

# Datenbasierte, prozessparallele Qualitätsprognose für spanend hergestellte Werkstücke mittels maschinellen Lernens

Vom Fachbereich Maschinenbau  
an der Technischen Universität Darmstadt  
zur  
Erlangung des Grades eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte

DISSERTATION

vorgelegt von

**Alexander Joachim Fertig, M. Sc.**

aus Miltenberg

Berichterstatter: Prof. Dr.-Ing. Matthias Weigold  
Mitberichterstatter: Prof. Dr.-Ing. Thomas Bergs MBA  
Tag der Einreichung: 26. Januar 2023  
Tag der mündlichen Prüfung: 09. Mai 2023

Fertig, Alexander Joachim: Datenbasierte, prozessparallele Qualitätsprognose für spanend hergestellte Werkstücke mittels maschinellen Lernens  
Darmstadt, Technische Universität Darmstadt

Jahr der Veröffentlichung der Dissertation auf TUprints: 2023

URN: urn:nbn:de:tuda-tuprints-241127

Tag der mündlichen Prüfung: 09.05.2023



Veröffentlicht unter CC BY-SA 4.0 International

<https://creativecommons.org/licenses/>

# Vorwort des Herausgebers

Durch die zunehmende Vernetzung und Digitalisierung der Fertigungsindustrie stellt jede Komponente in einer Werkzeugmaschine eine potenzielle Datenquelle dar, welche mit einer konsequenten Erfassung und Verarbeitung neue Ansätze und Anwendungen für datengetriebene Produktionssysteme ermöglicht. Eine zentrale Anwendung, die dadurch ermöglicht wird, ist der Einsatz von künstlicher Intelligenz (KI). Die Datenanalyse mittels KI ist eines der dominierenden Themen in der derzeitigen Produktionsforschung.

Die Qualitätsmessung zerspanter Bauteile ist in der Fertigungspraxis mit sehr hohen zeitlichen und kostenintensiven Aufwänden verbunden, da sie heute durch personalintensive Prüfungen von Maschinenbedienenden an den Werkzeugmaschinen, als auch in separaten Arbeitsvorgängen, auf Koordinatenmessgeräten durchgeführt werden. Während solche Schritte zur Qualitätssicherung einen hohen Aufwand erfordern, könnte mit dem Einsatz von KI ein Wendepunkt eintreten, der es Systemen ermöglicht, solch komplexe Aufgaben zu erlernen und während der Fertigungszeit automatisiert durchzuführen.

Das Ziel der vorliegenden Dissertation ist die Entwicklung eines Lösungskonzepts für datenbasierte Qualitätsprognosen von spanend hergestellten Werkstücken. Als Grundlage dienen Daten aus der numerischen Steuerung von Werkzeugmaschinen. Entwickelte Lösungen für die Kontextualisierung und Segmentierung ordnen die qualitätsrelevanten Datenpunkte den entsprechenden Prozessabschnitten am Bauteil zu und verknüpfen diese mit den zugehörigen Daten aus der Qualitätssicherung. Eine speziell konzipierte Prozesskette für das maschinelle Lernen (MLPK) übernimmt die Erstellung der prozessindividuellen Prognosemodelle. Die automatisierte Parametrierung der MLPK durch einen entwickelten Algorithmus ermöglicht die effiziente Erstellung optimierter Qualitätsprognosemodelle für individuelle Werkstücke. Das erarbeitete Gesamtkonzept wird auf Basis real gewonnener Daten unter produktionsnahen Bedingungen anhand von zwei Anwendungsfällen erprobt und die Realisierbarkeit eines darauf aufbauenden prozessparallelen Qualitätsüberwachungssystems mithilfe eines eigens entwickelten Softwareprototyps belegt.

Das entwickelte Vorgehen eröffnet neue Möglichkeiten für die Entwicklung und Umsetzung prozessparalleler Qualitätsprognosen und unterstützt Experten der Produktion dabei, das eigene Fachwissen zu nutzen, um unter Verwendung der vorgestellten Lösungen KI-basierte Prognosemodelle zu erstellen und in ein Qualitätsüberwachungssystem zu überführen.

Darmstadt, im Mai 2023

Prof. Dr.-Ing. Matthias Weigold



## Vorwort des Verfassers

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Produktionsmanagement, Technologie und Werkzeugmaschinen (PTW) der Technischen Universität Darmstadt.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Matthias Weigold für die wissenschaftliche Betreuung der Arbeit und umfassende Unterstützung. Die hervorragenden Rahmenbedingungen für die Forschungstätigkeiten und gewährten Freiheiten habe ich sehr geschätzt. Des Weiteren bedanke ich mich bei Herrn Prof. Dr.-Ing. Thomas Bergs MBA für die Übernahme des Korreferats.

Die Inhalte dieser Arbeit entstanden im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekts TensorMill. Allen beteiligten Partnern gilt mein Dank für die fachlichen Diskussionen und die Unterstützung.

Herzlicher Dank geht an meine Kolleginnen und Kollegen, insbesondere aus der Forschungsgruppe TEC und der ehem. Zerspanungstechnologie, die meine Zeit am Institut zu etwas ganz Besonderem gemacht haben. Ich bin dankbar für die vielen neuen Freundschaften und gemeinsamen Erinnerungen.

Meinem Bürokollegen Oliver Kohn danke ich besonders für seine Hilfsbereitschaft und umfassende fachliche sowie freundschaftliche Unterstützung. Deine hervorragende Arbeit im Bereich der Maschinenvernetzung und Datenakquise hat entscheidend zum Erfolg dieser Arbeit beigetragen. Mein weiterer Dank gilt Willi Wünschel, Dr.-Ing. Felix Geßner und Christopher Krebs für die kritische Durchsicht meiner Arbeit und das konstruktive Feedback. Weiterhin möchte ich mich bei Christoph Preis, Yujia Chen, Patrick Ostendarp, Bastian Schrimpf und Robert Bruns für die Zusammenarbeit danken.

Ein großer Dank geht an meine Freunde für die schönen und unvergesslichen Momente außerhalb der beruflichen Tätigkeit.

Meiner Familie, insbesondere meinen Eltern Yenny und Hans-Joachim und meinem Bruder Jürgen möchte ich von ganzem Herzen danken. Eure bedingungslose Unterstützung hat mir das Studium und damit auch die Promotion ermöglicht und mir in jeder Lebensphase den notwendigen Rückhalt gegeben.

Mein größter Dank gilt meiner Freundin Vanessa. Deine Liebe, Fürsorge und Loyalität haben mich höchst motiviert und mir während der Erstellung der Dissertation stets Halt gegeben. Ohne dich wäre das nicht möglich gewesen. Ich danke dir für deine Unterstützung und für alles, was du für mich bist!



# Inhaltsverzeichnis

<b>Formelzeichen und Abkürzungsverzeichnis</b>	<b>ix</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Stand des Wissens</b>	<b>5</b>
2.1 KI und deren Relevanz für Produktionsunternehmen . . . . .	5
2.1.1 Begriffserklärung: KI . . . . .	5
2.1.2 ML und das Potenzial von KI im Produktionsumfeld . . . . .	9
2.1.3 KI im Kontext der Qualitätssicherung . . . . .	12
2.2 Prozessüberwachung an Werkzeugmaschinen . . . . .	13
2.2.1 Spanende Werkzeugmaschinen . . . . .	13
2.2.2 Numerische Steuerungen . . . . .	16
2.2.3 Maschineninterne NC-Signaldaten . . . . .	18
2.2.4 Prozessüberwachung in der spanenden Fertigung . . . . .	20
2.2.5 Verfahren zur Signalverarbeitung und Merkmalsextraktion . . . . .	24
2.2.6 Aktuelle Ansätze und industrielle Lösungen zur Prozessüberwachung . . . . .	27
2.3 Werkstückqualität in der spanenden Fertigung . . . . .	29
2.3.1 Werkstückqualität und -prüfung . . . . .	29
2.3.2 Einflüsse auf die Werkstückqualität . . . . .	31
2.4 Datenbasierte Bewertung der Werkstückgestalt . . . . .	34
2.5 Zwischenfazit und Forschungsbedarf . . . . .	38
<b>3 Zielsetzung und Vorgehensweise</b>	<b>41</b>
<b>4 Entwicklungsrahmen für die datenbasierte Qualitätsüberwachung</b>	<b>45</b>
4.1 Versuchsumgebung und Referenzprozess . . . . .	45
4.1.1 Versuchsmaschinen . . . . .	45
4.1.2 Konzeption des Referenzprozesses . . . . .	47
4.2 Datenerfassung entlang der Wertschöpfungskette . . . . .	51
4.2.1 Datenerfassung mittels Edge-Computing . . . . .	51
4.2.2 Hochfrequente NC-Signaldaten (hf_data) . . . . .	55
4.2.3 NC-Variablendaten (lf_data) . . . . .	59
4.2.4 Qualitätsdaten . . . . .	60

<b>5</b>	<b>Lösungsgrundlagen</b>	<b>65</b>
5.1	Kontextualisierung der NC-Signaldaten . . . . .	65
5.2	Automatisierte Datensegmentierung . . . . .	70
5.3	Prozesskette für das maschinelle Lernen . . . . .	75
5.3.1	Problembeschreibung im Kontext der Qualitätsüberwachung . . . . .	75
5.3.2	Merkmalsextraktion . . . . .	78
5.3.3	Merkmalsselektion . . . . .	83
5.3.4	Algorithmenauswahl und -optimierung . . . . .	90
5.3.5	Evaluation der Prognosegenauigkeit . . . . .	99
5.3.6	Softwaretechnische Umsetzung der Prozesskette . . . . .	102
5.4	Datengenerierung . . . . .	107
<b>6</b>	<b>Umsetzung der prozessparallelen Qualitätsprognose</b>	<b>113</b>
6.1	Datenvorbereitung und Signalanalyse . . . . .	113
6.1.1	Segmentierungsalgorithmus und Datenqualität . . . . .	113
6.1.2	Verteilung der Werkstückqualität . . . . .	116
6.1.3	Signalauswahl und -reduktion . . . . .	118
6.2	Umsetzung der automatisierten Prognosemodellerstellung . . . . .	123
6.3	Algorithmus zur automatisierten Prozesskettenparametrierung . . . . .	127
6.3.1	Algorithmusbeschreibung . . . . .	128
6.3.2	Optimierung der Teilprozesse . . . . .	131
6.3.3	Bewertungsmethode . . . . .	132
6.3.4	Validierung des Optimierungsalgorithmus . . . . .	134
<b>7</b>	<b>Validierung und Realisierung</b>	<b>139</b>
7.1	Validierung des Gesamtkonzepts . . . . .	139
7.2	Implementierung eines Softwareprototyps . . . . .	143
7.3	Ergebnisdiskussion . . . . .	146
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>149</b>
8.1	Zusammenfassung . . . . .	149
8.2	Ausblick . . . . .	150
	<b>Literaturverzeichnis</b>	<b>153</b>
<b>A</b>	<b>Anhang</b>	<b>183</b>

# Formelzeichen und Abkürzungsverzeichnis

## Formelzeichen

Zeichen	Einheit	Bezeichnung
$a_p$	mm	Schnitttiefe
$id_{\text{wkz}}$	-	aktueller Werkzeugbezeichner
$l_{\text{wkz}}$	mm	Werkzeuglänge
$N_{\text{fs}}$	-	Anzahl der genutzten Merkmalsselektionsalgorithmen
$p_x, p_y, p_z$	mm	Positionskordinaten
$r_{\text{wkz}}$	mm	Werkzeugradius
$SW_{\text{fs,comm}}$	-	Schwellenwert zur Überschneidung bei gemeinsamer Merkmalsselektion
$SW_{\text{fs,prop}}$	-	Schwellenwert zur Überschneidung bei proprietärer Merkmalsselektion
$t_{\text{Ipo}}$	ms	Interpolationstakt
$MI$	-	Transinformation zwischen zwei Variablen (engl.: Mutual Information)
$t_{\text{Lage}}$	ms	Lageregeltakt
$VB$	$\mu\text{m}$	Verschleißmarkenbreite
$v_c$	m/min	Schnittgeschwindigkeit
$v_f$	mm/min	Vorschubgeschwindigkeit
$x_{\text{mks}}, y_{\text{mks}}, z_{\text{mks}}$	mm	Positionskordinaten im Maschinenkoordinatensystem
$x_{\text{npv}}, y_{\text{npv}}, z_{\text{npv}}$	mm	Translation der Nullpunktverschiebung
$x_{\text{trans}}, y_{\text{trans}}, z_{\text{trans}}$	mm	Rücktransformierte Positionskordinaten nach Translation
$x_{\text{wks}}, y_{\text{wks}}, z_{\text{wks}}$	mm	Positionskordinaten im Werkstückkoordinatensystem
$\varphi_{\text{npv}}, \theta_{\text{npv}}, \psi_{\text{npv}}$	°	Drehung der Nullpunktverschiebung

## Abkürzungen

ACC	Accuracy
AdaBoost	AdaBoost-Classifer
AE	Körperschallemission (engl.: Acoustic Emission)
AUC	Flächeninhalt unter der Kurve (engl.: Area under Curve)
BTSS	Bedienertafelschnittstelle
CAD	Computer Aided Design (dt.: rechnerunterstütztes Kontruieren)
CAM	Computer Aided Manufacturing (dt.: rechnerunterstützte Fertigung)
CommFeatSet	gemeinsamer Merkmalsatz (engl.: Common Feature Set)
CSV	Comma-separated values (Dateiformat)
CWT	kontinuierliche Wavelet-Transformation (engl.: Continuous Wavelet-transform)
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz
DL	Deep Learning
DS	Datensatz
DT	Decision Tree
DWT	diskrete Wavelet-Transformation (engl.: Discrete Wavelet Transform)
EC	Fehlereinfluss (engl.: Error Code)
EMD	empirische Modenzerlegung (engl.: Empirical Mode Decomposition)
EuPa	Europäisches Parlament
FEM	Finite-Element-Methode
FFT	Fast-Fourier-Transformation
FN	Falsch-Negativ (engl.: False Negative)
FP	Falsch-Positiv (engl.: False Positive)
FPR	Falsch-Positiv-Rate (engl.: False Positive Rate)

---

GNB	Gaussian Naive Bayes
hf_data	paketweise erfasste hochfrequente NC-Signaldaten (Siemens Originalbezeichnung: <i>sinumerik_hf_data</i> )
HHT	Hilbert-Huang-Transformation
i. O.	„in Ordnung“
IoT	Internet der Dinge (engl.: Internet of Things)
IA	Augenblicksamplitude (engl.: Instantaneous Amplitude)
IF	Augenblicksfrequenz (engl.: Instantaneous Frequency)
IG	Informationsgehalt (engl.: Information Gain)
IPC	Industrie-PC
IMF	intrinsische Modenfunktion (engl.: Intrinsic Mode Function)
JSON	JavaScript Object Notation
KI	künstliche Intelligenz
KMG	Koordinatenmessgerät
KMT	Koordinatenmesstechnik
KNN	<i>k</i> -Nearest Neighbours
lf_data	paketweise erfasste niederfrequente NC-Variablendaten (Siemens Originalbezeichnung: <i>sinumerik_lf_data</i> )
MCA	MachiningContextAdapter - am PTW entwickelte Applikation zur Datenerfassung
MKS	Maschinenkoordinatensystem
ML	maschinelles Lernen (engl.: Machine Learning)
MLP	Multilayer-Perceptron
MLPK	Prozesskette für das maschinelle Lernen
MQTT	Message Queuing Telemetry Transport
n. i. O.	„nicht in Ordnung“
NC	Numerische Steuerung (engl.: Numerical Control)
NCU	NC-Steuereinheit (engl.: Numerical Control Unit)
NPV	Nullpunktverschiebung

OBJ	Wavefont Object Format
PLC	Programmable Logic Controller (dt.: speicherprogrammierbare Steuerung)
PREC	Precision
PropFeatSet	proprietärer Merkmalsatz (engl.: Proprietary Feature Set)
PTW	Institut für Produktionsmanagement, Technologie und Werkzeugmaschinen
RF	Random Forest
REC	Recall
RidgeRe	Ridge-Classifier
RMS	Effektivwert (engl.: Root Mean Square)
ROC	Receiver-Operating-Characteristic
SPEC	Specificity
SPS	speicherprogrammierbare Steuerung
STFT	Short-Time-Fourier-Transformation
SVM	Support Vector Machine
TCP	Werkzeugmittelpunkt (engl.: Tool Center Point)
TN	Richtig-Negativ (engl.: True Negative)
TP	Richtig-Positiv (engl.: True Positive)
TPR	Richtig-Positiv-Rate (engl.: True Positive Rate)
TSFEL	Time Series Feature Extraction Library
WKS	Werkstückkoordinatensystem
WT	Wavelet-Transformation
WZM	Werkzeugmaschine
XT	Extra Trees

# 1 Einleitung

Das verarbeitende Gewerbe in Deutschland gilt, mit einem Anteil von über einem Fünftel an der gesamten Bruttowertschöpfung, als einer der führenden Wirtschaftszweige in Deutschland [SEIF18; STAT22]. Innerhalb dieser Branche stehen produzierende Unternehmen infolge von Globalisierung und steigender Digitalisierung zunehmend vor schwerer zu beherrschenden Herausforderungen. Die damit verbundene gestiegene Komplexität des täglichen Geschäftsablaufs verlangt nach schnellem unternehmerischen Handeln, um sich an die vorherrschenden volatilen wirtschaftlichen und politischen Rahmenbedingungen anzupassen. Dies setzt bei unternehmerischen Entscheidungen ein hohes Maß an Agilität sowie rechtzeitige und effektive Reaktionen auf kurzfristig eintretende Störungen voraus. Als wesentliche Auslöser gelten in diesem Zusammenhang neben den gestiegenen Schwankungen im Lieferangebot die zunehmende Dynamik in den Kundenanforderungen. [SCHU22]

Um diesen aktuellen Herausforderungen zu begegnen und Deutschland im internationalen Wettbewerb zu stärken, hat die Bundesregierung die „Strategie Künstliche Intelligenz“ herausgegeben. Künstliche Intelligenz (KI) gilt in diesem Zusammenhang als Treiber der Digitalisierung und wird als Schlüsseltechnologie für zusätzliches Wachstum und den nachhaltigen Erfolg der deutschen Wirtschaft angesehen. Ziel ist es, die Forschung und den Einsatz von KI in Unternehmen zu fördern und auszubauen, um im internationalen Vergleich auf diesem Gebiet führend zu sein. [BUND18; BUND20] Das damit zusammenhängende Potenzial von KI spiegelt sich zusätzlich in aktuellen Studien wieder, welche u. a. von einem weltweiten Marktpotenzial in Höhe von 13 Billionen USD bis 2030 für den Bereich KI ausgehen [BUGH18]. Mit dem Einsatz von KI im produzierenden Gewerbe wird ausgehend vom Jahr 2018 mit einer zusätzlichen Bruttowertschöpfung von knappen 32 Mrd. EUR innerhalb von fünf Jahren gerechnet [SEIF18]. Dies geht einher mit dem enormen Zuwachs an zur Verfügung stehenden Datenmengen, welche insbesondere durch die exponentiell fortschreitende Entwicklung leistungsfähigerer Hardware, stetig effizienter ausgewertet und verarbeitet werden können [BUND18].

Die gestiegenen Datenmengen erhöhen die Transparenz von Prozessen und Anlagen und führen somit zu einer zunehmenden Bandbreite an daraus ableitbaren Entscheidungen. In der industriellen Praxis gilt es die erfolgversprechenden Handlungen je nach gegebener Ist-Situation abzuleiten und die

richtigen Kompromisslösungen für die teilweise in Konkurrenz stehenden Zielvorgaben hinsichtlich Qualität, Produktivität und Verfügbarkeit zu erfüllen. Die steigende Komplexität der Systeme, die kürzer werdenden Entscheidungszeiträume und der folgenschwere Entscheidungscharakter erschweren es für den Menschen zunehmend, unmittelbare und flexible Lösungen herauszuarbeiten um die richtigen Entscheidungen zu treffen. KI ermöglicht in diesem Zusammenhang die Ableitung von datenbasierten Handlungsempfehlungen durch die Erkennung von Zusammenhängen in zugrundeliegenden Daten ohne die Notwendigkeit einer expliziten Modellierung. Lernende Modelle sind dazu konzipiert insbesondere bei komplexen und menschlich kaum greifbaren Situationen datenbasierte Entscheidungen zu treffen. Der Einsatz dieser innovativen Technologien hilft dementsprechend die Systemkomplexität zu durchdringen und die Leistungsgrenze von Produktionssystemen zu erhöhen (vgl. Abbildung 1.1). [BREC20a]

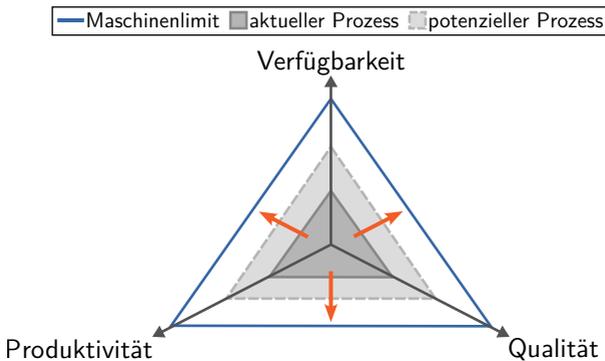


Abbildung 1.1: Erhöhung der Leistungsgrenze von Produktionssystemen durch den Einsatz innovativer Technologien, eigene Darstellung basierend auf [BREC20a].

Um das Potenzial aus der Nutzung der rasant anwachsenden Datenmengen möglichst auszuschöpfen und den komplexeren Herausforderungen zu begegnen, steigern zahlreiche Unternehmen die Entwicklung und Anwendung von Lösungen aus den Bereichen der Data Analytics und KI [SCHM20b]. In diesem Zusammenhang beschreibt die prädiktive Qualität (engl.: predictive quality) „die Befähigung des Anwenders zu einer datengetriebenen Vorhersage der produkt- und prozessbezogenen Qualität in der Herstellung und Nutzung physischer Produkte“ [SCHM20a] und eröffnet neue Möglichkeiten in der Anwendung von Qualitätsprognosen als Entscheidungsgrundlage für unter-

---

nehmerische Maßnahmen. Damit einher gehen die verbesserte Beherrschung der komplexer werdenden Prozesse sowie Einsparungen von Ressourcen in Form von reduzierten Aufwänden für die Qualitätsprüfung. Zusätzlich können durch Maßnahmen hinsichtlich prädiktiver Qualität Nachhaltigkeitsziele früher erreicht werden, indem Ausschuss verringert und die aufgewendeten Ressourcen infolge der verbesserten Prozessbeherrschung reduziert werden. [SCHM20b]

Ausgehend von diesen Gegebenheiten beschäftigen sich viele fertigende Unternehmen mit Fragestellungen hinsichtlich der notwendigen Schritte beginnend mit der Identifikation erforderlicher Daten, über die Datenerfassung hin zur Datenmodellierung, um effizient Lösungen für die datenbasierte Qualitätsprognose zu entwickeln und diese in der eigenen Produktion einzusetzen. Motiviert durch diese Fragestellung ergibt sich die Erarbeitung eines Lösungsvorschlags zur anwendungsnahen Entwicklung von KI-basierten Qualitätsprognosemodellen für Zerspanprozesse als Zielstellung dieser Arbeit. Die Schwerpunkte der nachfolgenden Ausführungen liegen dabei auf der kontextsensitiven Erfassung, Verarbeitung und Verknüpfung der Daten sowie auf einem möglichst hohen Automatisierungsgrad der charakteristischen Schritte hin zu finalen Prognosemodellen. Diese Arbeit soll Domänenexperten bei der Entwicklung und Umsetzung KI-basierter Lösungen für die Qualitätsüberwachung unterstützen und die Grundlagen für die Nutzung und Anwendung datenbasierter Methoden in der eigenen Produktion liefern. Durch den Einsatz der basierend auf dem vorgestellten Verfahren erstellten Prognosemodelle in einem prototypisch umgesetzten Qualitätsüberwachungssystem entsteht die Möglichkeit, die Ergebnisse dieser Arbeit anwendungsnah zu demonstrieren, um die daraus resultierenden Potenziale für datenbasierte Qualitätsprognosen in der spanenden Fertigung aufzuzeigen.



## 2 Stand des Wissens

Ausgehend von der Motivation, werden in diesem Kapitel die Grundlagen für die Entwicklung datenbasierter Qualitätsbewertungsansätze gelegt. Dazu wird in Kapitel 2.1.1 zunächst ein Verständnis über den Begriff der KI geschaffen und das darin liegende Potenzial für das Produktionsumfeld sowie im Speziellen für die Qualitätssicherung (Kapitel 2.1.2 und 2.1.3) herausgestellt. Daran anknüpfend thematisiert Kapitel 2.2 die notwendigen Grundlagen und Zusammenhänge hinsichtlich der Prozessüberwachung an Werkzeugmaschinen. Dafür werden die Signalflüsse in Werkzeugmaschinensteuerungen beschrieben, um diese als Datenquelle für aktuelle Ansätze zur Prozessüberwachung zu nutzen. Die Erläuterung des Qualitätsbegriffs im Kontext der spanenden Fertigung (Kapitel 2.3) dient schließlich dazu, den Stand der Forschung hinsichtlich Qualitätsüberwachung und Qualitätsbewertung von Zerspanprozessen darzustellen. In einem abschließenden Zwischenfazit wird der Forschungsbedarf abgeleitet, aus welchem die Zielstellung für die vorliegende Dissertationsschrift hervorgeht.

### 2.1 KI und deren Relevanz für Produktionsunternehmen

Infolge der zunehmenden Verbreitung von KI steigt deren Relevanz für das produzierende Gewerbe. Dieses Kapitel erklärt KI und die Abgrenzung zum maschinellen Lernen, um das Verständnis für die breiten Anwendungsmöglichkeiten dieser Methoden in der Produktion und insbesondere in der Qualitätssicherung zu verstehen.

#### 2.1.1 Begriffserklärung: KI

Für den Begriff „künstliche Intelligenz“ (engl.: Artificial Intelligence) existiert keine einheitliche Definition. Der Hintergrund dafür liegt im unterschiedlichen und komplexen Verständnis von KI, welches zudem einem stetigen und dynamischen Wandel unterliegt. Der US-amerikanische Informatiker John McCarthy nutzt 1955 den Begriff KI erstmalig in seinem Forschungsantrag für das Dartmouth Summer Research Project on Artificial Intelligence, mit dem

Tabelle 2.1: Definitionen KI.

COPELAND	KI ist „die Fähigkeit eines digitalen Computers [...] Aufgaben auszuführen, die normalerweise mit intelligenten Wesen in Verbindung stehen“ [COPE21].
LACKES	„KI beschäftigt sich mit Methoden, die es einem Computer ermöglichen, solche Aufgaben zu lösen, die, wenn sie vom Menschen gelöst werden, Intelligenz erfordern“ [LACK18].
DFKI	„KI ist die Eigenschaft eines IT-Systems, „menschenähnliche“, intelligente Verhaltensweisen zu zeigen“ [BUSC17].
Europäisches Parlament	„KI ist die Fähigkeit einer Maschine, menschliche Fähigkeiten wie logisches Denken, Lernen, Planen und Kreativität zu imitieren“ [PARL20].
MICROSOFT	„[...] Technologien, die menschliche Fähigkeiten im Sehen, Hören, Analysieren, Entscheiden und Handeln ergänzen und stärken“ [RONS20].
WICHERT	„Teilgebiet der Informatik, welches sich mit der Erforschung von Mechanismen des intelligenten menschlichen Verhaltens befaßt (Intelligenz)“ [WICH22].

Ziel, die „Aspekte des Lernens oder jede andere Eigenschaft von Intelligenz“ [MCCA06] zu beschreiben, um intelligente Systeme und Maschinen zu entwickeln. [MOCK21] Der daraufhin stattgefundene Workshop markiert den Startpunkt von KI als wissenschaftliche Disziplin [LAKE17]. Zuvor beschäftigt sich TURING 1950 mit der zentralen Frage: „Can machines think?“ in seiner Publikation „Computing Machinery and Intelligence“ [TURI50]. Dabei schlägt TURING ein Testverfahren vor, um beurteilen zu können, ob ein System intelligent ist. Heute bekannt als Turing Test. Bei diesem Test stellt ein Mensch über eine elektronische Leitung Fragen an einen anderen Menschen bzw. an eine KI. Der Computer hat den Test bestanden und kann dementsprechend nach Turing als intelligent eingestuft werden, wenn der menschliche Fragesteller mehrmals nicht mit Sicherheit sagen kann, ob er mit einer Person oder einer Maschine kommuniziert. [MCCO04; TURI50]

Je nach Blickwinkel existiert ein unterschiedliches Begriffsverständnis für KI (vgl. Tabelle 2.1). Die Definitionen von COPELAND und LACKES interpretieren KI als Fähigkeiten und Methoden, die es Computersystemen ermöglichen intelligente Aufgaben zu lösen. Die Systeme sollen also in die Lage versetzt werden, selbstständig zu lernen und sich ähnlich zu Menschen, intel-

ligent zu verhalten [PAAS20]. Das Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI) und das Europäische Parlament (EuPa) erweitern das Verständnis um menschliche bzw. menschenähnliche Fähigkeiten oder Verhaltensweisen. Diese spiegeln sich laut EUPA im logischen Denken, Lernen, Planen und in der Kreativität wider. MICROSOFT betrachtet KI weniger als eine Systemeigenschaft, sondern versteht KI als Zusammenfassung von Technologien, welche menschliche Fähigkeiten ergänzen und stärken können. Der Mensch und die Förderung seiner Fähigkeiten stehen hierbei im Mittelpunkt. [RONS20] WICHERT definiert KI pragmatisch als Forschungsgebiet der Informatik. Aus den genannten Definitionen gehen die verschiedenen Sichtweisen für den Begriff KI je nach Interpretation und Zielstellung hervor.

Trotzdem wird in jeder Definition von „Intelligenz“ bzw. „intelligentem Verhalten“ aus menschlicher Sicht gesprochen. Um ein besseres Verständnis für die angesprochene menschliche Intelligenz aufzubauen, dominieren neben einer Reihe von unterschiedlichen Beschreibungen die multidimensionalen und prozessorientierten Modelle der Intelligenz [ZIEG22]. In diesem Zusammenhang liefert GARDNER die Theorie der multiplen Intelligenzen in Form der acht folgenden eigenständigen Intelligenzdimensionen [GARD93; PAAS20; ZIEG22]:

- **sprachlich:** Verständnis von Sprache, Sensitivität gegenüber Wortbedeutungen.
- **logisch-mathematisch:** mathematische und logische Denkleistungen.
- **räumlich:** räumliches Denken, Fähigkeiten zur Raumwahrnehmung.
- **musikalisch:** musikalische, künstlerische Kompetenzen.
- **körperlich-kinästhetisch:** Bewegungskompetenzen, körperliche Geschicklichkeit.
- **selbstreflektiv:** Sensibilität gegenüber eigener Empfindungswelt.
- **zwischenmenschlich:** Fähigkeit, Stimmungen, Motivationen und Intentionen anderer Menschen zu erkennen.
- **naturalistisch:** Fähigkeit, Naturphänomene zu beobachten, zu unterscheiden, zu erkennen sowie eine Sensibilität dafür zu entwickeln.

Die Dimensionen beschreiben Leistungen in unterschiedlichsten Anwendungsfeldern. KI verfolgt demnach das Ziel, Systeme zu entwickeln, welche in den genannten Feldern intelligente Leistungen liefern. [PAAS20] Der Fokus liegt dabei nicht auf der Nachahmung der Verhaltensweise des menschlichen Gehirns, sondern auf der Entwicklung von Systemen, welche menschenähnliche,

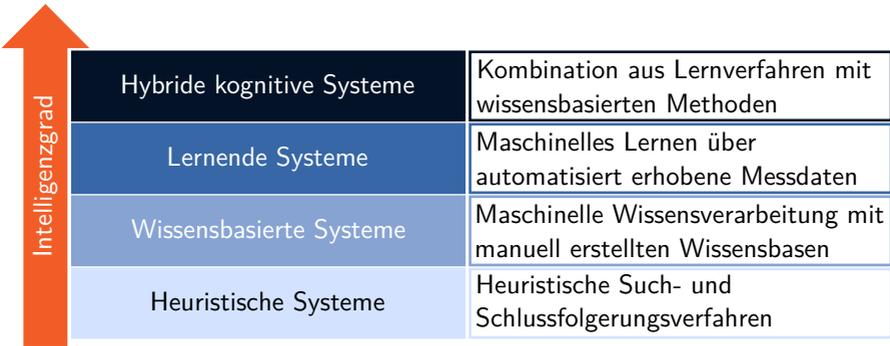


Abbildung 2.1: Die vier Phasen der KI, eigene Darstellung basierend auf [WAHL20].

intelligente Fähigkeiten besitzen oder in gewissen Disziplinen dem Menschen sogar überlegen sind [LAKE17].

Im Verlauf der Zeit kommt es jedoch zu ständigen Veränderungen des Verständnisses und der Dimension menschlicher Intelligenz. Infolgedessen lässt sich im Verlauf der KI-Entwicklung häufiger beobachten, dass sobald ein System eine Lösung für ein Problem liefern konnte, dieses nicht mehr zum Kern der menschlichen Intelligenz gezählt wird. Grund dafür ist die Feststellung, dass zur Lösung des Problems keine Intelligenz notwendig ist. [MCCO04] Ein prominentes Beispiel hierfür ist das Schachspielen. Logische Bewertungen von Schachzügen ermöglichten Computerprogrammen Schachweltmeister zu schlagen. Dies führte dazu, dass Schachspielen als „mechanistisches“ Schließen abgewertet wurde. [MOCK21] Aus diesen stetigen Veränderungen entstehen aus heutiger Sicht vier Entwicklungsstufen der KI mit steigendem Intelligenzgrad (vgl. Abbildung 2.1). Mit dem Erreichen der lernenden Phase ist es möglich, basierend auf einer qualitativ hochwertigen Datenbasis, ohne die Notwendigkeit einer manuell erstellten Wissensbasis, selbstlernende Systeme bei geringen Entwicklungsaufwänden und leichter Anpassbarkeit zu entwickeln. [WAHL17]

Die Vielzahl an Definitionen und die unterschiedlichen Ausprägungen unterstreichen die Problematik hinsichtlich eines einheitlichen und eindeutigen Verständnisses von KI. Zusammenfassend ist festzuhalten, dass KI seit ihrem Ursprung in den 50er Jahren des 20. Jahrhunderts verschiedene Entwicklungsphasen durchläuft und sich ihre Bedeutung stetig verändert bzw. mit den gegebenen Randbedingungen erweitert. Dabei dienen das Verhalten sowie die

intelligenten Fähigkeiten des Menschen als Anlehnung für das Verständnis von KI im jeweiligen Kontext des Einsatzgebiets.

## 2.1.2 ML und das Potenzial von KI im Produktionsumfeld

In öffentlichen Diskussionen werden maschinelles Lernen (engl.: Machine Learning) (ML) und KI häufig als Synonyme verwendet [KOTU19]. MITCHELL definiert ML als „[...] study of algorithms that allow computer programs to automatically improve through experience“ [MITC97]. ML nutzt demnach Algorithmen, welche Maschinen in die Lage versetzen aus Erfahrungen zu lernen und beschreibt ein Teilgebiet von KI. Die Algorithmen erhalten ihre Erfahrungen in Form von Daten. [KOTU19] Der Grundgedanke liegt darin, einem Algorithmus ein bestimmtes, aus früherer Erfahrung hervorgehendes Verhalten anzutrainieren, um den trainierten Algorithmus bzw. das resultierende Modell in die Lage zu versetzen, Ergebnisse auf einem noch nie zuvor gesehenen Datensatz zu erzeugen [RAZ20]. Die für das Training der Algorithmen genutzten Daten bezeichnen sich als Trainingsdaten. Dies bringt den Unterschied zur traditionellen Programmierung zum Vorschein. Während in der analytischen Programmierung die Eingangsdaten analysiert werden, um für den Algorithmus einen Satz von Anweisungen zu erstellen, welcher Eingangsdaten in Ausgangsdaten umsetzt, verknüpfen ML-Algorithmen die Eingangsdaten mit den Ausgangswerten, um ein Modell für die Umwandlung von Eingabe (engl.: Input) in die Ausgabe (engl.: Output) zu bilden (vgl. Abbildung 2.2). Als Deep Learning (DL) wird eine Unterklasse von ML-Algorithmen bezeichnet, welche auf tiefen neuronalen Netzen basiert. DL-Algorithmen eignen sich vor allem dafür, sehr komplexe Zusammenhänge in den Daten zu modellieren. [GOOD16; SCHM15]

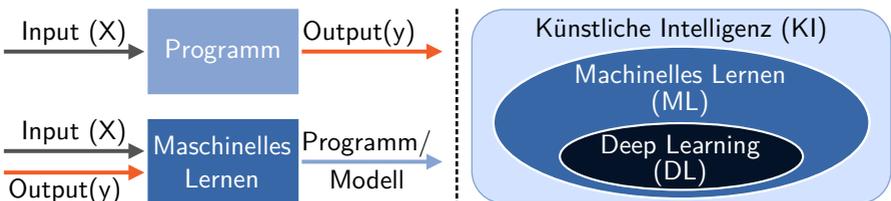


Abbildung 2.2: (links) Traditionelle Programmierung und ML, übersetzt und angelehnt an [KOTU19]. (rechts) Abgrenzung KI, ML und DL.

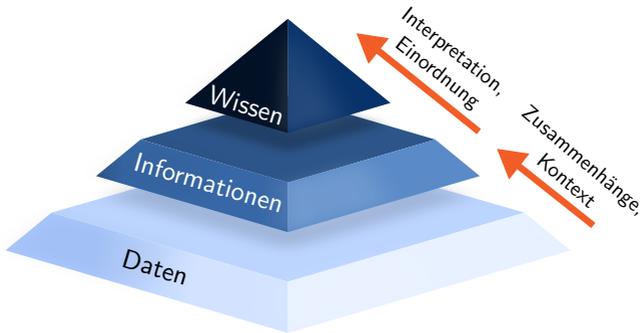


Abbildung 2.3: Zusammenhang zwischen Daten, Informationen und Wissen, eigene Darstellung basierend auf [MOCK21].

Das Trainieren von ML-Algorithmen mithilfe historischer Datensätze, erfordert große und besonders aussagekräftige Datenmengen. Dabei spielen sowohl die Verfügbarkeit als auch die Qualität an Daten eine essentielle Rolle für die Leistungsfähigkeit und damit die erzielten Ergebnisse des erstellten Modells. [HEIN19] Die Algorithmen nutzen die in den Daten enthaltenen Informationen – „Bedeutung, die Daten mit Hilfe von Regeln, die für diese Daten gelten, momentan zugewiesen wird“ [EN61499-1] – für das Erlernen der für das Modell benötigten Verhaltensweisen. Entsprechend lernen ML-Systeme die Bedeutung der Informationen, um daraus das nötige Wissen für die jeweilige Anwendung zu generieren. Daten erhalten Informationen indem diese in einen Zusammenhang, dem sog. Kontext, gesetzt werden. Den relevanten Kontext erhalten Daten über beschreibende Metadaten respektive Metainformationen. Durch die Einordnung bzw. Interpretation der Informationen auf Basis (menschlicher) Erfahrung entsteht schließlich das Wissen. [MOCK21] Abbildung 2.3 veranschaulicht den Zusammenhang zwischen Daten, Informationen und Wissen.

Diese Grundeigenschaft von ML eröffnet neue Lösungsmöglichkeiten von Problemen und Aufgaben, bei denen der zugrundeliegende Zusammenhang nicht bekannt und demnach eine exakte Modellierung unmöglich oder nur mit sehr großem Aufwand realisierbar ist [HEIN19]. Dies führte in Kombination mit den Entwicklungen der letzten Jahre, hinsichtlich leistungsfähigerer und kostengünstigerer Hardware sowie größerer Mengen an verfügbaren Daten und Informationen infolge der voranschreitenden Digitalisierung und Vernetzung zu einem enormen Anstieg der gesellschaftlichen und wirtschaftlichen Re-

relevanz von KI und ML. Weitere Faktoren sind die gesteigerte Leistungsfähigkeit der Algorithmen, die breiten Investitionen in KI von Universitäten, Regierungen, Startups sowie Tech-Giganten (z. B. Google, Amazon, Microsoft oder Facebook) und der erleichterte Zugang zur Thematik durch die Bereitstellung von Open-Source-Frameworks [DÖBE18; DRIS11; FUMO17]. Diese Entwicklungen bergen enorme Potenziale in Form von Effizienzsteigerungen, neuen Geschäftsmodellen und qualifizierten Arbeitsplätzen beim Einsatz von KI in Unternehmen [GERE19]. Infolge des Trends zur Digitalisierung und Industrie 4.0 nimmt die Verfügbarkeit an Daten vor allem im Bereich der Fertigung zu, was wiederum zur Verbreiterung der Einsatzmöglichkeiten von ML im produzierenden Gewerbe führt. Industrie 4.0 Anwendungen zielen dabei auf Verbesserungen hinsichtlich Qualität, Zeit und Kosten ab. Zusätzlich liegt dabei der Fokus auf dem Komplexitätshandling sowie auf Flexibilitätssteigerungen. [DELO16]

Es stellt sich die Frage nach den relevanten Einsatzbereichen von KI in produzierenden Unternehmen. Eine Studie von HATIBOGLU zeigt die in Abbildung 2.4 zusammengefassten relevanten Anwendungsfelder in produktionsnahen Bereichen auf [HATI19]. Auffällig dabei ist die Breite in der KI-Methoden im produzierenden Gewerbe zum Einsatz kommen. Vor allem gilt die Fertigung nach MCKINSEY als einer der wichtigsten Einsatzbereiche für KI [MCKI17]. Als die vier zentralen Kernbereiche innerhalb von Produktionsprozessen gelten KI-basierte prädiktive Wartungssysteme, kollaborative und kontextsensitive Robotik, Effizienz- und Ertragssteigerungen sowie KI-basierte Qualitätskontrollen. Eine entscheidende Rolle spielen dabei vor allem Technologien aus dem Bereich der Computer Vision und des ML. [SEIF18]

Instandhaltung	Logistik	Qualitätsmanagement und -kontrolle	Produkt- und Prozessentwicklung
Digitale Assistentensysteme	Automatisierungstechnik	Prozessoptimierung und -steuerung	Ressourcenplanung

Abbildung 2.4: Übersicht der KI Anwendungsfelder in produktionsnahen Bereichen, angelehnt an [HATI19].

### 2.1.3 KI im Kontext der Qualitätssicherung

Empirische Studien belegen einen positiven Zusammenhang zwischen einer Verbesserung der Produktqualität und einer gesteigerten Kundenzufriedenheit. Diese gilt als entscheidender Faktor für die Gewährleistung eines langfristigen Unternehmenserfolgs. [HINT09] Dieser positive Einfluss von hoher Produktqualität auf den Unternehmenserfolg spielt insbesondere für Unternehmen aus Hochlohnländern, wie beispielsweise Deutschland eine entscheidende Rolle, wodurch die wirtschaftliche Realisierung einer lückenlosen, zuverlässigen Qualitätssicherung besonders im Mittelpunkt steht. Gründe dafür sind u. a. der zunehmende globalisierte Wettbewerb, wodurch geographische Standortvorteile als Differenzierungsmöglichkeit von Konkurrenten wegfallen und steigende Kundenerwartungen an die Qualität, Zuverlässigkeit und leichte Handhabung der Produkte, welche auch zunehmend in Schwellenländern Einzug finden. [BRÜG15; MAJS19]. Die für diesen Zweck üblicherweise eingesetzte Fertigungsmesstechnik sieht sich jedoch zunehmend steigenden Anforderungen hinsichtlich Geschwindigkeit, Genauigkeit, Sicherheit und Flexibilität konfrontiert und stößt dabei zunehmend an ihre Grenzen [MAJS19]. Um dieser Problematik entgegen zu wirken, können datenbasierte Methoden helfen die aktuellen Herausforderungen zu überwinden und die Qualität der Produkte weiter zu erhöhen [SCHM20a]. Nach KÖKSAL und ROSTAMI lassen sich die Einsatzfelder von datenbasierten ML-Ansätzen für qualitätsbezogene Aufgaben in vier Kernbereiche einteilen [KÖKS11; ROST15]:

- Beschreibung der Qualität bzw. Untersuchung der Einflussfaktoren auf die Produkt- oder Prozessqualität
- Klassifikation der Produktqualität
- Vorhersage der Produktqualität
- Parameteroptimierung

Für die Umsetzung eines datenbasierten Ansatzes zur Qualitätssicherung bildet die Beschreibung der Qualität bzw. die Untersuchung der Einflussfaktoren auf die Produkt- bzw. Prozessqualität die Grundlage. Darauf aufbauend verknüpft ein Vorhersagemodell die verfügbaren qualitätsbezogenen Eingangsdaten mit der resultierenden Qualität des Produktes oder Prozesses. Unterschieden werden kann dabei in die Klassifikation von Qualitätsmerkmalen und die Vorhersage in Form von kontinuierlichen Werten (Regression). Schließlich ermöglicht ein datenbasierter Ansatz eine Identifikation und Optimierung der relevantesten Einflussparameter auf die resultierende Qualität. Diese Vorgehensweise eröffnet u. a. neue Wege im Hinblick auf die Reduktion

von Ausschuss durch frühes Eingreifen und die Verbesserung der Produktqualität. [KÖKS11; ROST15; SCHM20a] Für diese Arbeit stehen die ersten drei der genannten Kernbereiche im Fokus der Untersuchungen. Dabei erfolgt die Beschreibung der Grundlagen in den Kapiteln 2.3 und 2.4.

## 2.2 Prozessüberwachung an Werkzeugmaschinen

Die Prozessüberwachung an Werkzeugmaschinen liefert die grundlegenden Zusammenhänge für die Umsetzung von KI-Ansätzen zur Qualitätssicherung und basiert auf prozessnahen Informationen, welche durch entsprechende Signalverarbeitungsmethoden (Kapitel 2.2.5) eine Prozessbewertung zulassen. Diese Informationen können aus unterschiedlichen Datenquellen gewonnen werden. In diesem Kapitel werden ausgehend von der Definition für Werkzeugmaschinen (Kapitel 2.2.1) und der Funktionsweise von numerischen Steuerungen (Kapitel 2.2.2 und 2.2.3) die verschiedenen Zielgrößen und Aufgabenbereiche von Prozessüberwachungslösungen herausgestellt (Kapitel 2.2.4). Dies liefert das Verständnis für die anschließende Diskussion aktueller Prozessüberwachungsansätze aus Forschung und Industrie (Kapitel 2.2.6).

### 2.2.1 Spanende Werkzeugmaschinen

Eine Maschine, „[...] die der Fertigung mechanischer Komponenten definierter, reproduzierbarer Form mit Hilfe von Werkzeugen [...]“ [NEUG12] dient, wird als Werkzeugmaschine (WZM) bezeichnet. Für die Formgebung führen WZM zwischen dem Werkzeug und Werkstück Relativbewegungen aus. Die Relativbewegungen lassen sich in Prozess- und Vorschubbewegungen unterscheiden. [NEUG12] Neben dieser Erläuterung definiert DIN 69651 WZM als „[...] mechanisierte und mehr oder weniger automatisierte Fertigungseinrichtungen, die durch relative Bewegungen zwischen Werkzeug und Werkstück eine vorgegebene Form oder Veränderung am Werkstück erzeugen“ [DIN69651].

Entsprechend der Definitionen lässt sich die große Bandbreite unterschiedlicher WZM erschließen. DIN 69651 beschreibt die Bezeichnung von WZM gemäß der Einteilung von Fertigungsverfahren nach DIN 8580 in Hauptgruppen, Gruppen und Untergruppen (Abbildung 2.5). Ist eine WZM in der Lage nur ein Verfahren auszuführen, erfolgt die Bezeichnung nach dem entsprechenden Verfahren. Können hilfsweise weitere Verfahren auf der WZM ausgeführt werden, erfolgt die Bezeichnung nach dem Verfahren für das diese hauptsächlich gebaut und ausgerüstet ist, in Kombination mit dem Suffix „-zentrum“. Dementsprechend ist ein Fräszentrum beispielsweise auch in der

Lage zu Bohren. Bei der vollwertigen Beherrschung mehrerer Verfahren wird dem zuvor beschriebenen Wortteil („-maschine“ bzw. „-zentrum“) das Attribut „Bearbeitungs-“ beigefügt. Die Bezeichnung von Einzelmaschinen bzw. Universalmaschinen („[...] für ein breiteres Werkstückspektrum universell einsetzbare Werkzeugmaschinen [...]“ [DIN69651]) oder Mehrmaschinensystemen<sup>1</sup> erfolgt basierend auf dem Automatisierungsgrad nach Abbildung 2.6. [BREC19d; DIN69651]

Bei den in dieser Arbeit betrachteten WZM (vgl. Kapitel 4.1.1) handelt es sich nach diesen Definitionen um Fräs-Bearbeitungszentren. Die Universalmaschinen gehören zu den spanenden Maschinen für Werkzeuge mit geometrisch bestimmter Schneide und sind hauptsächlich für das Fertigungsverfahren Fräsen ausgelegt und ausgerüstet. Bei beiden Maschinen handelt es sich um vollautomatisierte Bearbeitungszentren mit jeweils eigenem Werkzeugspeicher und automatisiertem Werkzeugwechsel. Diese sind in der Lage neben Fräsoperationen u. a. zusätzlich Bohr-, Reib- und Senkprozesse durchzuführen. Eine nähere Beschreibung erfolgt in Kapitel 4.1.1.

---

<sup>1</sup>zusammengefasst bezeichnet als Fertigungssysteme

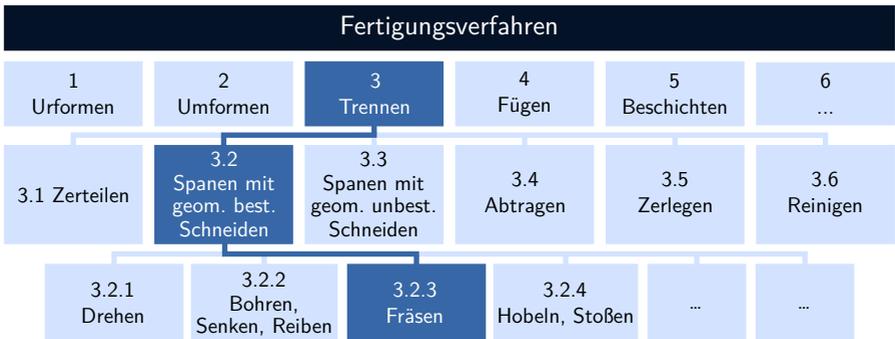


Abbildung 2.5: Einteilung der Fertigungsverfahren nach DIN8580. Unterscheidung von Hauptgruppen, Gruppen und Untergruppen von der ersten Zeile zur letzten Zeile, angelehnt an [DIN8580].

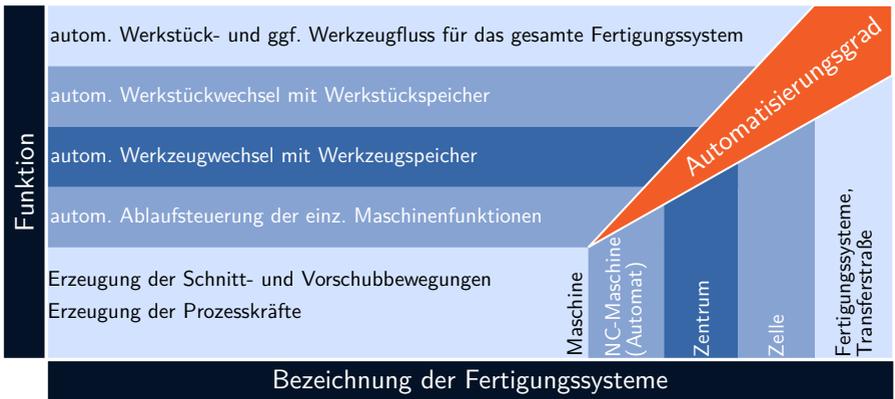


Abbildung 2.6: Bezeichnung der Fertigungssysteme nach dem Automatisierungsgrad, angelehnt an [BREC19d].

## 2.2.2 Numerische Steuerungen

Für die Ausführung der Relativbewegungen zwischen Werkzeug und Werkstück kommen numerische Steuerungen (engl.: Numerical Control) (NC) zum Einsatz. Die Aufgabe von NC besteht in der Interpretation und Aufbereitung von Eingangsinformationen in Form von NC-Programmen (Teileprogrammen), um diese in Weg- und Geschwindigkeits- sowie Schaltanweisungen für jede Achse umzuwandeln. Die in NC-Programmen festgelegten Fertigungsschritte eines Werkstücks entsprechen einer geordneten Folge an Anweisungen in Form von alphanumerischen Zeichen. [BREC21; ISO2806; SEKL12] Abbildung 2.7 veranschaulicht den Informationsfluss einer NC, welchen die Eingangsinformationen durchlaufen, um schließlich in den Bewegungen der Achsen zu resultieren.

Der **NC-Interpreter** übersetzt die im Teileprogramm enthaltenen Anweisungen (üblicherweise NC-Programme nach DIN 66025) in ein einheitliches, intern lesbares Format und decodiert diese nach geometrischen und technologischen Daten sowie Schaltinformationen. Die Schaltinformationen werden an die Anpassteuerung, heutzutage von speicherprogrammierbaren Steuerungen (SPS) realisiert, übergeben, welche die Steuerbefehle für die zu schaltenden Aggregate erzeugt und umsetzt. Typische Schaltbefehle sind M-Funktionen für z. B. Werkzeugauswahl, Kühlmittel ein/aus, Spindeldrehung usw.

Die geometrischen und technologischen Daten werden an die **Geometriedatenverarbeitung** übergeben. Diese erhält zu jedem NC-Satz die Lage der Bahnneckpunkte, die programmierte Bewegungsart sowie Vorschubgeschwindigkeit. In diesem Funktionsblock finden die Geschwindigkeitsführung, geometrische Transformation und Korrekturberechnungen statt. Die geometrische Transformation ermöglicht Korrekturen hinsichtlich relativer Werkstückpositionierung zum Maschinennullpunkt<sup>2</sup> und Werkzeugdaten unabhängig vom Maschinentyp. Die Nullpunktverschiebung (NPV) beschreibt die Lage des Werkstücknullpunkts in Bezug zum Maschinennullpunkt. Diese ermöglicht weitere Transformationen zwischen dem Maschinenkoordinatensystem (MKS) und dem Werkstückkoordinatensystem (WKS). Zusätzlich zur NPV gewährleistet die Werkzeugkorrektur die Unabhängigkeit der NC-Programmierung von den tatsächlichen Werkzeugabmessungen und besteht in der Regel aus der Werkzeuglängen- und Werkzeugradiuskorrektur. Darüber hinaus dient die Werkzeugkorrektur der Berechnung der Werkzeugwege zur Herstellung der programmierten Kontur unabhängig vom eingesetzten Werkzeug.

---

<sup>2</sup>in Abbildung 2.7 entsprechend Siemens Nomenklatur als FRAME-Transformation bezeichnet

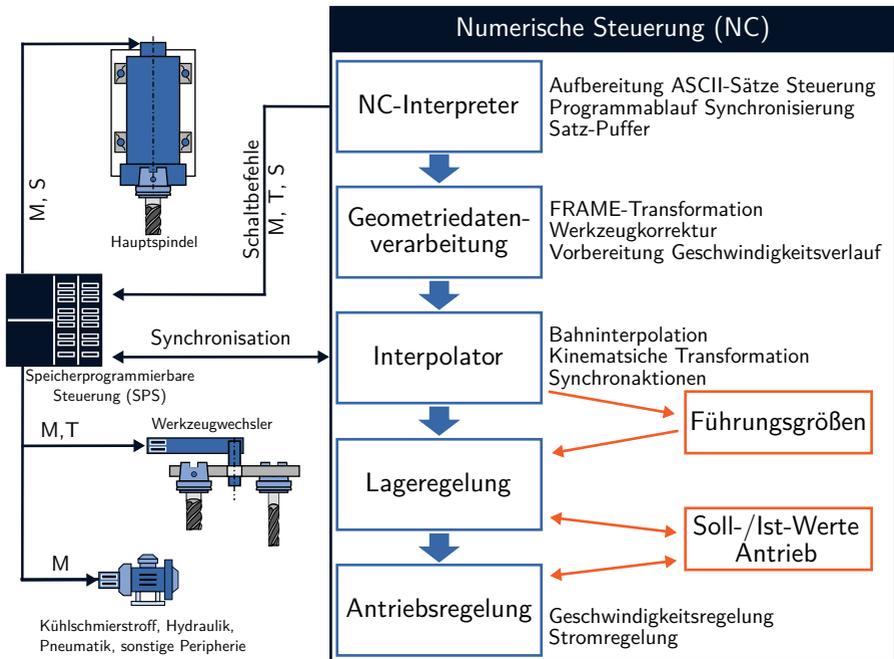


Abbildung 2.7: Informationsfluss einer NC, eigene Darstellung basierend auf [BREC21; WECK06].

Mit der Look-Ahead-Funktion werden mehrere Sätze im Voraus analysiert und Geschwindigkeits- sowie Richtungsänderungen der Vorschubbahn ermittelt. Auf diese Weise erfolgt die vorausschauende Anpassung der Geschwindigkeits- und Beschleunigungsprofile auf Basis der programmierten Vorschubgeschwindigkeit unter den gegebenen Randbedingungen. Dabei wird im Rahmen der Geschwindigkeitsführung überwacht, ob die Antriebe aufgrund ihrer begrenzten Dynamik in der Lage sind die vorgegebenen Geschwindigkeitsprofile einzuhalten. Ist dies nicht der Fall, erfolgt dementsprechend eine angepasste Geschwindigkeitsführung. [BREC21; SEKL12; SIEM18b]

Der **Interpolator** übernimmt die Berechnung der Positionssollwerte zu diskreten Zeitpunkten für die einzelnen Maschinenachsen. Die definierten Bahnabschnitte aus den NC-Sätzen werden abhängig vom Geschwindigkeitsprofil in Zwischenpunkte zerlegt. Dabei wird die Bahnbewegung, welche im Allgemeinen aus mehreren Achsen besteht, in stückweise Bewegungen der einzelnen

Maschinenachsen aufgeteilt und es entsteht eine voneinander abhängige Positionierung der Einzelachsen. Mögliche Interpolationsverfahren sind Geraden-, Kreis- oder Spline-Interpolationen. Im Interpolator erfolgt zusätzlich die Synchronisation zwischen den Schaltsignalen aus der SPS, der Arbeitsspindel und der Bewegungsausführung sowie die Reaktion auf äußere Einwirkungen, wie Vorschub- bzw. Drehzahlkorrekturen oder Verlagerungen mittels Kompensationsberechnungen. [BREC21; SEKL12; WOLF09] Der Interpolatortakt  $t_{\text{Ipo}}$  beschreibt die geltende Taktzeit für die Interpolation und beträgt ganzzahlige Vielfache des Lageregeltakts  $t_{\text{Lage}}$  [SIEM20]. Übliche Zykluszeiten für  $t_{\text{Ipo}}$  betragen 4–12 ms [REHS99].

Im Anschluss werden die interpolierten Sollpositionswerte an die Lageregelung und schließlich an die Antriebsregelung übergeben. Diese werden im folgenden Kapitel genauer beschrieben.

### 2.2.3 Maschineninterne NC-Signaldaten

Gemäß [ISO2806] kommen in NC kaskadierte Lageregelkreise mit unterlagerten Geschwindigkeits- und Stromrückführung zum Einsatz. Abbildung 2.8 zeigt den Regelkreis einer Vorschubachse. Innerhalb der Reglerkaskaden erfolgt für die Realisierung der programmierten NC-Prozessführung die Verarbeitung von Daten, welche im weiteren Verlauf dieser Arbeit angelehnt an die Ausführungen in [WELL19] als NC-Signaldaten bezeichnet werden. Diese sind zum einen die vom Interpolator vorgegebenen Führungsgrößen und zum anderen die von interner Sensorik erfassten, zurückgeführten Regelgrößen. Die rückgeführten Daten zu den Istpositionen der jeweiligen Achse werden entweder aus einem Drehgeber am Motor berechnet (indirektes Messprinzip) oder über einen direkten Positionsgeber auf der Achse zurückgespielt. Moderne WZM nutzen fast ausschließlich direkte Positionsgeber für die Regelung. Der Grund hierfür liegt in mechanisch und thermisch induzierten Abweichungen zwischen dem indirekten Drehgeber und dem Werkzeugschlitten. Die Drehzahlrückführung im Geschwindigkeitsregelkreis erfolgt üblicherweise durch numerisches Differenzieren des Wegsignals aus dem indirekten Drehgeber am Antrieb. Die Messung der Stromistwerte lässt sich über das Transformatorprinzip per Ringkerntransformator oder Hallelemente realisieren. [BAHM13; BREC21; KAEV05; WELL19; YOHA13] Für die Regelung der Hauptspindel kommt üblicherweise ein Drehzahlregelkreis mit unterlagertem Stromregelkreis zum Einsatz. In modernen Spindeln kann mithilfe eines Motorgebers die inkrementelle Spindelpositionierung rückgeführt werden. [BAHM13] Im Zuge der numerischen Steuerung und Regelung entstehen in jedem Takt Daten für

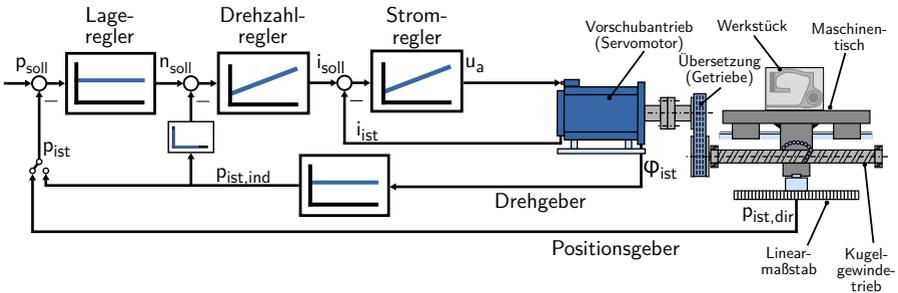


Abbildung 2.8: Maschineninterne Signalflüsse in der kaskadierten Regelung einer Vorschubachse, angelehnt an [BREC21; WELL19].

die Lageregelung, welche im nächsten Takt überschrieben und nicht weiter verwendet werden. Übliche Taktzeiten des Lagereglers liegen im Bereich von 1–4 ms [REHS99]. Neben den Signalen aus dem Lageregelkreis fallen innerhalb der Steuerung zahlreiche weitere Daten an. Dazu gehören Schaltbefehle, Geometriedaten, Technologiedaten und zusätzliche Daten aus der SPS wie beispielsweise Korrekturwerte, Kompensationstabellen oder Werkzeugdaten.

Um diese Daten für die Prozessanalyse und -überwachung nutzbar zu machen, gilt es diese strukturiert aufzuzeichnen. Als Grundlage dafür dienen offene NC, welche die Möglichkeit einer Funktionserweiterung durch die Integration zusätzlicher Softwarebausteine bieten. [PRIT01; PRIT93] Grundsätzlich wird in diesem Kontext zwischen innerer und äußerer Offenheit unterschieden. Äußere Offenheit adressiert vor allem den Anwender und beschreibt die Vereinheitlichung von externen Schnittstellen einer Steuerung. Grundeigenschaften hierfür sind neben der einheitlichen Programmierung von NC und SPS auch erweiterbare bzw. standardisierte Benutzerschnittstellen oder Antriebschnittstellen. Moderne NC nutzen bereits zahlreiche Normen und Standards zur Erfüllung der Anforderungen hinsichtlich äußerer Offenheit. Schnittstellen in Form der existierenden Feldbussysteme (z. B. PROFIBUS-MC, PROFINET, EtherCAT, SERCOS), standardisierter Schnittstellen wie OPC-UA (Open Platform Communications United Architecture) oder DDE (Dynamic Data Exchange) und die Programmierung von NC-Teilprogrammen nach DIN 66025 sind äußerer Offenheit zuzuordnen. Zusätzlich nutzen industrielle Überwachungssysteme häufig offene Schnittstellen an der SPS für den Zugriff auf Steuerungssignale zur Überwachung. [BREC21; KAEV05; WOLF09; YOHA13]

Für den Zugriff auf die steuerungsinternen Funktionsbausteine (NC-Kern) ist die innere Offenheit von Steuerungen notwendig. Eigenschaften innerer Offenheit sind weitreichende Zugriffsmöglichkeiten auf Steuerungs- und Prozessdaten, konfigurierbare Software mit offenen Schnittstellen und eine modular aufgebaute Hardware. [WECK93; WOLF09] Im Rahmen dieser Arbeit erfolgen die Analysen und Entwicklungen prototypisch an WZM mit Steuerungen vom Typ SIEMENS SINUMERIK 840D sl. Deshalb wird im Folgenden gezielt auf die innere Offenheit dieses Steuerungstyps und die damit verbundenen Möglichkeiten zur Erfassung interner Antriebsdaten eingegangen. Die Firma Siemens bietet für Kunden die Möglichkeit Compileryklen (in C++ erstellte Anwendungen) in einer proprietären Entwicklungsumgebung zu entwickeln, um die Funktionalität des NC-Kerns zu erweitern. Nach der Freigabe durch den Steuerungshersteller bieten die Compileryklen die gleiche Echtzeitfähigkeit wie der NC-Kern. Die Eingriffsmöglichkeiten mit Compileryklen reichen vom NC-Interpreter bis hin zur Lageregelung und können somit auf alle Komponenten des NC-Kern-Basissystems zugreifen. [BREC21; KAEV05; WOLF09; YOHA13]

## 2.2.4 Prozessüberwachung in der spanenden Fertigung

Unter einer Überwachung in der Fertigung (engl.: Monitoring) wird grundsätzlich die Verarbeitung von gemessenen Sensorsignalen zur Bestimmung des Prozesszustands verstanden [ULSO06]. Das Ziel dabei ist es, Maschinenstillstandszeiten zu minimieren und hohe Prozesssicherheiten, bei beaufsichtigten oder unbeaufsichtigten Bearbeitungen, zu gewährleisten, um insgesamt eine möglichst hohe Produktivität zu erreichen. Im Fokus der Überwachung können sowohl die Maschine als auch der Prozess stehen. Während es bei der Maschinenüberwachung (engl.: Condition Monitoring) vorrangig darum geht, den aktuellen Zustand der Komponenten zu beobachten, um eine frühzeitige Fehlererkennung zu gewährleisten, verfolgt die Prozessüberwachung (engl.: Process Monitoring) im Hinblick auf die Gewährleistung der Prozesssicherheit das Ziel, Prozessstörungen zu verhindern und zu beseitigen sowie die geforderte Produktqualität einzuhalten. [BREC21] Im Fokus dieser Arbeit steht die Qualität der gefertigten Werkstücke und somit der Prozess im Mittelpunkt, weshalb auf die Überwachung von Maschinenkomponenten im weiteren Verlauf nicht weiter eingegangen wird.

Aus Prozesssicht lassen sich verschiedene Überwachungsbereiche (engl.: Monitoring Scopes) unterscheiden [ABEL10; LIAN04; TETI10; TÖNS88]:

- **Werkzeugzustand:** Werkzeugverschleiß, Werkzeugbruch, Werkzeuggeometrie, Werkzeugtemperatur
- **Spanbildung:** Spanform, Spanbruch, Spanabfuhr, Spanverteilung
- **Prozess:** Prozessfehler, Prozesszustand, Rattern, Schnittbedingungen, Prozessvariationen
- **Werkstückzustand:** Abmaße, Form, Rohteilschwankungen, Werkstoffschwankungen
- **Oberflächenintegrität:** Oberflächengüte, Oberflächenrauheit, Oberflächenfehler

Für einen zusammenfassenden Einblick in die Forschungslandschaft wird auf die wissenschaftlichen Literaturübersichten [LIAN04], [TETI10], [ABEL10] und [KOVA11] verwiesen. [BYRN95], [PRIC99], [JANT02], [REHO05] und [ZHU09] ergänzen diese mit Zusammenfassungen zu den Entwicklungen hinsichtlich Werkzeugzustandsüberwachung.

Abhängig vom Ablauf lassen sich prozessunterbrechende bzw. intermittierende („offline“) und prozessbegleitende oder prozessparallele bzw. kontinuierliche („online“) Überwachungssysteme unterscheiden. Intermittierende Methoden führen die Überwachungsaufgabe in Nebenzeiten durch, wodurch Produktivitätsverluste durch verlängerte Nebenzeiten entstehen. Prozessbegleitende Systeme wirken diesem Nachteil entgegen, in dem diese den Zerspanprozess in der Hauptzeit, d. h. parallel zum laufenden Prozess überwachen. Die Begriffe In-Prozess (engl.: „in-process“) und „post-process“ beziehen sich auf den gesamten Fertigungsprozess als Einheit und unterscheiden somit nicht in Haupt- und Nebenzeiten. [KAEV05; SUWA99; TÖNS88]

Für die Entwicklung von Prozessüberwachungssystemen lassen sich drei grundsätzliche Aufgabenbereiche unterscheiden [CHO99; JEMI95; KAEV05] [KRÜG14; LIAN04; PRIC99; SUWA99; TETI10; TÖNS02; TÖNS88]:

- **Signalerfassung:** Auswahl geeigneter Sensorik für die Ermittlung des aktuellen Prozesszustands.
- **Signalverarbeitung und Merkmalsextraktion** (engl.: Signal Processing, Signal Conditioning, Feature Extraction): Aufbereitung der Sensorsignale durch Filterung, Kennwertberechnung für die Extraktion geeigneter Merkmale für die Diagnose des Prozesszustands.
- **Prozessbewertung und Entscheidungsfindung** (engl.: Decision Making, Action, Strategy): Auswertung und Analyse der Merkmale durch Modelle, Algorithmen oder anderweitig geeignete Strategien.

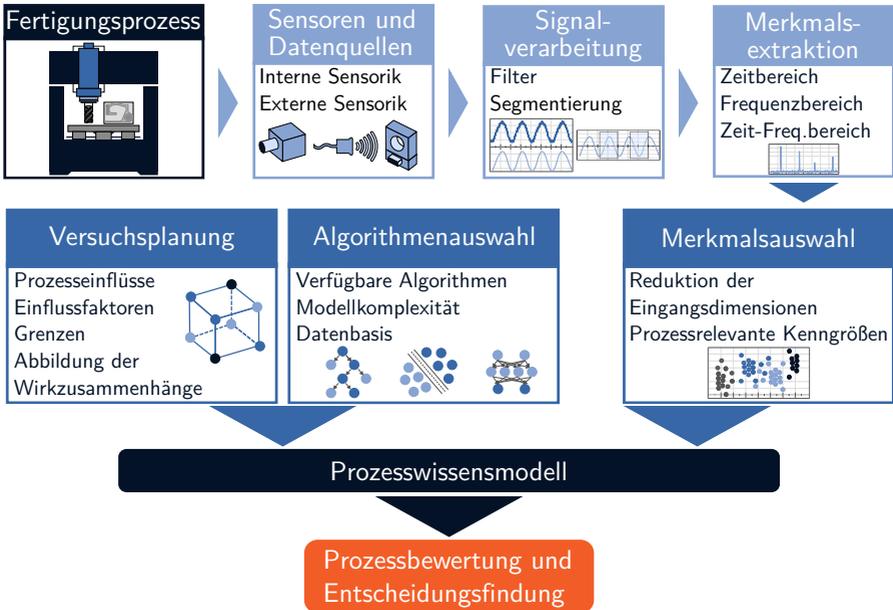


Abbildung 2.9: Grundlegende Vorgehensweise zur Entwicklung eines intelligenten Überwachungssystems für Bearbeitungsprozesse, eigene Darstellung basierend auf [ABEL10].

Infolge der steigenden Verbreitung von KI kommen neben den klassischen Ansätzen vermehrt Methoden und Algorithmen aus dem Bereich des ML in der Prozessüberwachung zum Einsatz. Prozessüberwachungssysteme können zusätzlich für die Datenaufnahme und intelligente Prozessbewertung eingesetzt werden. [KRÜG14] Für die Entwicklung kommen zusätzliche Aufgabenbereiche, wie eine effektive Versuchsplanung und die adäquate Modellauswahl für die intelligente Prozessüberwachung hinzu [ABEL10; KOVA11]. Abbildung 2.9 illustriert die erweiterte Vorgehensweise zur Entwicklung eines intelligenten Überwachungssystems nach [ABEL10].

Das Ziel beim Einsatz von Sensorik ist es, notwendige Informationen aus der Umgebung aufzunehmen, um Wissen über den Zustand des Bearbeitungsprozesses, der Maschine sowie über Wechselwirkungen zwischen Maschine und Prozess zu generieren [LITW11]. Bei der Auswahl und Platzierung der Sensorik ist darauf zu achten, den Prozess nicht zu beeinflussen und Signale aus dem Bearbeitungsprozess aufzunehmen, welche die Zustände möglichst

direkt wiedergeben [KRÜG14]. Neben der maschineninternen Sensorik (vgl. Kapitel 2.2.3) kommen häufig externe Sensoren zum Einsatz. Mit diesen Signalquellen lässt sich die vorhandene Ausstattung einer WZM erweitern, um Informationen aufzunehmen, die sich entweder nicht in den internen Signalen abbilden lassen oder der Zugriff auf interne Signale in ausreichender Menge und Abtastrate nicht vorhanden bzw. gewährleistet ist. Weitere Vorteile der prozessnahen Sensorintegration ergeben sich durch die Möglichkeit eine sehr große Bandbreite an Messgrößen abzubilden und es kommt im Vergleich zur internen Sensorik zu keinen elektrischen oder regelungstechnischen Einflüssen. [KAEV05; YOHA13] Typischerweise erfasste externe Messsignale sind Kraft- und Drehmomente, Körperschall, Schall, Beschleunigung zur Schwingungserfassung, Temperaturen, visuelle Signale mit Kameras und externe Sensoren zur Erfassung von Motorströmen und Leistungen [ABEL10; KAEV05; KOVA11; LIAN04; LITW11; REHS99; SUWA99; TETI10; YOHA13]. Die Eigenschaften von externen Sensoren sind neben den genannten Vorteilen mit erheblichen Nachteilen für den industriellen Einsatz verbunden:

- Integration, Installation und Inbetriebnahme der externen Sensorik führen zu hohen Kosten. Neben den direkten Kosten entstehen zusätzliche indirekte Kosten (Konstruktive Anpassungen für Sensorintegration, Signalübertragung). [KAEV05; YOHA13]
- Mögliche Einschränkung des Arbeitsraums, eingesetzten Werkzeug- und Werkstückspektrums, Be- und Entladevorgangs der Maschine durch die Integration der Sensorik [YOHA13].
- Mögliche Reduktion der Maschinensteifigkeit durch Integration der Sensorik im Kraftfluss der Maschine [KAEV05; KIRC92; YOHA13].
- Externe Sensorik ist unter Umständen anspruchsvollen Umgebungsbedingungen (Späne, Schmutz, mechanische Einflüsse, ...) ausgesetzt. Dadurch können Empfindlichkeit und Signalqualität leiden. [KAEV05]
- Hohe Aufwände bei der Zusammenführung mit Prozessinformationen. Besonders bei fehlenden Schnittstellen in der Maschine. [LITW11]

Die Signalverarbeitung und Merkmalsextraktion verfolgen das Ziel, Störanteile zu beseitigen und prozessrelevante Merkmale aus den erfassten Signalen zu extrahieren [KRÜG14; LITW11]. In diesen Schritten kommen analoge und digitale Filtermethoden zum Einsatz. Analoge Filter werden dazu eingesetzt, um hochfrequentes Rauschen und konstante Vorspannungen zu unterdrücken, das Signal innerhalb des Frequenzgangs des Sensors zu halten sowie eine Signalverzerrung durch Aliasing zu verhindern. Digitale Filter ermöglichen eine

zusätzliche Aufbereitung der Sensorsignale, um prozessrelevante Signalanteile herauszufiltern. Die anschließende Segmentierung der Signaldaten zielt darauf ab, die prozessrelevanten Signalabschnitte während des Eingriffs der Werkzeugschneiden zu erhalten. Nur diese Segmente enthalten die für die Überwachung notwendigen Informationen über die Wechselwirkungen zwischen Schneide und Werkstück. [ABEL10] Für die Wissensgenerierung aus den aufbereiteten Signalen bieten extrahierte Merkmale die Möglichkeit, die hohe Dimensionalität der Daten zu reduzieren und eine effiziente Repräsentation des Signals und der darin enthaltenen Informationen zu erhalten (vgl. Kapitel 2.2.5) [SCIE22; ZEBA20]. Die Merkmalsauswahl (engl.: Feature Selection) zielt darauf ab, die Menge an extrahierten Merkmalen auf eine Teilmenge an relevanten Merkmalen zu reduzieren. Die Grundlage dafür ist die Reduktion von Redundanzen und die Aussagekraft des einzelnen Merkmals in Bezug auf die Überwachungsaufgabe. [ABEL10; ZEBA20]

In der intelligenten Prozessüberwachung kommen KI-Modelle zum Einsatz, welche unter Nutzung von Trainingsdaten Wissensmodelle über den Prozess bilden und diese für die Prozessbewertung und Entscheidungsfindung nutzen. Die zur Generierung von Trainingsdaten notwendigen Versuchsreihen in der spanenden Fertigung sind jedoch kostenintensiv und zeitaufwendig, weshalb eine effektive Versuchsplanung essentiell ist, um den industriellen Einsatz eines KI-basierten Überwachungssystems zu ermöglichen. Dabei ist zu beachten, ein möglichst akkurates Abbild des Prozesses zu erreichen. Entscheidend ist neben der Versuchsplanung die passende Algorithmenauswahl, damit die Algorithmen in der Lage sind, die komplexen nicht linearen Zusammenhänge in den Daten zu erlernen und die resultierenden Wissensmodelle zuverlässige Vorhersagen liefern können. Die Zusammenführung der drei Schritte führt schließlich zu Prozesswissensmodellen, welche die zentrale Rolle im intelligenten Prozessüberwachungssystem einnehmen. [ABEL10; TETI10]

## 2.2.5 Verfahren zur Signalverarbeitung und Merkmalsextraktion

Das Ziel der Signalverarbeitung ist es, Merkmale bzw. Kennwerte oder geeignet aufbereitete Signale für die Entscheidungsfindung zur Verfügung zu stellen [KAEV05]. Aufgabe dabei ist, die Signalcharakteristik dimensionsreduziert in Form von Kennwerten, auch Merkmale (engl.: Features) genannt, möglichst aussagekräftig abzubilden, um diejenigen Merkmale zu identifizieren, welche mit dem aktuellen Prozesszustand korrelieren und dementsprechend die Entscheidungsfindung zulassen [LI06]. Entsprechend zielt die Merkmals-

extraktion darauf ab, die relevanten Informationen aus den Prozesssignalen herauszustellen und zu verdichten. Speziell im Hinblick auf Zerspanprozesse ist die Dimensionsreduktion der hochdimensionalen Zeitreihensignale für die Leistungsfähigkeit der Modelle von entscheidender Bedeutung. Je höher die Dimension in den Eingangsdaten für die Algorithmen, desto mehr Daten werden benötigt, um die Modellierungsaufgabe adäquat abbilden zu können und die Generalisierungsfähigkeit zu gewährleisten [KEOG17].

In diesem Zusammenhang wird vom „Fluch der Dimensionalität“ (engl.: Curse of Dimensionality) gesprochen [KEOG17]. Jedes zusätzliche Merkmal erhöht die Dimensionalität des Eingaberaums (engl.: Input Space), während jedes Trainingsbeispiel (Trainingsinstanz) lediglich die Menge an Datenpunkten im aufgespannten Eingaberaum erhöht. Bei einer festen Menge an Daten in Bezug auf die Anzahl an Instanzen erhöht sich die Komplexität für die Generalisierung exponentiell, wenn sich die Dimensionalität des Eingaberaums erhöht. Daraus lassen sich zwei Schlussfolgerungen ziehen: Zum einen kommt es zu einer verbesserten Generalisierungsfähigkeit der Modelle je größer die Anzahl an Instanzen in einem vorgegebenen Merkmalsraum ist. Zudem gilt es, lediglich Merkmale in der Analyse zu berücksichtigen, welche einen direkten Mehrwert bringen. Redundante oder nicht mit der Zielgröße korrelierende Merkmale sind demnach möglichst für die Modellbildung auszuschließen. [KEOG17; SARI21]

In der Signalverarbeitung lassen sich die Verfahren nach [LI06], [TET10] und [LAUR14] unterscheiden in:

- Verfahren im Zeitbereich
- Verfahren im Frequenzbereich
- Verfahren im Zeit-Frequenzbereich
- Weitere Verfahren

### **Verfahren im Zeitbereich**

Im Zeitbereich erfolgt die Merkmalsextraktion direkt ohne eine vorige Transformation der Zeitreihe. Üblicherweise werden statistische Kennwerte wie arithmetischer Mittelwert, Median, Maximum, Minimum, Standardabweichung, Schiefe, Kurtosis oder Effektivwert (engl.: Root Mean Square) (RMS) berechnet. Zusätzlich können spezielle Kennwerte wie Abstand der Extremwerte (Peak-to-Peak-Distanz), Crest-Faktor (Verhältnis des Maximums zum Mittelwert), Signalleistung, Permutationsentropie, Flächeninhalt unter der Kurve (engl.: Area under Curve) (AUC), Schwerpunkt (engl.: Centroid),

Nullstellen bzw. Schwellendurchgangsrate als Merkmale fungieren. Weitere Möglichkeiten ergeben sich durch den Einsatz von Korrelationsverfahren oder Verfahren zur Ähnlichkeitsbestimmung von Zeitreihen. Mithilfe der Autokorrelation wird ein Signal mit seiner zeitlich verschobenen Signalkopie gefaltet. Dies dient dazu innere Zusammenhänge im Signal aufzudecken. Durch Kreuzkorrelation lässt sich die Beziehung zwischen unterschiedlichen Signalen untersuchen. [BARA20; KAEV05; LI06; RMIL16; TETI10; ZHAN16] Aus dem Bereich der Ähnlichkeitsbestimmung ist das Dynamic Time Warping ein weit verbreitetes Verfahren für die Zeitreihenanalyse und wird häufig bei Klassifikationsaufgaben eingesetzt [HATA17; SENI08].

### **Analysen im Frequenzbereich**

Für Analysen im Frequenzbereich wird insbesondere die Fast-Fourier-Transformation (FFT) verwendet. Damit ist es möglich, das Leistungsdichtespektrum, Frequenzspektrum oder Amplitudenspektrum von Signalen zu bestimmen. Die FFT ermöglicht darüber hinaus die Cepstralanalyse, um Periodizitäten im Frequenzspektrum aufzuspüren. Für die Merkmalsberechnung werden statistische Kennwerte, wie Mittelwert, Median, Summe, Varianz bzw. Standardabweichung, Schiefe und Kurtosis des transformierten Zeitreihensignals oder weitere Kennwerte wie die spektrale Entropie erhoben. Zusätzlich dienen die dominanten Frequenzamplituden und relative Amplitudenpeaks als Eigenschaften der Zeitreihe. [BARA20; LAUR14; LI06; TETI10; ZHAN16]

### **Analysen im Zeit-Frequenzbereich**

Bei Anwendung der FFT erfolgt eine Mittlung der Frequenzzusammensetzung über die Dauer des Signals mit einer festen Auflösung des Frequenzspektrums. Für die Analyse von instationären Signalen lässt sich die FFT auf gefensterter Abschnitte des Signals anwenden. Die Short-Time-Fourier-Transformation (STFT) kommt zum Einsatz um Frequenzen und deren Amplituden auch zeitlich auflösen zu können. Bei Anwendung der STFT zusammen mit der chronologischen Anordnung der ermittelten Spektren in den jeweiligen Fensterabschnitten resultieren Spektrogramme. Diese dienen als visuelle Darstellung der Amplituden und Frequenzen in Abhängigkeit der Zeit. Die Wahl der Fensterbreite entscheidet bei der STFT über die Zeit- als auch Frequenzauflösung. Dies hat zur Folge, dass entweder eine hohe zeitliche oder eine hohe Frequenzauflösung vorliegt. Diese Unschärfe zwischen Zeit- und Frequenzauflösung lässt sich durch die Heisenbergsche Unschärferelation erklären. Um dieser Einschränkung zu begegnen wurde die Wavelet-Transformation entwickelt. Als Basis dienen Wavelets (mathematische Funktionen, die Sinus- und Kosinus-

funktionen ähneln), welche für die Untersuchung der enthaltenen Frequenzbestandteile gestaucht oder gedehnt werden können. Für eine hohe zeitliche und spektrale Auflösung dienen angepasste Fensterbreiten je nach Höhe der Frequenzanteile. Neben der kontinuierlichen Wavelet-Transformation, kommt die diskrete Wavelet-Transformation mit eingeschränkten Zeit- und Skalenparametern zum Einsatz, um die Rechenintensität für die Transformation zu reduzieren. Typisch verwendete Merkmale sind die ermittelten Wavelet-Koeffizienten selbst oder darauf berechnete Kennwerte wie Mittelwert, Standardabweichung bzw. Varianz, Kurtosis, Crest-Faktor, Peak-to-Peak-Distanz oder der Effektivwert. Weitere Methoden im Zeit-Frequenzbereich liefern die Wavelet-Packet-Transformation und die Hilbert-Huang-Transformation. [BASS07; CAO13; ESSE01; HUAN98; KALV10; LI06; MERT13; PENG06; SCHL05; SUSA18; TETI10]

### Weitere Verfahren

Als weiteres Verfahren für die Zeitreihenanalyse liefert die Singular Spectrum Analysis, welche eine Zeitreihe in drei unabhängige Signalkomponenten zerlegt: Trend (lokaler Mittelwert des Signals), Differenz zwischen ursprünglichem Signal und dem Trend (detrendiertes Signal), und ein strukturloses Rauschen ohne latente Struktur [TETI10]. Zusätzlich lassen sich Zeitreihen in Bilddaten transformieren, um diese beispielsweise in DL-Algorithmen für Klassifikationsaufgaben einzusetzen. Beispiele hierfür sind Gramian Angular Fields, Markov Transition Fields und Recurrence Plots. [HATA17; WANG15]

## 2.2.6 Aktuelle Ansätze und industrielle Lösungen zur Prozessüberwachung

In der Forschungslandschaft existieren bereits einige Ansätze für die Umsetzung von Prozessüberwachungslösungen. Vor allem wird die Nutzung von maschineninternen Signaldaten in den Forschungsarbeiten zur Detektion von Prozessfehlern untersucht. REHSE untersucht Methoden für die Werkzeugüberwachung bei Bohr- und Fräsprozessen. Neben der Werkzeugbruchüberwachung von Bohr- und Schaftfräswerkzeugen auf Basis interner Signaldaten, realisiert REHSE mithilfe einer Differenzbildung zwischen gemessenen Kraftsignalen aus externer Sensorik und berechneten Verläufen aus einem Kraftmodell nach Kienzle, die Prozessbewertung beim Fräsen mit Messerkopfstirnfräsern. Detektierte Prozessfehler sind Schneidenbrüche, fehlende Schneiden, radialer Schneidenversatz und Werkzeugverschleiß. [REHS99] SUWALSKI stellt ein Prozessüberwachungssystem auf Basis interner Signaldaten für Drehma-

schinen vor, welches die Detektion von Werkzeugverschleiß und Werkzeugbrüchen ermöglicht [SUWA99]. REUBER untersucht Methoden zur Überwachung von Schlichtfräsprozessen für die Zerspanung von Freiformflächen. Mit Hilfe eines spindelintegrierten Kraftmessrings und eines Körperschallsensors am Werkstück können Kollisionen, Überlastsituationen, Werkzeugbrüche und der Werkzeugverschleiß überwacht werden. [REUB01] KAEVER erweitert diese Ansätze hin zu einem steuerungsintegrierten Überwachungssystem für die spanende Fertigung. Grundlage für das sensorlose Konzept bilden steuerungsintegriert korrigierte Signale (Beschleunigungs-, Gleitreibungs- und Stillstandskorrektur), welche je nach Aufgabe mit festen Grenzen oder mitlaufenden Schwellen überwacht werden. [KAEV05] BRINKHAUS stellt, basierend auf statistischen Verfahren, ein Überwachungsverfahren zum Anlernen von Hüllkurven vor, welches sich insbesondere für Serienprozesse eignet [BRIN09]. Für die Realisierung von Überwachungsaufgaben für die Kleinserien- bzw. Einzelteilerfertigung nutzt SCHMIDT eine einflussgrößensensitive Prozesskraftsimulation für die prozessspezifische Parametrierung eines Überwachungssystems [SCHM11]. YOHANNES geht einen Schritt weiter und entwickelt ein weitgehend anlernfreies Überwachungssystem für die Kleinserienfertigung. Die Parametrierung erfolgt basierend auf online ermittelten Kontaktbedingungen zwischen Werkzeug und Werkstück. [YOHA13] LITWINSKI nutzt ein eigens entwickeltes sensorisches Spannsystem zur Umsetzung eines Überwachungssystems für die Einzelteilerfertigung. Damit ist es möglich Prozesskräfte, Beschleunigungen und Temperaturen zu erfassen. Die Überwachung wird durch die Kombination einer Materialabtragssimulation mit den Sensorsignalen aus dem Spannsystem unter der Nutzung von Methoden aus der statistischen Prozesskontrolle umgesetzt. [LITW11] Die vorgestellten Ansätze fokussieren sich auf Werkzeugfehler, Kollisionen, Rattern oder Fehler in Werkstücken, wie beispielsweise Fehlstellen in Form von Bohrungen.

Neben den forschungsseitigen Ansätzen verbreiten sich zunehmend industrielle Prozessüberwachungslösungen, welche interne Signaldaten für die Überwachung nutzen. Beispiele dafür sind das *ToolScope* [CERA22] vom Unternehmen Ceratizit, *CTM* [MARP22a] und *Genior Modular* [MARP22b] vom Unternehmen Marposs Monitoring Solution und *Liveinspect* [MCU 22] vom Unternehmen MCU.

Zusammenfassend lässt sich festhalten, dass auf dem Feld der Prozessüberwachung für die spanende Fertigung mit WZM in den letzten Jahrzehnten intensiv geforscht wurde. Die Nutzung von internen Signaldaten spielt dabei aufgrund der damit verbundenen Vorteile gegenüber externer Sensorik (vgl. Kapitel 2.2.4) eine immer wichtigere Rolle. Begünstigt wird diese Entwicklung

durch die offenen Steuerungssysteme, welche die Grundlage für die Überwachung mittels interner Signaldaten bilden. Der überwiegende Forschungsanteil liegt in der Werkzeugüberwachung und Prozessfehlerdetektion. Dabei wird in der Regel davon ausgegangen, dass die von den Überwachungssystemen als fehlerfrei deklarierten Prozesse dementsprechend Gutteile, welche die Qualitätsanforderungen erfüllen, hervorbringen. Die existierenden Ansätze sind basierend auf typischen Fehlerausprägungen erprobt und liefern sehr gute Ergebnisse. Dennoch existieren zahlreiche weitere Randbedingungen, welche die Maßhaltigkeit der gefertigten Werkstücke beeinflussen. Deshalb richten sich aktuelle Forschungsarbeiten von der Prozessüberwachung hin zur Qualitätsüberwachung aus. Ziel dieser Arbeiten ist es, Ansätze zu entwickeln, die es ermöglichen die gefertigte Qualität der Bauteile möglichst online und prozessparallel zu bestimmen. Für ein verbessertes Verständnis eines In-Prozess-Qualitätsüberwachungssystems dieser Art muss jedoch zunächst der Qualitätsbegriff im Kontext der Zerspanung erläutert werden.

## 2.3 Werkstückqualität in der spanenden Fertigung

Dieses Kapitel definiert die Werkstückqualität aus Sicht der spanenden Fertigung. Neben der allgemeinen Einführung in den Qualitätsbegriff werden in Kapitel 2.3.2 die Einflussgrößen auf die resultierende Werkstückqualität diskutiert.

### 2.3.1 Werkstückqualität und -prüfung

Gemäß DIN EN ISO 9000 ist unter Qualität der „Grad, in dem ein Satz inhärenter Merkmale eines Objekts Anforderungen erfüllt“ zu verstehen [ISO9000]. Bezogen auf ein Produkt definiert sich Qualität als Erfüllung von zuvor festgelegten und vorausgesetzten Anforderungen. Ein Produkt erfüllt die Qualität, wenn die Werte sämtlicher ihm innewohnender (inhärenter) Merkmale sich eignen, um die gegebenen Anforderungen zu erfüllen. [ISO9000; PAUC08] Qualität bezieht sich demnach auf die Übereinstimmung der Produkteigenschaften mit den vorgegebenen Forderungen (vgl. Abbildung 2.10). Für eine Klassifizierung der Qualität eines Produkts gilt es, den Erfüllungsgrad sämtlicher Einzelanforderungen zu bestimmen bzw. zu messen. Je nach Grad der Übereinstimmung wird von „guter“ (Produktmerkmale erfüllen alle Einzelanforderungen) oder „schlechter“ (nicht jedes Merkmal erfüllt die entsprechende Einzelanforderung) Qualität gesprochen. [BRÜG20]



Abbildung 2.10: Qualität als Grad der Übereinstimmung zwischen vorgegebenen Forderungen und dem Ergebnis, angelehnt an [BRÜG20].

Bezogen auf die spanende Fertigung entspricht die Erfüllung der Anforderungen allgemein die Herstellung von funktionsfähigen Bauteilen. Die Funktionsfähigkeit eines Bauteils charakterisiert sich über Oberflächen- und Randzoneigenschaften, welche aus fertigungstechnischer Sicht durch die Abweichungen des Werkstücks von der vorgegebenen Idealform ausgeprägt sind. Diese geometrischen Abweichungen von der idealen Oberfläche werden unter der Werkstückqualität zusammengefasst. Dabei lassen sich Maß-, Form-, Lage- und Rauheitsabweichungen sowie physikalische Randzoneigenschaften unterscheiden. [DENK11] Zahlreiche zufällige oder systematische Störgrößen auf den Fertigungsprozess resultieren in schwankenden Merkmalsausprägungen und beeinflussen damit die Produktqualität in Form von Gestaltabweichungen der Werkstückgestalt. Insbesondere nicht vorhersagbare, zufällige Störgrößen machen es erforderlich, Qualitätsmerkmale in der Fertigung zu prüfen. Je nach Verwendungszweck des Produkts kommen dabei Stichproben- oder 100 %-Prüfungen zum Einsatz. Vorgeschrieben ist die 100 %-Prüfung bei sicherheitskritischen Bauteilen. Zusätzlich wird eine 100 %-Prüfung bei hohen Fehleranteilen oder Bauteilen mit hoher wirtschaftlicher Bedeutung durchgeführt. [BRÜG20; KLOC18] Um die Produktqualität kontinuierlich zu überwachen, setzen Unternehmen Methoden zur statistischen Prozesssteuerung bzw. -lenkung ein. Das Ziel dabei ist, mittels der Aufnahme von qualitätsrelevanten Merkmalen, etwaige Fehler und Abweichungen frühzeitig zu erkennen, um darauf aufbauend korrigierend in den Prozess einzugreifen. In der Fertigung werden normalerweise sehr niedrige Fehlerquoten erreicht bzw. gefordert, weshalb eine kontinuierliche Qualitätsüberwachung meist nur über die 100 %-Prüfung bzw. durch Prüfstrategien über den gesamten Herstellungsprozess zu erreichen ist. Stichprobenprüfungen können diesen Qualitätsprüfanforderungen dementsprechend nicht gerecht werden. [BRÜG20] Die Qualität des Fertigungsergebnisses stellt somit eine grundlegende Bewertungsgröße für Abläufe in der Produktion dar. Eine wesentliche Herausforderung dabei ist

die Reduktion der notwendigen Aufwände für die Qualitätsprüfung und die Bereitstellung der Zustandsinformationen von 100 % der Bauteile zu jedem Zeitpunkt der Fertigung. [KRÜG14]

Eine aufwandsarme Realisierung einer 100 %-Prüfung der Werkstückqualität lässt sich durch In-Prozess-Messung der Qualitätsmerkmale umsetzen. Im Gegensatz zur Verwendung in der Prozessüberwachung bezeichnet In-Prozess-Messtechnik, die hauptzeitparallele Messung in der WZM während des Produktionsprozesses. In-Situ-Messtechnik misst ebenfalls in der WZM jedoch im Gegensatz zur In-Prozess-Messtechnik nicht hauptzeitparallel. Dazu wird der Produktionsprozess angehalten, um die Messung in der WZM durchzuführen. Für die In-Situ-Messung von Qualitätsmerkmalen kommen bei einfachen Messaufgaben maschinenintegrierte Messtaster zum Einsatz. Zusätzlich lassen sich Maßverkörperungen, Längenaufnehmer und Messuhren mittels manueller Messung durch die Werker einsetzen. Der große Nachteil dabei ist, dass die Qualitätsmerkmale nach dem Zerspanprozess, nicht prozessbegleitend geprüft werden. Zudem besteht die Möglichkeit, dass gegebene Prozessrandbedingungen eine In-Situ-Messung unmöglich machen. [MARX21] Für die direkte In-Prozess-Erfassung von Form- und Lageabweichungen existieren nach aktuellem Stand noch keine praxistauglichen Ansätze [KRÜG14].

Jedoch steckt in der In-Prozess-Messung ein hohes Potenzial für eine aufwandsarme kontinuierliche Überwachung der Werkstückqualität. Deshalb werden in den folgenden Kapiteln zunächst die relevanten Einflüsse auf die Werkstückqualität (Kapitel 2.3.2) vorgestellt und darauf aufbauend erste Forschungsansätze betrachtet, welche versuchen auf Basis von erfassten Prozessgrößen Qualitätsmerkmale indirekt abzuschätzen (Kapitel 2.4).

### 2.3.2 Einflüsse auf die Werkstückqualität

Der Zerspanprozess basiert auf dem Zusammenwirken von mechanischen, thermischen sowie chemischen Vorgängen und lässt sich als System mit multi-dimensionalen Eingangs- und Ausgangsgrößen beschreiben [AUER15] [DENK11]. PAUCKSCH, DENKENA und KLOCKE vertiefen die technologischen Zusammenhänge während des Zerspanprozesses in deren Fachliteratur. Diese sind für das weitere Verständnis in dieser Arbeit nicht erforderlich, weshalb für eine detaillierte Betrachtung auf die Werke [PAUC08], [DENK11] und [KLOC18] verwiesen wird. Die eingangsseitigen System- und Stellgrößen wirken von außen auf das System ein und beschreiben die Ursachen für die im Prozess ablaufenden Vorgänge [KLOC18; LUNZ20]. Während die Systemgrößen Abhängigkeiten von Maschine, Werkzeug und Werkstück aufweisen,

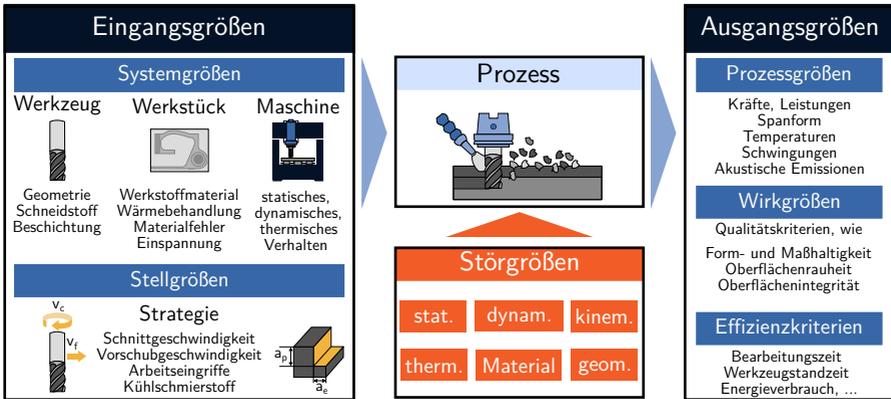


Abbildung 2.11: Der Zerspanprozess als Systemdarstellung, eigene Darstellung basierend auf [AUER15; DENK11].

repräsentieren die Stellgrößen die gewählte Prozessstrategie. Ausgangsseitig lassen sich Prozessgrößen wie Kräfte, Temperaturen, Spanform und Spanbildung, Schwingungen und Leistungen, Wirkgrößen in Form von Qualitätskriterien, wie Form- und Maßhaltigkeit des Werkstücks, Oberflächenrauheit und Oberflächenintegrität sowie Effizienzkriterien, wie Bearbeitungszeit, Werkzeugstandzeit und -standweg, Energieverbrauch, etc. unterscheiden. Neben den Input- und Outputgrößen unterliegt der Prozess zufälligen oder systematischen Störgrößen, die den Systemausgang beeinflussen und demnach basierend auf den Prozessgrößen Auswirkungen auf die Werkstückgestalt nehmen. [AUER15; AUER18; KLOC18] Störgrößen auf das System lassen sich nach AUERBACH in statische, dynamische, thermische und kinematische Störeinflüsse sowie Materialschwankungen und geometrische Abweichungen einteilen. Beispiele dafür sind Aufmaßschwankungen, Ungenauigkeiten in der Werkzeugherstellung, äußere Schwingungsanregungen oder schwankende Umgebungstemperaturen. [AUER15; HEIS92] Abbildung 2.11 fasst die beschriebenen Einflüsse auf den Zerspanprozess als Systemdarstellung zusammen. Die komplexen Wirkzusammenhänge und die zahlreichen Einflussgrößen führen bei spanend gefertigten Flächen zu geometrischen Abweichungen von der idealen Sollgeometrie, welche zusammengefasst als Gestaltabweichungen bezeichnet werden. [JORD20; KLOC18] Die entstehenden Gestaltabweichungen lassen sich nach DIN 4760 in sechs Gruppen einteilen (vgl. Abbildung 2.12).

Gestaltabweichungen der ersten Ordnung resultieren in makroskopischen Abweichungen und sind für die Tolerierung von Geometrieelementen in DIN EN

Gestaltabweichung	Art der Abweichung	Mögliche Entstehungsursachen
1. Ordnung: Formabweichungen 	Geradheits-, Ebenheits-, Rundheits-, Profil-, Zylinderformabweichung, u.a. (siehe DIN EN ISO 1101 / 5458)	Fehler in den Führungen der WZM, Durchbiegung der Maschine oder des Werkstückes, Härteverzug, Werkzeugabdrängung, Verschleiß
2. Ordnung: Welligkeit 	Wellen (siehe DIN 4761 / DIN EN ISO 4287)	Fehlerhafte Einspannung, Form- oder Laufabweichungen eines Fräsers, Schwingungen der WZM oder des Werkzeugs
3. Ordnung: Rauheit 	Rillen (siehe DIN 4761 / DIN EN ISO 4287)	Form der Werkzeugschneide, Vorschub oder Zustellung des Werkzeugs
4. Ordnung: Rauheit 	Riefen, Schuppen, Kuppen (siehe DIN 4761)	Vorgang der Spanbildung (Reißspan, Scherspan, Aufbau-schneide)
5. Ordnung: Rauheit Anmerkung: nicht mehr in einfacher Weise bildlich darstellbar	Gefügestruktur (siehe DIN 4761)	Kristallisationsvorgänge, Veränderung der Oberfläche durch chemische Einwirkung, Korrosionsvorgänge
6. Ordnung: Anmerkung: nicht mehr in einfacher Weise bildlich darstellbar	Gitteraufbau des Werkstoffs (siehe DIN 4761)	

Abbildung 2.12: Ordnungssystem für Gestaltabweichungen, angelehnt an [DIN4760].

ISO 1101 und DIN EN ISO 5458 nach Form- und Lagetoleranzen genormt. Für die Begrenzung der Formabweichungen einzelner Geometrielemente dienen Formtoleranzen, die sich in Geradheit, Ebenheit, Rundheit und Zylinderform untergliedern. Zusätzlich können Profile von beliebigen Linien oder Flächen toleriert werden. Lagetoleranzen dienen der Beschreibung des funktionalen Zusammenwirkens von Geometrieelementen bzw. der Tolerierung von Positionsabweichungen von Geometrieelementen gegenüber einer Referenz. Diese sind in Richtungs- (Parallelität, Rechtwinkligkeit, Neigung), Orts- (Position, Konzentrität, Koaxialität, Symmetrie) und Lauftoleranzen (Rundlauf, Planlauf, Lauf in vorgegebener Richtung, Gesamtrundlauf, Gesamtplanlauf) eingeteilt. Lauftoleranzen finden ihren Einsatz überwiegend bei rotationssymmetrischen Bauteilen. [ISO1101; ISO5458; KLOC18; WECK12] Die zugrundeliegenden Entstehungsursachen liegen meistens in systembedingten Fehlern, wie elastischen oder thermischen Verformungen von Maschinenführung, Werkstück oder Werkzeug, unsachgemäßer Aufspannung oder Ausrichtung des Werkstückes, Härteverzug oder Werkzeugverschleiß. Die Ermittlung von

Gestaltabweichungen erster Ordnung erfolgt durch Betrachtung der gesamten Ist-Oberfläche eines Formelements. [DENK11; DIN4760; KLOC18].

Welligkeiten als Gestaltabweichungen zweiter Ordnung charakterisieren sich durch überwiegend periodisch auftretende Abweichungen der gefertigten Ist-Oberfläche. Die Rauheit einer Oberfläche lässt sich durch Gestaltabweichungen der 3.–5. Ordnung erfassen. Welligkeiten und Rauheiten werden im Gegensatz zu den Gestaltabweichungen erster Ordnung anhand von repräsentativen Ausschnitten der Werkstückoberfläche beurteilt. In der Regel weisen reale Oberflächen eine Überlagerung aller Gestaltabweichungen auf. Dominierend in der spanenden Fertigung sind die Gestaltabweichungen der 1.–4. Ordnung. [DIN4760; KLOC18; PAUC08]

## 2.4 Datenbasierte Bewertung der Werkstückgestalt bei Zerspanprozessen

Für die Bewertung der gefertigten Oberflächenform kommt zum heutigen Zeitpunkt hauptsächlich spezialisierte Fertigungsmesstechnik zum Einsatz. Diese dient zur Messung und Prüfung der resultierenden Gestaltabweichungen von gefertigten Oberflächen. [MARX21] Zahlreiche Forschungsarbeiten widmen sich der Bewertung der Werkstückgestalt ohne zusätzliche Messtechnik parallel zum Prozess. Dies birgt Potenziale in kürzeren Reaktionszeiten zwischen Fehlerdetektion und Fehlerkorrektur, frühzeitiger Erkennung bzw. Vermeidung von Ausschuss und damit insgesamt der Reduktion von Produktionskosten [BREC19a; BREC19b]. In diesem Zusammenhang unterscheiden BENARDOS und VOSNIAKOS mit dem Fokus auf die Prädiktion der Oberflächenrauheit zerspanungstheoretische Ansätze als Grundlage analytischer Modelle zur Repräsentation der Oberfläche, experimentbasierte Ansätze, welche die Auswirkungen von verschiedenen Faktoren auf Basis experimenteller Untersuchungen analysieren und KI-basierte Ansätze, die auf Basis von Prozessdaten Zusammenhänge zwischen Prozessgrößen und Oberflächenabweichungen modellieren [BENA03]. Diese Einteilung lässt sich allgemeingültig auf die weiteren Ordnungen der Gestaltabweichungen übertragen, weshalb im Folgenden die aktuellen Forschungsarbeiten basierend auf der genannten Eingruppierung vorgestellt werden.

### Zerspanungstheoretische Ansätze

Für die Simulation der entstehenden Oberflächentopographie dienen analytisch erstellte geometrische Modelle, welche auf der Modellierung zerspa-

nungstheoretischer Aspekte, wie Prozesskinematik, Werkzeugeigenschaften und Spanbildungsmechanismen basieren [BENA03]. KLINE sieht als Hauptursache für Abweichungen in der Oberflächenausprägung die während des Zerspanprozesses entstehenden Zerspankräfte. Diese führen hauptsächlich zu Abdrängungen des Werkzeugs und zu Verformungen bei dünnwandigen Werkstücken, wodurch sich Abweichungen zur gewünschten Sollgeometrie ergeben. Zusätzlich können thermisch induzierte Verformungen am Werkstück entstehen. [KLIN82; KRÜG14] Grundlage für die Zerspankraftmodellierung bildet die Bestimmung des Spannungsquerschnitts, welcher proportional in die Schnittkraftberechnung eingeht. Die Absolutwertbestimmung der Kräfte stützt sich dabei auf empirisch gewonnene Zerspankraftkonstanten. [DESE70; GEY03; KAMM77; KIEN52; KLIN82; VICT56] KRÜGER fasst die unterschiedlichen Ansätze zur Simulation der Werkzeugabdrängung zusammen. Dafür können Steifigkeitskennwerte zum einen experimentell ermittelt und in die entsprechenden Systemgleichungen für Ersatzfedersteifigkeiten eingesetzt werden. Die Balkentheorie liefert einen weiteren Ansatz zur Ermittlung der Werkzeugsteifigkeit. Neben den analytischen Modellen ist es möglich, die Werkzeugabdrängung mit diskreten numerischen Modellen auf Basis der Finite-Element-Methode (FEM) abzubilden. [KRÜG14] Weitere Forschungsarbeiten zur analytischen Modellierung der Werkzeugabdrängung beim Schaftfräsen werden in [BUDA95], [WANG03], [LARU03], [RYU12] und [HASE18] vorgestellt. Für die Bestimmung der mechanisch und thermisch bedingten Verformung werden FEM-basierte numerische Verfahren verwendet. [DENK12; KLIN82; KRÜG13; KRÜG14] Detaillierte Modellierungsansätze beziehen die Spannsituation, Steifigkeiten von Maschine und Komponenten sowie erzwungene Schwingungen mit ein [LIAO01]. Nachteil dieser Methoden ist die a-priori Simulation, wodurch sich Störeinflüsse während des Prozesses nicht abbilden lassen. Um diese berücksichtigen zu können, werden Prozessinformationen aus Sensordaten benötigt, welche möglichst prozessnah die aktuellen Prozessbedingungen wiedergeben. KRÜGER und DENKNA entwickeln eine Methodik zur Online-Bewertung von Fräsprozessen auf Basis von externer Kraftsensorik unter Einbindung strukturmechanischer FEM-Werkzeugmodelle. Die realisierte Online-Überwachung der Werkzeugabdrängung ermöglicht Rückschlüsse auf die Güte und Formgenauigkeit der gefertigten Oberfläche. [DENK12; KRÜG13; KRÜG14] Um den Nachteilen externer Sensorik entgegen zu wirken, stellen KÖNIGS und BRECHER ein Verfahren zur virtuellen Qualitätsabschätzung mittels spindelintegrierter Kraftsensorik vor. Interne Encodersignale liefern dafür die aktuelle Werkzeugposition relativ zum Werkstück. [BREC19a; BREC19b; KÖNI18]

## Experimentelle Ansätze

Experimentelle Analysen untersuchen die Wirkungen der relevanten Einflussfaktoren auf die Qualitätseigenschaften. Jeder Faktor wird hinsichtlich dessen Einflussmechanismus auf das beobachtete Qualitätsmerkmal hin analysiert. Häufig kommen dafür Regressionsanalysen zum Einsatz, um auf Grundlage der experimentellen Ergebnisse eine Prozessmodellierung durchzuführen. Bei den untersuchten Eingangsgrößen handelt es sich vorwiegend um variierte Technologieparameter in Form von Schnitt- und Vorschubgeschwindigkeiten sowie Eingriffsgrößen. Zudem untersuchen verschiedene Forschungsarbeiten den Einfluss des voranschreitenden Werkzeugverschleißes auf die erzeugte Oberflächenrauheit. [BENA03] Neben der Untersuchung von technologischen Einflussgrößen untersuchen BEGGAN, GUO, MARINESCU und LI den Informationsgehalt von während des Zerspanprozesses extern aufgezeichneten Körperschallemissionen (engl.: Acoustic Emission) (AE) hinsichtlich erreichter Oberflächenqualität. Mithilfe von Methoden aus dem Zeit-Frequenzbereich lassen sich korrelierende Merkmale in Form spezifischer Frequenzmuster identifizieren sowie Informationen zur Energieverteilung gewinnen, welche Rückschlüsse auf die gefertigte Qualität zulassen. [BEGG99; GUO05; LI18; MARI08; MARI09] CHEN und BARTON nutzen als weitere Möglichkeit externe Schwingungssensorik für die Bestimmung der Oberflächenrauheit [BART21; CHEN12].

## Ansätze mittels künstlicher Intelligenz

Aktuelle KI-basierte Ansätze für die Bewertung der Werkstückoberfläche lassen sich gemäß den genutzten Eingangsdaten in zwei Gruppen einteilen. Die erste Gruppe von Forschungsarbeiten nutzt variierte Technologieparameter als Eingangsgrößen für das Modelltraining. LO, HO, KOVAC, KANT und VASANTH zeigen Ansätze basierend auf verschiedenen neuronalen Netzwerkarchitekturen zur Bestimmung der Oberflächenrauheit. Die Eingangsgrößen dabei sind prozessbeschreibende Technologieparameter wie die Schnittgeschwindigkeit  $v_c$ , Vorschubgeschwindigkeit  $v_f$ , Schnitttiefe  $a_p$  sowie je nach Ansatz die aktuelle Verschleißmarkenbreite VB am Werkzeug, der Werkstofftyp und die binäre Variable zur Nutzung von Kühlschmierstoff. [HO09; KANT15; KOVA13; LO03; VASA20] ELANGOVAN und KHORASANI ergänzen die statischen Eingangsparameter zusätzlich mit statistischen Kenngrößen von während des Prozesses aufgezeichneten Schwingungssignalen [ELAN15; KHOR17]. LI, ARNAIZ-GONZÁLES und DENKENA nutzen ähnliche Ansätze für die Vorhersage der makroskopischen Formabweichung beim Fräsen. Neben den angesprochenen Technologieparametern nutzt DENKENA zusätz-

lich die Materialabtragsrate und den projizierten Abstand zwischen Oberflächenpunkt und dem Werkzeugmittelpunkt (engl.: Tool Center Point) (TCP) entlang der Werkzeugachse. Diese werden aus einer Materialabtragssimulation gewonnen. Untersuchte Fertigungsverfahren sind Fräsen mit Schaft-, Kugelkopf- oder Planfräsern sowie Drehen. [ARNA16; DENK20; LI04]

Die Nutzung von Technologieparametern als Eingangsgrößen für die Vorhersage der Qualität deckt lediglich die Abhängigkeiten zwischen Stellgrößen und Wirkgrößen ab. Das Ergebnis des Prozesses hängt jedoch zusätzlich von während des Prozesses wirkenden Stör- und Einflussgrößen ab, welche Einfluss auf die resultierenden Prozessgrößen nehmen (vgl. Kapitel 2.3.2). Um diese in der Modellbildung zu berücksichtigen bringt die Forschungslandschaft erste Ansätze hervor, welche Prozessgrößen in Form von hochdimensionalen multivariaten Zeitreihen verarbeiten. MATHEWS und NETO trainieren Neuronale Netze mit Kräften, AE, Beschleunigungssignalen sowie mit Strom-, elektrischen Spannungs- und Leistungssignalen zur Vorhersage der Rundheit und des Durchmessers beim Bohren und Reiben [MATH99; NETO13]. Um das Potenzial bereits verbauter Sensorik herauszustellen, nutzen die Arbeiten von SCHUH, SCHORR und ZIEGENBEIN maschineninterne Signaldaten für Prädiktion der Form- und Lageabweichungen von gebohrten und geriebenen Bohrungen. Zu beobachten ist hierbei neben der Nutzung von neuronalen Netzen zusätzlich der Vergleich zu Algorithmen aus dem Bereich des ML. [SCHO20a; SCHO20b; SCHO20c; SCHO21a; SCHO21b; SCHU19; ZIEG20] BRECHER zeigt einen Ansatz zur Vorhersage eines kontinuierlichen Verlagerungsvektors bei Schaftfräsprozessen, welcher die Verlagerung zwischen Werkzeug und Werkstück widerspiegelt. Dazu werden maschineninterne Positions-, Strom-, und Spindeldrehzahl-signale genutzt, um ein neuronales Netz zu trainieren, welches den Verlagerungsvektor für jeden Interpolationsschritt prädiziert. Basierend auf dem Verlagerungsvektor lassen sich Geradheiten und Rundheiten (bei Kreisfräsprozessen) berechnen. Die Ergebnisse zeigen, dass dieses Verfahren bei Geradheiten deutlich bessere Vorhersagen liefert als bei Rundheiten. [BREC19c; BREC20b]

## 2.5 Zwischenfazit und Forschungsbedarf

Der Stand des Wissens zeigt die große Bandbreite an möglichen Zielgrößen von Prozessüberwachungslösungen in der spanenden Fertigung. Je nach Überwachungsaufgabe sind Entwicklungen in Richtung des Condition Monitoring zur Überwachung der Maschinenkomponenten und zum Process Monitoring mit dem Schwerpunkt auf dem Fertigungsprozess zu beobachten. Gemeinsames Ziel dabei ist die größtmögliche Transparenz über die aktuellen Zustände und Randbedingungen während des Prozesses, um bei auftretenden Störungen oder Abweichungen ein rechtzeitiges Eingreifen in den Produktionsablauf zu ermöglichen. Aktuelle Lösungen nutzen für die Ermittlung des Prozesszustands neben zusätzlicher Sensorik zunehmend bereits integrierte Maschinensensorik. Insbesondere die mit interner Sensorik verbundenen Vorteile (vgl. Kapitel 2.2.4) sorgen für die zunehmende Verbreitung von maschineninternen Signalen in Überwachungslösungen für Anwendungen in Forschung und Industrie. Hinzu kommt die Offenheit von aktuellen Steuerungssystemen, welche den direkten Zugriff auf prozessnahe NC-Signaldaten aus der Steuerung und dem Lageregelkreis ermöglicht. Diese Entwicklungen führen zu bereits existierenden steuerungsintegrierten Methoden und Lösungen zur Überwachung von Serienprozessen. Prozessmodelle ergänzen die reine Signalverarbeitung, um anlernfreie Prozessüberwachungssysteme zu realisieren. Zusammenfassend liefern diese klassischen Überwachungsansätze gute Ergebnisse hinsichtlich typischer Fehlerausprägungen, wie Werkzeugbrüche, Kollisionen oder Fehler in Werkstücken.

Die zahlreichen Einfluss- und Störgrößen auf den Zerspanprozess und die komplexen Wirkzusammenhänge (vgl. Kapitel 2.3.2) können jedoch auch bei Prozessen ohne die erwähnten Fehlerausprägungen zu Gestaltabweichungen an den gefertigten Oberflächen führen und damit zu einer möglichen Überschreitung der für die Funktionalität des Bauteils geforderten Toleranzen beitragen. Für die Erhöhung der Prozesstransparenz und die Reduktion von Kosten für die Qualitätsprüfung untersuchen aktuelle Forschungsansätze Möglichkeiten, die resultierende Werkstückqualität online und prozessparallel zu bestimmen. Zerspanungstheoretische Ansätze liefern dafür zunehmend leistungsfähigere physikalische Modelle für die Abbildung der vorherrschenden Prozessbedingungen und deren resultierenden Ausprägungen auf die gefertigten Werkstückoberflächen. Höhere Modellgenauigkeiten gehen mit erhöhter Modellkomplexität und damit zunehmender Berechnungszeit einher. Zudem lassen sich trotz immer höherer zur Verfügung stehenden Rechenleistung nicht alle Randbedingungen in physikalischen Prozessmodellen berücksichtigen. Expe-

rimentelle Ansätze dienen dazu, die Auswirkungen der Prozesseinflussgrößen auf das Fertigungsergebnis zu untersuchen, um damit zu einem besseren Prozessverständnis beizutragen. Im Vordergrund stehen dabei die Auswirkungen unterschiedlicher Technologieparameter und der Einfluss des aktuellen Werkzeugzustands auf die gefertigten Bauteiloberflächen. Diese Analysen untersuchen insbesondere Teilaspekte für den jeweiligen spezialisierten Anwendungsfall, wodurch die, basierend auf deren Ergebnissen, gebildeten Prozessmodelle lediglich die erprobten Zusammenhänge abbilden können.

Die steigende Bedeutung von KI im produktionstechnischen Kontext der Qualitätssicherung hält zunehmenden Einzug in der Forschungslandschaft. Die steigende Datenverfügbarkeit und die verbesserte Leistungsfähigkeit von KI-Algorithmen eröffnen neue Wege, um den genannten Herausforderungen entgegenzuwirken. KI-Algorithmen liefern für die Prozessüberwachung den entscheidenden Vorteil ein Abbild des aktuellen Prozesszustands alleinig aus prozessnah gewonnenen Signaldaten in Form eines trainierten Modells zu erzeugen. Die explizite Modellierung basierend auf tiefem Prozessverständnis entfällt beim Einsatz maschineller Lernmethoden. Dies erlaubt bei ausreichender Signalqualität ein umfassendes Prozessmodell inklusive von in den Daten widergespiegelten Einfluss- und Störgrößen zu erstellen. Aktuelle experimentelle Ansätze nutzen diese Perspektiven für den Einsatz von neuronalen Netzen in der Prozess- bzw. Qualitätsüberwachung. Insbesondere dienen Technologieparameter als Inputgrößen für das Training der Algorithmen hinsichtlich gemessener Kennwerte zur Oberflächenrauheit. Weitere Arbeiten ergänzen die Eingangparameter mit Signaldaten aus externer Sensorik oder ermittelten Werten aus Abtragssimulationen, um eine Verbesserung der Genauigkeit zu erzielen. Neben der Nutzung von hauptsächlich statischen Technologiegrößen und niederdimensionalen Eingangsräumen verbreiten sich zunehmend Forschungsansätze, die hochdimensionale Zeitreihendaten sowohl aus interner als auch aus externer Sensorik für die Qualitätsvorhersage nutzen. Das Hauptaugenmerk liegt insbesondere in der Vorhersage von Qualitätskenngrößen bei Bohrprozessen. Insgesamt lässt sich als Gemeinsamkeit der KI-basierten Ansätze beobachten, dass einzelne Geometrielemente bzw. Formtoleranzen im Fokus der Untersuchungen stehen. Zusätzlich konzentrieren sich die Autoren häufig auf einen Algorithmus, welcher für die jeweilige Problemstellung hin optimiert wird.

Zusammenfassend gehen aus dem aktuellen Stand zur KI-basierten Qualitätsüberwachung insbesondere Methoden und Ansätze hervor, die sich auf Basis von externer Sensorik vorwiegend mit der Vorhersage der Oberflächenrauheit befassen. Erste Ansätze unter der Nutzung maschineninterner NC-

Signal­daten zeigen zusammen mit den Vorteilen integrierter Sensorik das enorme Potenzial von ML-­Algorithmen für die Abbildung der Prozess­zusammenhänge und die Qualitäts­prädiktion anhand der Betrachtung einzelner Geometrie­elemente. Langfristiges Ziel muss es also sein, die Entwicklung und den Einsatz von KI-­basierten Methoden in der Qualitäts­überwachung span­nender Prozesse zu vereinfachen, um die Grundlage für einen industriellen Einsatz in Produktions­unternehmen zu legen. Hierbei bleibt offen, wie sich aus den zur Verfügung stehenden Daten die notwendigen Informationen ableiten lassen, um mithilfe von KI-­basierten Methoden Prozess­wissen­modelle zu erstellen, die es ermöglichen, eine online­fähige In-­Prozess-­Überwachung für allgemeine Zerspan­prozesse auf Werkzeug­maschinen zu realisieren. Neben der Frage nach den Informationen, existieren Lücken in der Forschungs­land­schaft hinsichtlich geeigneter Methoden zur Daten­vor­ver­ar­bei­tung, Auto­ma­tisier­barkeit der einzelnen Schritte von den Roh­daten hin zu den trainierten Modellen, Übertragbarkeit der Modelle auf weitere Maschinen und Prozesse sowie Grenzen der KI-­basierten Methoden.

### 3 Zielsetzung und Vorgehensweise

Die 100%-Prüfung der Qualität spanend hergestellter Werkstücke ist nach aktuellem Stand nur mit hohen zeitlichen und finanziellen Aufwänden für Produktionsunternehmen möglich. Insbesondere bei sicherheitskritischen Bauteilen ist die Qualitätsprüfung von jedem gefertigten Werkstück jedoch unabdingbar. Die Herausforderung liegt in der Reduktion des Ressourceneinsatzes für die Qualitätsprüfung ohne dabei die Prüfintervalle zu vergrößern. In der Regel wird demnach eine In-Prozess-Messung der Qualitätsmerkmale angestrebt, welche eine hauptzeitparallele Bestimmung der erreichten Werkstückqualität ohne einen nachfolgenden operativen Schritt für die Qualitätssicherung ermöglicht.

Zusätzlich steigt die Anzahl an verfügbaren Datenquellen in der Produktion durch die stetig voranschreitende Digitalisierung [TSAI14]. Hinzu kommt die steigende Leistungsfähigkeit datenbasierter Modellierungsmethoden aus der KI, welche die Möglichkeit bieten, die aufkommenden enormen Datenmengen zu verarbeiten. Das Potenzial in diesen Daten bleibt jedoch oftmals ungenutzt, da es an Expertise im Umgang mit Daten oder prozessübergreifendem Datenmanagement fehlt und somit der Zugriff auf notwendige Informationen für die datenbasierte Modellierung verborgen bleibt.

Hieraus ergibt sich die in der Einleitung erläuterte übergeordnete Zielstellung dieser Arbeit. Die Qualitätsprognose soll basierend auf NC-Signaldaten mittels Verfahren des ML erfolgen, um die Komplexität des Zerspanprozesses und dessen Einflussgrößen ausreichend abbilden zu können. Die untersuchte Fertigungstechnologie in dieser Arbeit ist das Fräsen mit drei linearen Vorschubachsen. Der Schwerpunkt wird dabei auf Gestaltabweichungen 1. Ordnung der erzeugten Bauteilgeometrie gelegt. Insbesondere die Werkstückqualität, ausgeprägt als Form- und Lagetoleranzen nach DIN EN ISO 1101 für verschiedene Geometrieelemente am gesamten Werkstück, wurde unter den genannten Gesichtspunkten in bisherigen Arbeiten nicht berücksichtigt. Ausgehend vom aktuellen Stand des Wissens und der identifizierten Forschungslücke lautet das Forschungsziel dieser Arbeit:

*Erforschung und Entwicklung von bauteil- und maschinenunabhängigen Lösungen für die In-Prozess-Qualitätsprognose von spanend hergestellten Werkstücken, basierend auf prozessparallel erfassten Daten aus der numerischen Steuerung einer Werkzeugmaschine, mittels trainierter Algorithmen aus dem Bereich des maschinellen Lernens.*

Für die Gewährleistung der industriellen Anwendbarkeit wird eine Vorgehensweise entwickelt, welche alle notwendigen Schritte von der Datenerzeugung über die Datenverknüpfung, Identifikation sowie Auswahl relevanter Daten hin zur Datenmodellierung und den prozessnahen Einsatz der KI-basierten Prognosemodelle abdeckt. Ziel ist es, einen möglichst hohen Automatisierungsgrad für die Teillösungen zu erreichen, um einen einfachen Zugang zu datenbasierten Qualitätsvorhersagemodellen zu erlauben. Ausgehend von der Zielstellung dieser Arbeit lassen sich die vier folgenden Forschungsfragen ableiten:

*FF1: Welche Daten aus entscheidenden Wertschöpfungsschritten (hier: Konstruktion und Arbeitsvorbereitung, Teilefertigung, Qualitätssicherung) werden benötigt, um datenbasierte Qualitätsprognosen für Zerspanprozesse zu ermöglichen?*

*FF2: Wie müssen die erfassten Daten vorverarbeitet werden, um diese in eine hochqualitative Datenbasis als Grundlage für die KI-Modellentwicklung zur Prädiktion der Werkstückqualität zu überführen?*

*FF3: Welche Teilprozesse sind notwendig, um KI-basierte Qualitätsprognosemodelle für spanend (hier: Fräsprozesse) hergestellte Werkstücke zu erstellen und deren produktive Einsatzfähigkeit zu evaluieren?*

*FF4: Wie lassen sich die Teilprozesse automatisieren, um für einen vorliegenden Bearbeitungsprozess optimierte Prozesswissensmodelle zu entwickeln und diese in einem prozessnahen datenbasierten Qualitätsüberwachungssystem einzusetzen?*

Für die wissenschaftliche Beantwortung dieser Fragestellungen ergibt sich die in Abbildung 3.1 veranschaulichte Gliederung für das weitere Vorgehen in dieser Arbeit. In Kapitel 4 wird zunächst der Entwicklungsrahmen beschrieben und auf die Erfassung der Daten entlang der entscheidenden Wertschöpfungs-

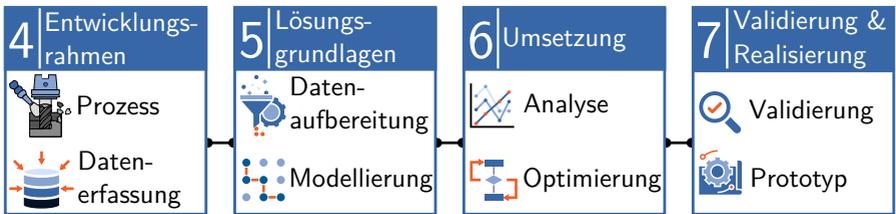


Abbildung 3.1: Gliederung und Kapitelstruktur der vorliegenden Arbeit.

schritte eingegangen. Kapitel 5 behandelt die notwendigen Lösungsgrundlagen für die Umsetzung der prozessbegleitenden Qualitätsüberwachung. Das Kapitel umfasst neben Methoden zur Datenaufbereitung den Aufbau der Prozesskette für das maschinelle Lernen, welche dazu dient die Qualitätsprognosemodelle zu erstellen und für deren produktiven Einsatz zu evaluieren. Schließlich folgt in Kapitel 6 die Umsetzung der prozessparallelen Qualitätsprognose basierend auf den zuvor konzipierten Lösungen anhand des in Kapitel 4.1 eingeführten Demonstratorszenarios. Aufbauend auf den Ergebnissen wird ein Optimierungsalgorithmus vorgestellt, welcher ermöglicht die Teilprozesse zu automatisieren, um optimierte Prozessmodelle zu entwickeln. In Kapitel 7 wird das erarbeitete Gesamtkonzept hinsichtlich dessen Übertragbarkeit auf weitere Werkstücke bzw. Prozesse validiert und es erfolgt die Realisierung der In-Prozess-Qualitätsbewertung in einem prototypisch umgesetzten, prozessnahen Qualitätsüberwachungssystem. In einer abschließenden Diskussion der Ergebnisse werden die Voraussetzungen für einen industriellen Einsatz sowie resultierende Potenziale und mögliche Hemmnisse diskutiert.



## **4 Entwicklungsrahmen für die datenbasierte Qualitätsüberwachung**

Das erste Teilziel in dieser Arbeit ist der Aufbau eines Entwicklungsrahmens für die datenbasierte Qualitätsüberwachung. Dabei bildet die Erfassung der notwendigen Daten und deren Verknüpfung mit Daten aus entscheidenden Wertschöpfungsschritten die Grundlage für die Beantwortung der ersten Forschungsfragestellung und die anschließende Generierung der datenbasierten Prozesswissensmodelle. Für die praxistaugliche Herleitung der verwendeten Verfahren wird in Kapitel 4.1 die Versuchsumgebung vorgestellt und darauf aufbauend auf die Konzeptionierung eines Referenzwerkstücks eingegangen, welches als durchgängiges Entwicklungsmuster im Rahmen dieser Arbeit dient. Anhand des Musterbauteils erfolgt ab Kapitel 4.2 die Einführung in die kontextsensitive Erfassung und Verknüpfung der Daten entlang der Wertschöpfungskette.

### **4.1 Versuchsumgebung und Referenzprozess**

Dieses Kapitel gibt einen Einblick in die genutzten Fräs-Bearbeitungszentren, anhand derer die Lösungen innerhalb dieser Arbeit entwickelt werden. Insbesondere die verbauten Steuerungen nehmen als Datenquelle für die datenbasierte Qualitätsüberwachung eine wesentliche Rolle ein. Im zweiten Teil wird ein Musterwerkstück inklusive zugehörigem Prozess eingeführt, um die nachfolgenden hergeleiteten Ansätze zu veranschaulichen und umzusetzen.

#### **4.1.1 Versuchsmaschinen**

Alle im Rahmen dieser Arbeit durchgeführten Entwicklungen und Versuchsreihen werden an zwei WZM unterschiedlicher Maschinenhersteller durchgeführt. Einen Überblick über die Konfigurationen und Eigenschaften beider Maschinen geben Abbildung 4.1 und 4.2. Bei beiden Maschinen handelt es sich um Fräs-Bearbeitungszentren (vgl. Kapitel 2.2.1), welche jeweils mit einer SINUMERIK 840D sl Steuerung der Firma Siemens ausgestattet sind.

Die zur Ausstattung des Instituts für Produktionsmanagement, Technologie und Werkzeugmaschinen (PTW) gehörende WZM DMC 850V vom Her-



<b>Verfahrenweg</b>		<b>Wegmesssystem</b>	
X, Y, Z	: 850 x 520 x 475 mm	Auflösung	: 0,01 $\mu\text{m}$
<b>WZ-System</b>		Eingabefinheit	: 0,1 $\mu\text{m}$
Werkzeugaufnahme : HSK-A63		<b>Steuerung</b>	: SIEMENS SINUMERIK 840D sl
<b>Antriebe</b>		Version	: V04.07 + SP 06
Vorschubgeschw.	: bis 42 m/min	lpo-Taktzeit $t_{\text{lpo}}$	: 4 ms
Nennleistung	: 35 kW	Lageregeltakt $t_{\text{Lage}}$	: 2 ms
Max. Drehmoment	: 130 Nm		
Max. Drehzahl	: 20.000 1/min		

Abbildung 4.1: Spezifikationen der Versuchsmaschine DMG MORI DMC 850V, eigene Darstellung basierend auf [DECK19].

steller DMG MORI ist ein vertikales Bearbeitungszentrum mit drei linearen Vorschubachsen, dessen Spindel sich in den drei Raumrichtungen X/Y/Z verfahren lässt (vgl. Abbildung 4.1). Der starre Werkstücktisch erlaubt die Bearbeitung von Werkstücken mit einem Gesamtgewicht von 1000 kg. [DECK19] Die DMC 850V dient in den weiteren Ausführungen der Entwicklung der Vorgehensweisen und Algorithmen am Beispiel eines universell einsetzbaren Bearbeitungszentrums mit drei Vorschubachsen.

Bei der G350 (2. Generation) handelt es sich um ein 5-Achs-Universal-Bearbeitungszentrum des Maschinenherstellers GROB-WERKE (vgl. Abbildung 4.2). Die horizontale Spindellage ermöglicht zusammen mit dem in zwei Raumrichtungen drehbaren Dreh-Schwenk-Tisch komplexe 5-Achs-Simultanbearbeitungsprozesse. Vorteile der Horizontalspindel sind u. a. der optimierte Spänefall und die Möglichkeit der Überkopfbearbeitung. Dabei verfährt die Spindel in X- und Z-Richtung. Die lineare Vorschubbewegung der Y-Achse bewegt den Tisch in vertikaler Richtung. Besonderheit hierbei ist der Antrieb der Y-Achse über zwei parallel geschaltete Antriebsmotoren. Durch ihren grundlegend unterschiedlichen Gesamtaufbau im Vergleich zur DMC 850V wird die ebenfalls zur Ausstattung des PTW gehörende Universalmaschine G350



<b>Verfahrenweg</b>		<b>Wegmesssystem</b>	
X, Y, Z	: 600 x 855 x 750 mm	Auflösung	: 0,01 $\mu\text{m}$
<b>WZ-System</b>		Eingabefeinheit	: 0,1 $\mu\text{m}$
Werkzeugaufnahme	: HSK-A63	<b>Steuerung</b>	: SIEMENS SINUMERIK 840D sl
<b>Antriebe</b>		Version	: V04.08 + SP 04
Vorschubg. X/Y/Z	: bis 70/45/90 m/min	lpo-Taktzeit $t_{lpo}$	: 4 ms
Vorschubg. A/B	: bis 35/50 U/min	Lageregeltakt $t_{Lage}$	: 2 ms
Nennleistung	: 32 kW		
Max. Drehmoment	: 206 Nm		
Max. Drehzahl	: 16.000 1/min		

Abbildung 4.2: Spezifikationen der Versuchsmaschine GROB G350, eigene Darstellung basierend auf [GROB20].

für Untersuchungen hinsichtlich Übertragbarkeit der Methoden auf weitere WZM sowie für die Verifikation der entwickelten Datenstrukturen und Algorithmen gewählt. Dies stellt sicher, dass die Lösungen in dieser Arbeit für einen breiten Einsatz geeignet und in weiteren Anwendungen nutzbar sind.

### 4.1.2 Konzeption des Referenzprozesses

Für die Entwicklung der in dieser Arbeit angestrebten Vorgehensweisen und Algorithmen, dient ein durchgängiges Musterwerkstück samt Fertigungsprozess, um die zentrale Datengrundlage für die datenbasierte Modellierung zu schaffen. Das Musterwerkstück besteht aus einer konstruierten Geometrie, welche sich aus zuvor festgelegten Anforderungen und Randbedingungen ableitet. Die Rahmenbedingungen in diesem Zusammenhang sind die Nutzung von TiAl6V4 als Werkstoff (vgl. Tabelle 4.1), Schaftwerkzeugen mit einem Durchmesser von 8 mm und die Fertigbarkeit auf einem 3-achsigen Bearbeitungszentrum sowie das Fräsen des Werkstücks aus einem Rohmaterialblock. Diese ergeben sich aus den erarbeiteten Entwicklungsrahmen des Projekts „TensorMill“, welche sich an branchenüblichen Anforderungen der Luft- und

Tabelle 4.1: Physikalische und mechanische Eigenschaften des eingesetzten Werkstoffs [HSM 22; KIND22; TIME21].

Bezeichnung	TiAl6V4
Werkstoffnummer	3.7164
Dichte	$4,43 \frac{g}{cm^3}$
E-Modul	$110.000 \frac{N}{mm^2}$
Streckgrenze $R_{p0.2}$	$830 \frac{N}{mm^2}$
Anwendung	Luft- & Raumfahrt

Raumfahrtindustrie orientieren [FERT23]. Basierend auf den Rahmenbedingungen werden Anforderungen definiert, welche sich aus der Zielstellung dieser Arbeit ableiten lassen. Für die Untersuchung von datenbasierten Modellierungsansätzen soll an der Werkstückgeometrie die Möglichkeit geschaffen werden, einen großen Umfang an Form- und Lagetoleranzen nach DIN EN ISO 1101 auszuwerten. Zusätzlich wird gefordert, eine möglichst geringe Anzahl an unterschiedlichen Werkzeugen und lediglich eine Aufspannung für die Fertigung des Musterwerkstücks zu nutzen. Als weitere Pflichtforderung für die Gewährleistung der breiten Anwendbarkeit steht, dass neben einfachen Geometrielementen wie Ebenen oder Zylindern zusätzlich allgemeine Elemente aus dem Bereich der Freiformflächen in der Konstruktion mit berücksichtigt werden sollen.

Aus den genannten Anforderungen und Rahmenbedingungen resultiert in einem iterativen Konstruktionsprozess das in Abbildung 4.3 gezeigte Musterwerkstück, welches als Referenzwerkstück für die weiteren Entwicklungen im Rahmen dieser Arbeit dient. Die Höhe der gezeigten Geometrie beträgt 8 mm und weist eine Länge von 48,5 mm in X-Richtung und 33 mm in Y-Richtung auf. Insgesamt lassen sich die 15 Geometrielemente in 11 Haupt- und 4 zusätzliche Geometrielemente einteilen. Zwischen LINE\_1 und LINE\_7 befindet sich ein dünnwandiger Steg mit einer Breite von 1,5 mm. LINE\_3 liegt um einen Winkel von  $0,5^\circ$  gedreht zur X-Achse. Zusätzlich befinden sich am Werkstück die beiden Splineflächen INNERSPLINE und OUTERSPLINE sowie eine gefräste Bohrung (INNERCIRCLE) und ein Außenzylinder (OUTERCIRCLE). Die treppenförmigen LINESTEPS dienen dazu unterschiedliche Tiefenzustellungen während des Prozesses abzudecken. Die Abstufungen variieren dabei von 8 mm hin zu 2 mm in einer Schrittweite von 2 mm.

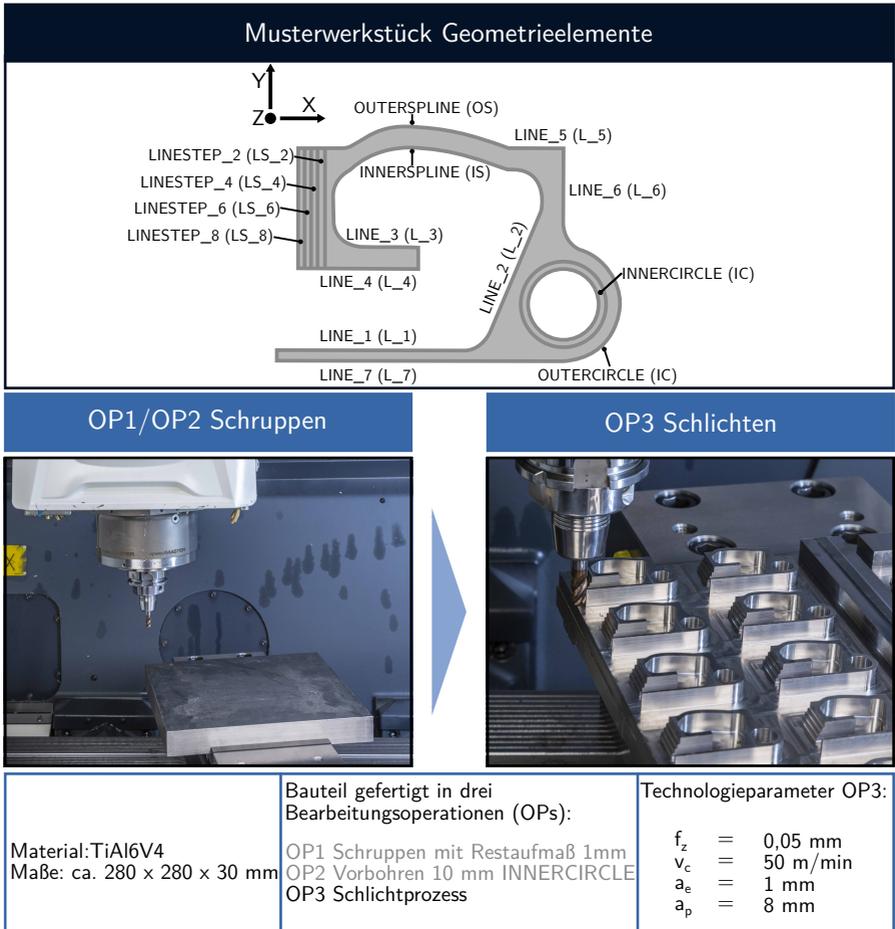


Abbildung 4.3: Musterwerkstück - Geometrie und Prozess.

Tabelle 4.2: Eigenschaften der eingesetzten Werkzeuge für den Schlichtprozess OP3.

Bezeichnung	WS_002787	WS_002790	WS_002792
Hersteller	Linner	Linner	Linner
Substrat	VHM	VHM	VHM
Beschichtung	AlTiSiN	AlTiSiN	AlTiSiN
Durchmesser in mm	8	8	8
Gesamtlänge in mm	64	64	64
Schneidenlänge in mm	14	19	19
Zähnezahl	4	5	3
Drallwinkel in °	40	41	40
Eckphase	45° x 0,3 mm	45° x 0,2 mm	45° x 0,2 mm

Die restlichen vier Geometrielemente entstehen durch die Verbindung der Hauptgeometrielemente zu einem zusammenhängenden Gesamtwerkstück.

Für die Herstellung des Musterwerkstücks ist ein Prozess vorgesehen, welcher aus drei Bearbeitungsoperationen (OP) besteht. OP1 umfasst das Schruppen, umgangssprachlich „aus dem Vollen“, entlang der Werkstückgeometrie mit einem resultierenden Restaufmaß von 1 mm. Das Restaufmaß ist bewusst gewählt, um geringe Bearbeitungskräfte zu erzeugen. Diese spiegeln sich in niedrigen Signalamplituden der NC-Signaldaten wider, um herausfordernde Bedingungen für die Entwicklung der Algorithmen zu schaffen. OP2 dient wie OP1 der Vorbereitung des Schlichtprozesses und erzeugt mittels eines Bohrwerkzeugs die Vohrbohrung mit einem Durchmesser von 10 mm für den INNERCIRCLE (Enddurchmesser 11 mm). OP3 stellt den endkonturrelevanten Schlichtprozess entlang der finalen Werkstückgeometrie dar. Tabelle 4.2 fasst die Eigenschaften der eingesetzten Werkzeuge für den Schlichtprozess zusammen. Die dabei genutzten Technologieparameter sind in Abbildung 4.3 aufgeführt und bewusst an industrieüblichen Werten angelehnt.

## 4.2 Datenerfassung entlang der Wertschöpfungskette

Dieses Unterkapitel legt das Fundament für den konzipierten Lösungsansatz: Die Erfassung der Daten entlang entscheidender Prozessschritte der Wertschöpfungskette. Dafür wird zunächst auf die Erfassung der NC-Signaldaten mit zugehörigem zerpannungstechnischen Kontext eingegangen. Für die Generierung der Qualitätsinformationen folgt in Kapitel 4.2.4 die Vorstellung der qualitätsrelevanten Datenquellen.

### 4.2.1 Datenerfassung mittels Edge-Computing

Für die Aufzeichnung und Verarbeitung von NC-Signaldaten wird eine am PTW entwickelte Lösung genutzt, die auf den bereitgestellten Schnittstellen der von Siemens entwickelten „Industrial Edge for Machine Tools“ (im Folgenden Industrial Edge genannt) innerhalb der Siemens Edge Computing Platform<sup>1</sup> aufbaut. Industrial Edge ist ein von der Firma Siemens entwickeltes linuxbasiertes Softwaresystem (vgl. Abbildung 4.4), welches auf industrietauglicher Hardware ausgeführt wird. Die am PTW genutzte Hardware ist ein Simatic Nanobox Industrie-PC (IPC) von Siemens<sup>2</sup>. Die vollständig containerisierte *Application engine* liefert verschiedene grundlegende *Runtime services* wie die Geräte- bzw. Applikationsverwaltung, Logging und Monitoring, Nachrichtendienste und Nutzerverwaltung. Darauf aufbauend ermöglichen integrierte *Automation adapter* die Verbindung und Kommunikation mit den entsprechend konfigurierten Automatisierungssystemen. Diese liefern über bereitgestellte Schnittstellen den Zugriff auf Parameter und hochfrequente Datenströme aus den verbundenen Automatisierungssystemen. Die containerbasierte Struktur ermöglicht schließlich die Entwicklung eigener sogenannter *Edge Apps*, welche über die IoT-Plattform Mindsphere von Siemens auf der Industrial Edge bereitgestellt werden können. [SIEM22a]

Die am PTW entwickelte Edge App *MachiningContextAdapter (MCA)* zur Aufzeichnung der NC-Signaldaten nutzt die Schnittstellen des Automation adapters *SinumerikAdapter*. Der SinumerikAdapter ist eine auf dem IPC installierte Industrial Edge Applikation, welche über den auf der NC-Steuer-einheit (engl.: Numerical Control Unit) (NCU) installierten Compilezyklus

---

<sup>1</sup>Die Edge Computing Platform umfasst die gesamte Plattformarchitektur für das Edge Device Management, Application Management, Automation Connectivity und Mindsphere IoT. Weiterführende Informationen finden sich in [SIEM22a].

<sup>2</sup>Die technischen Spezifikationen finden sich in Anhang A.1.

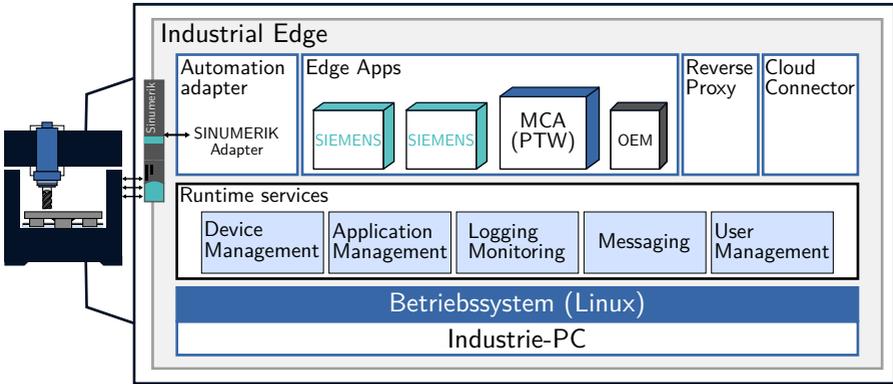


Abbildung 4.4: Architektur Industrial Edge for Machine Tools, eigene Darstellung basierend auf [SIEM22a].

(vgl. Kapitel 2.2.3) *HF\_Probe* über Ethernet via TCP/IP kommuniziert. Für die physische Verbindung von IPC und NCU wird steuerungseitig die X120 Schnittstelle genutzt, welche als Anschluss für das interne Maschinennetzwerk dient. Innerhalb der Industrial Edge-Umgebung ermöglicht der SinumerikAdapter auf die konfigurierten Daten in Form von strukturierten Datenpaketen über unterschiedliche Funktionen und Schnittstellen zuzugreifen. Je nach gewählter Funktion resultieren verschiedene maximale Abtastfrequenzen und damit Klassen für die erfassten Datenpakete. Die Kommunikation basiert zum einen auf dem MQTT<sup>3</sup>-Protokoll und zum anderen auf dem Request-Response-Verfahren, abhängig von der genutzten Funktionalität für den Zugriff auf die Datenpakete.

Datenpakete der Klasse *sinumerik\_hf\_data* (kurz: *hf\_data*) enthalten erfasste Datenpunkte aus dem Lageregelkreis der NC mit Abtastintervallen im Lageregeltakt  $t_{Lage}$  und werden per MQTT auf dem internen Datenbus bereitgestellt. Signale dieser Klasse erhalten von Siemens aufgrund der hochfrequenten<sup>4</sup> und äquidistanten Abtastung die Bezeichnung „high quality data“ und repräsentieren im Rahmen dieser Arbeit die hochfrequenten NC-Signaldaten (vgl. Kapitel 4.2.2). Für den Zugriff auf NC-Variablen über den Lageregelkreis hinaus, stellt der SinumerikAdapter den *Parameter-service* bereit. Die

<sup>3</sup>MQTT ist ein Nachrichtentransportprotokoll für das IoT. Die Grundlage liefert das Publish/Subscribe Prinzip, welches sich ideal für die Verbindung entfernter Geräte mit geringem Code Aufwand und minimaler Netzwerkbandbreite eignet. [MQTT22]

<sup>4</sup>Aufzeichnungsfrequenzen von bis zu 1000 Hz je nach  $t_{Lage}$  der NC

entsprechenden Datenpakete werden als *sinumerik\_lf\_data* (kurz: *lf\_data*) bezeichnet. Diese per Request-Response zur Verfügung gestellte Schnittstelle bietet den Zugriff auf sämtliche Datenpunkte, welche über die Bedientafelschnittstelle (BTSS)<sup>5</sup> von SIEMENS-NC abrufbereit sind. [SIEM22b]

Die in Python<sup>6</sup> entwickelte MCA-App (vgl. Abbildung 4.5) greift die konfigurierten Datenpunkte vom internen Datenbus ab. Dieser liefert alle 200 ms ein neues Datenpaket im Datenaustauschformat JSON<sup>7</sup> mit 100 Datenpunkten für jedes konfigurierte *hf\_data*-Signal. Parallel dazu werden die *lf\_data*-Pakete über den Parameter-service zyklisch niederfrequent<sup>8</sup> abgefragt. Nach Erhalt eines *hf\_data*-Paketes verknüpft die MCA-App dieses mit den entsprechenden innerhalb des letzten 200 ms Intervalls empfangenen *lf\_data*-Paketen, um diese gemeinsam im CSV<sup>9</sup>-Format auf dem IPC zwischenzupuffern und über eine integrierte Web-API<sup>10</sup> bereitzustellen. Der Dateipuffer auf dem IPC der Industrial Edge dient dazu, die Datenpakete auf dem IPC lokal zu speichern, bis diese über einen GET-Request Befehl an die API abgefragt werden. Die Synchronisation von *lf\_data* mit den *hf\_data* erfolgt mithilfe einer Taktzählvariable. Dafür liefert der installierte Compilezyklus *HF\_Probe* auf der Steuerung eine steuerungsinterne Variable, welche sich mit jedem durchgeführten Lageregeltakt der Maschine um 1 erhöht. Jedes auf dem Databus ankommende *hf\_data*-Paket, enthält diesen eindeutigen Taktzähler. Bei entsprechender Konfiguration der Schnittstelle für *lf\_data* enthalten zugehörige Datenpakete zusätzlich diesen Taktzähler. Dies erlaubt die Synchronisation beider Datenquellen innerhalb der MCA-App. Zusammenfassend ermöglicht die MCA-App eine äquidistante Erfassung von über 70 NC-Signalen<sup>11</sup> mit einer Abtastrate von 500 Hz (bei  $t_{Lage} = 2$  ms) und dazu synchronisierten *lf*-Daten.

---

<sup>5</sup>Eine ausführliche Dokumentation der zur Verfügung stehenden BTSS-Datenpunkte einer SIEMENS SINUMERIK-Steuerung liefert das Listenhandbuch für NC-Variablen und Nahtstellensignale [SIEM18c].

<sup>6</sup>Python ist eine interpretierte, objektorientierte Hochsprache mit dynamischer Semantik [PYTH22]. Weitere Informationen unter <https://www.python.org/>.

<sup>7</sup>JavaScript Object Notation - schlankes Austauschformat, welches sowohl für Menschen einfach lesbar und für Maschinen einfach zu verarbeiten ist. Weitere Informationen unter: <https://www.json.org/json-en.html>.

<sup>8</sup>Abfragen ca. alle 50 ms. Je höher die Abfragerate, desto stärker wird die NC-Hardware belastet, wodurch bei drohender Überlastung eine Priorisierung der Aufgaben für die Prozesssteuerung innerhalb der NC erfolgt.

<sup>9</sup>Comma-separated values (Dateiformat). Charakteristisch für dieses Dateiformat ist die Trennung der Datenpunkte durch Kommas. [SHAF05]

<sup>10</sup>Application Programming interface (dt.: Programmierschnittstelle).

<sup>11</sup>getestet an der WZM G350 (vgl. Kapitel 4.1.1)

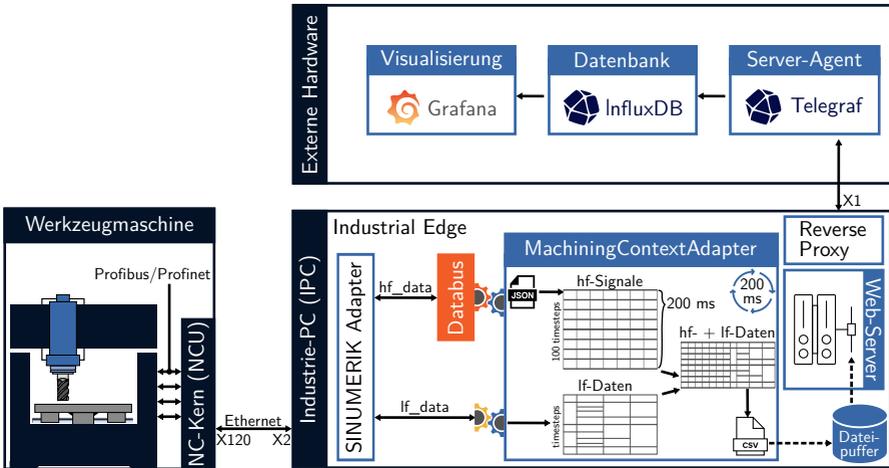


Abbildung 4.5: Erfassung interner NC-Signaldaten an WZM des PTW. X1 entspricht der Ethernet-Schnittstelle des IPC zum Maschinenetzwerk. X2 dient zur Verbindung an ein externes Netzwerk.

Über ein verfügbares HTTP-Plugin für den Server-Agent Telegraf<sup>12</sup> werden die gepufferten Daten über HTTP-GET-Requests kontinuierlich über die Web-API abgefragt und in der Datenbank gespeichert. Am PTW wird in diesem Fall die Zeitreihendatenbank InfluxDB genutzt, welche sich zusammen mit dem Telegraf auf einem externen virtuellen Server im Servercluster der Universität befindet. Die Bereitstellung der Daten via Web-API über einen Export-Server auf dem IPC bietet die Möglichkeit über entsprechende Clients (in diesem Fall der Telegraf) beliebig Anfragen zur Übertragung der Daten zu stellen und damit deren Weiterverarbeitung auf beliebiger externer (Cloud-)Infrastruktur umzusetzen.

Diese Ausführungen stellen den grundsätzlichen Aufbau zur Erfassung interner Signaldaten aus der NC im Rahmen dieser Arbeit dar. Diese liegen als Rohdaten vor und müssen für die weitere Nutzung kontextspezifisch interpretiert werden. Die folgenden Kapitel 4.2.2 und 4.2.3 beschreiben die kontextspezifische Interpretation und die darauf aufbauend notwendige Datenvorverarbeitung.

<sup>12</sup>Weitere Informationen unter: <https://www.influxdata.com/time-series-platform/telegraf/>.

## 4.2.2 Hochfrequente NC-Signaldaten (hf\_data)

Für die kontextspezifische Interpretation der kontinuierlich im Lageregeltakt erfassten hochfrequenten NC-Signaldaten muss die entsprechende Konfiguration des Automatisierungssystems betrachtet werden. Die Signale liegen aufgrund der Vielfalt an Automatisierungssystemen in proprietären Signalbenennungen des Herstellers vor. Jede Signalquelle erhält einen englischsprachigen Bezeichner gefolgt von einem Senkrechtstrich „|“ und zugehöriger Achse in Form eines ganzzahligen Wertes. Zum Beispiel CURRENT|1 für das Stromsignal der ersten Achse. [SIEM22b]

Die Zuordnung der Achsnummern zu den physikalischen Maschinenachsen lässt sich aus den Maschinendaten und damit dem Inbetriebnahmearchiv (IBN-Archiv) der NC entnehmen. Für die Konfiguration der Maschinenachsen in einer SINUMERIK-Steuerung sind vier prinzipielle Formen von Achsen zu unterscheiden: Maschinenachsen, Kanalachsen, Geometrieachsen und Zusatzachsen. Maschinenachsen repräsentieren die an einer Maschine vorhandenen Bewegungsachsen und teilen sich je nach Bewegungsart in Linear- und Rundachsen auf. Die Gesamtheit aller einem Kanal zugeordneten Maschinen, Geometrie- und Zusatzachsen bilden die Gruppe der Kanalachsen. Während die Maschinenachsen die physikalische Seite abbilden, handelt es sich bei Geometrie- und Zusatzachsen um die programmtechnische Seite des Bearbeitungsprozesses. Das kartesische Basiskoordinatensystem eines Kanals wird durch die Geometrieachsen gebildet, welche sich in der Regel direkt den Maschinenachsen zuordnen lassen. Die Gruppe der Zusatzachsen fasst alle Achsen zusammen, welche keine Geometrieachsen sind. Dabei ist kein geometrischer Zusammenhang definiert. Abbildung 4.6 zeigt die Maschinendaten und die Zuordnung der Geometrie- zu Kanal- und der Kanal- zu Maschinenachsen für die in dieser Arbeit betrachteten Maschinen. Jede Achse erhält dabei einen eindeutigen Namen. [SIEM17]

Daraus können die Achsnummern der Signalquellen den Maschinenachsen zugeordnet werden. Für die einkanalige NC der DMC 850V resultiert die Zuordnung der Geometrieachsen X, Y, und Z zu den Maschinenachsen und damit den Achsnummern 1, 2 und 3. Die Spindelachse liegt auf der 6. Kanalachse. Die Geometrieachsen der G350 sind ebenfalls den ersten 3 Maschinenachsen zugeteilt. Unterschiede ergeben sich in der Spindelachse auf Kanalachse 8 und den beiden Drehachsen A und B des Werkstücktisches auf den Kanalachsen 7 und 5. Eine Besonderheit zeigt sich in Kanalachse 30 durch die zweite Y-Achse „Y\_2“, welche der Achse des parallelgeschalteten zweiten Antriebsmotors entspricht. Zusätzlich liegt Achse „ZM“ auf der Achsnummer 20. Die ZM-Achse

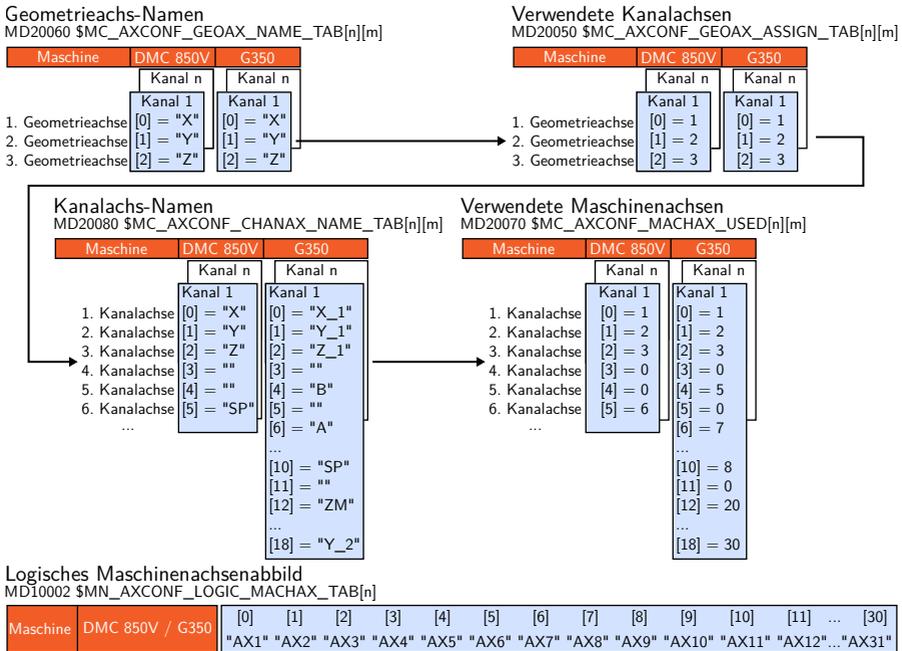


Abbildung 4.6: Achszuordnung innerhalb der NC der WZM, eigene Darstellung basierend auf [SIEM17].

ist in diesem Fall die Tür des Werkzeugwechsellunnels und wird im weiteren Verlauf nicht betrachtet. Zusätzlich zu den Maschinen- und Zusatzachsen im ersten Kanal verfügt die G350 über weitere Achsen auf den weiteren Kanälen 2 – 4. Dabei handelt es sich um zusätzliche Achsen, für Werkzeugwechsler und -magazin, welche für diese Arbeit nicht relevant sind. Basierend auf der Zuordnung der Signalnamen zu den Maschinenachsen können die erfassten Signaldaten mit der entsprechenden Quelle aus dem Lageregelkreis (vgl. Kapitel 2.2.3) verknüpft werden.

Abbildung 4.7 bildet angelehnt an Abbildung 2.8 die Signalnamen samt Achszuordnung am Beispiel des Lageregelkreises für die X-Achse ab. Parallel dazu resultieren die Achszuordnungen für Y- und Z-Achse. Der erweiterte Regelkreis zeigt zusätzlich zu den in Kapitel 2.2.3 vorgestellten Signalen, verfügbare Signale aus der dynamischen Vorsteuerung (engl.: Feed Forward Control), welche mit dem Zusatz „FFW“ versehen sind. Die dynamische Vorsteuerung ermöglicht den axialen Schleppfehler und damit die blei-

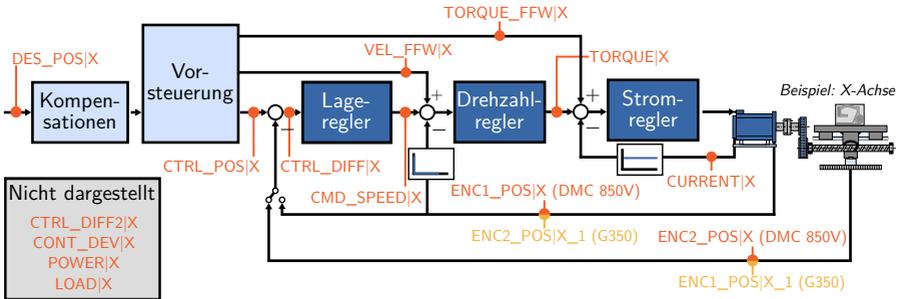


Abbildung 4.7: hf-Signale aus dem Lageregelkreis.

bende Regelabweichung zu reduzieren. Für die eingesetzte Drehzahl- und Momentenvorsteuerung werden zusätzliche Geschwindigkeits- bzw. Stromsollwerte auf die jeweiligen Eingänge geschaltet, sodass bei Vorgängen mit konstanter Geschwindigkeit sowie Beschleunigungsvorgängen eine größtmögliche Reduktion des resultierenden Schleppabstands erreicht wird. [SIEM18a] Zusätzlich zur Achszuordnung lassen sich den Maschinendaten noch weitere Informationen entnehmen, welche für die spätere Analyse und die Untersuchungen der Übertragbarkeit relevant sind. Neben den Signalen zur Vorsteuerung liefert die Datenerfassung zwei Signalgruppen für die Sollpositionen. DES\_POS-Signale enthalten Lagesollwerte nach dem Feininterpolator. CTRL\_POS enthält darüber hinaus die kompensierten Lagesollwerte, welche als Eingangssignale für den Lageregler dienen. Als Kompensationsverfahren existieren dabei die Temperatur-, Lose- und Nickkompensation sowie interpolatorische Kompensationen, wie Spindelsteigungsfehler- bzw. Messsystemfehler-, Durchhang- bzw. Winkligkeitsfehler- und Zylinderfehlerkompensation. [SIEM18a; SIEM22b] In Bezug auf die Regelabweichung liefert der HF\_Probe die Signale CTRL\_DIFF, CTRL\_DIFF2 und CONT\_DEV. CTRL\_DIFF enthält in diesem Zusammenhang die Regelabweichung inklusive kompensierter DSC<sup>13</sup>-Effekte. CTRL\_DIFF2 liefert die unkompensierten Schleppfehler und CONT\_DEV entspricht einer steuerungsintern berechneten Konturabweichung aus dem konturrelevanten Schleppfehler (Lagesollwert nach Feininterpolator – Lageistwert) und der zugehörigen Differenz zum Streckenmodell aus der Vorsteuerung. [SIEM18c; SIEM22b]

<sup>13</sup>Die „Dynamic Servo Control“ (DSC) eliminiert die systembedingt vorhandene Totzeit der zwischen NC und Antrieb üblicherweise verwendeten Drehzahlsollwertschnittstelle durch die Verlagerung des Lagereglers in den Antrieb [SIEM06].

Anzahl der Geber für die Lageistwerterfassung MD30200 \$MA_NUM_ENCS		Geber-Typ der Istwerterfassung MD30240 \$MA_ENC_TYPE		Linearmaßstab MD31000 \$MA_ENC_IS_LINEAR		Messsystem Istwerterfassung MD31040 \$MA_ENC_IS_DIRECT	
Maschine		Maschine		Maschine		Maschine	
DMC 850V	G350	DMC 850V	G350	DMC 850V	G350	DMC 850V	G350
[AX1] = 2	[AX1] = 2	[0,AX1] = 1	[0,AX1] = 4	[0,AX1] = 0	[0,AX1] = 1	[0,AX1] = 0	[0,AX1] = 1
[AX2] = 2	[AX2] = 2	[1,AX1] = 4	[1,AX1] = 1	[1,AX1] = 1	[1,AX1] = 0	[1,AX1] = 1	[1,AX1] = 0
[AX3] = 2	[AX3] = 2	[0,AX2] = 1	[0,AX2] = 4	[0,AX2] = 0	[0,AX2] = 1	[0,AX2] = 0	[0,AX2] = 1
[AX6] = 1	[AX5] = 1	[1,AX2] = 4	[1,AX2] = 1	[1,AX2] = 1	[1,AX2] = 0	[1,AX2] = 1	[1,AX2] = 1
	[AX7] = 1	[0,AX3] = 1	[0,AX3] = 4	[0,AX3] = 0	[0,AX3] = 1	[0,AX3] = 0	[0,AX3] = 0
	[AX8] = 1	[1,AX3] = 4	[1,AX3] = 1	[1,AX3] = 1	[1,AX3] = 0	[1,AX3] = 1	[1,AX3] = 0
	[AX20] = 1	[0,AX6] = 1	[0,AX5] = 4	[0,AX6] = 0	[0,AX5] = 0	[0,AX6] = 0	[0,AX5] = 1
	[AX30] = 1	[1,AX6] = 0	[1,AX5] = 0	[1,AX6] = 0	[1,AX5] = 0	[1,AX6] = 0	[1,AX5] = 0
			[0,AX7] = 4		[0,AX7] = 0		[0,AX7] = 1
			[1,AX7] = 0		[1,AX7] = 0		[1,AX7] = 0
			[0,AX8] = 1		[0,AX8] = 0		[0,AX8] = 0
			[1,AX8] = 0		[1,AX8] = 0		[1,AX8] = 0
			[0,AX20] = 4		[0,AX20] = 0		[0,AX20] = 0
			[1,AX20] = 0		[1,AX20] = 0		[1,AX20] = 0
			[0,AX30] = 1		[0,AX30] = 0		[0,AX30] = 0
			[1,AX30] = 0		[1,AX30] = 0		[1,AX30] = 0

0: Simulation 1: Rohsignalgeber 4: Absolutgeber allg.	0: rotatorische Lageistwerterfassung 1: lineare Lageistwerterfassung	0: Lageistwerterfassung am Motor 1: direkte Lageistwerterfassung
---	---	---

Abbildung 4.8: Parametrierung Lagegeber, eigene Darstellung basierend auf [SIEM18g].

Beim Vergleich der beiden betrachteten Bearbeitungszentren lässt sich zusätzlich die unterschiedliche Parametrierung der Lagegeber beobachten. Die zugehörigen Informationen sind ebenfalls dem IBN-Archiv zu entnehmen. Abbildung 4.8 fasst die relevanten Maschinendaten und die zugehörigen Werte für beide WZM zusammen. Alle Geometrieachsen der Maschinen besitzen nach MD30200 zwei Geber für die Lageistwerterfassung (Spindel und Drehachsen jeweils einen). MD30240 enthält die Information, über welche Art der Istwerterfassung die jeweiligen Lagegeber verfügen. Während bei der DMC 850V jeweils der erste Geber das Rohsignal liefert, ist es bei der G350 der zweite Geber. Dies geht einher mit dem Maschinendatum MD31000 für den Linearmaßstab, welches für jeden Absolutgeber den Wert für die lineare Lageistwerterfassung enthält. Dabei zeigt sich zusätzlich, dass die zweite Y-Achse der G350 nur mit einem rotatorischen Lageistwertgeber am Motor ausgestattet ist und demnach als parallel geschaltete Zusatzachse agiert. Für die Lageregelung basierend auf dem direkten Messsystem ist somit lediglich ein Glasmaßstab in der Maschine verbaut. Aus den Achsparametrierungen geht zusammenfassend hervor, dass die Hersteller beider Bearbeitungszentren verschiedene Ansätze bei der Erstellung des SPS-Projekts wählen. Für die späteren Untersuchungen zur Übertragbarkeit ist zu beachten, dass bei

der DMC 850V die direkten Lageistwertgeber auf dem hf-Signal ENC2\_POS und bei der G350 auf ENC1\_POS liegen. Dementsprechend entgegengesetzt verhalten sich die rotatorischen Geber an den Motoren. Im weiteren Verlauf der Arbeit werden die entsprechenden Signale aus der direkten Lageistwertbestimmung mit linENC\_POS und die Signale der Drehwinkelgeber als rotENC\_POS bezeichnet.

### 4.2.3 NC-Variablendaten (lf\_data)

Die erfassten Signaldaten aus dem Lageregelkreis entstehen nach der Vorverarbeitung innerhalb der NC. Zuvor erfolgte die Aufbereitung der Eingabedaten aus dem NC-Programm im NC-Interpreter, welche daraufhin in der Geometriedatenverarbeitung und dem Interpolator verschiedene Vorverarbeitungs- und Transformationsschritte durchlaufen (vgl. Abbildung 2.7). Die Verarbeitungsvorgänge beziehen aus dem NC-Programm hervorgehende zusätzliche Daten mit ein, um notwendige NC-Schaltbefehle auszulösen und die Transformation der eingehenden Verfahrenweisungen in Sollpositionsdaten durchzuführen. Dabei fließen neben den Achs- und Pfadkoordinaten vor allem der Interpolationstyp (z. B. *G00*, *G01*, *G02*, *G03*, *SPLINE*), Technologiewerte (z. B. *F*, *S*), die Spindeldrehrichtung und weitere M-Befehle (z. B. *M03*, *M04*, *M07*, *M08*), Nullpunktverschiebungen (z. B. *G54* ff., *SUPA*) sowie Werkzeugdaten und -korrekturen (z. B. Werkzeugname, -länge, -radius, *G41*, *G42*) mit ein [WELL19].

Diese und weitere in der NC verarbeiteten Daten lassen sich über die lf\_data-Schnittstelle erfassen. Im Sinne der Forschungsfragestellungen dieser Arbeit liefern gezielt ausgewählte, aus den zahlreichen zur Verfügung stehenden NC-Variablen<sup>14</sup> zusätzliche Informationen, um die als rohe Zeitreihen vorliegenden Signale aus dem Lageregelkreis in den entsprechenden Prozesskontext zu setzen. Tabelle 4.3 fasst die identifizierten Variablen zusammen. Für das bessere Verständnis im weiteren Verlauf dieser Arbeit werden für die Pfade aus der Steuerung Variablenamen vergeben. Die Frame-Variablen „linshift“ und „rotation“ liefern die aktuellen Werte der aktiven NPV. Für den Zugriff auf die jeweiligen Werte der Maschinenachsen ist die Achsnummer mit anzugeben (z. B. /Channel/ActualFrame/linShift[u1,1] für die lineare Verschiebung der X-Achse).

In der SINUMERIK-Steuerung existieren neben der Translation und Rotation zusätzliche Einstellmöglichkeiten für die Transformation des MKS in

<sup>14</sup>BTSS-Datenpunkte (vgl. Kapitel 4.2.1)

Tabelle 4.3: NC-Variablen für die Einordnung der hf-Signaldaten in den Prozesskontext.

Variablenpfad <i>SINUMERIK 840D sl</i>	Variablen- namen	Erklärung
/Channel/ActualFrame/linShift[u1,ax <sub>i</sub> ]	$x_{npv}, z_{npv}, y_{npv}$	Translation aktiver Frame
/Channel/ActualFrame/rotation[u1,ax <sub>i</sub> ]	$\varphi_{npv}, \theta_{npv}, \psi_{npv}$	Drehung aktiver Frame
/Channel/State/actToolIdent[u1,1]	$id_{wkz}$	Bezeichner aktuelles Werkzeug
/Channel/State/actToolLength1[u1,1]	$l_{wkz}$	aktive Werkzeuglänge
/Channel/State/actToolRadius[u1,1]	$r_{wkz}$	aktiver Werkzeugradius
/Channel/ProgramInfo/progName[u1,1]	ProgName	Programmname aktuell aktives Programm
mit ax <sub>i</sub> = Achsnummer der Achszuordnung (vgl. Abbildung 4.6)		

das WKS. Die Feinverschiebung (/Channel/ActualFrame/linShiftFine) ermöglicht eine zusätzliche Translation innerhalb der aktiven Hauptverschiebung. Darüber hinaus kann die aktive NPV gespiegelt (/Channel/ActualFrame/mirrorImgActive) und um beliebige Faktoren skaliert (/Channel/ActualFrame/scaleFact) werden. Diese Funktionen werden im Verlauf dieser Arbeit nicht weiter berücksichtigt.

#### 4.2.4 Qualitätsdaten

Die grundsätzliche Aufgabe von Fertigungsmesstechnik besteht darin, die Qualität von gefertigten Produkten sicherzustellen. Das zentrale Ziel dabei ist es zu verhindern, dass Werkstücke, welche die Qualitätsanforderungen nicht erfüllen, zum nächsten Schritt in der Produktion bzw. zum Kunden gelangen. In der Fertigungsmesstechnik dominieren das Messen und Prüfen von Werkstücken bzw. der Werkstückgestalt in Form der messtechnischen Ermittlung von Qualitätsmerkmalen. [MARX21; PFEI10; WECK12] Je nach Ausprägung der Abweichung (vgl. Kapitel 2.3.2) untergliedert sich die Gestalt von Werkstücken in die Grob- und Feingestalt. Die aus Standardgeometrieelementen, wie z. B. Ebene, Zylinder oder Torus zusammengesetzte Grobgestalt

lässt sich durch Prüfung von Maß- und Formabweichungen der Elemente bewerten. Zusätzlich wird deren Lage zueinander erfasst und mit der entsprechenden Sollvorgabe verglichen. Gestaltabweichungen höherer Ordnung (vgl. Kapitel 2.3.2), wie Welligkeiten und Rauheiten werden unter der Feingestalt zusammengefasst. [WECK12]

Für die Bestimmung der Gestaltabweichungen von Werkstücken lassen sich in der Fertigungsmesstechnik Messzeuge, Messgeräte, Hilfsmittel, Normale und Lehren unterscheiden. Je nach geforderter Genauigkeit, Eignung und dem Einsatzort sowie dem Umfang der Prüfung ist ein Großteil dieser Messmittel auf bestimmte Messaufgaben spezialisiert. [WECK12] Steigende Qualitätsanforderungen an die Produkte, sowie das Bestreben nach reproduzierbaren und benutzerunabhängigen Messergebnissen im Zusammenhang mit zunehmendem Kostendruck erfordern automatisierbare und universell einsetzbare Messgeräte, welche in der Lage sind auch komplexe Werkstücke bei hoher Aussagesicherheit zu prüfen. Für die geometrische Prüfung von Werkstücken haben sich deshalb für den industriellen Einsatz Geräte und Verfahren aus der Koordinatenmesstechnik (KMT) etabliert, deren typische Vertreter Koordinatenmessgeräte (KMG) darstellen. Diese gehören zur Gruppe der am weitesten verbreiteten Messmittel in der modernen Fertigungsmesstechnik [MARX21]. Während KMT für die Messung von Gestaltabweichungen 1. Ordnung (vgl. Kapitel 2.3.2) eingesetzt wird, kommt Oberflächenmesstechnik für die Bestimmung von Gestaltabweichungen höherer Ordnung zum Einsatz. [MARX21; PFEI10; WECK12; WELL19] Die im weiteren Verlauf beschriebenen Vorgehensweisen werden in diesem Zusammenhang anhand eines KMG entwickelt und validiert.

Wie in Abbildung 4.9 dargestellt, erfassen KMG Messpunkte auf der Werkstückoberfläche, welche zuvor durch die Zerlegung des konstruierten Werkstücks in ideale Geometrielemente definiert wird. Die punktweise Abtastung der realen Flächenelemente kann taktil oder optisch erfolgen. Im ersten Schritt werden die erfassten Oberflächenpunkte in ein gemeinsames Koordinatensystem transformiert. Dies ermöglicht die Approximation der Istgeometrie durch ideale Ersatzgeometrielemente. Die notwendige Ausgleichsrechnung für die Approximation der Istgeometrielemente basiert üblicherweise auf dem Verfahren nach Gauß, welchem als mathematisches Kriterium die kleinste Fehlerquadratsumme zugrunde liegt. Dieses kann für jegliche Geometrielemente als Ausgleichsverfahren genutzt werden und bietet eine schnelle Datenverarbeitung sowie eine hohe Stabilität der Messergebnisse. Je nach Anwendung kommen zusätzlich die Hüll-, Pferch- oder die Tschebyscheffbedingung zum Einsatz. Die berechneten Ersatzelemente reagieren empfindlich auf

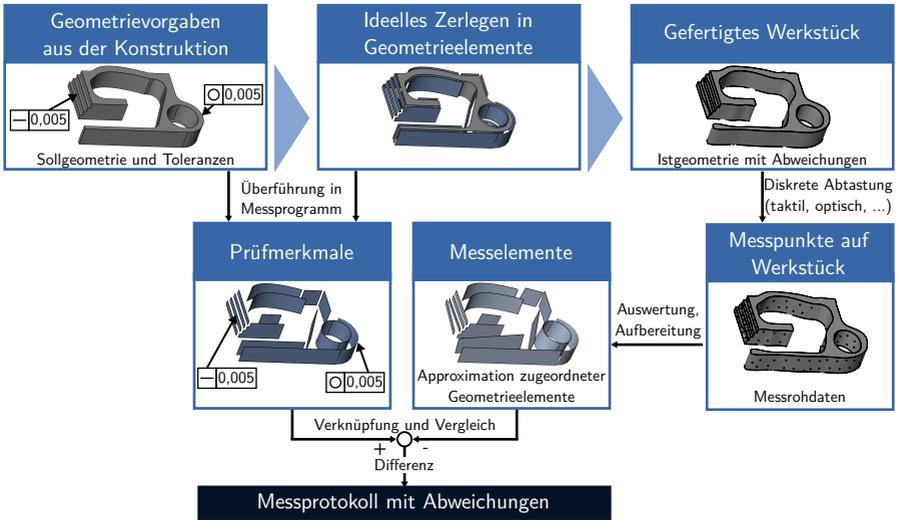


Abbildung 4.9: Grundprinzip der Koordinatenmesstechnik, eigene Darstellung basierend auf [WECK12; WELL19].

Ausreißer (z. B. durch Verschmutzung), weshalb es sinnvoll sein kann, Ausreißer durch ein gewähltes Verfahren herauszufiltern. Dafür wird die Streuung der Messpunkte um das Ersatzelement analysiert und mithilfe von mathematischen Verfahren stark abweichende Punkte entfernt, wodurch eine Glättung der aus den Messpunkten resultierenden Kontur stattfindet. Zusätzlich ermöglicht die Filterung eine Trennung von Gestaltabweichungen unterschiedlicher Ordnungen. Die zugeordneten geometrisch idealen Geometrielemente können direkt zur Informationsbereitstellung genutzt werden oder als Bezug für die Bestimmung von Maßen bzw. zur Berechnung von Form- und Lageabweichungen dienen. Die Ermittlung der Abweichung zwischen idealer Sollgeometrie und dem approximierten Istgeometrieelement erfolgt durch den prüfzielabhängigen Vergleich der entsprechenden Messelemente mit den zuvor im Prüfplan festgelegten Prüfmerkmalen. Je nach Art des Prüfmerkmals, können die Abweichungen direkt am zugeordneten Geometrieelement (z. B. Durchmesser, Rundheit, Geradheit, Ebenheit) oder durch die Verknüpfung von zwei bzw. mehreren idealisierten Geometrieelementen bestimmt werden. Das Prüfergebnis wird schließlich in Form eines Messprotokolls, üblicherweise im PDF-Format, ausgegeben. [MARX21; NEUM05; PFEI10; WECK12; WELL19]

Messprotokolle enthalten neben den Messwerten zusätzliche Informationen zum gemessenen Produkt, Auftrag, Kunde sowie zu den Prüfmerkmalen und deren Bezeichnung. Zusätzlich sind das Istmaß, die obere (OTOL, engl.: UPP-TOLLIM) und untere Toleranzgrenze (UTOL, engl.: LOWTOLLIM) sowie die Nennwerte und die Messunsicherheit zu jedem gemessenen Merkmal im Prüfprotokoll eingetragen. Die grafische Darstellung der Ergebnisse ermöglicht eine erleichterte Beurteilung z. B. in „gut“ und „schlecht“. [MARX21; WECK12] Nachteil des im PDF-Format vorliegenden Messprotokolls, ist die komplexe automatisierte Extraktion und Übertragung der enthaltenen Informationen, da Darstellung und Struktur für eine schnelle Beurteilung der Qualität durch entsprechendes Fachpersonal ausgelegt sind. Um dem entgegenzuwirken eignen sich im Verlauf der Zeit verschiedene entstandene zeichencodierte Transferformate, welche teilweise herstelllerspezifisch aufgebaut sind. Jedoch konnte sich für die Strukturierung von Messprotokollen keine Standardisierung durchsetzen. [NEUM05; WECK12]

Für die Übertragung der enthaltenen qualitätsrelevanten Informationen aus dem Messprotokoll dient im Rahmen dieser Arbeit ein entwickeltes Datenaustauschformat basierend auf einer CSV-Dateistruktur. Die erfassten Werte aus dem Messprotokoll werden für jedes gemessene Bauteil in Form einer eigenen CSV-Datei gesichert. Die Kopfzeile (engl.: Header) enthält die Bezeichner für die entsprechenden kommaseparieren Werte in den nachfolgenden Zeilen. Jede Zeile enthält die Daten zu einem ausgewerteten Prüfmerkmal. Jedes Prüfmerkmal ist dabei einem Geometrieelement am Bauteil zugeordnet. Für die eindeutige Zuordnung der Zeileneinträge zum jeweiligen physikalischen Produkt enthält jede Zeile acht Einträge, welche über die Bezeichner interpretiert werden können. Tabelle 4.4 fasst die Bezeichner und deren Erläuterung zusammen.

Aufgrund der Limitationen durch die eingesetzte Messsoftware PC-DMIS der Hexagon Metrology GmbH für die Strukturierung der Messdatenexporte werden die exportierten Daten mittels eines in Python programmierten Softwarebausteins in die vorgestellte Datenstruktur transformiert. Dieser Softwarebaustein ermöglicht die automatisierte Restrukturierung der von PC-DMIS erstellten CSV-Exportdatei infolge einer durchgeführten Bauteilvermessung. Die entstehenden Dateien werden im Hintergrund als CSV-Dateien in einem vom Nutzer gewählten Verzeichnis abgelegt. Das Verzeichnis kann sich dabei sowohl lokal auf dem Messrechner als auch per Ethernetverbindung in einem Netzlaufwerk des Unternehmensnetzwerks befinden. Der Softwarebaustein greift auf das Verzeichnis zu und liest neu erzeugte CSV-Messprotokolle ein, transformiert diese in die oben genannte Datenstruktur und speichert

Tabelle 4.4: Datenpunkte im entwickelten CSV-Austauschformat für qualitätsrelevante Messdaten.

Bezeichner	Erklärung
MATNUM	Materialnummer
SERNUM	Seriennummer
geomELEMENT	zugehöriges Geometrieelement
TOLNAME	Bezeichner Prüfmerkmal
AXIS	zugehörige Bezugsachse
NOMVAL	Nennmaß
MEAS	Istmaß
UPPTOLLIM	obere Toleranzgrenze
LOWTOLLIM	untere Toleranzgrenze
DEV	Abmaß
OUTTOL	Abmaß außerhalb der Toleranzgrenze

die Ergebnisse jeweils in CSV-Dateien ab. Diese Vorgehensweise ermöglicht einen automatisierten Datentransfer und ist unabhängig von der genutzten Messsoftware. Der entwickelte Softwarebaustein muss bei Nutzung einer unterschiedlichen Messsoftware lediglich einmalig an das proprietär gestaltete Exportformat der entsprechenden Software angepasst werden.

Die aufbereiteten Qualitätsdaten liefern zusammen mit den Daten aus der NC die notwendigen realen Daten aus Teilefertigung und nachgelagerter Qualitätssicherung zur Beantwortung von hervorgehenden Fragestellungen aus FF1 und bilden die Grundlage für datenbasierte Qualitätsprognosen in der Zerspanung. Durch technologiespezifische Einordnung der hochfrequenten Signaldaten (vgl. Kapitel 4.2.2) und deren Zusammenführung mit NC-Variablen (vgl. Kapitel 4.2.3) bei der Erfassung, wird die kontextsensitive Interpretation der Daten aus der Maschinensteuerung ermöglicht. Für die prozess- bzw. geometrielementindividuelle Verknüpfung der Zeitreihendaten aus dem Prozess mit den Qualitätsdaten werden im folgenden Kapitel zusätzliche Daten aus Konstruktion und Arbeitsvorbereitung miteinbezogen, um eine Lösung für die automatisierte Signaldatensegmentierung zu entwickeln und abschließend die erste Forschungsfragestellung zu beantworten.

# 5 Lösungsgrundlagen

Die in diesem Kapitel entwickelten Lösungsgrundlagen bilden das Fundament für die Umsetzung der prozessparallelen Qualitätsprognose. Dabei stehen zunächst mit der zweiten Forschungsfrage verbundene Lösungskonzepte für die Kontextualisierung und Vorverarbeitung der NC-Signaldaten im Fokus. Das im Folgenden vorgestellte Kontextualisierungskonzept (Kapitel 5.1) ermöglicht, ausgehend von der CAD/CAM-Prozesskette, die Anreicherung der NC-Signaldaten mit, für die weitere Datenverarbeitung notwendigen, kontextbeschreibenden Metadaten. Der darauf aufbauend entwickelte Algorithmus für die Feinsegmentierung (Kapitel 5.2) extrahiert in einem weiteren Schritt der Datenvorverarbeitung automatisiert diejenigen Datenpunkte, welche in die Prognosemodellerstellung einfließen. Diese dienen schließlich als Eingangsgrößen für die entwickelte Prozesskette für das maschinelle Lernen (Kapitel 5.3), welche die notwendigen Teilprozesse für die Erstellung KI-basierter Qualitätsprognosemodelle spanend hergestellter Werkstücke umfasst und damit zur Beantwortung der dritten Forschungsfrage beiträgt. Das Kapitel schließt mit der Generierung von Datensätzen für die nachfolgende Umsetzung der Gesamtlösung ab.

## 5.1 Kontextualisierung der NC-Signaldaten<sup>1</sup>

Das Ziel der Kontextualisierung ist die automatisierte Zuordnung der entstehenden Zeitreihenabschnitte (in diesem Fall die hochfrequenten NC-Signaldaten `hf_data`) zu den entsprechenden Teilprozessen bzw. gefertigten Geometrieelementen am Werkstück. Die Kontextualisierung beschreibt demzufolge die Anreicherung der Signaldaten mit kontextbeschreibenden Metadaten für den zugrundeliegenden Anwendungsfall. Fehlen Kontextinformationen bei der Erfassung interner Signaldaten einer WZM, ist die nachträgliche Zuordnung der rohen Zeitreihensignale zu den jeweiligen Geometrieelementen innerhalb eines Werkstücks lediglich mit hohem manuellen Aufwand möglich. Eine praktikable Umsetzung bei Bearbeitungen von Werkstücken, welche über einfache Bauteile hinausgehen, ist somit nicht mehr realisierbar.

---

<sup>1</sup>Erstveröffentlichung in [FERT22a].

NC-Programme enthalten Angaben zur Erzeugung der Werkzeug- bzw. Werkstückbewegungen, technologische Befehle wie Spindeldrehzahl und Vorschubgeschwindigkeit sowie Anweisungen für einen Werkzeugwechsel bzw. das Schalten von Zusatzfunktionen wie z. B. Kühlschmierpumpe, Späneförderer etc. [BREC21]. Diese abgearbeiteten Befehle liefern entscheidende Informationen für die Interpretation des aktuellen Prozesskontexts. Deren satzweise Aufbereitung innerhalb des Informationsflusses der NC (vgl. Kapitel 2.2.2) durch den NC-Interpreter (vgl. Abbildung 2.7) ermöglicht einen taktsynchronisierten Zugriff auf die verarbeiteten Informationen. Die einzelnen NC-Sätze werden innerhalb der Steuerung nacheinander abgearbeitet und führen zum gewünschten Prozessablauf bzw. zur Fertigung der finalen Werkstückgeometrie. Aktuelle Arbeiten nutzen bereits die abgearbeiteten NC-Sätze für die Anreicherung der Prozessdaten mit Kontextinformationen. Zusätzlich kommen häufig Trigger im NC-Programm in Form von M-Befehlen zum Einsatz für Start und Stopp der Datenaufzeichnung. [CERA22; HÄNE20; SIEM19b; WELL19]

Für die Erstellung von NC-Programmen kommt heutzutage die CAM-Prozessplanung zum Einsatz, welche im ersten Schritt maschinenunabhängig generierte Verfahrenweisungen für das Teileprogramm<sup>2</sup> erzeugt. Ein Prozessor prüft das maschinell erzeugte Teileprogramm auf formale Fehler und berechnet die Bahn des Werkzeug Bezugspunktes. Zusätzlich werden sämtliche geometrische und arithmetische Anweisungen erstellt, technologische Schnittwerte bestimmt und Kollisionsberechnungen durchgeführt. Als Ergebnis steht die CLDATA<sup>3</sup> in Form eines maschinenunabhängigen Zwischencodes. Diese maschinenunabhängigen Daten werden durch den individuell auf die jeweilige Maschine abgestimmten Postprozessor auf die von der Maschinensteuerung interpretierbaren Codierung angepasst. Als praxisrelevante Alternativen dazu, nutzen zahlreiche Programmiersysteme anstelle der CLDATA einen maschinenneutralen an DIN 66025 angelehnten NC-Code, welcher durch entsprechend konfigurierte Postprozessoren in ein maschinenspezifisches NC-Programm überführt wird. [BREC21] Diese Vorgehensweise führt in erster Linie bei komplexeren Werkstücken dazu, dass eine einfache Interpretation der maschinell generierten Verfahrenweisungen nicht mehr möglich ist. Zusätzlich fehlen häufig Informationen zur eindeutigen Identifikation des aktuell gefertigten Werkstücks insbesondere bei Fertigung mehrerer Bauteile innerhalb einer Serie. Die Kontextinformationen für die Zuordnung der Prozesssignale zu den Geometrieelementen am Werkstück liegen in diesem Fall zwar vor,

<sup>2</sup>in einer NC-Programmiersprache erstelltes Quellprogramm [BREC21]

<sup>3</sup>Cutter Location Data [DIN66215-1]

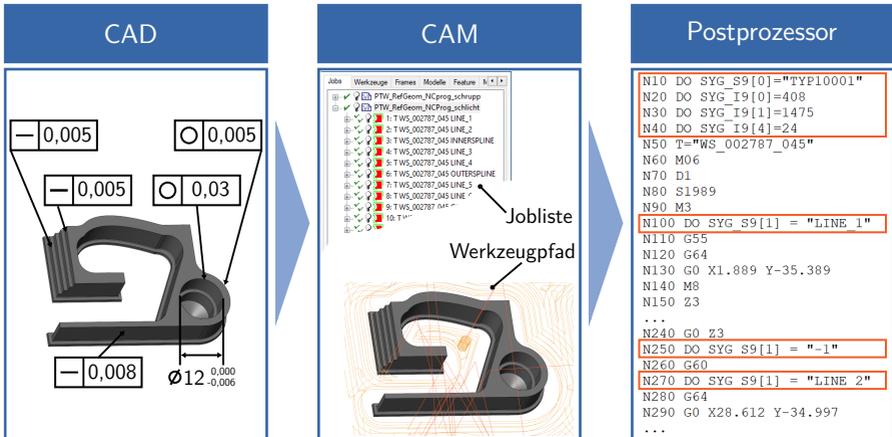


Abbildung 5.1: Schematische Vorgehensweise für die Anreicherung des NC-Programms mit zusätzlichen Metainformationen (orangefarbene Kästen).

jedoch steigt der dafür notwendige manuelle Aufwand mit der Anzahl und der Komplexität an NC-Sätzen. Zudem liegen die Geometriemerkmalen lediglich implizit in den Verfahrensanweisungen vor und erschweren zusätzlich die angestrebte Zuordnung, wodurch eine automatisierte Kontextualisierung über beliebige Werkstückgeometrien hinweg nicht umsetzbar ist.

Das NC-Programm bildet jedoch als Eingangsdatenquelle eine ideale und einfache Möglichkeit, Informationen von vorgelagerten Prozessen entlang der CAD/CAM-Prozesskette in die Werkzeugmaschinensteuerung zu transferieren. Dies schafft die Grundlage für die im Rahmen dieser Arbeit umgesetzte Methode zur Kontextualisierung interner Maschinensignale. Dazu wird das NC-Programm mit zusätzlichen werkstück- und prozessbeschreibenden Metainformationen angereichert. Abbildung 5.1 stellt die schematische Vorgehensweise für die Anreicherung des NC-Programms dar. Die als Minimalzusammenstellung identifizierten Kontextvariablen sind die *Materialnummer*, *Seriennummer* sowie eindeutige Identifikatoren für die Geometrielemente, im weiteren Verlauf als *geomElemID* bezeichnet, eines Werkstücks. Die Materialnummer ermöglicht die eindeutige Identifikation für ein Material. In diesem Zusammenhang erhält jeder Werkstücktyp eine individuelle Materialnummer. Die Seriennummer dient nach ISO 8000 der „Identifizierung eines einzelnen Vorkommens eines Erzeugnisses“ [ISO8000-2] und repräsentiert somit die

Identifikationsnummer für ein einzelnes physikalisches Bauteil einer Serie. Bei der Erstellung des maschinenspezifischen NC-Programms werden Material- und Seriennummer in den Programmkopf des NC-Programms geschrieben. Darüber hinaus ist es möglich, an dieser Stelle je nach Bedarf und produktionsorganisatorischer Umsetzung zusätzliche Variablen, z. B. Auftragsnummer, Charge, etc. zu hinterlegen. Die `geomElemIDs` werden unmittelbar vor dem ersten zugehörigen NC-Satz der entsprechenden Zerspanungsoperation für das jeweilige Geometrieelement gesetzt. Am Ende der zu einem Geometrieelement gehörigen NC-Sätze ist diese Variable wieder auf den Ursprungszustand (engl.: Default Value) zu setzen. Die Zuweisung der Variablenwerte erfolgt während der CAM-Programmierung und wird schließlich automatisiert vom Postprozessor, welcher zuvor für Kontextualisierung angepasst wurde, in das NC-Programm übernommen. Dabei ist darauf zu achten, dass die finalen Schlichtprozesse der aus der Prüfplanung hervorgehenden qualitätsrelevanten Geometrieelemente bei der CAM-Programmierung mit entsprechend individuellen NC-Operationen (engl.: Jobs) programmiert werden.

Um die Kontextvariablen bzw. den dazugehörigen Speicherbereich in der Steuerung synchron zu den Bearbeitungssätzen zyklisch schreiben und lesen zu können, dienen innerhalb einer SIEMENS SINUMERIK-Steuerung anwenderdefinierte Variablen für Synchronaktionen, sogenannte synchronaktionsfähige GUD-Variablen<sup>4</sup> (S-GUD). Die Anzahl der zur Verfügung stehenden S-GUD-Variablen wird datentyp- und zugriffsspezifisch über die in Tabelle 5.1 aufgeführten Maschinendaten parametrisiert. Zusätzliche Parametrierungsoptionen sind [SIEM18e] zu entnehmen. Dafür stehen innerhalb der Steuerung für jeden der neun verfügbaren GUD-Bausteine sechs Datentypen zur Verfügung. Tabelle 5.1 führt lediglich die im Rahmen dieser Arbeit parametrisierten S-GUD-Variablen für den neunten Baustein auf. Als letzter zur Verfügung stehender Baustein ist dieser oftmals durch den Maschinenhersteller werkseitig noch nicht in Benutzung, wodurch das Risiko minimiert wird, bei nachträglicher Anpassung, bestehende Speicherbereiche innerhalb der Steuerung zu überschreiben. Grundsätzlich ist es jedoch möglich einen beliebigen Baustein für die eingeführten S-GUD-Variablen zu nutzen. Die parametrisierten Variablen `SYG_I9[0]` entsprechen in diesem Fall der Seriennummer, `SYG_S9[0]` der Materialnummer und `SYG_S9[1]` der `geomElemID`.

Die Erfassung der eingeführten Kontextvariablen über die `lf_data`-Schnittstelle erfordert die Kenntnis des BTSS-Variablenpfads für S-GUD-Variablen innerhalb der NC. Dieser ist nicht explizit in den Handbüchern des Steue-

---

<sup>4</sup>in allen Programmen geltende anwenderdefinierte Variablen [SIEM18e]

Tabelle 5.1: Parameterierung der S-GUD-Variablen [SIEM18f].

Baustein	Adresse <i>SINUMERIK 840D sl</i>	Anzahl	Variablenname
GUD9	MD 18661[8] \$MM_MM_NUM_SYNACT_GUD_INT	1	SYG_I9[i]
GUD9	MD 18665[8] \$MM_MM_NUM_SYNACT_GUD_STRING	2	SYG_S9[i]
mit i = 0 bis (<Anzahl> - 1)			

Tabelle 5.2: BTSS-Variablenpfade der S-GUD-Kontextvariablen.

Kontextvariable	Variablenname	BTSS-Variablenpfad <i>SINUMERIK 840D sl</i>
Seriennummer	SYG_I9[0]	/NC/_N_CH_GD9_ACX/SYG_I9[1]
Materialnummer	SYG_S9[0]	/NC/_N_CH_GD9_ACX/SYG_S9[1]
geomElemID	SYG_S9[1]	/NC/_N_CH_GD9_ACX/SYG_S9[2]
Indizes steigen beim Auslesen um 1.		

rungsherstellers aufgeführt. Jedoch lassen sich die notwendigen Pfade per Verbindung zum OPC-UA Server der Steuerung auslesen. Die Suche nach „SYG“ im Variablenbaum führt zu den in Tabelle 5.2 aufgeführten OPC-UA-Variablenpfaden, welche in diesem Fall den BTSS-Variablenpfaden der lf\_data-Schnittstelle entsprechen.

Für den Aufruf von anwenderspezifischen NC-Programmen beim Auftreten ausgewählter Ereignisse in der Steuerung existiert innerhalb der Steuerung die Funktion „Ereignisgesteuerte Programmaufrufe“ (PROG\_EVENT). Diese Funktionalität wird im Rahmen der Kontextualisierung genutzt, um zu gewährleisten, dass die gesetzten S-GUD-Speicherbereiche bei „Programm-Ende“ sowie bei „Kanal-Reset“ wieder ihren Ursprungswert annehmen. Im Rahmen dieser Arbeit ist der Ursprungszustand für Integer Variablen auf den Zahlenwert -1 und für String Variablen auf den Wert „-1“ festgelegt. Dafür ist es notwendig den Anwenderzyklus „CYCPE1US.SPF“ anzupassen bzw. falls dieser noch nicht existiert anzulegen und die Befehle für das Zurücksetzen

der parametrisierten Variablen einzupflegen. Der im Rahmen dieser Arbeit angepasste Anwenderzyklus „CYCPE1US.SPF“ ist Anhang A.2 zu entnehmen. Weiterführende Informationen zu den ereignisgesteuerten Programmaufrufen finden sich in [SIEM19a].

## 5.2 Automatisierte Datensegmentierung<sup>5</sup>

Die kontextsensitive Erfassung aus Kapitel 5.1 bildet die Voraussetzung für eine exakte Zuordnung der entstehenden Datenpunkte bei der Schlichtbearbeitung der zugehörigen qualitätsrelevanten Geometrieelemente. Abbildung 5.2 (a) stellt die kontextbasiert gefilterten Positionsdaten des TCP für die Schlichtprozesse aller Geometrieelemente eines Musterwerkstücks dar. Die Zuordnung der Prozessabschnitte basiert gemäß Kapitel 5.1 auf dem mit Kontextdaten angereicherten NC-Programm, wodurch es aus technologischen Gesichtspunkten und Einschränkungen der lf\_data-Schnittstelle nicht möglich ist, ein präzises datenpunktgenaues „Schneiden“ (engl.: „Slicing“) der Zeitreihensegmente durchzuführen. Besonders deutlich wird dies bei einer detaillierteren Betrachtung der Geometrieelemente „gestufte Linienzüge“ (LINESTEPS, vgl. Abbildung 4.3) am Musterwerkstück, welche jeweils die gleiche Länge aufweisen. Der TCP-Pfad weist nach der Filterung für jeden LINESTEP eine unterschiedliche Anzahl an Datenpunkten in den Datenreihen auf. Basierend auf der Prozessführung gilt jedoch, dass die Anzahl an qualitätsrelevanten Datenpunkten für jeden der vier LINESTEPS übereinstimmend sein muss. Als qualitätsrelevante Datenpunkte in diesem Zusammenhang werden diejenigen hf\_data-Datenpunkte definiert, welche zwischen Schneideneintritt und Schneidenaustritt des Prozesses für das jeweilige Geometrieelement entstehen und erfasst werden. Diese Datenpunkte repräsentieren die Werkzeug/Werkstück-Interaktion und enthalten somit die wesentlichen Informationen für die angestrebte Qualitätsvorhersage. Erfasste Datenpunkte vor Schneideneintritt bzw. nach Schneidenaustritt, sowie zu Zustellbewegungen des Werkzeugs gilt es dabei möglichst präzise und automatisiert zu entfernen, um die Datenbasis auf die relevanten Datenpunkte zu reduzieren. Die Nutzung der lf\_data-Schnittstelle für die Kontextualisierung limitiert darüber hinaus die Präzision für die Datensegmentierung, aufgrund der geringen und nicht äquidistanten Abtastrate.

Für die automatisierte Datensegmentierung und Zuordnung der kontextbasiert gefilterten Datenpunkte zu den jeweiligen qualitätsrelevanten Geometrieelementen eines Bauteils dient ein entwickelter Segmentierungsalgorithmus, welcher sich in zwei grundsätzliche Schritte einteilen lässt. Der erste

<sup>5</sup>Erstveröffentlichung in [FERT22b].

Schritt umfasst die Transformation der im MKS vorliegenden Achspositionssignale in das WKS bzw. Koordinatensystem des TCP durch Rückwärtstransformation der kinematischen Kette der WZM [SIEM18d; WELL19]. Diese Transformation ist erforderlich, um im zweiten Schritt die Segmentierung basierend auf den Werkstückgeometriedaten, welche durch die Konstruktion des Werkstücks in Form von CAD-Modellen zur Verfügung stehen, durchzuführen. Die Translation berechnet sich gemäß

$$\begin{pmatrix} x_{\text{trans}} \\ y_{\text{trans}} \\ z_{\text{trans}} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -x_{\text{npv}} \\ 0 & 1 & 0 & -y_{\text{npv}} \\ 0 & 0 & 1 & -(z_{\text{npv}} + l_{\text{wkz}}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{\text{mks}} \\ y_{\text{mks}} \\ z_{\text{mks}} \\ 1 \end{pmatrix}, \quad (5.1)$$

mit  $x_{\text{npv}}$ ,  $y_{\text{npv}}$ ,  $z_{\text{npv}}$  als Werte der NPV für die X, Y und Z-Achse.  $l_{\text{wkz}}$  entspricht dem aktuellen Kompensationswert für die Werkzeuglänge (vgl. Tabelle 4.3), welcher vor dem Prozess extern mit entsprechenden Werkzeugeinstellgeräten oder maschinenintern mit Werkzeugeinmesssystemen bestimmt wird.

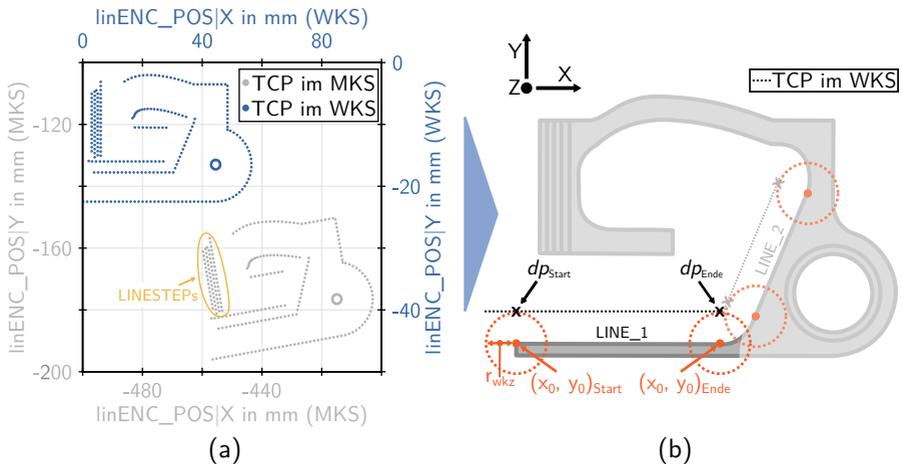


Abbildung 5.2: Schematische Darstellung des Segmentierungsalgorithmus.  
 a) Rücktransformation des TCP vom MKS in das WKS.  
 b) Schematische Darstellung der Segmentierungsvorschrift am Beispiel von LINE\_1.

$$\begin{aligned}
 R_x(\varphi)^T &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) & 0 \\ 0 & -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 R_y(\theta)^T &= \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 R_z(\psi)^T &= \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned} \tag{5.2}$$

Die Rückwärtsrotation geschieht unter Nutzung der Rotationsmatrizen aus Gleichung 5.2 gemäß [DIEB06; SLAB99]

$$\begin{pmatrix} x_{\text{wks}} \\ y_{\text{wks}} \\ z_{\text{wks}} \\ 1 \end{pmatrix} = R_x(\varphi_{\text{npv}})^T \cdot R_y(\theta_{\text{npv}})^T \cdot R_z(\psi_{\text{npv}})^T \cdot \begin{pmatrix} x_{\text{trans}} \\ y_{\text{trans}} \\ z_{\text{trans}} \\ 1 \end{pmatrix}, \tag{5.3}$$

mit  $\varphi_{\text{npv}}$ ,  $\theta_{\text{npv}}$ ,  $\psi_{\text{npv}}$  als eingestellte Werte für die Drehung der NPV um die Achsen X, Y, Z. Die Rotationsmatrizen  $R_x(\varphi)^T$ ,  $R_y(\theta)^T$ ,  $R_z(\psi)^T$  basieren auf der Rotation nach Roll-, Nick-, und Gier-Winkel (engl.: Roll, Pitch, Yaw Angle), welche von den im Rahmen dieser Arbeit verwendeten Steuerungen für die rotatorische Transformation genutzt werden. Die verwendete Einstellung, wie die Drehungen um die drei Geometrieachsen festgelegt sind, lässt sich aus dem IBN-Archiv (vgl. Tabelle 5.3) entnehmen. Die Transformationen werden auf die positionsverwandten Signale der Signalgruppen DES\_POS, CTRL\_POS, linENC\_POS, rotENC\_POS angewendet. Zusätzlich werden die rotatorischen Transformationen für die Signalgruppen CMD\_SPEED, CONT\_DEV, CTRL\_DIFF und CTRL\_DIFF2 durchgeführt. Insgesamt entstehen dadurch neben dem TCP-Pfad in Soll- und Istposition (vgl. Abbildung 5.2 (a) „blau“) zusätzlich Geschwindigkeiten und Regelabweichungen im WKS. Auf Grundlage der transformierten Positionssignale erfolgt die Segmentierung der Zeitreihensignale. Dafür werden die Start- und Endkoordinaten der Grenzfläche für jedes qualitätsrelevante Geometrielement im WKS benötigt, welche im Voraus aus dem CAD-Modell des Gesamtwerkstücks in

Tabelle 5.3: Maschinendatum für Drehreihenfolge der Geometrieachsen im FRAME [SIEM20].

Maschine	Adresse <i>SINUMERIK 840D sl</i>	Wert	Winkel
DMC 850V	MD 10600 \$MN_FRAME_ANGLE_INPUT_MODE	1	RPY
G350	MD 10600 \$MN_FRAME_ANGLE_INPUT_MODE	1	RPY

Form von X, Y, und Z-Koordinaten extrahiert werden. Für jedes Geometrieelement dient jeweils der extrahierte Start- und Endpunkt als Mittelpunkt eines berechneten Kreises mit dem Radius des eingesetzten Werkzeuges. Ziel ist es, den Schnittpunkt zwischen dem rücktransformierten TCP-Pfad und den berechneten Kreisen zu bestimmen, welche jeweils den ersten (Start) und den letzten (Ende) qualitätsrelevanten Schnitt der Werkzeugschneiden für das Geometrieelement darstellen. Für die Bestimmung der Schnittpunkte erfolgt für jeden vorgefilterten (gemäß Kontextualisierung aus Kapitel 5.1) und transformierten Positionsdatenpunkt die Berechnung des Abstands  $d$  zum jeweiligen Kreis gemäß

$$d^2 = ((p_x - x_0)^2 + (p_y - y_0)^2) - r_{\text{wkz}}^2, \quad (5.4)$$

mit  $x_0, y_0$  jeweils als Koordinaten des Start- bzw. Endpunkts des betrachteten Geometrieelements und  $r_{\text{wkz}}$  als Werkzeugradius des eingesetzten Schaftfräswerkzeugs. Für die Positionskoordinaten  $p_x$  und  $p_y$  werden die erfassten DES\_POS-Signalwerte genutzt. Diese bilden den geplanten Prozessablauf in Form von interpolierten Sollpositionsdaten ab und unterliegen somit keiner zusätzlichen Beeinflussung durch etwaige Störgrößen. Stehen die DES\_POS-Signale nicht zur Verfügung, können alternativ auch linENC\_POS- bzw. rotENC\_POS-Signale mit leichten Abstrichen in der Genauigkeit für die Segmentierung genutzt werden. Die finale Bestimmung der Segmentierungsgrenze für Start und Ende erfolgt schließlich durch die Ermittlung des Datenpunkts  $dp$  mit dem geringsten Abstand  $d$ , folglich des Datenpunkts mit dem minimalen Abstand zwischen Werkzeugumfang und geschlichteter Werkstückoberfläche gemäß

$$dp = \min(|d_0^2|, |d_1^2|, |d_2^2|, \dots, |d_n^2|), \quad (5.5)$$

mit  $n$  als Anzahl der vorgefilterten Datenpunkte des betrachteten Geometrieelements. Zusammengefasst werden die Schnittpunkte zwischen einem Kreis

mit dem Radius des eingesetzten Werkzeugs und des 2D-TCP-Pfads bestimmt, welche die Grenzen des Werkzeugeingriffs markieren. Abbildung 5.2 (b) illustriert schematisch die Funktionsweise des Algorithmus am Beispiel von LINE\_1. Im Fall übereinstimmender Start- und Endkoordinaten, wie beispielsweise beim INNERCIRCLE, wird die Zeitreihe vor der Ermittlung der Schnittpunkte mittig geteilt und mit der entsprechenden ersten Hälfte bzw. der zweiten Hälfte der Punktemenge werden die Schnittpunkte wie zuvor beschrieben ermittelt.

Es resultieren präzise gefilterte Zeitreihensegmente, welche jeweils dem entsprechenden Geometrielement an einem Werkstück zugeordnet sind. Zusammen mit den Qualitätsdaten (vgl. Kapitel 4.2.4) entstehen durch Kontextualisierung und Segmentierung hochqualitative Datensätze, um damit KI-basierte Prognosemodelle für die Prädiktion der Werkstückqualität zu erstellen und die zweite Forschungsfragestellung zu beantworten.

## 5.3 Prozesskette für das maschinelle Lernen

Die Prozesskette für das maschinelle Lernen (MLPK) berücksichtigt die wesentlichen Teilprozesse von der Merkmalsextraktion, über die -selektion hin zum Modelltraining und der Evaluation der Prognosegüte im Zusammenhang mit der dritten Forschungsfragestellung. Dabei wird jeder Teilprozess an den zugrundeliegenden Anwendungsfall adaptiert, sodass die abschließende Implementierung eine automatisierte Modellbildung, ausgehend von den zuvor gewonnenen segmentierten Datensätzen, zulässt. Initial erfolgt dafür die Herleitung des maschinellen Lernproblems im Zusammenhang mit der angestrebten Qualitätsüberwachung.

### 5.3.1 Problembeschreibung im Kontext der Qualitätsüberwachung

Die grundlegende Fähigkeit von maschinellen Lernalgorithmen, aus Ein- und Ausgangsdaten das notwendige Wissen für die jeweilige Anwendung in Form von Modellen zu generieren, entfaltet ihr höchstes Potenzial insbesondere bei einem Einsatz in komplexen Anwendungen, bei denen die zugrundeliegenden Zusammenhänge zwischen Eingang und Ausgang nicht umfänglich bekannt sind bzw. eine exakte Modellierung lediglich mit sehr hohem Aufwand oder gar nicht möglich ist (vgl. Kapitel 2.1.2). Die im Rahmen dieser Arbeit betrachtete Qualitätsbewertung auf Basis maschineninterner Signaldaten weißt aufgrund diverser Effekte und Störgrößen solche komplexen physikalischen Zusammenhänge auf, welche sich nicht vollständig analytisch beschreiben lassen. Die genannte Eigenschaft prädestiniert maschinelle Lernverfahren für die Erstellung von Prozesswissensmodellen, um die komplexen Wirkzusammenhänge während eines Zerspanprozesses abzubilden (vgl. Kapitel 2.3.2) und somit das Ziel einer automatisierten In-Prozess-Bewertung der Werkstückqualität zu erreichen. Aufgrund ihrer breiten Anwendbarkeit und der Vielzahl an existierenden Algorithmen hat sich eine grundsätzliche Einteilung (vgl. Abbildung 5.3) in die drei Kategorien überwachtes Lernen, unüberwachtes Lernen und verstärkendes Lernen etabliert. [BUXM21; WROB13]

Algorithmen aus der Kategorie des **überwachten Lernens** (engl.: Supervised Learning) nutzen für die Modellbildung „beschriftete“ Daten während des Trainingsvorgangs. Dementsprechend erhalten die Algorithmen Trainingsdatensätze, welche zu den jeweiligen zugeführten Eingabewerten, die gewünschten Lösungen bzw. Ausgabewerte („Labels“) enthalten. Die Modellbildung erfolgt durch die Erkennung und das Lernen der Zusammenhänge zwischen

Eingabe- und Ausgabewerten. Im Anschluss an das Training dient ein dem Algorithmus unbekannter, zuvor aus dem vorhandenen Datensatz abgespaltener Testdatensatz zur Überprüfung der Prognosegüte und Evaluation des trainierten Modells. Das überwachte Lernen verfolgt das Ziel einer möglichst präzisen Prognose für zwei grundsätzlich zu unterscheidende überwachte Lernaufgaben. Die Klassifikation zielt darauf ab, die Eingabedaten zu klassifizieren und diese dementsprechend zuvor definierten Klassen bzw. diskreten Zielwerten zuzuordnen. Ziel von Regressionsaufgaben hingegen ist es, eine numerische Zielgröße bzw. kontinuierliche reelle Zielwerte vorherzusagen. [BUXM21; GÉRO20; SCHO21a; WROB13]

Lernverfahren aus dem Bereich des **unüberwachten Lernen** (engl.: Unsupervised Learning) erhalten während der Trainingsphase keine Labels zu den Trainingsdaten und zielen darauf ab, Muster sowie unbekannte Zusammenhänge und Strukturen in den Eingangsdaten aufzudecken. Das unüberwachte Lernen ist dementsprechend eher deskriptiver Natur und die Algorithmen kommen insbesondere bei Komprimierungsverfahren zum Einsatz, um entweder irrelevante bzw. unwichtige Merkmale aus den Daten herauszufiltern oder wesentliche Zusammenhänge und Abhängigkeiten in großen Datenmengen zu finden. Typische unüberwachte Lernaufgaben sind die Dimensionsreduktion, die Anomaliedetektion oder die algorithmische Visualisierung von komplexen Datenmengen. [BUXM21; GÉRO20; WROB13]

Die letzte der drei Kategorien stellt das **bestärkende Lernen** (engl.: Reinforcement Learning) dar. Innerhalb des bestärkenden Lernens lernt der Algorithmus bzw. Agent optimale Handlungsvorschriften aus dessen gemachten Erfahrungen während der Interaktion mit einer Umgebung bzw. Umwelt. Die Grundlage für das Erlernen der bestmöglichen Handlungsstrategie bilden positive Anreize bzw. Belohnungen und Bestrafungen für die Wahl einer bestimmten Handlung (Aktion). Die Intention beim Trainieren des Algorithmus ist es, diejenigen Aktionen und Handlungsanweisungen zu finden, welche die Belohnungs- bzw. Anreizfunktion maximieren und so die optimale Lösung für die vorliegende Aufgabenstellung darstellen. [BUXM21; GÉRO20; WROB13]

Übertragen auf die In-Prozess-Bewertung der gefertigten Werkstückqualität lässt sich die zugrundeliegende Aufgabenstellung in ein überwachtes Lernproblem überführen. Die prozessparallel aufgezeichneten Signaldaten liegen nach erfolgter Kontextualisierung (vgl. Kapitel 5.1) und Feinsegmentierung (vgl. Kapitel 5.2) für jedes qualitätsrelevante Geometrieelement in Form multivariater, zu jedem qualitätsrelevanten Geometrieelement zugeordneter Zeitreihenabschnitte vor, welche nach Abbildung 5.3 die Datenpunkte bzw. Instan-

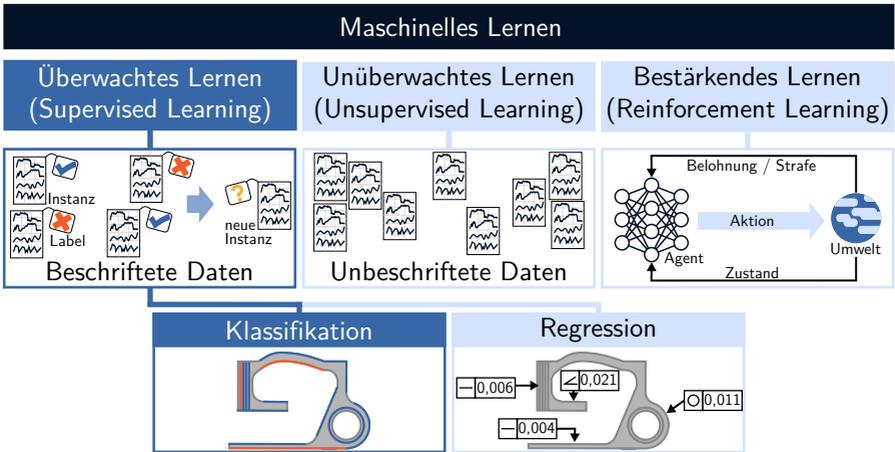


Abbildung 5.3: Einteilung maschineller Lernverfahren.

zen des Datensatzes darstellen. Die Ermittlung der erreichten Werkstückqualität mittels KMT liefert zu jedem Datenpunkt je nach Anzahl der jeweiligen Prüfmerkmale  $1 \dots n$  Maße, welche als Label des Datensatzes fungieren. Diese können unverarbeitet als kontinuierliche Zielwerte den Algorithmen für die Umsetzung von Regressionen vorgegeben oder durch Abgleich mit den Toleranzvorgaben in die Klassen „in Ordnung“ (i. O.) und „nicht in Ordnung“ (n. i. O.) eingeteilt werden. Es resultiert ein binäres Klassifikationsproblem.

Die wesentliche Hauptaufgabe in der Qualitätsüberwachung besteht darin die Qualität der gefertigten Produkte hinsichtlich „gut“ / „schlecht“ zu prüfen, um n. i. O.-Werkstücke frühzeitig zu erkennen und zu verhindern, dass diese in die nächsten Schritte der Produktion oder zum Kunden gelangen (vgl. Kapitel 2.3.1 und 4.2.4). Zusätzlich hat aus produktionstechnischer Sicht eine schnelle und zuverlässige klassifizierende Aussage über die Maßhaltigkeit des gefertigten Werkstücks die höchste Relevanz für die Bewertung der Qualität. Aus diesen Gründen wird im weiteren Verlauf dieser Arbeit die Zielstellung in ein Klassifikationsproblem aus dem Bereich des überwachten Lernens überführt. Das Ziel ist es, innerhalb eines Werkstücks die qualitätsrelevanten Geometrielemente hinsichtlich i. O. bzw. n. i. O. (vgl. Abbildung 5.3) mittels Modellvorhersagen prozessparallel zu prüfen, um schnelle Reaktionen für den Produktionsablauf zu ermöglichen.

### 5.3.2 Merkmalsextraktion

Die Umsetzung eines intelligenten Überwachungssystems für Bearbeitungsprozesse (vgl. Kapitel 2.2.4) sieht im Anschluss an die Identifikation und Erfassung der relevanten Signale, die Signalaufbereitung durch Filterung, Verstärkung und Segmentierung vor. Die Hauptnutzung der erfassten Signale aus der internen Sensorik besteht darin, möglichst präzise die entsprechenden Größen für die Lageregelung widerzuspiegeln, um die Aufgabe der Achsregelung durchzuführen. Deshalb stellen die aus dem Lageregelkreis erfassten Signaldaten im Sinne der Signalverarbeitung bereits aufbereitete Signale dar. Die Notwendigkeit einer weiteren Aufbereitung mittels Filterung oder Verstärkung entfällt somit. Erforderlich ist lediglich eine Segmentierung der vorliegenden Zeitreihendaten bezogen auf den Prozesskontext, welche bereits in Kapitel 5.2 beschrieben und für jedes zu untersuchende Geometrieelement an einem Bauteil umgesetzt ist. Für die Reduktion der Dimensionen der vorliegenden multivariaten Zeitreihensegmente, im weiteren Verlauf als Rohdaten bezeichnet, dient die Extraktion von Merkmalen, welche die enthaltenen Informationen der jeweiligen Zeitreihen in einer komprimierten Anzahl an zeitunabhängigen Skalaren wiedergeben.

Die Literatur, ausgehend von Kapitel 2.2.5, empfiehlt für Prozessüberwachungsaufgaben die Berechnung von Merkmalen im Zeit-, Frequenz- und Zeit-Frequenzbereich. Dabei existieren starke Unterschiede im Hinblick darauf, welche Merkmale sich für den spezifischen Anwendungsfall eignen. Diese gilt es größtenteils manuell für den vorliegenden Prozess zu identifizieren und auszuwählen. Allgemeine Aussagen können hierfür aufgrund der zahlreichen Prozesseinflussgrößen jedoch nicht getroffen werden. Um den manuellen Aufwänden entgegen zu wirken und einen möglichst hohen Automatisierungsgrad der Merkmalsextraktion zu erreichen, erfolgt die Berechnung der Merkmale in dieser Arbeit mithilfe der Softwarebibliothek (engl.: Library) TSFEL [BARA20], welche in der Programmiersprache Python zur Verfügung steht. Mithilfe einer globalen JSON-Konfigurationsdatei (`features.json`) erfolgt die Definition der zu berechnenden Merkmale auf dem vorliegenden Datensatz. Dabei ist eine Einteilung der Merkmale nach Domäne (engl.: Domain) in statistische (engl.: statistical) und zeitbasierte (engl.: temporal) Kennwerte aus dem Zeitbereich sowie spektrale (engl.: spectral) Merkmale, welche auf den mittels FFT transformierten Zeitreihen berechnet werden, vorgesehen. Diese Funktionalität wird genutzt, um die aus der Literatur hervorgehenden und domänenspezifisch relevanten Merkmale aus den Rohdaten zu berechnen.

Die implementierten Merkmale sind in den Tabellen A.3 und A.4 im Anhang zusammengefasst.

Für die Berücksichtigung der Informationen aus dem Zeit-Frequenzbereich erfolgt die Erweiterung von TSFEL um die Domäne „temporal-spectral“. Zum Einsatz kommen die diskrete Wavelet-Transformation (engl.: Discrete Wavelet Transform) (DWT) und die Hilbert-Huang-Transformation (HHT). Die DWT zerlegt das Eingangssignal durch wiederholte Hoch- und Tiefpassfilterung in einzelne Frequenzbereiche. Nach jedem Transformationsschritt liegen in der Dekompositionsebene (engl.: Decomposition Level) die hochpassgefilterten Signalanteile kodiert als Wavelet-Koeffizienten vor und das Ausgangssignal der Tiefpassfilterung dient als Ausgangsbasis für die nächste Transformationsstufe. Der resultierende Vektor der Wavelet-Koeffizienten in jeder Dekompositionsebene enthält halb so viele Einträge wie das Ursprungssignal. Die Grundlage dafür bildet die Multiplikation der Wavelet-Funktion (auch als Basis- oder Mutter-Wavelet bezeichnet)

$$\psi_{\tau,s} = \frac{1}{\sqrt{s}} \cdot \psi\left(\frac{t-\tau}{s}\right), \quad (5.6)$$

mit dem Ausgangssignal  $x(t)$  für die kontinuierliche Wavelet-Transformation (engl.: Continuous Wavelet-transform) (CWT). Für eine Wavelet-Funktion existieren eine Vielzahl an unterschiedlichen Basis-Wavelets. Durch Verschiebung des Wellenpakets  $\psi$  entlang der Zeitachse erfolgt die zeitliche Auflösung des Signals. Dabei beinhaltet der Verschiebungsparameter  $\tau$  den Aufenthaltsort entlang der Zeitachse. Der Skalierungsparameter  $s$  stellt ein Maß für den analysierten Frequenzbereich dar. Durch dessen Variation wird das Wellenpaket gestreckt bzw. gestaucht. Das Integral der erzeugten Funktion entspricht dem Wavelet-Koeffizienten

$$c(\tau, s) = \frac{1}{\sqrt{|s|}} \cdot \int_{-\infty}^{\infty} x(t) \cdot \psi\left(\frac{t-\tau}{s}\right) dt. \quad (5.7)$$

Der Unterschied der DWT zur CWT liegt in der Verwendung diskreter Werte für Skalierungs- und Verschiebungsparameter. Die DWT benötigt somit deutlich weniger Rechenzeit als die CWT und bietet umfassende Möglichkeiten für die Auswertung von Sensorsignalen, weshalb diese in der praktischen Anwendung als primäre Wahl zum Einsatz kommt. [ESSE01; KLOC18; MERT13]

Die Implementierung der DWT erfolgt mit der Funktion `pywt.wavedec()` aus dem Softwarepaket `PyWavelets`. Als Eingabeargumente lassen sich das

zu untersuchende Signal, Art des Basis-Wavelets (**wavelet**) und die Anzahl an Dekompositionsebenen (**level**) definieren. Ausgangsseitig steht der hochpassgefilterte Signalanteil in Form des Wavelet-Koeffizientenvektors für jede Dekompositionsebene sowie das Residuum in Form des tiefpassgefilterten Signalanteils. [LEE19] Für die Ermittlung charakteristischer Merkmale schließt sich aufgrund der relativ großen Anzahl an Koeffizienten bzw. Datenpunkten die Berechnung von Kennwerten für die Koeffizientenvektoren jeder Dekompositionsebene sowie dem Residuum an. In Anlehnung an [TET10] erfolgt die Berechnung von Mittelwert, Effektivwert, Standardabweichung, Kurtosis, Crest-Faktor und der Peak-to-Peak-Distanz. Anhang A.4 zeigt die implementierten Funktionen zur Erweiterung von TSFEL um die DWT.

Ein weiteres Verfahren der Signalverarbeitung innerhalb der Zeit-Frequenzanalyse ist die HHT [TET10]. Ausgangspunkt für das zweistufige Verfahren ist die empirische Modenzerlegung (engl.: Empirical Mode Decomposition) (EMD) [HUAN98]. Die EMD zerlegt das vorliegende Signal in eine endliche Anzahl intrinsischer Modenfunktionen (engl.: Intrinsic Mode Functions) (IMF), basierend auf dem Sifting-Algorithmus. Für die Identifikation einer IMF werden zunächst alle lokalen Maxima im Signal über einen kubischen Spline zur oberen Einhüllenden verbunden. Analog entsteht die untere Einhüllende unter der Nutzung der lokalen Minima. Anschließend erfolgt die Berechnung des Mittelwerts zwischen der oberen und unteren Einhüllenden für jeden Zeitpunkt des vorliegenden Signals, wodurch die resultierende Zeitreihe der Mittelwerte  $m_1$  entsteht. Durch Differenzbildung zwischen dem Ursprungssignal  $x(t)$  und  $m_1$  entsteht die erste Komponente

$$h_1(t) = x(t) - m_1. \quad (5.8)$$

Erfüllt  $h_1(t)$  die Bedingungen für eine IMF, wird  $h_1$  als IMF gespeichert. Die erste Bedingung besagt, dass die Anzahl an Extremstellen und Nulldurchgängen sich höchstens um 1 unterscheiden darf. Als zweite Bedingung für eine IMF gilt, dass der Mittelwert zwischen beiden durch die lokalen Maxima und Minima entstehenden Einhüllenden stets Null sein muss. Durch dieses Berechnungsverfahren für IMF können infolge des Bezugssystemwechsels von einem rechtwinkligen in ein gekrümmtes Koordinatensystem Anomalien entstehen, welche sich durch multiple Extremstellen zwischen aufeinanderfolgenden Nulldurchgängen charakterisieren. Um solch unerwünschte Schwankungen durch asymmetrische Wellenformen zu vermeiden, muss der Sifting-Prozessschritt so oft wie nötig durchgeführt werden, bis keine dieser Anomalien mehr auftre-

ten. Dabei dient im nächsten Schritt  $h_1$  als Ausgangssignal für die nächste Iteration  $k$  des Sifting-Algorithmus, wodurch die Berechnungsvorschrift

$$h_{1(k-1)}(t) - m_{1k} = h_{1k} \quad (5.9)$$

resultiert. Dieser Prozess wird so lange wiederholt, bis ein zuvor festgelegtes Abbruchkriterium erfüllt ist. Als erste Option dient die Definition eines Grenzwerts der resultierenden Standardabweichung  $std$  zweier aufeinanderfolgender Differenzsignale  $h_1(t)$  gemäß

$$std_k = \frac{\sum_{t=0}^T |h_{1(k-1)}(t) - h_{1k}(t)|^2}{\sum_{t=0}^T h_{1(k-1)}^2(t)}. \quad (5.10)$$

Typische Grenzwerte für  $std$  liegen bei 0,2 bis 0,3. Ein zweites mögliches Abbruchkriterium berücksichtigt im Gegensatz dazu die Eigenschaften einer IMF und basiert auf der S-Zahl, welche die Anzahl an aufeinanderfolgenden Schritten des Sifting-Prozesses beschreibt, in welchen sich die Anzahl von Extrema und Nullstellen höchstens um 1 unterscheidet. Die S-Zahl wird im Vorfeld festgelegt und sollte nach empirischen Untersuchungen von HUANG im Wertebereich zwischen 4 - 8 liegen. [HUAN05a; HUAN05b; HUAN98]

Nach erfolgreicher Durchführung des Sifting-Algorithmus kann im zweiten Schritt die Hilbert-Transformation auf jede IMF angewendet werden, um die Augenblicksfrequenz (engl.: Instantaneous Frequency) (IF) zu berechnen. Die Hilbert-Transformierte eines Signals  $x(t)$  berechnet sich nach folgender Gleichung

$$\mathcal{H}[x(t)] = \frac{1}{\pi} PV \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau, \quad (5.11)$$

in welcher  $PV$  der cauchysche Hauptwert des Integrals darstellt. Mit der Hilbert-Transformation lässt sich das analytische Signal definieren als

$$z(t) = x(t) + iy(t) = a(t)e^{i\theta(t)}, \quad (5.12)$$

mit der Augenblicksamplitude (engl.: Instantaneous Amplitude) (IA)  $a(t)$  und der Phasenfunktion  $\theta(t)$ :

$$a(t) = \sqrt{x^2 + y^2} \quad \text{und} \quad \theta(t) = \arctan \frac{y}{x}. \quad (5.13)$$

Die IF  $\omega$  ist bestimmt durch

$$\omega = \frac{d\theta}{dt}. \quad (5.14)$$

Nach Durchführung der Hilbert-Transformation für jede IMF, entspricht das Originalsignal  $x(t)$  dem Realteil  $\mathcal{R}$  nach folgender Gleichung:

$$x(t) = \mathcal{R} \left\{ \sum_{j=1}^n a_j(t) e^{i \int \omega_j(t) dt} \right\}. \quad (5.15)$$

Diese Zeit-Frequenzverteilung der Amplitude wird als Hilbert-Amplituden-Spektrum oder kurz Hilbert-Spektrum bezeichnet. Durch Kombination von EMD mit der Hilbert-Spektralanalyse entsteht schließlich die HHT. Diese eignet sich insbesondere für die Zeit-Frequenzanalyse von nicht linearen und instationären Signaldaten. Ihr Vorteil liegt darin, dass für die Extraktion von Informationen aus den IMF, im Gegensatz zur Wavelet-Transformation, kein a priori Wissen über das Signal notwendig ist, da die ermittelten IMF per Definition im Signal enthaltene Komponenten widerspiegeln. [HUAN05a; HUAN05b]

Für die Implementierung der HHT wird das Softwarepaket EMD: Empirical Mode Decomposition genutzt. Die Funktion `emd.sift.mask_sift()` führt den Sifting-Algorithmus für die Ermittlung der IMF, unter Berücksichtigung der zuvor gewählten Anzahl an zu ermittelnden IMF (`max_ims`), durch. Die ausgangsseitige Liste an IMF dient als Eingabe in die Funktion `emd.spectra.frequency_transform()`, welche die IF, IA und die Augenblicksphasen der zuvor ermittelten IMF berechnet. Diese werden als Input für die Funktion `emd.spectra.hilberthuang()` genutzt, um die Energieverteilungen im Signal über die Zeit und Frequenz zu erhalten. Als Ergebnis steht eine zweidimensionale Matrix, welche die Hilbert-Huang-Transformation beinhaltet. [QUIN21] Aufgrund der hohen Ausgangsdimension der HHT-Matrix ( $n \times nbins$  mit  $n$  als Anzahl der Datenpunkte im Signal und  $nbins$  als Anzahl an zuvor definierten Frequenzbereichen) und dem Ziel der automatisierten Weiterverarbeitung, erfolgt abschließend die Berechnung charakteristischer Kennwerte angelehnt an die zuvor beschriebene DWT. Die Forschungslandschaft liefert für diesen Ansatz kaum Anhaltspunkte, welche Kennwerte sich für die Merkmalsextraktion im Anschluss an eine HHT eignen. Die Auswertung des gewonnenen Spektrums erfolgt überwiegend visuell basierend auf graphischen Darstellungen in Form von Spektrogrammen [BASS07; CAO13; KALV10; PENG06; SUSA18; TETI10]. Für die automatisierte Weiterverarbeitung der Informationen erfolgt deshalb in Anlehnung an die identifizierten

Kennwerte bei der DWT die Berechnung von Mittelwert, Effektivwert, Standardabweichung, Crest-Faktor und der Peak-to-Peak-Distanz. Zusätzlich wird die absolute Energie (engl.: Absolute Energy, vgl. Anhang A.3) als Kennwert basierend auf den Ergebnissen aus [DALE16; PENG06] implementiert.

### 5.3.3 Merkmalsselektion

Die automatisierte Merkmalsextraktion führt zu einer Vielzahl an gewonnenen Merkmalen aus den Rohdaten. Aufgrund der automatisierten Berechnung liefert in der Regel ein Großteil der Merkmale keine relevanten Informationen in Bezug auf die Vorhersageaufgabe oder verhält sich indifferent bei veränderten Prozessbedingungen [TETI10]. Zusätzlich vergrößert sich durch eine erhöhte Anzahl an Eingangsdimensionen der Bedarf an benötigten Trainingsdaten und die Gefahr der Überanpassung (engl.: Overfitting) steigt infolge der resultierenden hohen Komplexität für die Generalisierung (vgl. Kapitel 2.2.5). Dies kann bis hin zu einer Verschlechterung der Prognosegenauigkeit führen, falls bedeutungslose Kennwerte in die Modellbildung mit einfließen [SCHO21a]. Um eine möglichst hohe Prognosegenauigkeit bei entsprechend hoher Generalisierungsfähigkeit der Modelle zu erzielen, ist es deshalb notwendig, an die automatisierte Merkmalsextraktion eine geeignete Merkmalsselektion anzuschließen. Dies ermöglicht, lediglich diejenigen Merkmale mit der höchsten Bedeutung für die Prognose der Zielgröße in die Modellierung mit einzubeziehen. In industriellen Anwendungen sollte die Auswahl automatisiert, möglichst ohne manuelle Analysen, jedoch unter Berücksichtigung der vorliegenden Prognoseaufgabe erfolgen. [TETI10]

Grundsätzlich lassen sich Algorithmen für die Merkmalsselektion angelehnt an die in Abbildung 5.3 gezeigte Einteilung der ML-Verfahren in überwachte, unüberwachte und halbüberwachte Verfahren gliedern [TANG14]. Für die in dieser Arbeit adressierte Qualitätsprognose sind beschriftete Datensätze für eine Klassifikation vorhanden, weshalb Verfahren aus dem Bereich der überwachten Merkmalsselektion zum Einsatz kommen. Diese lassen sich in Filter-, Wrapper- und Embedded-Ansätze kategorisieren. **Filter-Ansätze** berücksichtigen für die Einstufung der Merkmale ein unabhängiges Kriterium und bewerten Teilmengen von Merkmalen, ohne die Einbeziehung von Lernalgorithmen. Die Einstufung kann dabei für jedes Merkmal univariat, d. h. unabhängig des Merkmalsraums oder multivariat stapelweise erfolgen. Darauf folgend werden im zweiten Schritt die am höchsten bewerteten Merkmale ausgewählt, um Prognosemodelle zu erstellen. Im Gegensatz dazu nutzen **Wrapper-Ansätze** spezifische Lernalgorithmen für die Evaluation und Aus-

wahl eines Merkmalsatzes. Dabei werden typischerweise die Auswahl einer Teilmenge an Merkmalen und die Evaluation mithilfe eines Lernalgorithmus iterativ durchlaufen, bis die gewünschte Qualität der Prognose erreicht ist. In der Regel erreichen Wrapper-Ansätze dadurch eine höhere Prognosegenauigkeit als Filter-Ansätze, jedoch erhöht sich der Rechenaufwand für jedes Merkmal  $m$  mit einer Ordnung von  $\mathcal{O}(2^m)$ . **Embedded-Ansätze** kombinieren die Vorteile von Filter- und Wrapper Ansätzen, indem Lernalgorithmen mit eingebetteter Merkmalsselektion zum Einsatz kommen. Diese zeichnen sich durch eine effizientere Berechnung aus, da die genutzten Algorithmen zusätzlich zur Prognose, eine Auswahl der relevantesten Merkmale liefern können und beziehen im Gegensatz zu den Filter-Ansätzen die Wechselwirkungen mit dem Prognosemodell mit ein. [KUMA14; SING20; TANG14]

Für die prozess- bzw. geometrieelementindividuelle Auswahl der relevanten Merkmale wird ein an die Prognoseaufgabe ausgerichteter Selektionsalgorithmus<sup>6</sup> entwickelt und implementiert. Dieser Schritt ist notwendig, da aufgrund der unterschiedlichsten Anwendungsfälle und der Vielzahl an zur Verfügung stehenden Möglichkeiten zur Reduktion des Merkmalsraums, nicht zuvor bestimmt werden kann, welche Auswahlmethode sich für den spezifischen Anwendungsfall optimal eignet. Das Ziel der entwickelten Handlungsvorschrift ist es deshalb, die Vorteile aus den unterschiedlichen Ansätzen zu kombinieren, um automatisiert individuell ausgewählte Merkmalsätze für die Modellerstellung einzelner Geometrieelemente zu erhalten.

Zunächst werden mittels eines Varianzschwellenwertfilters alle Merkmale entfernt, welche über den untersuchten Datensatz eine Varianz von 0 aufweisen. Die Werte der Merkmale sind demnach über alle Instanzen eines Geometrieelements konstant und liefern somit keine zusätzlichen Informationen für die Prognoseaufgabe. Auf dem reduzierten Merkmalsatz je Geometrieelement wird ein Satz  $S_{fs}$  an Merkmalsselektionsalgorithmen angewendet.  $S_{fs}$  beinhaltet eine Teilmenge der  $n_{fs}$  implementierten und zur Verfügung stehenden Merkmalsselektionsalgorithmen, welche bei Anwendung jeweils einen spezifischen Merkmalsatz als Ergebnis liefern. Der finale proprietäre Merkmalsatz (engl.: Proprietary Feature Set) (PropFeatSet) für ein Geometrieelement entsteht mittels Bestimmung der Häufigkeit der Merkmale in den Ergebnissätzen der angewandten Merkmalsselektionsalgorithmen. Dafür wird ein Schwellenwert  $SW_{fs,prop}$  für die Häufigkeit der Nennung eines Merkmals in den Ergebnissätzen gesetzt, ab dem ein Merkmal in den finalen Merkmalsatz mit aufgenommen wird. Dieser kann im Wertebereich von 1 bis

<sup>6</sup>Erstveröffentlichung in [FERT22b].

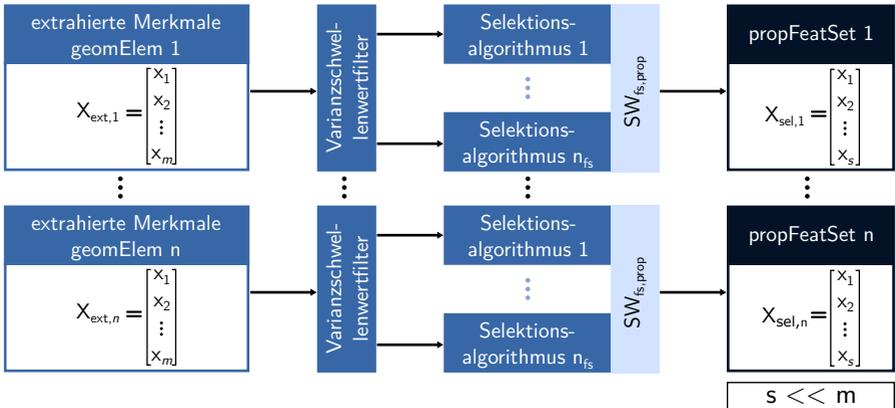


Abbildung 5.4: Schematische Darstellung des Merkmalsselektionsalgorithmus, angelehnt an [FERT22c].

zur Anzahl der zur Verfügung stehenden Merkmalsselektionsalgorithmen variieren. Abbildung 5.4 verdeutlicht die Handlungsvorschrift des entwickelten Algorithmus für die Merkmalsselektion. Zusätzlich ermöglicht eine ähnliche Vorgehensweise unter der Nutzung der PropFeatSet aller Geometrielemente zusammen mit einem zweiten Schwellenwert  $SW_{fs,comm}$  die Ermittlung eines gemeinsamen Merkmalsatzes (engl.: Common Feature Set) (CommFeatSet), für die Untersuchung der Leistungsfähigkeit eines geometrieunabhängigen Merkmalsatzes, über alle Geometrielemente hinweg.

Die spezifische Merkmalsselektion basiert auf Algorithmen aus den drei erwähnten Bereichen, um eine gewisse Bandbreite an zu Verfügung stehenden Ansätzen abdecken zu können und verschiedene Auswahlmechanismen miteinander zu kombinieren. Für die Implementierung der Filter- und Embedded-Ansätze dient das Modul `sklearn.feature_selection` aus der Software-Bibliothek Scikit-learn [PEDR12]. Die Funktion `VarianceThreshold()` selektiert die Merkmale, wie zuvor erwähnt, basierend auf dem Varianzschwellenwertfilter.

### Algorithmen für die Merkmalsselektion

Die Funktion `SelectKBest()` ermöglicht eine filterbasierte, univariate Merkmalsselektion (`uniStat`) auf Grundlage unterschiedlicher Bewertungsfunktionen (`score_func`), je nach zugrundeliegendem statistischen Testverfahren. In diesem Fall basiert die Auswahl auf der Bestimmung der Transinformation

$MI$  (engl.: Mutual Information) für jede der unabhängigen Variablen in Bezug auf die abhängige Zielvariable. Besteht zwischen beiden Variablen eine statistische Abhängigkeit, dann lässt sich mit den enthaltenen Informationen der einen Variable, mit einer bestimmten Wahrscheinlichkeit auf die andere Variable schließen. Mit steigender Abhängigkeit steigen die Werte von  $MI$ . Bei statistischer Unabhängigkeit zwischen den Variablen beträgt  $MI = 0$ . Dies bildet eine rechenineffiziente und beliebte Methode für die univariate Merkmalsselektion ab und liefert im Vergleich zur Nutzung alternativer statistischer Testverfahren, wie beispielsweise dem F-Test, den entscheidenden Vorteil auch nicht lineare statistische Abhängigkeiten erfassen zu können. Die Auswahl der Merkmale erfolgt basierend auf der absteigend sortierten Auflistung der Merkmale nach den Werten für  $MI$  und einer vom Nutzer gewählten Anzahl an aufzunehmenden Merkmalen. [ERTA20; KRAS04; ROSS14; ROY19; TANG14] Im Rahmen dieser Arbeit wurde die Anzahl auf 50 Merkmale begrenzt und hat sich als erfolgversprechend gezeigt.

Bei der Lasso-Regression (Lasso) handelt es sich um eine Regressionsmethode aus dem Bereich der Shrinkage-Methoden. Der Hauptvorteil dieser Methoden gegenüber einer einfachen linearen Regression besteht darin, Prognosemodelle mit allen Prädiktoren (Merkmalen) zu erstellen. Durch Regularisierungsverfahren werden die Koeffizienten der nicht relevanten Merkmale nahe oder auf 0 „geschrumpft“, wodurch Shrinkage-Methoden sich prädestinieren eine modellbasierte Merkmalsselektion entsprechend der Embedded-Ansätze durchzuführen. Grundsätzlich lassen sich die Ridge- und die Lasso-Regression basierend auf den verwendeten Straftermen unterscheiden. Für die Bestimmung der Koeffizienten  $\beta_j$  des zugrundeliegenden linearen Regressionsmodells

$$f(X) = \beta_0 + \sum_{j=1}^p \beta_j x_j, \quad (5.16)$$

dient die Methode der kleinsten Quadrate (engl.: Ordinary Least Squares), welche darauf abzielt die Koeffizienten entsprechend zu bestimmen, um die Summe der Residuen der Quadrate

$$RSS(\beta) = \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 \quad (5.17)$$

zu minimieren. Die Ridge-Regression „schrumpft“ die Koeffizienten durch Hinzufügen eines Strafterms abhängig ihrer Größe. Die Ridge-Koeffizienten

bestimmen sich demnach durch Minimierung des Residuums der erweiterten Quadratsumme nach

$$\hat{\beta}^{ridge} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}, \quad (5.18)$$

wobei  $\lambda \geq 0$  den separat zu bestimmenden Regularisierungsparameter darstellt. Je Größer der Wert von  $\lambda$  desto höher ist der Effekt der Regularisierung und die Werte der Koeffizienten werden stärker in Richtung 0 gedrückt, wodurch die Flexibilität der Ridge-Regression und damit die Varianz sinkt. [HAST09; JAME21] Dabei resultiert durch den gewählten Strafterm der entscheidende Nachteil, dass die Koeffizienten zwar in Richtung 0 gedrückt werden, jedoch diesen Wert niemals erreichen. Dies führt zu erschwerter Interpretation der Modelle bei einer hohen Anzahl an eingangsseitigen Merkmalen. Um diesem Nachteil entgegen zu wirken, nutzt die Lasso-Regression einen alternativen Strafterm. Die Lasso-Koeffizienten bestimmen sich aus

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}. \quad (5.19)$$

Demnach wird der Ridge  $L_2$ -Strafterm  $\sum_{j=1}^p \beta_j^2$  bei Lasso durch den  $L_1$ -Strafterm  $\sum_{j=1}^p |\beta_j|$  ersetzt, wodurch die Lasso-Regression in der Lage ist, bestimmte Koeffizienten auf den Wert 0 zu setzen. Dies eröffnet die Möglichkeit einer direkten Merkmalsselektion basierend auf der Auswertung der Modellkoeffizienten. [HAST09; JAME21] Für die Merkmalsauswahl wurde von EFRON der Least Angle Regression (LARS) Algorithmus entwickelt, welcher sämtliche Lasso-Schätzungen berechnet und sich durch eine hohe Effizienz in dessen Berechnung auszeichnet. Die Vorteile liegen insbesondere in der Merkmalsauswahl aus hochdimensionalen Datensätzen, welche infolge der hohen Anzahl an automatisiert extrahierten Merkmalen (vgl. Kapitel 5.3.2) resultieren. [EFRO04] Für die Implementierung der auf Lasso basierten Merkmalsselektion wird deshalb das Objekt `LassoLarsCV()` aus dem Modul `sklearn.linear_model` mit integrierter Kreuzvalidierung für eine zusätzliche Verbesserung der Generalisierungsfähigkeit gewählt. Die Merkmalsauswahl erfolgt schließlich mit der Funktion `SelectFromModel()`, welche als Eingabeparameter das zuvor gewählte Objekt erhält.

Die logistische Regression (LogisRe) ist ein Regressionsverfahren zur Prognose von kategorialen Variablen. Die Grundlage für die Prognose bildet die

Modellierung der Wahrscheinlichkeit  $\pi(X)$  für das Auftreten eines bestimmten Ereignisses  $y$  entsprechend einer gegebenen Wertekombination des Eingangsvektors  $X$ . Angelehnt an die lineare Regression besteht das logistische Regressionsmodell aus der linearen systematischen Komponente  $z(X)$  nach Gleichung 5.16 mit  $X = (x_1, \dots, x_j)$  als Merkmalsvektor und der nicht linearen logistischen Funktion

$$\pi = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}. \quad (5.20)$$

Die logistische Funktion wird verwendet, um den reellen Wertebereich von  $z(X)$  in eine Wahrscheinlichkeit mit dem Wertebereich  $[0, 1]$  zu transformieren. Das logistische Regressionsmodell

$$\pi(X) = \frac{1}{1 + e^{-z(X)}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_j x_j)}} \quad (5.21)$$

resultiert schließlich durch Einsetzen von Gleichung 5.16 in Gleichung 5.20. Für die Schätzung der Koeffizienten dient die Maximum-Likelihood-Methode, welche ermöglicht Schätzwerte für die unbekanntenen Koeffizienten so zu bestimmen, um die Wahrscheinlichkeit der beobachteten Daten zu maximieren. Dabei gilt es, die Likelihood-Funktion, welche die Koeffizienten auf Wahrscheinlichkeiten mit der die Zielwerte auftreten können abbildet, zu maximieren. [BACK21; HUDE20] Im Anschluss an die Modellbildung können die relevanten Merkmale über die Werte der Koeffizienten identifiziert werden. Jedes Merkmal, dessen Parameter den Wert 0 aufweist gilt als redundantes Merkmal und fällt dementsprechend aus der finalen Auswahl heraus. Dafür kommen in der logistischen Regression ebenfalls Regularisierungsverfahren zum Einsatz. Im vorliegenden Fall (Anzahl der Merkmale  $\gg$  Anzahl an Trainingsbeispielen) kommt basierend auf den Ergebnissen von ZOU und HASTIE die „elastic net“ Regularisierung zum Einsatz. Diese zeichnet sich durch die Kombination der  $L_1$ - und  $L_2$ -Strafterme aus und schafft es in realen Anwendungsfällen häufig die Ergebnisse der Lasso-Regularisierung zu übertreffen. Der elastic net Strafterm mit dem Regularisierungsparameter  $\alpha$  resultiert in

$$(1 - \alpha) |\beta_j| + \alpha \beta_j^2, \quad (5.22)$$

welcher eine Kombination der Lasso- und Ridge-Strafterme darstellt. [ZOU05] Die Implementierung erfolgt mit der Funktion `SelectFromModel()` und dem Eingabeobjekt `LogisticRegression()` inklusive des Strafterms `penalty='elasticnet'` aus dem Modul `sklearn.linear_model`.

Bei Boruta handelt es sich um einen effizienten Wrapper-Ansatz, der es ermöglicht, sämtliche relevante Merkmale für die Prognoseaufgabe, basierend auf dem Vergleich der Relevanz der tatsächlichen Merkmale mit zufällig berechneten Werteverteilungen, zu ermitteln. Der um einen Random Forest (vgl. Kapitel 5.3.4) herum aufgebaute Wrapper erstellt zunächst zufällige Kopien für jedes Merkmal, sogenannte „Shadow Features“ bzw. im Deutschen Schattenmerkmale, welche zusätzlich zu den tatsächlichen Merkmalen für das Erstellen des internen Prognosemodells genutzt werden. Der Algorithmus hinter dem Random Forest ist in der Lage, die prozentuale Bedeutung der Merkmale (engl.: Feature Importance) für die Prognose zu liefern. Diese Eigenschaft wird genutzt, um den Wert des am höchsten bewerteten Schattenmerkmals zu ermitteln und mit den Werten der tatsächlichen Merkmale zu vergleichen. Übertrifft der Wert eines Merkmals die Bedeutung des am höchsten bewerteten Schattenmerkmals, gilt es als nützlich für die Prognose und wird damit als „Treffer“ gewertet. Diese Vorgehensweise wird wiederholt, bis die zuvor gewählte Anzahl an Iterationen durchlaufen ist. Anhand der Normalverteilung der Anzahl an Treffern für jedes tatsächliche Merkmal, werden basierend auf einem Hypothesentest die Merkmale mit den häufigsten Treffern auf einem zuvor definierten Signifikanzniveau (in diesem Fall  $\alpha = 0,05$ ) ausgewählt. Dies ermöglicht eine statistische Absicherung der Merkmalsauswahl. [KURS10] Die Implementierung erfolgt mit der Funktion `BorutaPy()` aus dem gleichnamigen Softwarepaket.

Insgesamt stehen damit vier Merkmalsselektionsalgorithmen mit unterschiedlichen Auswahlmechanismen, für den Einsatz in der zuvor beschriebenen automatisierten Handlungsvorschrift zur spezifischen Ermittlung von Merkmalsätzen für jedes qualitätsrelevante Geometrieelement am Bauteil, zur Verfügung. Deren Nutzung für die Auswahl kann mit den eingeführten Schwellenwerten in beliebiger Kombination erfolgen, wodurch eine flexible Anpassung an den zur Verfügung stehenden Datensatz ermöglicht wird. Als Ergebnis stehen individuelle Merkmalsätze für jedes Geometrieelement, welche in die Prognosemodellerstellung einfließen.

### 5.3.4 Algorithmenauswahl und -optimierung

Für die Modellbildung basierend auf den ausgewählten Merkmalen existiert eine Vielzahl an verschiedenen Algorithmen aus dem Bereich des ML. In diesem Zusammenhang beschreibt das von WOLPERT formulierte No-Free-Lunch-Theorem, dass die durchschnittliche Leistung zweier unterschiedlicher Algorithmen über jedes potenzielle Problem hinweg identisch ist [WOLP96; WOLP97]. Übertragen auf ML impliziert dies, dass keine Aussage über einen Algorithmus getroffen werden kann, welcher in der gesamten Breite der Vorhersageprobleme die beste Performanz liefert [BROW21; WOLP02]. Als Konsequenz auf das No-Free-Lunch-Problem hat die Vielfalt an Anwendungsfällen und Daten aus der realen Welt zu einer Vielzahl an entwickelten - und zur stetigen Erforschung weiterer - Algorithmen geführt [MURP12]. Im Bereich des traditionellen ML kommen insbesondere Algorithmen mit flachen Strukturen (engl.: Shallow Structure) zum Einsatz, welche angelehnt an Kapitel 5.3.2 und 5.3.3 eine explizite Merkmalsextraktion und -selektion erfordern. Die Prognosegenauigkeit hängt stark von den domänenspezifisch extrahierten und ausgewählten Merkmalen ab. Algorithmen mit tiefen Strukturen (engl.: Deep Structure) lernen die Zusammenhänge direkt aus den Rohdaten und bilden über die verschiedenen Schichten abstrakte Merkmalsrepräsentationen während der Trainingsphase, wodurch die vorherige Merkmalsselektion wegfällt. Der entscheidende Nachteil dieser Methoden ist der hohe Bedarf an Trainingsbeispielen und Rechenleistung im Vergleich zu den traditionellen Methoden mit flachen Strukturen. [SCHO21a; WANG18]

Basierend auf den eigenen Ergebnissen aus [FERT22b] und den Arbeiten von SCHUH, SCHORR, ZIEGENBEIN und BRECHER (vorgestellt in Kapitel 2.4) sowie der Zusammenfassung von bereits erfolgreichen implementierten Forschungsansätzen aus [SARI21], werden für die Lösung des Klassifikationsproblems im Rahmen dieser Arbeit Methoden des ML verwendet, die zu der Gruppe der traditionellen Shallow-Structure-Algorithmen gehören. Die folgenden Methoden werden innerhalb der Prozesskette für die Erstellung der Prognosemodelle berücksichtigt: Support Vector Machine (SVM),  $k$ -Nearest Neighbours (KNN), Ridge-Classifer (RidgeRe), Gaussian Naive Bayes (GNB), Multilayer-Perceptron (MLP), Decision Tree (DT) sowie die Ensemble-Learning-Methoden Random Forest (RF), Extra Trees (XT) und AdaBoost-Classifer (AdaBoost). Innerhalb eines Ensembles werden einzelne Entscheidungen mehrerer Klassifikatoren mittels unterschiedlicher Ansätze zu einer gemeinsamen finalen Entscheidung kombiniert. Oftmals liefern Ensembles bessere Prognosegüten als die einzelnen Klassifikatoren. [DIET00] Im Folgenden wird

die Funktionsweise der implementierten Algorithmen kurz vorgestellt. Deren Nutzung innerhalb der Prozesskette basiert auf Basis der von Scikit-learn zur Verfügung stehenden Implementierung für die jeweiligen Methoden.

### Support Vector Machine (SVM)

Die Grundidee hinter einer SVM liegt darin zwei Klassen unter Verwendung einer minimalen Anzahl von Eingangsdatenpunkten (Stützvektoren, engl.: Support Vectors) durch eine Hyperebene, wie in Abbildung 5.5 gezeigt, zu trennen. Die klassentrennende Hyperebene maximiert dabei den Abstand (engl.: Margin) zu den Stützvektoren. Die mithilfe der Stützvektoren eindeutig definierte Hyperebene bildet damit die Entscheidungsgrenze zwischen den beiden Klassen ab. Sind die Datenpunkte nicht linear trennbar, wird durch eine Transformation der Datenpunkte in einen höherdimensionalen Raum mithilfe geeigneter Kernel-Funktionen, das nicht lineare Problem in ein lineares Problem überführt. Dieser sogenannte Kernel-Trick ermöglicht schließlich eine lineare Trennung der transformierten Datenpunkte mittels einer höherdimensionalen Hyperebene. Dadurch ist es möglich flexiblere nicht lineare Entscheidungsgrenzen zu ermitteln. Typische Kernel-Funktionen sind die Polynomiale-, Sigmoid- oder Radial-Basis-Kernel. [GUYO06; HAST09; JOSH20]

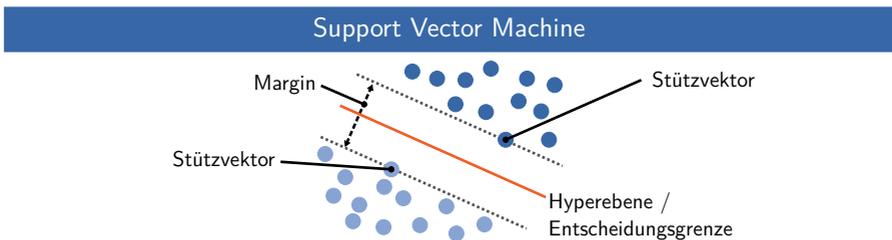


Abbildung 5.5: Darstellung der Funktionsweise einer SVM, eigene Darstellung basierend auf [HAST09].

### **$k$ -Nearest Neighbours (KNN)**

Bei dem KNN handelt es sich um einen speicherbasierten Algorithmus. Für die Prognose eines unbekanntem Datenpunkts  $x_0$  werden die  $k$  Datenpunkte aus dem vorherigen Training gesucht, welche je nach gewählter Metrik den geringsten Abstand zu  $x_0$  aufweisen. Die prognostizierte Klasse ist diejenige Klasse, welche als Mehrheit in den ermittelten nächsten Nachbardatenpunkten vorhanden ist. Der KNN ist demnach einer der einfachsten ML-Algorithmen, kann jedoch in häufigen Fällen hohe Prognosegüten aufweisen. [HAST09; JOSH20] Abbildung 5.6 zeigt beispielhaft die Klassifikation mittels KNN.

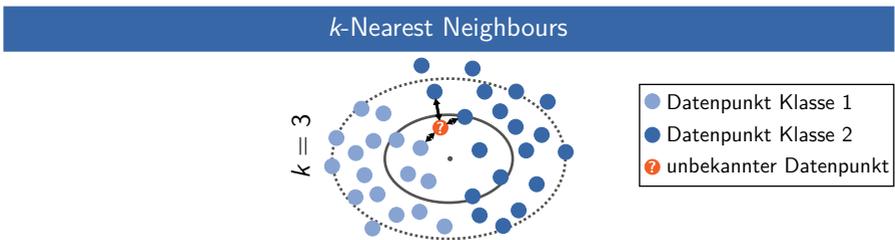


Abbildung 5.6: Klassifikation mittels KNN, eigene Darstellung basierend auf [JOSH20].

### **Ridge-Classifier (RidgeRe)**

Die Ridge-Regression ist wie in Kapitel 5.3.3 beschrieben im eigentlichen Sinn eine Regressionsmethode. Für eine Ridge-basierte Klassifikation erfolgt die Transformation der binären Zielwerte in den Wertebereich  $\{-1, 1\}$ , wodurch das Problem wie eine Regression behandelt wird. Das Vorzeichen des prognostizierten Regressionswerts entspricht schließlich der vorhergesagten Klassenzugehörigkeit. [BOIS22c]

### **Gaussian Naive Bayes (GNB)**

Die naiven Bayes-Algorithmen für die Klassifikation basieren auf der Anwendung des Bayes-Theorems und gehören zur Gruppe der probabilistischen Klassifikationsverfahren. Der Satz von Bayes beschreibt den mathematischen Zusammenhang, um aus einer bedingten Wahrscheinlichkeit  $P(y|X)$  die umgekehrte bedingte Wahrscheinlichkeit  $P(X|y)$  zu berechnen. [LUDW20]  $y$  sei dabei die Klassenzugehörigkeit und  $X = \{x_1, x_2, \dots, x_m\}$  der Merkmalsvektor

mit  $m$  der Anzahl an Merkmalen, dann gilt für die bedingte Wahrscheinlichkeit  $P(y|X)$  einer Klassenzugehörigkeit  $y$  unter der Bedingung der Eingabewerte  $X$ :

$$P(y|X) = \frac{P(y) P(X|y)}{P(X)} = \frac{P(y) P(x_1, x_2, \dots, x_m|y)}{P(x_1, x_2, \dots, x_m)}. \quad (5.23)$$

Unter der naiven bedingten Unabhängigkeitsannahme

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_m) = P(x_i|y), \quad (5.24)$$

für alle  $i$ , kann die Beziehung aus Gleichung 5.23 vereinfacht werden zu

$$P(y|x_1, x_2, \dots, x_m) = \frac{P(y) \prod_{i=1}^m P(x_i|y)}{P(x_1, x_2, \dots, x_m)}. \quad (5.25)$$

Aufgrund der Tatsache, dass  $P(X)$  als Eingabe konstant ist, resultiert folgende Klassifizierungsregel:

$$\hat{y} = \operatorname{argmax}_y P(y) \prod_{i=1}^m P(x_i|y). \quad (5.26)$$

$P(y)$  und  $P(x_i|y)$  können mittels Maximum a posteriori Schätzung ermittelt werden, wobei erstes die relative Häufigkeit der Klasse in der Trainingsmenge darstellt. [BOIS22d; ZHAN04] Je nach zugrundeliegender Annahme über die Verteilung von  $P(x_i|y)$  unterscheiden sich verschiedene Bayes-Klassifikatoren. Im Fall des genutzten GNB liegt die Annahme zugrunde, dass die Wahrscheinlichkeit der Merkmale

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}} \quad (5.27)$$

normalverteilt ist.  $\sigma_y$  und  $\mu_y$  werden mittels Maximum-Likelihood-Methode geschätzt. [BOIS22d; RUNK15]

## Multilayer-Perceptron (MLP)

In Anlehnung an den Aufbau biologischer Netzwerke von Nervenzellen menschlicher bzw. tierischer Gehirne bestehen künstliche neuronale Netze aus vernetzten Neuronen, welche in mehreren Schichten angeordnet und schichtweise verbunden sind. Die Grundlage dafür liefert ROSENBLATT durch die Vorstellung des Perzeptrons (engl.: Perceptron), welches über einen aufgenommenen Input einen Output errechnen konnte. Das in Abbildung 5.7 links dargestellte Perzeptron, als einfachstes künstliches neuronales Netz, erwartet eingangsseitig  $m$  Werte, welche jeweils mit den Gewichten  $w_i$  multipliziert und anschließend aufsummiert werden. Die Ausgabe entsteht basierend dem Vergleich der gewichteten Summe mit einem Schwellenwert. Ein künstliches neuronales Netz oder MLP entsteht schließlich durch die in Abbildung 5.7 rechts dargestellte schichtweise Vernetzung des einfachen Perzeptrons mit einer Eingabeschicht, verdeckten Schichten und einer Ausgabeschicht. Die Neuronen der Eingabeschicht nehmen die Werte der Merkmale auf und leiten diese an die Neuronen der ersten verdeckten Schicht weiter. Jedes Neuron ermittelt aus der eingangsseitigen gewichteten Summe den eigenen Ausgangswert durch eine nicht lineare Aktivierungsfunktion. Jede verdeckte Schicht besteht dabei aus einer Anzahl an Neuronen, welche aus der zuvor gewählten Netzwerktopologie resultiert. Die Werte werden durch die jeweiligen Schichten geleitet, bis das Neuron der Ausgabeschicht die zu prognostizierende Zielvariable ausgibt. Das Trainieren des Netzwerks erfolgt durch Anpassung der Gewichte basierend auf dem ermittelten Prognosefehler nach einem Durchlauf des Trainingsdatensatzes durch das MLP. Der Fehler wird rückwärts durch jede Schicht rückgeführt bzw. propagiert, um die Gewichte in jeder Schicht dementsprechend zu aktualisieren, sodass der Prognosefehler minimiert wird. Dieser Algorithmus wird als Backpropagation bezeichnet. [JOSH20; ROSE58; RUNK15; SONN22]

## Decision Tree (DT)

Entscheidungsbäume (engl.: Decision Trees) nutzen für die Vorhersage eine im Training ermittelte baumartige Struktur (Abbildung 5.8 links), bei der an jedem Knoten eine Entscheidung getroffen wird. Somit bilden DT heuristische Strukturen ab, welche durch eine Abfolge von Entscheidungen in einer bestimmten Reihenfolge aufgebaut sind. Das Ziel bei der Erstellung eines DT ist es, die Entscheidungsregeln für jeden Knoten basierend auf den Trainingsdaten zu ermitteln. Die grundsätzliche Vorgehensweise besteht darin, in jedem Schritt das Merkmal mit dem höchsten Informationsgehalt (engl.: Information Gain) (IG) bezogen auf die zugehörige Klasse zu ermitteln. Dafür

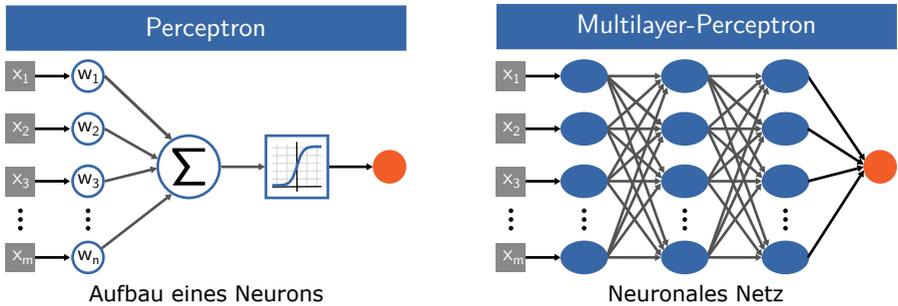


Abbildung 5.7: Perceptron und Multilayer-Perceptron, eigene Darstellung basierend auf [JOSH20].

wird jedes Merkmal auf dessen IG bezogen auf die Klassenprognose hin untersucht und als erster Knoten das Merkmal mit dem höchsten IG gewählt. In der nächsten Stufe erfolgt die gleiche Vorgehensweise für die entstandenen Tochterknoten. Diese Prozedur wird so lange wiederholt bis ein Abbruchkriterium erfüllt ist. Bei keinem zu erwartendem Informationsgewinn bzw. keinen weiteren vorhandenen Merkmalen für die Aufteilung wird das Baumwachstum abgebrochen. Zusätzliche Abbruchkriterien sind, wenn ein Knoten „rein“ (engl.: „pure“) ist, d. h. alle Trainingsdatensätze zu diesem Knoten der selben Klassen angehören, oder eine zuvor gewählte maximale Tiefe der Baumstruktur erreicht ist. Für die jeweilige Bestimmung des IG stehen verschiedene Berechnungsmetriken zur Verfügung. Am häufigsten kommen der Gini-Index und die Entropie zum Einsatz. [HAST09; JOSH20; YADA18]

## Random Forest (RF)

Der RF (vgl. Abbildung 5.8 rechts) ist eine von BREIMAN entwickelte Methode aus dem Bereich des Ensemble-Learnings [BREI01]. Ensemble-Learning-Methoden zeichnen sich durch die Aggregation mehrerer Prognosemodelle aus, um im Fall der Klassifikation per Mehrheitsentscheidung die finale Prognose zu erstellen. Häufig kann die Prognosegenauigkeit durch den Einsatz von Ensemble-Learning-Methoden deutlich verbessert und die Überanpassung des Modells reduziert werden. Grundsätzlich lassen sich die drei Haupttypen Bagging, Random Forest und Boosting für die Erstellung von Ensembles unterscheiden. Beim Bagging werden die Trainingsinstanzen zufällig in eine festgelegte Anzahl an Trainingssets mit Zurücklegen, d. h. eine Instanz kann in mehreren Sets (engl.: „Bootstrap Samples“) vorkommen, aufgeteilt.

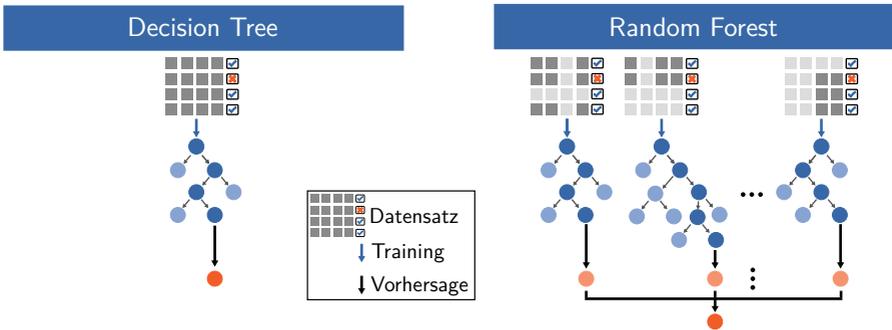


Abbildung 5.8: Aufbau und Funktionsweise DT und RF, eigene Darstellung basierend auf [BREI01; HAST09; JOSH20; YADA18].

Basierend auf jedem ermittelten Trainingsset (engl.: „Bag“) wird ein Entscheidungsbaum trainiert. Die einzelnen DT werden schließlich aggregiert und die finale Prognose per Mehrheitsentscheidung durchgeführt. Diese Vorgehensweise verbessert die Robustheit des Algorithmus gegenüber Ausreißern. Beim RF kommt zusätzlich zum herkömmlichen Bagging – entsteht aus einer Kombination der Wörter „bootstrap“ und „aggregating“ – hinzu, dass in jedem Schritt der Knotenerstellung eines einzelnen DT lediglich eine zufällige Auswahl an Merkmalen berücksichtigt wird. Dies erhöht die Prognosegenauigkeit zusätzlich, falls zwischen den Merkmalen gewisse Abhängigkeiten bestehen, welche das Ergebnis negativ beeinflussen können. Weiterhin erhöht sich dadurch zusätzlich die Robustheit gegenüber Ausreißern. [FORS19; JOSH20]

### Extremely Randomized Trees (XT)

XT ist eine stark verwandte Methode zum RF. Die Unterschiede liegen darin, dass zum einen für die Bildung der einzelnen Entscheidungsbäume alle Instanzen aus dem Trainingsdatensatz genutzt werden. Zusätzlich erfolgt für die Ermittlung der Entscheidungsgrenzen in den jeweiligen Knoten der einzelnen DT zufällig. Dabei erfolgt die Berücksichtigung einer zufälligen Auswahl an Merkmalen für die Bildung eines einzelnen DT. Es wird schließlich das Merkmal für den Knoten ausgewählt, welches basierend auf der zufällig ermittelten Entscheidungsgrenze am Besten hinsichtlich der Prognoseaufgabe abschneidet. [BOIS22b; GEUR06]

## AdaBoost

Im Gegensatz zum RF nutzt der AdaBoost-Algorithmus (engl.: Adaptive Boosting) ein Ensemble aus einfachen schwachen Lernmodellen (engl.: Weak Learner). In der Regel kommen einfache DT, welche lediglich aus einem Knoten und zwei Blättern bestehen, zum Einsatz. Aufgrund deren Aufbau werden diese als „Stümpfe“ (engl.: „Stumps“) bezeichnet. Die finale Prognose wird nicht mehr per einfacher Mehrheitsentscheidung getroffen, sondern die Entscheidung erfolgt basierend auf einer gewichteten Prognose über alle schwachen Lernmodelle hinweg (vgl. Abbildung 5.9). Für die Erstellung eines Prognosemodells unter der Nutzung von AdaBoost werden alle Trainingssamples mit Gewichten  $w_i$  ausgestattet. Initial sind diese über alle Instanzen hinweg gleich. Im ersten Schritt wird das Merkmal mit der höchsten Prognosegenauigkeit ermittelt und der erste Stumpf damit gebildet. Basierend auf dem Gesamtfehler, welcher aus der Summe der Gewichte sämtlicher falsch prognostizierten Instanzen entsteht, wird die individuelle Aussagekraft bzw. im Englischen „Amount of Say“ für den Stumpf bestimmt. Im Anschluss erfolgt die individuelle Anpassung der Gewichte für die Trainingsinstanzen auf Grundlage falsch prognostizierter Instanzen. Dabei erhalten fehlerhaft vorhergesagte Instanzen höhere Gewichte als fehlerfrei prognostizierte Beispiele. Die aktualisierte Gewichtsverteilung fließt schließlich in die Erstellung des nächsten Stumpfs ein. Diese Schritte werden solange iteriert, bis alle Datenpunkte korrekt klassifiziert oder die zuvor gewählte Anzahl an Iterationen bzw. Weak Learnern erreicht ist. [BOIS22b; FREU97; HAST09; MAKL19; SCHA13]

Jeder der vorgestellten Algorithmen (ausgenommen GNB) besitzt Hyperparameter, welche je nach gewählter Konfiguration und zugrundeliegender Datenbasis zu unterschiedlich guten Prognosemodellen führen. Hyperparameter parametrieren vor Start des Trainings den Algorithmus und demnach ist es nicht möglich diese während des Trainingsprozesses anzupassen. Es gilt für jeden Algorithmus die optimale Kombination an Hyperparametern durch Ausprobieren zu finden, um die Prognosegüte der Modelle für den individuellen Anwendungsfall zu maximieren. Für die Suche nach den optimalen Parametern haben sich im Bereich des ML im Wesentlichen die zwei Methoden Gittersuche (engl.: Grid Search) und randomisierte Suche (engl.: Random Search) etabliert. Für die Gittersuche erfolgt im Vorhinein die Definition des Suchraums in Form von diskreten Werten für jeden zu testenden Hyperparameter in Form eines Rasters, welches als `param_grid` bezeichnet wird. Die Gittersuche trainiert und testet mit dem zugrundeliegenden Datensatz alle Parameterkombinationen für den ausgewählten Algorithmus durch und

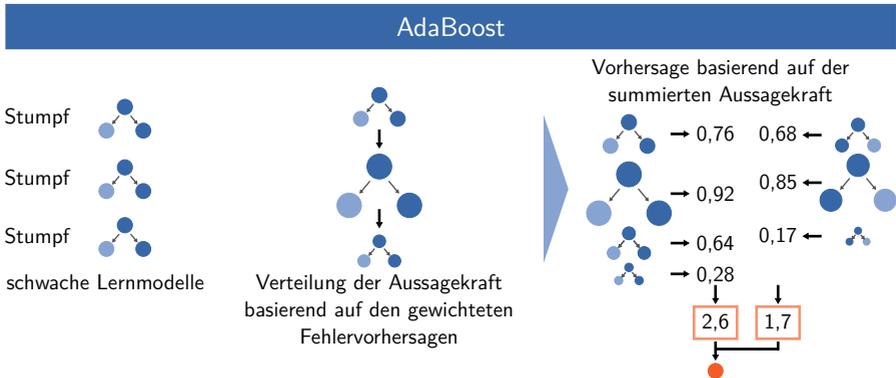


Abbildung 5.9: Aufbau und Funktionsweise AdaBoost, eigene Darstellung basierend auf [BOIS22b; FREU97; HAST09; MAK19; SCHA13].

liefert diejenige Wertekombination zurück, welche zur höchsten Prognosegenauigkeit führt. Die randomisierte Suche führt mittels zufälligen, vor jedem Durchlauf neu bestimmten, Kombinationen die Suche nach den optimalen Parametern durch. [BROW17; GÉRO20; KRAU19] Im Rahmen dieser Arbeit erfolgt die Hyperparameteroptimierung mittels Gittersuche, damit Vergleichbarkeit und Reproduzierbarkeit der Ergebnisse gewährleistet sind. Für die Implementierung dient die Funktion `GridSearchCV()` aus dem Modul `sklearn.model_selection` mit einer 3-fachen Kreuzvalidierung. Die Kreuzvalidierung ist eine Methode, welche den Trainingsdatensatz zufällig in  $k$  Teildatensätze (engl.: „Folds“) aufteilt. Das Training erfolgt schließlich  $k$ -mal, wobei in jedem Durchlauf ein anderer Teildatensatz (auch Validierungsdatensatz genannt) für die Evaluation herangezogen wird. [GÉRO20] Die innerhalb der implementierten Prozesskette definierten Hyperparameterraster für jeden Algorithmus sind in Anhang A.5 zusammengefasst.

### 5.3.5 Evaluation der Prognosegenauigkeit

Für die Evaluation der Prognosegenauigkeit nach dem Trainieren der Algorithmen bedarf es an interpretierbaren und vergleichbaren Kennzahlen, welche es ermöglichen die Modellperformanz bezogen auf den Anwendungsfall zu bewerten und schließlich mittels Vergleich eine Algorithmenauswahl treffen zu können. Die Evaluation der Prognosegüte basiert auf dem vor Beginn der Modellentwicklung separierten Testdatensatz. Dieser stellt für die Modelle unbekannte Daten dar, wodurch die Generalisierungsfähigkeit der trainierten Algorithmen evaluiert werden kann. Die Grundlage für die Ermittlung der Performanzmetriken bildet die Anwendung der Modelle auf dem Testdatensatz. Mittels Vergleich zwischen dem wahren (engl.: True Value) und dem prognostizierten (engl.: Predicted Value) Wert erfolgt schließlich die Evaluation der Prognosegenauigkeit.

Für die vorliegende binäre Klassifikationsaufgabe (vgl. Kapitel 5.3.1) lassen sich zwei Klassen unterscheiden. Erfüllt ein Geometrieelement alle geforderten Toleranzen ist es somit aus produktionstechnischer Sicht i. O. und bildet die positive Klasse ab, welche numerisch mit einer 1 beschrieben wird. Überschreitet eine der ermittelten Abweichungen die zulässige Toleranz gilt das Geometrieelement als n. i. O. Dieser negativen Klasse wird numerisch die 0 zugeordnet. Bei der Klassifizierung lassen sich folgende vier Fälle unterscheiden [BUXM21; FAWC06]:

- Richtig-Positiv (engl.: True Positive) (TP): Das Geometrieelement ist tatsächlich i. O. (1) und wird vom Modell als i. O. (1) klassifiziert.
- Richtig-Negativ (engl.: True Negative) (TN): Das Geometrieelement ist tatsächlich n. i. O. (0) und wird vom Modell als n. i. O. (0) klassifiziert.
- Falsch-Positiv (engl.: False Positive) (FP): Das Geometrieelement ist tatsächlich n. i. O. (0) und wird vom Modell fälschlicherweise als i. O. (1) klassifiziert.
- Falsch-Negativ (engl.: False Negative) (FN): Das Geometrieelement ist tatsächlich i. O. (1) und wird vom Modell fälschlicherweise als n. i. O. (0) klassifiziert.

Durch die Ermittlung der Anzahl an eingetretenen Fällen im Testdatensatz kann die  $2 \times 2$  Konfusionsmatrix (engl.: Confusion Matrix) konstruiert werden, welche in den Zellen jeweils die Anzahl der Instanzen zum jeweiligen Fall beinhaltet. Anhand der Konfusionsmatrix (vgl. Abbildung 5.10) lassen sich durch die Bildung von Verhältnissen die relevanten Kennzahlen für die Leistungsfähigkeit der Modelle ermitteln. [FAWC06; JOSH20; SAMM10; TING10]

		Vorhersage		
		Positiv (i.O.)	Negativ (n.i.O.)	
Wahrheit	Positiv (i.O.)	Richtig-Positiv (TP)	Falsch-Negativ (FN)	<b>Recall</b> $\frac{TP}{TP + FN}$
	Negativ (n.i.O.)	Falsch-Positiv (FP)	Richtig-Negativ (TN)	<b>Specificity</b> $\frac{TN}{TN + FP}$
		<b>Precision</b> $\frac{TP}{TP + FP}$		<b>Accuracy</b> $\frac{TP + TN}{TP + TN + FP + FN}$

Abbildung 5.10: Konfusionsmatrix und abgeleitete Performanzmetriken, eigene Darstellung basierend auf [BUXM21; TING10].

Die Accuracy (ACC) ist eine zusammenfassende Metrik dafür, inwieweit die Prognosen mit den realen Werten übereinstimmen und wird berechnet nach

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \quad (5.28)$$

Für eine detaillierte Bewertung, inwiefern ein Modell in der Lage ist die jeweilige Klasse zu identifizieren, lassen sich Precision (PREC) und Recall (REC) berechnen. Die

$$PREC = \frac{TP}{TP + FP} \quad (5.29)$$

lässt Rückschlüsse auf die Menge der korrekt vorhergesagten positiven Ereignisse ziehen. Der REC (Gleichung 5.30) dahingegen ermöglicht die Beurteilung, welcher Anteil der i. O.-Geometrieelemente durch das Modell identifiziert werden konnte.

$$REC = \frac{TP}{TP + FN} \quad (5.30)$$

Diese Metriken fokussieren sich auf die Positiv-Klasse, welche im vorliegenden Anwendungsfall die maßhaltigen Geometrieelemente beschreibt. Jedoch sind insbesondere die FP-Vorhersagen von n. i. O.-Geometrieelementen die aus fertigungstechnischen Aspekten kritischen Vorhersagen, da bei einem produktiven Einsatz der Modelle ein Bauteil, welches nicht alle Toleranzanforderungen erfüllt, durch eine FP-Vorhersage fälschlicherweise als i. O. eingestuft wird. Das Bauteil könnte demnach in den nächsten Schritt der Produktion gelangen. Die Weiterverarbeitung sowie Montage eines Werkstücks, das außerhalb

seiner Toleranzen gefertigt wurde, erfüllt die gewünschten Funktionen nicht und hält den wirkenden Betriebslasten unter Umständen nicht stand. Dies gilt es möglichst zu vermeiden, da durch eine derartige Falscheinschätzung, infolge der nicht garantierten Funktionsfähigkeit, potenziell ein hoher technischer, sicherheitskritischer und ökonomischer Schaden für das Unternehmen und dessen Kunden entstehen kann. Aus diesem Grund fließt zusätzlich die Specificity (SPEC)

$$SPEC = \frac{TN}{TN + FP} \quad (5.31)$$

in die Bewertung der Modellvorhersagen mit ein. Darüber hinaus wird die absolute Anzahl an FP, welche es aufgrund der kritischen Bedeutung im Kontext des Anwendungsfalls zu minimieren gilt, mit betrachtet.

Die Receiver-Operating-Characteristic (ROC) ist weit verbreitetes Instrument, um die Leistungsfähigkeit von Klassifikatoren zu beurteilen. Die ROC-Kurve entsteht durch Auftragen der Richtig-Positiv-Rate (engl.: True Positive Rate) (TPR) als Synonym für den REC, gegen die Falsch-Positiv-Rate (engl.: False Positive Rate) (FPR). Die FPR entspricht dem Anteil an FP zu allen negativen Datenpunkten im Datensatz, welcher sich dementsprechend durch  $1 - SPEC$  ermitteln lässt. Reduziert sich der Schwellenwert eines Klassifikators, hat dies eine höhere Anzahl an Positiv-Vorhersagen zur Folge, wodurch sich der REC erhöht. Im gleichen Zug verschlechtert sich jedoch die SPEC und umgekehrt. Zwischen REC und SPEC besteht demnach ein indirekt proportionaler Zusammenhang, welcher je nach gewähltem Schwellenwert die Verläufe der beiden Metriken bestimmt. Für die Ermittlung der ROC-Kurve führt eine Erhöhung der FPR zu einer Erhöhung der TPR. Abbildung 5.11 stellt beispielhafte ROC-Kurven unter Berücksichtigung sämtlicher Schwellenwerte dar. Ein zufällig prognostizierender Klassifikator wird in diesem Diagramm mit der gestrichelten Linie dargestellt. Je besser ein Modell zwischen beiden Klassen unterscheiden kann, desto weiter entfernt sich die Kurve in Richtung der linken oberen Ecke. Für einen Vergleich verschiedener ROC-Kurven bzw. Modelle lässt sich der AUC berechnen. Je näher der ROC AUC Wert bei 1 liegt desto besser kann das Modell zwischen den beiden Klassen unterscheiden. Ein perfektes Klassifikationsmodell erreicht demnach einen  $ROC\ AUC = 1$ . [FAWC06; FLAC17; GÉRO20; LARN21]

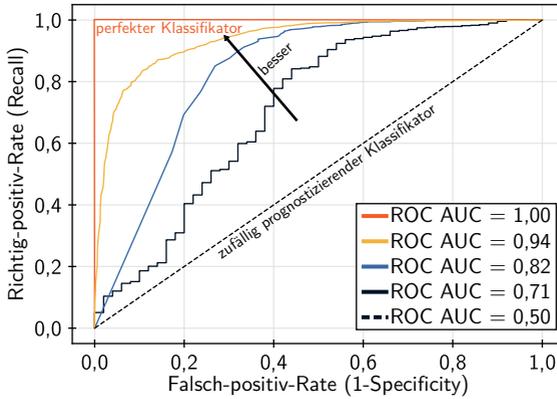


Abbildung 5.11: Beispielhafte ROC-Kurven für unterschiedlich leistungsfähige Klassifikationsmodelle.

### 5.3.6 Softwaretechnische Umsetzung der Prozesskette

Die Implementierung der MLPK mit den beschriebenen Teilprozessen aus den Kapiteln 5.3.2 – 5.3.5 wird modular in der Programmiersprache Python durchgeführt. Der modulare Aufbau ermöglicht neben der verbesserten Übersicht insbesondere eine flexible Erweiterung mit zusätzlichen Modulen und Funktionen. Den Ausgangspunkt bildet ein Datenordner („data“) mit zwei Unterverzeichnissen, welche zum einen die gemäß Kapitel 5.2 segmentierten Zeitreihendaten in tabellarisch strukturierten Dateien für jedes gefertigte Werkstück und zum anderen die zugehörigen Messdaten aus dem KMG (strukturiert gemäß Kapitel 4.2.4) enthalten.

Die Steuerung der Prozesskette übernehmen die zwei Hauptmodule `feature engineering.py` und `featuremachinelearning.py`, welche über die Konfigurationsdatei `PipelineConfig.json` parametrieren werden. Diese enthält für jeden Teilprozess ein individuelles Konfigurationsobjekt, welches die notwendigen Einstellungen und Parameter für den Ablauf der Prozesskette bereitstellt. Damit werden unter anderem Dateipfade konfiguriert, die Domäne der Merkmalsextraktion, die Skalierungsmethode und die Einstellungen zum Merkmalsselektionsalgorithmus (vgl. Kapitel 5.3.3) parametrieren sowie der Umfang des Trainingssets gewählt. Zusätzlich enthält das Objekt `MachineLearningConfig` die zu nutzenden ML-Algorithmen zusammen mit den individuellen Hyperparameterarrastern für die Gittersuche. Eine beispielhafte `PipelineConfig.json` ist Anhang A.6 zu entnehmen.

### **Merkmalsextraktion und -selektion (featureengineering.py)**

Das Modul `featureengineering.py` führt die Extraktion der Merkmale, die Erstellung der benötigten Datensätze für das ML und die Merkmalsselektion durch. Der schematische Aufbau der Teilmodule ist in Abbildung 5.12 dargestellt. Für die Merkmalsextraktion wird die Klasse `FeatureExtractorTSFEL` aus dem Modul `featureextractionTSFEL` importiert und mittels der `FeatEngConfig` das erzeugte Klassenobjekt initialisiert. Die Methode `processFeatureExtraction()` berechnet die Merkmale, basierend auf den konfigurierten Merkmalen aus der `features.json` (vgl. Kapitel 5.3.2) einschließlich der gewählten Signale sowie der Domäne aus der `FeatEngConfig`, für jedes Geometrieelement innerhalb eines Werkstücks. Für die Berechnung der Merkmale über mehrere Werkstücke hinweg dient eine `for`-Schleife, welche über das Softwarepaket `multiprocessing` parallelisiert ist. Dies ermöglicht eine parallele Berechnung der Merkmale mit  $N_{\text{threads}} = N_{\text{Kerne}} - 1$ , womit je nach gewählter Rechenhardware die Nutzung sämtlicher zur Verfügung stehenden Prozessorkerne gewährleistet wird. Ausgangsseitig entsteht für jedes Werkstück eine tabellarisch strukturierte Datei, welche die ermittelten Merkmalsvektoren für jedes Geometrieelement enthält.

Im Vorfeld der Merkmalsselektion ist es zunächst notwendig, den vorliegenden Merkmalsvektoren die entsprechenden Zielwerte für die Prognose zuzuordnen und den Datensatz in einen Trainings- und einen Testdatensatz aufzuteilen. Die Aufteilung vor der Merkmalsauswahl hat den Vorteil, dass ab diesem Zeitpunkt ausschließlich der Trainingsatz für die algorithmische Auswertung der Daten genutzt wird. Dies gewährleistet, dass die genutzten Algorithmen – sowohl für die Selektion als auch für die Prognosemodellerstellung – zu keinem Zeitpunkt Instanzen aus dem Testdatensatz verwenden. Dies ist wesentlich, um eine Verzerrung (engl.: Bias) der Ergebnisse bei der abschließenden Evaluation auf dem Testdatensatz zu verhindern und die Generalisierungsfähigkeit der Modelle adäquat beurteilen zu können. Bei einer Nutzung von Testdaten in einem der vorherigen Schritte (Skalierung, Selektion, Modelltraining) passen sich die Algorithmen zusätzlich an diese Daten an, was fälschlicherweise in zu optimistischen Performanzschätzungen münden kann. [BOIS22a] Dafür trennt die Methode `BuildXDatasets()` aus der mittels `DataSetConfig` instanziierten Klasse `XDataSetBuilder` des Moduls `BuildDataSets` die vorliegenden Werkstückdatensätze in jeweils einen Gesamtdatensatz für jedes Geometrieelement auf. Die Umwandlung der KMG-Messdaten, in die für die Klassifikation benötigten binären Zielwerte je Geometrieelement, übernimmt die Funktion `buildyClassificationLabels()` aus der Klasse `yLabelBuilder`. Die Ergebnisse aus den beiden Schritten fließen in die `TrainTestDataSet-`

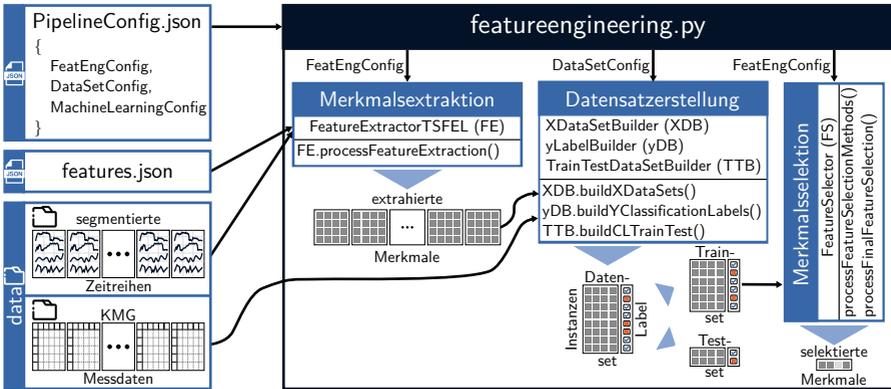


Abbildung 5.12: Schematische Darstellung des Moduls für die Merkmalsextraktion und -selektion.

`Builder`-Klasse ein, woraufhin eine geschichtete (engl.: stratified) Aufteilung in individuelle Trainings- und Testdatensätze für jedes Geometrieelement unter Berücksichtigung der Häufigkeitsverteilung der Zielwerte erfolgt. Die Aufteilung basiert gemäß konfiguriertem Anteil an Instanzen für das Training aus der `DataSetConfig`. Die geschichtete Vorgehensweise gewährleistet, dass sowohl im Trainings- als auch im Testdatensatz die prozentualen Häufigkeitsverteilungen der Klassen aus dem Gesamtdatensatz beibehalten werden, wodurch insbesondere im Fall kleiner Datensätze mit unausgeglichene Klassenverteilungen höhere Prognosegüten erreicht werden können.

Die `FeatureSelector`-Klasse stellt den in Kapitel 5.3.3 vorgestellten Algorithmus für die geometrieelementindividuelle Merkmalsselektion bereit. Für die Anwendung des Varianzschwellenwertfilters und der mittels `FeatEngConfig` konfigurierten Selektionsalgorithmen (vgl. Abbildung 5.4) dient die Methode `processFeatureSelectionMethods()`. Diese führt vor der algorithmischen Auswahl eine Skalierung der Merkmale durch, um den Skalenbereich der Werte über die verschiedenen Merkmale hinweg anzugleichen. In der Regel sind datengetriebene Algorithmen nicht in der Lage, mit numerischen Eingabedaten auf unterschiedlichen Skalen zielführend umzugehen [GÉRO20]. Dafür stehen innerhalb der MLPK die Standardisierung (`StandardScaler()`) und Normalisierung (`MinMaxScaler()`) aus dem Scikit-learn-Modul `sklearn.preprocessing` zur Verfügung. Bei der Standardisierung wird den Merkmalswerten zunächst der Mittelwert der Merkmalsverteilung abgezogen und schließlich durch deren Standardabweichung geteilt. Dies resultiert in einem

Mittelwert von 0 und einer Varianz von 1 für das standardisierte Merkmal über sämtliche Instanzen hinweg. Die Normalisierung verschiebt und skaliert die Werte eines Merkmals in ein resultierendes Intervall von  $[0, 1]$ . Die zu nutzende Skalierungsmethode wird ebenfalls über die `FeatEngConfig` vorgegeben. Die Methode `processFinalFeatureSelection()` nutzt die je Algorithmus und Geometrieelement ausgewählten Merkmale und führt die Erstellung der finalen Merkmalssätze unter Berücksichtigung des konfigurierten Schwellenwerts  $SW_{fs,prop}$  durch. Als Ergebnis stehen je Geometrieelement individuell selektierte Merkmale, welche in die anschließende Prognosemodellerstellung einfließen.

### Maschinelles Lernen (`featuremachinelearning.py`)

Für die Erstellung der Prognosemodelle ist das Modul `featuremachinelearning.py` (vgl. Abbildung 5.13) verantwortlich, welches als Hauptkomponente die Klasse `CLFeatureLearner` aus dem Modul `featureclassification` nutzt. Die `MachineLearningConfig` übergibt dem Objekt die zu nutzenden Algorithmen samt Hyperparameterraster während der Instanziierung. In einer darauffolgenden `for`-Schleife erfolgt die individuelle Prognosemodellerstellung für jedes Geometrieelement mittels klassenzugehöriger Methoden. Diese `for`-Schleife ist ebenfalls parallelisiert implementiert, um das Algorithmentraining für mehrere Geometrieelemente parallel durchführen zu können. In jeder Schleifeniteration laufen vier Teilprozessschritte ab. Im ersten Schritt müssen mit der Funktion `readTrainTestDataset()` Trainings- und Testdatensatz für das entsprechende Geometrieelement unter Berücksichtigung der ausgewählten Merkmale eingelesen werden. Das Trainieren der Algorithmen auf dem Trainingsdatensatz gemäß Kapitel 5.3.4 samt zuvor in der Selektion bestimmter Skalierungsmethode übernimmt die Methode `runTrainingClassification()`. Die erstellten Prognosemodelle werden schließlich von `runPredictionClassification()` genutzt, um die modellbasierten Prognosen sowohl auf dem Trainings- als auch auf dem Testdatensatz durchzuführen. Dies ermöglicht den Vergleich der Prognosegüten zwischen den Modellen auf bereits bekannten mit unbekanntem Daten und dient als Instrument für die Einschätzung, ob und inwiefern eine Über- bzw. Unteranpassung der Modelle aufgetreten ist. Die Berechnung der Metriken gemäß Kapitel 5.3.5 zur Evaluation der Prognosegenauigkeit wird durch die `reportMetrics()` Funktion durchgeführt. Am Ende der Durchläufe resultieren bei der Nutzung sämtlicher beschriebenen Algorithmen aus Kapitel 5.3.4 neun Prognosemodelle je Geometrieelement mit den individuell erreichten

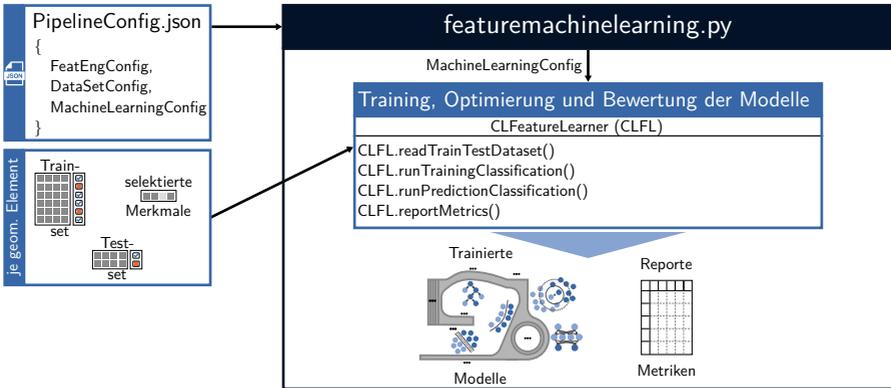


Abbildung 5.13: Schematische Darstellung des Moduls für die Prognosemodellherstellung.

Prognosekennzahlen, in Form von zusammenfassenden Dateireporten (metricsReports) in Tabellenform.

Die entwickelte Prozesskette läuft vollständig automatisiert ab und benötigt lediglich eine initiale Parametrierung der `PipelineConfig.json`. Die Parallelisierung zeitaufwendiger und sich wiederholender Teilprozesse ermöglicht eine effiziente Ausnutzung der zur Verfügung stehenden Prozessorkerne, wodurch sich eine Berechnungsdauer, je nach Anzahl an Datensätzen und gewählter Algorithmenkonfiguration, im niedrigen zweistelligen Minutenbereich<sup>7</sup> selbst auf marktüblich verbauten Prozessoren (z. B. Intel Core i7-1165G7 mit 4 Kernen und 8 logischen Prozessoren) realisieren lässt. Die Datenübergabe zwischen den Modulen erfolgt durch Schreib- und Lesevorgänge auf die Festplatte, wodurch die vollständig unabhängige Lauffähigkeit der einzelnen Module gewährleistet und die benötigte Kapazität des Arbeitsspeichers während der Berechnung reduziert wird.

Mit der Zusammenführung und Implementierung der Teilprozesse aus den vorangegangenen Ausführungen zur MLPK wird die Möglichkeit geschaffen, automatisiert KI-basierte Qualitätsprognosemodelle zu erstellen und die Frage nach den notwendigen Teilprozessen aus der dritten Forschungsfragestellung beantwortet.

<sup>7</sup>bei der Nutzung von 150 Trainingsinstanzen je Geometrieelement des Referenzwerkstücks aus Kapitel 4.1.2

## 5.4 Datengenerierung

Die datenbasierten Prognosemodelle benötigen in der Trainingsphase eine entsprechend repräsentative Datengrundlage, welche im Kontext der Klassifikation jeweils Datenpunkte für die zu prognostizierenden Klassen i. O. und n. i. O. enthält. Außerdem gilt es, zusätzliche Daten für das Testen der Modelle bereitzustellen, um eine Evaluation der Modelle sowie eine begründete Modellauswahl treffen zu können. Für die Generierung einer Datenbasis wird im Folgenden ein experimentelles Vorgehen entwickelt und im Rahmen dieser Arbeit durchgeführt.

Kapitel 2.3.2 beschreibt den Zerspanprozess als Systemdarstellung und diskutiert die verschiedenen Einflussgrößen auf die Werkstückqualität. Eingangs- und Störgrößen auf den Prozess beeinflussen das Ergebnis und die in diesen Fall relevanten Wirkgrößen in Form der Qualitätskriterien (vgl. Abbildung 2.11). Störgrößen resultieren in der Regel aus veränderten Eingangsgrößen oder treten zufällig auf. Deshalb wird für eine möglichst realitätsnahe experimentelle Datengenerierung gezielt Einfluss auf die Eingangsgrößen, welche durch System- und Stellgrößen charakterisiert sind, genommen. Dies dient dem Zweck, Unterschiede in Form- und Maßhaltigkeit der gefertigten Werkstücke herbeizuführen. Ausgehend von Kapitel 2.3.2 unter der Einbeziehung von [BENA03] lassen sich werkzeug-, werkstück-, maschinen- und strategiebedingte Einflussfaktoren unterscheiden. Die aus der Literaturrecherche basierend auf [BREC17], [DENK11], [ELLI19], [GROSS06], [HEIS92], [KLOC18] und [PAUC08] identifizierten detaillierten Einflussfaktoren aus den einzelnen Gruppen sind in Abbildung 5.14 zusammengetragen.

Diese werden unter Berücksichtigung einer einfachen und übertragbaren Umsetzbarkeit mit realitätsnahen Fehlereinflüssen sowie unter Beachtung sicherheitskritischer Aspekte in einen Versuchsplan transferiert. Dieser sieht die Fertigung ohne zusätzliche Prozesseinflüsse für die Erzeugung von maßhaltigen Werkstücken sowie die Herstellung von Bauteilen unter der Einwirkung von qualitätsbeeinflussenden Fehlereinflüssen vor. Tabelle 5.4 fasst die vorgesehenen Fehlereinflüsse (engl.: Error Codes) (EC) zusammen.

Im Wesentlichen gilt für die datenbasierte Modellierung der Grundsatz: „Je höher die Anzahl an für die Prognoseaufgabe relevanten Trainingsdaten, desto leistungsfähiger die damit antrainierten Modelle“. Jedes zusätzlich gefertigte Werkstück außerhalb des geplanten Produktionsablaufs erzeugt höhere Kosten, deshalb gilt es die Trainingsdatenbasis für die Modellierung auf ein notwendiges Minimum zu reduzieren. Dies ist insbesondere im Hinblick auf die gezielte Herstellung von Ausschussteilen für Umsetzung der binären Klassifi-

kation ein entscheidender Faktor. Für einen effizienten Versuchsablauf wird aus den genannten Gesichtspunkten das in Kapitel 4.1.2 vorgestellte Musterwerkstück auf den zur Verfügung stehenden Materialblöcken (vgl. Abbildung 4.3) schachbrettförmig für die Fertigung angeordnet. Darüber hinaus werden die Rohmaterialblöcke verdreht ( $10^\circ$  um die Z-Achse der Maschine) auf dem Maschinentisch aufgespannt. Somit sind bei jeder Vorschubbewegung am Werkstück jeweils die X- und Y-Achse beteiligt. Jeder Materialblock bietet den Platz für  $2 \times 10$  Musterwerkstücke je Seite. Insgesamt können mit einem Block bei Nutzung von Vorder- und Rückseite 40 Musterwerkstücke gefertigt werden. Die Anordnung innerhalb einer Gruppe von 10 Werkstücken (entspricht einer Versuchskonfiguration) erfolgt in zwei Spalten à 5 Werkstücken (vgl. Abbildung 5.15). Jede Gruppe wird in einer Aufspannung und einem Durchlauf gefräst, woraufhin eine  $180^\circ$  Drehung des Materialblocks um die Z-Achse die Fertigung der zweiten Gruppe in der selben Aufspannung ermöglicht. Der Gesamtversuchsdurchlauf umfasst 5 Materialblöcke mit insgesamt 200 gefertigten Musterwerkstücken. 15 der 20 Gruppen unterliegen dabei den 15 Fehlereinflüssen aus Tabelle 5.4. Die restlichen 5 Gruppen, aufgeteilt auf verschiedene Materialblöcke und Seiten, werden gemäß des zugrundeliegenden Fertigungsprozesses ohne künstlich eingebrachte Fehlereinflüsse hergestellt.

Im Anschluss an die Fertigung der Werkstücke dient das zur Ausstattung des PTW gehörende KMG Leitz PMM 864 mit der Messsoftware PC-DMIS (vgl. Kapitel 4.2.4) zur Vermessung der Geometrielemente. Die Vermessung jedes



Abbildung 5.14: Einflussfaktoren auf die Werkstückqualität.

der 15 Geometrielemente resultiert in der Auswertung von vorgegebenen 41 Form- und Lagetoleranzen je Werkstück. Anhang A.7 fasst die definierten Form- und Lagetoleranzen zusammen.

Im Rahmen dieser Arbeit werden drei Versuchsdurchläufe für die Generierung von insgesamt drei Datensätzen (DS) auf den beiden WZM DMC 850V und G350 (vgl. Kapitel 4.1.1) durchgeführt. Diese sind in Tabelle 5.5 zusammen mit den relevanten Informationen zusammengefasst. Während der Fertigung kommt für die prozessparallele Datenaufzeichnung die in Kapitel 4.2.1 beschriebene Datenerfassung mittels Kontextualisierung (vgl. Kapitel 5.1) zum Einsatz. Für die weiteren Analysen durchlaufen die gewonnenen Datensätze die automatisierte Datensegmentierung (vgl. Kapitel 5.2) und die qualitätsbezogenen Messdaten werden durch den in Kapitel 4.2.4 beschriebenen Softwarebaustein zur Verfügung gestellt.

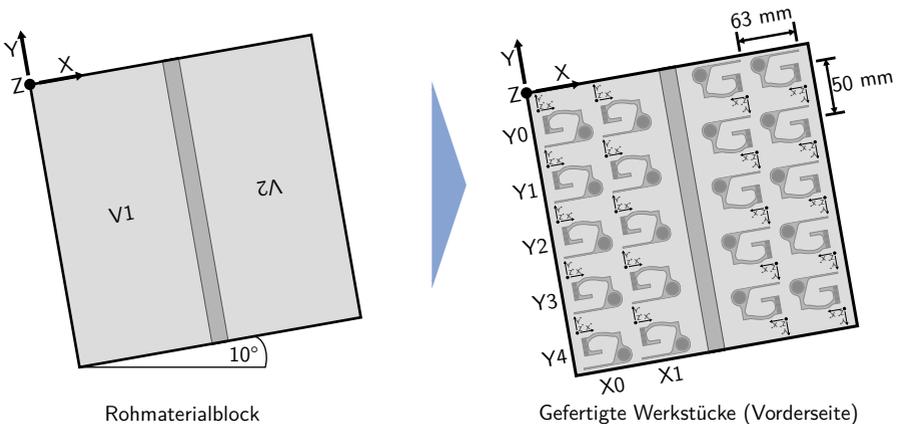


Abbildung 5.15: Anordnung der Musterwerkstücke auf einem Materialblock am Beispiel der Vorderseite.

Tabelle 5.4: Fehlereinflüsse zur Generierung einer Datengrundlage.

EC	Einfluss- kategorie	Bezeichnung	Kurzbeschreibung
00	-	-	Fertigung ohne Beeinflussung. (WS_002787)
11	werkzeug- bedingt	Falsches Werkzeug 1	Werkzeug mit $N_z = 5$ Schneiden. (WS_002790)
12	werkzeug- bedingt	Falsches Werkzeug 2	Werkzeug mit $N_z = 3$ Schneiden. (WS_002792)
13	werkzeug- bedingt	Auskraglänge	Einspannung des Werkzeugs mit einer verlängerten Auskraglänge in Höhe von 5 mm.
14	werkzeug- bedingt	Radiusmanipulation	Manipulation des Werkzeugradius in der Korrekturtabelle. $r_{\text{wkz,manipuliert}} = r_{\text{wkz,original}} + 0,1 \text{ mm}$
15	werkzeug- bedingt	Längenmanipulation	Manipulation der Werkzeuglänge in der Korrekturtabelle. $l_{\text{wkz,manipuliert}} = l_{\text{wkz,original}} - 1 \text{ mm}$
21	werkstück- bedingt	NPV-Verschiebung	Verschiebung der NPV vor Schruppen in X-Richtung um 0,1 mm.
22	werkstück- bedingt	NPV-Rotation	Rotation der NPV vor Schruppen um Z-Achse um $0,1^\circ$ .
23	werkstück- bedingt	Verkipfung 1	Verkipfung des Materialblocks um X-Achse durch einseitige Unter- lage einer Metallfolie (Dicke = 0,25 mm) vor Schruppen.

EC	Einfluss-kategorie	Bezeichnung	Kurzbeschreibung
24	werkstück- bedingt	Verkippung 2	Verkippung des Materialblocks um X-Achse durch einseitige Unterlage einer Metallfolie (Dicke = 0,25 mm) vor Schlichten.
25	werkstück- bedingt	Aufmaßschwankung	Verbleibendes Aufmaß für das Schlichten in Höhe von 1,5 mm.
31	maschinen- bedingt	CEC	Manipulation der Kompensationstabelle für die Durchgangskompensation (engl.: Cross Error Compensation) (vgl. Anhang A.8).
41	strategie- bedingt	Vorschub	Erhöhung des Zahnvorschubs um 20%. $f_{z,manipuliert} = f_{z,original} \times 1,2$
42	strategie- bedingt	Schnittgeschwindigkeit	Erhöhung der Schnittgeschwindigkeit um 20%. $v_{c,manipuliert} = v_{c,original} \times 1,2$
43	strategie- bedingt	Kühlschmierstoff	Schlichten ohne Kühlschmierstoff.
44	strategie- bedingt	G-Code	G-Code mit zeitoptimiertem Beschleunigungsverhalten (vgl. Anhang A.9).

Tabelle 5.5: Generierte Datensätze im Rahmen dieser Arbeit.

Bezeichnung	Maschine	Anzahl Werkstücke	Fertigungszeitraum
DS <sub>DMG1</sub>	DMC 850V	200	07.2020 - 08.2020 (Sommer)
DS <sub>GROB</sub>	G350	200	02.2021 (Winter)
DS <sub>DMG2</sub>	DMC 850V	200	03.2022 (Winter / Frühling)



# 6 Umsetzung der prozessparallelen Qualitätsprognose

In den vorigen Kapiteln wurden neuartige Verfahren zur kontextsensitiven Signalverarbeitung vorgestellt, welche eine Aufbereitung der erfassten Rohdaten aus der Werkzeugmaschinensteuerung für die datenbasierte Modellbildung ermöglichen. Die vorgestellte MLPK nutzt die aufbereiteten Signal- und Qualitätsdaten zusammen für eine automatisierte Ende-zu-Ende-Prognosemodellierung. In diesem Kapitel wird durch die Anwendung der entwickelten Lösungen gezeigt, welche Prognosegenauigkeiten die erstellten Modelle auf den generierten Datensätzen erzielen.<sup>1</sup> Infolge der zahlreichen Möglichkeiten für die Parametrierung der Prozesskette wird darauf aufbauend demonstriert, wie mittels entwickeltem Algorithmus eine automatisierte Parametrierung sowie eine Verbesserung der Prognosegüte mittels optimierter Prozesswissensmodelle erreicht werden kann.

## 6.1 Datenvorbereitung und Signalanalyse

Im Vorfeld der Prognosemodellierung gilt es zunächst, die generierten Datensätze zu analysieren, um mögliche Probleme oder Fehlstellen frühzeitig zu erkennen. Zusätzlich schafft eine vorab durchgeführte Datenanalyse ein verbessertes Verständnis der zugrundeliegenden Datenbasis.

### 6.1.1 Segmentierungsalgorithmus und Datenqualität

In Kapitel 5.2 wurde ein Algorithmus basierend auf kontextualisierten Signal- und Qualitätsdaten vorgestellt, der eine automatisierte Segmentierung der Zeitreihendaten ermöglicht. Dieser dient dazu, für die Qualitätsprognose irrelevante Datenpunkte für jedes Geometrieelement zu entfernen. Abbildung 6.1 (a) illustriert das Ergebnis anhand des Spindelstromsignals der LINESTEPS am Beispiel eines Werkstücks aus DS<sub>DMG1</sub>. Es ist zu erkennen, dass die durch kontextbasierte Filterung (vgl. Kapitel 5.1) erhaltenen Zeitreihen unterschiedlicher Länge nach der Segmentierung eine einheitliche Anzahl an Datenpunkten aufweisen. Der Algorithmus bezieht für das Schneiden der Zeitreihen sämtliche

---

<sup>1</sup>Ergebnisse aus Kapitel 6.1 und Kapitel 6.2 teilweise erstveröffentlicht in [FERT22b].

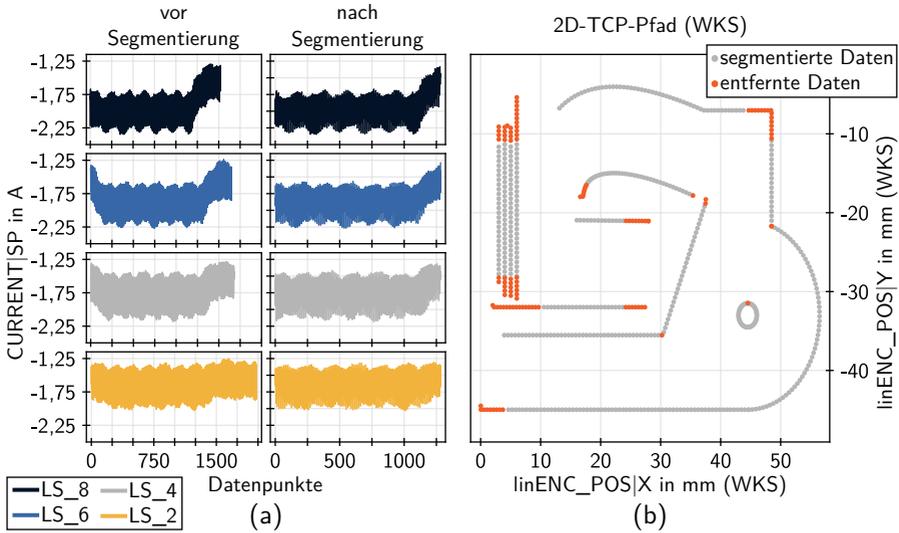


Abbildung 6.1: Ergebnisse der Segmentierung. (a) Vergleich des Spindelstroms am Beispiel der LINESTEPS vor und nach der Segmentierung. (b) 2D-TCP-Pfad für ein gesamtes Werkstück.

qualitätsrelevanten Geometrielemente mit ein. Die orange markierten Abschnitte aus Abbildung 6.1 (b) repräsentieren die durch den Algorithmus als irrelevant identifizierte und dementsprechend entfernten Datenpunkte über ein gesamtes Werkstück hinweg. Die Segmentierungsvorschrift lässt sich demnach auf die verschiedenen Geometrien anwenden und identifiziert zuverlässig die relevanten Datenpunkte für die Qualitätsprognose.

Abbildung 6.2 zeigt die Verteilung der Anzahl an Datenpunkten je Geometrielement über die drei generierten Datensätze hinweg. Für jeden Datensatz ist der Mittelwert über die gesamten Werkstücke<sup>2</sup> unabhängig von der genutzten WZM nahezu konstant. Bei Betrachtung der Standardabweichung (vgl. Abbildung 6.2 (b)) fällt auf, dass die Streuung von  $DS_{DMG1}$  über  $DS_{GROB}$  hin zu  $DS_{DMG2}$  deutlich abnimmt. Während der Durchführung der ersten beiden Versuchsdurchläufe konnten in der Datenerfassung (vgl. Kapitel 4.2.1) im Nachhinein noch jeweils Schwachstellen identifiziert werden, welche zu zufällig verteilten fehlenden Datenpunkten geführt haben. Während der Herstellung der Werkstücke aus  $DS_{DMG2}$  ist die Zuverlässigkeit der Datenaufzeichnung durch die vorige Behebung der Probleme deutlich erhöht worden.

<sup>2</sup>In diesem Fall 180 je Datensatz, da zwei Fehlereinflüsse aufgrund von Geschwindigkeitserhöhungen zu deutlich kürzeren Prozesszeiten und damit zu Zeitreihen geringerer Länge führen.

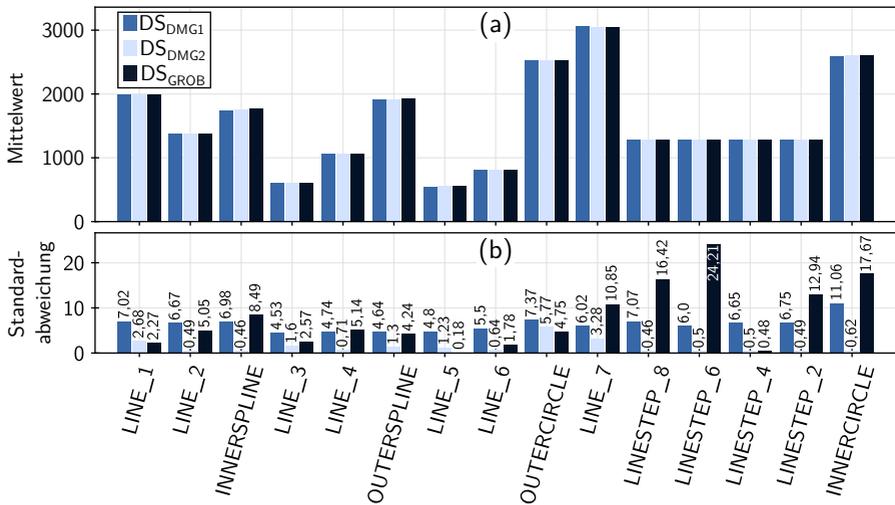


Abbildung 6.2: Verteilung der Anzahl an Datenpunkten je Geometrieelement und Datensatz. (a) Mittelwert über alle Werkstücke je Datensatz. (b) Standardabweichung der Verteilung.

Dies spiegelt sich auch in der Schwankung der Anzahl an Datenpunkten über Datensatz DS<sub>DMG2</sub> wider. Die Standardabweichungen liegen im Bereich zwischen 0,18 und 5,77<sup>3</sup> Datenpunkten je Geometrieelement. Daraus lässt sich schließen, dass bei zuverlässiger Datenaufzeichnung der entwickelte Segmentierungsalgorithmus präzise und automatisiert die erfassten Zeitreihen für die Prognosemodellerstellung vorbereitet. Die Analyse der Zyklusähler, welche sich inkrementell in jedem Abtastzeitpunkt um 1 erhöhen, ermöglicht eine Beurteilung über die Menge an fehlenden Datenpunkten. In DS<sub>DMG1</sub> fehlen bei Betrachtung der 180 Werkstücke 3707 (0,09%) Abtastzeitpunkte. DS<sub>GROB</sub> weist in diesem Fall weniger Fehlstellen als DS<sub>DMG1</sub> auf, jedoch erstrecken sich die einzelnen Fehlstellen über deutlich mehr Zeitpunkte. Während die größte Fehlstelle in DS<sub>DMG1</sub> 4 Abtastzeitpunkte beträgt, zeigt sich in DS<sub>GROB</sub> die längste Fehlstelle in Höhe von 299 Abtastzeitpunkten. Dies resultiert in 1527 (0,04%) fehlenden Abtastzeitpunkten innerhalb von DS<sub>GROB</sub>. Der Verlust von Datenpunkten konnte durch die Behebung der Probleme in der Datenerfassung bei DS<sub>DMG2</sub> auf 0 reduziert werden.

<sup>3</sup>EC14 Radiusmanipulation (vgl. Tabelle 5.4) führt zu einer ca. 50 ms längeren Prozesszeit für den OUTERCIRCLE im Vergleich zum Prozess ohne Fehlereinfluss.

Zusammenfassend verdeutlichen diese Ergebnisse, trotz der geringen Fehler in der Initialphase, die Nutzbarkeit und Zuverlässigkeit der vorgestellten kontextualisierten Datenerfassung (vgl. Kapitel 4.2.1 - 4.2.3 und 5.1) für die kontextsensitive Aufzeichnung hochfrequenter NC-Signaldaten. Der entwickelte Segmentierungsalgorithmus ermöglicht darüber hinaus eine zuverlässige und automatisierte Datenvorbereitung für die prädiktive Qualitätsprognose.

## 6.1.2 Verteilung der Werkstückqualität

Grundsätzlich ist es für überwachte ML-Verfahren notwendig den Eingabewerten die entsprechenden Zielwerte bzw. Labels zuzuordnen. Zudem benötigen die Algorithmen im Fall der Klassifikation für jede zu prognostizierende Klasse eine ausreichende Menge an Trainingsbeispielen im Datensatz. Für die angestrebte Qualitätsprognose entsprechen die Labels der binär codierten Qualität der Geometrieelemente, wodurch diese in die zwei Klassen i. O. und n. i. O. eingeordnet werden. Abbildung 6.3 fasst die resultierende Verteilung der erreichten Werkstückqualität je Geometrieelement für die jeweiligen Datensätze zusammen. Im Wesentlichen generiert der in Kapitel 5.4 entwickelte Versuchsplan auf beiden Maschinen Gut- und Ausschussteile. Über alle Geometrieelemente hinweg sind n. i. O.-Beispiele in den Datensätzen vorhanden. Die generierten Datensätze auf der DMC 850V weisen ähnliche Klassenverteilungen auf. Wohingegen die Fertigung der Werkstücke auf dem Bearbeitungszentrum G350 unter den gleichen technologischen Randbedingungen bei 10 der 15 Geometrieelemente zu einer leicht erhöhten Anzahl an n. i. O.-Elementen führt.

Aus der detaillierten Analyse der Fehlereinflüsse geht hervor, dass die eingebrachten Prozessstörungen sich je Maschine unterschiedlich auswirken. Die Heatmaps aus Abbildung 6.4 verdeutlichen, dass insbesondere die werkzeugbedingten Fehlereinflüsse (EC1x) zu einer Erhöhung des Ausschusses führen. Für die Generierung einer Datenbasis mit ausreichend n. i. O.-Instanzen sind diese demnach besonders geeignet und lassen sich verhältnismäßig einfach in den Prozess implementieren. Die werkstückbedingten Fehler (EC2x) beeinflussen in einem deutlich geringeren Maße die erreichte Qualität. EC22 erzeugt im Fall von  $DS_{DMG1}$  kein Ausschussteil. Der deutlichste Unterschied zwischen beiden Maschinen wird durch den Fehler „Verkippung 2“ (EC24) verursacht.  $DS_{GROB}$  weist diesbezüglich eine erheblich höhere Anzahl an n. i. O.-Geometrieelementen auf als  $DS_{DMG1}$ . Im Gegensatz dazu, reagiert der Prozess auf der DMC 850V deutlich empfindlicher auf die strategiebedingten Fehlereinflüsse (EC4x) mit einer Verschlechterung der Qualität. Zusammen-

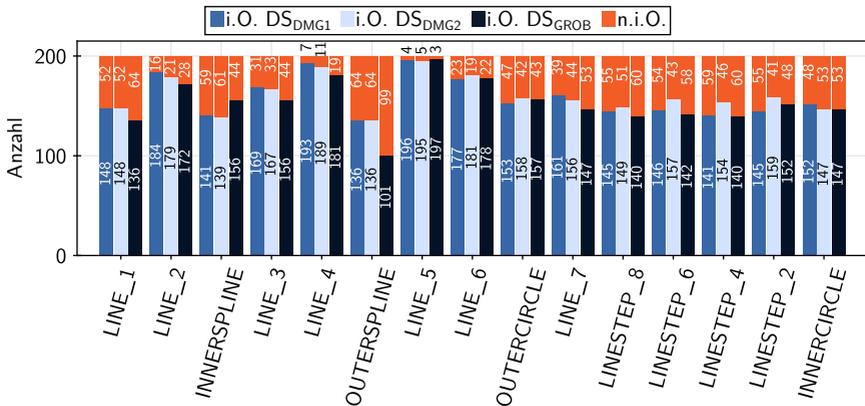


Abbildung 6.3: Verteilung der erreichten Werkstückqualität je Geometrieelement.

fassend lässt sich festhalten, dass jeder der definierten Fehlereinflüsse aus dem entwickelten Versuchsplan (vgl. Tabelle 5.4) auf unterschiedliche Weise den Prozess hinsichtlich resultierender Qualität beeinflusst. Dadurch wird das Ziel erreicht, einen Datensatz mit ausreichend Ausschussteilen zu erzeugen. Die Komplexität der angestrebten Klassifikationsaufgabe zeigt sich zudem durch die Ausschusselemente in EC00, welche planmäßig ohne Prozessbeeinflussung entstanden sind. Diese bilden zufällig aufgetretene, nicht erfasste Störungen und damit realitätsnahe Bedingungen ab.

Insgesamt handelt es sich bei den generierten Klassenverteilungen (vgl. Abbildung 6.3) um unausgeglichene (engl.: imbalanced) Datensätze je Geometrieelement, welche aufgrund der Unterrepräsentanz von Ausschussteilen eine besondere Herausforderung für das Training von überwachten ML-Algorithmen darstellen. Besonders auffällig dabei sind LINE\_4 und LINE\_5 mit jeweils sehr wenigen Ausschussinstanzen. In der Regel liegen jedoch aus der Fertigung überwiegend unausgeglichene Datensätze vor, da die Fehlerraten im eingefahrenen Zustand auf ein Minimum reduziert werden, wodurch lediglich wenige Fehlerbeispiele zustande kommen [EBER14]. Die generierten Datensätze basierend auf dem erstellten Versuchsplan repräsentieren somit ein realitätsnahe Abbild und ermöglichen durch das Vorhandensein von Gut- und Ausschussteilen für jedes Geometrieelement den Einsatz von überwachtem ML.

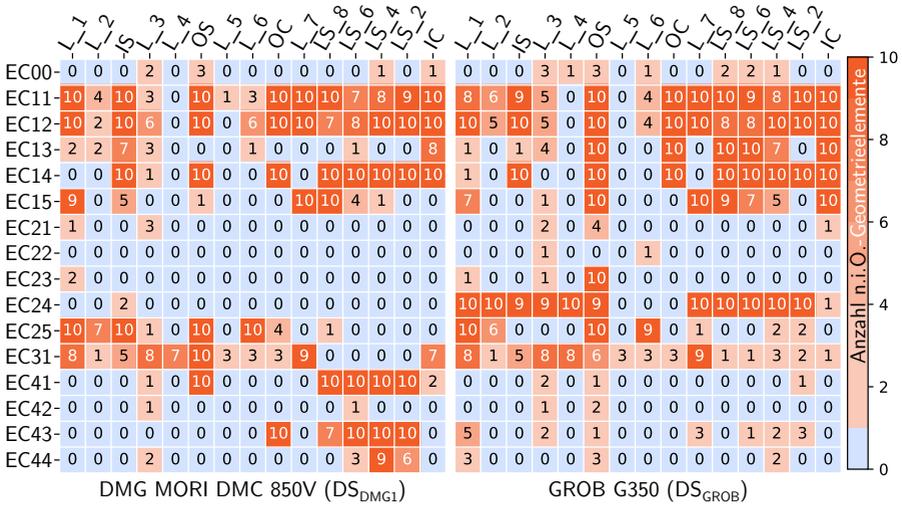


Abbildung 6.4: Heatmap der Verteilung von n. i. O.-Geometrieelementen je EC.

### 6.1.3 Signalauswahl und -reduktion

Die erfassten NC-Signaldaten werden unterschiedlich stark durch die Zerspanung beeinflusst und unterscheiden sich deshalb in dem enthaltenen Informationsgehalt über den Fräsprozess [FERT20]. Für den Einsatz der Daten in einem ML-System ist es demzufolge notwendig, aus den zur Verfügung stehenden Signalen aus dem Lageregelkreis (vgl. Kapitel 4.2.2) diejenigen Signale zu identifizieren, welche direkten Einfluss auf die zu prognostizierende Zielvariable ausüben bzw. Informationen zum Zustand der Zielvariable liefern. Auf die Zerspanung übertragen, bildet der im Signal enthaltene Informationsgehalt über den aktuellen Prozesszustand folglich die Ausgangsbasis für die Signalauswahl (vgl. Kapitel 2.2.4). Zusätzlich erschweren redundante Signale die Modellbildung und können die Prognosegenauigkeit negativ beeinflussen, weshalb diese möglichst für die Erstellung der Prognosemodelle zu vermeiden sind (vgl. Kapitel 2.2.5).

Um eine geeignete und übertragbare Signalauswahl für die Qualitätsprognose zu treffen, wird in den folgenden Ausführungen eine werkstück- und zielgrößenunabhängige Vorgehensweise gewählt. Diese basiert im ersten Schritt auf einer domänenspezifischen Bewertung unter Berücksichtigung des jeweiligen Signalur sprunges. Signale, welche Sollwerte für den Prozess vorgeben, unter-

liegen bei gleichbleibender Prozessführung keinen Veränderungen und liefern dementsprechend keine relevanten Informationen für die Qualitätsprognose. Folglich werden die Signalgruppen DES\_POS und CTRL\_POS aussortiert. TORQUE\_FFW und VEL\_FFW stellen Signale aus der dynamischen Vorsteuerung dar, welche basierend auf dem vorbereiteten Prozessverlauf zur Reduktion des Schleppfehlers beitragen und unabhängig von den resultierenden Messungen bzw. vom Zustand der Regelstrecke sind. Entsprechend liefern diese prozessunabhängige Informationen und leisten somit keinen direkten Beitrag für die Qualitätsprognose. LOAD entspricht der prozentualen Auslastung des jeweiligen Antriebs, welche bei Schlichtprozessen sehr geringe Werte annimmt und dient hauptsächlich dazu, einfach zu interpretierenden Rückmeldungen über potenzielle Komponentenüberlastungen zu liefern. Die POWER-Signale enthalten die geglättete Antriebswirkleistung [SIEM18c] und werden überwiegend für energetische Betrachtungen der WZM eingesetzt. Eine detaillierte Analyse der zugehörigen Signaldaten zeigt, dass diese beim betrachteten Schlichtprozess (bis auf vereinzelte Ausreißer) den Wert 0 aufweisen. Durch Einbeziehung in die Modellierung wird kein Mehrwert erwartet, weshalb diese Signale ebenfalls aussortiert werden.

Die verbleibenden Signalgruppen bilden prozessbeeinflusste Messgrößen aus dem Lageregelkreis ab und liefern potenziell wertvolle Informationen über den Prozess für die angestrebte Qualitätsvorhersage. Um Redundanzen und direkte Abhängigkeiten zu identifizieren, schließt sich eine lineare Korrelationsanalyse zwischen den Signalen basierend auf der Berechnung des Pearson-Korrelationskoeffizienten an. Dieser wird nach [BENE08] auf Grundlage der Arbeiten von [PEAR96], [DUNN74] und [RODG88] definiert als

$$r_{xy} = \frac{E^2[xy]}{\sigma_x^2 \cdot \sigma_y^2} \quad \text{mit} \quad W = \{r_{xy} \in \mathbb{R} \mid -1 \leq r_{xy} \leq 1\} \quad (6.1)$$

wobei  $E^2[xy]$  der Kreuzkorrelation zwischen den Variablen  $x$  bzw.  $y$  und  $\sigma_x^2$  bzw.  $\sigma_y^2$  der jeweiligen Varianz in den Signalen entsprechen. Daraus resultiert schließlich die Berechnungsformel

$$r_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (6.2)$$

wobei  $\bar{x}$  und  $\bar{y}$  den arithmetischen Mittelwerten der Signale entsprechen. Zwei Signale definieren sich im Rahmen dieser Arbeit ab einem betragsmäßigen Korrelationswert von

$$|r_{xy}| \geq 0,95 \quad (6.3)$$

als hoch korreliert. Je nach Vorzeichen herrscht eine positive bzw. negative Korrelation zwischen den untersuchten Variablen.

Abbildung 6.5 fasst das Ergebnis der Korrelationsanalyse zusammen. Die identifizierten Korrelationen lassen sich durch den physikalischen Zusammenhang der jeweiligen Signalaarungen erklären. Die Signale `linENC_POS` (Linearmaßstab auf der Vorschubachse) und `rotENC_POS` (Drehgeber am Antrieb) geben jeweils die aktuelle Ist-Position der Achse an zwei unterschiedlichen Messstellen an. Die mechanische Kopplung der beiden Signale führt zum Korrelationskoeffizient  $r_{xy} = 1$  für jede Signalaarung (X-, Y- und Z-Achse). Für die nachfolgende Modellbildung werden dementsprechend die jeweiligen linearen Positionssignale `linENC_POS` der beteiligten Achsen mit einbezogen, da diese Signale durch ihre Nähe zum Prozess potenziell stärkeren Einflüssen unterliegen. `TORQUE` und `CURRENT` weisen im betrachteten Anwendungsfall eine Soll- und Istbeziehung (vgl. Abbildung 4.7), ähnlich der von Soll- und Istpositionen auf, weshalb es dabei ebenfalls zu stark positiven Korrelationen kommt. Trotz der starken Korrelation zeigt eine visuelle Analyse am Beispiel von `OUTERSPLINE` und `INNERCIRCLE` (vgl. Abbildung 6.6) unterschiedliche Steigungen und qualitative Verläufe insbesondere nach Verfahrenänderungen der beteiligten Achsen. Darüber hinaus resultiert bei Betrachtung der zugehörigen Spindelsignale ein reduzierter Korrelationswert in Höhe von 0,88. Aus diesen Gründen kann nicht davon ausgegangen werden, dass eine der beiden Signalgruppen für sich, sämtliche relevanten Informationen liefert, weshalb beide Gruppen mit in die Modellbildung einfließen. Die vorherrschende starke positive Korrelation zwischen `CMD_SPEED` und `CTRL_DIFF2` lässt sich anhand der Gegenüberstellung der beiden Signalgruppen in Abbildung 6.7 nachvollziehen. Aufgrund der unterschiedlichen Werteskalen sind die Signale für einen vereinfachten Vergleich mittels Standardisierung auf die gleiche Skala transformiert dargestellt. Der Verlauf von `CTRL_DIFF2` folgt qualitativ der Sollgeschwindigkeit mit dem Unterschied, dass dieses Signal mit hochfrequenten Anteilen der Regelabweichung überlagert ist. Wie bereits in Kapitel 4.2.2 erwähnt, liefert die zur Regelabweichung gehörende Signalgruppe `CTRL_DIFF` die kompensierte Regelabweichung und `CTRL_DIFF2` die zugehörigen unkompensierten Werte. Demnach entspricht der hochfrequente Anteil in `CTRL_DIFF2` den Werten aus `CTRL_DIFF`. Infolge der hohen Korrelation von `CTRL_DIFF2` mit der Sollgeschwindigkeit `CMD_SPEED` kann `CTRL_DIFF2` aussortiert werden, um lediglich die Sollgeschwindigkeiten in der Modellbildung mit zu berücksichtigen. Zudem handelt es sich nach [SIEM22b] bei `CTRL_DIFF2` um exklusive Signale der Datenerfassungslösung Industrial Edge, welche bei der Wahl ei-

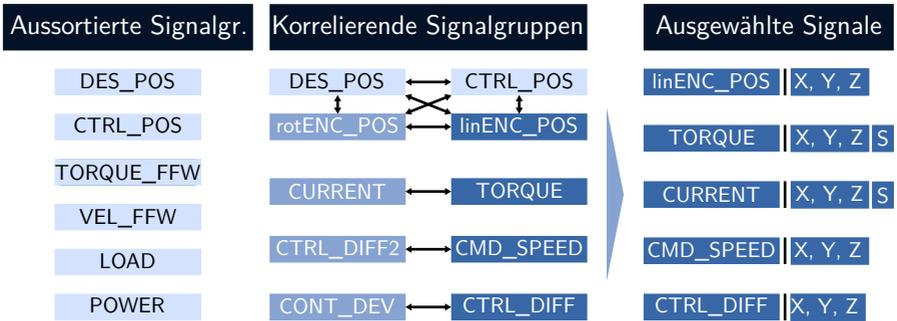


Abbildung 6.5: Korrelierende Signalgruppen und Signalauswahl.

ner alternativen Datenaufzeichnung an SINUMERIK-Steuerungen nicht zur Verfügung stehen. Die Sollgeschwindigkeit der Spindel zeigt keine signifikanten Schwankungen in den Signalwerten über die Werkstücke hinweg, weshalb dieses Signal ebenfalls aussortiert wird. Die letzte aussortierte Signalgruppe ist die steuerungsinterne Konturabweichung `CONT_DEV`, da deren Berechnung basierend auf der Regelabweichung `CTRL_DIFF` stattfindet (vgl. Kapitel 4.2.2) und damit die enthaltenen Informationen für die Prognose in den für die Modellerstellung ausgewählten Signalen bereits enthalten sind.

Zusammenfassend gehen aus der Signalauswahl die fünf in Abbildung 6.5 rechts gezeigten Signalgruppen zu den jeweiligen Maschinenachsen hervor, welche die Grundlage für das anschließende Modelltraining bilden. Aufgrund von Problemen bei der Datenerfassung zum Zeitpunkt der Versuchsdurchläufe auf der GROB Maschine, fehlen die `TORQUE`-Signale in `DS_GROB`. Darüber hinaus werden die Drehachsen der WZM G350 nicht weiter berücksichtigt, da für deren 3-achsige Nutzung eine Klemmung dieser Achsen notwendig ist und diese demnach nicht am Prozess beteiligt sind.

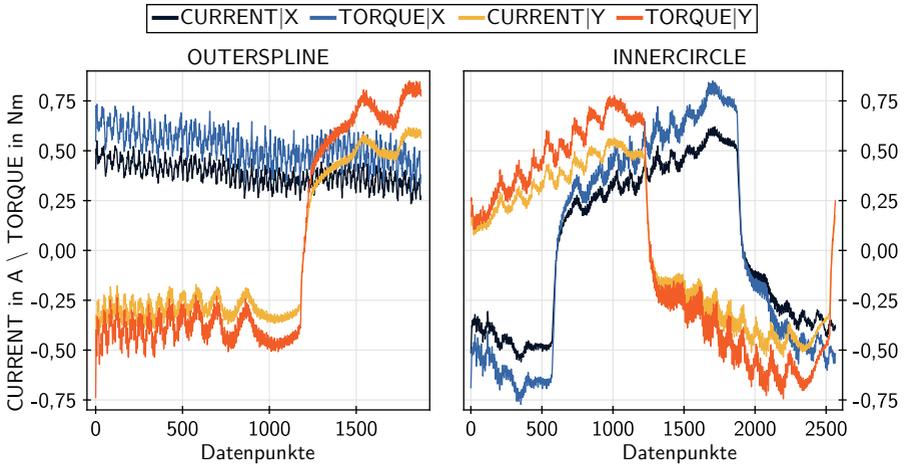


Abbildung 6.6: Unterschiede in Strom- und Drehmomentenverläufen der X- und Y-Achse am Beispiel der Geometrielemente OUTER SPLINE und INNERCIRCLE.

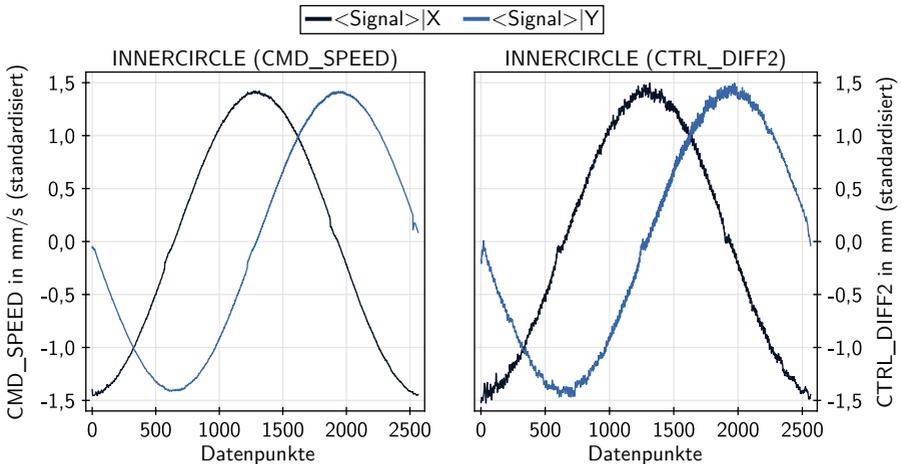


Abbildung 6.7: Gegenüberstellung CMD\_SPEED und CTRL\_DIFF2 der X- und Y-Achse am Beispiel des Geometrielements INNERCIRCLE.

## 6.2 Umsetzung der automatisierten Prognosemodellerstellung

Ausgangspunkt für die automatisierte Prognosemodellerstellung bildet die entwickelte MLPK aus Kapitel 5.3.6. Nach Erfassung (vgl. Kapitel 5.4) und Segmentierung der Signaldaten gemäß Kapitel 5.2 sowie deren Verknüpfung mit aufbereiteten Qualitätsdaten (vgl. Kapitel 4.2.4) liegen die gelabelten Datensätze vor, welche die benötigten eingangsseitigen Daten für die Ausführung der Pipeline darstellen. Die Parametrierung der MLPK erfolgt angelehnt an [FERT22b] und sieht folgende relevante Einstellungen vor:

- Domänen Merkmalsextraktion: statistisch, zeitbasiert, spektral (vgl. Kapitel 5.3.2). 60 Merkmale (vgl. Anhang A.3)
- Selektionsalgorithmen:  $S_{fs} = [\text{uniStat}, \text{LogisRe}]$  mit  $SW_{fs,prop} = 2$  und  $SW_{fs,comm} = 5$  (vgl. Kapitel 5.3.3)
- Skalierungsmethode: Standardisierung
- Aufteilung Datensätze: 150 im Trainings- und 50 im Testdatensatz (`train_size : 0.75`)

Zusätzlich enthält die `PipelineConfig.json` die ausgewählten Signale aus Abbildung 6.5 gemäß Kapitel 6.1.3. Für die automatisierte Erstellung der Prognosemodelle wird die parametrierte MLPK jeweils auf die generierten Datensätze aus Kapitel 5.4 angewendet. Dabei entspricht  $DS_{DMG1}$  dem Datensatz aus [FERT22b].

Abbildung 6.8 fasst die erreichte Prognosegenauigkeit auf den entsprechenden Testdatensätzen aufgeschlüsselt nach Geometrieelement und Algorithmus für  $DS_{DMG1}$  unter der Verwendung des `PropFeatSet` zusammen. Die Ergebnisse zeigen insgesamt leichte Verbesserungen im Gegensatz zu den dargestellten Werten aus [FERT22b]. Hintergrund dafür ist, dass zum Zeitpunkt der Veröffentlichung eine etwas geringere Anzahl an Merkmalen ( $43 \text{ Merkmale} \times 17 \text{ Signale} = 731$  zu  $60 \text{ Merkmale} \times 17 \text{ Signale} = 1020$ ) sowie ein reduziertes Hyperparameterraster der Algorithmen berücksichtigt wurde. Bei Betrachtung der Performanzmetriken ACC, PREC und REC lassen sich für jedes Geometrieelement mit Ausnahme von `LINE_3` jeweils Modelle mit hohen Werten über 90 % identifizieren. Teilweise werden für diese Metriken Prognosegenauigkeiten von 100 % erreicht. Insbesondere die hohen Werte für PREC und REC lassen darauf schließen, dass die Algorithmen in der Lage sind, die Klassenzugehörigkeit von i. O.-Geometrieelementen zu identifizieren. Je nach Geometrieelement schneiden die Modelle unterschiedlich gut ab, wobei keiner

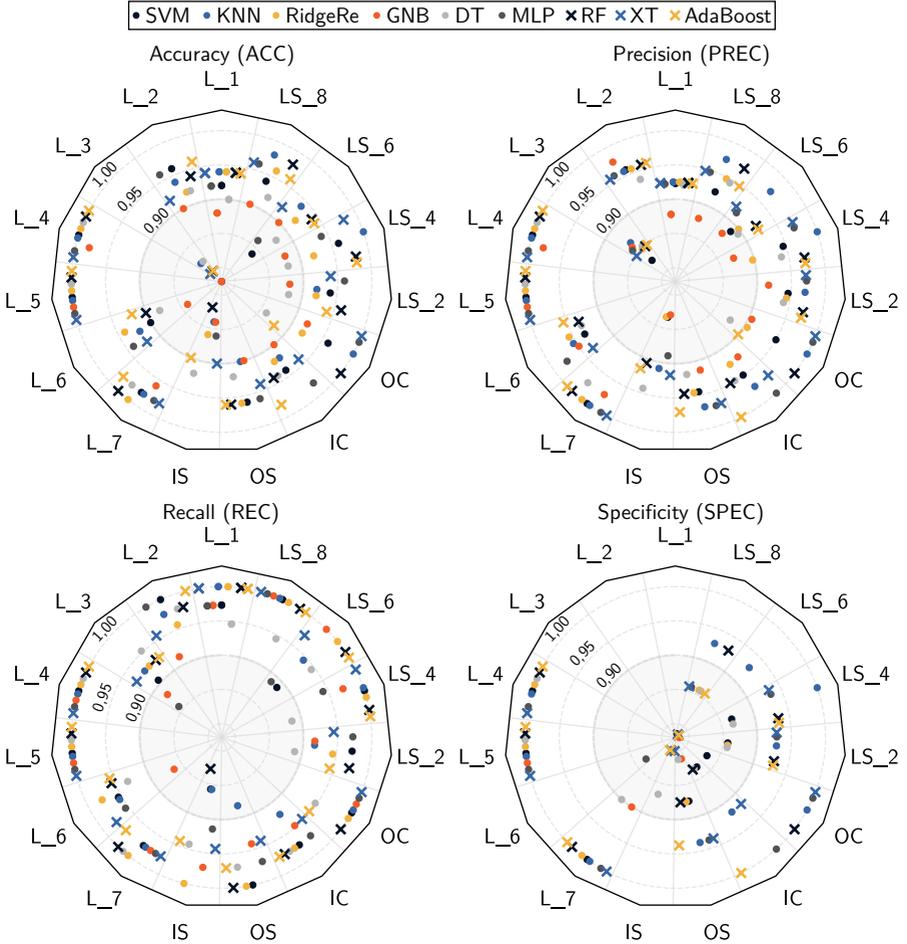


Abbildung 6.8: Erreichte Prognosegenauigkeiten je Algorithmus und Geometrieelement für  $DS_{DMG1}$ .

der Algorithmen mit besonderen Prognosegenauigkeiten über sämtliche Geometrielemente hinweg heraussticht. Zusätzlich zeigen die Ergebnisse, dass teilweise mehrere Algorithmen in der Lage sind, die Klassifikation für ein Geometrieelement bei gleicher Vorhersagegüte zu erzielen. Für die Prognose der Qualität ist es hingegen von höherer Bedeutung, die Modelle in die Lage zu versetzen, zuverlässig n. i. O.-Werkstücke zu detektieren. Die resultierenden Werte der SPEC, der dafür relevanten Bewertungsmetrik, fallen deutlich geringer aus, als diejenigen der zuvor diskutierten Metriken. Die Skala aus Abbildung 6.8 endet jeweils im Kreismittelpunkt bei ca. 0,8. Die fehlenden Punkte weisen dementsprechend Werte unter 80 % auf. Dies bedeutet im Umkehrschluss, dass im Gegensatz zur Identifikation von Gutteilen, Fehlteile mit den erstellten Modellen nicht ausreichend detektiert werden können. Daraus lässt sich schließen, dass die Daten von n. i. O.-Instanzen starke Ähnlichkeiten zu i. O.-Geometrieelementen aufweisen und aufgrund ihrer Unterrepräsentation in den Datensätzen (vgl. Kapitel 6.1.2) für die Modelle komplexer zu identifizieren sind bzw. die Algorithmen deren Abweichungen zu Gutteilen noch nicht ausreichend modellieren. Dieses Ungleichgewicht der Klassenbeispiele wirkt sich zudem negativ auf die erreichbaren Werte für die SPEC bereits bei wenigen FP-Vorhersagen aus. Der prozentuale Anteil ist aufgrund der geringen Anzahl an n. i. O.-Beispielen im Testset bei bereits einer Fehlvorhersage höher als bei Evaluation der weiteren Metriken. Trotzdem lassen sich für einige Geometrieelemente Modelle identifizieren, welche in der Lage sind, nahezu fehlerfreie Vorhersagen auf den unbekanntenen Daten des Testset zu treffen. Dies zeigt das Potenzial der datenbasierten Modellierung für die Qualitätsprognose.

Für die Evaluation der generierten Modelle auf den weiteren Datensätzen, fasst Abbildung 6.9 die ausgewerteten Metriken für jedes der 135 Modelle (9 Modelle  $\times$  15 Geometrieelemente = 135 Modelle) für die einzelnen Datensätze zusammen. Neben der zusätzlichen Berücksichtigung des ROC AUC wird für das verbesserte Verständnis jeweils der Mittelwert der aufgetragenen Metrik über sämtliche Modelle gebildet. Die erreichten Prognosegenauigkeiten von  $DS_{DMG1}$  und  $DS_{DMG2}$  befinden sich in ähnlichen Wertebereichen. Die MLPK ist demnach in der Lage ähnlich leistungsfähige Modelle mit einem zweiten Datensatz aus der selben WZM zu generieren. Deutlich zu erkennen ist zum einen wiederum der deutliche Abfall der SPEC bis hin zu teilweise 0%. In diesen Fällen ist das dahinterliegende Modell nicht in der Lage fehlerhafte Geometrieelemente zu erkennen. Zum anderen zeigt sich, dass der Durchschnitt des ROC AUC über alle Modelle bei Werten über 90 % liegt. Bei einer entsprechenden Auswahl resultieren Modelle, welche in der Lage sind,

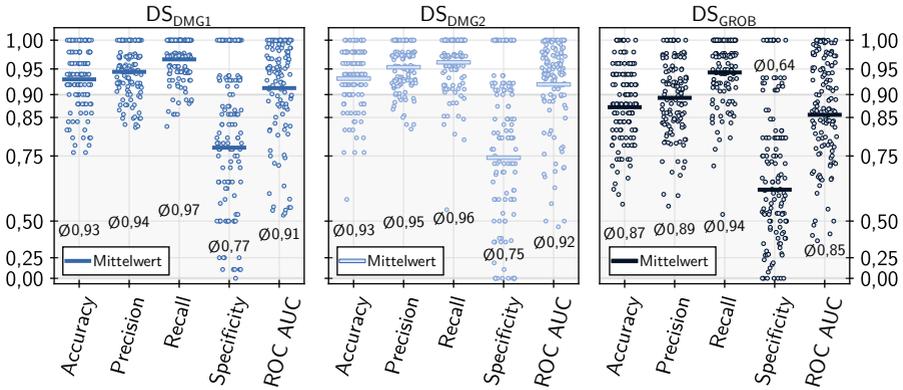


Abbildung 6.9: Vergleich der Prognosegenauigkeiten zwischen den Datensätzen. Jeder Punkt repräsentiert ein resultierendes Modell für ein Geometrieelement nach dem Durchlauf der MLPK.

beide Zielklassen zu unterscheiden. Die Prognosegenauigkeiten für DS<sub>GROB</sub> fallen im Vergleich deutlich ab und vier der fünf ausgewerteten Durchschnittsmetriken liegen unter 90 %. Gründe dafür können die größeren Fehlstellen in den erfassten Signaldaten (vgl. Kapitel 6.1.1), wodurch sich die Merkmalsausprägungen über den Datensatz hinweg instabiler verhalten, die fehlenden TORQUE-Signale (vgl. Kapitel 6.1.3) oder die möglicherweise ungünstig gewählte Parametrierung der MLPK für die WZM von GROB sein.

Bei der Nutzung des gemeinsamen CommFeatSet über alle Geometrielemente hinweg zeigt Tabelle 6.1, dass die erreichten durchschnittlichen Prognosegenauigkeiten deutlich gegenüber der individualisierten Merkmalsselektion abfallen. Dies bekräftigt die Annahme der hohen Komplexität in der Entscheidungsfindung, wodurch eine für jedes Geometrieelement individuell durchgeführte Merkmalsauswahl im Fall der vorliegenden Datensätze zu leistungsfähigeren Modellen führt. Die gemeinsam ausgewählten Merkmale eignen sich aufgrund der geringeren Prognosegenauigkeiten der Modelle nicht für einen Einsatz über beliebige Geometrielemente hinweg, weshalb dieser Ansatz in den weiteren Ausführungen nicht weiter verfolgt wird.

Zusammenfassend ermöglicht die entwickelte MLPK im Zusammenspiel mit den zugrundeliegenden vorverarbeiteten Daten aus NC und Qualitätssicherung eine automatisierte individualisierte Modellbildung für jedes qualitätsrelevante Geometrieelement an einem Bauteil. Aus der Diskussion der Ergebnisse geht hervor, dass die gewählte Parametrierung der MLPK bereits

Tabelle 6.1: Durchschnittswerte der Metriken. Vergleich PropFeatSet und CommFeatSet.

DS	Merkmalssatz	Durchschnittswert				
		ACC	PREC	REC	SPEC	ROC AUC
DS <sub>DMG1</sub>	PropFeatSet	0,93	0,94	0,97	0,77	0,91
	CommFeatSet	0,91	0,94	0,95	0,74	0,89
DS <sub>DMG2</sub>	PropFeatSet	0,93	0,95	0,96	0,75	0,92
	CommFeatSet	0,86	0,90	0,93	0,60	0,80
DS <sub>GROB</sub>	PropFeatSet	0,87	0,89	0,94	0,64	0,86
	CommFeatSet	0,83	0,88	0,89	0,59	0,84

leistungsfähige Modelle liefern kann, deren Prognosegenauigkeit jedoch für einen produktiven Einsatz noch nicht ausreichend ist. Die MLPK liefert zahlreiche Parametrierungsmöglichkeiten, die bei geeigneter Parameterkombination das Potenzial enthalten, die Leistungsfähigkeit der generierten Modelle zu steigern. Bei einem Durchlauf stehen zudem je Geometrieelement (in diesem Fall) 9 Modelle zur Verfügung. Es gilt daraus, das für die Qualitätsprädiktion passende Modell für den produktiven Einsatz auszuwählen. Im weiteren Verlauf dieser Arbeit wird dafür ein Algorithmus samt Modellbewertungs- und -auswahlmethode vorgestellt, welcher die Möglichkeit bietet, automatisiert die Parametrierung der MLPK vorzunehmen, um damit die am besten geeigneten Modelle zu erhalten und auszuwählen.

## 6.3 Algorithmus zur automatisierten Prozesskettenparametrierung<sup>4</sup>

Die Zielsetzung der automatisierten Parametrierung ist die Identifikation idealer Konfigurationsparameter für die entwickelte MLPK zur Erstellung optimierter Prognosemodelle für den genutzten Datensatz aus der spanenden Fertigung. Der Schwerpunkt liegt dabei auf dem in dieser Arbeit adressierten Anwendungsfall zur Prognose der Werkstückqualität basierend auf internen NC-Signaldaten von WZM. Neben der Parametrierung gilt es zum Zweck eines höchstmöglichen Automatisierungsgrads, die leistungsfähigsten Prognosemodelle für jedes untersuchte qualitätsrelevante Geometrieelement mittels einer domänenspezifisch angepassten Bewertungsmethode auszuwäh-

<sup>4</sup>Erstveröffentlichung in [FERT22c].

len. Dies ermöglicht Domänenexperten effizient optimierte Qualitätprognosemodelle für individuelle Werkstücke und WZM ohne manuelles Eingreifen zu erstellen. Damit steht die Beantwortung der vierten Forschungsfragestellung im Fokus der folgenden Ausführungen. Das nächste Kapitel erläutert zunächst die zugrundeliegenden Einzelschritte des entwickelten Optimierungsalgorithmus, woraufhin in Kapitel 6.3.2 eine detaillierte Beschreibung der untersuchten Parameterkombinationen erfolgt. Die Bewertungsmethode aus Kapitel 6.3.3 ermöglicht die Sortierung und Auswahl der Einzelmodelle nach ihrer Leistungsfähigkeit für die Qualitätsprognose. Abschließend stellt Kapitel 6.3.4 die Ergebnisse nach Anwendung des Algorithmus auf die drei zur Verfügung stehenden Datensätze dar.

### 6.3.1 Algorithmusbeschreibung

Ausgangspunkt des entwickelten Algorithmus ist die initial parametrisierte MLPK entsprechend Kapitel 6.2. Das Grundkonzept des Algorithmus besteht darin, Durchläufe (im Folgenden als „Run“ bezeichnet) der MLPK in unterschiedlichen Parametrisierungen zu berechnen, um basierend auf den erreichten Prognoseergebnissen auf einem unbekanntem Validierungsdatensatz geeignete Werte für die entsprechenden Parameter zu identifizieren. Ziel dabei ist es, diejenige Parametrisierung zu ermitteln, welche zu Modellen mit den höchsten Prognosegenauigkeiten führt. Die Suche der optimierten Parameterwerte erfolgt sequentiell durch schrittweise Optimierung in einzeln ablaufenden Optimierungsschritten (im Folgenden als „optStep“ bezeichnet), beginnend bei der Merkmalsextraktion hin zur Merkmalsselektion. Diese Vorgehensweise wird gewählt, da in Anbetracht einer vollfaktoriellen Untersuchung in diesem Fall über 600.000 Durchläufe der MLPK notwendig sind, was zu einem nicht vertretbaren Rechenaufwand führt. Der Ablauf eines optSteps sieht im Anschluss an die Berechnung aller in diesem Optimierungsschritt vorgesehenen Runs eine Evaluation der Modellprognosegenauigkeiten gemäß der in Kapitel 6.3.3 vorgestellten Bewertungsmethode vor. Diese Auswertung lässt Rückschlüsse auf die entsprechend leistungsfähigste Parameterkombination zu, woraufhin eine Anpassung der MLPK-Konfiguration erfolgt. Abbildung 6.10 illustriert den beschriebenen Ablauf des Algorithmus in Form eines Programmablaufplans.

Insgesamt werden die folgenden Optimierungsschritte berücksichtigt:

- Optimierung der Merkmalsextraktion
  - 1.1 `optStepDWT`: Optimierung der Hyperparameter für die DWT (`wavelet`, `level`)
  - 1.2 `optStepHHT`: Optimierung der Hyperparameter für die HHT (`max_imfs`)
  2. Optimierung der Domäne `temporal-spectral`
  3. Optimierung der zu nutzenden Domänen und Fensterfunktion
- Optimierung der Merkmalsselektion
  4. Optimierung der zu nutzenden Merkmalsselektionsalgorithmen,  $SW_{fs,prop}$  sowie Skalierungsmethode
- Finaler Durchlauf der MLPK mit optimierter Parameterkonfiguration
  5. Finaler Durchlauf (Finalrun) und Auswahl der leistungsfähigsten Prognosemodelle je Geometrieelement

Die Optimierungsschritte werden in der `configTable_optSteps` definiert. Durch die Auflistung der durchzuführenden Schritte in einer externen Konfigurationsdatei ist es möglich, diese bei Bedarf flexibel anzupassen. Für jeden Optimierungsschritt liefert die zugehörige `configTable_«optStep»_Runs` die Parameterwerte für jeden Run der MLPK innerhalb dieses `optSteps`. Dementsprechend wird die MLPK in einer individuell angepassten Konfiguration gemäß der Anzahl an konfigurierten Runs durchlaufen. Jeder Durchlauf liefert am Ende die Prognoseergebnisse für jedes Modell in Form der vorgestellten Metriken aus Kapitel 5.3.5, berechnet auf dem Validierungsdatensatz. Dabei entsteht ein tabellenförmiger `metricsReport` je Geometrieelement (vgl. Kapitel 5.3.6).

Nach Beendigung jeglicher Runs im Rahmen eines `optSteps` folgt die Bewertung der Resultate zur Identifikation der erfolgreichsten Parameterkombination. Dafür werden die entstandenen `metricsReports` eingelesen und die Einträge gemäß der in Kapitel 6.3.3 beschriebenen Bewertungsmethode durch Zuordnung von Rängen auf Grundlage der jeweiligen Prognosegüte in eine Reihenfolge gebracht. Dies ermöglicht die Identifikation derjenigen Parameterwerte, welche zu Modellen mit den höchsten Prognosegenauigkeiten führen. Zum Abschluss einer Iteration erfolgt die Neuparametrierung der MLPK mit den ermittelten Werten für den nächsten Optimierungsschritt. Dieser Prozess wird so lange durchlaufen, bis alle `optSteps` abgeschlossen sind und der finale Durchlauf die optimiert trainierten Modelle je Geometrieelement ausgibt.

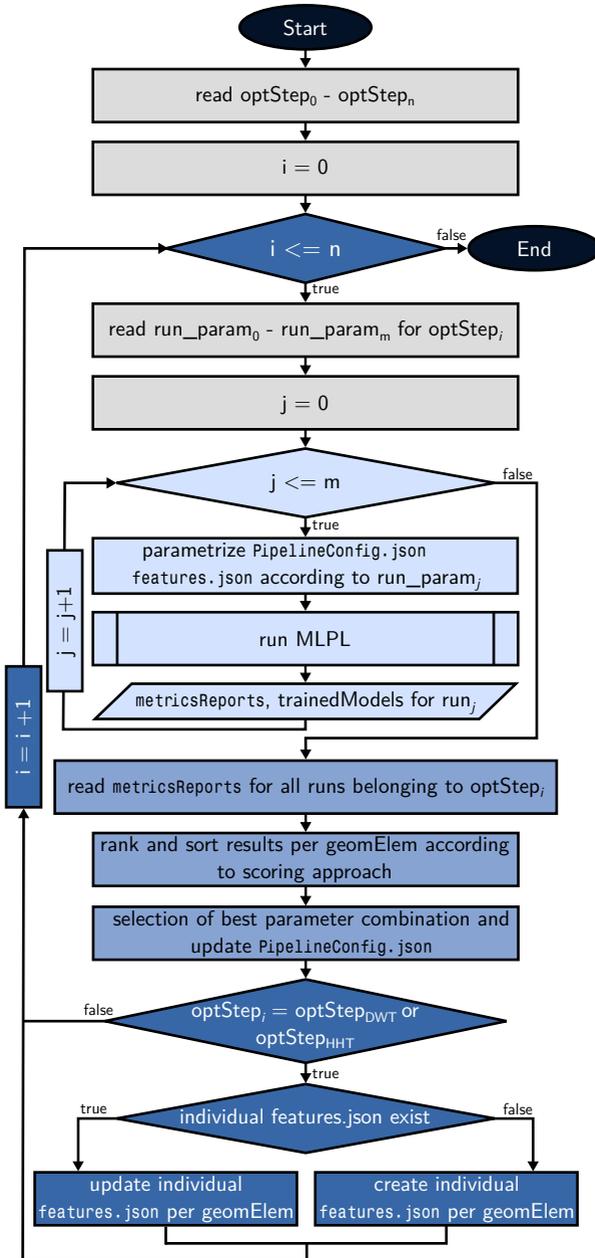


Abbildung 6.10: Algorithmus zur automatisierten Parametrierung der MLPK (engl.: MLPL), angelehnt an [FERT22c].

### 6.3.2 Optimierung der Teilprozesse

Der grundlegende Ablauf zur Optimierung sieht vor, für jeden Teilprozess der MLPK schrittweise an die Prognoseaufgabe angepasste Werte für die jeweiligen Konfigurationsparameter auszuwählen. Wie bereits erwähnt, handelt es sich bei DWT und HHT um Transformationsmethoden aus der Zeit-Frequenzdomäne, welche für eine adäquate Informationsbereitstellung mittels geeigneter Wahl der jeweiligen Hyperparameter, an die zu untersuchenden Daten adaptiert werden müssen. Zu diesem Zweck sieht der `optStepDWT` Runs unter veränderten Basis-Wavelets und Dekompositionsebenen vor. Diese beziehen die häufig verwendete Daubechies (db) Wavelet-Familie zusammen mit Coiflet- (coif), Symlet- (sym) und Biorthogonal- (bior) Wavelets in jeweils neun Ausprägungen (ausgenommen sym mit 8 Ausprägungen, da sym1 nicht existiert) und den Dekompositionsebenen 3 - 7 mit ein. [KARA13; PAHU20; SHAO11] Bei Durchführung dieses `optSteps` wird in der Konfiguration der `temporal-spectral` Domäne die Nutzung der HHT deaktiviert. Aufgrund der unterschiedlich resultierenden Verläufe der erfassten NC-Signaldaten, bedingt durch den charakteristischen Prozessablauf, werden für jedes Geometrieelement spezifische DWT-Hyperparameter gewählt, welche zu einer individuellen Parametrierung in den zugehörig erstellten `features.json` führen. Zusätzlich konnte im Rahmen der Masterthesis von PREIS gezeigt werden, dass die Individualisierung der Hyperparametrierung zu höheren Prognosegenauigkeiten führt [PREI22]. Parallel dazu wird die Anzahl an zu extrahierenden IMF ( $n_{\text{IMF}} = \{1, 2, \dots, 7\}$ ) im Rahmen von `optStepHHT` individualisiert an die Signalverläufe jedes Geometrieelements angepasst. Die Ergebnisse der ersten beiden Optimierungsschritte fließen in die darauffolgende Entscheidung mit ein, ob mittels HHT extrahierte Merkmale, zusätzlich zur DWT in den weiteren Schritten mit berücksichtigt werden sollen. Vorversuche von PREIS haben gezeigt, dass die DWT als Extraktionsmethode deutlich gegenüber der HHT überlegen ist. Zusätzlich erzielte die Nutzung beider Methoden keine Verbesserung im Vergleich zur alleinigen Nutzung der DWT. [PREI22] Dennoch ist es nicht auszuschließen, dass bei einer veränderten Datenbasis die HHT zusätzliche mehrwertbringende Informationen liefert, weshalb der Optimierungsschritt für die Domäne `temporal-spectral` in die Handlungsvorschrift mit einbezogen wird. Der letzte Schritt für die Optimierung der Merkmalsextraktion dient schließlich der Auswahl, welche Domänen in die Modellbildung mit einfließen sollen. Dafür erfolgt die Iteration über jede der vier zur Verfügung stehenden Domänen einzeln sowie über alle gemeinsam. Zusätzlich prüft der `optStep` die Kombination von `temporal-spectral` mit jeweils einer weiteren Domäne. In der Spektralanalyse kommen zur Ver-

besserung der Analysequalität häufig die Fensterfunktionen Hanning, Hamming und Blackman zum Einsatz [PODD14]. Diese werden in den Runs mit `spectral` Merkmalen zusätzlich iteriert, wodurch insgesamt 14 Parameterkombination in diesem `optStep` resultieren.

Die Optimierung der Parametrierung für die Merkmalsselektion basiert auf den möglichen Kombinationen der vier implementierten Selektionsalgorithmen (vgl. Kapitel 5.3.3).  $SW_{fs,prop}$  wird entsprechend der Anzahl der pro run verwendeten Algorithmen adaptiert. Bei Nutzung eines Algorithmus folgt demnach  $SW_{fs,prop} = 1$ . Ab zwei bis vier genutzten Algorithmen iteriert  $SW_{fs,prop}$  zwischen den Werten 2 und 4. Zur Berücksichtigung der zwei Skalierungsmethoden (vgl. Kapitel 5.3.6) erfolgt für jede der 21 Kombinationen jeweils ein Durchlauf unter Standardisierung bzw. Normalisierung der Merkmalswerte, sodass insgesamt 42 Durchläufe für die Optimierung der Merkmalsselektion benötigt werden. Der Finalrun durchläuft zur Erstellung und Auswahl der finalen Prognosemodelle die MLPK, unter Nutzung der durch den Algorithmus identifizierten optimierten Parameterkonfiguration.

Der modulare Aufbau und die Parametrierung über externe Konfigurationsdateien ermöglicht eine flexible Erweiterung der Schritte des Algorithmus. Die einfachen Anpassungsmöglichkeiten der Konfiguration gewährleistet eine breite Anwendbarkeit sowie bei Bedarf die Vergrößerung des Suchbereichs für die Parameterwerte.

### 6.3.3 Bewertungsmethode

Nach dem Durchlauf sämtlicher Iterationen innerhalb eines Optimierungsschritts ist es für die Identifikation der Parameterwerte, welche zu den höchsten Prognosegenauigkeiten führen erforderlich, die Ergebnisse der durchgeführten Runs mittels der erzeugten `metricsReports` auszuwerten. Diese enthalten die in Kapitel 5.3.5 vorgestellten Klassifikationsmetriken, ermittelt auf dem Validierungsdatensatz. Für die Analyse werden die Ergebnisse temporär zu einer Gesamttabelle mit

$$N_{res,optStep} = N_{geomElems} \times N_{MLAlgorithmen} \times N_{Runs} \quad (6.4)$$

Zeilen zusammengeführt, wobei  $N_{geomElems}$  der Anzahl an Geometrieelementen,  $N_{MLAlgorithmen}$  der Anzahl an konfigurierten ML-Algorithmen und  $N_{Runs}$  der Anzahl an MLPK Durchläufen innerhalb des aktuellen `optSteps` entspricht. Um je Geometrieelement das Modell zu bestimmen, welches die beste Prognoseleistung erbringt, werden jedem Modell je Geometrieelement entsprechend dessen Anzahl an FP-Vorhersagen sowie dessen Werte für SPEC,

ACC und ROC AUC Ränge zugeordnet. Ein niedrigerer Wert für die Anzahl an FP sowie höhere Werte der restlichen Metriken führen zu einem niedrigeren Rang und damit zu einer bevorzugten Auswahl als leistungsfähigstes Modell eines Geometrieelements. Erreichen mehrere Modelle die gleichen Werte, so wird jeweils ein Zwischenrang gemäß der zufälligen Reihenfolge innerhalb der Ergebnistabelle bestimmt. Jedes dieser Modelle erhält den durch Mittelwertbildung der zuvor ermittelten Zwischenränge gebildeten Rang. Dies gewährleistet eine Bewertung gemäß der jeweiligen Leistungsfähigkeit und eliminiert den Einfluss der Reihenfolge während der Rangzuordnung. Für die Modellauswahl erfolgt je Geometrieelement eine aufsteigende Sortierung der Ränge schrittweise nach dem Rang der FP und die Summe aller Ränge über sämtliche Metriken hinweg. Ist diese Sortierung aufgrund von gleichen Werten nicht ausreichend, berücksichtigt die Sortierungsstrategie zusätzlich nacheinander den Rang des ROC AUC, den Rang der SPEC sowie den Rang der ACC. Die höchste Gewichtung des Rangs für FP begründet sich mit deren besonders kritischen Einstufung im Rahmen eines potenziell produktiven Einsatzes der Modelle (vgl. Kapitel 5.3.5).

Diese individuelle Bewertung erlaubt es im Rahmen der Optimierungsschritte `optStepDWT` und `optStepHHT` die Hyperparameter, für die Transformationsmethoden aus dem Zeit-Frequenzbereich individuell für jedes Geometrieelement zu bestimmen und dementsprechend in die jeweilige Konfigurationsdatei (`features.json`) zu schreiben. Dazu wird das gemäß Sortierung leistungsfähigste Modell identifiziert und die zugrundeliegende Parametrierung gewählt. Innerhalb der restlichen Optimierungsschritte erfolgt die Bewertung global basierend auf der Leistungsfähigkeit über sämtliche Geometrieelemente hinweg. Dazu wird die Gesamtzahl an FP der gemäß Sortierung identifizierten Modelle je Geometrieelement für jeden Run innerhalb des Optimierungsschritts berechnet und schließlich diejenige Parametereinstellung gewählt, welche zur geringsten Anzahl an FP führt. Die resultierenden Parameterwerte dienen schließlich zur Aktualisierung der globalen `PipelineConfig.json`.

Die entwickelte, auf den Anwendungsfall der Qualitätsvorhersage adaptierte, domänenspezifische Bewertungsmethode ermöglicht die automatisierte Identifikation der leistungsfähigsten Prognosemodelle je Geometrieelement innerhalb eines Optimierungsschritts. Die zugrundeliegenden konfigurierten Parameterwerte werden für die Parametrierung der entsprechenden Konfigurationsdateien genutzt, um diese im darauffolgenden Optimierungsschritt für optimierte Durchläufe der MLPK zu berücksichtigen.

### 6.3.4 Validierung des Optimierungsalgorithmus

Für die Validierung wird der entwickelte Optimierungsalgorithmus für die automatisierte Erstellung der Prognosemodelle auf den generierten Datensätzen aus Kapitel 5.4 angewendet. Für die Gewährleistung der Vergleichbarkeit erfolgt die Wahl des gleichen Verhältnisses zwischen Trainings- und Testdatensatz in Anlehnung an Kapitel 6.2 mit 75 % an Trainingsdaten. Aufgrund der geringen zur Verfügung stehenden Datenmenge je Datensatz und den teilweise in geringer Anzahl enthaltenen n. i. O.-Beispielen sind die Voraussetzungen nicht gegeben, um die einzelnen Datensätze weiter zu unterteilen. Dies hätte eine geringere Trainingsdatenmenge (mit wiederum weniger n. i. O.-Instanzen) zur Folge, wodurch sich die datenbasierte Modellbildung aufgrund der komplexen Wirkzusammenhänge zwischen den eingangsseitigen NC-Signaldaten und den ausgangsseitigen Qualitätsdaten zusätzlich erschwert. Aus diesen Gründen entsprechen in diesem Fall die für eine Anwendung des Optimierungsalgorithmus notwendigen Validierungsdatsätze den Testdatensätzen.

Abbildung 6.11 fasst die Prognosegenauigkeiten nach Durchführung des Optimierungsalgorithmus zusammen. Die dargestellten Punkte bilden angelehnt an Abbildung 6.9 die erreichten Metriken auf dem Testdatensatz der jeweils 135 resultierenden Modelle nach dem Finalrun ab. Zusätzlich zu den Mittelwerten über sämtliche Modelle sind entsprechend die zuvor resultierenden Mittelwerte aus der nicht optimierten MLPK (vgl. Kapitel 6.2) mit entsprechend heller eingefärbten Balken eingetragen. Die abgebildeten orangefarbenen Pfeile heben die deutlichen Verbesserungen infolge der Optimierung hervor. Insbesondere SPEC und ROC AUC zeigen deutliche Verbesserungen in jedem der drei Datensätze. Je nach Datensatz fällt in der Gesamtübersicht die Wirkung der Optimierung auf die restlichen Metriken unterschiedlich stark aus. Für eine detaillierte Beschreibung fasst Tabelle 6.2 die prozentuale Verbesserung der ermittelten Mittelwerte in Relation zu den Werten der nicht optimierten MLPK zusammen. Ausgenommen der beiden marginalen Verschlechterungen für den durchschnittlichen REC auf  $DS_{DMG1}$  und  $DS_{DMG1}$  verbessern sich sämtliche Durchschnittsmetriken über alle Modelle und Geometrielemente hinweg. Die deutlichsten Verbesserungen für SPEC und ROC AUC spiegeln das Optimierungsziel der Minimierung von FP wider, da aufgrund der zugrundeliegenden Klassenverteilungen insbesondere die SPEC starkem Einfluss von FP unterliegt.

Die Optimierungsergebnisse bringen für jeden der drei betrachteten Datensätze hervor, dass für die finale Prognosemodellerstellung lediglich berechnete

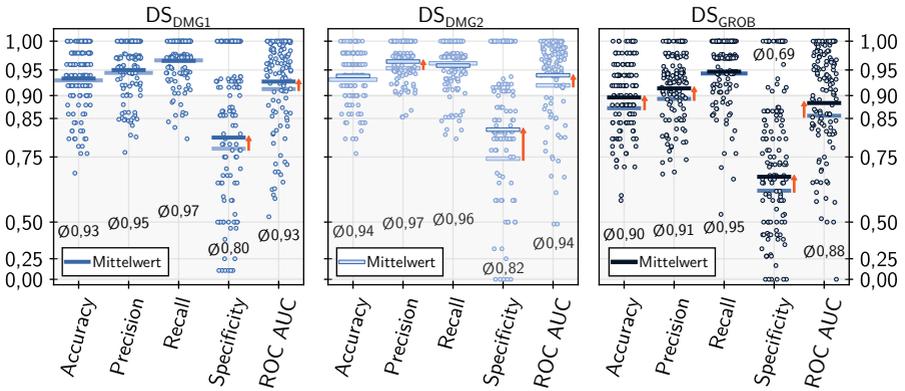


Abbildung 6.11: Prognosegenauigkeiten nach Durchführung des Optimierungsalgorithmus.

Tabelle 6.2: Prozentuale Verbesserung der durchschnittlichen Metriken nach Durchführung des Optimierungsalgorithmus.

DS	durchschnittliche Verbesserung				
	ACC	PREC	REC	SPEC	ROC AUC
DS <sub>DMG1</sub>	0,38 %	0,55 %	-0,09 %	3,81 %	1,62 %
DS <sub>DMG2</sub>	0,75 %	1,25 %	-0,45 %	10,51 %	2,12 %
DS <sub>GROB</sub>	2,63 %	2,40 %	0,35 %	7,74 %	3,30 %

Merkmale auf den mittels DWT transformierten Zeitreihen Berücksichtigung finden. Eine zusätzliche Nutzung von Merkmalen aus den anderen Domänen oder durch HHT ermittelte Merkmale führt im untersuchten Szenario nicht zu verbesserten Prognosegenauigkeiten der Modelle. Die an die jeweiligen Datenverläufe der Geometrielemente individuell adaptierte DWT stellt sich in diesem Anwendungsfall als leistungsfähigste Merkmalsextraktionstrategie dar. Auffällig ist dabei, dass sich die final identifizierten Basis-Wavelets und Dekompositionsebenen je Geometrielement zwischen den Datensätzen unterscheiden. Eine tabellarische Übersicht dieser Ergebnisse ist in Anhang A.10 dargestellt. Dies lässt die Schlussfolgerung zu, dass es trotz der gleichen Prozessführung zu Unterschieden in den Daten zwischen den Datensätzen und WZM kommt. Gründe dafür können neben der unterschiedlichen Bauweise der Maschinen, zusätzlich die Unsicherheiten während der Datenaufzeichnung

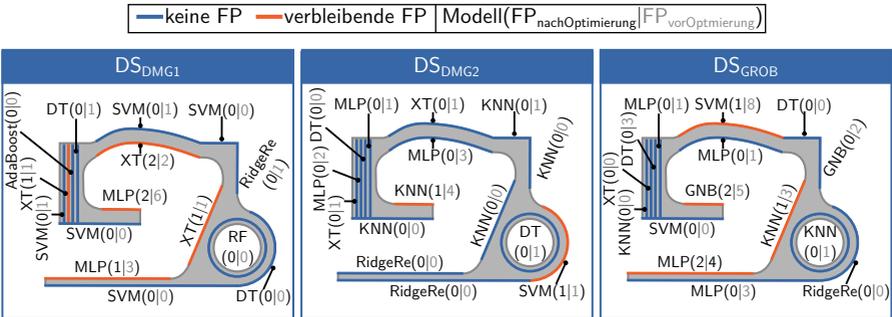


Abbildung 6.12: Resultierende Modelle und FP-Vorhersagen je Geometrieelement.

der Datensätze  $DS_{DMG1}$  und  $DS_{GROB}$  (vgl. Kapitel 6.1.1) sein. Die fehlenden Datenpunkte wirken sich negativ auf die Stabilität der Transformation in den Frequenz- und Zeit-Frequenzbereich, aufgrund des damit verbundenen Informationsverlusts, aus.

Die diskutierten Ergebnisse liefern dennoch lediglich einen Gesamteindruck über die Leistungsfähigkeit des Optimierungsalgorithmus. Bei genauerer Betrachtung von Abbildung 6.11 lassen sich zahlreiche Fälle identifizieren, in welchen die Modelle sehr niedrige Werte für die Metriken, insbesondere für die SPEC, aufweisen. Deshalb ist es notwendig, aus der Gesamtmenge von trainierten Modellen das leistungsfähigste Modell für jedes Geometrieelement im Kontext der Qualitätsprognose zu ermitteln. Die vorgestellte Bewertungsmethode aus Kapitel 6.3.3 liefert die in Abbildung 6.12 gezeigten Modelle je Geometrieelement samt FP auf dem Testdatensatz. Der Optimierungsalgorithmus reduziert die FP-Anzahl verglichen mit den Modellen aus der nicht optimierten MLPK deutlich. Die 7 FP auf  $DS_{DMG1}$  verteilen sich auf 5, die 2 FP auf  $DS_{DMG2}$  auf 2 Geometrieelemente. Die Modelle für  $DS_{GROB}$  präzisieren 6 FP auf dem Testdatensatz. Die deutliche Verbesserung der FP auf allen Datensätzen zeigt das Potenzial der entwickelten Methode zur automatisierten Modellerstellung und anschließenden Auswahl. Tabelle 6.3 fasst die Ergebnisse zu den restlichen, während der Optimierung berücksichtigten, Metriken zusammen. Gezeigt ist jeweils der prozentual angegebene Durchschnitt der erreichten Prognosegüte auf den Modellen unbekanntem Testdaten, unter der Nutzung der final ausgewählten Modelle über alle Geometrieelemente hinweg. Zum Großteil lassen sich Werte von über 95 % beobachten.

Dies spiegelt die hohe Prognoseleistung der Modelle wider und bestätigt, dass die entwickelte Methode für die Optimierung der MLPK in der Lage ist, die Anforderung einer automatisierten Bildung von optimierten Modellen mit entsprechend hoher Prognosegüte für die Qualitätsprädiktion zu erfüllen und den enormen Aufwand einer manuellen Datenanalyse und -modellierung zu reduzieren bzw. zu ersetzen. Dies beantwortet zusammen mit den weiteren Ergebnissen aus Kapitel 6 insbesondere die Teilaspekte hinsichtlich automatisierter Entwicklung und Auswahl optimierter Prognosemodelle aus Forschungsfrage FF4.

Tabelle 6.3: Durchschnittswerte der Metriken über alle Geometrielemente für die finalen Prognosemodelle.

DS	Durchschnittswert				
	ACC	PREC	SPEC	ROC AUC	FP/geomELEM
DS <sub>DMG1</sub>	97,47 %	98,77 %	94,79 %	97,32 %	0,47
DS <sub>DMG2</sub>	98,00 %	99,66 %	98,56 %	98,85 %	0,13
DS <sub>GROB</sub>	92,00 %	98,49 %	96,74 %	94,92 %	0,40



# 7 Validierung und Realisierung

In diesem Kapitel wird die Validierung des erarbeiteten Gesamtkonzepts beschrieben. Für die Beurteilung der Übertragbarkeit der entwickelten Lösungen ausgehend von der Zielstellung dieser Arbeit folgt die Anwendung der entwickelten Lösungen auf ein weiteres Werkstück in Kapitel 7.1. Der implementierte Softwareprototyp im zweiten Abschnitt demonstriert, wie die erstellten Prognosemodelle dazu genutzt werden können, um diese verbunden mit der vierten Forschungsfragestellung in einem prototypisch umgesetzten prozessnahen Überwachungssystem für die In-Prozess-Qualitätsbewertung anzuwenden. Abschließend folgt im letzten Abschnitt die Ergebnisdiskussion inklusive Fazit.

## 7.1 Validierung des Gesamtkonzepts anhand des Werkstücks „Tiefe Tasche“

Mit der Übertragung auf das Werkstück „Tiefe Tasche“ wird die grundsätzliche Übertragbarkeit der entwickelten Lösungen auf ähnliche Zerspanprozesse geprüft. Bei der in Abbildung 7.1 dargestellten Geometrie handelt es sich um ein in seiner Komplexität reduziertes, für die Lehre am PTW eingesetztes Strukturbauteil aus der Luft- und Raumfahrtbranche, bestehend aus sieben qualitätsrelevanten Geometrieelementen. Die spezifizierten Toleranzen sind in Anhang A.11 aufgeführt. Die Fertigung findet auf der DMC 850V unter Verwendung von Rohmaterialblöcken aus dem Vergütungsstahl 42CrMo4 (1.7225) statt. Die tiefe Tasche wird zur aufwandsreduzierten Generierung einer Datenbasis gitterförmig (7x7 Werkstücke je Block) angeordnet. Nach dem Schruppen folgt das Endkonturschlichten mit Werkzeugen der Hoffmann Group [HOFF22] und den aus Abbildung 7.1 oben rechts hervorgehenden Technologieparametern. Insgesamt werden 392 Werkstücke verteilt auf 8 Materialblöcken unter verschiedenen Fehlereinflüssen angelehnt an Tabelle 5.4 hergestellt, wodurch die in Abbildung 7.1 unten dargestellte Verteilung von i. O.- und n. i. O.-Geometrieelementen resultiert. Anschließend erfolgt die Vermessung der Bauteile auf dem KMG und die Erstellung der Messprotokolle gemäß Kapitel 4.2.4. Durch Zusammenführung der Daten aus der Fertigung und Qualitätssicherung entsteht der Datensatz  $DS_{DMG3}$ .

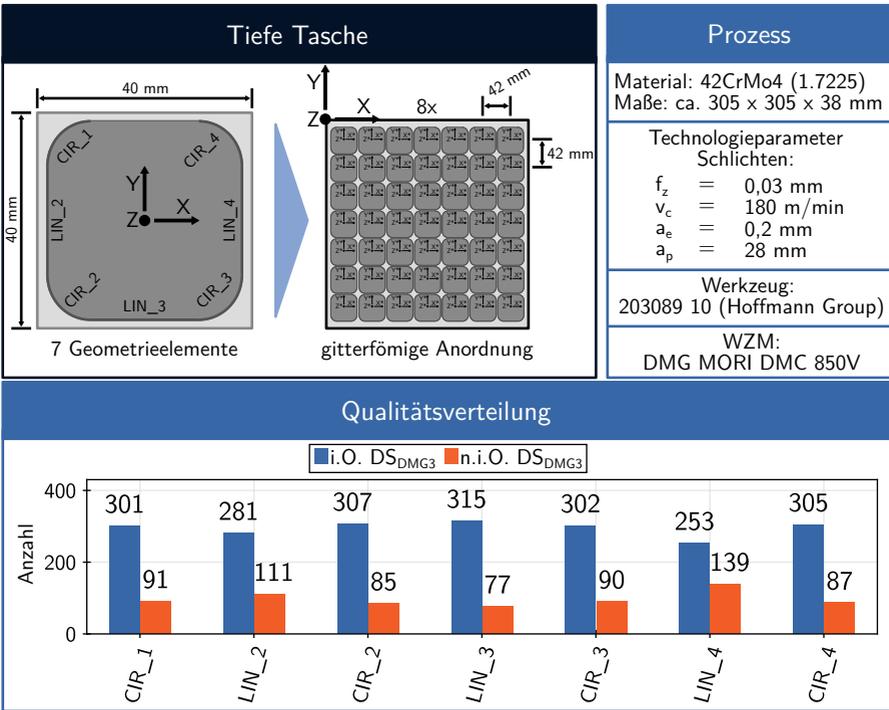


Abbildung 7.1: Werkstück „Tiefe Tasche“ – Geometrie, Prozess und Qualitätsverteilung.

Die während des Prozesses eingebrachten Fehlereinflüsse mit dem zugrundeliegenden Versuchsplan führen auch im zweiten Anwendungsfall zu einer Datenbasis mit Vertretern beider Zielklassen, sodass eine Bildung von Klassifikationsmodellen für die binäre Qualitätsprognose möglich ist.

Die Daten aus der NC werden wie in Kapitel 5.1 kontextualisiert aufgezeichnet und anschließend mittels des entwickelten Segmentierungsalgorithmus aus Kapitel 5.2 je Geometrieelement segmentiert. Bei dem Vergleich zwischen den Geometrieelementen vom gleichen Grundtyp in Abbildung 7.2 links mit beispielhaft dargestelltem Verlauf des Antriebsstroms aus der Y-Achse der Maschine ist zu erkennen, dass die segmentierten Zeitreihen die gleiche Länge aufweisen. Die symmetrische Konstruktion des Werkstücks impliziert bei der gewählten Prozessführung unter konstanter Vorschubgeschwindigkeit für Geometrieelemente, welche durch TCP-Pfadabschnitte gleicher Länge gefe-

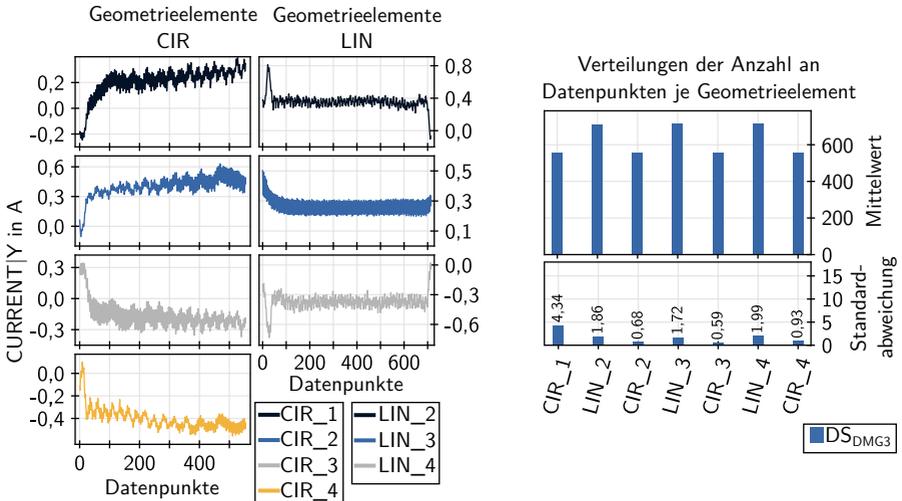


Abbildung 7.2: Ergebnisse Segmentierung und Datenqualität für den Anwendungsfall „Tiefe Tasche“.

tigt werden, die gleiche Dauer des zugehörigen Prozessabschnitts und damit die übereinstimmende Länge der individuellen Zeitreihendaten. Mit dem Segmentierungsalgorithmus ist es bei kontextsensitiver Datenerfassung demnach für ein alternatives Bauteil möglich, automatisiert die hochfrequenten Zeitreihendaten aus der NC den entsprechenden Geometrielementen zuzuordnen und unnötige Datenpunkte zu entfernen. Abbildung 7.2 rechts spiegelt die Präzision der Segmentierung über den gesamten Datensatz hinweg wider. Ausgewertet auf 378 Werkstücken (14 Werkstücke unterliegen Fehlereinflüssen mit erhöhter Schnittgeschwindigkeit und werden deshalb nicht in die Auswertung mit einbezogen) sind die Mittelwerte der Anzahl an Datenpunkten über alle Linienelemente annähernd konstant. Gleiches gilt für die Viertelkreiselemente. Zusätzlich weist die Standardabweichung Werte im niedrigen einstelligen Bereich auf. Die relativ zu den restlichen Geometrielementen erhöhte Standardabweichung in CIR\_1 lässt sich durch eine detaillierte Fehlstellenanalyse in den erfassten Rohsignalen erklären. Ähnlich zu den Ergebnissen auf DS<sub>DMG1</sub> (vgl. Kapitel 6.1.1) sind infolge der Schwachstellen bei der Datenerfassung während des Versuchsdurchlaufs, Fehlstellen in Höhe von 2301 Abtastzeitpunkten (0,140 %) in DS<sub>DMG3</sub> aufgetreten. In CIR\_1 lässt sich die größte Fehlstelle in Höhe von 81 fehlenden Abtastzeitpunkten beobachten, welche zum relativ erhöhten Wert in der Standardabweichung führt.

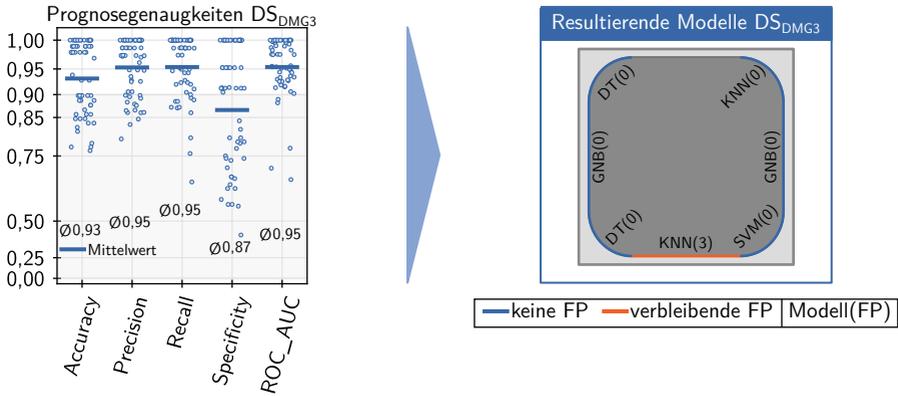


Abbildung 7.3: Prognosegenauigkeiten und resultierende Modelle für den Anwendungsfall „Tiefe Tasche“.

Die Fertigung erfolgte von Ende September 2020 bis Anfang Oktober 2020, zeitlich gesehen zwischen der Herstellung der Werkstücke aus  $DS_{DMG1}$  und  $DS_{GROB}$  und unterlag demnach den beschriebenen Fehlern in der Datenerfassung (vgl. Kapitel 6.1.1).

Im letzten Schritt der Validierung wird die Prognosemodellerstellung basierend auf dem entwickelten Algorithmus zur automatisierten Prozesskettenparametrierung aus Kapitel 6.3 durchgeführt. Dazu ist es lediglich notwendig Ordnerpfade in der `PipelineConfig.json` (vgl. Kapitel 5.3.6) anzupassen. Die restlichen Konfigurationseinstellungen werden für den Durchlauf des Algorithmus identisch zu den vorigen gewählt. Abbildung 7.3 fasst die resultierenden Ergebnisse zusammen. Die Mittelwerte der Prognosegenauigkeiten für alle Modelle (Abbildung 7.3 links) liegen in ähnlichen Wertebereichen wie bei der vorigen Anwendung auf den Datensätzen zum Musterwerkstück. Bei Analyse der vom Optimierungsalgorithmus final erstellten und ausgewählten Modelle (Abbildung 7.3 rechts) ist zu beobachten, dass diese auf dem Testdatensatz lediglich drei FP prognostizieren, welche allein im Geometrielement `LIN_3` auftreten. Insgesamt erreichen die resultierenden Modelle für das Werkstück „Tiefe Tasche“ die in Tabelle 7.1 zusammengefassten Prognosegenauigkeiten. Die hohen Werte für die Metriken spiegeln die hohe Prognoseleistung der Modelle auf dem unbekanntem Testdatensatz und damit die Effektivität des Optimierungsalgorithmus für den Einsatz in einem alternativen Anwendungsfall wider.

Tabelle 7.1: Durchschnittswerte der Metriken nach Durchlauf des Optimierungsalgorithmus für die „Tiefe Tasche“.

DS	Durchschnittswert				
	ACC	PREC	SPEC	ROC AUC	FP/geomELEM
DS <sub>DMG3</sub>	93,73 %	99,45 %	97,74 %	97,12 %	0,43

## 7.2 Implementierung eines Softwareprototyps

Zur beispielhaften Umsetzung der prozessparallelen Qualitätsprognose wird ein Softwareprototyp in der Programmiersprache Python implementiert. Für die Gestaltung der Anwenderoberfläche kommt das Framework Dash<sup>1</sup> zum Einsatz, welches eine schnelle und einfache Möglichkeit bietet, datenbasierte Web-Applikationen (Apps) zu erstellen. Die Programmierung der Funktionsbausteine im Backend erfolgt unter der Verwendung des Web-Frameworks Django<sup>2</sup>. Abbildung 7.4 zeigt die Benutzeroberfläche des entwickelten Softwareprototyps.

Im Zentrum der Anwenderoberfläche steht eine interaktive 3D-Ansicht des Bauteils für die visuelle Darstellung der modellbasierten Prognoseergebnisse. Diese visualisiert die 3D-Geometrie des gesamten Werkstücks inklusive der eingefärbten Flächen zu den qualitätsrelevanten Geometrieelementen. Die Flächen werden entsprechend der Modellvorhersagen Blau für i. O. bzw. Orange für n. i. O. eingefärbt. Für die 3D-Ansicht des Werkstücks dient das Wavefont Object Format (OBJ), welches mit der Dateiendung `.obj` vorliegt und häufig zur programm- und plattformübergreifenden Übermittlung von 3D-Modellen genutzt wird [HECK14]. Die Realisierung der Darstellung erfolgt mit der Funktion `Mesh3d()` aus dem Modul `graph_objects` der Software-Bibliothek Plotly<sup>3</sup>. Initial ist es notwendig, für jede innerhalb der Anwendung zur Verfügung stehenden Materialnummer, die 3D-Daten des gesamten Bauteils für die Gesamtdarstellung, sowie für den Zugriff auf die qualitätsrelevanten Flächen, die 3D-Daten der jeweiligen Geometrieelemente in Form von OBJ Dateien bereitzustellen. Die graphische Ansicht basiert auf der Darstellung des Gesamtbauteils mit überlagerten Plots der Geometrieelemente, wodurch der Zugriff auf die Geometriedaten der einzelnen Flächen ermöglicht wird.

<sup>1</sup><https://dash.plotly.com/introduction>

<sup>2</sup><https://www.djangoproject.com/>

<sup>3</sup><https://plotly.com/graphing-libraries/>

**Baureihe**  
Auswahl Materialnummer  
TYP10001

**KI-Modellübersicht**  
Performanz Metriken

geomELEMENT	ALOP	ACC	SPEC
INNERCIRCLE	DT	0.98	1
INNERSTEP	MIP	1	1
LINESTEP_2	DT	0.98	1
LINESTEP_4	DT	0.98	1
LINESTEP_6	MIP	1	1
LINESTEP_8	MIP	1	1
LINE_1	Ridgefe	1	1
LINE_2	KNN	0.96	1
LINE_3	KNN	0.92	0.875
LINE_4	KNN	1	1
LINE_5	KNN	1	1
LINE_6	KNN	1	1
LINE_7	Ridgefe	1	1

**Export - Qualitätsreport**  
Export Bauteillansicht  
png  
Export Vorhersage  
pdf csv

**Bauteillansicht**  
TYP10001  
INNERCIRCLE  
INNERSTEP  
LINESTEP\_2  
LINESTEP\_4  
LINESTEP\_6  
LINESTEP\_8  
LINE\_1  
LINE\_2  
LINE\_3  
LINE\_4  
LINE\_5  
LINE\_6  
LINE\_7  
OUTERCIRCLE  
OUTERSTEP

**Live-Anbindung**  
live

Seriennummer	Produktionsdatum
1458	15:55 23.03.2022 CET
1460	18:16 23.03.2022 CET
1461	18:17 23.03.2022 CET
1462	18:18 23.03.2022 CET
1464	18:21 23.03.2022 CET
1466	18:23 23.03.2022 CET
1467	18:24 23.03.2022 CET
1468	18:26 23.03.2022 CET
1469	18:27 23.03.2022 CET
1470	18:28 23.03.2022 CET
1472	08:48 24.03.2022 CET
1473	08:49 24.03.2022 CET
1475	08:52 24.03.2022 CET
1476	08:53 24.03.2022 CET
1478	08:56 24.03.2022 CET

**Konsole**  
start app  
no Materialnummer selected (estimated time: 4.7s)  
loading Materialnummer: TYP10001 ... (estimated time: 1.8s)  
predicting Seriennummer: 1475 ... (estimated time: 1.8s)

selected ID ID-seen nID nID-seen

Abbildung 7.4: Browserbasierte Benutzeroberfläche des Softwareprototyps.

Die Tabelle auf der rechten Seite zeigt die Historie bereits gefertigter Werkstücke aufgeschlüsselt nach Seriennummer und Produktionsdatum zur ausgewählten Materialnummer. Die Farben der Zeilen signalisieren dem Anwender je Seriennummer die im Anschluss an die Fertigung prognostizierte Qualität, bezogen auf das gesamte Werkstück. Eine orangefarbene Einfärbung weist darauf hin, dass mindestens eines der Geometrieelemente durch die Modelle als n. i. O. vorhergesagt wurde und stellt somit einen Kurzüberblick über den historischen Verlauf der Qualität dar. Für eine detaillierte Analyse initiiert ein Klicken auf eine der Zeilen der rechten Spalte die Aktivierung der zuvor abgelegten Prognosemodelle und es findet die Prädiktion für alle Geometrieelemente statt. Dieser Vorgang benötigt je nach Komplexität des Bauteils wenige Sekunden. Die entsprechend der Modellvorhersagen eingefärbten Flächen ermöglichen dem Anwender eine graphische Auswertung und Bewertung von qualitätsbezogenen Problemen. Dadurch wird der Anwender in die Lage versetzt, rasche Entscheidungen für den weiteren Produktionsablauf zu treffen. Für die zusätzliche Entscheidungsunterstützung liefert die linke Tabelle die Prognoseleistungen der Modelle für jedes Geometrieelement auf dem genutzten Testdatensatz innerhalb der Modellerstellungsphase. Die Konsole dient der zusätzlichen Informationsbereitstellung während der Interaktion mit der App.

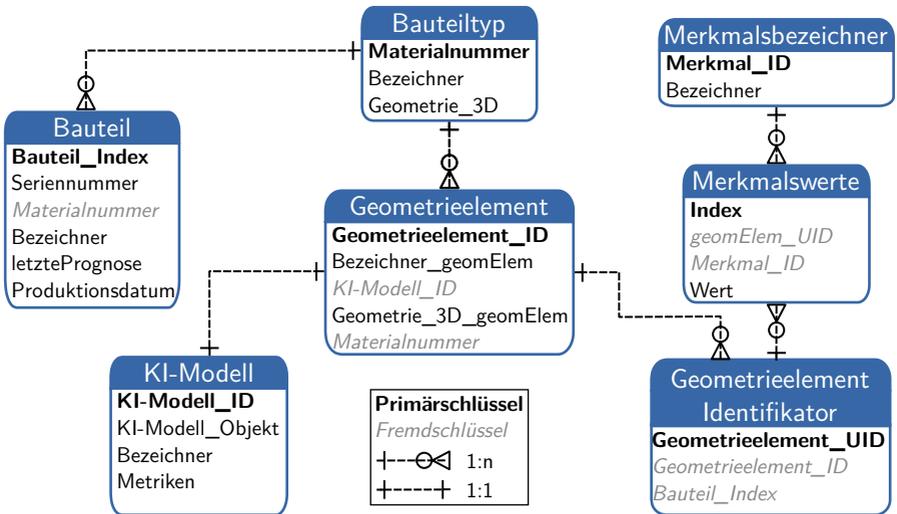


Abbildung 7.5: Entity-Relationship-Modell der Datenbank.

Im Backend dient eine PostgreSQL<sup>4</sup>-Datenbank, welche die Daten für die Webanwendung strukturiert bereitstellt. Das in Abbildung 7.5 gezeigte Entity-Relationship-Modell der Datenbank ermöglicht eine effiziente Datenhaltung sowie flexible Erweiterung der Anwendung für weitere Bauteile. Benötigte Daten sind demnach die 3D-Daten des Bauteils inklusive Geometrieelemente, die trainierten Prognosemodelle samt Metriken auf dem Testdatensatz der Modellerstellungsphase sowie die zugehörigen vorverarbeiteten Daten aus der WZM in Form von berechneten Merkmalen, welche die Eingangsgrößen für die Qualitätsprognose darstellen.

Für eine online Prognose unmittelbar nach der Fertigung ist es möglich, mittels Anbindung über einen ETL-Prozess (extract, transform, load) die in kontextualisierter Form gespeicherten Daten aus der Influx-Datenbank (vgl. Kapitel 4.2.2) abzufragen. Die Weiterverarbeitung der Daten gliedert sich in die Segmentierung (vgl. Kapitel 5.2) und Merkmalsextraktion (vgl. Kapitel 5.3.2) als Zwischenprozesse vor deren Übertragung in die PostgreSQL-Datenbank des Softwareprototyps. Bei erfolgreicher Vorverarbeitung und Übermittlung der Daten erfolgt schließlich die Qualitätsprognose innerhalb der App. Im Anhang A.12 sind zusätzliche Ausschnitte der Benutzeroberfläche dargestellt, die einen Einblick in die Funktionsweise des Softwareprototyps geben.

<sup>4</sup><https://www.postgresql.org/>

## 7.3 Ergebnisdiskussion

Die Ergebnisse aus den vorangegangenen Kapiteln bestätigen, dass eine In-Prozess-Qualitätsüberwachung für die spanende Fertigung mit WZM, basierend auf maschineninternen NC-Signaldaten möglich ist. Im Verlauf der Kapitel 4 bis 7 wurden die vier im Rahmen der Zielsetzung aus Kapitel 3 aufgestellten Forschungsfragen beantwortet.

Das vorgestellte Gesamtkonzept liefert, mittels durchgängiger Informationsbereitstellung aus den entscheidenden Wertschöpfungsschritten, die Grundlage für eine Erzeugung von hochqualitativen Datensätzen als Voraussetzung für die Erstellung von Qualitätsprognosemodellen. Ein hochqualitativer Datensatz zeichnet sich im Kontext der Qualitätsprädiktion durch folgende Eigenschaften aus:

- Präzise zugeordnete NC-Signaldatensegmente zu den entsprechenden Prozessabschnitten bzw. Geometrieelementen am Werkstück.
- Verknüpfung von Zielvariablen bzw. Labeln aus der Qualitätssicherung für jedes Zeitreihensegment.

Die Kontextualisierung aus Kapitel 5.1 zusammen mit dem in Kapitel 5.2 realisierten Segmentierungsalgorithmus erlauben die Anreicherung der erfassten rohen NC-Signale mit den notwendigen prozessbeschreibenden Metadaten, sodass für jedes Geometrieelement an einem Werkstück die während der Fertigung entstandenen Datenpunkte explizit zur Verfügung stehen. Unter Einhaltung der vorgeschlagenen Datenstruktur für Qualitätsdaten aus Kapitel 4.2.4 kann die Verknüpfung der gewonnenen Zeitreihensegmente mit den zugehörigen Zielvariablen aus der Qualitätssicherung eindeutig und automatisiert umgesetzt werden. Dadurch entsteht die Grundlage für die KI-basierte Prognosemodellerstellung in Form von hochqualitativen Datensätzen, unabhängig davon ob eine Klassifikation oder Regression angestrebt wird. Nachträgliche manuelle Aufwände für die Zusammenführung dieser Daten sowie für die Übertragung auf weitere Werkstücke entfallen hierbei. Dies bedeutet in der Anwendung, dass für die Erzeugung einer nutzbaren Datengrundlage lediglich Initialaufwände in der Datenstrukturierung und -speicherung notwendig werden.

Mittels der entwickelten MLPK (vgl. Kapitel 5.3) im Kern des Optimierungsalgorithmus aus Kapitel 6.3 lassen sich basierend auf den gewonnenen Datensätzen automatisiert leistungsfähige Prognosemodelle für die Qualitätsvorhersage erstellen. Diese ermöglichen im produktiven Einsatz die Klassifikation einzelner Geometrieelemente am Werkstück in i. O. und n. i. O. an-

hand von prozessparallel erfassten Daten aus der NC einer WZM. Für die Erstellung der Prognosemodelle analysiert die entwickelte Algorithmik die eingangsseitigen Daten, identifiziert für jedes qualitätsrelevante Geometrieelement die einflussreichsten Merkmale und trainiert prozessindividuelle Prognosemodelle. Dafür stehen 17 domänenspezifisch ausgewählte Signale (vgl. Kapitel 6.1.3) zur Verfügung, welche zuvor mittels der gezeigten Verfahren vorverarbeitet und zur Informationsbereitstellung auf prozessbeschreibende Kennwerte reduziert werden. Die Ergebnisse zeigen, dass sich die Analyse der Daten im Zeit-Frequenzbereich unter der Nutzung der diskreten Wavelet-Transformation maßgeblich eignen, um relevante Informationen aus den Signalen für die Qualitätsprognose zu extrahieren. Die Anwendung auf ein weiteres Bauteil bestätigt die Übertragbarkeit der zugrundeliegenden Lösungen. Manuelle Aufwände während der Datenvorbereitung und Modellerstellung entfallen und beschränken sich auf domänenspezifische Aufgaben für die Datengenerierung während Prozess und Qualitätssicherung. Die Anwendung der gewonnenen Prozesswissensmodelle in einem prototypisch umgesetzten Qualitätsüberwachungssystem veranschaulicht exemplarisch die Umsetzung einer In-Prozess-Qualitätsprognose in einem anwendungsnahen Demonstratorszenario. Zusammenfassend führen die entwickelten Lösungen zur Erfüllung des in Kapitel 3 definierten Forschungsziels: Die Erforschung und Entwicklung von übertragbaren Lösungen für die In-Prozess-Qualitätsprognose von spanend hergestellten Werkstücken, basierend auf prozessparallel erfassten Daten aus der NC einer WZM, mittels trainierter Algorithmen aus dem Bereich des ML. Die gewählte Implementierung der Lösungen schließt dabei jegliche Form- und Lagetoleranzen, welche direkt an einem zugeordneten Geometrieelement bestimmt werden mit ein. Die Berücksichtigung von Lagetoleranzen, welche durch Verknüpfung mehrerer Geometrielemente entstehen, erfolgt demnach nicht im Rahmen dieser Arbeit. Eine Erweiterung des Verfahrens hinsichtlich dieser Toleranzen ist möglich, jedoch gilt es dabei zu beachten, dass die damit verbundene Zusammenführung von NC-Signaldatensegmenten mehrerer Geometrielemente zu einer Erhöhung der Dimensionalität des Eingaberaums führt, wodurch sich die Komplexität der Modellbildung erhöht.

Die Ergebnisse auf den zugrundeliegenden Datensätzen zeigen, dass es trotz der hohen Prognoseleistung in diesem Fall nicht möglich ist, mit den erstellten Modellen Prognosegenauigkeiten von 100 % zu erreichen. Dabei werden die höchsten Prognoseleistungen bei individueller Modellauswahl je Geometrieelement erzielt. Dies lässt den Schluss zu, dass je nach gegebener Situation und Informationsdichte in den Daten eine angepasste Algorithmenauswahl für das Erzielen der höchsten Prognoseleistung erforderlich ist. Die Analysen der

Fehlvorhersagen deuten darauf hin, dass die Menge an Trainingsbeispielen zu gering ist, um die Bandbreite an möglichen qualitätsbeeinflussenden Störgrößen vollständig abzudecken. Zudem führen unzulässig große Veränderungen in den Eingangsdaten zu einer Verschlechterung der Prognosegüte, wodurch Anpassungen oder erneute Trainings der Modelle durchgeführt werden müssen. Die Unsicherheiten in der Datenaufzeichnung, die Veränderungen in den Umgebungsbedingungen der WZM sowie der voranschreitende Verschleiß der Maschinenkomponenten führten in den Signaldaten zu wesentlichen Veränderungen, welche sich in den abweichenden Ergebnissen zwischen  $DS_{DMG1}$  und  $DS_{DMG2}$  widerspiegeln. Hierfür ist die zukünftige Untersuchung von Forschungsfragestellungen zur Detektion von Veränderungen in den Daten hin zu einem produktiven Einsatz unabdingbar.

Das vorgestellte Verfahren liefert die Grundlage, um unter realen Produktionsbedingungen nutzbare Datensätze mit einer hohen Anzahl an Trainingsbeispielen zu generieren. Dabei liegt die größte Herausforderung in der Praxis, die notwendige Dateninfrastruktur möglichst fehlerfrei und über die Produktion hinweg umzusetzen. Zusätzlich ist es bei der Entwicklung einer KI-basierten Qualitätsprognose mit Verfahren aus dem Bereich des überwachten Lernens essentiell, für jedes Werkstück die zugehörigen Labels der adressierten Zielgröße zu ermitteln und fehlerfrei den Eingangsdaten zuzuordnen. Fehlen diese vereinzelt oder ist die verwendbare Datenbasis zu gering, besteht das Risiko, dass die Algorithmen nicht in der Lage sind die komplexen Zusammenhänge zwischen den NC-Signaldaten und der Werkstückqualität zu erlernen. Die Schwierigkeit entsteht in der Organisation, da insbesondere für die Erstellung der Zielwerte aus der Qualitätssicherung heraus, hohe initiale Mess- und Prüfaufwände zusätzlich zu den üblichen Prüfintervallen entstehen. Darüber hinaus benötigen Klassifikationsalgorithmen während des Trainings Instanzen aus jeder Zielklasse, demnach auch Werkstücke welche nicht die Toleranzanforderungen erfüllen. In realen Produktionsumgebungen sind Fehlteile selten bis kaum zu finden, wodurch die Gewinnung einer repräsentativen Datenbasis zusätzlich erschwert wird. Der entwickelte Versuchsplan aus Kapitel 5.4 berücksichtigt typische Fehlereinflüsse während der Zerspannung, sodass gezielt Fehlteile gefertigt werden und bietet eine Hilfestellung für die effiziente Generierung einer nutzbaren hochqualitativen Datenbasis, bestehend aus den zuvor angesprochenen hochqualitativen Datensätzen und ausreichend Trainingsinstanzen für jede Zielklasse (in diesem Fall für i. O. und n. i. O.). Der Mehraufwand für die bewusste Fertigung von Fehlteilen muss jedoch unter realen Bedingungen gegenüber dem potenziellen Nutzen der prozessparallelen Qualitätsüberwachung abgewogen werden.

# 8 Zusammenfassung und Ausblick

In diesem Kapitel werden die im Rahmen dieser Arbeit systematisch untersuchten Lösungsansätze und Ergebnisse zur automatisierten Erstellung von Prognosemodellen für die prozessparallele Qualitätsüberwachung zusammengefasst. Abschließend wird der hieraus ableitbare, weitere Forschungsbedarf in Form eines Ausblicks dargestellt.

## 8.1 Zusammenfassung

Im Rahmen dieser Dissertationsschrift wurde ein anwendungsnahe Lösungskonzept erforscht, welches erlaubt eine In-Prozess-Bewertung hinsichtlich der Qualität spanend gefertigter Werkstücke mittels Verfahren des ML umzusetzen.

Für die datenbasierte Qualitätsprognose stehen erfasste Signaldaten aus der NC im Vordergrund, welche aus den Regelkreisen der Vorschubachsen sowie der Spindel gewonnen werden. Die Grundlage des Verfahrens bildet eine fortschrittliche Vorgehensweise zur Kontextualisierung der Rohdaten, um hochwertige Analysen zu ermöglichen. Dazu fließen geometrische und fertigungsrelevante Informationen aus der Konstruktion über den CAM-Prozess in das NC-Teileprogramm ein. Durch die realisierte Verknüpfung dieser Daten mit den Signaldaten aus dem Lageregelkreis entstehen aufbereitete Datensätze zu jedem gefertigten Bauteil. Ein neuartiger Segmentierungsalgorithmus filtert darauf aufbauend präzise für die Qualitätsprognose relevante Datenpunkte, sodass automatisiert aufbereitete Signaldatensätze als Eingangsgrößen für die Modellierung entstehen. Mittels durchgängiger Zusammenführung der kontextualisierten Zeitreihendaten mit Qualitätsdaten über gemeinsame Metadaten werden hochqualitative Datensätze für den unmittelbaren Einsatz von KI-basierten Methoden generiert. Dieses Verfahren zur automatisierten Datenvorbereitung beseitigt mit den zugrundeliegenden Teillösungen die Notwendigkeit manueller Vorbereitungsschritte herkömmlicher Ansätze und trägt zur breiten Anwendbarkeit datenbasierter Analysemethoden für NC-Zerspanprozesse bei.

Ein weiterer zentraler Aspekt ist das Erstellen und die Evaluation von Prozesswissensmodellen für die In-Prozess-Prognose der Werkstückqualität ba-

sierend auf den prozessparallel aufbereiteten Daten. Für die Bildung von individuellen Prognosemodellen dient im Rahmen dieser Forschungsarbeit eine entwickelte Prozesskette für das maschinelle Lernen, welche die wesentlichen Schritte von der Merkmalsextraktion und -selektion über das Trainieren verschiedener Algorithmen hin zur Bewertung und Auswahl der Prognosemodelle für die Qualitätsprädiktion automatisiert durchführt. Dafür wurden wissenschaftliche Erkenntnisse aus den Bereichen der Datenanalyse, der künstlichen Intelligenz, der Fertigungstechnik sowie der Messtechnik zusammengeführt und in einen Algorithmus für die automatisierte Parametrierung der im Kern stehenden Prozesskette überführt. Dieser erlaubt es, die gewonnenen Datensätze zu nutzen, um ohne zusätzliche manuelle Aufwände leistungsfähige Prognosemodelle für den Einsatz in der Qualitätsüberwachung zu erstellen.

Das vorgestellte Gesamtkonzept wurde auf Basis real gewonnener Daten unter produktionsnahen Bedingungen anhand von zwei Anwendungsfällen erfolgreich erprobt und die Realisierbarkeit eines darauf aufbauenden prozessparallelen Qualitätsüberwachungssystems mithilfe eines eigens entwickelten Softwareprototyps belegt. Dies legt den Grundstein für die Übertragung der erarbeiteten anwendungsnahen Lösungen auf weitere Werkstücke und Maschinen in den industriellen Kontext, sodass die breite Anwendbarkeit möglich wird.

Zusammenfassend versetzt das entwickelte Verfahren Domänenexperten in die Lage, das eigene Fachwissen zu nutzen, um unter Verwendung der vorgestellten Lösungen ohne zusätzliches Expertenwissen KI-basierte Prognosemodelle für die In-Prozess-Qualitätsüberwachung zu erstellen und diese in ein Qualitätsüberwachungssystem zu überführen.

## 8.2 Ausblick

Die anwendungsnah entwickelten Lösungen bilden die Grundlage für eine potenziell breite industrielle Anwendung prozessparalleler Qualitätsprognosen. Im Fokus stehen hierbei Domänenexperten, welche mithilfe der erzielten Ergebnisse aus dieser Arbeit in der Umsetzung datenbasierter Lösungen für Zerspanprozesse unterstützt werden. Perspektivisch ergeben sich daraus Möglichkeiten ohne tiefes Verständnis über Datenanalysen bzw. KI-basierte Methoden das vorhandene Expertenwissen zu nutzen, um datengetriebene Qualitätsüberwachungssysteme umzusetzen und zukünftig Qualitätsmessprozesse einzusparen. Die Lösungen wurden so konzipiert, dass diese auf ein breites Spektrum von Maschinen und Werkstücken übertragbar sind. Hierbei gilt es

anzusetzen und im industriellen Kontext hochqualitative Datensätze für ein breites Bauteil-, Werkstoff- und Werkzeugspektrum mit einer deutlich höheren Anzahl an Beispielen zu generieren, um die Methoden zur Erstellung der Prognosemodelle sowie die Prognoseleistungen der Modelle bei einem langfristigen Produktiveinsatz auf verschiedensten Anlagen zu bewerten. Zusätzlich können weitere Datensätze aus realen Fertigungsszenarien als Basis dienen, um Regressionsverfahren zu entwickeln, welche Rückschlüsse auf driftende Qualitätsveränderungen zulassen und um darüber hinaus weitere Methoden für Ursachenanalysen bei abweichender Qualität zu erforschen. Diese könnten zukünftig genutzt werden, um Maßnahmen gegen systematische Fehlereinfüsse einzuleiten und die Qualität der Produkte zu erhöhen.

Neben dem industriellen Einsatzpotenzial ergeben sich wissenschaftliche Perspektiven für die Weiterentwicklung des Gesamtkonzepts. Denkbar ist eine Erweiterung der Zielgrößen auf weitere Qualitätskriterien, wie die Oberflächenbeschaffenheit und Randzoneneigenschaften. Aufgrund deren mikroskopischen Struktur kann es erforderlich werden, zusätzliche externe Sensorik mit in zukünftige Untersuchungen einzubeziehen, um höherfrequente Phänomene erfassen zu können. Dabei sollte der Fokus insbesondere auf geeigneten Synchronisationsstrategien zwischen maschineninterner und externer Sensorik gelegt werden, um die kontextbasierte Verknüpfung der externen Signaldaten zu garantieren und den erreichten Automatisierungsgrad der vorgestellten Lösungen sicherzustellen.

Darüber hinaus sind die datenbasierten Prognosen der Modelle mit gewissen Unsicherheiten verbunden. Diese treten auf, wenn ML-Modelle mit einer begrenzten Quantität und Qualität an Trainingsdaten trainiert werden [TRIP20]. Dadurch sinkt die Robustheit der Modellvorhersagen und die erforderlichen technisch nachvollziehbaren, fehlertoleranten Entscheidungen gegenüber natürlichen Störungen sind nicht mehr gewährleistet [MACK21]. Deshalb gilt es, weitere Forschungstätigkeiten in Richtung robustheitsbewertender Kenngrößen auszurichten, um die Robustheit der Modellprognosen zu verbessern bzw. einschätzen zu können und im produktiven Einsatz entsprechende Handlungen einzuleiten. Erfolgreiche Ergebnisse könnten neben einer Detektion von Fehlvorhersagen, zusätzlich die Erkennung von auftretenden Veränderungen in Form von Drifts bzw. Fehlstellen in den eingangsseitigen Daten ermöglichen. Dies lässt Rückschlüsse auf sinkende Prognoseleistungen der eingesetzten Modelle zu, sodass nach definierten Einsatzperioden Nachjustierungen der Modelle in Form von zusätzlichen bzw. erneuten Trainingsdurchläufen erfolgen können.

Die vorliegende Dissertationsschrift stellt abschließend mit den erarbeiteten Lösungen zur Datenstrukturierung und -modellierung einen ersten Grundstein auf dem Weg zu übergreifend nutzbaren hochqualitativen Datensätzen in digitalisierten industriellen Prozessen dar. Damit wird ein entscheidender Beitrag zur Industrialisierung von KI im Kontext Gaia-X konformer Daten- und Serviceökosysteme geleistet.

# Literaturverzeichnis

## Literaturquellen

- [ABEL10] J. V. Abellan-Nebot und F. Romero Subirón. „A review of machining monitoring systems based on artificial intelligence process models“. In: *The International Journal of Advanced Manufacturing Technology* 47.1-4 (2010), S. 237–257.
- [ARNA16] Á. Arnaiz-González, A. Fernández-Valdivielso, A. Bustillo und L. N. López de Lacalle. „Using artificial neural networks for the prediction of dimensional error on inclined surfaces manufactured by ball-end milling“. In: *The International Journal of Advanced Manufacturing Technology* 83.5-8 (2016), S. 847–859.
- [AUER15] T. Auerbach, S. Gierlings, D. Veselovac, R. Seidner, S. Kamps und F. Klocke. „Concept for a Technology Assistance System to Analyze and Evaluate Materials and Tools for Milling“. In: *Proceedings of the ASME Turbo Expo*. 2015.
- [AUER18] T. Auerbach. „Automatisierte, datenbasierte Wissensgenerierung beim Fräsen“. Dissertation. 2018.
- [BACK21] K. Backhaus, B. Erichson, S. Gensler, R. Weiber und T. Weiber. *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. 16., vollst. überarb. und erw. Auflage. Wiesbaden: Springer Gabler, 2021.
- [BAHM13] W. Bahmann. *Werkzeugmaschinen Kompakt: Baugruppen, Einsatz und Trends*. 21. Auflage. Wiesbaden: Springer, 2013.
- [BARA20] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz und H. Gamboa. „TSFEL: Time Series Feature Extraction Library“. In: *SoftwareX* 11.10 (2020), S. 100456.
- [BART21] D. Barton, J. Federhen und J. Fleischer. „Retrofittable vibration-based monitoring of milling processes using wavelet packet transform“. In: *Procedia CIRP* 96.3 (2021), S. 353–358.

- [BASS07] A. M. Bassiuny und X. Li. „Flute breakage detection during end milling using Hilbert–Huang transform and smoothed nonlinear energy operator“. In: *International Journal of Machine Tools and Manufacture* 47.6 (2007), S. 1011–1020.
- [BEGG99] C. Beggan, M. Woulfe, P. Young und G. Byrne. „Using Acoustic Emission to Predict Surface Quality“. In: *The International Journal of Advanced Manufacturing Technology* 15.10 (1999), S. 737–742.
- [BENA03] P. G. Benardos und G.-C. Vosniakos. „Predicting surface roughness in machining: a review“. In: *International Journal of Machine Tools and Manufacture* 43.8 (2003), S. 833–844.
- [BENE08] J. Benesty, J. Chen und Y. Huang. „On the Importance of the Pearson Correlation Coefficient in Noise Reduction“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.4 (2008), S. 757–765.
- [BOIS22a] J. du Boisberranger, J. Van den Bossche, L. Estève, T. J. Fan, A. Gramfort, O. Grisel und Y. Halchenko. User Guide Scikit learn - Common pitfalls and recommended practices. 2022. URL: [https://scikit-learn.org/stable/common\\_pitfalls.html](https://scikit-learn.org/stable/common_pitfalls.html) (Zuletzt geprüft am: 05.08.2022).
- [BOIS22b] J. du Boisberranger, J. Van den Bossche, L. Estève, T. J. Fan, A. Gramfort, O. Grisel und Y. Halchenko. User Guide Scikit learn - Ensemble Methods. 2022. URL: <https://scikit-learn.org/stable/modules/ensemble.html> (Zuletzt geprüft am: 05.08.2022).
- [BOIS22c] J. du Boisberranger, J. Van den Bossche, L. Estève, T. J. Fan, A. Gramfort, O. Grisel und Y. Halchenko. User Guide Scikit learn - Linear Models. 2022. URL: [https://scikit-learn.org/stable/modules/linear\\_model.html#ridge-regression](https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression) (Zuletzt geprüft am: 05.08.2022).
- [BOIS22d] J. du Boisberranger, J. Van den Bossche, L. Estève, T. J. Fan, A. Gramfort, O. Grisel und Y. Halchenko. User Guide Scikit learn - Naive Bayes. 2022. URL: [https://scikit-learn.org/stable/modules/naive\\_bayes.html](https://scikit-learn.org/stable/modules/naive_bayes.html) (Zuletzt geprüft am: 05.08.2022).

- 
- [BREC17] C. Brecher, M. Fey und D. Christoffers. „Einfluss der Komponentendämpfung auf das Gesamtsystem Werkzeugmaschine“. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 112.12 (2017), S. 839–842.
- [BREC19a] C. Brecher, H.-M. Eckel, T. Motschke, M. Fey und A. Epple. „Estimation of the virtual workpiece quality by the use of a spindle-integrated process force measurement“. In: *CIRP Annals* 68.1 (2019), S. 381–384.
- [BREC19b] C. Brecher, M. Königs und S. Neus. „Prozessparallele Qualitätssicherung in der spanenden Fertigung.“ In: *Werkstoffe in der Fertigung* 56.3 (2019), S. 21–22.
- [BREC19c] C. Brecher, J. Ochel, V. Lohrmann und M. Fex. „Merkmalsbasierte Qualitätsprädiktion durch maschinelles Lernen: Anwendung künstlicher neuronaler Netze zur prozessparallelen virtuellen Prüfung von Qualitätsmerkmalen anhand maschineninterner Daten“. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 114.11 (2019), S. 784–787.
- [BREC19d] C. Brecher und M. Weck. *Werkzeugmaschinen Fertigungssysteme 1: Maschinenarten und Anwendungsbereiche*. 9. Auflage. Berlin und Heidelberg: Springer, 2019.
- [BREC20a] C. Brecher, B. M. H. Biernat, M. Fey, J. Ochel und M. Wiesch. „Analytics in der Produktion“. In: *Internet of Production - Turning Data into Value*. Fraunhofer-Gesellschaft, 2020, S. 186–208.
- [BREC20b] C. Brecher, J. Ochel, V. Lohrmann und M. Fex. „Machinelles Lernen zur Prädiktion der Bauteilqualität: Erweiterung eines Ansatzes zur merkmalsbasierten Qualitätsprädiktion durch künstliche Neuronale Netze“. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 115.11 (2020), S. 834–837.
- [BREC21] C. Brecher und M. Weck. *Werkzeugmaschinen Fertigungssysteme 3: Mechatronische Systeme, Steuerungstechnik und Automatisierung*. 9. Auflage. Berlin und Heidelberg: Springer, 2021.
- [BREI01] L. Breiman. „Random forests“. In: *Machine learning* 45.1 (2001), S. 5–32.

- [BRIN09] J.-W. Brinkhaus. „Statistische Verfahren zur selbstlernenden Überwachung spanender Bearbeitungen in Werkzeugmaschinen“. Dissertation. Leibniz Universität Hannover, 2009.
- [BROW17] J. Brownlee. What is the Difference Between a Parameter and a Hyperparameter? 2017. URL: <https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/> (Zuletzt geprüft am: 08.08.2022).
- [BROW21] J. Brownlee. No Free Lunch Theorem for Machine Learning. 2021. URL: <https://machinelearningmastery.com/no-free-lunch-theorem-for-machine-learning/> (Zuletzt geprüft am: 03.08.2022).
- [BRÜG15] H. Brüggemann. Grundlagen Qualitätsmanagement: Von den Werkzeugen über Methoden zum TQM. 2., überarb. u. erw. Auflage. Wiesbaden: Springer Vieweg, 2015.
- [BRÜG20] H. Brüggemann und P. Bremer. Grundlagen Qualitätsmanagement: Von den Werkzeugen über Methoden zum TQM. 3. Auflage. Wiesbaden: Springer Vieweg, 2020.
- [BUDA95] E. Budak und Y. Altintas. „Modeling and avoidance of static form errors in peripheral milling of plates“. In: *International Journal of Machine Tools and Manufacture* 35.3 (1995), S. 459–476.
- [BUGH18] J. Bughin, J. Seong, J. Manyika, M. Chui und R. Joshi. Notes from the AI frontier: Modeling the impact of AI on the world economy. 2018. URL: <https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-modeling-the-impact-of-ai-on-the-world-economy> (Zuletzt geprüft am: 30.10.2022).
- [BUND18] Bundesregierung. Strategie Künstliche Intelligenz der Bundesregierung - Stand: November 2018. 2018.
- [BUND20] Bundesregierung. Strategie Künstliche Intelligenz der Bundesregierung - Fortschreibung 2020. 2020.
- [BUSC17] F. Buschbacher und M. Weber. Künstliche Intelligenz: Wirtschaftliche Bedeutung, gesellschaftliche Herausforderungen, menschliche Verantwortung. Berlin und Kaiserslautern, 2017. URL: [https://www.dfki.de/fileadmin/user\\_upload/import/9744\\_171012-KI-Gipfpapier-online.pdf](https://www.dfki.de/fileadmin/user_upload/import/9744_171012-KI-Gipfpapier-online.pdf) (Zuletzt geprüft am: 10.01.2022).

- 
- [BUXM21] P. Buxmann und H. Schmidt, Hrsg. Künstliche Intelligenz: Mit Algorithmen zum wirtschaftlichen Erfolg. 2. Auflage. Berlin und Heidelberg: Springer, 2021.
- [BYRN95] G. Byrne, D. Dornfeld, I. Inasaki, G. Ketteler, W. König und R. Teti. „Tool Condition Monitoring (TCM) — The Status of Research and Industrial Application“. In: *CIRP Annals* 44.2 (1995), S. 541–567.
- [CAO13] H. Cao, Y. Lei und Z. He. „Chatter identification in end milling process using wavelet packets and Hilbert–Huang transform“. In: *International Journal of Machine Tools and Manufacture* 69 (2013), S. 11–19.
- [CERA22] CERATIZIT S.A. Volle Prozesskontrolle mit ToolScope: Digitale Überwachung für Ihre Fertigung. 2022. URL: <https://cuttingtools.ceratizit.com/de/de/services/toolscope.html> (Zuletzt geprüft am: 24.01.2022).
- [CHEN12] C.-C. Chen, N.-M. Liu, K.-T. Chiang und H.-L. Chen. „Experimental investigation of tool vibration and surface roughness in the precision end-milling process using the singular spectrum analysis“. In: *The International Journal of Advanced Manufacturing Technology* 63.5-8 (2012), S. 797–815.
- [CHO99] D.-W. Cho, S. J. Lee und C. N. Chu. „The state of machining process monitoring research in Korea“. In: *International Journal of Machine Tools and Manufacture* 39.11 (1999), S. 1697–1715.
- [COPE21] B. J. Copeland. artificial intelligence. 2021. URL: <https://www.britannica.com/technology/artificial-intelligence> (Zuletzt geprüft am: 10.01.2022).
- [DALE16] J. Dalei und K. B. Mohanty. „Fault classification in SEIG system using Hilbert-Huang transform and least square support vector machine“. In: *International Journal of Electrical Power & Energy Systems* 76 (2016), S. 11–22.
- [DAS16] O. Das. Speaker Recognition: Tech. rep. 2016. URL: [https://cirma.stanford.edu/~orch/ Documents / speaker\\_recognition\\_report.pdf](https://cirma.stanford.edu/~orch/ Documents / speaker_recognition_report.pdf) (Zuletzt geprüft am: 06.11.2022).
- [DECK19] DECKEL MAHO Seebach GmbH. Betriebsanleitung - DMG MORI DMC850V. 2019.

- [DELO16] Deloitte GmbH Wirtschaftsprüfungsgesellschaft. Manufacturing 4.0: Meilenstein, Must-Have oder Millionengrab? Warum bei M4.0 die Integration den entscheidenden Unterschied macht. 2016. URL: [https://www2.deloitte.com/content/dam/Deloitte/de/Documents/operations/DELO-2267\\_Manufacturing-4.0-Studie\\_s.pdf](https://www2.deloitte.com/content/dam/Deloitte/de/Documents/operations/DELO-2267_Manufacturing-4.0-Studie_s.pdf) (Zuletzt geprüft am: 14. 01. 2022).
- [DENK11] B. Denkena. Spanen: Grundlagen. 3. Auflage. Berlin und Heidelberg: Springer, 2011.
- [DENK12] B. Denkena, M. Krüger, D. Bachrathy und G. Stepan. „Model based reconstruction of milled surface topography from measured cutting forces“. In: *International Journal of Machine Tools and Manufacture* 54-55.6 (2012), S. 25–33.
- [DENK20] B. Denkena, M.-A. Dittrich, M. Lindauer, J. Mainka und L. Stürenburg. „Using AutoML to Optimize Shape Error Prediction in Milling Processes“. In: *Proceedings of the Machining Innovations Conference (MIC) 2020*. 2020.
- [DESE70] L. Deselaers. „Beitrag zur Kenntnis der Schnittkräfte beim Drehen, Hobeln und Bohren“. Dissertation. Universität Karlsruhe, 1970.
- [DIEB06] J. Diebel. „Representing attitude: Euler angles, unit quaternions, and rotation vectors“. In: *Matrix* 58.15-16 (2006), S. 1–35.
- [DIET00] T. G. Dietterich. „Ensemble methods in machine learning“. In: *International workshop on multiple classifier systems*. Hrsg. von J. Kittler und F. Roli. Springer Berlin Heidelberg, 2000, S. 1–15.
- [DÖBE18] I. Döbel, M. Leis, M. M. Vogelsang, D. Neustroev, H. Petzka, A. Riemer, S. Rüping, A. Voss, M. Wegele und J. Welz. Maschinelles Lernen. Eine Analyse zu Kompetenzen, Forschung und Anwendung. München, 2018.
- [DRIS11] M. Driscoll. Building data startups: Fast, big, and focused. Low costs and cloud tools are empowering new data startups. 2011. URL: <http://radar.oreilly.com/2011/08/building-data-startups.html> (Zuletzt geprüft am: 07. 02. 2022).
- [DUNN74] O. J. Dunn und V. Clark. Applied statistics: analysis of variance and regression. New York: Wiley, 1974.

- 
- [EBER14] S. Eberlin und B. Hock. Zuverlässigkeit und Verfügbarkeit technischer Systeme: Eine Einführung in die Praxis. Wiesbaden: Springer Vieweg, 2014.
- [EFRO04] B. Efron, T. Hastie, I. Johnstone und R. Tibshirani. „Least angle regression“. In: *The Annals of Statistics* 32.2 (2004), S. 407–499.
- [ELAN15] M. Elangovan, N. R. Sakthivel, S. Saravanamurugan, B. Nair und V. Sugumaran. „Machine Learning Approach to the Prediction of Surface Roughness Using Statistical Features of Vibration Signal Acquired in Turning“. In: *Procedia Computer Science* 50.7 (2015), S. 282–288.
- [ELLI19] J. Ellinger, M. Benker, J. Rowe, R. Kleinwort, M. F. Zäh und S. Unsleber. „Zustandsüberwachung von Vorschubantrieben mithilfe eingebetteter Sensoren“. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 114.3 (2019), S. 115–118.
- [ERTA20] H. Ertan. Which features to use in your model? 2020. URL: <https://medium.com/@hertan06/which-features-to-use-in-your-model-350630a1e31c> (Zuletzt geprüft am: 26.07.2022).
- [ESSE01] C. Esser. Wavelet Transformation von Standbildern. Hrsg. von W. Effelsberg. Studienarbeit. 2001. URL: <https://pi4.informatik.uni-mannheim.de/pi4.data/content/animations/wavelet/WaveletDemo.pdf> (Zuletzt geprüft am: 24.01.2022).
- [FAWC06] T. Fawcett. „An introduction to ROC analysis“. In: *Pattern Recognition Letters* 27.8 (2006), S. 861–874.
- [FERT20] A. Fertig, C. Bauerdick und M. Weigold. „In-Process Quality Monitoring During Turning Based on High Frequency Machine Data“. In: *Proceedings of the Machining Innovations Conference (MIC) 2020*. 2020.
- [FERT22a] A. Fertig, O. Kohn, B. Brockhaus und M. Weigold. „Consistent Contextualisation of Process and Quality Information for Machining Processes“. In: *Production at the Leading Edge of Technology*. Hrsg. von B.-A. Behrens, A. Brosius, W.-G. Drossel, W. Hintze, S. Ihlenfeldt und P. Nyhuis. Springer International Publishing, 2022, S. 195–202.

- [FERT22b] A. Fertig, M. Weigold und Y. Chen. „Machine Learning based quality prediction for milling processes using internal machine tool data“. In: *Advances in Industrial and Manufacturing Engineering* 4 (2022), S. 100074.
- [FERT22c] A. Fertig, C. Preis und M. Weigold. „Quality prediction for milling processes: automated parametrization of an end-to-end machine learning pipeline“. In: *Production Engineering* (2022).
- [FERT23] A. Fertig u. a. Intelligente Vernetzung zur autonomen Fräsbearbeitung von Strukturbauteilen - Ergebnisbericht des BMBF Verbundprojektes TensorMill. Darmstadt, 2023.
- [FLAC17] P. A. Flach. „ROC Analysis“. In: *Encyclopedia of Machine Learning and Data Mining*. Hrsg. von C. Sammut und G. I. Webb. Boston: Springer US, 2017, S. 1109–1116.
- [FORS19] D. Forsyth. Applied Machine Learning. 1st edition. Cham: Springer, 2019.
- [FREU97] Y. Freund und R. E. Schapire. „A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting“. In: *Journal of Computer and System Sciences* 55.1 (1997), S. 119–139.
- [FUMO17] D. Fumo. Why Is Everyone Talking About Artificial Intelligence? 2017. URL: <https://towardsdatascience.com/why-is-everyone-talking-about-ai-73bab31bf9c1> (Zuletzt geprüft am: 13.01.2022).
- [GARD93] H. Gardner. Frames of mind: The theory of multiple intelligences. 2nd edition. New York: Basic Books, 1993.
- [GERE19] D. Geretshuber und H. Reese. Künstliche Intelligenz in Unternehmen. 2019. URL: <https://www.pwc.de/de/digitale-transformation/kuenstliche-intelligenz/studie-kuenstliche-intelligenz-in-unternehmen.pdf> (Zuletzt geprüft am: 13.01.2022).
- [GÉRO20] A. Géron. Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme. 2. Auflage. Heidelberg: O’Reilly, 2020.
- [GEUR06] P. Geurts, D. Ernst und L. Wehenkel. „Extremely randomized trees“. In: *Machine learning* 63.1 (2006), S. 3–42.

- 
- [GEY03] C. Gey. „Prozessauslegung für das Flankenfräsen von Titan“. Dissertation. Leibniz Universität Hannover, 2003.
- [GOOD16] I. Goodfellow, Y. Bengio und A. Courville. Deep Learning. Cambridge: MIT Press, 2016.
- [GROB20] Grob-Werke GmbH & Co. KG, Hrsg. Technische Dokumentation - G352 Betriebsanleitung. 2020.
- [GROSS06] K. Großmann, A. Mühl und M. Löser. „Prognose von Stabilitätsgrenzen für das Fräsen“. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 101.7-8 (2006), S. 416–421.
- [GUO05] Y. B. Guo und S. C. Ammula. „Real-time acoustic emission monitoring for surface damage in hard machining“. In: *International Journal of Machine Tools and Manufacture* 45.14 (2005), S. 1622–1627.
- [GUYO06] I. Guyon, S. Gunn, M. Nikravesh und L. A. Zadeh, Hrsg. Feature extraction: Foundations and applications. Bd. 207. Berlin, Heidelberg und New York: Springer, 2006.
- [HÄNE20] A. Hänel, T. Schnellhardt, E. Wenkler, A. Nestler, A. Brosius, C. Corinth, A. Fay und S. Ihlenfeldt. „The development of a digital twin for machining processes for the application in aerospace industry“. In: *Procedia CIRP* 93 (2020), S. 1399–1404.
- [HASE18] C. Hasenfratz. „Modellgestützte Prozessauslegung zum linearen Schaftfräsen von tiefen Kavitäten in TiAl6V4-Verdichterscheiben“. Dissertation. Technische Universität Darmstadt, 2018.
- [HAST09] T. Hastie, R. Tibshirani und J. H. Friedman. The elements of statistical learning: Data mining, inference, and prediction. 2nd edition. New York: Springer, 2009.
- [HATA17] N. Hatami, Y. Gavet und J. Debayle. „Classification of Time-Series Images Using Deep Convolutional Neural Networks“. In: *CoRR* abs/1710.00886 (2017).
- [HATI19] B. Hatiboglu, S. Schuler, A. Bildstein und M. Hämmerle. Einsatzfelder von künstlicher Intelligenz im Produktionsumfeld. Hrsg. von F. I. Fraunhofer IAO. 2019.

- [HECK14] H. Heckner und M. Wirth. Vergleich von Dateiformaten für 3D-Modelle. Hrsg. von Center for Digital Fabrication. 2014. URL: <https://fdokument.com/document/vergleich-von-dateiformaten-fr-3d-von-dateiformaten-fr-3d-modelle-cedifa-arbeitsbericht.html?page=1> (Zuletzt geprüft am: 24.09.2022).
- [HEIN19] M. Heinzmann und E. Marquardt. Maschinelles Lernen: VDI Statusreport 2019. 2019.
- [HEIS92] U. Heisel, O. Novaski und G. Roth. „Einflußfaktoren auf die Werkstückqualität und Güte beim Feinstdrehen“. In: *Dima* 46.12 (1992), S. 59–63.
- [HENR03] U. Henriksson. Power Spectrum and Bandwidth. 2003. URL: <https://www.commsys.isy.liu.se/TSDT45/Material/UlfsSpectrum2003.pdf> (Zuletzt geprüft am: 06.11.2022).
- [HINT09] H. H. Hinterhuber und K. Matzler, Hrsg. Kundenorientierte Unternehmensführung: Kundenorientierung - Kundenzufriedenheit - Kundenbindung. 6., überarb. Auflage. Wiesbaden: Springer Gabler, 2009.
- [HO09] W.-H. Ho, J.-T. Tsai, B.-T. Lin und J.-H. Chou. „Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid Taguchi-genetic learning algorithm“. In: *Expert Systems with Applications* 36.2 (2009), S. 3216–3222.
- [HOFF22] Hoffmann SE. VHM-Fräser mit Spanteilern TPC TiAlN (203089). 2022. URL: <https://www.hoffmann-group.com/DE/de/hom/Monozerspanung/VHM-Fr%C3%A4ser/VHM-Fr%C3%A4ser-mit-Spanteilern-TPC-TiAlN/p/203089> (Zuletzt geprüft am: 17.09.2022).
- [HSM 22] HSM Hochleistungswerkstoffe. Werkstoffdatenblatt 3.7164 / Ti6Al4V. 2022. URL: [https://www.hsm-stahl.de/fileadmin/user\\_upload/datenblatt/HSM\\_Datenblatt\\_3.7164.pdf](https://www.hsm-stahl.de/fileadmin/user_upload/datenblatt/HSM_Datenblatt_3.7164.pdf) (Zuletzt geprüft am: 28.06.2022).
- [HUAN05a] N. E. Huang und N. O. Attoh-Okine, Hrsg. The Hilbert-Huang transform in engineering. Boca Raton: Taylor & Francis Group, 2005.
- [HUAN05b] N. E. Huang und S. S. P. Shen, Hrsg. Hilbert-Huang transform and its applications. Bd. 5. River Edge: World Scientific, 2005.

- 
- [HUAN98] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N.-C. Yen, C.-C. Tung und H. Liu. „The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis“. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454 (März 1998), S. 903–995.
- [HUDE20] M. von der Hude. Predictive analytics und data mining: Eine Einführung mit R. Wiesbaden und Heidelberg: Springer Vieweg, 2020.
- [INDU14] IndustryArena GmbH. Forumsbeitrag: Variable bei Reset mit Wert belegen. 2014. URL: <https://de.industryarena.com/siemens/forum/variable-bei-reset-mit-wert-belegen--72715.html> (Zuletzt geprüft am: 22.05.2022).
- [JAME21] G. James, D. Witten, T. Hastie und R. Tibshirani. An Introduction to Statistical Learning: With Applications in R. 2nd edition. New York: Springer US und Imprint Springer, 2021.
- [JANT02] E. Jantunen. „A summary of methods applied to tool condition monitoring in drilling“. In: *International Journal of Machine Tools and Manufacture* 42.9 (2002), S. 997–1010.
- [JEMI95] K. Jemielniak und J. Kosmol. „Tool and process monitoring-state of art and future prospects“. In: *Scientific papers of the institute of mechanical engineering and automation of the Technical University of Wroclaw* 61 (1995), S. 90–112.
- [JORD20] W. Jorden und W. Schütte. Form- und Lagetoleranzen: Geometrische Produktspezifikationen (ISO GPS) in Studium und Praxis. 10., überarb. und erw. Auflage. München: Hanser, 2020.
- [JOSH20] A. V. Joshi. Machine Learning and Artificial Intelligence. Cham: Springer, 2020.
- [KAEV05] M. Kaefer. „Steuerungsintegrierte Fertigungsprozeßüberwachung bei spanender Bearbeitung“. Dissertation. RWTH Aachen, 2005.
- [KALV10] T. Kalvoda und Y.-R. Hwang. „A cutter tool monitoring in machining process using Hilbert–Huang transform“. In: *International Journal of Machine Tools and Manufacture* 50.5 (2010), S. 495–501.

- [KAMM77] H. Kamm. „Beitrag zur Optimierung des Messerkopffräsens“. Dissertation. Universität Karlsruhe, 1977.
- [KANT15] G. Kant und K. S. Sangwan. „Predictive Modelling and Optimization of Machining Parameters to Minimize Surface Roughness using Artificial Neural Network Coupled with Genetic Algorithm“. In: *Procedia CIRP* 31 (2015), S. 453–458.
- [KARA13] S. Karam und R. Teti. „Wavelet Transform Feature Extraction for Chip form Recognition during Carbon Steel Turning“. In: *Procedia CIRP* 12 (2013), S. 97–102.
- [KEOG17] E. Keogh und A. Mueen. „Curse of Dimensionality“. In: *Encyclopedia of Machine Learning and Data Mining*. Hrsg. von C. Sammut und G. I. Webb. Boston: Springer US, 2017, S. 314–315.
- [KHOR17] A. Khorasani und M. R. S. Yazdi. „Development of a dynamic surface roughness monitoring system based on artificial neural networks (ANN) in milling operation“. In: *The International Journal of Advanced Manufacturing Technology* 93.1-4 (2017), S. 141–151.
- [KIEN52] O. Kienzle. „Die Bestimmung von Kräften und Leistungen an spanenden Werkzeugen und Werkzeugmaschinen.“ In: *VDI-Z* 94.11/12 (1952).
- [KIND22] Kind Aerospace Materials. Werkstoffdatenblatt Ti6Al4V. 2022. URL: [www.kind-aerospace.de/pdf/Werkstoffdatenblatt\\_120402.pdf](http://www.kind-aerospace.de/pdf/Werkstoffdatenblatt_120402.pdf) (Zuletzt geprüft am: 28.06.2022).
- [KIRC92] A. Kirchheim. „Prozessüberwachung bei der Mehrschnittbearbeitung“. Dissertation. Technische Universität Berlin, 1992.
- [KLIN82] W. A. Kline, R. E. DeVor und I. A. Shareef. „The Prediction of Surface Accuracy in End Milling“. In: *Journal of Engineering for Industry* 104.3 (1982), S. 272–278.
- [KLOC18] F. Klocke. *Fertigungsverfahren 1: Zerspanung mit geometrisch bestimmter Schneide*. 9. Auflage. Berlin und Heidelberg: Springer, 2018.
- [KOKO00] S. Kokoska und D. Zwillinger. *CRC Standard Probability and Statistics Tables and Formulae*. Boca Raton: CRC Press, 2000.

- 
- [KÖKS11] G. Köksal, İ. Batmaz und M. C. Testik. „A review of data mining applications for quality improvement in manufacturing industry“. In: *Expert Systems with Applications* 38.10 (2011), S. 13448–13467.
- [KÖNI18] M. Königs und C. Brecher. „Process-parallel virtual quality evaluation for metal cutting in series production“. In: *Procedia Manufacturing* 26.2 (2018), S. 1087–1093.
- [KOTU19] V. Kotu und B. Deshpande. *Data science: Concepts and practice*. 2nd edition. Cambridge: Morgan Kaufmann Publishers, 2019.
- [KOVA11] P. Kovac, I. Maňková, M. Gostimirović, M. Sekulić und B. Savkovic. „A Review of Machining Monitoring Systems“. In: *Journal of Production Engineering* 14 (2011), S. 1–6.
- [KOVA13] P. Kovac, D. Rodic, V. Pucovsky, B. Savkovic und M. Gostimirovic. „Application of fuzzy logic and regression analysis for modeling surface roughness in face milling“. In: *Journal of Intelligent Manufacturing* 24.4 (2013), S. 755–762.
- [KRAS04] A. Kraskov, H. Stögbauer und P. Grassberger. „Estimating mutual information“. In: *Phys. Rev. E* 69 (6 2004), S. 066138.
- [KRAU19] J. Krauß, M. Frye, G. T. D. Beck und R. H. Schmitt. „Selection and Application of Machine Learning- Algorithms in Production Quality“. In: *Machine Learning for Cyber Physical Systems*. Hrsg. von J. Beyerer, C. Kühnert und O. Niggemann. Bd. 9. Springer Berlin Heidelberg, 2019, S. 46–57.
- [KRÜG13] M. Krüger und B. Denkena. „A model-based approach for monitoring of shape deviations in peripheral milling“. In: *The International Journal of Advanced Manufacturing Technology* 67.9-12 (2013), S. 2537–2550.
- [KRÜG14] M. Krüger. „Modellbasierte Online-Bewertung von Fräsprozessen“. Dissertation. Leibniz Universität Hannover, 2014.
- [KUMA14] V. Kumar und S. Minz. „Feature Selection: A literature Review“. In: *The Smart Computing Review* 4.3 (2014).
- [KURS10] M. B. Kurşa und W. R. Rudnicki. „Feature Selection with the Boruta Package“. In: *Journal of Statistical Software* 36.11 (2010), S. 1–13.

- [LACK18] R. Lackes und M. Siepermann. Künstliche Intelligenz (KI): Was ist Künstliche Intelligenz (KI)? 2018. URL: <https://wirtschaftslexikon.gabler.de/definition/kuenstliche-intelligenz-ki-40285/version-263673> (Zuletzt geprüft am: 10.01.2022).
- [LAKE17] G. Lakemeyer. Künstliche Intelligenz. Bd. 261. Berlin: Konrad-Adenauer-Stiftung, 2017.
- [LARN21] A. J. Larner. The 2x2 Matrix: Contingency, Confusion and the Metrics of Binary Classification. 1st edition. Cham: Springer, 2021.
- [LARU03] A. Larue und B. Anselmetti. „Deviation of a machined surface in flank milling“. In: *International Journal of Machine Tools and Manufacture* 43.2 (2003), S. 129–138.
- [LAUR14] C. H. Lauro, L. C. Brandão, D. Baldo, R. A. Reis und J. P. Davim. „Monitoring and processing signal applied in machining processes – A review“. In: *Measurement* 58.10 (2014), S. 73–86.
- [LEE19] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt und A. O’Leary. „PyWavelets: A Python package for wavelet analysis“. In: *Journal of Open Source Software* 4.36 (2019), S. 1237.
- [LI04] X. Li, X. Guan und Y. Li. „A Hybrid Radial Basis Function Neural Network for Dimensional Error Prediction in End Milling“. In: *Advances in Neural Networks - ISNN 2004*. Hrsg. von D. Hutchison u. a. Bd. 3174. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, S. 743–748.
- [LI06] C. J. Li. „Signal Processing in Manufacturing Monitoring“. In: *Condition Monitoring and Control for Intelligent Manufacturing*. Hrsg. von L. Wang, R. X. Gao und D. T. Pham. Bd. 21. Springer London, 2006, S. 245–265.
- [LI18] Z. Li, G. Wang und G. He. „Surface quality monitoring based on time-frequency features of acoustic emission signals in end milling Inconel-718“. In: *The International Journal of Advanced Manufacturing Technology* 96.5-8 (2018), S. 2725–2733.
- [LIAN04] S. Y. Liang, R. L. Hecker und R. G. Landers. „Machining Process Monitoring and Control: The State-of-the-Art“. In: *Journal of Manufacturing Science and Engineering* 126.2 (2004), S. 297–310.

- 
- [LIAO01] Y. G. Liao und S. J. Hu. „An Integrated Model of a Fixture-Workpiece System for Surface Quality Prediction“. In: *The International Journal of Advanced Manufacturing Technology* 17.11 (2001), S. 810–818.
- [LITW11] K. M. Litwinski. „Sensorisches Spannsystem zur Überwachung von Zerspanprozessen in der Einzelteillfertigung“. Dissertation. Leibniz Universität Hannover, 2011.
- [LO03] S.-P. Lo. „An adaptive-network based fuzzy inference system for prediction of workpiece surface roughness in end milling“. In: *Journal of Materials Processing Technology* 142.3 (2003), S. 665–675.
- [LUDW20] W. Ludwig-Mayerhofer. Bayes-Theorem. 2020. URL: [https://wlm.userweb.mwn.de/Ilmes/ilm\\_b13.html](https://wlm.userweb.mwn.de/Ilmes/ilm_b13.html) (Zuletzt geprüft am: 04.08.2022).
- [LUNZ20] J. Lunze. Regelungstechnik. 12., überarb. Auflage. Berlin und Heidelberg: Springer Vieweg, 2020.
- [MACK21] M. Mackert und M. Mikoleit. Whitepaper on Robust AI Assessment. Hrsg. von Deutsche Telekom AG. Bonn, 2021.
- [MAJS19] V. D. Majstorovic, N. Durakbasa, Y. Takaya und S. Stojadinovic. „Advanced Manufacturing Metrology in Context of Industry 4.0 Model“. In: *Proceedings of the 12th International Conference on Measurement and Quality Control - Cyber Physical Issue*. Hrsg. von V. D. Majstorovic. Cham: Springer International Publishing AG, 2019, S. 1–11.
- [MAKL19] C. Maklin. AdaBoost Classifier Example In Python. 2019. URL: <https://towardsdatascience.com/machine-learning-part-17-boosting-algorithms-adaboost-in-python-d00faac6c464> (Zuletzt geprüft am: 08.08.2022).
- [MARI08] I. Marinescu und D. A. Axinte. „A critical analysis of effectiveness of acoustic emission signals to detect tool and workpiece malfunctions in milling operations“. In: *International Journal of Machine Tools and Manufacture* 48.10 (2008), S. 1148–1160.

- [MARI09] I. Marinescu und D. Axinte. „A time–frequency acoustic emission-based monitoring technique to identify workpiece surface malfunctions in milling with multiple teeth cutting simultaneously“. In: *International Journal of Machine Tools and Manufacture* 49.1 (2009), S. 53–65.
- [MARP22a] MARPOSS Monitoring Solutions GmbH. Werkzeug- und Prozessüberwachungssystem - CTM. 2022. URL: <https://artis.de/ger/product/tool-and-process-monitoring-system> (Zuletzt geprüft am: 24.01.2022).
- [MARP22b] MARPOSS Monitoring Solutions GmbH. Werkzeug- und Prozessüberwachungssystem - Genior Modular. 2022. URL: <https://artis.de/ger/product/tool-and-process-monitoring-system-2> (Zuletzt geprüft am: 24.01.2022).
- [MARX21] M. Marxer, C. Bach und C. P. Keferstein. Fertigungsmesstechnik: Alles zu Messunsicherheit, konventioneller Messtechnik und Multisensorik. 10., vollst. überarb. und erw. Auflage. Wiesbaden: Springer Vieweg, 2021.
- [MATH99] P. Mathews und M. Shunmugam. „Neural-network approach for predicting hole quality in reaming“. In: *International Journal of Machine Tools and Manufacture* 39.5 (1999), S. 723–730.
- [MCCA06] J. McCarthy, M. L. Minsky, N. Rochester und C. E. Shannon. „A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955“. In: *AI Magazine* 27.4 (2006).
- [MCCO04] P. McCorduck. *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. Boca Raton, London und New York: CRC Press, 2004.
- [MCKI17] McKinsey & Company. *Smartening up with Artificial Intelligence (AI) - What’s in it for Germany and its Industrial Sector*. 2017. URL: <https://www.mckinsey.com/~media/mckinsey/industries/semiconductors/our%20insights/smartening%20up%20with%20artificial%20intelligence/smartening-up-with-artificial-intelligence.ashx> (Zuletzt geprüft am: 14.01.2022).
- [MCU 22] MCU GmbH & Co. KG. *Prozessüberwachungssystem Toolinspect*. 2022. URL: <https://www.mcu-gmbh.de/produkte/toolinspect> (Zuletzt geprüft am: 24.01.2022).

- 
- [MERT13] A. Mertins. Signaltheorie: Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung. 3., überarb. u. erw. Auflage. Wiesbaden: Springer Vieweg, 2013.
- [MITC97] T. M. Mitchell. Machine Learning. McGraw-Hill Science/Engineering/Math, 1997.
- [MOCK21] A. Mockenhaupt. Digitalisierung und Künstliche Intelligenz in der Produktion: Grundlagen und Anwendung. Wiesbaden: Springer Vieweg, 2021.
- [MQTT22] mqtt.org. MQTT: The Standard for IoT Messaging. 2022. URL: <https://mqtt.org/> (Zuletzt geprüft am: 01.03.2022).
- [MURP12] K. P. Murphy. Machine learning: A probabilistic perspective. Cambridge: MIT Press, 2012.
- [NETO13] F. C. Neto, T. M. Gerônimo, C. Cruz, P. R. Aguiar und E. Bianchi. „Neural Models for Predicting Hole Diameters in Drilling Processes“. In: *Procedia CIRP* 12.2 (2013), S. 49–54.
- [NEUG12] R. Neugebauer. Aufbau, Funktion und Anwendung von spannenden und abtragenden Werkzeugmaschinen. 1. Auflage. Berlin und Heidelberg: Springer, 2012.
- [NEUM05] H. J. Neumann. Präzisionsmesstechnik in der Fertigung mit Koordinatenmessgeräten. 2., überarb. Auflage. Bd. 646. Renningen: expert verlag, 2005.
- [OLIP06] T. E. Oliphant. Guide to NumPy. 2006. URL: <https://web.mit.edu/dvp/Public/numpybook.pdf> (Zuletzt geprüft am: 07.11.2022).
- [PAAS20] G. Paass und D. Hecker. Künstliche Intelligenz: Was steckt hinter der Technologie der Zukunft? Wiesbaden: Springer Vieweg, 2020.
- [PAHU20] R. Pahuja und R. Mamidala. „Quality monitoring in milling of unidirectional CFRP through wavelet packet transform of force signals“. In: *Procedia Manufacturing* 48 (2020), S. 388–399.

- [PAN09] Y. N. Pan, J. Chen und X. L. Li. „Spectral entropy: A complementary index for rolling element bearing performance degradation assessment“. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 223.5 (2009), S. 1223–1231.
- [PARL20] E. Parlament, Hrsg. Was ist künstliche Intelligenz und wie wird sie genutzt? 2020. URL: <https://www.europarl.europa.eu/news/de/headlines/society/20200827STO85804/was-ist-kunstliche-intelligenz-und-wie-wird-sie-genutzt> (Zuletzt geprüft am: 11.01.2022).
- [PAUC08] E. Paucksch, S. Holsten, M. Lin und F. Tikal. Zerspantechnik: Prozesse, Werkzeuge, Technologien. 12., vollst. überarb. und erw. Auflage. Wiesbaden: Vieweg + Teubner, 2008.
- [PEAR96] K. Pearson. „Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia“. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 187 (1896), S. 253–318.
- [PEDR12] F. Pedregosa u. a. „Scikit-learn: Machine Learning in Python“. In: *arXiv* (2012).
- [PEET11] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis und S. McAdams. „The Timbre Toolbox: extracting audio descriptors from musical signals“. In: *The Journal of the Acoustical Society of America* 130.5 (2011), S. 2902–2916.
- [PENG06] Y. Peng. „Empirical Model Decomposition Based Time-Frequency Analysis for the Effective Detection of Tool Breakage“. In: *Journal of Manufacturing Science and Engineering* 128.1 (2006), S. 154–166.
- [PFEI10] T. Pfeifer und R. Schmitt. Fertigungsmesstechnik. 3., überarb. und erw. Auflage. München: Oldenbourg, 2010.
- [PODD14] P. Podder, T. Khan, M. Khan und M. Rahman. „Comparative Performance Analysis of Hamming, Hanning and Blackman Window“. In: *International Journal of Computer Applications* 96 (2014), S. 1–7.
- [PRIC99] P. Prickett und C. Johns. „An overview of approaches to end milling tool monitoring“. In: *International Journal of Machine Tools and Manufacture* 39.1 (1999), S. 105–122.

- 
- [PRIT01] G. Pritschow, Y. Altintas, F. Jovane, Y. Koren, M. Mitsuishi, S. Takata, H. van Brussel, M. Weck und K. Yamazaki. „Open Controller Architecture - Past, Present and Future“. In: *CIRP Annals* 50.2 (2001), S. 463–470.
- [PRIT93] G. Pritschow, C. Daniel, G. Junghans und W. Sperling. „Open System Controllers – A Challenge for the Future of the Machine Tool Industry“. In: *CIRP Annals* 42.1 (1993), S. 449–452.
- [PYTH22] Python Software Foundation. What is Python? Executive Summary. 2022. URL: <https://www.python.org/doc/essays/blurp/> (Zuletzt geprüft am: 01.07.2022).
- [QUIN21] A. J. Quinn, V. Lopes-dos-Santos, D. Dupret, A. C. Nobre und M. W. Woolrich. „EMD: Empirical Mode Decomposition and Hilbert-Huang Spectral Analyses in Python“. In: *Journal of Open Source Software* 6.59 (2021), S. 2977.
- [RAZ20] A. K. Raz, J. Llinas, R. Mittu und W. F. Lawless. „Engineering for emergence in information fusion systems: A review of some challenges“. In: *Human-Machine Shared Contexts*. Bd. 358. Elsevier, 2020, S. 241–255.
- [REHO05] A. G. Rehorn, J. Jiang und P. E. Orban. „State-of-the-art methods and results in tool condition monitoring: a review“. In: *The International Journal of Advanced Manufacturing Technology* 26.7-8 (2005), S. 693–710.
- [REHS99] M. Rehse. „Flexible Prozeßüberwachung bei der Bohr- und Fräsbearbeitung in einer autonomen Produktionszelle“. Dissertation. RWTH Aachen, 1999.
- [REUB01] M. Reuber. „Prozessüberwachung beim Schlichtfräsen von Freifromflächen“. Dissertation. RWTH Aachen, 2001.
- [RMIL16] W. Rmili, A. Ouahabi, R. Serra und R. Leroy. „An automatic system based on vibratory analysis for cutting tool wear monitoring“. In: *Measurement* 77 (2016), S. 117–123.
- [RODG88] J. L. Rodgers und W. A. Nicewander. „Thirteen Ways to Look at the Correlation Coefficient“. In: *The American Statistician* 42.1 (1988), S. 59–66.

- [RONS20] J. Ronsdorf. Microsoft erklärt: Was ist künstliche Intelligenz? Definition & Funktionen von KI. 2020. URL: <https://news.microsoft.com/de-de/einfach-erklaert-was-ist-kuenstliche-intelligenz/> (Zuletzt geprüft am: 11.01.2022).
- [ROSE58] F. Rosenblatt. „The perceptron: a probabilistic model for information storage and organization in the brain.“ In: *Psychological Review* 65(6) (1958), S. 386–408.
- [ROSS14] B. C. Ross. „Mutual Information between Discrete and Continuous Data Sets“. In: *PLOS ONE* 9.2 (2014), S. 1–5.
- [ROST15] H. Rostami, J.-Y. Dantan und L. Homri. „Review of data mining applications for quality assessment in manufacturing industry: support vector machines“. In: *International Journal of Metrology and Quality Engineering* 6.4 (2015).
- [ROY19] S. Roy. The relationship between correlation, mutual information, and p-values. 2019. URL: <https://scottroy.github.io/the-relationship-between-correlation-mutual-information-and-p-values.html> (Zuletzt geprüft am: 26.07.2022).
- [RUNK15] T. A. Runkler. Data Mining. Wiesbaden: Springer, 2015.
- [RYU12] S. H. Ryu. „An analytical expression for end milling forces and tool deflection using Fourier series“. In: *The International Journal of Advanced Manufacturing Technology* 59.1-4 (2012), S. 37–46.
- [SAMM10] „Accuracy“. In: *Encyclopedia of Machine Learning*. Hrsg. von C. Sammut und G. I. Webb. Boston: Springer US, 2010, S. 9–10.
- [SARI21] E. Sarikaya, B. Brockhaus, A. Fertig, H. Ranzau, P. Stanula und J. Walther. Data Driven Production – Application Fields, Solutions and Benefits. Darmstadt, 2021.
- [SCHA13] R. Schapire. „Explaining AdaBoost“. In: *Empirical Inference*. Springer Berlin Heidelberg, 2013, S. 37–52.
- [SCHL05] S. Schlagner und U. Strehlau. „Fourier-Analyse versus Wavelet-Analyse“. In: *PAMM* 5.1 (2005), S. 125–126.
- [SCHM11] C. Schmidt. „Einflussgrößensensitive Simulation und Überwachung von Fräsprozessen“. Dissertation. Leibniz Universität Hannover, 2011.

- 
- [SCHM15] J. Schmidhuber. „Deep learning in neural networks: an overview“. In: *Neural networks : the official journal of the International Neural Network Society* 61 (2015), S. 85–117.
- [SCHM20a] J. Schmitt, J. Böning, T. Borggräfe, G. Beitingner und J. Deuse. „Predictive model-based quality inspection using Machine Learning and Edge Cloud Computing“. In: *Advanced Engineering Informatics* 45.4 (2020), S. 101101.
- [SCHM20b] R. H. Schmitt, R. Kurzhals, M. Ellerich, G. Nilgen, P. Schlegel, E. Dietrich, J. Krauß, A. Latz, J. Gregori und N. Miller. „Predictive Quality - Data Analytics in produzierenden Unternehmen“. In: *Internet of Production - Turning Data into Value*. Fraunhofer-Gesellschaft, 2020, S. 226–253.
- [SCHO20a] S. Schorr, M. Möller, J. Heib und D. Bähre. „In-process Quality Control of Drilled and Reamed Bores using NC-Internal Signals and Machine Learning Method“. In: *Procedia CIRP* 93.13 (2020), S. 1328–1333.
- [SCHO20b] S. Schorr, M. Möller, J. Heib und D. Bähre. „Quality Prediction of Drilled and Reamed Bores Based on Torque Measurements and the Machine Learning Method of Random Forest“. In: *Procedia Manufacturing* 48 (2020), S. 894–901.
- [SCHO20c] S. Schorr, M. Möller, J. Heib, S. Fang und D. Bähre. „Quality Prediction of Reamed Bores Based on Process Data and Machine Learning Algorithm: A Contribution to a More Sustainable Manufacturing“. In: *Procedia Manufacturing* 43 (2020), S. 519–526.
- [SCHO21a] S. Schorr. „Prozessparallele Prognose der Werkstückqualität mithilfe von NC-internen Daten und maschinellem Lernen“. Dissertation. Universität des Saarlandes, 2021.
- [SCHO21b] S. Schorr, M. Möller, J. Heib und D. Bähre. „Comparison of Machine Learning Methods for Quality Prediction of Drilled and Reamed Bores Based on NC-Internal Signals“. In: *Procedia CIRP* 101 (2021). 9th CIRP Conference on High Performance Cutting, S. 77–80.
- [SCHU19] G. Schuh, P. Scholz, S. Schorr, D. Harman, M. Möller, J. Heib und D. Bähre. „Prediction of Workpiece Quality: An Application of Machine Learning in Manufacturing Industry“. In: *6th International Conference on Computer Science, Engineering*

*and Information Technology (CSEIT-2019)*. Aircc Publishing Corporation, 2019, S. 189–202.

- [SCHU22] G. Schuh, V. Zeller und V. Stich, Hrsg. Digitalisierungs- und Informationsmanagement. Bd. 9. Berlin und Heidelberg: Springer Vieweg, 2022.
- [SCIE22] ScienceDirect. Feature Extraction. 2022. URL: <https://www.sciencedirect.com/topics/engineering/feature-extraction> (Zuletzt geprüft am: 22.01.2022).
- [SEIF18] I. Seifert, M. Bürger, L. Wangler, S. Christmann-Budian, M. Rohde, P. Gabriel und G. Zinke. Potenziale der künstlichen Intelligenz im produzierenden Gewerbe in Deutschland. 2018. URL: [https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/potenziale-kuenstlichen-intelligenz-im-produzierenden-gewerbe-in-deutschland.pdf?\\_\\_blob=publicationFile&v=8](https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/potenziale-kuenstlichen-intelligenz-im-produzierenden-gewerbe-in-deutschland.pdf?__blob=publicationFile&v=8) (Zuletzt geprüft am: 14.01.2022).
- [SEKL12] P. Sekler. „Modellbasierte Berechnung der Systemeigenschaften von Maschinenstrukturen auf der Steuerung“. Dissertation. Universität Stuttgart, 2012.
- [SENI08] P. Senin. „Dynamic time warping algorithm review“. In: *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855.1-23 (2008), S. 40.
- [SHAF05] Y. Shafranovich. RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files. 2005. URL: <https://datatracker.ietf.org/doc/html/rfc4180> (Zuletzt geprüft am: 02.05.2022).
- [SHAO11] H. Shao, X. Shi und L. Li. „Power signal separation in milling process based on wavelet transform and independent component analysis“. In: *International Journal of Machine Tools and Manufacture* 51.9 (2011), S. 701–710.
- [SIEM06] Siemens AG. SINUMERIK 840D sl SINAMICS S20 CNC Teil1 - Inbetriebnahmehandbuch. 2006.
- [SIEM17] Siemens AG. SINUMERIK 840D sl Inbetriebnahme CNC: NC, PLC, Antrieb - Inbetriebnahmehandbuch. 2017.
- [SIEM18a] Siemens AG. SINUMERIK 840D sl / 828D Erweiterungsfunktionen - Funktionshandbuch. 2018.
- [SIEM18b] Siemens AG. SINUMERIK 840D sl Grundlagen - Programmierhandbuch. 2018.

- 
- [SIEM18c] Siemens AG. SINUMERIK 840D sl NC-Variablen und Nahtstellensignale - Listenhandbuch. 2018.
- [SIEM18d] Siemens AG. SINUMERIK 840D sl Sonderfunktionen - Funktionshandbuch. 2018.
- [SIEM18e] Siemens AG. SINUMERIK 840D sl, 828D Arbeitsvorbereitung - Programmierhandbuch. 2018.
- [SIEM18f] Siemens AG. SINUMERIK 840D sl, 828D Synchronaktionen - Funktionshandbuch. 2018.
- [SIEM18g] Siemens AG. SINUMERIK 840D sl, SINAMICS S120 Maschinendaten und Parameter - Listenhandbuch. 2018.
- [SIEM19a] Siemens AG. SINUMERIK 840D sl, Basisfunktionen - Funktionshandbuch. 2019.
- [SIEM19b] Siemens AG. SINUMERIK Analyze MyWorkpiece /Capture - Bedienhandbuch. 2019.
- [SIEM20] Siemens AG. SINUMERIK 840D sl, SINAMICS S120 Maschinendaten und Parameter - Listenhandbuch. 2020.
- [SIEM22a] Siemens AG. Industrial Edge for Machine Tools, a Siemens Edge Computing Platform for the Machine Tool Domain. 2022. URL: <https://documentation.mindsphere.io/resources/html/manage-my-sinumerik-edge-app-management/en-US/user-docu/industrialedge.html> (Zuletzt geprüft am: 01.03.2022).
- [SIEM22b] Siemens AG. Sinumerik Adapter - Developer Manual. 2022. URL: <https://documentation.mindsphere.io/resources/html/manage-my-sinumerik-edge-app-publishing/en-US/developer-docu/sinumerikadapter.html#Databus%20message%20structure%20of%20data%20packages> (Zuletzt geprüft am: 03.03.2022).
- [SING20] A. Singh. Feature Selection For Dimensionality Reduction (Embedded Method). 2020. URL: <https://medium.com/analytics-vidhya/feature-selection-for-dimensionality-reduction-embedded-method-e05c74014aa> (Zuletzt geprüft am: 25.07.2022).
- [SLAB99] G. G. Slabaugh. „Computing Euler angles from a rotation matrix“. In: *Retrieved on August 6.2000* (1999), S. 39–63.

- [SONN22] D. Sonnet. *Neuronale Netze kompakt*. Wiesbaden: Springer, 2022.
- [STAT22] Statistisches Bundesamt. *Industriesektor in Deutschland weiterhin stark: Bruttowertschöpfung Verarbeitendes Gewerbe*. 2022. URL: <https://www.destatis.de/Europa/DE/Thema/Industrie-Handel-Dienstleistungen/Industrie.html> (Zuletzt geprüft am: 30.10.2022).
- [SUSA18] A. Susanto, C.-H. Liu, K. Yamada, Y.-R. Hwang, R. Tanaka und K. Sekiya. „Application of Hilbert–Huang transform for vibration signal analysis in end-milling“. In: *Precision Engineering* 53 (2018), S. 263–277.
- [SUWA99] I. Suwalski. „Steuerungintegriertes Prozeßüberwachungssystem für Drehmaschinen“. Dissertation. Technische Universität Berlin, 1999.
- [TANG14] J. Tang, S. Alelyani und H. Liu. „Feature selection for classification: A review“. In: *Data Classification: Algorithms and Applications* (2014), S. 37–64.
- [TETI10] R. Teti, K. Jemielniak, G. O’Donnell und D. Dornfeld. „Advanced monitoring of machining operations“. In: *CIRP Annals* 59.2 (2010), S. 717–739.
- [TIME21] TIMET Savoie S.A. *Approved Certificate TIMETAL 6-4*. 2021. (Zuletzt geprüft am: 28.06.2022).
- [TING10] K. M. Ting. „Confusion Matrix“. In: *Encyclopedia of Machine Learning*. Hrsg. von C. Sammut und G. I. Webb. Boston: Springer US, 2010, S. 209–209.
- [TÖNS02] H. K. Tönshoff, T. Friemuth und J. C. Becker. „Process Monitoring in Grinding“. In: *CIRP Annals* 51.2 (2002), S. 551–571.
- [TÖNS88] H. K. Tönshoff, J. P. Wulfsberg, H. Kals, W. König und C. A. van Luttervelt. „Developments and Trends in Monitoring and Control of Machining Processes“. In: *CIRP Annals* 37.2 (1988), S. 611–622.
- [TRIP20] S. Tripathi, D. Muhr, B. Manuel, F. Emmert-Streib, H. Jodlbauer und M. Dehmer. *Ensuring the Robustness and Reliability of Data-Driven Knowledge Discovery Models in Production and Manufacturing*. 2020.

- 
- [TSAI14] C.-W. Tsai, C.-F. Lai, M.-C. Chiang und L. T. Yang. „Data Mining for Internet of Things: A Survey“. In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), S. 77–97.
- [TURI50] A. M. Turing. „I.-Computing Machinery and Intelligence“. In: *Mind* LIX.236 (1950), S. 433–460.
- [ULSO06] A. G. Ulsoy. „Monitoring and Control of Machining“. In: *Condition Monitoring and Control for Intelligent Manufacturing*. Hrsg. von L. Wang, R. X. Gao und D. T. Pham. Bd. 119. London: Springer London, 2006, S. 1–32.
- [VASA20] X. A. Vasanth, P. S. Paul und A. S. Varadarajan. „A neural network model to predict surface roughness during turning of hardened SS410 steel“. In: *International Journal of System Assurance Engineering and Management* 11.3 (2020), S. 704–715.
- [VICT56] H. Victor. „Beitrag zur Kenntnis der Schnittkräfte beim Drehen, Hobeln und Bohren“. Dissertation. Leibniz Universität Hannover, 1956.
- [WAHL17] W. Wahlster. Künstliche Intelligenz versus menschliche Intelligenz: Wie lernen, verstehen und denken Computer? Vorlesungsreihe: Künstliche Intelligenz für den Menschen Digitalisierung mit Verstand. Mainz, 2017. URL: [http://www.dfki.de/wwdata/Gutenberg\\_Stiftungsprofessur\\_Mainz\\_2017/Lernende\\_Maschinen.pdf](http://www.dfki.de/wwdata/Gutenberg_Stiftungsprofessur_Mainz_2017/Lernende_Maschinen.pdf) (Zuletzt geprüft am: 05.12.2022).
- [WAHL20] W. Wahlster und C. Winterhalter. Deutsche Normungsroadmap Künstliche Intelligenz. Berlin und Frankfurt, 2020.
- [WANG03] M.-Y. Wang und H.-Y. Chang. „A simulation shape error for end milling AL6061-T6“. In: *The International Journal of Advanced Manufacturing Technology* 22.9-10 (2003), S. 689–696.
- [WANG15] Z. Wang und T. Oates. „Imaging Time-Series to Improve Classification and Imputation“. In: *CoRR* (2015).
- [WANG18] J. Wang, Y. Ma, L. Zhang, R. X. Gao und D. Wu. „Deep learning for smart manufacturing: Methods and applications“. In: *Journal of Manufacturing Systems* 48 (2018). Special Issue on Smart Manufacturing, S. 144–156.
- [WECK06] M. Weck und C. Brecher. Werkzeugmaschinen 4: Automatisierung von Maschinen und Anlagen. 6. Auflage. Berlin und Heidelberg: Springer Vieweg, 2006.

- [WECK12] A. Weckenmann, Hrsg. Koordinatenmesstechnik: Flexible Strategien für funktions- und fertigungsgerechtes Prüfen. 2., vollst. überarb. Auflage. München: Hanser, 2012.
- [WECK93] M. Weck, T. Pfeifer, W. Eversheim und W. König. „Die offene Steuerung - Zentraler Baustein leistungsfähiger Produktionsanlagen.“ In: *Wettbewerbsfaktor Produktionstechnik*. Hrsg. von W. Eversheim, W. König, T. Pfeifer und M. Weck. Düsseldorf: VDI-Verlag, 1993.
- [WELC67] P. Welch. „The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms“. In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), S. 70–73.
- [WELL19] F. Wellmann. „Datengetriebene, kontextadaptive Produktivitätssteigerung von NC-Zerspanprozessen“. Dissertation. RWTH Aachen, 2019.
- [WICH22] A. Wichert. Künstliche Intelligenz. 2022. URL: <https://www.spektrum.de/lexikon/neurowissenschaft/kuenstliche-intelligenz/6810> (Zuletzt geprüft am: 11. 01. 2022).
- [WOLF09] J. Wolf. „Steuerungintegrierte, adaptive Programmausführung einer aufgabenorientierten Programmierung in STEP-NC“. Dissertation. RWTH Aachen, 2009.
- [WOLP02] D. H. Wolpert. „The Supervised Learning No-Free-Lunch Theorems“. In: *Soft Computing and Industry*. Hrsg. von R. Roy, M. Köppen, S. Ovaska, T. Furuhashi und F. Hoffmann. Springer London, 2002, S. 25–42.
- [WOLP96] D. Wolpert. „The Lack of A Priori Distinctions Between Learning Algorithms“. In: *Neural Computation* 8 (März 1996).
- [WOLP97] D. H. Wolpert und W. G. Macready. „No free lunch theorems for optimization“. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), S. 67–82.
- [WROB13] S. Wrobel, T. Joachims und K. Morik. „12 Maschinelles Lernen und Data Mining“. In: *Handbuch der Künstlichen Intelligenz*. Hrsg. von G. Görz, J. Schneeberger und U. Schmid. München: Oldenbourg Wissenschaftsverlag, 2013.
- [YADA18] P. Yadav. Decision Tree in Machine Learning. 2018. URL: <https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96> (Zuletzt geprüft am: 05. 08. 2022).

- 
- [YOHA13] B. Yohannes. „Industrielle Prozessüberwachung für die Kleinserienfertigung“. Dissertation. Leibniz Universität Hannover, 2013.
- [ZEBA20] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari und J. Saeed. „A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction“. In: *Journal of Applied Science and Technology Trends* 1.2 (2020), S. 56–70.
- [ZHAN04] H. Zhang. „The Optimality of Naive Bayes“. In: Bd. 2. Jan. 2004.
- [ZHAN16] C. Zhang, X. Yao, J. Zhang und H. Jin. „Tool Condition Monitoring and Remaining Useful Life Prognostic Based on a Wireless Sensor in Dry Milling Operations“. In: *Sensors (Basel, Switzerland)* 16.6 (2016).
- [ZHU09] K. Zhu, Y. S. Wong und G. S. Hong. „Wavelet analysis of sensor signals for tool condition monitoring: A review and some new results“. In: *International Journal of Machine Tools and Manufacture* 49.7-8 (2009), S. 537–553.
- [ZIEG20] A. Ziegenbein, A. Fertig, J. Metternich und M. Weigold. „Data-based process analysis in machining production: Case study for quality determination in a drilling process“. In: *Procedia CIRP* 93 (2020). 53rd CIRP Conference on Manufacturing Systems 2020, S. 1472–1477.
- [ZIEG22] A. Ziegler und K. A. Heller. Intelligenz. 2022. URL: <https://www.spektrum.de/lexikon/psychologie/intelligenz/7263> (Zuletzt geprüft am: 10.01.2022).
- [ZOU05] H. Zou und T. Hastie. „Regularization and Variable Selection via the Elastic Net“. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67.2 (2005), S. 301–320. (Zuletzt geprüft am: 27.07.2022).

## Normen und Richtlinien

- [DIN4760] DIN 4760:1982-6. Gestaltabweichungen; Begriffe, Ordnungssystem.
- [DIN66025-1] DIN 66025-1:1983-1. Programmaufbau für numerisch gesteuerte Arbeitsmaschinen.
- [DIN66215-1] DIN 66215-1:1974-8. Programmierung numerisch gesteuerter Arbeitsmaschinen durch CLDATA: Allgemeiner Aufbau und Satztypen.
- [DIN69651] DIN 69651:1981-3. Werkzeugmaschinen für die Metallbearbeitung. Teil 1: Begriffe.
- [DIN8580] DIN 8580:2020-1. Fertigungsverfahren - Begriffe, Einteilung.
- [EN61499-1] DIN EN 61499-1:2014-9. Funktionsbausteine für industrielle Leitsysteme – Teil 1: Architektur.
- [ISO1101] DIN EN ISO 1101:2017-10. Geometrische Produktspezifikation (GPS) - Geometrische Tolerierung - Tolerierung von Form, Richtung, Ort und Lauf.
- [ISO2806] DIN ISO 2806:1996-4. Industrielle Automatisierungssysteme - Numerische Steuerung von Maschinen - Begriffe.
- [ISO5458] DIN EN ISO 5458:2018-11. Geometrische Produktspezifikationen (GPS) - Geometrische Tolerierung - Elementgruppen und kombinierte geometrische Spezifikation.
- [ISO8000-2] ISO 8000-2:2020-6. Data quality: Part2: Vocabulary.
- [ISO9000] DIN EN ISO 9000:2015-11. Qualitätsmanagementsysteme - Grundlagen und Begriffe.

## Studentische Arbeiten

- [CHEN21] Y. Chen. „Erarbeitung und Erprobung verschiedener Ansätze zur Vorhersage der Werkstückqualität in der Zerspaltung auf Basis von künstlicher Intelligenz“. Masterthesis. TU Darmstadt, 2021.
- [OSTE21] P. Ostendarp. „Entwicklung eines Modells zur Vorhersage der Werkstückqualität in der Zerspaltung auf Basis von künstlicher Intelligenz“. Masterthesis. TU Darmstadt, 2021.

- [PREI22] C. Preis. „Erarbeitung und Konzeptionierung einer Methode zur systematischen Entwicklung von Machine Learning Pipelines für die Vorhersage der Werkstückqualität in der Zerspanung“. Masterthesis. TU Darmstadt, 2022.
- [ROSS21] T. Rosswinkel. „Entwicklung einer Methode zur Übertragbarkeit von KI-basierten Modellen zwischen Werkzeugmaschinen“. Masterthesis. TU Darmstadt, 2021.
- [WÜNS22] W. Wünschel. „Untersuchung der Robustheit von Machine Learning Modellen für die Vorhersage der Werkstückqualität in der Zerspanung“. Masterthesis. TU Darmstadt, 2022.



# A Anhang

## A.1 Spezifikationen des verwendeten IPC für die Datenerfassung

Tabelle A.1: Technische Daten SIMATIC IPC227E.

Bezeichnung	Wert
Produktbezeichnung	SIMATIC IPC227E
Prozessor	Intel Celeron N2930 (4C/4T) mit TPM
Laufwerk	SSD mit 240 GB
Hauptspeicher	8 GB DDR3L
Versorgungsspannung	DC 24 V
Industrial-Ethernet	2x 10/100/1000 Mbit/s Ethernet RJ45
Schnittstellen	1x USB 3.0 / 3x USB 2.0; CFast slot
Artikelnummer	6ES7647-8BD31-0CA1

## A.2 Anwenderzyklus CYCPE1US.SPF

Anwenderzyklus für das Zurücksetzen der Kontextvariablen bei Programm Ende und Bedienertafel Reset, angelehnt an [INDU14].

Systempfad: „NC/CUS.DIR/CYCPE1US.SPF“

```
 ; #####  
PROC CYCPE1US  
 ; Programm Ende           Bedienertafel Reset  
IF ($P_PROG_EVENT == 2) OR ($P_PROG_EVENT == 3)  
  SYG_I9[0] = -1           ; Seriennummer  
  SYG_S9[0] = "-1"        ; Materialnummer  
  SYG_S9[1] = "-1"        ; geomElemID  
ENDIF  
M17  
 ; #####
```

### A.3 Berechnete Merkmale für die Merkmalsextraktion

Tabelle A.2: Formeln zur Berechnung der implementierten statistischen Merkmale [BARA20; KOKO00; OLIP06].

Statistische Merkmale	
Merkmal	Formel
arithmetischer Mittelwert	$\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i = \text{mean}(X)$
quadratischer Mittelwert	$rms = \sqrt{\frac{\sum_{i=0}^n x_i^2}{n}}$
Median	$\tilde{x} = \text{median}(X)$
Standardabweichung	$std = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$
Schiefe	$v = \frac{1}{n} \sum_{i=0}^n \left(\frac{x_i - \bar{x}}{std}\right)^3 = \text{skew}(X)$
Kurtosis	$w = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{STD}\right)^4 = \text{kurt}(X)$
Maximum	$max = \text{max}(X)$
Minimum	$min = \text{min}(X)$
mittlere absolute Abweichung	$\tilde{d}_{0,5} = \frac{1}{n} \sum_{i=0}^n  x_i - \tilde{x} $
Median Abweichung	$\tilde{d}_{med} = \text{median}( x_i - \tilde{x} )$

Tabelle A.3: Formeln zur Berechnung der implementierten Merkmale im Zeitbereich [BARA20].

Zeitbereich (temporal)	
Merkmalsname	Formel
Signalenergie	$E = \sum_{i=0}^n x_i^2$
Flächeninhalt unter der Kurve	$AUC = \sum_{i=0}^n (t_i - t_{i-1}) \cdot \frac{x_i - x_{i-1}}{2}$
Autokorrelation	$c_{auto} = \sum_{n \in \mathbb{Z}} x(n) \cdot \overline{x(n-l)},$ mit $\overline{x(n-l)}$ als konjugiert komplexe Funktion von $x_i$ und $l$ als Verschiebungsfaktor
Schwerpunkt (engl.: centroid)	$C = \frac{\sum_{i=0}^n t_i \cdot x_i^2}{\sum_{i=0}^n x_i^2}$
(Shannon-)Entropie	$H(X) = - \sum_{i=0}^n p(x_i) \log_2 p(x_i)$
Mittelwert absolute Differenz (Mean abs diff)	$MAD = \frac{1}{n} \sum_{i=0}^n  x_{i+1} - x_i $
Mittelwert Differenz	$MD = \frac{1}{n} \sum_{i=0}^n x_{i+1} - x_i$
Median absolute Differenz	$medAD = median( x_{i+1} - x_i )$
Median diff	$medD = median(x_{i+1} - x_i)$
Peak-to-Peak-Distanz	$p2p =  max(X) - min(X) $
Distanz	$dist = \sum_{i=0}^{n-1} \sqrt{1 + (x_{i+1} - x_i)^2}$
Steigung (engl.: slope)	Anpassung der linearen Gleichung $y = mx + t$ an die Daten, Rückgabe der Steigung $m$
Summe absolute Differenzen	$sumAD = \sum_{i=0}^{n-1}  x_{i+1} - x_i $
Totale Energie	$TE = \frac{\sum_{i=0}^n x_i^2}{t_n - t_0}$
Scheitelfaktor / Crest-Faktor	$\frac{max(X)}{rms(X)}$

Für die Berechnung der Merkmale im Frequenzbereich wird das Signal zunächst mittels FFT transformiert. Basierend auf dem resultierenden Signal im Frequenzbereich  $f_{\text{Sig}}$ , erfolgt die Berechnung der in Tabelle A.4 zusammengefassten Merkmale. Die Spektrale Leistungsdichte (engl.: power spectrum density)  $PSD_{\text{Sig}}$  wird basierend auf [WELC67] berechnet. Aufgrund der teilweisen komplexen Berechnungsvorschrift wird auf die aufgeführten Quellen verwiesen.

Tabelle A.4: Quellen und Formeln zur Berechnung der implementierten Merkmale im Frequenzbereich.

Frequenzbereich (spectral)	
Merkmal	Quelle / Formel
Grundfrequenz (engl.: fundamental frequency)	[BARA20]
Amplitude der Grundfrequenz	[BARA20]
Linear prediction cepstral coefficients (LPCC)	[DAS16]
Mel-Frequenz-Cepstrum-Koeffizienten (MFCC)	[DAS16]
Maximum PSD	$\max(PSD_{\text{Sig}})$
Maximumfrequenz	[BARA20]
Medianfrequenz	[BARA20]
Leistungsbandbreite	[HENR03]
Schwerpunktwellenlänge (engl.: spectral centroid)	[PEET11]
Spektrale Abnahme (engl.: spectral decrease)	[PEET11]
Spektrale Distanz (engl.: spectral distance)	[BARA20]
Spektrale Entropie (engl.: spectral entropy)	[PAN09]
Spektrale Steigung (engl.: spectral slope)	[PEET11]
Spektrale Schiefe (engl.: spectral skewness)	[PEET11]
Spektrale Kurtosis (engl.: spectral kurtosis)	[PEET11]
Spektrale Standardabweichung (engl.: spectral spread)	[PEET11]
Spektraler Fluss (engl.: spectral variation)	[PEET11]
Mittelwert Frequenzspektrum	$\text{mean}(f_{\text{Sig}})$
Kurtosis PSD	$\text{kurt}(PSD_{\text{Sig}})$
Mittelwert PSD	$\text{mean}(PSD_{\text{Sig}})$
Schiefe PSD	$\text{skew}(PSD_{\text{Sig}})$

## A.4 Erweiterung TSFEL

Erweiterungsmodul `own_features.py` für TSFEL. Enthält neben der Implementierung für DWT und HHT zusätzlich genutzte Merkmale für die Domänen `temporal` und `spectral`.

```

from tsfel.feature_extraction.features_utils import *
import pywt
import tsfel
import numpy
import emd

@set_domain("domain", "temporal")
def crest_factor(signal):
    """Computes the crest factor of the signal.
    Parameters
    -----
    signal:
        The time series to calculate the feature of.
    Returns
    -----
    float
        crest factor
    """

    feature = np.max(signal) / (np.sqrt(np.sum(np.array(signal)
        ** 2) / len(signal)))

    return feature

@set_domain("domain", "spectral")
def mean_frequency(signal, fs):
    """Computes mean frequency of the signal.
    Feature computational cost: 2
    Parameters
    -----
    signal : nd-array
        Input from which MEAN frequency is computed
    fs : int
        Sampling frequency
    Returns
    -----
    float
        mean frequency using cumsum
    """

    f, fmag = calc_fft(signal, fs)

    return np.mean(f)

```

```

@set_domain("domain", "spectral")
def mean_PSD(signal, fs):
    """Computes mean of Power Spectrum Density via
    Welch's method.
    Feature computational cost: 2
    Parameters
    -----
    signal : nd-array
        Input from which PSD is computed
    fs : int
        Sampling frequency
    Returns
    -----
    float
        mean of Power Spectrum Density
    """

    freq, psd = welch(signal, fs=fs)

    return np.mean(psd)

@set_domain("domain", "spectral")
def kurtosis_PSD(signal, fs):
    """Computes kurtosis of Power Spectrum Density via
    Welch's method.
    Feature computational cost: 2
    Parameters
    -----
    signal : nd-array
        Input from which PSD is computed
    fs : int
        Sampling frequency
    Returns
    -----
    float
        Kurtosis of Power Spectrum Density
    """

    freq, psd = welch(signal, fs=fs)

    return scipy.stats.kurtosis(psd)

@set_domain("domain", "spectral")
def skewness_PSD(signal, fs):
    """Computes skewness of Power Spectrum Density via
    Welch's method.
    Feature computational cost: 2
    Parameters
    -----
    signal : nd-array

```

```

        Input from which PSD is computed
    fs : int
        Sampling frequency
    Returns
    -----
    float
        Skewness of Power Spectrum Density
    """

    freq, psd = welch(signal, fs=fs)

    return scipy.stats.skew(psd)

@set_domain("domain", "spectral")
def amplitude_spectral_peak(signal, fs):
    """Computes actual signal amplitude of spectral peak.
    fft peak=A*N/2
    A is the signal amplitude, N is the number
    of sampling points
    Feature computational cost: 2
    Parameters
    -----
    signal : nd-array
        Input from which PSD is computed
    fs : int
        Sampling frequency
    Returns
    -----
    float
        actual signal amplitude of spectral peak
    """

    complex_array = fft(np.array(signal))
    peak = abs(complex_array[np.argmax(complex_array)])

    return peak * 2 / len(signal)

@set_domain("domain", "spectral")
def mean_frequency(signal, fs):
    """Computes mean frequency of the signal.
    Feature computational cost: 2
    Parameters
    -----
    signal : nd-array
        Input from which MEAN frequency is computed
    fs : int
        Sampling frequency
    Returns
    -----
    float

```

```

    """ mean frequency using cumsum
    """
    f, fmag = calc_fft(signal, fs)

    return np.mean(f)

@set_domain("domain", "temporal-spectral")
def dwt(signal, dwt_level, waveletname):
    """Computes the dwt features of the signals dwt-coefficients.
    Parameters
    -----
    signal:
        The time series to calculate the feature of.

    Returns
    -----
    list
        features of dwt-coefficients for each decomposition level
    """
    list_coeff = pywt.wavedec(signal, waveletname, level=dwt_level)
    coeff_features_list = []
    for i in range(len(list_coeff)):
        mean = tsfel.calc_mean(list_coeff[i])
        rms = tsfel.rms(list_coeff[i])
        std = tsfel.calc_std(list_coeff[i])
        kurtosis = tsfel.kurtosis(list_coeff[i])
        crestFactor = crest_factor(list_coeff[i])
        pk_pk_dist = tsfel.pk_pk_distance(list_coeff[i])
        coeff_feat = numpy.array([mean, rms, std,
                                kurtosis, crestFactor, pk_pk_dist])
        coeff_features_list.append(coeff_feat)

    return coeff_features_list

def hht(signal, fs, imfs_max, min_frequency, max_frequency,
        count_frequency_bins):
    """Computes Hilbert-Spectrum of the signal using the
    Hilbert-Huang-Transformation
    Parameters
    -----
    signal : nd-array
        Input from which HHT is computed
    fs : int
        Sampling frequency
    max_imfs: int
        Maximum number of IMFs that will be calculated

    min_frequency: int
        minimum frequency fot Hilbert-Transform

```

```

max_frequency: int
    max frequency for Hilbert-Transform
count_frequency_bins: int
    number of frequency bins, which will be examined
    between min and max frequency

Returns
-----
features of hht-features
"""
sample_rate=fs

imf = emd.sift.mask_sift(signal, max_imfs=imfs_max)
IMF_Features_List = []

# Carrier frequency histogram definition
freq_edges, freq_centres
    = emd.spectra.define_hist_bins(min_frequency,
    max_frequency, count_frequency_bins, 'linear')
# Compute frequency statistics
IP, IF, IA
    = emd.spectra.frequency_transform(imf, sample_rate,
    'hilbert')
f, hht
    = emd.spectra.hilberthuang(IF, IA, freq_edges,
    mode='amplitude', sum_time=False)
mean = tsfel.calc_mean(hht)
rms = tsfel.rms(hht)
std = tsfel.calc_std(hht)
crestFactor = crest_factor(hht)
pk_pk_dist = tsfel.pk_pk_distance(hht)
abs_en = tsfel.abs_energy(hht)
IMF_Features = numpy.array([mean, rms, std, crestFactor,
    pk_pk_dist, abs_en])
IMF_Features_List.append(IMF_Features)
return IMF_Features_List

```

## A.5 Hyperparameterraster für die jeweiligen Algorithmen

Suchbereich der Hyperparameter für die Gittersuche je genutztem Algorithmus als Auszug aus der `PipelineConfig.json`. Der folgende Code Abschnitt zeigt das Objekt `CL_algorithms` aus dem übergeordneten Objekt `MachineLearningConfig`, welches neben den zu nutzenden Algorithmen den Suchbereich für die jeweiligen Hyperparameter parametrisiert.

```

"CL_algorithms" : {
  "SVM" : {
    "svm__kernel" : [ "rbf", "poly", "sigmoid" ],
    "svm__C" : [ 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,
                  0.9, 1 ],
    "svm__gamma" : [ "auto", 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,
                     0.7, 0.8, 0.9, 1 ]
  },
  "KNN" : {
    "knn__n_neighbors" : [ 3, 4, 5, 6, 7, 8 ]
  },
  "RidgeRe" : {
    "ridge__alpha" : [ 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,
                      0.7, 0.8, 0.9, 1 ]
  },
  "GNB" : {},
  "DT" : {
    "dt__max_depth" : [ 3, 4, 5, 6, 8, 10 ]
  },
  "RF" : {
    "rfc__max_features" : [ "auto", "sqrt", "log2", 2, 3,
                             4, 5 ],
    "rfc__n_estimators" : [ 10, 20, 30, 40, 50, 100 ],
    "rfc__criterion" : [ "gini", "entropy", "log_loss" ]
  },
  "XT" : { "xtc__max_features" : [ "auto", "sqrt", "log2", 2, 3,
                                   4, 5 ],
           "xtc__n_estimators" : [ 10, 20, 30, 40, 50, 100 ],
           "xtc__criterion" : [ "gini", "entropy", "log_loss" ]
  },
  "AdaBoost" : {
    "ada__base_estimator" : [ "DecisionTreeClassifier",
                              "ExtraTreesClassifier" ],
    "ada__n_estimators" : [ 50, 100, 150 ]
  },
  "MLP" : {
    "MLP__hidden_layer_sizes" : [[ 10, 10, 10 ], [ 10, 10,
                                                       10 ], [ 50, 50 ], [ 50, 50, 50 ] ],
    "MLP__activation" : [ "tanh", "relu" ],
    "MLP__solver" : [ "adam" ],
    "MLP__max_iter" : [ 10000 ]
  }
}

```

## A.6 Beispielhafte PipelineConfig.json

Die beispielhaft dargestellte PipelineConfig.json dient der initialen Parametrierung der implementierten MLPK aus Kapitel 5.3.6. Die „...“ dienen in diesem Fall als Platzhalter für die vereinfachte Darstellung.

```
{
  "Initialization": {
    "MachineToolID": "DMC850V",
    "configDir": "/config",
    "DataSetName": "Dataset",
    "FeatSelExperimentName": "Features",
    "MLExperimentName": "MLExperiment"
  },
  "FeatEngConfig": {
    "ModePipeline": {"FeatExtraction":1,
                    "DataSetBuilding":1,
                    "FeatSelection":1,
                    "FinalFeatSelection":1
    },
    "dirpath": "/data/sliced",
    "saveFormat": "csv",
    "dstDir": "/data/FeatEng",
    "hf_cycle_time": 2,
    "domain": "None",
    "window_func": "None",
    "individual_geomELEM_features": "yes",
    "Use_IMF_Full_Auto_Experiment": "no",
    "signals_to_use": [...],
    "FeatSel__methods": ["uniStat", "LogisRe"],
    "params_FeatureSelection": {
      "feat_scaling": "StandardScaler",
      "uniStat__scorefunc": "mutual_info_classif",
      "LogisRe__penalty": "elasticnet"
    },
    "finalFeatSel__methods": ["uniStat", "LogisRe"],
    "finalFeatSel__modes": ["perElement", "commFeat"],
    "value_counting_tresholds": {
      "perElement": 2,
      "commonFeat": 5
    },
    "geomElements": [...],
    "geomElementsGroups": {
    }
  },
}
```

```
"DataSetConfig" : {
  "geomElements" : [...],
  "dirpaths" : {
    "X" : "/data/FeatEng/Features",
    "y" : "/data/CMM"
  },
  "dstDir" : "/data/MLDatasets",
  "train_size" : 0.75
},
"MachineLearningConfig" : {
  "dstDir" : "/resultsML",
  "Scaling__method" : "StandardScaler",
  "CL_algorithms" : {...}
}
}
```

## A.7 Form- und Lagetoleranzen am Musterwerkstück

Tabelle A.5: Ausgewertete Form- und Lagetoleranzen am Musterwerkstück (1 / 2). In runden Klammern gesetzte Achsen entsprechen der Bezugsdefinition.

geomELEMENT	TOLNAME	AXIS	NOMVAL	UPPTOLLIM	LOWTOLLIM
LINE_1	Geradheit	-	0,000	0,006	0,000
LINE_1	Ebenheit	-	0,000	0,015	0,000
LINE_1	Parallelität	(X)	0,000	0,008	0,000
LINE_1	Rechtwinkligkeit	(Y)	0,000	0,008	0,000
LINE_2	2D_Winkel	A	293,0183	0,025	-0,005
INNERSPLINE	Lage_PT0	Y	-13,4128	0,000	-0,070
INNERSPLINE	Lage_PT1	Y	-10,9566	0,000	-0,120
INNERSPLINE	Lage_PT2	Y	-10,9715	0,000	-0,120
INNERSPLINE	Lage_PT3	Y	-12,9552	0,150	0,000
LINE_3	Geradheit	-	0,000	0,005	0,000
LINE_3	2D_Winkel	A	0,500	0,020	-0,050
LINE_4	Geradheit	-	0,000	0,005	0,000
LINE_4	Parallelität	(X)	0,000	0,008	0,000
OUTERSPLINE	Lage_PT0	Y	-10,3944	0,020	-0,015
OUTERSPLINE	Lage_PT1	Y	-8,0153	0,000	-0,100
OUTERSPLINE	Lage_PT2	Y	-7,9694	0,010	-0,040
OUTERSPLINE	Lage_PT3	Y	-9,7787	0,200	0,000
LINE_5	Geradheit	-	0,000	0,005	0,000
LINE_5	Parallelität	(X)	0,000	0,008	0,000
LINE_6	Geradheit	-	0,000	0,005	0,000
LINE_6	Rechtwinkligkeit	(C)	0,000	0,008	0,000
OUTERCIRCLE	Durchmesser	-	16,000	0,050	0,000
OUTERCIRCLE	Rundheit	-	0,000	0,005	0,000

Tabelle A.6: Ausgewertete Form- und Lagetoleranzen am Musterwerkstück (2 / 2). In runden Klammern gesetzte Achsen entsprechen der Bezugsdefinition.

geomELEMENT	TOLNAME	AXIS	NOMVAL	UPPTOLLIM	LOWTOLLIM
LINE_7	Geradheit	-	0,000	0,008	0,000
LINE_7	Ebenheit	-	0,000	0,018	0,000
LINE_7	Parallelität	(X)	0,000	0,020	0,000
LINE_7	Rechtwinkligkeit	(Y)	0,000	0,020	0,000
LINESTEP_8	Geradheit	-	0,000	0,005	0,000
LINESTEP_8	Parallelität	(Y)	0,000	0,008	0,000
LINESTEP_8	Lage	X	7,000	-0,010	-0,050
LINESTEP_6	Geradheit	-	0,000	0,005	0,000
LINESTEP_6	Parallelität	(Y)	0,000	0,008	0,000
LINESTEP_6	Lage	X	8,000	-0,010	-0,050
LINESTEP_4	Geradheit	-	0,000	0,005	0,000
LINESTEP_4	Parallelität	(Y)	0,000	0,008	0,000
LINESTEP_4	Lage	X	9,000	-0,010	-0,050
LINESTEP_2	Geradheit	-	0,000	0,005	0,000
LINESTEP_2	Parallelität	(Y)	0,000	0,008	0,000
LINESTEP_2	Lage	X	10,000	0,010	-0,030
INNERCIRCLE	Durchmesser	-	11,000	0,000	-0,060
INNERCIRCLE	Rundheit	-	0,000	0,030	0,000

## A.8 Manipulierte Durchhangskompensation

Folgende Abbildung stellt die manipulierten Werte für die Kompensation der Y-Achse bei Verfahrensbewegung der X-Achse dar. Die Manipulation erfolgt basierend auf den Ausführungen zur Durchhangskompensation gemäß [SIEM18a]. Dafür dienen 100 Stützstellen, welche jeweils einen Abstand von 1 mm ( $\$AN\_CEC\_STEP[0]=1$ ) aufweisen. Für die Konfiguration der Kompensation wurden die Basisachse X mit der Systemvariable  $\$AN\_CEC\_INPUT\_AXIS[0]=(AX1)$  und die Kompensationsachse Y mit  $\$AN\_CEC\_OUTPUT\_AXIS[0]=(AX2)$  definiert.

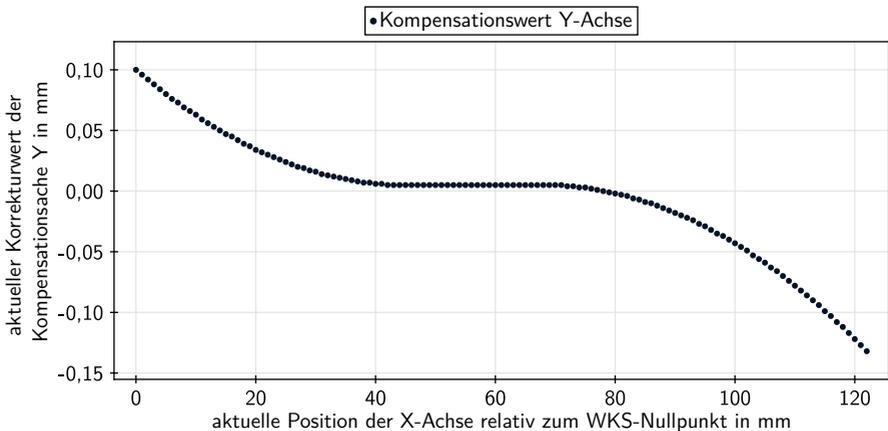


Abbildung A.1: Verlauf der manipulierten Durchhangskompensation für EC31 (vgl. Tabelle 5.4)

## A.9 G-Code Zeilen für zeitoptimiertes Beschleunigungsverhalten

Eingefügte G-Code Zeilen für die Manipulation des Beschleunigungsverhaltens und Deaktivierung der Vorsteuerung für EC44 (vgl. Tabelle 5.4). Die jeweilige Wirkung der Befehle auf die Bahnführung lässt sich [SIEM18b] entnehmen.

```
N10 BRISK
N20 VELOLIMA [AX1]=100
N30 VELOLIMA [AX2]=100
N40 ACCLIMA [AX1]=100
N50 ACCLIMA [AX2]=100
N60 JERKLIMA [AX1]=100
N70 JERKLIMA [AX2]=100
N80 FFWOF
N90 CPRECOF
```

## A.10 Vergleich der optimierten DWT-Hyperparameter zwischen den Datensätzen

Mittels Optimierungsalgorithmus identifizierte Basis-Wavelets und Dekompositionsebenen je Geometrieelement für die drei Datensätze  $DS_{DMG1}$ ,  $DS_{DMG2}$  und  $DS_{GROB}$ . Die Bezeichnungen folgen der Namensgebung aus dem Softwarepaket PyWavelets [LEE19].

Tabelle A.7: Identifizierte Basis-Wavelets und Dekompositionsebenen je Datensatz.

geomELEMENT	$DS_{DMG1}$		$DS_{DMG2}$		$DS_{GROB}$	
	Wavelet	Ebene	Wavelet	Ebene	Wavelet	Ebene
LINE_1	coif1	6	bior1.1	7	sym4	5
LINE_2	sym5	4	db9	6	bior3.1	6
INNERSPLINE	bior2.2	4	sym9	6	bior2.8	4
LINE_3	sym7	6	bior2.8	3	bior2.4	5
LINE_4	bior1.1	3	bior1.1	3	bior1.1	3
OUTERSPLINE	bior1.3	4	db6	3	sym6	5
LINE_5	bior1.1	3	bior1.1	3	bior1.1	3
LINE_6	coif5	6	bior1.5	5	db8	6
OUTERCIRCLE	bior1.1	3	bior3.3	5	bior1.1	3
LINE_7	bior1.1	3	bior1.1	3	sym8	6
LINESTEP_8	bior1.3	4	coif5	7	bior3.1	3
LINESTEP_6	sym8	4	sym4	3	bior1.5	5
LINESTEP_4	bior1.1	3	bior1.1	4	db8	7
LINESTEP_2	bior3.3	5	bior2.6	3	bior2.6	6
INNERCIRCLE	bior2.6	3	db8	5	sym4	4

## A.11 Form- und Lagetoleranzen „Tiefe Tasche“

Tabelle A.8: Ausgewertete Form- und Lagetoleranzen am Werkstück „Tiefe Tasche“. In runden Klammern gesetzte Achsen entsprechen der Bezugsdefinition.

geomELEMENT	TOLNAME	AXIS	NOMVAL	UPPTOLLIM	LOWTOLLIM
CIR_1	Radius	-	10,000	-0,015	-0,040
CIR_1	Rundheit	-	0,000	0.010	0,000
LIN_2	Geradheit	-	0,000	0,010	0,000
LIN_2	Parallelität	(Y)	0,000	0,010	0,000
LIN_2	2D_Winkel	Y	0,000	0,010	0,010
CIR_2	Radius	-	10,000	-0,015	-0,040
CIR_2	Rundheit	-	0,000	0.010	0,000
LIN_3	Geradheit	-	0,000	0,010	0,000
LIN_3	Parallelität	(X)	0,000	0,010	0,000
LIN_3	2D_Winkel	X	0,000	0,010	0,010
CIR_3	Radius	-	10,000	-0,015	-0,040
CIR_3	Rundheit	-	0,000	0.010	0,000
LIN_4	Geradheit	-	0,000	0,010	0,000
LIN_4	Parallelität	(Y)	0,000	0,010	0,000
LIN_4	2D_Winkel	Y	0,000	0,010	0,010
CIR_4	Radius	-	10,000	-0,015	-0,040
CIR_4	Rundheit	-	0,000	0.010	0,000

## A.12 Ausschnitte der Benutzeroberfläche des implementierten Softwareprototyps



Abbildung A.2: Initiale Benutzeroberfläche des Softwareprototyps.

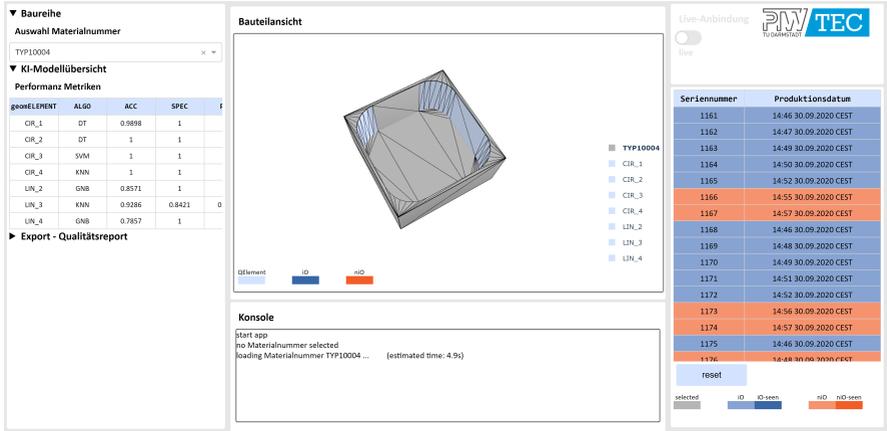


Abbildung A.3: Bauteil „TYP10004“ (Tiefe Tasche) - Qualitätsrelevante Flächen vor Prognose.



Abbildung A.4: Bauteil „TYP10004“ (Tiefe Tasche) - Qualitätsrelevante Flächen nach Prognose.

## Lebenslauf

### *Persönliche Daten*

Name: Alexander Joachim Fertig  
Geburtsdatum und -ort: 27. Dezember.1993 in Miltenberg  
Staatsangehörigkeit: deutsch

### *Berufstätigkeit*

01/2018 – dato: Wissenschaftlicher Mitarbeiter am Institut für Produktionsmanagement, Technologie und Werkzeugmaschinen (PTW) der Technischen Universität Darmstadt

### *Praktikum*

11/2016 – 04/2017 TRUMPF SE + Co. KG, Ditzingen

### *Studium*

10/2011 – 05/2018 Studium des allgemeinen Maschinenbaus, Technische Universität Darmstadt  
Abschluss: Master of Science

### *Schulbildung*

09/2003 – 07/2011 Johannes-Butzbach-Gymnasium Miltenberg  
Abschluss: Allgemeine Hochschulreife