene Elektrolytsimulationsauswertung sowie die Methodik der Flanschprognose.
Weiterhin wird eine bestehende Kantenkorrosionsprognose um den Elektrolyteinfluss erweitert, indem mehrere Bereiche in Abhängigkeit des Wetting-Werts und des Winkels ausgewertet werden. Auch hier bestätigt sich anhand des Abgleichs an einem Realfahrzeug die gewählte Einbindung der CFD-Simulation.
Mit den beschriebenen Methodiken ist es nun erstmals möglich, das Korrosionsverhalten einer kompletten Fahrzeugkarosserie zu prognostizieren.

8.2 Ausblick

Das entwickelte System ist modular aufgebaut und bietet sehr gute Möglichkeiten, in allen Teilen erweitert zu werden.
So bietet die entwickelte Matrix im Bereich der Klassifizierungen von korrosionsrelevanten Einflüssen die Möglichkeit, weitere Korrosionseinflüsse zu berücksichtigen.
Dazu müssen gegebenenfalls weitere Korrosionsrisikobereiche und Korrosionsvermeidungsmaßnahmen berücksichtigt werden.
Darüber hinaus kann die Architektur der entwickelten Netzwerke weiter analysiert und optimiert werden um einen noch höheren MCC zu erreichen. Im Kontext dieser Arbeit wurde die Architektur der Neuronalen Netze über die Schichtanzahl im Rahmen des Hyperparametertunings betrachtet. Andere Schichtarchitekturen oder Aktivierungsfunktionen können geringfügige Optimierungen des Ergebnisses ermöglichen.

Auch im Bereich der Elektrolytsimulation kann das entwickelte System weiterentwickelt werden. Für die in der vorliegenden Arbeit eingesetzte Elektrolytsimulation existieren Untersuchungen, die die Simulationsergebnisse bestätigen. Deshalb kann eine Äquivalenz von ToW und wt angenommen werden. Realtests bestätigen diese Annahmen, können im Kontext der Korrosionssimulation aber noch bestätigt werden. Es besteht weiterhin die Möglichkeit das Modul der Elektrolytsimulationsauswertung auf weitere Simulationen anzuwenden. Die bestehende Grundlogik sowie die Datenarchitektur kann hierfür genutzt werden. So besteht zum Beispiel die Möglichkeit eine Lack- oder KTL-Simulation in das System zu integrieren. Dadurch könnte das Korrosionsrisiko weiter verfeinert werden.
Neben dem Korrosionsrisiko können diese zusätzlichen Informationen auch im Bereich der KI-basierten Korrosionsprognose zu einer verbesserten und genaueren Prognose führen. Neben dieser Verbesserung existieren auch weitere Möglichkeiten. So kann die Materialbetrachtung der Flanschprognose weiter unterteilt werden. Bisher werden in der Korrosionsprognose vier unterschiedliche Kategorien der Materialien betrachtet. Diese Kategorisierung kann verfeinert werden. Dazu muss die Datengrundlage der FEM-Simulationen verbreitert werden, auf der die Korrosionsprognose basiert. Darüber hinaus können weitere Einflussfaktoren wie Beschichtungen berücksichtigt werden. Dies ermöglicht eine genauere Validierung der Kantenkorrosionsprognose.
Das in dieser Arbeit entwickelte System bietet eine ausgezeichnete Basis für weitere Arbeiten, die das System erweitern. Es kann einen wichtigen Beitrag leisten die Entwicklungszyklen in der Automobilproduktion zu verkürzen.

A FEM-Datengrundlage

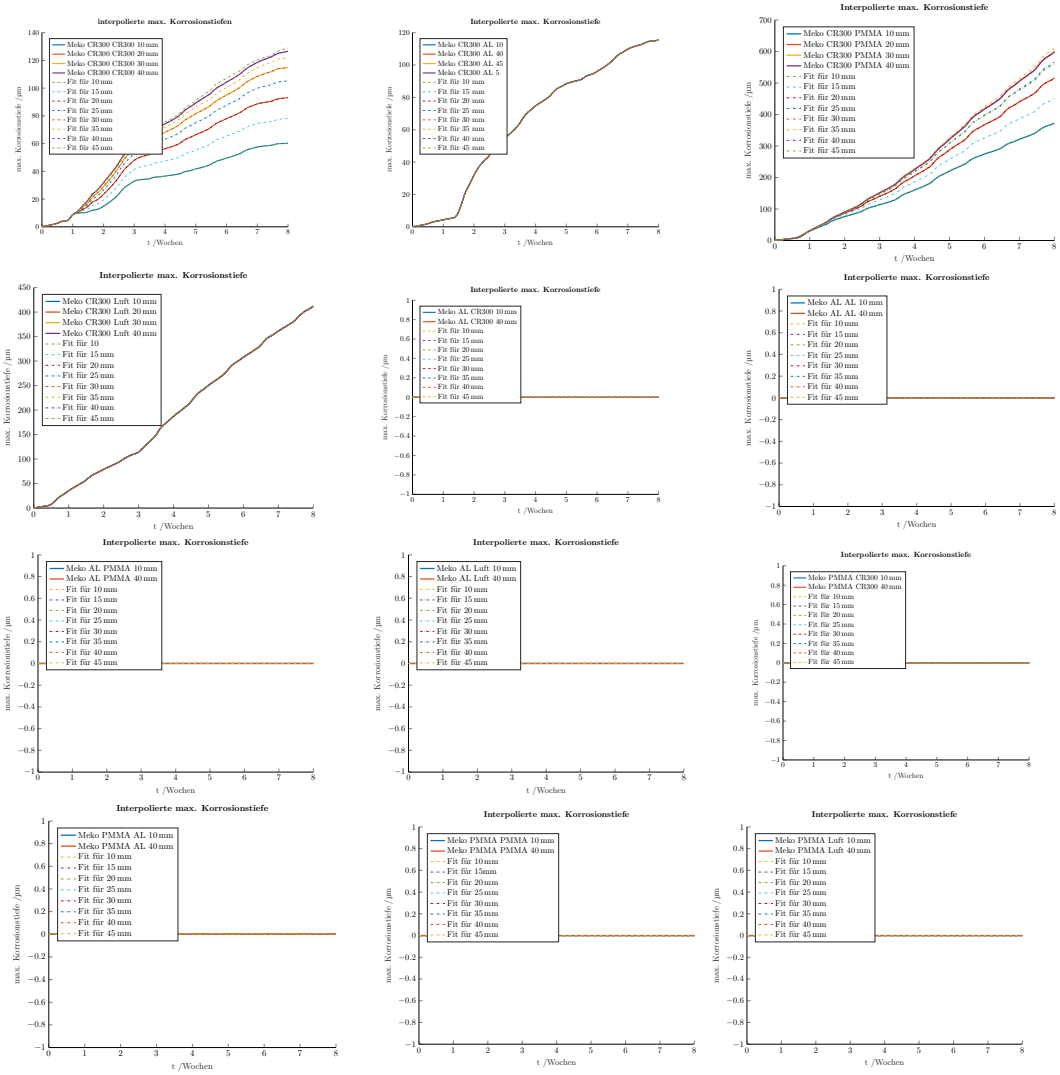
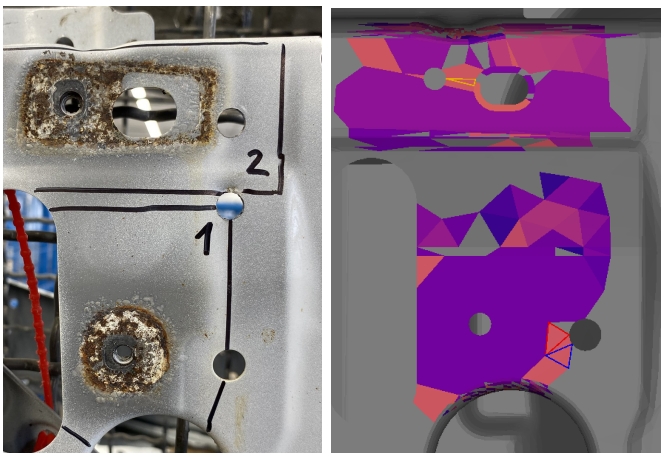


Abb. A.1: Die Daten, die zum Training des Netzwerks benutzt werden

B Evaluierte Stellen

- Stelle 1 und 2: Die Korrosionstiefen der Stellen 1 und 2 (Abbildung B.1) sind in Tabelle B.1 enthalten.



(a) Realverhalten der Stelle 1 und 2.

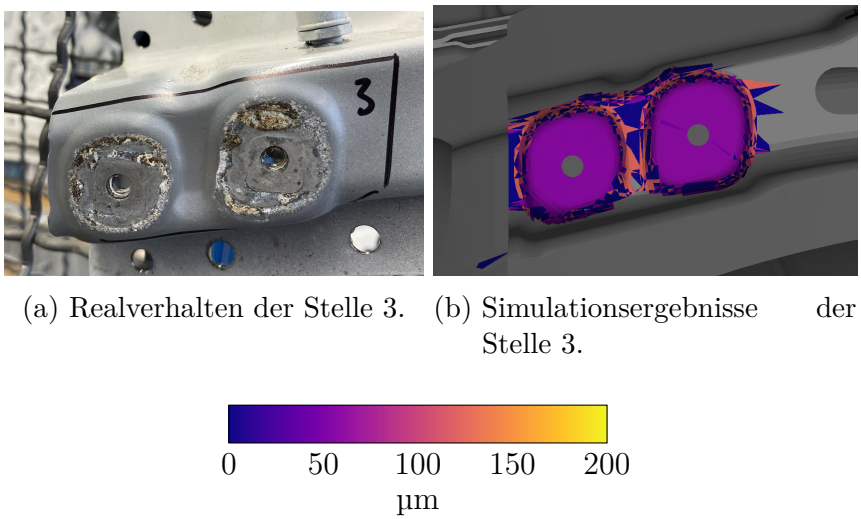
(b) Simulationsergebnisse der Stelle 1 und 2.

Abb. B.1: Flanschstellen 1 und 2 im Realtest und in der Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
1	154,9	125	109
2	110,9	109,3	108

Tabelle B.1: Prognose und Realergebnisse der Flanschstellen 1 und 2

- Stelle 3: Die Korrosionstiefen der Stelle 3 (Abbildung B.2) sind in Tabelle B.2 gezeigt.



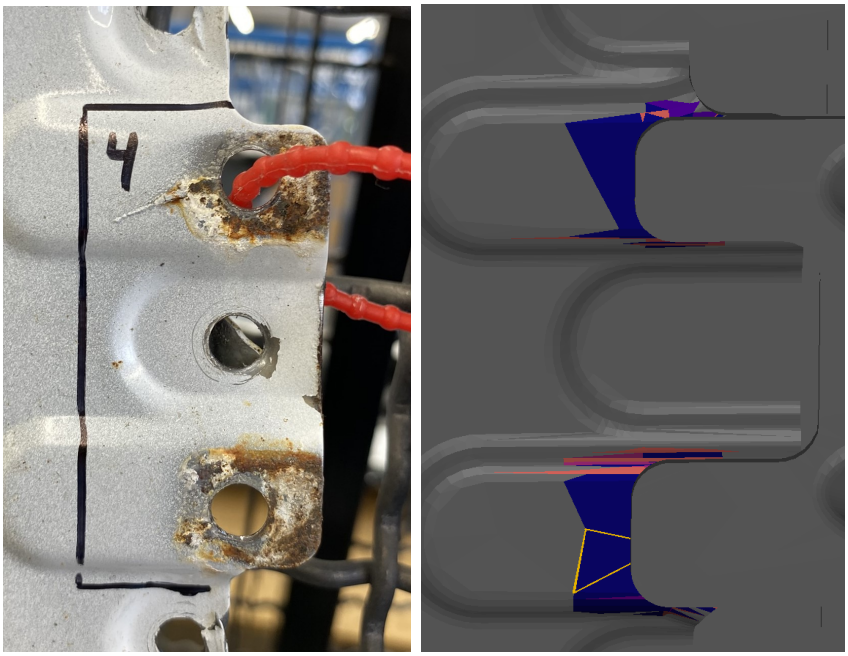
(a) Realverhalten der Stelle 3. (b) Simulationsergebnisse der Stelle 3.

Abb. B.2: Flanschstelle 3 im Realverhalten und als simuliertes Verhalten

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
3	41	50,8	51

Tabelle B.2: Prognose und Realergebnisse der Flanschstellen 3

- Flanschstelle 4: Die Korrosionstiefen der Stelle 4 (Abbildung B.3) sind in Tabelle B.3 enthalten.



(a) Realverhalten der Flanschstelle 4.

(b) Simulationsergebnisse der Flanschstelle 4.

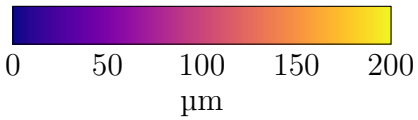
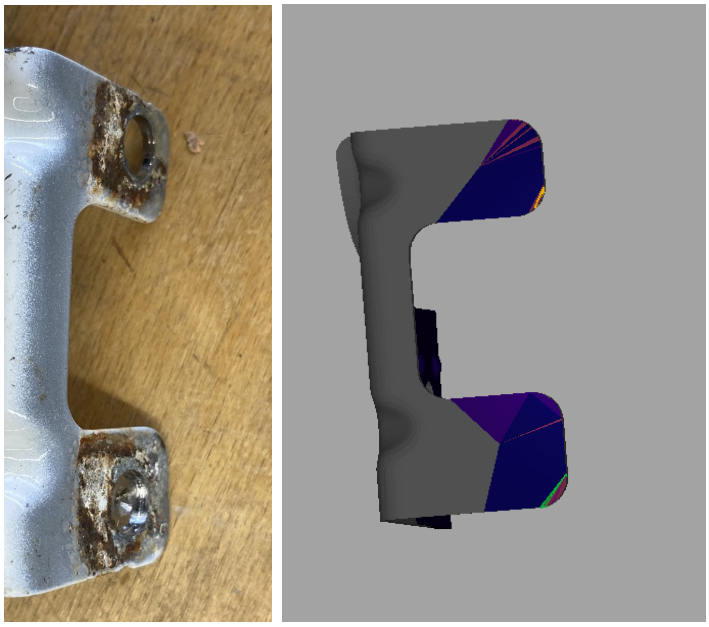


Abb. B.3: Flanschstelle 4 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
4	80,6	0	116

Tabelle B.3: Prognose und Realergebnisse der Flanschstelle 4

- Flanschstelle 5: Die Korrosionstiefen der Flanschstelle 5 (Abbildung B.4) sind in Tabelle B.4 enthalten.



(a) Realverhalten der Flanschstelle 5.

(b) Simulationsergebnisse der Flanschstelle 5.

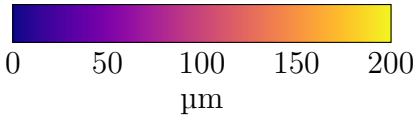
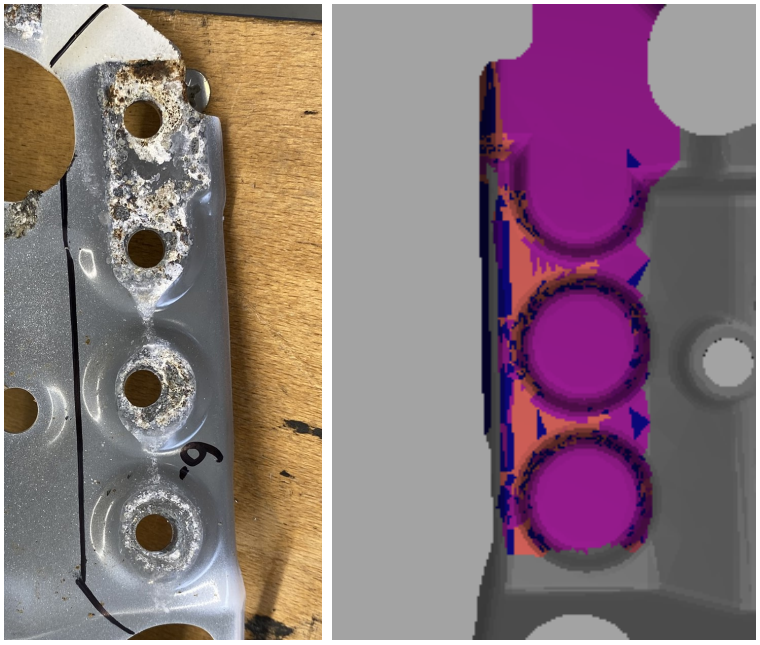


Abb. B.4: Flanschstelle 5 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
5	123,5	0	90

Tabelle B.4: Prognose und Realergebnisse der Flanschstelle 5

- Flanschstelle 6: Die Korrosionstiefen der Flanschstelle 6 (Abbildung B.5) sind in Tabelle B.5.



(a) Realverhalten der Flanschstelle 6.

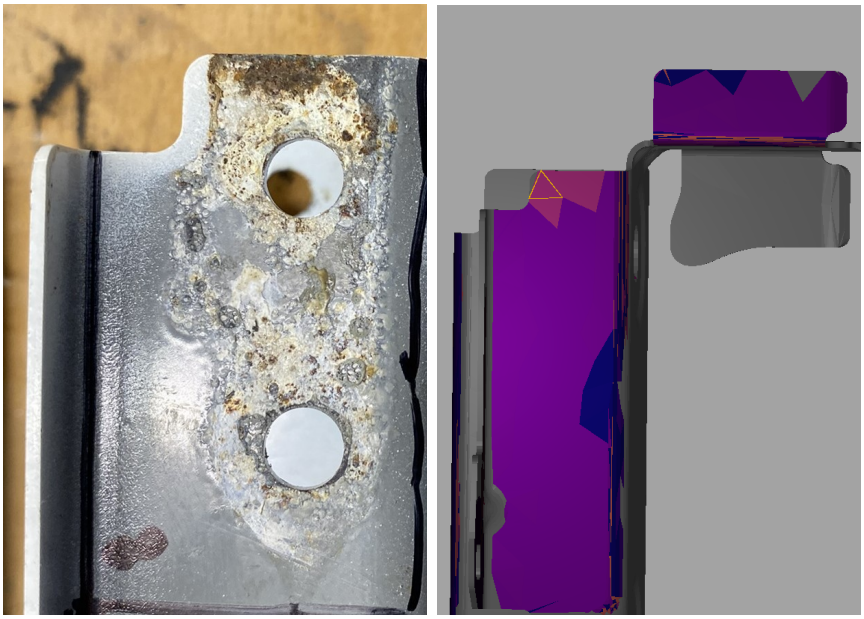
(b) Simulationsergebnisse der Flanschstelle 6.

Abb. B.5: Flanschstelle 6 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
6	65,5	23	70

Tabelle B.5: Prognose und Realergebnisse der Flanschstelle 6

- Flanschstelle 8: Die Korrosionstiefen der Flanschstelle 8 (Abbildung B.6) sind in Tabelle B.6.



(a) Realverhalten der Flanschstelle 8.

(b) Simulationsergebnisse der Flanschstelle 8.

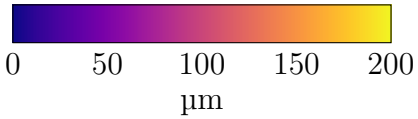
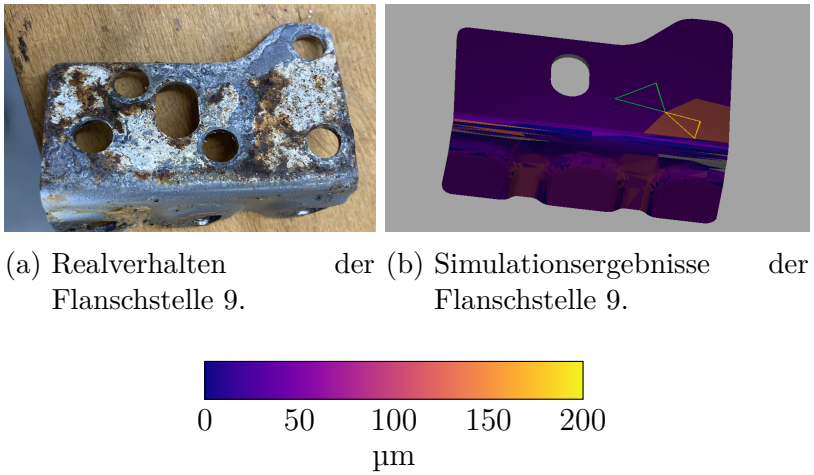


Abb. B.6: Flanschstelle 8 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
8	26,3	3,5	83

Tabelle B.6: Prognose und Realergebnisse der Flanschstelle 8

- Flanschstelle 9: Die Korrosionstiefen der Flanschstelle 9 (Abbildung B.7) sind in Tabelle B.7.



(a) Realverhalten der Flanschstelle 9.

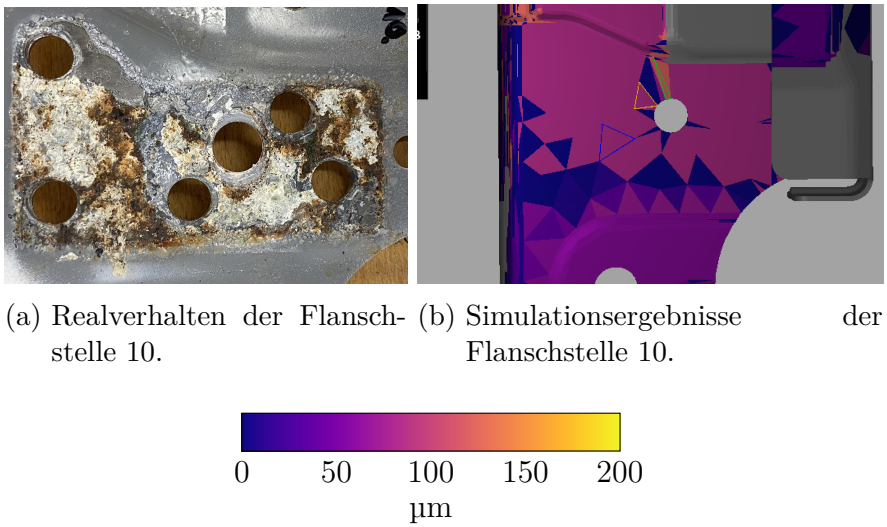
(b) Simulationsergebnisse der Flanschstelle 9.

Abb. B.7: Flanschstelle 9 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
9	76,1	29	116

Tabelle B.7: Prognose und Realergebnisse der Flanschstelle 9

- Flanschstelle 10: Die Korrosionstiefen der Flanschstelle 10 (Abbildung B.8) sind in Tabelle B.8.



(a) Realverhalten der Flanschstelle 10.

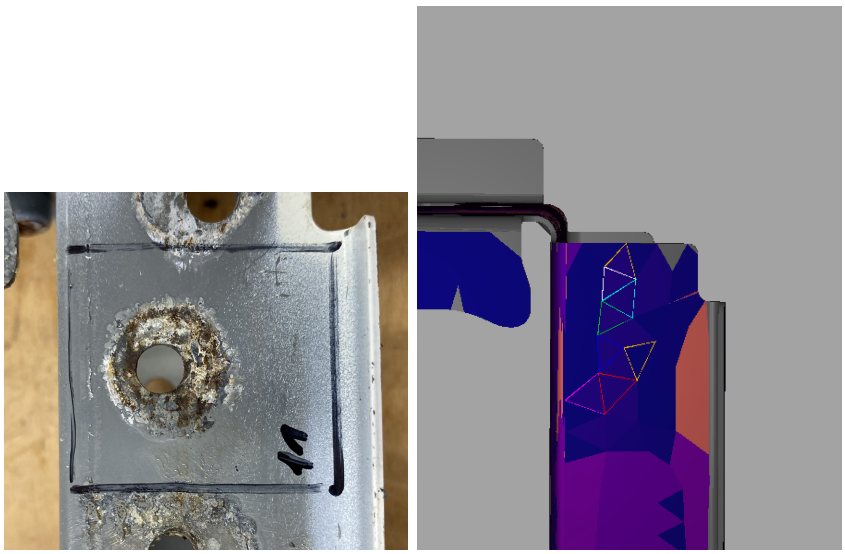
(b) Simulationsergebnisse der Flanschstelle 10.

Abb. B.8: Flanschstelle 10 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
10	76,1	29	116

Tabelle B.8: Prognose und Realergebnisse der Flanschstelle 10

- Flanschstelle 11: Die Korrosionstiefen der Flanschstelle 11 (Abbildung B.9) sind in Tabelle B.9.



(a) Realverhalten der Flanschstelle 11.

(b) Simulationsergebnisse der Flanschstelle 11.

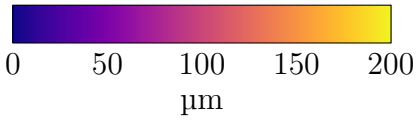
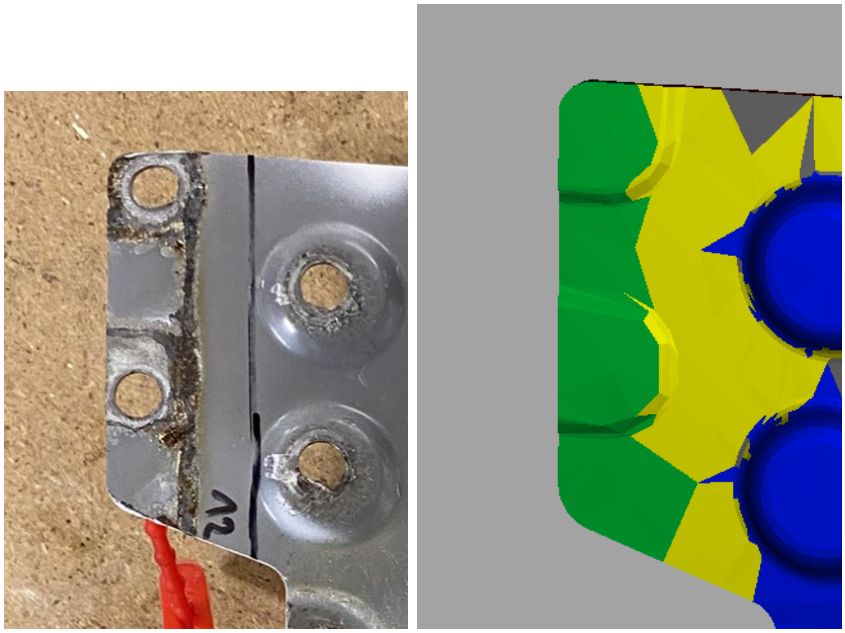


Abb. B.9: Flanschstelle 11 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
11	32,1	85,4	0

Tabelle B.9: Prognose und Realergebnisse der Flanschstelle 11

- Flanschstelle 12: Die Korrosionstiefen der Flanschstelle 12 (Abbildung B.10) sind in Tabelle B.10.



(a) Realverhalten der Flanschstelle 12.

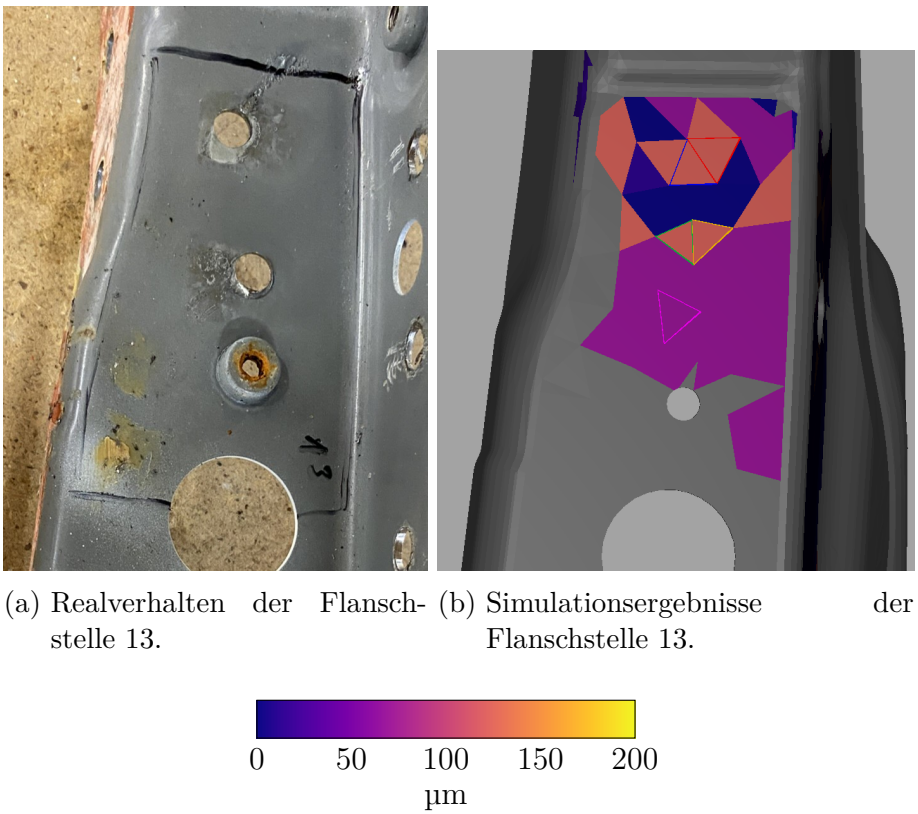
(b) Segmentierungsergebnisse der Flanschstelle 12 zeigen einen klebstoffgeschützen Bereich.

Abb. B.10: Flanschstelle 12 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
12	38,5	30,4	0

Tabelle B.10: Prognose und Realergebnisse der Flanschstelle 12

- Flanschstelle 13: Die Korrosionstiefen der Flanschstelle 13 (Abbildung B.11) sind in Tabelle B.11 gezeigt.



(a) Realverhalten der Flanschstelle 13.

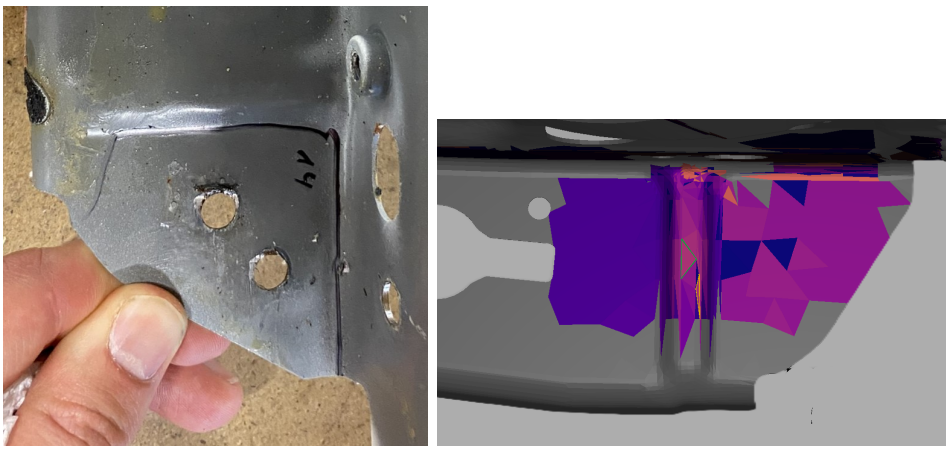
(b) Simulationsergebnisse der Flanschstelle 13.

Abb. B.11: Flanschstelle 13 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
13	0	85,3	86

Tabelle B.11: Prognose und Realergebnisse der Flanschstelle 13

- Flanschstelle 14: Die Korrosionstiefen der Flanschstelle 14 (Abbildung B.12) sind in Tabelle B.12.



(a) Realverhalten der Flanschstelle 14.

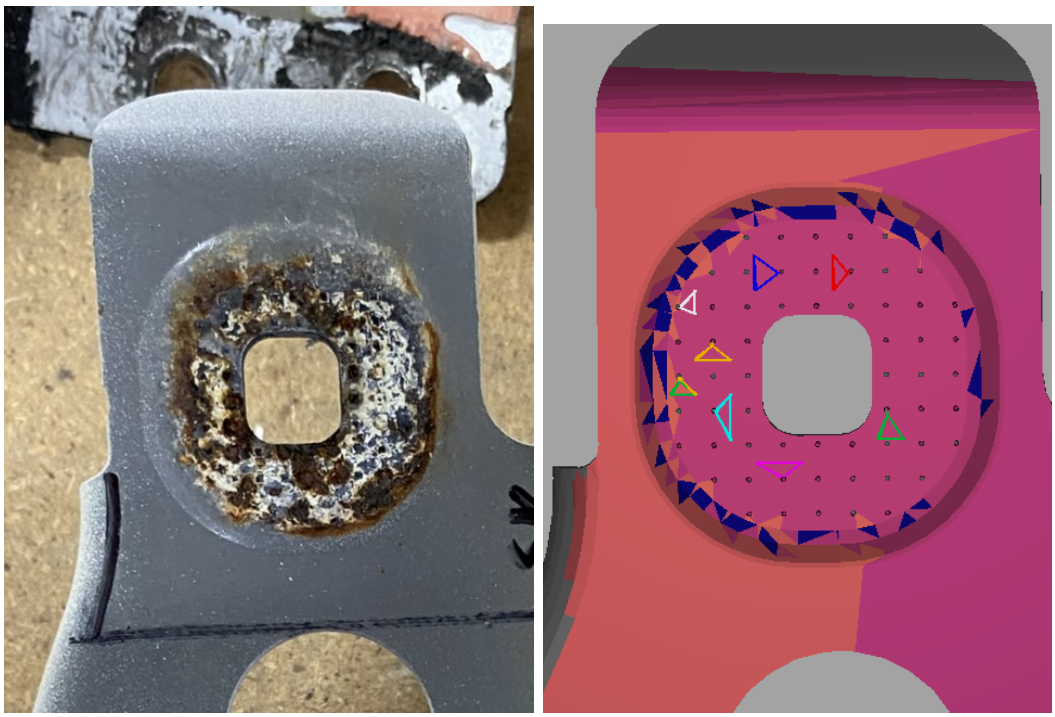
(b) Simulationsergebnisse der Flanschstelle 14.

Abb. B.12: Flanschstelle 14 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
14	0	55,7	116

Tabelle B.12: Prognose und Realergebnisse der Flanschstelle 14

- Flanschstelle 15: Die Korrosionstiefen der Flanschstelle 15 (Abbildung B.13) sind in Tabelle B.13.



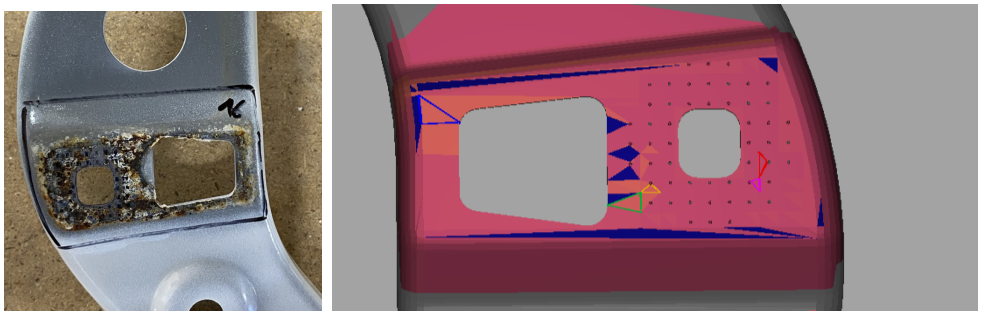
(a) Realverhalten der Flanschstelle 15. (b) Simulationsergebnisse der Flanschstelle 15.

Abb. B.13: Flanschstelle 15 im Realverhalten und Prognose

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
15	68,8	62,7	116

Tabelle B.13: Prognose und Realergebnisse der Flanschstelle 15

- Flanschstelle 16: Die Korrosionstiefen der Flanschstelle 16 (Abbildung B.14) sind in Tabelle B.14.



(a) Realverhalten der Flanschstelle 16.

(b) Simulationsergebnisse der Flanschstelle 16.

Abb. B.14: Flanschstelle 16 im Realverhalten und Prognose

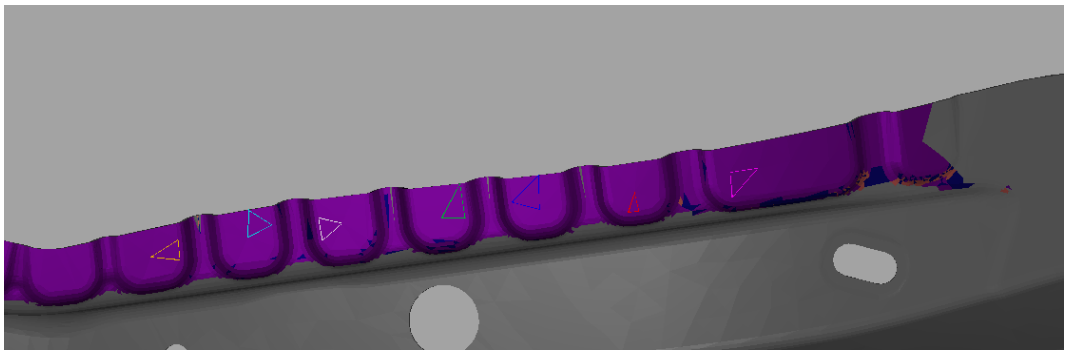
Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
16	52,7	124,9	116

Tabelle B.14: Prognose und Realergebnisse der Flanschstelle 16

- Flanschstelle 17: Die Korrosionstiefen der Flanschstelle 17 (Abbildung B.15) sind in Tabelle B.15.



(a) Realverhalten der Flanschstelle 17.



(b) Simulationsergebnisse der Flanschstelle 17.

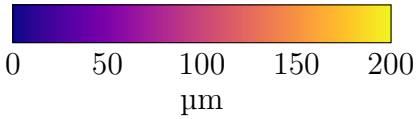


Abb. B.15: Flanschstelle 17 im Realverhalten und Prognose

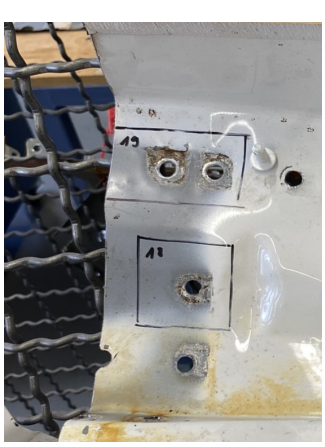
Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
17	106,4	22,8	51

Tabelle B.15: Prognose und Realergebnisse der Flanschstelle 17

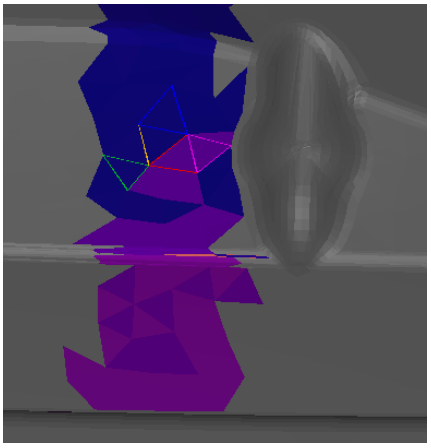
- Flanschstellen 18 und 19: Die Korrosionstiefen der Flanschstellen 18 und 19 (Abbildung B.16) sind in Tabelle B.16.

Flanschstelle	Ergebnis Realtest 1 \ µm	Ergebnis Realtest 2 \ µm	Ergebnis Simulation \ µm
18	75	–	35
19	110,6	45,4	53

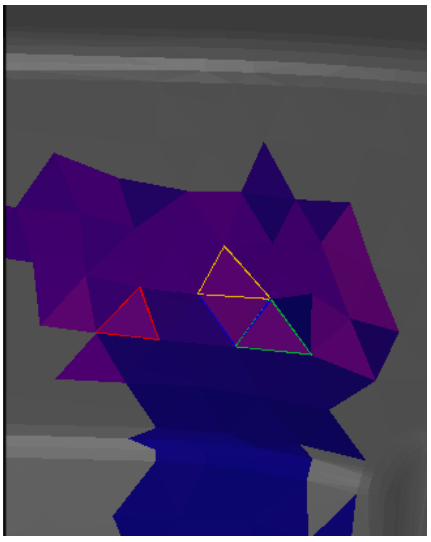
Tabelle B.16: Prognose und Realergebnisse der Flanschstellen 18 und 19



(a) Realverhalten der Flanschstelle 18 und 19



(b) Simulationsergebnisse der Flanschstelle 18



(c) Simulationsergebnisse der Flanschstelle 19

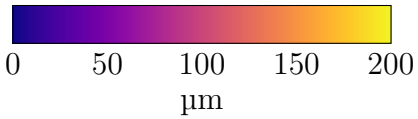


Abb. B.16: Flanschstellen 18 und 19 im Realverhalten und als simuliertes Verhalten

B.1 Hiflsberechnungen für Geometrien

Die Dreiecksfläche eines Dreiecks i im Raum wird wie folgt berechnet:

$$\begin{aligned} A_i &= \frac{|AB * AC|}{2} \\ &= \frac{|AB| * |AC| * |\sin\Theta|}{2} \\ &= \frac{1}{2}((y_{AB} * z_{AC} - z_{AB} * y_{AC})^2 + \\ &\quad (z_{AB} * x_{AC} - x_{AB} * z_{AC})^2) + \\ &\quad (x_{AB} * y_{AC} - y_{AB} * x_{AC})^2))^{\frac{1}{2}} \end{aligned} \tag{B.1}$$

C Literaturverzeichnis

[1] Westkämper, E. (2006). *Einführung in die Organisation der Produktion*. Springer.

[2] Jodlbauer, H. (2008). *Produktionsoptimierung*. Springer.

[3] Lotter, B. und Wiendahl, H.-P. (2012). *Montage in der industriellen Produktion*. Springer. Standardwerk zur industriellen Montage, behandelt Gestaltung, Planung und Optimierung moderner Montagesysteme.

[4] Kaesche, H. (2003). *Corrosion of Metals: Physicochemical Principles and Current Problems*. Engineering Materials and Processes. Springer.

[5] Koch, G., Brongers, M., Thompson, N., Virmani, Y. und Payer, J. (2001). Corrosion Cost and Preventive Strategies in the United States.

[6] Faerber, E. und Wobben, G.-D. (2015). *Einfluss der Korrosion auf die passive Sicherheit von Pkw*.

[7] Kapfer, K. (2022). *Untersuchung und Modellierung der Korrosion an Kanten und Flanschen unter unterschiedlichen Klimabedingungen und räumlichen Anordnungen*. Dissertation, Der Fakultät für Werkstoffwissenschaft und Werkstofftechnologie an der Technischen Universitaet Bergakademie Freiberg.

[8] Wünsche, M. (2020). *Numerische Korrosionssimulation von konstruktiven Schwachstellen für die industrielle Anwendung*. Dissertation, Technische Universitaet Dresden, Dresden.

[9] Giessgen, T. (2021). *Korrosionssimulation von pressgehärteten Stählen und Filiformfäden auf Aluminium*. Dissertation, Technischen Universitaet Hamburg-Harburg.

[10] Bösch, N.-C. (2016). *Numerische Simulation von Korrosionsprozessen für die industrielle Anwendung*. Dissertation, Technischen Universitaet Hamburg-Harburg, Hamburg.

[11] Waibel, L. (2021). *Segmentierung und Bewertung korrosionskritischer Konstruktionen einer Gesamtfahrzeugstruktur mittels Methoden der Künstlichen Intelligenz*. Dissertation, Fakultät für Mathematik und Wirtschaftswissenschaften Institut für Numerische Mathematik Universitaet Ulm.

[12] Waibel, L., Mittelbach, A. und Funken, S. (2021a). Semantic segmentation of corrosive critical designs in body-in-white structures for corrosion simulation. *Materials and Corrosion*, 72(5):788–804. doi:https://doi.org/10.1002/maco.202012134.

[13] Waibel, L., Kapfer, K., Hepfner, C., Mittelbach, A., Funken, S. und Krüger, L. (2021b). Edge delamination width prediction of 3D body-in-white part by finite element-based corrosion simulation and neural networks. *Materials and Corrosion*, 73:68–81. doi:10.1002/maco.202112637.

[14] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[15] Müller, A. und Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.

[16] Geron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA.

[17] Shapira, L., Shamir, A. und Cohen-Or, D. (2008). Consistent Mesh Partitioning and Skeletonisation Using the Shape Diameter Function. *Vis. Comput.*, 24(4):249–259. doi:10.1007/s00371-007-0197-5.

[18] Das, P. (2023). *Zeitreihen: Datenerzeugender Prozess*, Seiten 255–268. Springer Nature Singapore, Singapore. doi:10.1007/978-981-99-5940-2_9.

[19] Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77–89. doi:10.1016/S0034-4257(97)00083-7.

[20] Taha, A. A. und Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15:1–28.

[21] Powers, D. M. W. (2019). What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes.

[22] Matthews, B. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451. doi:https://doi.org/10.1016/0005-2795(75)90109-9.

[23] Draper, N. und Smith, H. (1998). *Applied regression analysis*. A Wiley-Interscience publication. Wiley, New York, NY [u.a.], 3. ed.. Auflage.

[24] Waszczyszyn, Z. (1999). Fundamentals of Artificial Neural Networks. In Waszczyszyn, Z., Herausgeber, *Neural Networks in the Analysis and Design of Structures*, Seiten 1–51, Vienna. Springer Vienna.

[25] Goodfellow, I., Bengio, Y. und Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[26] LEONTARITIS, I. J. und BILLINGS, S. A. (1985). Input-output parametric models for non-linear systems Part I: deterministic non-linear systems. *International Journal of Control*, 41(2):303–328. doi:10.1080/0020718508961129.

[27] Rumelhart, D. E. und McClelland, J. L. (1987). *Learning Internal Representations by Error Propagation*, Seiten 318–362. The MIT Press.

[28] Rumelhart, D. E., Hinton, G. E. und Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

[29] Liu, D. und Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528. doi:10.1007/BF01589116.

[30] Hagan, M. und Menhaj, M. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993. doi:10.1109/72.329697.

[31] Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533. doi:https://doi.org/10.1016/S0893-6080(05)80056-5.

[32] Johansson, E., Dowla, F. und Goodman, D. (1991). BACK-PROPAGATION LEARNING FOR MULTILAYER FEED-FORWARD NEURAL NETWORKS USING THE CONJUGATE GRADIENT METHOD. *International Journal of Neural Systems*, 02(04):291–301. doi:10.1142/S0129065791000261.

[33] Kingma, D. P. und Ba, J. (2017). Adam: A Method for Stochastic Optimization.

[34] Bergstra, J. und Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.*, 13(null):281–305.

[35] Trouvain, N., Pedrelli, L., Dinh, T. und Hinaut, X. (2020). *ReservoirPy: An Efficient and User-Friendly Library to Design Echo State Networks*, Seiten 494–505. Springer International Publishing. doi:10.1007/978-3-030-61616-8_40.

[36] Weihs, C., Luebke, K. und Czogiel, I. (2006). Response Surface Methodology for Optimizing Hyper Parameters. *Technical reports*.

[37] Nareyek, A. (2004). *Choosing Search Heuristics by Non-Stationary Reinforcement Learning*, Seiten 523–544. Springer US, Boston, MA. doi:10.1007/978-1-4757-4137-7-25.

[38] Holzinger, A. (2018). Explainable AI (ex-AI). *Informatik Spektrum*, 41(2):138–143. doi:10.1007/s00287-018-1102-5.

[39] van der Velden, B. H., Kuijf, H. J., Gilhuijs, K. G. und Viergever, M. A. (2022). Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470. doi:https://doi.org/10.1016/j.media.2022.102470.

[40] Lundberg, S. und Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. doi:10.48550/ARXIV.1705.07874.

[41] Ribeiro, M. T., Singh, S. und Guestrin, C. (2016). "Why Should I Trust You?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. doi:10.1145/2939672.2939778.

[42] Fryer, D., Strümke, I. und Nguyen, H. (2021). Shapley values for feature selection: The good, the bad, and the axioms. *Ieee Access*, 9:144352–144360.

[43] Lundberg, S. M. und Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. und Garnett, R., Herausgeber, *Advances in Neural Information Processing Systems 30*, Seiten 4765–4774. Curran Associates, Inc.

[44] Angelov, P., Soares, E., Kasegov, A. E. u a. (2021). Explainable artificial intelligence: an analytical review. *WIREs Data Mining and Knowledge Discovery*, 11(5):e1424. doi:10.1002/widm.1424.

[45] Xu, Y., Jie, Z., Ma, L., Jin, X. u a. (2020). Explainable Object-Induced Action Decision for Autonomous Vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 9523–9532. doi:10.1109/CVPR42600.2020.00954.

[46] Shah, N., Girdhar, R. und Song, S. (2022). JEDAI: A System for Skill-Aligned Explainable Robot Planning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Demo Track*.

[47] Hossfeld, M. und Wortmann, A. (2024). A Universal Framework for Skill-Based Cyber–Physical Production Systems. *Journal of Manufacturing and Materials Processing*, 8(5):221. doi:10.3390/jmmp8050221.

[48] Kale, A., Simmhan, Y. und Moreau, L. (2023). Provenance documentation to enable explainable and trustworthy AI: A literature review. *Data Intelligence*, 5(1):139–180. doi:10.1162/dint_a_00174.

[49] Bersani, M. M., Bucchiarone, A. und Taibi, D. (2023). Architecting Explainable Service Robots. In *European Conference on Software Architecture (ECSA 2023)*, Band 14212 in *Lecture Notes in Computer Science*, Seiten 153–169. Springer. doi:10.1007/978-3-031-43964-6_10.

[50] 8044, N. D. E. I. (2015). Korrosion von Metallen und Legierungen - Grundbegriffe. Technischer bericht, DIN Deutsches Institut für Normung e.V. doi:10.31030/2585719.

[51] Deshpande, K. B. (2010). Validated numerical modelling of galvanic corrosion for couples: Magnesium alloy (AE44)–mild steel and AE44–aluminium alloy (AA6063) in brine solution. *Corrosion Science*, 52(10):3514–3522. doi:https://doi.org/10.1016/j.corsci.2010.06.031.

[52] 2906, N. V. (1994). Schnittflächenqualität beim Schneiden, Beschneiden und Lochen von Werkstücken aus Metall. Technischer bericht, VDI-Gesellschaft Produktion und Logistik.

[53] Schneider, J. (2025). FEM-Simulation. unpublished.

[54] Ashcroft, W. (2019). *Industrial Polymer Applications: Essential Chemistry and Technology*. RSC.

[55] Habenicht, G. (2009). *Einteilung und Aufbau der Klebstoffe*, Seiten 3–14. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-540-85266-7.

[56] Revie, R. und Uhlig, H. (2008). *Cathodic Protection*, Seiten 251–267. John Wiley & Sons, Ltd. doi:10.1002/9780470277270.ch13.

[57] Larché, N., Vlot, M., Goodwin, F. und Thierry, D. (2009). Corrosion performance of Zn–Al–Mg coatings in open and confined zones in conditions simulating automotive applications. *Materials and Corrosion*, 61:412 – 420. doi:10.1002/maco.200905425.

[58] Fontana, M. (1986). *Corrosion Engineering*. Materials Science & Engineering. McGraw-Hill.

[59] Alavi, A. und Cottis, R. (1987). The determination of pH, potential and chloride concentration in corroding crevices on 304 stainless steel and 7475 aluminium alloy. *Corrosion Science*, 27(5):443–451. doi:https://doi.org/10.1016/0010-938X(87)90088-6.

[60] Gießgen, T. (2021). Korrosionssimulation von pressgehärteten Stählen und Filiformfäden auf Aluminium. doi:10.15480/882.3255.

[61] Protopopoff, E. und Marcus, P. (2003). Electrode Potentials. In *Corrosion: Fundamentals, Testing, and Protection*. ASM International. doi:10.31399/asm.hb.v13a.a0003579.

[62] Lindström, R., Svensson, J.-E. und Johansson, L.-G. (2002). The Influence of Salt Deposits on the Atmospheric Corrosion of Zinc. The Important Role of the Sodium Ion. *Journal of The Electrochemical Society*, 149:B57–B64. doi:10.1149/1.1432670.

[63] (2012). DIN EN ISO 9223:2012-05, Korrosion von Metallen und Legierungen - Korrosivität von Atmosphären - Klassifizierung, Bestimmung und Abschätzung.

[64] Pedeferri, P. (2018). *Corrosion Science and Engineering*. Springer Cham.

[65] Wickert, D. (2021). *Beitrag zur simulativen Bestimmung der Belastung von Fahrzeugkomponenten durch korrosive Medien am virtuellen Prototyp*, Band 19. Cuvillier Verlag.

[66] Steinmetz, M. und Mueller, E. (1993). On the capabilities and limits of smoothed particle hydrodynamics. *Astronomy and Astrophysics*, 268:391–410.

[67] Batchelor, G. K. (2000). *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library. Cambridge University Press.

[68] Stokes, G. G. (2009). *On the Theories of the Internal Friction of Fluids in Motion, and of the Equilibrium and Motion of Elastic Solids*, Seite 75–129. Cambridge Library Collection - Mathematics. Cambridge University Press.

[69] Streitberger, H.-J. und Dössel, K.-F., Herausgeber (2008). *Automotive Paints and Coatings*. Wiley-VCH, Weinheim, 2. Auflage. doi:10.1002/9783527622375.

[70] Doganay, E. (2021). *Prognose von Unterwanderungswerten (U/2) aus der Korrosionsprüfung VDA 233-102 basierend auf elektrochemischen und physikalischen Charakterisierungen des Korrosionsschutzsystems*. Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, Aachen. doi:10.18154/RWTH-2021-06240. Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2021.

[71] Feßmann, J. und Orth, H. (2002). *Angewandte Chemie und Umwelttechnik für Ingenieure: Handbuch für Studium und betriebliche Praxis*. ecomed Sicherheit. ecomed.

[72] Messier, R. (1994). Electrodeposition—the materials science of coatings and substrates. By Jack W. Dini, Noyes Publications, Park Ridge, NJ 1993, 367 pp., hardcover, US $ 78, ISBN 0-8155-1320-8. *Advanced Materials*, 6(1):88–89. doi:https://doi.org/10.1002/adma.19940060127.

[73] Wüstenberg, D. (2001). *Temporärer Korrosionsschutz von Halbzeugen, Bauteilen und Anlagen*, Kapitel 4.2, Seiten 1715–1756. John Wiley & Sons, Ltd. doi:https://doi.org/10.1002/9783527625659.ch4b.

[74] Harlfinger, R. (2001). *Straßenfahrzeuge*, Kapitel 5.15.1, Seiten 2975–3000. John Wiley & Sons, Ltd. doi:https://doi.org/10.1002/9783527625659.ch5p.

[75] Grimm, T. (2004). *User's guide to rapid prototyping*. Society of Manufacturing Engineers.

[76] Nagel RN, Braithwaite WW, K. P. (1980). Initial graphics exchange specification (IGES) version 1.0. NBSIR 80-1978 (R). Technischer bericht, National Bureau of Standards.

[77] Pratt, M. (2000). Introduction to ISO 10303 - The STEP Standard for Product Data Exchange.

[78] Rodrigues (1840). Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, Seiten 380–440.

[79] Möller, T. und Trumbore, B. (1997). Fast, Minimum Storage Ray-Triangle Intersection. *Journal of Graphics Tools*, 2(1):21–28. doi:10.1080/10867651.1997.10487468.

[80] Koecher, M. und Krieg, A. (2007). *Ebene Geometrie*. Springer-Lehrbuch. Springer, Berlin, Germany, 3. Auflage.

[81] Shellshear, E. und Ytterlid, R. (2014). Fast Distance Queries for Triangles, Lines, and Points using SSE Instructions. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):86–110.

[82] Sutherland, I. E. und Hodgman, G. W. (1974). Reentrant polygon clipping. *Commun. ACM*, 17(1):32–42. doi:10.1145/360767.360802.

[83] and, B. B. (1986). The Surveyor's Area Formula. *The College Mathematics Journal*, 17(4):326–337. doi:10.1080/07468342.1986.11972974.

[84] MatLab (2010). patternnet - Generate pattern recognition network. Technischer bericht, MatLab.

[85] Coelho, L. B., Zhang, D., Ingelgem, Y. V., Steckelmacher, D., Nowé, A. und Terryn, H. (2022). Reviewing machine learning of corrosion prediction in a data-oriented perspective. *npj Materials Degradation*, 6(1). doi:10.1038/s41529-022-00218-4.

[86] Kordijazi, A., Zhao, T., Zhang, J., Alrfou, K. und Rohatgi, P. (2021). A Review of Application of Machine Learning in Design, Synthesis, and Characterization of Metal Matrix Composites: Current Status and Emerging Applications. *JOM*, 73(7):2060–2074. doi:10.1007/s11837-021-04701-2.

[87] Weinmann, M., Jutzi, B., Hinz, S. und Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS journal of photogrammetry and remote sensing*, 105:286–304. doi:10.1016/j.isprsjprs.2015.01.016.

[88] Blomley, R., Jutzi, B. und Weinmann, M. (2016). CLASSIFICATION OF AIRBORNE LASER SCANNING DATA USING GEOMETRIC MULTI-SCALE FEATURES AND DIFFERENT NEIGHBOURHOOD TYPES.

ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, III-3:169–176. doi:10.5194/isprs-annals-III-3-169-2016.

[89] West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M. und Iverson, A. E. (2004). Context-driven automated target detection in 3D data. In Sadjadi, F. A., Herausgeber, *Automatic Target Recognition XIV*, Band 5426, Seiten 133 – 143. International Society for Optics and Photonics, SPIE. doi:10.1117/12.542536.

[90] Wickert, D. (2017). Aufbau und Validierung eines numerischen Strömungs-modells für eine Fahrzeugtürdurchströmung mittels SPH-Methode.

[91] Sven (2024). octree - partitioning 3D points into spatial subvolumes. MATLAB Central File Exchange.

[92] Krei, D. (2024). Weiterentwicklung und Parametrisierung einer Validierungsmethodik für eine partikelbasierte CFD-Simulation. Masterarbeit, Universität Stuttgart, Institut für Fahrzeugtechnik Stuttgart, Stuttgart. Unveröffentlichte Masterarbeit, betreut von Dr.-Ing. Konstantin Kapfer (Mercedes-Benz AG) und Dr.-Ing. Timo Kuthada (IFS Stuttgart).

[93] Jacobson, A. u a. (2024). gptoolbox: Geometry Processing Toolbox. http://github.com/alecjacobson/gptoolbox.

[94] Cormen, T., Leiserson, C., Rivest, R. und Stein, C. (2009). *Introduction to Algorithms, third edition*. Computer science. MIT Press.

[95] 9227, N. D. E. I. (2017). Korrosionsprüfungen in künstlichen Atmosphären - Salzsprühnebelprüfungen. Technischer bericht, DIN Deutsches Instiut für Normung e.V. doi:10.31030/2585719.

[Gollé-Leidreiter et al.] Gollé-Leidreiter, M., Schneider, J., Kapfer, K., Mittelbach, A., Mandel, M., Krüger, L. und Kowalski, J. Corrosion Prediction Approach Combining FEM-Simulations and AI-Modules for Complex Automotive Geometries. *Materials and Corrosion*, n/a(n/a). doi:https://doi.org/10.1002/maco.70127.

[97] MathWorks (2025). Design Time Series NARX Feedback Neural Networks. `https://de.mathworks.com/help/deeplearning/ug/design-time-series-narx-feedback-neural-networks.html`. Accessed: 2025-06-13.

Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung eines KI-basierten Systems zur automatischen, simulationsgestützten Vorhersage des Korrosionsverhaltens an Gesamtfahrzeugen. Dabei werden konstruktive Maßnahmen, Elektrolytbeanspruchung und spezifische Bauteileigenschaften berücksichtigt. Das System adressiert eine wesentliche Forschungslücke in der Automobilentwicklung, da bisher keine Methode existierte, die eine ganzheitliche Vorhersage des Korrosionsverhaltens eines Gesamtfahrzeugs unter realitätsnahen Bedingungen erlaubt.
Die Methodik umfasst drei Kernkomponenten, die Klassifizierung korrosionsgeschützter Bereiche, die Auswertung der Elektrolytbeanspruchung und die KI-basierte Korrosionsprognose.
Für die Klassifizierung der korrosionsvermeidenden Maßnahmen (Klebstoff, Nahtabdichtung, Wachs) wurden Machine-Learning-Modelle mit Transfer-Learning entwickelt. Diese erreichen hohe Genauigkeiten mit einem Matthews Correlation Coefficient (MCC) von bis zu 0,96. Durch die Integration von Shapley-Werten wurde die Erklärbarkeit verbessert und die Trainingszeit eines Flanschnetzwerks um rund 18 % reduziert, ohne signifikanten Genauigkeitsverlust.
Zur Beschreibung der Elektrolytbeanspruchung wurde ein Algorithmus entwickelt, der die Ergebnisse partikelbasierter Strömungssimulationen auf die Fahrzeugoberfläche überträgt. Das resultierende System ermöglicht eine ortsaufgelöste Bewertung der Elektrolyteinwirkung auf einzelne Oberflächenelemente.
Die anschließende Korrosionsprognose basiert auf simulierten und gemessenen Korrosionsverläufen und trainiert ein NARX-Modell. Der Abgleich mit realen Fahrzeugtests ergibt eine mittlere Abweichung von 57 µm, die sich nach Korrektur produktionsbedingter Einflüsse auf 17 µm reduziert. Damit wird eine sehr hohe Übereinstimmung zwischen Simulation und Realität erreicht. Auch die Erweiterung der Kantenprognose um Elektrolyteinflüsse zeigte konsistente Ergebnisse.
Das auf Basis dieser Methodiken implementierte Gesamtsystem arbeitet modular und weist eine Laufzeit von rund einem Tag pro Gesamtfahrzeug auf. Es ist damit praxistauglich für den industriellen Entwicklungsprozess und ermöglicht eine reproduzierbare, nachvollziehbare Bewertung korrosionskritischer Bereiche. Die Arbeit leistet somit einen wesentlichen Beitrag zur Digitalisierung und Automatisierung der Korrosionsbewertung im Automobilbau, indem sie simulative, datenbasierte und erklärbare Methoden zu einem integrierten digitalen Zwilling vereint.

Abstract

The aim of this work is the development of an AI-based system for the automated, simulation-supported prediction of corrosion behavior in complete vehicles. The system considers design measures, electrolyte exposure, and specific component properties. It addresses a major research gap in automotive development, as no existing method has yet enabled a holistic prediction of a vehicle's corrosion behavior under realistic conditions.
The methodology consists of three core components: the classification of corrosion protection measures, the evaluation of electrolyte exposure, and the AI-based corrosion prediction.
For the classification of corrosion protection measures (adhesive, PVC, wax), machine-learning models with transfer learning were developed. These achieve high accuracies with a Matthews Correlation Coefficient (MCC) of up to 0.96. By integrating Shapley values, model explainability was improved and the training time of a flange network was reduced by around 18
To characterize the electrolyte exposure, an algorithm was developed that maps the results of particle-based flow simulations onto the vehicle surface. The resulting system enables a spatially resolved assessment of the electrolyte impact on individual surface elements.
The subsequent corrosion prediction is based on simulated and measured corrosion progressions and uses a NARX model for training. Comparison with real vehicle tests shows a mean deviation of 57 µm, which is reduced to 17 µm after correcting for production-related influences. This demonstrates a very high level of agreement between simulation and reality. The extension of the edge prediction to include electrolyte influences also showed consistent results.
The overall system implemented based on these methods operates modularly and has a runtime of approximately one day per complete vehicle. It is thus suitable for practical use in industrial development processes and enables a reproducible, transparent assessment of corrosion-critical areas.
This work therefore makes a significant contribution to the digitalization and automation of corrosion assessment in automotive engineering by integrating simulation-based, data-driven, and explainable methods into a unified digital twin.