

Entwicklung einer modularen datengetriebenen Optimierungskette für Fertigungsprozesse am Beispiel der Gießereiindustrie

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Maschinenbau und Verfahrenstechnik der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades

eines

Doktors der Ingenieurwissenschaften

Dr.-Ing.

genehmigte Dissertation

von

Felix Gandha Salentin

aus

Bielefeld

Gutachter: Univ.-Prof. Dr. rer. nat. Johannes Gottschling
Prof. Dr. rer. nat. Robert Martin

Tag der mündlichen Prüfung: 9. März 2023

————— *Diese Seite wurde absichtlich leer gelassen.* —————

Kurzfassung

Die industrielle Produktion steht vor zahlreichen globalen Herausforderungen. Ein dynamischer Markt mit komplexen Kundenanforderungen sowie steigende Rohstoff- und Energiepreise sind nur einige der treibenden Faktoren intelligenter Prozesstechnologien. Digitalisierungsmaßnahmen zur Schaffung adaptiver und robuster Fertigungsumgebungen erfordern die effektive Nutzung von Prozessdaten. In der vorliegenden Dissertation wurde ein methodisches Grundkonzept entwickelt, das in der Lage ist, Fertigungsprozesse mit prädiktiven Modellen des maschinellen Lernens zu optimieren. Es besteht aus drei verketteten Modulen, die jeweils eine Funktion der Wissensgenerierung aus Daten zur Prozessoptimierung erfüllen und stufenweise eingesetzt werden können. Jedes Modul implementiert modellagnostische Verwertungsansätze, die sich mit substituierbaren Analyse- und Optimierungsmethoden realisieren lassen.

Das Datenvorverarbeitungsmodul hat zur Aufgabe, eine ganzheitliche, mit maschinellem Lernen analysierbare Datenbasis zu schaffen und erste relevante Informationen zu extrahieren. Es liefert die Rahmenbedingungen der prozessspezifischen Aufbereitung, einer robusten Störgrößenkompensation und einer thresholdbasierten Verdichtung der Rohdaten auf Basis unüberwachten Lernens. Das Modellierungsmodul befasst sich mit der datengetriebenen Systemidentifikation komplexer Prozesse. Neben Ansätzen der Modellentwicklung und -evaluation wird die Nutzung empirischer Modellinterpretationsverfahren beschrieben, die der Plausibilitätsprüfung und der Verifizierung von Wirkzusammenhängen dienen. Abschließend erfolgt die Konzipierung des Optimierungsmoduls, das auf den vorangegangenen Ergebnissen fußt. Gradientenfreie Verfahren realisieren die systematische Optimierung. Sie nutzen Prinzipien der virtuellen DoE, die im Rahmen analytischer Modellierung Anwendung findet. Die zugrunde liegende Kostenfunktion enthält über die Zielkriterien und Nebenbedingungen hinaus Komponenten, die die synthetisierten Prozessvarianten mit der Grundverteilung der Daten vergleichen und so ihre Plausibilität und verfahrenstechnische Umsetzbarkeit bewerten. Ferner wird die Optimierung insoweit mathematisch manipuliert, als die Zahl der anzupassenden Steuerungsgrößen für einen Prozesseingriff gelenkt werden kann.

Die entwickelte Optimierungskette wird am Beispiel eines Schleudergießprozesses verifiziert. Das Ziel der Optimierung ist die Variation der Prozessparameter zur Fertigung duktiler Gusseisenrohre mit einem möglichst homogenen Wandstärkenprofil. Als Störgrößenkompensation für prozessspezifische Einflüsse bei der Wandstärkenmessung dient die robuste Hauptkomponentenanalyse. Sie ist ferner in der Lage, Sensorausfälle auszugleichen und die Prognosegenauigkeit signifikant zu verbessern. Die Wandstärken lassen sich mit einem CatBoost-Regressor positionsabhängig mit hoher Genauigkeit vorhersagen. Das datengetriebene Modell kann bis zu 96 % der Gesamtvarianz erklären. Verfahrensspezifische Stellgrößen und die Flüssigeisentemperatur konnten als Haupteinflüsse identifiziert werden. Die systematische Optimierung erfolgt durch die Variation von 15 Steuerungsgrößen. Ein Partikelschwarmalgorithmus liefert den besten Kompromiss aus der Erfüllung der Zielvorgabe und der Zuverlässigkeit der Konvergenz. Die Wandstärken können so im relevanten Bereich erheblich reduziert werden. Abschließend werden Regularisierungen der Kostenfunktionen angewandt. Hierdurch lassen sich dünnbesetzte Handlungsempfehlungen generieren, die nur noch die Veränderung von drei Stellgrößen erfordern. Die Qualität des resultierenden Prozessergebnisses nimmt jedoch ab.

Abstract

Industrial production faces numerous global challenges. A dynamical market with complex customer demands alongside rising prices for raw materials and energy are among the driving factors for intelligent production technologies. Measures for digitalization to create adaptive and robust manufacturing environments require efficient usage of process data. In this dissertation, a methodical concept is developed that optimizes manufacturing processes with predictive machine learning models. It consists of three interconnected modules that each represent a functionality to extract knowledge from data for optimization and can be used gradually. Each module implements model-agnostic approaches and can be put into practice with multiple techniques for analysis and optimization.

The data preprocessing module's task is to generate a data pool that can be analyzed with machine learning and to extract relevant information. It provides the general framework of process-specific data-preparation, robust disturbance handling and a threshold-based compression of the data by means of unsupervised learning. The modelling module addresses data-driven system-identification of complex processes. Besides model development and evaluation, empirical procedures for model-interpretation are described that serve as an assessment of plausibility and verification of cause-effect correlations. Finally, an optimization module is engineered that is based on the previously obtained results. The systematic optimization is implemented with gradient-free methods. They utilize principles of a virtual design of experiments that are usually applied in analytical modelling. The underlying cost-function consists of target entities and constraints, but also measures that project generated process variants onto the distribution of the data to classify anomalies and assess whether they are physically feasible and technically viable. Moreover, the cost-function is manipulated mathematically to attain the ability to lower the amount of control variables necessary for a manual process adjustment.

The optimization chain is verified using data from a centrifugal casting process. To this end, process parameters are to be varied so that ductile cast iron pipes with a homogeneous wall thickness distribution are produced. Process specific disturbances affecting the measurements are handled with a robust principal component analysis. It also renders the possibility to counterbalance sensor outages and thus significantly improve the prediction performance. Wall thicknesses along the pipes can be highly accurately predicted with a CatBoost regressor. The data-driven model explains up to 96 % of the variance. Process related control inputs and the iron temperature at entry into the mould were identified as the main influencing factors. The systematic optimization was performed with 15 control variables. A particle swarm optimization yields the best compromise between the quality of the obtained solution and the reliability of convergence. A substantial reduction of wall thickness in relevant areas was attained. Finally, regularizations were applied to the cost-function. These mathematical modifications yield sparse adjustments to the process that only require three control variables to be changed. However, compromises regarding the quality of the output must be tolerated.

Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Forschungszentrum Allgäu der Hochschule für angewandte Wissenschaften Kempten in Zusammenarbeit mit dem Lehrstuhl Mathematik für Ingenieure der Universität Duisburg-Essen.

Mein Dank gilt allen Wissenschaftlern, die dieses Vorhaben begleitet haben. Mein Doktorvater, Herr Prof. Dr. rer. nat. Johannes Gottschling, hat meine Forschungstätigkeit unermüdlich unterstützt und es mir ermöglicht, Ideen anwendungsnahe zu verwirklichen. Die fachlichen Diskussionen mit meinem Zweitgutachter, Herrn Prof. Dr. rer. nat. Robert Martin, waren überaus wertvoll; durch sie lernte ich, Ergebnisse stets kritisch zu hinterfragen. Herrn Prof. Dr.-Ing. Dierk Hartmann (Labor für Werkstofftechnik und Betriebsfestigkeit der Hochschule Kempten), in dessen Arbeitsgruppe ich tätig sein durfte, verdanke ich einen unschätzbaren fachlichen und menschlichen Wissensschatz. Herr Prof. Dr.-Ing. Frank Schmidt hat dankenswerterweise den Vorsitz der Prüfungskommission und Herr Prof. Dr.-Ing. Arun Nagarajah die Funktion des Mitprüfers übernommen.

Besonders hervorheben möchte ich den Beitrag meiner Kollegen und Freunde am Forschungszentrum Allgäu. Ausdrücklich erwähnt seien Frau Melina Graner, Herr Marvin Sandt und Herr Florian Huber. Sie haben mir immer wieder Denkanstöße und konstruktive Kritik gegeben. Ohne ihre Unterstützung hätte ich diese Dissertation nicht fertigstellen können.

Ein herzlicher Dank gilt auch meiner Familie, insbesondere meinen Eltern, Dr. rer. soc. Pataya Ruenkaew und Dr. rer. soc. Kurt Salentin, ohne die ich nicht der Mensch wäre, der ich heute bin. Sie haben mir das Selbstvertrauen gegeben, die Ziele zu erreichen, die ich mir setze.

Inhaltsverzeichnis

Abbildungsverzeichnis	xi
Tabellenverzeichnis	xiii
Abkürzungsverzeichnis	xv
Formelzeichenverzeichnis	xvii
1 Einleitung	1
1.1 Problemstellung und Motivation	1
1.2 Zielsetzung	3
1.3 Aufbau der Arbeit	4
2 Theoretische Grundlagen	7
2.1 Mathematische Optimierung	7
2.1.1 Mehrkriterielle Optimierung	7
2.1.2 Gradientenfreie Optimierung	9
2.2 Grundlagen des maschinellen Lernens	11
2.2.1 Paradigmen des maschinellen Lernens	11
2.2.2 Modellvalidierung	13
2.2.3 Gütekriterien	14
2.2.4 Featureauswahl	16
2.2.5 Empirische Modellinterpretation	17
2.2.6 Mehrgrößenprognosen	19
2.3 Statistische und maschinelle Lernmodelle	20
2.3.1 Lineare Regression	20
2.3.2 Neuronale Netze	21
2.3.3 Entscheidungsbäume	22
2.3.4 Ensembleverfahren	23
2.4 Datenvorverarbeitung und Dimensionsreduktion	26
2.4.1 Lineare Korrelation	26
2.4.2 Singulärwertzerlegung	27
2.4.3 Hauptkomponentenanalyse	28
2.4.4 Robuste Hauptkomponentenanalyse	29
2.4.5 Autoencoder	31
2.4.6 Schnelle Fourier-Transformation	32
2.4.7 Verdichtung des Messraums	32

3	Stand der Technik und der Forschung	35
3.1	Prozessoptimierung in der Gießereiindustrie	35
3.1.1	Qualitätsvorhersage mit maschinellem Lernen	35
3.1.2	Simulationsgetriebene Prozessoptimierung	38
3.2	Anomalieerkennung für industrielle Prozessdaten	40
3.3	Dimensionsreduktion und Messraumverdichtung	40
3.4	Industrielle Data-Mining-Standards	41
3.5	Verwandte Forschungsarbeiten	42
4	Entwicklung der Optimierungskette	45
4.1	Rahmenbedingungen und Lösungsweg	45
4.1.1	Anforderungen	45
4.1.2	Entwicklungsansatz	48
4.1.3	Datenmanagement und Schnittstellen	51
4.2	Datenvorverarbeitungsmodul	53
4.2.1	Prozessspezifische Aufbereitung	53
4.2.2	Datentransformation	56
4.2.3	Thresholding der Prozessdaten	58
4.3	Modellierungsmodul	61
4.3.1	Datengetriebene Modellentwicklung und -optimierung	62
4.3.2	Prozessspezifische Modellinterpretation	62
4.4	Optimierungsmodul	65
4.4.1	Prinzip der systematischen Optimierung	65
4.4.2	Definition einer Kostenfunktion	68
4.4.3	Bereitstellung der Handlungsempfehlungen	70
4.5	Implementierungsaspekte	73
4.5.1	Modulübergreifende Bewertung	73
4.5.2	Bereitstellung der Module	74
5	Anwendung der Entwicklungsmethode	77
5.1	Grundlagen der Fallstudie	77
5.1.1	Gießtechnische Grundlagen	77
5.1.2	Auslegung der Fallstudie	79
5.1.3	Beschreibung des Prozesses	80
5.1.4	Beschreibung der Datenbasis	82
5.2	Datenvorverarbeitung und Dimensionsreduktion	83
5.2.1	Störgrößenkompensation und Rauschreduktion	84
5.2.2	Reduktion und Rekonstruktion der Messungen	90
5.2.3	Handhabung von Sensorausfällen	91
5.3	Wandstärkenmodellierung	94
5.3.1	Einzelprognosen	94
5.3.2	Performanceanalyse der Modelle	97
5.3.3	Modellierung des Gesamtprofils	100
5.4	Sensitivitätsanalysen	101
5.4.1	Plausibilitätsprüfung mit virtuellen Parametervariationen	101
5.4.2	Systematische Evaluation von Einflussgrößen	103
5.5	Systematische Optimierung des Wandstärkenverlaufs	108

5.5.1	Optimierungsstrategie	108
5.5.2	Ergebnisse der Optimierung	112
5.5.3	Generierung von Handlungsempfehlungen	120
6	Zusammenfassung und Ausblick	127
6.1	Zusammenfassung	127
6.2	Ausblick	129
	Literatur	131

Abbildungsverzeichnis

1.1	Prinzip der datengetriebenen Regelung	5
1.2	Aufbau der Arbeit	6
2.1	Paretofront	8
2.2	Maschinelles Lernen in der Prozessindustrie	12
2.3	Beispiel eines neuronalen Netzes	22
2.4	Beispiele für Ensemblelernverfahren	24
2.5	Beispiel für einen Autoencoder	31
3.1	Simulationsgestützte Gießereiprozessoptimierung	39
3.2	CRISP-DM	41
4.1	Entwicklungsansatz	49
4.2	Datenlage in einer Fertigungssituation	50
4.3	Datenmanagement	52
4.4	Beispielprozess zur Datenzusammenführung	54
4.5	Aufbereitung der Qualitätsdaten	56
4.6	Anwendung der Dimensionsreduktion auf Prognosefunktionen	57
4.7	Nutzung der Rang- und Dimensionsreduktion	59
4.8	Modellentwicklung	63
4.9	Prinzip der systematischen Optimierung	67
4.10	Struktur des Assistenzsystems	72
4.11	Modulare Bewertung der Optimierungskette	74
5.1	Übersicht über wichtige Gießverfahren	78
5.2	Projektziele	80
5.3	Übersicht des Prozesses	81
5.4	Verfahrensablauf beim Schleudergießen	82
5.5	Störgrößenkompensation	84
5.6	Singulärwerte und kumulierte Energie	85
5.7	Vergleich der Rangreduktion von SVD und RPCA	87
5.8	Vergleich der Ränge von SVD und RPCA	88
5.9	Rauschreduktion mit FFT	89
5.10	Dimensionsreduktion mit Autoencodern	90
5.11	Verdichtung der Messung	92
5.12	Simulation von Sensorausfällen	93
5.13	Einzelprognose bei $l = 580$ mm	95

5.14	Vergleich der Einzelprognosen bei $l = 2080$ mm	96
5.15	Histogramm der Prognoseresiduen	97
5.16	Vergleich der Einzelprognosen bei $l = 4280$ mm	98
5.17	Prognose des gesamten Wandstärkenverlaufes	100
5.18	Einzel-DoEs	102
5.19	SHAP-Werte für die Prognose bei $l = 4080$ mm	103
5.20	Featurerelevanz für die Prognose bei $l = 4080$ mm	104
5.21	Partielle Abhängigkeit der Eisentemperatur	105
5.22	Partielle Abhängigkeit der Kipperrückfahrposition	106
5.23	SHAP-Werte für die Prognose bei $l = 2080$ mm	107
5.24	Hauptkomponenten der Wandstärkendaten	108
5.25	Auswirkung der Wandstärkenkorrektur	109
5.26	Optimierungsstrategie	110
5.27	Beschränkungsfunktion für die Nebenbedingungen	111
5.28	Optimierungstopographie für Nelder-Mead	113
5.29	Optimierungsergebnis für Nelder-Mead	114
5.30	Prozessvariante mit Artefakten	114
5.31	Konvergenz der Partikelschwarmoptimierung	116
5.32	Optimierungsergebnis des Partikelschwarmalgorithmus	116
5.33	Kostenfunktion des genetischen Algorithmus	117
5.34	Auswirkung der Optimierung als Boxplot	118
5.35	Auswirkung der Optimierung als Boxplot mit echten Daten	119
5.36	Handlungsempfehlung ohne Beschränkung	120
5.37	Handlungsempfehlung mit ℓ_2 -Regularisierung	121
5.38	Handlungsempfehlung mit ℓ_1 -Regularisierung	122
5.39	Handlungsempfehlung mit ℓ_0 -Regularisierung	123
5.40	Auswirkung der ℓ_k -Regularisierung auf die Kostenfunktion	124

Tabellenverzeichnis

2.1	Konfusionsmatrix für Binärklassifikation	14
2.2	Begriffsbildung zu Mehrgrößenprognosen	19
2.3	Kostenfunktionen für lineare Regression	21
3.1	Publikationen aus der Gießereitechnik	36
4.1	Komponenten der Kostenfunktion	68
5.1	Vergleich der Einzelmodelle	99
5.2	Übersicht der Optimierungsergebnisse	125

Abkürzungsverzeichnis

BN	Bayes'sche Netze
CART	Classification and Regression Trees
CRISP-DM	Cross Industry Standard for Data Mining
DAE	Denoising Autoencoder
DoE	Versuchsplanung (Design of Experiment)
DTW	Dynamic Time Warping
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
FFT	Schnelle Fourier-Transformation (Fast Fourier-Transform)
GAN	Generative Adversarial Nets
IoT	Internet of Things
KDD	Knowledge Discovery in Databases
k-NN	k-Nearest-Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
MAE	Mittlere absolute Abweichung (Mean Absolute Error)
MES	Manufacturing Execution System
MLP	Mehrlagiges Perzeptron (Multilayer Perceptron)
MPC	Model Predictive Control
NoSQL	Not only SQL
PCA	Hauptkomponentenanalyse (Principal Component Analysis)
PDP	Partielle Abhängigkeit (Partial Dependence Plot)
PFI	Permutationsfeaturerelevanz (Permutation Feature Importance)
PSO	Partikelschwarmoptimierung
RDBMS	Relationale Datenbanksysteme
ReLU	Rectified Linear Unit
RMSE	Mittlere quadratische Abweichung (Root Mean Square Error)
RPCA	Robuste Hauptkomponentenanalyse

RRMSE	Relative mittlere quadratische Abweichung
SAE	Sparse Autoencoder
SCADA	Supervisory Control and Data Acquisition
SHAP	Shapley Additive Explanations
SPE	Squared Prediction Error (Q-Statistik)
SPS	Speicherprogrammierbare Steuerungen
SVD	Singulärwertzerlegung (Singular Value Decomposition)
SVM	Support Vector Machine
TAN	Tree Augmented Naive
TPOT	Tree-based Pipeline Optimization Tool
VAE	Variational Autoencoder

Formelzeichenverzeichnis

Skalare

ACC	Korrektklassifikationsrate (Accuracy)
c_1	kognitive Komponente des Partikelschwarms
c_2	soziale Komponente des Partikelschwarms
FN	falsche Negative (false negatives)
FP	falsche Positive (false positives)
K_{th}	Threshold der RPCA
l	Messpunkt am Rohr in x-Richtung
R^2	Bestimmtheitsmaß
REC	Trefferquote (Recall)
t	Zeitpunkt
TN	wahre Negative (true negatives)
TP	wahre Positive (true positives)
w	Trägheit eines Partikels
α	y-Achsenabschnitt der Regressionsgeraden
α_i	Gewichtungsfaktor für die Kostenfunktion
β	Steigung der Regressionsgeraden
ε_r	relativer Threshold der RPCA
λ	Eigenwert (2.4.3), Skalierungsfaktor
μ	Skalierungsfaktor
$\rho_{\mathbf{x}_i \mathbf{x}_j}$	Korrelationskoeffizient nach Pearson für die Variablen \mathbf{x}_i und \mathbf{x}_j
σ_i	i -ter Singulärwert

Dimensionen

d	Anzahl der funktional abhängigen Variablen (Spalten von \mathbf{Y})
k	Anzahl zu optimierender Zielgrößen
m	Anzahl der Zeilen der Datenmatrix
n	Anzahl der funktional unabhängigen Variablen (Spalten von \mathbf{X})
p	Funktionsauswertungen bei der Sensitivitätsanalyse
q	Anzahl der optimalen Prozessvarianten
r	reduzierter / intrinsischer Rang

Normen

$\ \cdot\ _0$	ℓ_0 -Pseudo-Norm eines Vektors \mathbf{x} , von Null verschiedene Elemente in \mathbf{x}
$\ \cdot\ _1$	ℓ_1 -Norm eines Vektors, gegeben durch $\ \mathbf{x}\ _1 = \sum_{i=1}^n x_i $
$\ \cdot\ _2$	ℓ_2 -Norm eines Vektors, gegeben durch $\ \mathbf{x}\ _2 = \sqrt{\sum_{i=1}^n x_i^2}$
$\ \cdot\ _F$	Frobenius-Norm einer Matrix, gegeben durch $\ \mathbf{X}\ _F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m x_{ij}^2}$
$\ \cdot\ _*$	Nuklear-Norm einer Matrix, gegeben durch $\ \mathbf{X}\ _* = \text{trace}(\sqrt{\mathbf{X}^T \mathbf{X}}) = \sum_{i=1}^n \sigma_i$

Operatoren und Funktionen

f	Prognosefunktion
f_a	Aktivierungsfunktion
f_f	Fitnessfunktion
f_s	Sensitivitätsfunktion
\mathcal{F}	Fourier-Transformation
g_i	Beschränkungsfunktion für \mathbf{x} , $g_i(\mathbf{x}) \leq 0$
g_l	untere Beschränkungsfunktion
g_u	obere Beschränkungsfunktion
h_i	Einzelmodell bei Boosting (2.3.4), Beschränkungsfunktion für \mathbf{x} , $h_i(\mathbf{x}) = 0$
J	Kostenfunktion
L	Verlustfunktion
pds	partielle Abhängigkeit
p_X	Wahrscheinlichkeitsdichte von X
Q	Q-Statistik
Γ	Optimalitätskriterium der Zielgröße
ζ	Verteilungskriterium der unabhängigen Variablen
Ξ	ℓ_k -Regularisierung der Kostenfunktion
φ	Encoder (2.4.5), Transformation $\mathbf{X} \rightarrow \mathbf{Z}$
ψ	Decoder (2.4.5), Transformation $\mathbf{Y} \rightarrow \mathbf{T}$
Π	Nebenbedingungen der Zielgröße
Υ	Verteilungskriterium der abhängigen Variablen

Mengen

\mathcal{F}	Menge der gültigen Lösungen bei der Optimierung
X_C	Menge nicht untersuchter Variablen bei der Sensitivitätsanalyse
X_S	Menge untersuchter Variablen bei der Sensitivitätsanalyse
\mathcal{Y}	Menge der gültigen Zielfunktionswerte bei der Optimierung

Matrizen und Vektoren

B	mittelwertfreie Datenmatrix
C	Messmatrix
$\text{cov}(\mathbf{X})$	Kovarianzmatrix von \mathbf{X}
f	vektorwertige Prognosefunktion
I	Einheitsmatrix
L	Matrix kohärenter Strukturen der RPCA
S	Matrix des Rauschens der RPCA
T	Matrix der transformierten Zielgrößen
U	linke Singulärvektoren von \mathbf{X} , $\mathbf{U} \in \mathbb{R}^{m \times m}$
P	rechte und linke Singulärvektoren der Kovarianzmatrix von \mathbf{X}
$\tilde{\mathbf{U}}$	reduzierte linke Singulärvektoren von \mathbf{X} , $\tilde{\mathbf{U}} \in \mathbb{R}^{m \times r}$
V	rechte Singulärvektoren von \mathbf{X} , $\mathbf{V} \in \mathbb{R}^{n \times n}$
$\tilde{\mathbf{V}}$	reduzierte rechte Singulärvektoren von \mathbf{X} , $\tilde{\mathbf{V}} \in \mathbb{R}^{n \times r}$
X	Datenmatrix, $\mathbf{X} \in \mathbb{R}^{m \times n}$
$\tilde{\mathbf{X}}$	rangreduzierte Datenmatrix, $\tilde{\mathbf{X}} \in \mathbb{R}^{m \times n}$
x	Prozessinstanz, Prozessvariante
\mathbf{x}_i	i -ter Spaltenvektor der Datenmatrix
$\mathbf{x}^{(i)}$	i -ter Zeilenvektor der Datenmatrix
\mathbf{x}_{opt}	optimale Prozessvariante
Y	Matrix funktional abhängiger Variablen
Z	Matrix der Hauptkomponenten, transformierte Prozessdaten
\mathbf{z}_i	i -te Hauptkomponente
θ	Regressionskoeffizienten zur Rekonstruktion von $\tilde{\mathbf{X}}$
Θ	Matrix der Regressionskoeffizienten
ξ	reduzierte Messung von \mathbf{x}
ϱ	Auswahlvektor der linear unkorrelierten Spalten von \mathbf{X}
Σ	Matrix der Singulärwerte von \mathbf{X} , $\Sigma \in \mathbb{R}^{m \times n}$
$\tilde{\Sigma}$	reduzierte Matrix der Singulärwerte von \mathbf{X} , $\tilde{\Sigma} \in \mathbb{R}^{r \times r}$
Φ	Datenmatrix ausgewählter Features

Kapitel 1

Einleitung

1.1 Problemstellung und Motivation

Intelligente Fertigungsumgebungen sind das Gebot der Stunde. Die industrielle Produktion in Deutschland der 2020er Jahre ist weitreichenden globalen Herausforderungen ausgesetzt. Der Konkurrenzdruck neuer Wettbewerber aus Schwellenländern wächst, die Kundenanforderungen werden immer komplexer. Weiterhin stellen Rohstoff- und Energieeffizienz vor dem Hintergrund hoher Produktionskosten heute Grundanforderungen an industrielle Prozesse. Die Sicherung des Wirtschaftsstandorts Deutschland erfordert die Weiterentwicklung von Prozesstechnologie und -beherrschung hin zu intelligenten und adaptiven Fertigungsumgebungen. Die Themen Industrie 4.0 und Digitalisierung dominieren den Diskurs über Technologien der Zukunft (Reinheimer 2017, S. 91). Ein Pfeiler für die erfolgreiche Umsetzung des Internets der Dinge ist die Sammlung und die effektive Nutzung von Prozessdaten zur Optimierung bestehender Produktionssysteme (Andelfinger und Hänisch 2017, S. 11). Datengetriebene Ansätze in Verbindung mit Methoden des maschinellen Lernens ermöglichen es, aus bereits bestehenden und stetig anfallenden Datenmengen Wissen zu extrahieren und Zusammenhänge zu erkennen, die sich auf Basis einer rein analytischen Modellbildung und klassischen Statistik nicht ableiten ließen. Dies verheißt neues Potenzial, mit technischen Systemen steuernd zu interagieren. Besonders deutsche kleine und mittlere Unternehmen (KMU) können von diesen Methoden profitieren und sich auf diese Weise die Wettbewerbsfähigkeit auf einem globalen Markt sichern (Becker et al. 2017, S. 7). Aufgrund unterschiedlicher Hürden wird das Potenzial datengetriebener Optimierungsverfahren allerdings noch nicht vollständig ausgeschöpft. So schaffen es Umfragen zufolge nur knapp 20 % der angegangenen Projekte bis in die Produktion (Escobar et al. 2021).

Die Gießereiindustrie braucht intelligentere Fertigungstechnologien. Deutschland ist der europaweit größte und weltweit viertgrößte Produzent von Gussteilen. Aufgrund ihrer Funktion als Zulieferer für Schlüsselbranchen wie den Automobil- und Maschinenbau erwächst aus der mittelständisch geprägten Gießereiindustrie ein enormes wirtschaftliches Nutzungspotenzial. Die sich wandelnde Nachfrage, globale Krisen und die Entwicklung neuer Antriebstechnologien, aber auch standortbedingte Besonderheiten wie der Emissionshandel sind nur einige der treibenden Faktoren für einen Paradigmenwechsel in der Branche hin zu intelligenten, adaptiven Fertigungstechnologien (Kallien et al. 2018). Die frühzeitige Planung von Prozessen und Produkten mit sowohl simulationsgestützten als

auch rein datengetriebenen Methoden ist in das Standardportfolio mittelständischer Gießereien zu überführen. Ferner müssen die Analyseverfahren mit Expertenwissen kombinierbar sein, um den Anforderungen der komplexen und heterogenen Produktionslandschaft gerecht zu werden.

Warum sind fortgeschrittene Technologien noch nicht implementiert? Für die heutzutage noch weitestgehend statische, intuitionsbasierte Steuerung der Fertigungsprozesse ohne computergestützte Entscheidungsgrundlage liegen unterschiedliche Gründe vor. Häufig fehlt es schon an einer intelligenten Vernetzung der Teilprozesse, die die Datenquellen darstellen (Konrad et al. 2013). Klassische analytische Modellbildung konnte bereits in vielen Teilen der Produkt- und Fertigungsplanung erfolgreich umgesetzt werden und gehört mittlerweile als fester Bestandteil zur Inbetriebnahme und zur kontinuierlichen Prozessverbesserung. Die Verfahren werden jedoch bei hochgradig nichtlinearen Zusammenhängen und Millionen von Freiheitsgraden kombinatorisch nicht mehr tragbar (Tonn et al. 2021). Zudem stehen Simulationen nicht für jeden Fertigungsprozess kommerziell zur Verfügung und erlauben bei nicht bekannten Systemkenntnissen darüber hinaus keine Darstellung von Optimierungsproblemen in geschlossener Form.

Die Herausforderung des Gießereiwesens liegt in der Beschaffenheit seiner Daten. Aufgrund der rapide gestiegenen industriellen Datenverfügbarkeit und nutzbaren Rechenleistung werden komplexe Optimierungsprobleme vermehrt mit Methoden des maschinellen Lernens gelöst. Hinsichtlich der Datenbeschaffenheit in Gießerei bleiben jedoch noch große Hürden, die es zu nehmen gilt (Saleem et al. 2015):

- P₁: **Datenheterogenität** Eine hohe Anzahl an Einflussparametern aus heterogenen Datenquellen wird mit unterschiedlicher Qualität und Periodizität erhoben. Eine einheitliche Betrachtung ist nicht unmittelbar möglich.
- P₂: **Umgebungsbedingungen:** Schwierige Messumgebungen beeinflussen die Datenaufnahme und damit ihre Güte. Einige Einflussgrößen werden nicht aufgenommen, sodass eine logische Lücke zwischen Erfassung und Wirkungsmuster besteht.
- P₃: **Fehlende analytische Beschreibbarkeit:** Einige Fertigungsprozesse lassen sich aufgrund ihres physikalischen Aufbaus nicht anforderungsgerecht simulieren. Die meisten externen Störungen können analytisch nicht abgebildet werden.
- P₄: **Komplexe Wechselwirkungen:** Kollinearitäten und wechselseitige Abhängigkeiten zwischen Einflussgrößen erschweren das Nachvollziehen von Wirkungsketten und lassen sich nicht trivial darstellen. Zielgrößen wirken teils konträr.

Maßgeschneiderte Lösungen sind erforderlich. Die genannten Probleme P₁ und P₂ wirken sich in direktem Maße auf die Datenqualität aus und erschweren eine zielführende Analyse. Ferner erlaubt die Datensituation der meisten KMU noch keine holistische Betrachtung der Prozesse, da operative Daten und Fertigungsparameter nicht miteinander kombiniert werden können. So ermöglichen die auf dem Markt verfügbaren Dashboard-Lösungen zur Online-Datenverarbeitung (engl. *Online Analytical Processing*, OLAP) nur die Nutzung deskriptiver Statistik ohne Data-Mining zur Wissensextraktion. Analyseergebnisse lassen sich nicht interoperabel zwischen Subprozessen austauschen, und es obliegt weiterhin dem Prozessexperten, die richtigen Schlussfolgerungen aus ihnen zu

ziehen und in direkte Maßnahmen zur Steuerung der Fertigung zu überführen (Gröger et al. 2012).

Für die Bereitstellung eines nutzbaren Basisdatenbestandes können verschiedene Methoden einer vorgelagerten Datenaufbereitung herangezogen werden. Sie erfordern jedoch häufig eine analytische Systembeschreibung oder nicht erfüllbare Annahmen an die Charakteristik der Störgrößen. Andere Verfahren wurden bislang nur unter Laborbedingungen oder mit synthetischen Daten getestet. Seitens einer zur Modellierung nachgelagerten Prozessoptimierung liegen die Vorstellungen der digital bereits gut aufgestellten Großunternehmen und der KMU teils weit auseinander. Meist liegt kein Bestreben vor, Prozesse völlig autonom zu betreiben, sodass Assistenzsysteme in Form von Handlungsempfehlungen für den Maschinenbediener bereitgestellt werden. Sie können nicht pauschal entworfen werden, sondern müssen an die Anforderungen des Personals vor Ort und die Umgebungsbedingungen abgestimmt werden.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist der Entwurf und die Verifizierung einer modularen datengetriebenen Optimierungskette, die den Besonderheiten der Gießereiindustrie Rechnung trägt. Der Ansatz berücksichtigt die oben genannten Probleme. Hierzu wird eine generische und datengetriebene methodische Grundlage entwickelt, die den Erkenntnisgewinn aus Fertigungsprozessen unterstützt und eine Optimierung durch gezielte Handlungsanweisungen ermöglicht. Ein wichtiger Beitrag hierfür ist die Stärkung des menschlichen Verständnisses für Ursache-Wirkungszusammenhänge. Die Entscheidungsfindung für Prozesseingriffe und weitere Maßnahmen soll beschleunigt und verbessert werden. Der Subjektivität des menschlichen Urteilsvermögens soll entgegengewirkt und objektive Entscheidungsgrundlagen bereitgestellt werden. Ziel ist es, Modelle zu trainieren, die sowohl für Prozessingenieure als auch für Fertigungsmitarbeiter interpretierbar und in der Praxis zur prädiktiven Verbesserung der Produktion nutzbar sind. Die Ziele der Arbeit werden nachfolgend definiert:

- Z₁: Entwicklung von Methoden der Systemidentifikation von Fertigungsprozessen durch datengetriebene Modellierung
- Z₂: Methoden zur Messraumverdichtung bei multidimensionalen Problemstellungen und optimierte Platzierung von Sensorik für eine Optimierung
- Z₃: Entwurf einer Kostenfunktion, die das Optimierungsziel erfasst
- Z₄: Methodik zur verteilungsunabhängigen Störgrößenkompensation
- Z₅: Anwendungsgerechte Generierung von Handlungsempfehlungen als Ergebnis einer datengetriebenen Optimierung der Prozessvariante
- Z₆: Bereitstellung der Handlungsempfehlung

Der zu erarbeitende Teil der Methodik, der über den Stand der Technik hinausgeht, unterteilt sich in drei Module, die sich an den drei in der Literatur ausgezeichneten Hauptanwendungsbereichen für die datengetriebene Optimierung technischer Prozesse orientieren (Brunton und Kutz 2019, S. 345):

- Rechenleistung und kostspielige Sensorik stehen nicht uneingeschränkt zur Verfügung. Das *Datenvorverarbeitungsmodul* beschäftigt sich mit der ressourcenökonomischen Platzierung von Sensorik (Z_2), um zu ermitteln, welche Zustands- und Zielgrößen im Prozess gemessen werden müssen, um den maximalen Informationsgehalt zu bewahren, aber gleichzeitig Ressourcen- und Rechenaufwand einzusparen (vgl. P_1). Darüber hinaus wird eine domänenspezifische und verteilungsunabhängige Störgrößenkompensation (Z_4) konzeptioniert, die fehlerhafte Daten erkennt und bereinigt (vgl. P_2).
- Im *Modellierungsmodul* erfolgt die Herstellung einer Beziehung zwischen dem Ein- und Ausgangsverhalten eines Prozesses (Z_1). Dies erfolgt durch Verfahren des maschinellen Lernens unter stetiger Prüfung der Ergebnisse mit Expertenwissen. Es handelt sich dabei um eine Form der datengetriebenen Systemidentifikation, die ein weitreichendes Spektrum von Einflüssen beschreiben kann (vgl. Z_3). Methoden der empirischen Modellevaluation stützen die Ergebnisse durch statistische Absicherung und fungieren als Plausibilitätsprüfung.
- Die Planung des Prozesses und proaktive Eingriffe in ihn werden vom *Prozessoptimierungsmodul* bereitgestellt (Z_5). Es befasst sich mit der Verarbeitung des Prozessausgangs, meist in Form von Messungen, zu einem Steuersignal, das über die Aktorik oder den Mensch in das System zurückgeführt wird (Z_6). Zudem muss es berücksichtigen, welche Prozessgrößen zum Zeitpunkt des Eingriffes noch gesteuert werden können, um die Zielgröße zu beeinflussen. Letztlich entscheidet die Kostenfunktion (Z_3) über die Erreichbarkeit der Optimierungsziele, die Konvergenz zur optimalen Lösung sowie die physikalische Plausibilität und technische Machbarkeit der synthetisierten Prozessvariante (vgl. P_4).

Abbildung 1.1 zeigt das Zusammenwirken der Module in abstrahierter Form. Sie sind an der Darstellung eines verallgemeinerten Regelkreises angelehnt. Es werden keine inneren Zustände des Systems für die Dynamik des Prozesses auf analytischer oder datengetriebener Basis betrachtet. Der Prozess wird als zustandslos angenommen und lediglich anhand der Eingangsgrößen und der geschätzten Störungen modelliert.

1.3 Aufbau der Arbeit

Der Rest der Arbeit ist wie folgt aufgebaut, vergleiche dazu Abbildung 1.2: Nachdem die Zielsetzung im einleitenden Kapitel fixiert wurde, wird in Kapitel 2 das theoretische Grundlagenwissen vermittelt, das zum weiteren Verständnis notwendig ist. Es umfasst wesentliche Elemente der Datenaufbereitung, wie solche auf Basis der Hauptkomponentenanalyse und klassische frequenzbasierte Verfahren. Des Weiteren erfolgt eine Einführung in die Grundkonzepte des maschinellen Lernens mit dem Schwerpunkt auf überwachtem Lernen zur Modellierung und der empirischen Modellinterpretation zum Verständnis von Black-Box-Systemen. Darüber hinaus werden mathematische Verfahren zur gradientenfreien Optimierung multivariater Funktionen vorgestellt, die die Grundlage für eine spätere Prozessoptimierung darstellen.

Kapitel 3 fasst den aktuellen Stand der Forschung und der Technik für diese Problemstellung zusammen. Die Optimierungsstrategien für Gießereiprozesse, welche sich

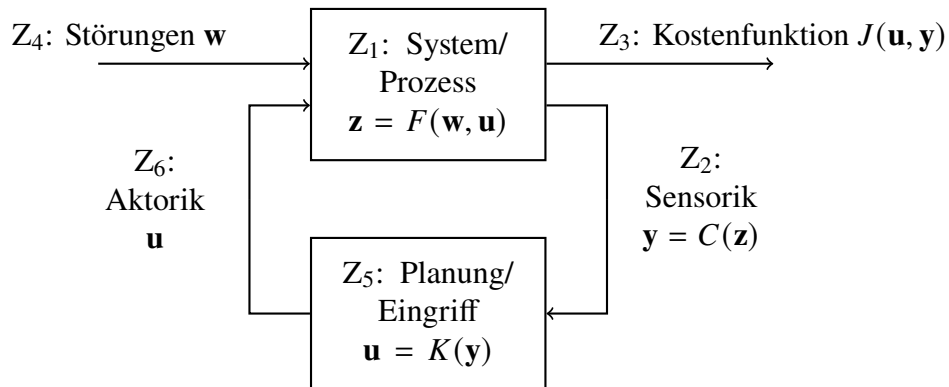


Abbildung 1.1: Prinzip der datengetriebenen Regelung für dynamische Systeme. Sensor-signale y werden durch K in geeignete Maßnahmen umgewandelt und über die Aktorik u realisiert. Eine analytische Beschreibung von Prozessen lässt sich in Gießereien nur in wenigen Fällen umsetzen.

in simulationsgestützte Bauteilauslegung und datengetriebene Fehlerdiagnose unterteilen, werden auf Grundlage aktueller Veröffentlichungen untersucht. Daneben werden Konzepte der Datenanalyse in anderen Prozessindustrien aufgezeigt, die sich auf Gießereiprozesse übertragen ließen, aber noch nicht systematisch genutzt werden. Abschließend werden neuartige Entwicklungen aus dem Bereich der datengetriebenen Ingenieurwissenschaften aufgezeigt, deren Potenzial zur Anwendung in technischen Prozessen noch nicht ausgeschöpft wird. Das Ziel des Kapitels ist die Identifikation des Forschungsbedarfes.

In Kapitel 4 wird die Optimierungskette auf Grundlage der bis dahin gewonnenen Erkenntnisse konzipiert. Die notwendigen Maßnahmen für die Implementierung der in Abschnitt 1.2 skizzierten dreiteiligen Methodik werden schrittweise erarbeitet. Unter Berücksichtigung der prozessspezifischen Rahmenbedingungen werden die Verfahren aus Kapitel 2 und 3 so kombiniert und modifiziert, dass sie auf Gießereiprozesse angewendet werden können. Die entwickelte Methodik wird schließlich in Kapitel 5 im Rahmen einer Fallstudie mit Daten einer Schleudergießerei verifiziert. Dabei steht die Performance der Methodik bei der Anwendung auf reale Prozessdaten im Vordergrund. Die neu gewonnenen Erkenntnisse werden rückwirkend genutzt, um die Praktikabilität der Optimierungskette zu bewerten. Während zuvor nur das bekannte Fachwissen systematisiert wurde, legen die Kapitel 4 und 5 den eigenen Beitrag dieser Arbeit dar.

Die Ergebnisse werden abschließend in Kapitel 6 zusammengefasst und evaluiert. Anschließend wird ein Ausblick für zukünftige Entwicklungen formuliert.

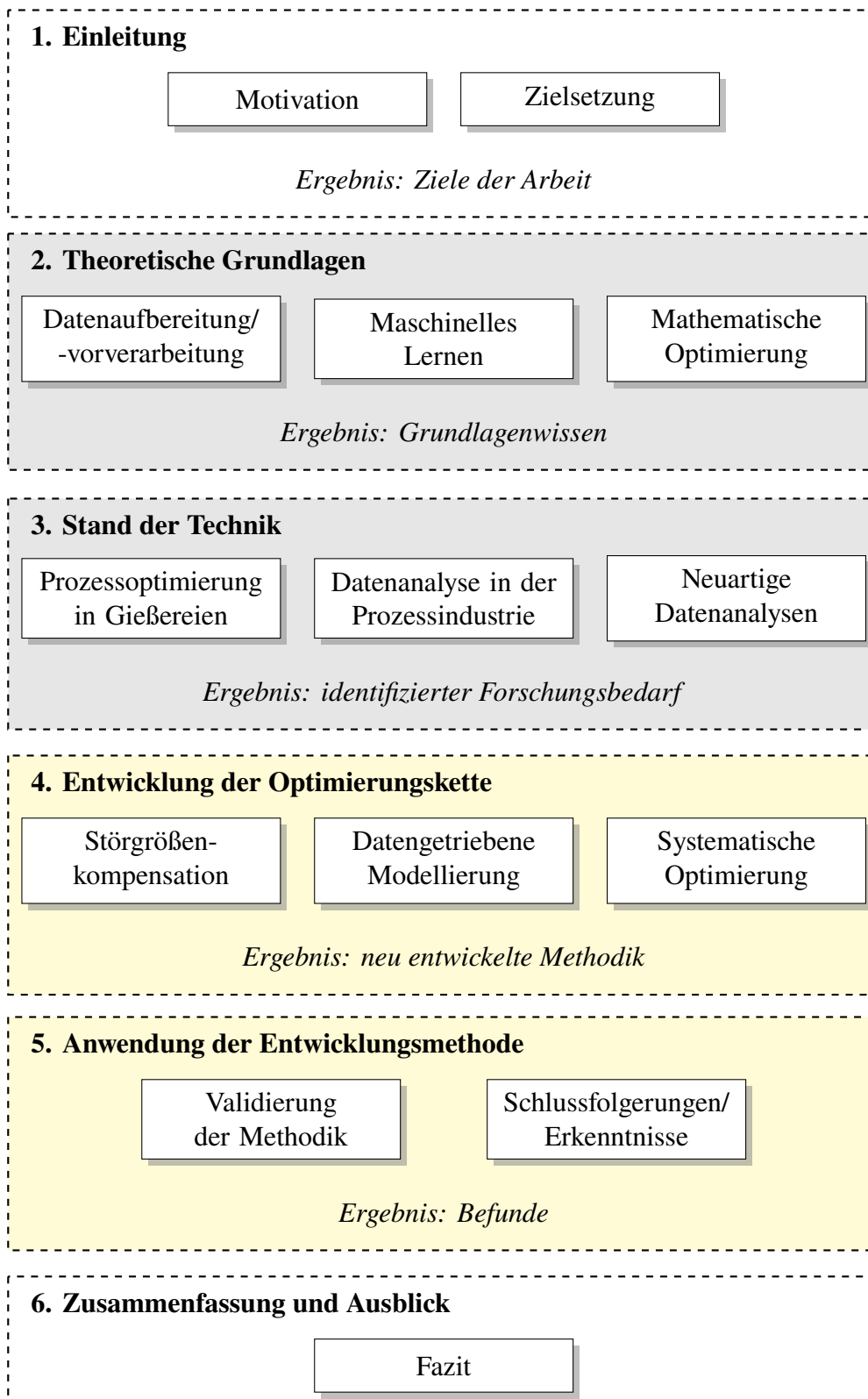


Abbildung 1.2: Aufbau der Arbeit. Die Begriffe am Ende der Kapitel beschreiben die erarbeiteten Zwischenergebnisse.

Kapitel 2

Theoretische Grundlagen

In diesem Kapitel wird das theoretische Grundlagenwissen vermittelt, das erforderlich ist, um die Funktionsweise der in dieser Arbeit zur Verwendung kommenden Verfahren zu verstehen. Zunächst erfolgt die Beschreibung des mathematischen Rahmenwerks für die Prozessoptimierung. Es folgt eine Einführung in das maschinelle Lernen als zentrales Werkzeug der Modellierung komplexer Fertigungsprozesse. Neben den wichtigsten Verfahren werden Methoden der empirischen Modellevaluation vorgestellt. Als nächstes werden Dimensionsreduktionsverfahren eingeführt, die latente Merkmale aus Daten extrahieren und es ermöglichen, den Messraum zu verkleinern.

2.1 Mathematische Optimierung

Mathematische Optimierung bildet den Grundstein für maschinelles Lernen. In diesem Abschnitt erfolgt die Definition der Problemstellungen und die Vorstellung der meistgenutzten Lösungsansätze.

2.1.1 Mehrkriterielle Optimierung

Mehrkriterielle Optimierung bezeichnet ein Optimierungsproblem mit mehreren, oft konträren Zielgrößen, die es gleichzeitig bezüglich einer Halbordnung zu minimieren gilt. Die mathematische Formulierung hierzu lautet nach Branke et al. (2008)

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{F}} \underbrace{\begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_k(\mathbf{x}) \end{bmatrix}}_{=\mathbf{f}(\mathbf{x})}. \quad (2.1)$$

Dabei sind $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ die Einzelzielfunktionen mit $k \geq 2$, die es gleichzeitig zu minimieren gilt. Der n -elementige *Entscheidungsvektor* \mathbf{x} mit den n freien Optimierungsparametern gehört zum Raum der gültigen Lösungen $\mathcal{F} \in \mathbb{R}^n$. Dieser wird durch

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m, \quad h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, p\} \quad (2.2)$$

beschrieben. Daraus ergibt sich der Abbildungsbereich der zulässigen Menge der Zielfunktionen \mathcal{Y} (Pieper 2017, S. 44).

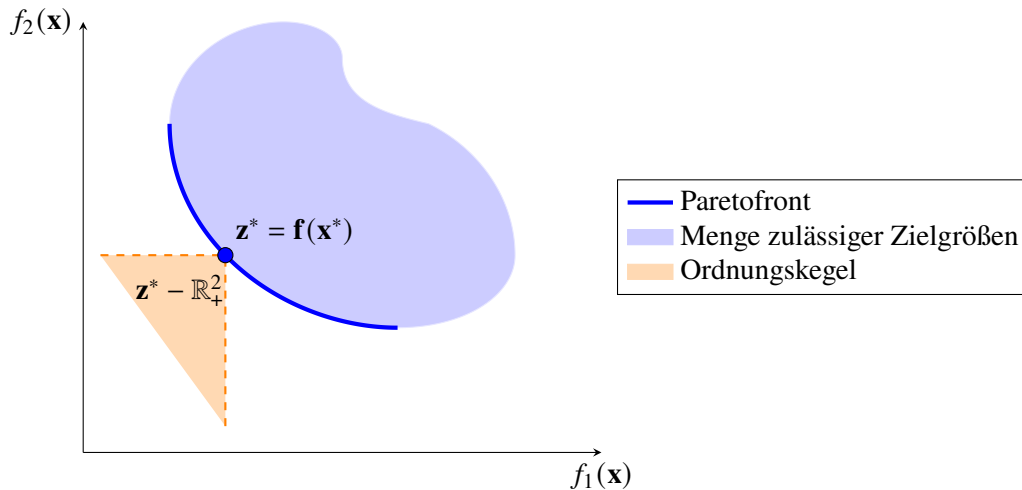


Abbildung 2.1: Paretofront am Beispiel eines zweidimensionalen Optimierungsproblems in Anlehnung an Mietinen (1998, S. 31).

Ein Zielvektor gilt als optimal, wenn keine der Komponenten $f_i(\mathbf{x})$ mehr verbessert werden kann, ohne mindestens eine der anderen Komponenten zu verschlechtern. Die Lösung $\mathbf{x}^* \in \mathcal{F}$ heißt *pareto-optimal*, wenn keine andere zulässige Lösung $\mathbf{x} \in \mathcal{F}$ existiert, sodass $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ für $i = 1, \dots, k$ und $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ für mindestens einen Index j ist (Branke et al. 2008). Die Funktionswerte $f(\mathbf{x}^*)$ aller Pareto-optimalen Punkte bilden die sogenannte *Pareto-menge* oder *Paretofront*. Abbildung 2.1 zeigt sie für ein zweidimensionales Beispiel. Mit dem Ordnungskegel

$$\mathbb{R}_+^k = \{\mathbf{z} \in \mathbb{R}^k \mid z_i \geq 0 \text{ für } i = 1, \dots, k\} \quad (2.3)$$

kann die Paretooptimalität alternativ durch

$$\mathbf{z}^* - \mathbb{R}_+^k \setminus \{0\} \cap \mathcal{Y} = \emptyset \quad (2.4)$$

definiert werden. In mancher Literatur wird die Forderung $f_i(\mathbf{x}^*) \leq f_i(\mathbf{x})$ als „utopische Lösung“ bezeichnet. Für $k = 1$ wird das mehrkriterielle zu einem einkriteriellen Optimierungsproblem und die utopische Lösung entspräche dem globalen Optimum dieser Zielfunktion. Bei $k > 1$ existiert letztere jedoch in der Regel nicht, da die Zielfunktionen zueinander im Konflikt stehen (Ngatchou et al. 2005).

Ein Lösungsverfahren für mehrkriterielle Optimierungsprobleme ist die Umwandlung in ein einkriterielles Problem, beispielsweise durch die Bildung einer gewichteten Summe. Für die Ermittlung der Paretofront haben sich genetische Algorithmen etabliert (Deb et al. 2002), die in Abschnitt 2.1.2 näher erläutert werden. Blank und Deb (2020) haben eine Python-Umgebung für mehrkriterielle Optimierung entwickelt, die verschiedene gradientenfreie Verfahren unterstützt.

Das Ergebnis einer mehrkriteriellen Optimierung ist die Menge pareto-optimaler Lösungen. Es obliegt darüber hinaus dem Anwender, Gewichtungen der einzelnen Zielgrößen vorzunehmen und ein Ranking zu erstellen, sodass eine Variante hervorgeht, die den Anforderungen der Problemstellung genügt (Ehrgott 2005, S. 6–7). Vor dem Hintergrund der Zielsetzung der Arbeit werden zahlreiche Bedingungen an die Kostenfunktion und die mit ihr bewerteten Prozessvarianten gestellt. Weiterhin muss die Substituierbarkeit der

Optimierungsalgorithmen gewährleistet werden. Aus diesen Gründen werden nachfolgend nur noch umgewandelte einkriterielle Problemstellungen berücksichtigt.

2.1.2 Gradientenfreie Optimierung

In zahlreichen Ingenieursanwendungen treten Optimierungsprobleme auf, deren Kostenfunktionen nur als Ausgang eines Black-Box-Modells zur Verfügung stehen. Sie sind meist diskret, unstetig oder rauschbehaftet und enthalten nicht genug Informationen für eine anforderungsgerechte Differenzierung. *Gradientenfreie Optimierungsverfahren* liefern in diesen Fällen zuverlässige Ergebnisse, benötigen jedoch im Regelfall mehr Rechenzeit als gradientenbasierte Ansätze.

Genetische Algorithmen

Genetische Algorithmen (GA) sind ein mathematisches Optimierungsverfahren, das den Prozess der natürlichen Auslese nachahmt und oft für maschinelles Lernen genutzt wird. Einer Population von Individuen, auch Generation genannt, wird eine Aufgabe übertragen, deren Bewertung durch eine Kostenfunktion erfolgt. Jedem Individuum wird dabei ein Grad der Fitness zugewiesen. Unter vorgegebenen Regeln schreiten erfolgreiche Handlungsstrategien bzw. Entscheidungsvektoren in die nächste Generation voran. Es hat sich in vielen Anwendungen gezeigt, dass der abzusuchende Raum häufig so groß ist, dass gradientenbasierte Verfahren keine brauchbaren Ergebnisse mehr liefern und globale Optima unter Umständen nicht mehr gefunden werden. Eine Nutzung von Monte-Carlo-Sampling wäre in diesem Fall sehr rechenintensiv (Brunton und Kutz 2019, S. 356–357).

Genetische Algorithmen sind eine Art evolutionärer Algorithmen, um Parameter in einem Input-Output-Mapping zu identifizieren. Ein Individuum repräsentiert dabei eine Parameterkombination eines zu optimierenden Modells. Bei der *genetischen Programmierung* hingegen wird die Struktur des Modells ebenfalls bei der Optimierung berücksichtigt. Für die erste Generation werden die Individuen zufällig zusammengestellt und anschließend aufgrund ihrer Performance durch die Kostenfunktion bewertet. Individuen mit geringen assoziierten Kosten haben eine höhere Chance, in die nächste Generation fortzuschreiten (Duriez et al. 2017, S. 14–15).

Die genauen Bedingungen, unter denen dies geschieht, sind wie folgt definiert:

- **Elitismus:** Einige Individuen mit einem hohen Fitnesswert gehen direkt in die nächste Generation über. So wird sichergestellt, dass sich ihre Performance nicht mehr verschlechtert.
- **Replikation:** Individuen werden mit einer Wahrscheinlichkeit proportional zu ihrer Fitness in die nächste Generation übernommen.
- **Kreuzung:** Zwei Individuen werden aufgrund ihrer Performance ausgewählt und einige zufällige Parameter ausgetauscht. Die beiden gehen mit den getauschten Werten in die nächste Generation.
- **Mutation:** Individuen gehen mit zufällig veränderten Werten in die nächste Generation. So wird sichergestellt, dass der Parameterraum hinreichend weit erforscht wird.

Nelder-Mead-Verfahren

Das *Nelder-Mead-Verfahren* oder *Downhill-Simplex-Verfahren* wurde von Nelder und Mead (1965) eingeführt und wird seitdem zur gradientenfreien nichtlinearen Optimierung eingesetzt. An die Zielfunktion werden keine Anforderungen gestellt. Gemäß des Namens wird im \mathbb{R}^n ein Simplex mit $n + 1$ Eckpunkten aufgespannt und durch die Berechnung hinweg evaluiert. Das Verfahren hat nichts mit der Simplex-Methode bei der linearen Programmierung gemeinsam (Nocedal und Wright 2006, S. 238–240). Ein Simplex beschreibt die Verallgemeinerung eines Tetraeders für den n -dimensionalen Raum und stellt in diesem das einfachste Polytop dar, dessen Eckpunkte nicht in einer Hyperebene liegen dürfen. Für den \mathbb{R}^2 führt dies beispielsweise zu einem Dreieck mit einer von Null verschiedenen Fläche (Scholz 2018, S. 120).

Das Optimierungsverfahren gibt Regeln vor, nach denen der Simplex seine Form und Position verändert, um zur besten Lösung zu gelangen. Für jeden Simplex werden verschiedene charakteristische Punkte und die dazugehörigen Funktionswerte berechnet. Anschließend wird der Punkt mit dem höchsten Zielfunktionswert durch einen der ermittelten charakteristischen Punkte ersetzt. Für gewöhnlich bewegt sich der Simplex im Raum und behält dabei seine Größe bei. Sobald er sich jedoch dem Optimum nähert, verkleinert er sich (Kochenderfer und Wheeler 2019, S. 106). Ausgehend vom Vertex mit dem höchsten Funktionswert \mathbf{x}_n^i wird der neue Punkt \mathbf{x}_n^{i+1} durch Reflexion, Expansion oder Kontraktion entlang einer Gerade gebildet, die durch \mathbf{x}_n^i und den Massenschwerpunkt des Simplex verläuft. Alternativ kann der gesamte Simplex geschrumpft werden, wenn der gespiegelte Punkt schlechter ist als die Übrigen. Die genauen Rechenvorschriften hierfür sind Scholz (2018, S. 120–121) zu entnehmen.

Für den Nutzer beschränkt sich der Aufwand bei der Optimierung durch Nelder-Mead auf einfache Funktionsauswertungen und erfordert keine Gradienten. Somit kann das Verfahren auch bei unstetigen und verrauschten Funktionen zur Anwendung kommen. Die unter Umständen langsame und nicht abschätzbare Konvergenz werden dem Verfahren jedoch als Mängel attestiert (Jarre und Stoer 2019, S. 495). Eine Abwandlung des Algorithmus unter Nutzung eines anderen Stoppkriteriums hat Powell (1998) entwickelt. Maßnahmen zur praktischen Umsetzung unter besonderer Berücksichtigung der computergestützten Berechnung finden sich in Gao und Han (2012).

Partikelschwarm-Optimierung

Die *Partikelschwarm-Optimierung* (PSO), ursprünglich vorgestellt von Kennedy und Eberhart (1995), bedient sich der Theorie des sozialen und kognitiven Verhaltens einzelner Individuen in einem Kollektiv. Auf einer hohen Abstraktionsebene beschreibt das Partikelschwarm-Modell die Fähigkeit, Zusammenschlüsse zu bilden und so Probleme zu lösen. Das Verhaltensmodell der Individuen umfasst im Wesentlichen die Prinzipien Evaluieren, Vergleichen und Nachahmen. So sind Organismen in der Lage, wichtige Merkmale ihrer Umgebung zu bewerten, ihre Performance mit der anderer zu vergleichen und ihr Verhalten bei Erfolg angemessen zu imitieren, um schwere Problemstellungen zu lösen (Kennedy et al. 2001, S. 287–289).

Bei der einfachen Partikelschwarm-Optimierung steht jeder Partikel für einen mög-

lichen Lösungsvektor des Problems. Die Position¹ $\mathbf{x}_i(t)$ des Partikels i zum diskreten Zeitpunkt t wird durch Addition einer Geschwindigkeit \mathbf{v}_i verändert:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (2.5)$$

Der Geschwindigkeitsvektor berechnet sich durch

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1r_1(t)(\mathbf{y}_i(t) - \mathbf{x}_i(t)) + c_2r_2(t)(\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)). \quad (2.6)$$

Er enthält eine *kognitive Komponente* mit Gewichtung c_1 , die die beste Position der Vergangenheit beinhaltet, und eine *soziale Komponente* mit Gewichtung c_2 , die Informationen über die Partikel in der unmittelbaren Nachbarschaft verarbeitet. Je nach Ausführungsvariante können Partikel lokal oder global miteinander verglichen werden. r_1 und r_2 sind Zufallswerte, die der Bewegung einen stochastischen Charakter verleihen. \mathbf{y}_i ist die beste gefundene Position des Partikels und $\hat{\mathbf{y}}_i$ die des Schwarms. w beschreibt die Trägheit des Partikels (Engelbrecht 2007, S. 289–293). Eine Übersicht über die Anwendungsgebiete bietet Poli (2008). Über die Konvergenz lassen sich im Vorhinein nur begrenzt Aussagen treffen. Umfangreiche Untersuchungen zu notwendigen und hinreichenden Bedingungen an die Schwarmparameter wurden hierzu von Schmitt (2015) durchgeführt. Ein Python-Interface mit dem Namen *PySwarms*, das auf der Nomenklatur von Engelbrecht (2007) aufbaut, wurde von Miranda (2018) entwickelt.

2.2 Grundlagen des maschinellen Lernens

In diesem Abschnitt werden die wichtigsten Aspekte des maschinellen Lernens hinsichtlich Strategien zur Anwendung, Bewertungskriterien für trainierte Prognosefunktionen und Voraussetzungen für ihre effektive Nutzbarkeit erläutert. Des Weiteren werden einige wichtige Methoden zur empirischen Evaluation von Black-Box-Modellen vorgestellt, die erlernte Wirkzusammenhänge offenlegen.

2.2.1 Paradigmen des maschinellen Lernens

Als maschinelles Lernen werden Algorithmen bezeichnet, die Informationen aus Daten extrahieren und verarbeiten. Sie ermöglichen die Automatisierung von Aufgaben und die Erweiterung von Domänenwissen (Brunton et al. 2020). Es wird zwischen drei Lernparadigmen unterschieden, die im Folgenden näher erläutert werden. Abbildung 2.2 zeigt Beispiele für die Umsetzungen.

Überwachtes Lernen

Überwachtes Lernen kann als Prozess verstanden werden, in dem die Beziehungen zwischen den Ein- und Ausgangsgrößen sowie der Parameter eines Systems anhand einer begrenzten Anzahl von Beobachtungen statistisch ermittelt und zu einem Modell formuliert werden (Cherkassky und Mulier 2007, S. 21). Um die Genauigkeit der Modellprognose zu optimieren, werden die prognostizierten stetig mit den wahren Werten verglichen. Der

¹Abweichende Formelzeichenverwendung für \mathbf{x} , \mathbf{y} und \mathbf{v}

Lernprozess besteht darin, den Fehler zwischen Prognose und wahrem Wert so weit wie möglich zu minimieren. Im Falle der Klassifikation wird die Klassenvariable auch als *Label* bezeichnet.

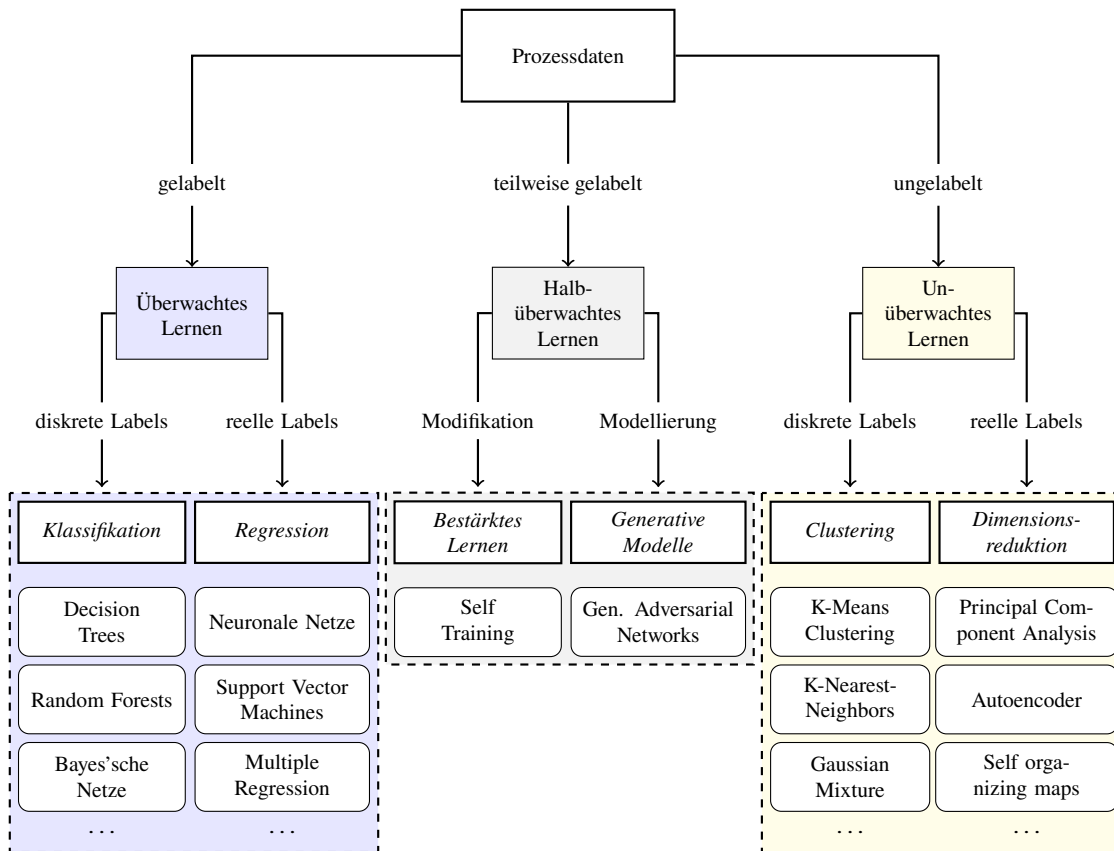


Abbildung 2.2: Eine Auswahl der Methoden des maschinellen Lernens in der Prozessindustrie in Anlehnung an (Ge et al. 2017; Brunton et al. 2020).

Unüberwachtes Lernen

Zweck des *unüberwachten Lernens* ist es, verborgene Muster oder innere Strukturen in Daten zu finden (Russell und Norvig 2016). Damit sollen Rückschlüsse gezogen werden, wenn keine expliziten Ausgangsvariablen vorliegen, beispielsweise weil ihre Ermittlung zu viele Ressourcen in Anspruch nähme. Ein wichtiges Verfahren ist das sogenannte *Clustering*, welches verwendet wird, um neuartige Muster oder Gruppierungen in Datenmengen zu entdecken (Brunton und Kutz 2019).

Halbüberwachtes Lernen

Je nach Literatur wird *Bestärktes Lernen* (engl. *Reinforced Learning*) als eigene Lernstrategie aufgeführt (Raschka und Mirjalili 2021, S. 30) oder dem *halbüberwachten Lernen* untergeordnet (Brunton et al. 2020). Im Mittelpunkt steht ein Agent in einem System, der dort eigenständig Entscheidungen trifft und sich möglichst autonom bewegen soll. Sein Ziel ist es, Aktionen zu erlernen und dabei nur gelegentlich oder ganz am Ende

Rückmeldungen zu erhalten. Prominente Beispiele sind Bots für Videospiele. Ein weiteres Themengebiet, das dem halbüberwachten Lernen zuzuordnen ist, sind generative Modelle. Sie erlernen die Verbundwahrscheinlichkeitsdichte einer gelabelten oder ungelabelten Datenmenge und sind dann in der Lage, Stichproben dieser Menge eigenständig zu generieren (Ng und Jordan 2001).

2.2.2 Modellvalidierung

Für den Lernvorgang wird die Gesamtdatenmenge in einen *Trainings-*, einen *Validierungs-* und einen *Testdatenbestand* unterteilt. Die Teildatenmengen zum Training und zur Validierung werden im Folgenden als *Lerndaten* bezeichnet. Sie dienen der Bestimmung der optimalen Hyperparameter, fungieren als Lernmaterial für den Algorithmus und werden zwecks Anpassung seiner Parameter mehrmals durchlaufen. Die Testdaten hingegen sind dem Modell zunächst völlig unbekannt und werden zu seiner finalen Bewertung herangezogen.

Klassische Validierung

Bei der *einfachen* oder *klassischen Validierung*, auch *Holdout-Methode* genannt, erfolgt die Unterteilung der Trainings- und Validierungsdaten statisch. So lässt sich ein Modell mit einer gegebenen Hyperparameterkombination trainieren und seine Performance anhand der Validierungsmenge zwischenbewerten. Erst nach abgeschlossenem Training wird das gelernte Wissen auf die Testdaten angewandt, um so die Generalisierbarkeit zu prüfen. Die Aufteilung von Trainings- zu Validierungsdaten erfolgt häufig in einem Verhältnis 70/30 und wird in der Literatur auch als *Percentage Split* bezeichnet (Witten et al. 2017, S. 559).

Kreuzvalidierung

Eine Alternative bietet die *Kreuzvalidierung*. Sie unterteilt die Gesamtdatenmenge in k Teilmengen, wovon $k - 1$ zum Training genutzt werden. Auf der übrigen Teilmenge wird dann die Validierung durchgeführt und der Fehler bestimmt. Anschließend wird der Prozess neu gestartet, bis jede Menge einmal zur Validierung genutzt wurde. Üblicherweise wird $k = 10$ gesetzt (Kohavi 1995). Wird versucht, die Klassenverteilung jedes Mal näherungsweise gleich zu halten, handelt es sich um eine stratifizierte Kreuzvalidierung (Witten et al. 2017, S. 167).

Overfitting

Während des Lernprozesses fällt der Fehler beim Training üblicherweise geringer aus als beim Validieren. Mitunter nehmen Fehler des Trainings jedoch ab, während sie beim Validieren stagnieren oder zunehmen. Der Algorithmus ist dann zu sehr auf die Trainingsdaten abgestimmt und nicht flexibel genug, unbekannte Daten zu bearbeiten oder seine Handlungslogik zu verallgemeinern. Dieses Phänomen wird *Overfitting* genannt und ist ein Indikator für die Untauglichkeit eines bestimmten Modells, den vorliegenden Sachverhalt adäquat abzubilden. Ein Beispiel hierfür ist die Annäherung eines punktwise definierten Graphs durch eine kontinuierliche Funktion. Wählt man als Approximation ein Polynom sehr hohen Grades, kann dieses die vorhandenen Punkte zwar sehr genau

abbilden, Werte außerhalb des trainierten Bereichs werden unter Umständen aber weit verfehlt. Somit ist die Komplexität des Prognosemodells in Anbetracht der Minimierung des Validierungsfehlers zu proportionieren (Bhadeshia 2009; Cawley und Talbot 2010).

2.2.3 Gütekriterien

Im Folgenden werden Gütekriterien für Prognosefunktionen bei Klassifikation und Regression vorgestellt.

Klassifikation: Konfusionsmatrix

Um das Ergebnis einer Klassifikation zur veranschaulichen, wird eine Konfusionsmatrix oder Fehlermatrix genutzt, in der richtig und falsch prognostizierte Klassen gegenübergestellt werden. Ein Beispiel für eine *Binärklassifikation*, in der nur zwei Vorhersagen getroffen werden können, zeigt Tabelle 2.1. Es liegen vier mögliche Fälle vor: *true po-*

Tabelle 2.1: Beispiel einer Konfusionsmatrix für eine Binärklassifikation

		Prognose	
		Positiv	Negativ
Messung	Positiv	TP	FN
	Negativ	FP	TN

sitives (TP) und *true negatives (TN)* sind jeweils richtige Prognosen der entsprechenden Klassen. Wurde ein positiver Datensatz als negativ klassifiziert, handelt es sich ein *false negative (FN)*. Analog werden positiv klassifizierte Datensätze, die negativ sind, *false positives (FP)* genannt (Fawcett 2006). Die *Korrektklassifikationsrate* (engl. *Accuracy*) berechnet sich durch

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \quad (2.7)$$

Sie beschreibt den Anteil aller richtig durchgeführten Prognosen (Runkler 2015, S. 91). Die Genauigkeit muss jedoch im Hinblick auf Zufallseffekte geprüft werden, da auch Standard-Klassifikatoren, die ihre Vorhersagen nur aufgrund der Häufigkeitsverteilungen der Klassen durchführen, bereits in einigen Fällen korrekt liegen. Dazu hat Cohen (1960) die *Kappa-Statistik* entwickelt, die die Vorhersagegenauigkeit um einen Erwartungswert der Zufallsvorhersage korrigiert und anzeigt, inwiefern die gewählte Klassifikationsmethode über reines Raten hinausgeht.

Der Anteil der korrekt klassifizierten positiven Instanzen wird *Trefferquote* (engl. *Recall*) genannt und berechnet sich zu

$$REC = \frac{TP}{TP + FN}. \quad (2.8)$$

Regression: Fehlersummen und Korrelation

Die grundlegenden Mechanismen der Klassifikation können auf die Regression übertragen werden. Zur Beurteilung der Prognosegüte einer Regression reicht die Feststellung, ob ein Fehler vorliegt, jedoch nicht mehr aus. Es muss untersucht werden, wie groß dieser ausfällt. Dazu werden unterschiedliche Fehlersummen herangezogen, die die Abweichung zwischen Prognose $\hat{y} = f(\mathbf{x})$ und wahrem Wert y quantifizieren (Witten et al. 2017, S. 194). Ein typisches Gütemaß ist der *Root Mean Square Error (RMSE)*, der auch für mathematische Optimierungsverfahren wie z.B. *least squares* genutzt wird, da er leicht zu handhaben ist. Er berechnet sich durch

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}. \quad (2.9)$$

Die Berechnung der Wurzel erfolgt, um bei der Prognose die gleiche Einheit wie y zu erhalten. Dabei ist m die Anzahl der Instanzen $\mathbf{x}^{(i)}$ in der betrachteten Datenmenge und f die Prognosefunktion. Die zugrunde liegende Distanzmetrik ist die ℓ_2 -Norm bzw. der euklidische Abstand. Große Ausreißer fallen somit stärker ins Gewicht und können die Interpretation erschweren (Géron 2019, S. 44–45).

Obwohl der RMSE der bevorzugte Fehlerindex für Regressionen ist, wird der *Mean Absolute Error (MAE)*

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \quad (2.10)$$

ebenfalls genutzt, vorwiegend, wenn zahlreiche Ausreißer vorliegen, da er diesen gegenüber unempfindlicher ist. Die Berechnung basiert auf der ℓ_1 -Norm. Sie wird auch *Manhattan-Distanz* genannt, da der Abstand zweier Punkte zueinander anhand der Einzelkoordinaten in einem Schachbrettmuster bestimmt wird (Kuhn und Johnson 2013, S. 159). Je höher der Index der ℓ_k -Norm,

$$\|\mathbf{y} - \hat{\mathbf{y}}\|_k = \sqrt[k]{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^k} \quad (2.11)$$

desto mehr werden starke Ausreißer bei der Fehlerberechnung berücksichtigt. Bei normalverteilten Größen, in denen Ausreißer selten auftreten, hat sich die Nutzung des RMSE jedoch gegenüber dem MAE bewährt. Beide Gütemaße haben die gleiche Größenordnung wie die vorherzusagende Variable. Sie sollten daher nur eingesetzt werden, um die Performance verschiedener Prognosetools für eine Zielgröße miteinander zu vergleichen (Hyndman und Koehler 2006). Um eine Fehlermetrik zu erhalten, die unabhängig von der Skalierung der Datenmenge ist, wird beispielsweise der *Mean Absolute Percentage Error (MAPE)*

$$MAPE = \frac{1}{m} \sum_{i=1}^m \left| \frac{y_i - \hat{y}_i}{\max(\varepsilon, |y_i|)} \right| \quad (2.12)$$

genutzt. Hierbei fungiert ε als Ersatz für den Fall, dass $y_i = 0$.

Die Vergleichbarkeit und die Robustheit der Gütemaße motivieren eine Erfassung der Prognosegenauigkeit über den statistischen Zusammenhang zwischen Prognose und Messung. Das Bestimmtheitsmaß $R^2 \in [0, 1]$ für zwei Vektoren \mathbf{x} und \mathbf{y} berechnet sich durch

$$R^2 = \frac{\sum_{i=1}^m (h(x_i) - \bar{y})^2}{\sum_{i=1}^m (y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^m (y_i - h(x_i))^2}{\sum_{i=1}^m (y_i - \bar{y})^2}. \quad (2.13)$$

Dabei beschreibt $h(x)$ ein lineares Regressionsmodell, das x auf y abbildet. Das Bestimmtheitsmaß R^2 gibt an, wie groß der Anteil der Gesamtvarianz in einer Datenmenge ist, der durch das lineare Modell erklärt werden kann. Es kann genutzt werden, um bei einer direkten Gegenüberstellung der wahren und geschätzten Werte die Qualität der Prognose zu beurteilen. $R^2 = 0$ ergibt sich genau dann, wenn konstant der Mittelwert von y geschätzt wurde. Ein nichtlinearer Zusammenhang kann in diesem Fall dennoch vorliegen. Bei einer einfachen Regression entspricht das Bestimmtheitsmaß dem Quadrat des Korrelationskoeffizienten ρ_{xy} (Mittag 2017, S. 258–260). Siehe dazu auch Abschnitt 2.4.1. Als Alternative kann die anteilig erklärte Varianz des Modells (engl. *Fraction of Variance Explained*, FVE)

$$FVE = 1 - \frac{\text{Var}\{\mathbf{y} - \hat{\mathbf{y}}\}}{\text{Var}\{\mathbf{y}\}} \quad (2.14)$$

genutzt werden.

Die Güte einer Regression kann durch die Korrelation zwischen Prognose und wahren Werten anschaulicher quantifiziert werden. Ein entscheidender Vorteil ist die Skalierungsunabhängigkeit: Wird eine der Spalten mit einem konstanten Faktor multipliziert, ändert sich die Korrelation nicht. Die zuvor behandelten Gütekriterien reagieren hierauf deutlich empfindlicher (Witten et al. 2017, S. 196). Dies ist einerseits ein Vorteil hinsichtlich der Robustheit. Andererseits kann das Bestimmtheitsmaß so auch über eine hohe Abweichung der Prognose gegenüber den realen Messungen hinwegtäuschen, wenn zwischen den Punkten ein starker linearer Zusammenhang besteht. In diesem Fall gibt es die Möglichkeit, die Verteilungsdichte der Residuen mit einzubeziehen. Sie entsprechen im Idealfall einer symmetrischen mittelwertfreien Normalverteilung mit geringer Varianz (Bangert 2021, S. 29–31).

2.2.4 Featureauswahl

Die Prognosegüte maschineller Lernmodelle hängt maßgeblich von der Auswahl der funktional unabhängigen Variablen, auch Features genannt, ab. Werden zu wenig Daten ausgewählt, leidet die Genauigkeit, zu große Mengen und redundante Features verlangsamen den Lernprozess. Das Ziel einer automatischen Auswahl ist es, bei n Features aus den $2^n - 1$ möglichen Untermengen diejenige zu finden, die die Grundverteilung so gut wie möglich widerspiegelt. Die nachfolgenden Ansätze gehen von einer initialen Variablenauswahl aus, die hinsichtlich ihrer Relevanz für das Modell bereits bewertet wurde. Bei der schrittweisen

Vorwärtsselektion wird mit einer leeren Menge angefangen die besten Features hinzugenommen. Analog werden bei der *Rückwärtseliminierung* die schlechtesten Features nach und nach entfernt. Eine gleichzeitige Durchführung ist ebenfalls möglich. Bei baumbasierten Modellen kann die Relevanz eines Merkmals an der Höhe des zugehörigen Asts abgelesen werden. Je wichtiger ein Merkmal für die Entscheidung zur Unterteilung der Daten ist, desto höher befindet sich der jeweilige Zweig (Han und Kamber 2012, S. 103–105). Eine genaue Erläuterung zur Funktionsweise von Entscheidungsbäumen folgt in Abschnitt 2.3.3.

2.2.5 Empirische Modellinterpretation

Einige Verfahren des maschinellen Lernens liefern Prognosefunktionen, die sich intuitiv visualisieren lassen. So werden beispielsweise erlernte kausale Zusammenhänge bei Modellen mit Baumstruktur in einer für den Menschen leicht verständlichen Form dargestellt. Maschinelle Lernalgorithmen mit sogenannten *Black-Box-Modellen* treffen häufig genauere Vorhersagen, bieten aber kaum Möglichkeiten zur Interpretation bei höherdimensionalen Datenmengen (Goldstein et al. 2014). In der Praxis häufig eingesetzte Verfahren zur Modellinterpretation sind die partielle Abhängigkeit, die Featurerelevanz und SHAP-Werte (Molnar et al. 2021). Sie werden im folgenden Abschnitt näher erläutert.

Statistische Versuchsplanung

Die *statistische Versuchsplanung* (engl. *Design of Experiments*, DoE) beschäftigt sich mit der Planung und Analyse von physikalischen oder digitalen Experimenten. Es gilt, den Einfluss der Parameter \mathbf{X} auf den Modellausgang \mathbf{Y} zu untersuchen, wenn das Modell komplex oder gar unbekannt ist. Dieses Ziel soll durch eine zufällige und zugleich homogene Analyse des Raumes, den die Parameter und ihre Grenzen aufspannen, erreicht werden (Braunholz 2021, S. 25–26). Dabei soll die Zahl der durchzuführenden Experimente möglichst klein gehalten werden (Bangert 2012, S. 34). Wird das Konzept auf technische Prozesse übertragen, ist die Wahl der Prozessparameter so vorzunehmen, dass sich die Zielgrößen stets innerhalb einer gegebenen Toleranz befinden, so wenig Schwankungen wie möglich unterliegen und der Einfluss von Störgrößen minimiert wird (Montgomery 2017, S. 3).

Partielle Abhängigkeit

Eine Methode der Modellauswertung, die den Einfluss einzelner funktional unabhängiger Variablen untersucht, ist die *partielle Abhängigkeit* von Friedman (2001). Sie wertet eine Prognosefunktion durch speziell gewählte Parametervariationen so aus, dass die Beziehung zwischen funktional abhängigen und unabhängigen Größen isoliert betrachtet werden kann.

Sei X_S die Menge der zu untersuchenden funktional unabhängigen Variablen und X_C die Komplementmenge dazu, sodass $X_S \cup X_C$ den gesamten Variablenraum beschreibt. Dann ist die partielle Abhängigkeit $pd_S : \mathbb{R} \rightarrow \mathbb{R}$ der Prognosefunktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$

gegenüber den Features in X_S definiert durch

$$pd_S(x_S) := \mathbb{E}_{X_C} \{f(x_S, X_C)\} = \int_{\mathbb{R}^{n-1}} f(x_S, v) p_{X_C}(v) dv \quad (2.15)$$

mit $x_S \in \mathbb{R}$ und $v \in \mathbb{R}^{n-1}$. Da die wahre Wahrscheinlichkeitsdichte p_{X_C} in der Regel nicht bekannt ist, wird die Schätzung

$$pd_S(x_S) \approx \frac{1}{m} \sum_{i=1}^m f(x_S, \mathbf{x}_C^{(i)}) \quad (2.16)$$

genutzt. Dabei ist $\mathbf{x}_C^{(i)} \in \mathbb{R}^{n-1}$ der i -te Zeilenvektor der Datenmatrix bezüglich der Menge X_C und m die Anzahl der Zeilen. Die Bestimmung eines Wertes von $pd_S(x_S)$ erfolgt durch die Mittelwertbildung der Prognose und aller m Parametervariationen von X_C in \mathbf{X} . Der Rechenaufwand steigt bei p diskretisierten Werten von X_S mit $O(m \cdot p)$, allerdings lässt sich der Prozess parallelisieren (Greenwell 2017).

Kerst et al. (2018a) nutzen den Ansatz der *Sensitivitätsanalyse* zur Modellevaluation. Dazu werden die funktional unabhängigen Variablen in X_C auf einen festen Wert $\bar{\mathbf{x}}_C$ wie beispielsweise Median oder Mittelwert gesetzt. Anschließend wird der Verlauf der Sensitivitätsfunktion $f_S : \mathbb{R} \rightarrow \mathbb{R}$ der Prognose

$$f_S(x_S) := f(x_S, \bar{\mathbf{x}}_C) \quad (2.17)$$

für p Variationen von x_S untersucht. Der Vorteil liegt in der einfachen Berechnung, da für jeden Wert von x_S nur eine Prognose durchgeführt werden muss. Somit beträgt der Rechenaufwand $O(p)$. Zur Beurteilung der Featurerelevanz kann für beide Ansätze die *Schwankungsbreite*, also die Differenz zwischen Minimum und Maximum von f_S , herangezogen werden. So kann festgestellt werden, welches Feature die funktional abhängige Variable am stärksten beeinflusst.

Die Aussagekraft von f_S variiert mit der Verteilung der funktional unabhängigen Variablen in X_S . Abschnitte mit wenigen zugrunde liegenden Daten eignen sich nicht zur Interpretation des Modells, da viele Lernalgorithmen nicht in der Lage sind, präzise zu extrapolieren. Weiterhin ist die Annahme der Unabhängigkeit zwischen den Features notwendig für die Untersuchung. Liegen jedoch Wechselwirkungen zwischen Einflussgrößen vor, liefert die partielle Abhängigkeit irreführende Ergebnisse (Goldstein et al. 2014). Um dem entgegenzuwirken, eignen sich Verfahren der Dimensionsreduktion, die in Kapitel 2.4 behandelt werden.

Permutationsrelevanz

Eine Methodik, die keinen expliziten Zusammenhang zwischen Zielgröße und zu untersuchendem Feature in Form einer partiellen Abhängigkeit erfordert, ist die von Breiman (2001) für *Random Forests* vorgestellte *Permutationsfeaturerelevanz* (engl. *Permutation Feature Importance*, PFI). Ihr liegt die Abnahme in der Prognosegüte eines Modells zugrunde, im Falle, dass die Spalteneinträge einer funktional unabhängigen Variable zufällig miteinander vermischt (permutiert) werden. Das Verfahren ist modellunabhängig auf alle trainierten Prognosefunktionen anwendbar und eignet sich insbesondere für nichtlineare

Modelle des maschinellen Lernens. Der Zusammenhang zwischen Feature und Modelausgang wird dabei so verfälscht, dass die Abhängigkeit des Modells vom jeweiligen Feature untersucht werden kann. Gegenstand der Untersuchung ist nicht die physikalische Bedeutung eines Parameters, sondern nur seine Wichtigkeit für das jeweilige Modell. Auch hier tritt die Problematik von korrelierenden Einflussgrößen auf. Selbst im Falle der Permutation hat die Prognosefunktion durch die Korrelation immer noch indirekten Zugriff auf das zu analysierende Feature, sodass beide Features als weniger wichtig gewertet werden, als sie es tatsächlich sind (Pedregosa et al. 2011). Modellagnostische Interpretationsmethoden können zu Fehlinterpretationen führen, wenn sie falsch eingesetzt werden. Neben Multikollinearität der Variablen sind Rauschen, die Dimensionalität des Datenraumes und die Generalisierbarkeit des Modells zu berücksichtigen (Molnar et al. 2021). Ein alternativer Ansatz, der die Konfidenzintervalle berücksichtigt, wurde von Altmann et al. (2010) entwickelt.

SHAP-Werte

Shapley Values, eine Methode, die den Beitrag jedes Features zu einer einzelnen Vorhersage bemisst, wurde von Shapley (1953) entworfen. Die Idee entstammt der Spieltheorie und versucht, die Differenz einer Prognose zum Durchschnittswert der abhängigen Variablen den Features anteilig zuzuordnen. Eine auf diesem Verfahren basierende modellagnostische Methode für Verfahren des maschinellen Lernens sind *Shapley Additive Explanations* (SHAP) von Lundberg und Lee (2017). Die Verteilung der Featurerelevanzen erfolgt linear und additiv anhand einer sogenannten Koalition von Features. Letztere wird so zusammengestellt, dass die aggregierten Beiträge der finalen Prognose entsprechen. Aufgrund recheneffizienter Implementierungen ermöglichen SHAP-Werte die gleichzeitige globale und lokale Interpretation eines Modells basierend auf Auswertungen auf Instanzebene. Aus Sicht des Anwenders wird die visuelle Bewertung lokaler Einflüsse unter Berücksichtigung der Verteilung für mehrere unabhängige Variablen zugleich zugelassen.

2.2.6 Mehrgrößenprognosen

Eine Übersicht der englischsprachigen Begriffsbildung bei Mehrgrößenprognosen wird durch Pedregosa et al. (2011) gegeben. Die Unterscheidung der Problemstellungen erfolgt nach Anzahl und Kardinalität der Zielgrößen und ist in Tabelle 2.2 gezeigt.

Tabelle 2.2: Begriffsbildung zu Mehrgrößenprognosen aus Pedregosa et al. (2011)

	Zielgrößen	Kardinalität
Multiclass classification	1	>2
Multilabel classification	>1	2
Multiclass-multioutput classification	>1	>2
Multioutput regression	>1	reell

Für Klassifikationsprobleme mit mehrdimensionalen Labels werden *Klassifikationsketten* genutzt. Sie verbinden einzelne Prognosetools miteinander, um eine gerichtete

Struktur zu bilden, sodass Vorhersagen eines Modells zu Features eines anderen werden. Die Performance der verketteten Klassifikation hängt maßgeblich von ihrer Topologie ab, welche mit Methoden analog zu Bayes'schen und neuronalen Netzen optimiert werden kann (Read et al. 2011). Bei der Mehrgrößenregression können wie bei der mehrkriteriellen Optimierung lokale Transformationsmethoden eingesetzt werden, die die Aufgabenstellung in ein einkriterielles Problem umwandeln. Alternativ lassen sich Algorithmen mit eindimensionalem Ausgang so adaptieren, dass sie dazu befähigt, mehrdimensionale Zielgrößen und Abhängigkeiten zwischen ihnen vorherzusagen. Die meistgenutzten Verfahren sind eine abgewandelte Support-Vector-Regression und mehrdimensionale Entscheidungsbäume. Letztere haben gegenüber Einzelmodellen den Vorteil, dass sie schlanker aufgebaut werden können und Kollinearitäten besser abbilden (Borchani et al. 2015).

2.3 Statistische und maschinelle Lernmodelle

Während die statistische Modellierung sich mit der Formalisierung von Beziehungen zwischen Variablen befasst, liegt der Schwerpunkt beim maschinellen Lernen in der Entwicklung prädiktiver Modelle und der Maximierung ihrer Genauigkeit. In diesem Abschnitt werden ausgewählte Implementierungen beider Verfahren vorgestellt, die in dieser Arbeit Verwendung finden. Es werden die lineare Regression, neuronale Netze sowie Entscheidungsbäume und ihre Modifikationen beschrieben. Für Informationen zu anderen Methoden wird an dieser Stelle auf die Fachliteratur, beispielsweise Witten et al. (2017) oder Hastie et al. (2017) verwiesen.

2.3.1 Lineare Regression

Die lineare Regression beschreibt als eines der grundlegendsten Verfahren der statistischen Modellierung eine ℓ_2 -optimale Lösung des überbestimmten Problems²

$$\mathbf{Ax} + \boldsymbol{\varepsilon} = \mathbf{y} \quad (2.18)$$

mit einer Datenmatrix \mathbf{A} , einem Koeffizientenvektor \mathbf{x} , dem Rauschen $\boldsymbol{\varepsilon}$ und einem Modellausgang \mathbf{y} . Die Minimierung der Kostenfunktion

$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{y}\|_2 \quad (2.19)$$

wird durch Bildung der Pseudoinversen von \mathbf{A} mit

$$\mathbf{x} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{y} \quad (2.20)$$

realisiert. Eine Bedingung für die Schätzbarkeit von \mathbf{x} ist, dass \mathbf{A} den vollen Spaltenrang n besitzt, da sich $\mathbf{A}^T \mathbf{A}$ sonst nicht invertieren ließe. Es kann nachgewiesen werden, dass die Schätzung erwartungsgetreu ist, sofern keine linearen Abhängigkeiten unter den Variablen bestehen. Die Vorteile der Lösung liegen in der einfachen analytischen Handhabbarkeit des Modells und dem beschränkten Rechenaufwand.

²Abweichende Notation in diesem Abschnitt.

Tabelle 2.3: Übersicht wichtiger Kostenfunktionen und Korrekturterme für die lineare Regression.

Name	Kostenfunktion
Least Squares	$J(\mathbf{x}) = \ \mathbf{Ax} - \mathbf{y}\ _2$
Ridge-Regression (Hoerl und Kennard 1970)	$J(\mathbf{x}) = \ \mathbf{Ax} - \mathbf{y}\ _2 + \alpha\ \mathbf{x}\ _2$
LASSO (Tibshirani 1996)	$J(\mathbf{x}) = \ \mathbf{Ax} - \mathbf{y}\ _2 + \lambda\ \mathbf{x}\ _1$
Elastic Net (Zou und Hastie 2005)	$J(\mathbf{x}) = \ \mathbf{Ax} - \mathbf{y}\ _2 + \lambda\ \mathbf{x}\ _1 + \alpha\ \mathbf{x}\ _2$

Um die Form des Lösungsvektors zu beeinflussen, können der Kostenfunktion Strafterme hinsichtlich der ℓ_1 - oder der ℓ_2 -Norm von \mathbf{x} hinzugefügt werden. Tabelle 2.3 zeigt eine Auswahl möglicher Maßnahmen. Bei der *Ridge-Regression* (Hoerl und Kennard 1970), auch *Tikhonov-Regularisierung* genannt, wird ein ℓ_2 -Korrekturterm $\alpha\|\mathbf{x}\|_2$ zugeführt, der die Größe der Koeffizienten der Lösung begrenzen soll. Er gewährt Robustheit gegenüber Kollinearitäten innerhalb der unabhängigen Variablen und kann nachweislich den Validierungsfehler senken (Bishop 2006, S. 11). Mit Hilfe des *Least Absolute Shrinkage and Selection Operator* (LASSO) nach Tibshirani (1996) kann die Modellkomplexität durch einen ℓ_1 -Normalisierungsfaktor $\lambda\|\mathbf{x}\|_1$ beschränkt werden. Durch LASSO tendieren Lösungen \mathbf{x} dazu, weniger von Null verschiedene Einträge zu besitzen und somit die Zahl der relevanten Features zu verringern sowie Overfitting zu verhindern. Beim *Elastic Net* werden ℓ_1 - und ℓ_2 -Strafterme miteinander kombiniert (Zou und Hastie 2005).

2.3.2 Neuronale Netze

Neuronale Netze sind eine modellbasierte Simulation des menschlichen Nervensystems. Dieses ist aus Zellen aufgebaut, die im Modell Neuronen genannt werden. Biologische Neuronen sind über Synapsen miteinander verbunden. Ein künstliches neuronales Netz lernt, indem die Stärke der Verbindungen verändert wird. Die Charakterisierung neuronaler Netze erfolgt durch seine Topologie, den Aufbau der Neuronen und die Lernregeln (Aggarwal 2015, S. 236). Für den Aufbau eines Netzes werden mehrere Neuronen übereinander angeordnet und in Schichten organisiert. Es gibt folglich eine Eingabeschicht, eine Ausgabeschicht und dazwischen eine oder mehrere verdeckte Schichten. Das Verhalten eines einzelnen Neurons i zur Erzeugung des Ausgangssignals

$$y_i = N_{\mathbf{w}_i, f_a}^i(\mathbf{x}, \theta_i) \in \mathbb{R} \quad (2.21)$$

mit dem Input x_j kann durch einen Gewichtungsfaktor \mathbf{w}_i , einen Schwellwert bzw. Bias θ_i und die Aktivierungsfunktion f_a beschrieben werden. Das Ausgangssignal y_i wird als Eingang für eine weitere Einheit verwendet oder ist eine Komponente der Ausgabe des Netzes. Es berechnet sich durch

$$y_i = f_a \left(\sum_{j=1}^n x_j w_{ij} + \theta_i \right). \quad (2.22)$$

Je größer der Betrag von w_{ij} ist, desto höher ist der Einfluss des sendenden Neurons. Um das Netz für vorgegebene Ein- und Ausgangsdaten zu trainieren, werden die Gewichte

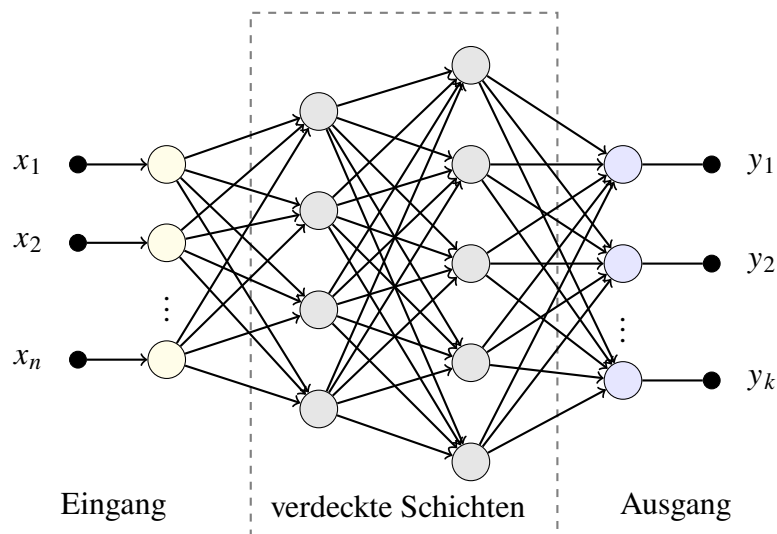


Abbildung 2.3: Beispiel eines neuronalen Netzes mit n Eingängen, k Ausgängen und zwei verdeckten Schichten.

systematisch angepasst, bis der Vorhersagefehler minimal ist. Dies erfolgt meist durch das Gradientenabstiegsverfahren (Richter 2019, S. 230). Bekannte Implementierungen neuronaler Netze sind TensorFlow von Google Brain (Abadi et al. 2016) und PyTorch von Meta AI³.

2.3.3 Entscheidungsbäume

Entscheidungsbäume (engl. *Decision Trees*) gehören zu den grundlegendsten Verfahren des maschinellen Lernens, die sowohl Klassifikation als auch Regression ermöglichen. Sie sind einfach zu interpretieren und können theoretisch manuell entworfen werden. Weiterhin sind sie ein wesentlicher Bestandteil der später vorgestellten Ensembleverfahren, die mehrere Baumstrukturen miteinander kombinieren. Ein weiterer Vorteil von Entscheidungsbäumen ist, dass sie meist keine Datenvorverarbeitung wie Skalierung und Zentrierung erfordern. Eine Baumstruktur besteht aus mehreren Eltern- und Kindsknoten. Abgehende Äste eines Knotens repräsentieren jeweils eine weitere Unterteilung der Gesamtdaten, indem innerhalb eines Features ein Threshold gesetzt wird, der die Entscheidungsgrundlage bildet (Géron 2019, S. 177–187).

Weit verbreitete Methoden zum Entwurf von Entscheidungsbäumen sind *Classification And Regression Trees* (CART) nach Breiman (1998) sowie C4.5 (Quinlan 1993). Auf CART wird nachfolgend näher eingegangen. Diese Implementierung erlaubt nur binäre Bäume, in denen aus jedem Knoten genau zwei weitere entstehen. Die Gesamtdatenmenge mit m Instanzen wird in m_1 und m_2 große Untermengen aufgeteilt. Um diese Trennung ideal vorzunehmen ist die Kostenfunktion

$$J(k, t_k) = \frac{m_1}{m} G_1 + \frac{m_2}{m} G_2 \quad (2.23)$$

³<https://pytorch.org/>

bei Klassifikation und

$$J(k, t_k) = \frac{m_1}{m} \text{MSE}_1 + \frac{m_2}{m} \text{MSE}_2 \quad (2.24)$$

bei Regression für jedes Feature k und einen Threshold t_k zu minimieren. G_i ist dabei ein Indikator für die Verunreinigung des Knotens i , also ein quantitatives Maß für die Zahl der Falschklassifikationen, die durch die Entscheidung entstehen. Beispiele für Berechnung der Verunreinigung sind der *Gini-Index* und die *Entropie*. Für genauere Definitionen sei auf Hastie et al. (2017, S. 309) verwiesen. Für Regressionsprobleme wird stattdessen der MSE herangezogen, der die Abweichung der konstanten Schätzung zu allen Instanzen innerhalb des Knotens angibt.

Der CART-Algorithmus gehört zu den geizigen Optimierungsverfahren, da er versucht, an der Spitze des Baumes die bestmögliche Teilung der Daten vorzunehmen. Es wird aber im Nachhinein nicht überprüft, ob diese Lösung in tieferen Ebenen angesichts der Verunreinigungen zu einem globalen Optimum führt. In dieser Hinsicht einen optimalen Entscheidungsbaum zu entwerfen, ist jedoch mit einem Rechenaufwand verbunden, der exponentiell mit der Zahl der Instanzen wächst, weswegen das CART-Verfahren in der Praxis als hinreichend erachtet wird. Die maximale Tiefe des Baumes kann durch einen Hyperparameter vorgegeben werden. Alternativ wird der Algorithmus beendet, wenn die Verunreinigung bzw. der MSE der Knoten nicht mehr reduziert werden kann. Overfitting entsteht bei Entscheidungsbäumen meist durch zu dichte Aststrukturen. Sie können im Nachhinein durch *Pruning* gestutzt werden (Aggarwal 2015, S. 297–298).

2.3.4 Ensembleverfahren

Bootstrap Aggregating (kurz: *Bagging*) wurde von Breiman (1996) vorgestellt und war eines der ersten Ensembleverfahren für performantere Vorhersagemodelle. Bagging beschreibt die Vorgehensweise, in der Bootstrapping in Verbindung mit einem Regressor oder einem Klassifikator genutzt wird, um ein Ensemble zu generieren. Das Prinzip ist oben in Abbildung 2.4 gezeigt. Es wird eine feste Zahl von Stichproben aus den Originaldaten entnommen und mit jeder von ihnen ein Modell, beispielsweise ein Entscheidungsbaum, trainiert. Zur Vorhersage einer neuen Instanz werden die Einzelprognosen gemittelt (Kuhn und Johnson 2013, S. 192). Durch den Prozess der Aggregation wird die Varianz der Vorhersage reduziert. Weiterhin gewährt Bagging Stabilität für baumbasierte Modelle, die in ihrer einfachen Form empfindlich auf Schwankungen in den Eingangsdaten reagieren (James et al. 2021, S. 314).

Boosting ist eine weitere mächtige Methode, um mehrere Basisklassifikatoren, manchmal auch *Weak-Learner* genannt, miteinander zu kombinieren und so ihre Leistung signifikant zu verbessern. Ursprünglich für Klassifikationsprobleme konzipiert, kann es auch zur Regression eingesetzt werden. Prominente Beispiele, in denen Boosting genutzt wird, sind *AdaBoost* (Freund und Schapire 1997) und *Gradient Boosting* (Friedman 2001). Der wichtigste Unterschied zum Bagging ist, dass die Einzelmodelle sequentiell trainiert und dann miteinander verknüpft werden. Jedes Modell versucht dabei, den Vorhersagefehler des vorherigen Modells zu korrigieren. Im Anschluss werden die Weak-Learner zu einem Meta-Modell zusammengefügt. Bei AdaBoost werden die Teildatenmengen mit einem Gewichtungsfaktor versehen, der proportional ansteigt, wenn die dazugehörige Prognose

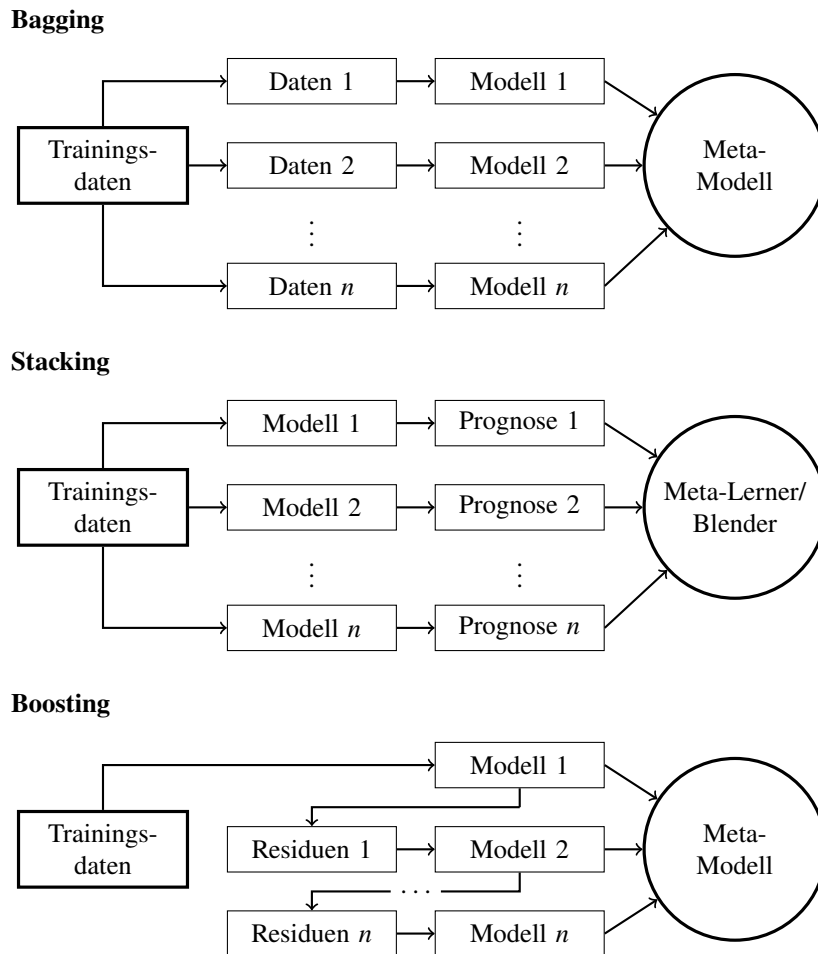


Abbildung 2.4: Vereinfachte Darstellung dreier Beispiele für Ensemblelernen. Bagging teilt die Trainingsdaten in Partitionen auf und trainiert Einzelmodelle. Stacking nutzt die Prognosen mehrerer Modelle zum Training eines Meta-Lerners. Boosting entwickelt das Meta-Modell aus der sequentiellen Anwendung einzelner Lerner.

fehlerbehaftet war (Bishop 2006, S. 657). Im Gegensatz zum Bagging hängen die Einzelmodelle, meist Entscheidungs-bäume, stark voneinander ab. Die Additivität des Modells kann auch in kleineren und weniger verflochtenen Baumstrukturen resultieren, was die Interpretation erleichtert (James et al. 2021, S. 437–348).

Stacked Generalization (kurz: *Stacking*) nach Wolpert (1992) stellt eine Alternative zur Kombination von Vorhersagemodellen dar. Es findet in der Literatur weniger Beachtung als Bagging und Boosting, da sich bislang keine generische Vorgehensweise zur Anwendung durchgesetzt hat. Im Normalfall werden keine gleichartigen Verfahren wie Entscheidungs-bäume genutzt, sondern unterschiedliche Lernansätze miteinander kombiniert. Statt eine Mehrheitsentscheidung oder eine gewichtete Mittelwertbildung zur Formung einer Meta-Prognose heranzuziehen, fungieren die Vorhersagen der Einzelmodelle als Trainingsdaten für einen *Meta-Lerner*, manchmal auch *Blender* genannt. Die Stacking-Methode erlernt somit auch, welche Modelle verlässliche Vorhersagen liefern und wie diese miteinander zu verknüpfen sind (Witten et al. 2017, S. 497–499). Für die Entwicklung der Einzelmodelle

und des Meta-Lerners werden die Trainingsdaten in zwei Partitionen unterteilt (Aggarwal 2015, S. 384).

Gradient Boosting

Bei *Gradient Boosting* handelt es sich um ein Ensembleverfahren mit dem Ziel, mehrere schwache Lerner bzw. Einzelmodelle Weak-Lerner zu einem starken Vorhersagemodell zu kombinieren. Die Vorgehensweise erfolgt iterativ und jedes Modell versucht, den gemachten Vorhersagefehler des Vorgängers zu korrigieren. Mit jeder Stufe der Optimierung wird ein neuer Lerner hinzugefügt und mit den Residuen der vorangegangenen Prognose trainiert, um den Restfehler des Ensemble-Modells zu reduzieren. Als Einzelmodelle werden häufig Entscheidungsbäume gewählt.

Die Modellbildung erfolgt in M Stufen mit einer Verlustfunktion $L(y, f(\mathbf{x}))$ für die Abweichung zwischen y und $f(\mathbf{x})$. Für jede Stufe $1 \leq i \leq M$ soll der Vorhersagefehler des Zwischenmodells $f_i(\mathbf{x})$ korrigiert werden, sodass die Prognose stückweise genauer wird. Dazu wird im Folgenden die Implementierung von Gradient Boosting anhand der Methode von Hastie et al. (2017, S. 359–361) skizziert. Zunächst wird das erste Modell

$$f_0 = \arg \min_{\gamma} \sum_{i=1}^m L\left(y_i, \gamma\left(\mathbf{x}^{(i)}\right)\right) \quad (2.25)$$

mit den Trainingsdaten aus \mathbf{X} initialisiert. Die Ermittlung der i -ten Prognosefunktion erfolgt nach

$$f_i(\mathbf{x}) = f_{i-1}(\mathbf{x}) + h_i(\mathbf{x}) \quad (2.26)$$

für $i = 1, \dots, M$. Dazu wird das hinzugenommene Modell $h_i(\mathbf{x})$ auf die Residuen bzw. die Restfehler des vorangegangenen Modells als Zielgröße trainiert:

$$h_i = \arg \min_{\gamma} \sum_{i=1}^m L\left(y_i, f_{i-1}\left(\mathbf{x}^{(i)}\right) + \gamma\left(\mathbf{x}^{(i)}\right)\right) \quad (2.27)$$

Das finale Modell $f(\mathbf{x})$ ist durch

$$f(\mathbf{x}) = f_M(\mathbf{x}) = f_0(\mathbf{x}) + \sum_{i=1}^M h_i(\mathbf{x}) \quad (2.28)$$

gegeben.

Im Rahmen verschiedener Open-Source-Projekte wurden Modifikationen der Boosting-Verfahren entwickelt, die teils eine deutliche Verbesserung der Prognoseperformance mit sich bringen. *Extreme Gradient Boosting* (kurz: *XGBoost*) ist eine frei verfügbare Bibliothek mit Gradient Boosting mit Schnittstellen zu unterschiedlichen Programmiersprachen. Sie zeichnet sich durch unter anderem durch Möglichkeiten aus, die Äste der Entscheidungsbäume effizient zu stutzen und die Trainingsvorgänge zu parallelisieren (Chen und Guestrin 2016). Eine weitere vielgenutzte Boosting-Erweiterung ist die *CatBoost*-Bibliothek der Firma Yandex. Sie unterstützt die gängigen Programmiersprachen, zeichnet sich durch die Handhabung kategorischer Features aus und bietet eine permutationsbasierte Alternative zum klassischen Algorithmus, die unempfindlich gegenüber Overfitting ist (Prokhorenkova et al. 2019). Eine weitere Implementierung mit dem Namen *LightGBM*, die auf große Datenmengen und Skalierbarkeit spezialisiert ist, wurde ursprünglich von Microsoft entwickelt (Ke et al. 2017).

Random Forests

Ein weiterer Ansatz zur Klassifikation und Regression, der auf Entscheidungsbäumen beruht, sind *Random Forests*. Sie wurden erstmalig von Breiman (2001) vorgestellt und bedienen sich der Bagging-Methode zur Kombination der Ergebnisse mehrerer Bäume. Die Klassifikation wird über eine Mehrheitsabstimmung und die Regression mit Hilfe einer Mittelwertbildung realisiert. Über Bootstrapping wird jeder Einzelbaum mit einer Teildatenmenge trainiert. Dies hat zur Folge, dass unterschiedliche Hyperparameterkombinationen entstehen und somit Overfitting vorgebeugt wird (Grus 2020, S. 221–222).

2.4 Datenvorverarbeitung und Dimensionsreduktion

Die Problematik, die hochdimensionale Vektorräume und Datenmengen hinsichtlich der Analyse mit sich bringen, wurde bereits von Bellman (1984) als Fluch der Dimensionalität grundlegend beschrieben. Aus einer Vielzahl von Attributen folgen ein hoher Rechenaufwand und damit lange Laufzeiten. Des Weiteren ist die visuelle Wahrnehmungsfähigkeit auf wenige Dimensionen beschränkt (Cleve und Lämmel 2016, S. 204). Eine Dimensionsreduktion sollte daher durch das Ausblenden von Attributen oder die Ermittlung dominanter Korrelationen erfolgen.

2.4.1 Lineare Korrelation

Für die Modellauswertung und die Interpretation von Wirkungszusammenhängen ist es prinzipiell vonnöten, dass die funktional unabhängigen Variablen nicht in einem hohen Umfang linear miteinander korrelieren. Ein Maß für die Stärke des linearen Zusammenhangs zwischen zwei Merkmalen einer Datenmenge ist der *empirische Korrelationskoeffizient*, der auch als *Bravais-Pearson-Korrelationskoeffizient* bezeichnet wird (Fahrmeir et al. 2016, S. 126–131). Er ist gegeben durch

$$\rho_{\mathbf{x}_i \mathbf{x}_j} = \frac{\sum_{k=1}^m (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^m (x_{ik} - \bar{x}_i)^2 \sum_{k=1}^m (x_{jk} - \bar{x}_j)^2}} \quad (2.29)$$

und es gilt

$$-1 \leq \rho_{\mathbf{x}_i \mathbf{x}_j} \leq 1. \quad (2.30)$$

Dabei sind \mathbf{x}_i und \mathbf{x}_j Spalten der Messdatenmatrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. Die Standardabweichungen im Nenner dienen der Normierung. Bei $\rho_{\mathbf{x}_i \mathbf{x}_j} > 0$ liegt ein gleichsinniger Zusammenhang bzw. eine *positive Korrelation* vor, $\rho_{\mathbf{x}_i \mathbf{x}_j} < 0$ bezeichnet einen gegensinnigen Zusammenhang bzw. eine *negative Korrelation* der Merkmale. \mathbf{x}_i und \mathbf{x}_j sind linear unkorreliert, wenn $\rho_{\mathbf{x}_i \mathbf{x}_j} = 0$. Eine perfekte lineare Abhängigkeit bzw. eine *vollständige Korrelation* ist bei $|\rho_{\mathbf{x}_i \mathbf{x}_j}| = 1$ gegeben, wobei diese in der Realität aufgrund unvorhersehbarer und nicht gemessener Einflüsse nur selten vorliegt. Nichtlineare Zusammenhänge werden vom Korrelationskoeffizienten nur schlecht erfasst, sodass auch kleine $\rho_{\mathbf{x}_i \mathbf{x}_j}$ nicht zwingend eine statistische Unabhängigkeit ausdrücken (Mittag 2017, S. 126–129).

2.4.2 Singulärwertzerlegung

Bei der *Singulärwertzerlegung* (engl. *Singular Value Decomposition*, SVD) handelt es sich um eine numerisch stabile Matrixzerlegung, in der sich die sogenannten *Singulärwerte* ablesen lassen. Die ist SVD datengetrieben und auf das jeweilige Problem zugeschnitten. Darüber hinaus lässt sie sich einfach mit Methoden der linearen Algebra interpretieren (Brunton und Kutz 2019, S. 3).

Für jede Matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ existiert eine eindeutige SVD, sodass

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \quad (2.31)$$

Die Matrix $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ besitzt nur nichtnegative Einträge und enthält auf der Hauptdiagonalen der oberen $n \times n$ -Untermatrix die *Singulärwerte* $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ der Matrix \mathbf{X} und außerhalb Nullen. $\mathbf{U} \in \mathbb{R}^{m \times m}$ und $\mathbf{V} \in \mathbb{R}^{n \times n}$ sind orthogonale Matrizen, da \mathbf{X} als reell angenommen wird. Es gilt

$$\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_{m \times m} \quad (2.32)$$

$$\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_{n \times n}. \quad (2.33)$$

Die Quadrate der Singulärwerte von \mathbf{X} sind die Eigenwerte von $\mathbf{X}^T\mathbf{X}$. Der Rang von \mathbf{X} entspricht der Zahl der von Null verschiedenen Singulärwerte.

Die Tatsache, dass \mathbf{U} und \mathbf{V} unitär sind, ist für die Anwendung der *Dimensionsreduktion* von entscheidender Bedeutung. Die Spalten von \mathbf{U} werden *linke Singulärvektoren* und die Spalten von \mathbf{V} *rechte Singulärvektoren* genannt:

$$\mathbf{X} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_m \\ | & | & & | \end{bmatrix}}_{=\mathbf{U}} \underbrace{\begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_n \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}}_{=\mathbf{\Sigma}} \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \\ | & | & & | \end{bmatrix}^T}_{=\mathbf{V}^T} \quad (2.34)$$

In technischen Prozessen liegen häufig *schlanke* Matrizen mit $m \gg n$ vor, sodass die SVD von \mathbf{X} umgeschrieben werden kann als

$$\mathbf{X} = \begin{bmatrix} \hat{\mathbf{U}} & \mathbf{U}^\perp \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Sigma}} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^T. \quad (2.35)$$

Diese Darstellung wird auch *ökonomische Variante* der SVD genannt und repräsentiert die Matrix \mathbf{X} exakt. Sie ist aufgrund der Reduktion von \mathbf{U} auf $\hat{\mathbf{U}}$ effizienter zu handhaben. Für die Berechnung der SVD existieren numerische Verfahren (Golub und van Loan 2013, S. 486–494). Sie sind aufwendig und erfolgen vor allem bei großen Messwertmatrizen computergestützt. Viele Programmiersprachen, wie beispielsweise Python, erlauben die Nutzung der ökonomischen Variante (Virtanen et al. 2020).

Matrix-Näherung

Eine nützliche Eigenschaft der SVD ist, dass sie eine rangreduzierte Näherung von \mathbf{X} bietet. Die Rang- n -Matrix aus Gleichung (2.34) lässt sich als Summe von n Rang-1-Matrizen $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ darstellen. Werden jedoch nur die ersten r Summanden berücksichtigt, ergibt sich die rangreduzierte Matrix

$$\tilde{\mathbf{X}} := \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T \quad (2.36)$$

mit $\text{rank}(\tilde{\mathbf{X}}) = r$. Ziel ist es, ein geeignetes r zu finden, sodass $\tilde{\mathbf{X}} := \tilde{\mathbf{U}} \tilde{\Sigma} \tilde{\mathbf{V}}^T$ alle dominanten Singulärwerte enthält. Es wurde nachgewiesen, dass die rangreduzierte Matrix $\tilde{\mathbf{X}}$ die beste Rang- r -Näherung für \mathbf{X} liefert (Eckart und Young 1936):

$$\arg \min_{\text{rank}(\tilde{\mathbf{X}})=r} \|\mathbf{X} - \tilde{\mathbf{X}}\|_F = \tilde{\mathbf{U}} \tilde{\Sigma} \tilde{\mathbf{V}}^T \quad (2.37)$$

Dabei enthalten $\tilde{\mathbf{U}}$ und $\tilde{\mathbf{V}}$ die ersten r Spalten von \mathbf{U} bzw. \mathbf{V} . $\tilde{\Sigma}$ ist der $r \times r$ -Unterblick von Σ . $\|\cdot\|_F$ ist die Frobenius-Norm. Anschaulich werden beim Abschneiden der Terme die ersten r Singulärwerte behalten, während die restlichen auf Null gesetzt werden. Die Rangreduktion ist dabei nicht in den Dimensionen von $\tilde{\mathbf{X}}$ sichtbar, sondern zeigt sich in den gestutzten Matrizen $\tilde{\mathbf{U}}$, $\tilde{\Sigma}$ und $\tilde{\mathbf{V}}^T$.

2.4.3 Hauptkomponentenanalyse

Die *Hauptkomponentenanalyse* (engl. *Principal Component Analysis*, PCA) ist eine der häufigsten Nutzungen der SVD. Sie wurde erstmals von Pearson (1901) als Minimierung des Abstands der Datenpunkte zu ihrer Projektion in einen niedrigdimensionalen Hauptkomponentenraum definiert. Unabhängig davon wurde sie von Hotelling (1933) über die Varianz der orthogonal projizierten Daten hergeleitet. Ziel ist es, die Spalten bzw. Variablen von \mathbf{X} in latente Metadaten

$$\mathbf{Z} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_n \\ | & | & & | \end{bmatrix} = \mathbf{B} \begin{bmatrix} | & | & \cdots & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_n \\ | & | & & | \end{bmatrix} \quad (2.38)$$

zu transformieren. Dabei ist \mathbf{B} die mittelwertfreie Datenmatrix

$$\mathbf{B} = \mathbf{X} - \bar{\mathbf{X}} \quad (2.39)$$

von \mathbf{X} . Die neuen Variablen \mathbf{z}_i werden *Hauptkomponenten* (Principal Components) genannt. Sie sind Linearkombinationen der ursprünglichen mittelwertfreien Spalten von \mathbf{B} . Die Gewichte der Kombinationen werden auch *Lasten* genannt und sind von null verschieden. Weiterhin erfolgt die Bestimmung der Hauptkomponenten \mathbf{z}_i so, dass sie paarweise orthogonal und nach ihrer Varianz absteigend angeordnet sind (Strang 2010, S. 106–107).

Um die Varianz der ersten Hauptkomponente zu maximieren, wird das Optimierungsproblem

$$\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|=1} \|\mathbf{B}\mathbf{w}\|_2^2 \quad (2.40)$$

gelöst. \mathbf{w}_1 entspricht damit dem ersten Singulärvektor von \mathbf{X} . Alle weiteren Komponenten \mathbf{w}_i werden durch iterative Subtraktion der Hauptkomponenten von \mathbf{B} berechnet. In der Praxis erfolgt die Berechnung der \mathbf{w}_i als die Eigenvektoren der empirischen (Spalten-) Kovarianzmatrix

$$\text{cov}(\mathbf{B}) = \frac{1}{m-1} \mathbf{B}^T \mathbf{B} \quad (2.41)$$

von \mathbf{B} .

Hinsichtlich der Nutzung der PCA in der Prozessindustrie liegen Einschränkungen vor. Prozessvariablen, die aufgrund von Nichtlinearitäten und Rückkopplungen keiner Normalverteilung folgen, beeinträchtigen die Leistung der klassischen PCA. Weiterhin stößt sie angesichts starker Ausreißer selbst bei normalverteilten Messgrößen an ihre Grenzen (Xu et al. 2015). Aus diesem Grund wird im nächsten Abschnitt eine Abwandlung der PCA vorgestellt, die eine höhere Robustheit diesbezüglich gewährt.

2.4.4 Robuste Hauptkomponentenanalyse

Optimierungsansätze hinsichtlich der ℓ_2 -Norm verhalten sich empfindlich gegenüber Ausreißern und fehlerhaften Daten. Auch die Bestimmung der Hauptkomponenten der PCA unterliegt dieser Problematik. Aus diesem Grund haben Candès et al. (2011) eine Methodik entwickelt, die eine Datenmatrix \mathbf{X} in eine Matrix \mathbf{L} mit niedrigem Rang, die die kohärenten Strukturen von \mathbf{X} enthält, und eine dünnbesetzte Matrix \mathbf{S} , die das Rauschen repräsentiert, zerlegt. Es gilt

$$\mathbf{X} = \mathbf{L} + \mathbf{S}. \quad (2.42)$$

Das mathematische Optimierungsproblem wird als

$$\min_{\mathbf{L}, \mathbf{S}} \text{rank}(\mathbf{L}) + \|\mathbf{S}\|_0, \quad \text{sodass } \mathbf{L} + \mathbf{S} = \mathbf{X} \quad (2.43)$$

formuliert. Da es sich hierbei um keine konvexe Optimierung handelt, wird eine konvexe Relaxierung vorgenommen. Gleichung (2.42) wird damit zu

$$\min_{\mathbf{L}, \mathbf{S}} \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1, \quad \text{sodass } \mathbf{L} + \mathbf{S} = \mathbf{X} \quad (2.44)$$

und konvergiert zur gleichen Lösung, wenn \mathbf{X} analog zur PCA normiert wurde und $\lambda = 1/\sqrt{\max(m, n)}$. Dabei ist $\|\cdot\|_*$ die Nuklear-Norm und fungiert als konvexe Hülle für die Rangfunktion. Für dieses spezielle Problem wurden Algorithmen entwickelt, die \mathbf{L} und \mathbf{S} iterativ ermitteln. Dazu werden zunächst ein Schrumpfoperator

$$\mathcal{S}_\tau(\mathbf{X}) = \text{sign}(\mathbf{X}) \max(|\mathbf{X}| - \tau, 0) \quad (2.45)$$

mit der elementweisen Bestimmung der Vorzeichen bzw. des Absolutbetrags und ein Singulärwert-Threshold-Operator

$$\text{SVT}_\tau(\mathbf{X}) = \mathbf{U} \mathcal{S}_\tau(\boldsymbol{\Sigma}) \mathbf{V}^T \quad (2.46)$$

mit der Singulärwertzerlegung von \mathbf{X} gebildet. Für den Algorithmus zur Bestimmung der Matrizen nach Candès et al. (2011) werden die Koeffizienten μ , λ mit

$$\mu = \frac{m \cdot n}{4 \sum_{i=1}^m \sum_{j=1}^n |x_{ij}|} \quad (2.47)$$

$$\lambda = \frac{1}{\sqrt{\max(m, n)}}, \quad (2.48)$$

der Threshold K_{th} mit

$$K_{th} = 10^{-7} \cdot \|\mathbf{X}\|_F \quad (2.49)$$

und die RPCA-Matrizen mit

$$\mathbf{L}_0 = \mathbf{S}_0 = \mathbf{Y}_0 = \mathbf{0}_{m \times n} \quad (2.50)$$

initialisiert. Anschließend erfolgt die iterative Berechnung solange $\|\mathbf{X} - \mathbf{L}_k - \mathbf{S}_k\|_F > K_{th}$ und $k < 1000$:

$$\mathbf{L}_{k+1} = SVT_{1/\mu} \left(\mathbf{X} - \mathbf{S}_k + \frac{1}{\mu} \mathbf{Y}_k \right) \quad (2.51)$$

$$\mathbf{S}_{k+1} = \mathcal{S}_{\lambda/\mu} \left(\mathbf{X} - \mathbf{L}_k + \frac{1}{\mu} \mathbf{Y}_k \right) \quad (2.52)$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu (\mathbf{X} - \mathbf{L}_{k+1} - \mathbf{S}_{k+1}) \quad (2.53)$$

Im Gegensatz zur SVD kann der reduzierte Rang $r = \text{rank}(\mathbf{L})$ nur implizit durch den Threshold vorgegeben werden. Weiterhin ist die Bestimmung der Zerlegung aufgrund der Optimierungsstrategie nur im Offlinefall möglich. Es ist für hohe Ränge r ein erheblicher Rechenaufwand zu erwarten.

Optimale Rangreduktion

Eine zentrale Fragestellung bei der Dimensionsreduktion durch SVD oder PCA besteht in der Wahl des richtigen Ranges r , sodass die Dynamik des zugrundeliegenden Systems bzw. die Gesamtvarianz der Datenmenge \mathbf{X} im notwendigen Umfang erhalten bleibt. Cattell (1966) entwickelte zu diesem Zweck den *Scree-Test*, der graphisch die Wichtigkeit der Eigenwerte der Hauptkomponenten untersucht. Dazu werden die Eigenwerte in absteigender Reihenfolge aufgetragen und nach sogenannten *Ellbogen* (Knickstellen) im Verlauf gesucht. Alle Eigenwerte rechts der Knickstelle werden als nicht relevant betrachtet und für die weitere Analyse gleich null gesetzt. Horn (1965) versuchte, durch eine *Parallelanalyse* nur solche Eigenwerte als signifikant zu klassifizieren, die größer sind als die, die einer Normalverteilung entstammen.

Anderen Verfahren wählen den Threshold r so, dass die kumulierte Energie der Eigen- bzw. Singulärwerte

$$\text{kumulierte Energie} = \frac{\sum_{i=1}^r \sigma_i}{\sum_{i=1}^n \sigma_i} \quad (2.54)$$

einen statischen Anteil, z.B. 80 %, der Gesamtvarianz abdeckt (Brunton und Kutz 2019, S. 9). An den aufgeführten Ansätzen lässt sich mangelnde Objektivität kritisieren, da sie Interpretationsspielraum geben und ihnen Sachbezug fehlt.

Gavish und Donoho (2014) wählen den numerischen Rang r anhand des Median-Singulärwerts und berechnen so einen *Optimal Hard Threshold*. Je nach Dimension von \mathbf{X} und den Kenntnissen über das Rauschen der Daten werden unterschiedliche Verfahren zur optimalen Rangreduktion genutzt.

2.4.5 Autoencoder

Bei einem *Autoencoder* handelt es sich um ein spezielles Neuronales Netz mit dem Ziel der Dimensions- und Rauschreduktion durch unüberwachtes Lernen. Sie werden manchmal autoassoziative neuronale Netze genannt (Bishop 2006, S. 592), da der Eingang und der Ausgang des Netzes gleichgesetzt werden und sie damit näherungsweise eine Identitätsabbildung darstellen (Boulevard und Kamp 1988). Strukturell unterteilen sich Autoencoder in einen Encoder und einen Decoder, die ihrerseits jeweils aus einer oder mehreren verdeckten Schichten bestehen können.

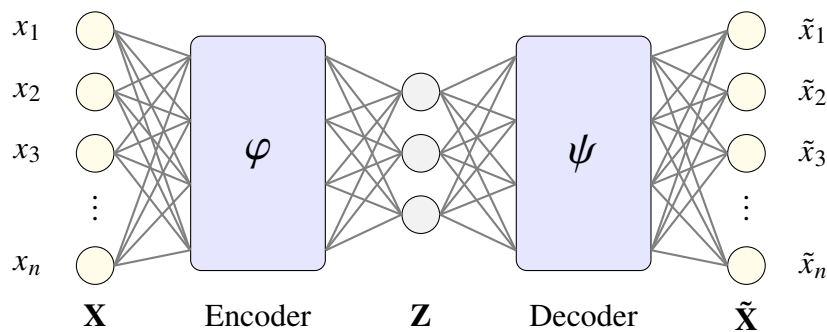


Abbildung 2.5: Beispiel für einen Autoencoder in Anlehnung an Champion et al. (2019). Der Encoder erzeugt die dimensionsreduzierte Darstellung $\mathbf{Z} = \varphi(\mathbf{X})$ von \mathbf{X} . Durch den Decoder werden die Daten $\tilde{\mathbf{X}} = \psi(\mathbf{Z})$ rekonstruiert.

Der Encoder φ erzeugt eine niedrigdimensionale Repräsentation der Daten $\mathbf{Z} = \varphi(\mathbf{X})$, die im Bottleneck \mathbf{Z} entnommen werden kann, während der Decoder ψ versucht, die Originaldaten $\tilde{\mathbf{X}} = \psi(\mathbf{Z})$ zu rekonstruieren. In seiner ursprünglichen Form und unter der Nutzung linearer Übertragungsfunktionen ähnelt die Funktionsweise der Hauptkomponentenanalyse, wobei Autoencodern eine bessere Performance bei der Dimensionsreduktion nachgewiesen werden konnten (Hinton und Salakhutdinov 2006). Ihre Stärke und Potenzial liegen jedoch in der Findung nichtlinearer Dimensionsreduktionen.

Eine Auflistung wichtiger Modifikationen von Autoencodern ist in Brunton und Kutz (2019, S. 223) gegeben. *Variational Autoencoder* (VAE) sind generative Modelle für unüberwachtes Lernen komplizierter Verteilungen. Die Optimierung erfolgt über das stochastische Gradientenabstiegsverfahren (Kingma und Welling 2013). *Denoising Autoencoder* (DAE) nehmen verrauschten Input als Trainingsdaten und versuchen, die rauschfreien Originaldaten zu rekonstruieren. Das Rauschen wird zum Trainingszweck absichtlich hinzugegeben (Vincent et al. 2008). *Sparse Autoencoder* (SAE) haben mehr versteckte Units als Eingänge, um nützliche Strukturen zu erlernen. Eine schlanke kohärente Darstellung

der Daten wird durch Thresholding der schwachen Neuronen herbeigeführt (Ranzato et al. 2006).

2.4.6 Schnelle Fourier-Transformation

Die Fourier-Transformierte $\mathcal{F}(\omega)$ einer zeitabhängigen Funktion $f(t)$ berechnet sich durch

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt. \quad (2.55)$$

Sie zerlegt ein Zeitsignal in seine Frequenzanteile und kann genutzt werden, um hochfrequente Anteile und schwache Amplitude zu identifizieren und herauszufiltern. Für diskrete Zeitsignale, also Zahlenfolgen der Form $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]$ eignet sich die diskrete Fourier-Transformation (DFT)

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{j2\pi}{N}kn} \quad (2.56)$$

Das Ergebnis ist ein N -elementiger Vektor mit den Komponenten X_k . Bei der *schnellen Fourier-Transformation* (FFT) handelt es sich um einen Verfahren, die DFT zeit- und rechenefizient mit $O(n \log(n))$ zu bestimmen (Isermann und Münchhof 2011, S. 82–86). Sie bildet die Grundlage für zahlreiche Methoden zur Datenfilterung und Signalverarbeitung. Es existieren verschiedene Realisierungen der FFT; am bekanntesten ist die Implementierung nach Cooley und Tukey (1965). Die FFT kann zur Komprimierung eines Signals genutzt werden, indem schwache Frequenzen, deren Amplitude unter einem vorgegebenen Threshold liegen, ausgeblendet werden. Anschließend wird die gefilterte FFT, die aus den verbleibenden dominanten Frequenzen besteht, in den Zeitbereich zurücktransformiert, um ein rauschreduziertes Signal zu erhalten.

2.4.7 Verdichtung des Messraums

Die bisherigen Verfahren zur Dimensionsreduktion liefern zwar eine komprimierte Darstellung großer Datenmengen bei bekannten kohärenten Strukturen, sie erfordern aber weiterhin die Messungen des gesamten Variablenraumes. Seit Mitte der 2000er Jahre werden aufgrund von Weiterentwicklungen von Verfahren in der angewandten Mathematik zunehmend Ansätze verfolgt, aus einer komprimierten Messung den ursprünglichen Datenraum zu rekonstruieren. Die grundlegende Annahme für die Anwendbarkeit ist, dass ein Signal \mathbf{x} in einer bestimmten Domäne k -dünnbesetzt ist und sich durch

$$\mathbf{x} = \Psi \mathbf{s} \quad (2.57)$$

ausdrücken lässt. Dabei ist \mathbf{s} ein dünnbesetzter Vektor mit k von null verschiedenen Einträgen. Die Basis Ψ kann aus Fourierkoeffizienten oder der Matrix \mathbf{U} der SVD bestehen. Ziel ist es, nur einen Unterraum

$$\{\mathbf{y} = \mathbf{C}\mathbf{x}\} \quad (2.58)$$

zu messen und das Ursprungssignal daraus zu rekonstruieren. Dies erfolgt durch die Zusammenführung von \mathbf{C} und Ψ zu

$$\mathbf{y} = \mathbf{C}\Psi\mathbf{s} = \Theta\mathbf{s} \quad (2.59)$$

und anschließender Lösung des Optimierungsproblems

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0, \quad \text{sodass } \mathbf{y} = \mathbf{C}\Psi\mathbf{s} \quad (2.60)$$

hinsichtlich der ℓ_0 -Norm, sodass dünnbesetzte \mathbf{s} favorisiert werden.

Zusammenfassung

In diesem Kapitel wurden die theoretischen Grundlagen beschrieben, die für die Methodenentwicklung notwendig sind. Es wurden die Grundzüge der gradientenfreien Optimierung erläutert, die es ermöglicht, Zielfunktionen ohne detaillierte analytische Kenntnisse zu minimieren. Die aufgeführten Beispiele umfassen das Nelder-Mead-Verfahren, genetische Algorithmen und die Partikelschwarmoptimierung. Sie lassen sich nutzen, um die Eingangsgrößen für Prognosefunktionen so zu variieren, bis ein vorgegebenes Ergebnis erzielt wird.

Im Rahmen der Grundlagen des maschinellen Lernens wurden allgemeine Strategien, Prinzipien der Validierung eines Modells auf Basis seiner Vorhersagen und Gütekriterien zur quantitativen Bewertung seiner Genauigkeit vorgestellt. Die empirische Modellinterpretation stützt sich auf modellagnostische Verfahren: die partielle Abhängigkeit fungiert als Erwartungswert der Prognosefunktion bei Betrachtung eines einzelnen Features, während SHAP-Werte versuchen, den Vorhersagewert den Features anteilig zuzuordnen. Beide werden genutzt, um später Haupteinflussgrößen eines Prozesses zu identifizieren und die Optimierung zu planen. Beispiele für Modelle des maschinellen Lernens wurden mit dem Schwerpunkt auf Ensembleverfahren aufgeführt. Diese verknüpfen zumeist Entscheidungsbäume.

Die Datenvorverarbeitung und die Dimensionsreduktion dienen dazu, maschinelle Lernmodelle performanter und robuster zu machen. Neben standardmäßigen Verfahren zur Rauschfilterung wurden Ansätze des unüberwachten Lernens aufgezeigt, die in der Lage sind, kohärente Strukturen aus Daten zu extrahieren. Methoden wie die RPCA entstammen neuartigen Entwicklungen der vergangenen 10 Jahre und versprechen Abhilfe bei Störgrößenkompensation zu schaffen, bei denen klassische Ansätze scheitern.

Kapitel 3

Stand der Technik und der Forschung

In diesem Kapitel wird der Stand der Technik im Bereich der Digitalisierung und der datengetriebenen Optimierungsmaßnahmen in der Prozessindustrie, insbesondere der Gießereibranche, dargestellt. Es wird erläutert, welche der im vorangegangenen Kapitel vorgestellten Verfahren die fertige Industrie bereits nutzt und welche lediglich die angewandte Forschung einsetzt. Es erfolgt zunächst eine Gegenüberstellung datengetriebener und durch Simulation gestützter Methoden zur Untersuchung von Gussfehlern. Darüber hinaus werden etablierte Anwendungsfälle und Paradigmen klassischer Datenanalyse in der Prozessindustrie gezeigt. Darauf folgt eine Übersicht datengetriebener Methoden, die bis dato nur unter Laborbedingungen und simulierten Umgebungen Einsatz finden. Abschließend werden verwandte akademische Forschungsarbeiten vorgestellt. Ziel des Kapitels ist es, ein Defizit im Stand der Technik in der Gießereiindustrie aufzuzeigen und das Potenzial sowohl neuartiger als auch bewährter Verfahren der datengetriebenen Optimierung zu erschließen.

3.1 Prozessoptimierung in der Gießereiindustrie

Im Vorfeld ist anzumerken, dass sich viele Verbesserungsmaßnahmen auf die Anlagenmodernisierung und Energienutzung konzentrieren und für diese Arbeit nicht weiter berücksichtigt werden. Die nachfolgend vorgestellten Arbeiten können in datengetriebene Fehlererkennung und -vorhersage (Abschnitt 3.1.1) sowie Bauteil-, Prozess- und Topologieoptimierung mithilfe von softwaregestützter Gießsimulation (Abschnitt 3.1.2) untergliedert werden.

3.1.1 Qualitätsvorhersage mit maschinellem Lernen

Eine Vielzahl der internationalen Veröffentlichungen zur Nutzung von maschinellem Lernen zur Gussfehlererkennung entstammt den Aktivitäten des metallurgischen Forschungsinstituts AZTERLAN und der Universität Bilbao. Die meisten Arbeiten beschäftigen sich mit der einkriteriellen Optimierung von Prozesskennwerten oder Bauteilqualitäten. Dies erfolgt chargen- oder bauteilbezogen sowohl als Klassifikation als auch als Regression. Den in der späteren Fallstudie der Dissertation verfolgten Ansatz, einzelne Bauteile virtuell und rein datengetrieben zu planen, gibt es bislang noch nicht.

Tabelle 3.1: Ausgewählte Publikationen zur Nutzung von maschinellem Lernen in der Gießereiindustrie. In manchen Veröffentlichungen fehlen vollständige Angaben zum Datenumfang. Bei der Erkennung von Oberflächenfehlern existieren keine Variablen im eigentlichen Sinne.

Publikation	Schwerpunkt	Datenumfang		Bestes Verfahren
		Zeilen	Spalten	
Gutiérrez et al. (2017)	Vorhersage von Mikrolunkern	74	7	BN (TAN)
Nieves et al. (2012)	Vorhersage von Mikrolunkern und Zugfestigkeit	951	24	BN (TAN) mit Stacking
Nieves et al. (2014)	Vorhersage von Schlackenbildung	2261	120	SVM mit Stacking
Pastor-López et al. (2015)	Erkennung von Oberflächenfehlern	6150	-	Random Forest
Pastor-López et al. (2019a)	Erkennung von Oberflächenfehlern	5785	-	Random Forest / Simple Segmentation
Pastor-López et al. (2019b)	Erkennung von Oberflächenfehlern	5785	-	k-NN ($k = 5$) / FFT
Pastor-López et al. (2021)	Erkennung von Oberflächenfehlern	5785	-	SVM / Co-Occurrence Matrix
Penya et al. (2008a)	Vorhersage von Mikrolunkern	1280 / 17.187	43 / 46	BN (einzig untersucht)
Penya et al. (2008b)	Vorhersage von Mikrolunkern	7368	k/A	BN (einzig untersucht)
Santos et al. (2013)	Vorhersage von Schlackenbildung	k/A	120	BN (TAN)
Santos et al. (2009a)	Vorhersage der Zugfestigkeit	889	25	MLP
Santos et al. (2009b)	Vorhersage von Mikrolunkern	951	24	BN (TAN)
Tonn et al. (2021)	Vorhersage von Grafitentartungen	130	30	Logistische Regression
Becker et al. (2011)	Sand-, Gas- und Fremdeinschlüsse / Formfehler	2275	460	Self Organizing Maps

Penya et al. (2008a) sagen Mikrolunker bei mit Sphäroguss gefertigten Sicherheitskomponenten für Bremssysteme der Firma Fuchosa S.L. mit maschinellem Lernen aufgrund der Prozessparameter bestehend aus Informationen der Schmelze, des Gießens und des Formstoffs vorher. Die genutzten Verfahren umfassen neuronale Netze, Support Vector Machines, Bayes'sche Netze und Random Forests. Grundlage der Versuche sind chargenweise Röntgenuntersuchungen von Einzelteilen. In weiteren Arbeiten prognostizieren sie Zugfestigkeiten gegossener Bremscheiben mit neuronalen Netzen. Prädiktive Modelle sollen diesem Fall die Zerstörungsprüfung der Einzelteile ersetzen (Santos et al. 2009a). Die metallurgischen Zusammenhänge und relevanten Einflussgrößen für die Festigkeit und andere Materialeigenschaften sind bereits experimentell untersucht und qualitativ bekannt (Wüller 2015). Eine Validierung der ermittelten Zusammenhänge mittels empirischer Modellinterpretationsverfahren wird jedoch nicht durchgeführt.

Die Vorhersage von Oberflächenfehlern an Gussteilen und ihre Kategorisierung in unvollständige Formfüllung, Kaltlauf und Einschlüsse erfolgt anhand von Bilderkennung und Klassifikation mit maschinellem Lernen. Dazu dient ein eigens entworfener Versuchsaufbau, der ein 3D-Profil der Gussteile erstellt und daraus eine Höhenmatrix berechnet. Diese wird mit von der Forschungsgruppe entwickelten Segmentierungsverfahren in Features umgewandelt. Die FFT wird zur Vorverarbeitung der Bilddaten genutzt und extrahiert relevante Teilbereiche, die von anderen Modellen klassifiziert werden (Pastor-López et al. 2019a).

Weitere Arbeiten befassen sich mit der Vorhersage von Schlackenbildung in Gussteilen anhand der Prozessparameter (Santos et al. 2013). Sie nutzen die gleichen Verfahren wie für die Vorhersage von Mikrolunkern. Später wird der Ansatz ausgeweitet, um im Rahmen von Model *Predictive Control* (MPC) die Vorhersage des stationären Endzustands durch einen hybriden Klassifikator zu verbessern und Prozesse auf diese Weise zu steuern (Nieves et al. 2014). Die aufgeführten Verfahren erzielen in einer experimentellen Umgebung teils sehr hohe Genauigkeiten. Informationen zur tatsächlichen Realisierung im Produktionsumfeld und zur nachhaltigen Ausschussreduktion durch eine langfristige Nutzung lassen sich jedoch nicht finden.

Das menschliche Auffassungsvermögen und die von ihm abhängige Zuverlässigkeit manuell erhobener Qualitätsmessungen sind Forschungsgegenstände im Rahmen der praktischen Umsetzbarkeit von maschinellem Lernen für Gießereiprozesse. Untersuchungen der Colorado State University befassen sich mit den Rahmenbedingungen für eine sinnvolle Anwendung überwachten Lernens im Druckguss. Da Methoden der Computer-Vision noch nicht flächendeckend eingesetzt werden, entfällt die visuelle Inspektion in größeren Teilen weiterhin auf Menschen. Selbst nach aufwändigen Schulungen sind diese lediglich in der Lage, 50 – 80 % der fehlerhaften Teile richtig zu identifizieren (Blondheim 2021). Darüber hinaus wird in Zusammenarbeit mit Mercury Castings das Auftreten größerer Lunker (engl. *macro porosity*), die für gewöhnlich erst bei Röntgenuntersuchungen oder mechanischer Nachbearbeitung freigelegt werden, thematisiert. Ihrem Vorkommen wird eine stochastische Charakteristik zugesprochen. Zonen mit Lunkern können grob vorhergesagt werden, Größe und Form jedoch nicht (Blondheim und Monroe 2022). Diese Problematik zeigt die Grenzen einer datengetriebenen Optimierung auf.

Forschungsprojekte der Universität Duisburg-Essen und der Hochschule Kempten befassen sich mit der datengetriebenen Analyse und Optimierung von Gießereiprozessen. In einem Verbundprojekte mit mittelständischen Gießereien werden zwecks digita-

ler Gussteilrückverfolgbarkeit an automatischen Formanlagen eine Markiereinheit entwickelt, die sich in Formkästen einbringen lässt (Sandt et al. 2020). Die so sichergestellte Verknüpfung von Prozessparametern und Gussteilqualität kann zur Formstoffsteuerung in Grünsandgießereien zur Erhöhung der Rohstoff- und Energieeffizienz eingesetzt werden (Bührig-Polaczek et al. 2021). Eine Methode zur Ermittlung von Key-Performance-Indikatoren mit einer Verifizierung anhand von Gießereidaten wurde von Kerst et al. (2018a) entwickelt. Sie wird anschließend genutzt, um die mechanischen Eigenschaften von duktilem Gusseisen mit Kugelgrafit auf Grundlage der chemischen Analyse vorherzusagen (Kerst et al. 2018b). Die Anwendbarkeit der Methoden wurde ebenfalls zur Prognose von Formstoffkennwerten und einer Veranschaulichung der Wirkzusammenhänge mit einer Sensitivitätsanalyse gezeigt (Kerst und Gottschling 2020). Datengetriebene Optimierung in Verbindung mit maschinellem Lernen werden zur Steigerung der Energie- und Rohstoffeffizienz im induktiven Schmelzprozess genutzt (Kaufmann und Tonnius 2021). Hierzu werden die ofenspezifischen Wechselwirkungen modelliert und eine kostengünstige Auswahl der Einsatzmaterialien getroffen (Kaufmann et al. 2022). Die Entwicklung einer cloudbasierten Handelsplattform für Schrotte der Recyclingindustrie als Beitrag für eine ressourceneffiziente Kreislaufwirtschaft wurde von Gottschling et al. (2020) entwickelt.

3.1.2 Simulationsgetriebene Prozessoptimierung

Mithilfe analytischer Simulationsmodelle lässt sich der Prozessverlauf beim Gießen und Erstarren unter gegebenen Annahmen detailliert nachbilden. Gegenstand der Modellierung sind unter anderem die Strömung der Schmelze in die Form, der Temperaturverlauf der Schmelze bei der Formfüllung und die Erstarrung des Gussteils. Weiterhin lassen sich andere Fertigungsschritte wie die Wärmebehandlung und die Kernfertigung in synthetischen Umgebungen simulieren. Die dafür notwendigen numerischen Berechnungen beruhen auf bekannten Zusammenhängen der Thermo- und Fluidodynamik.

Die klassische Gießprozesssimulation hat zur Aufgabe, Verbesserungen beim Entwurf von Gussteilen hervorzubringen, Risiken beim Gießen und Erstarren abzuschätzen und den Einfluss von Fertigungsparametern zu analysieren. Des Weiteren kann sie dabei helfen, den Rohstoff- und Energieeinsatz durch Ausschussreduktion zu verringern und bestehendes Prozesswissen zu erweitern. Konkrete Projekte beschäftigen sich mit der Optimierung der mechanischen Eigenschaften anhand verschiedener Prozessparameter und der Schaffung von Prozesstransparenz (Wagner und Sturm 2019). Die Gießprozesssimulation kann für eine virtuelle Versuchsplanung (engl. *Design of Experiments*, DoE) genutzt werden. So lassen sich auch für schwer beherrschbare Teilprozesse adäquate Lösungen finden. Eine große Herausforderung hierbei ist die hohe Variantenvielfalt der Prozesse bzw. die Zahl der abzubildenden Parameterkombinationen. Um dem entgegenzuwirken obliegt es den Experten, Zielvorgaben festzulegen und Ausgangssituationen so zu definieren, dass der Berechnungsaufwand tragbar bleibt (Velazquez Blandino et al. 2019).

Hahn und Sturm (2012) klassifizieren Optimierungsstrategien mit Simulationsmodellen. Bei der *eindimensionalen Optimierung* werden schrittweise Verbesserungen durch aufeinanderfolgende Versuche unter Fertigungsbedingungen erreicht. Unklarheiten verbleiben hinsichtlich der Findung globaler Optima und der Robustheit des Prozessfensters. Die Prozedur der Variantenfestlegung, Simulation und anschließender Auswertung wird wiederholt, bis die gewünschte Gussteilqualität erreicht ist. Die Anzahl der Versuche ist

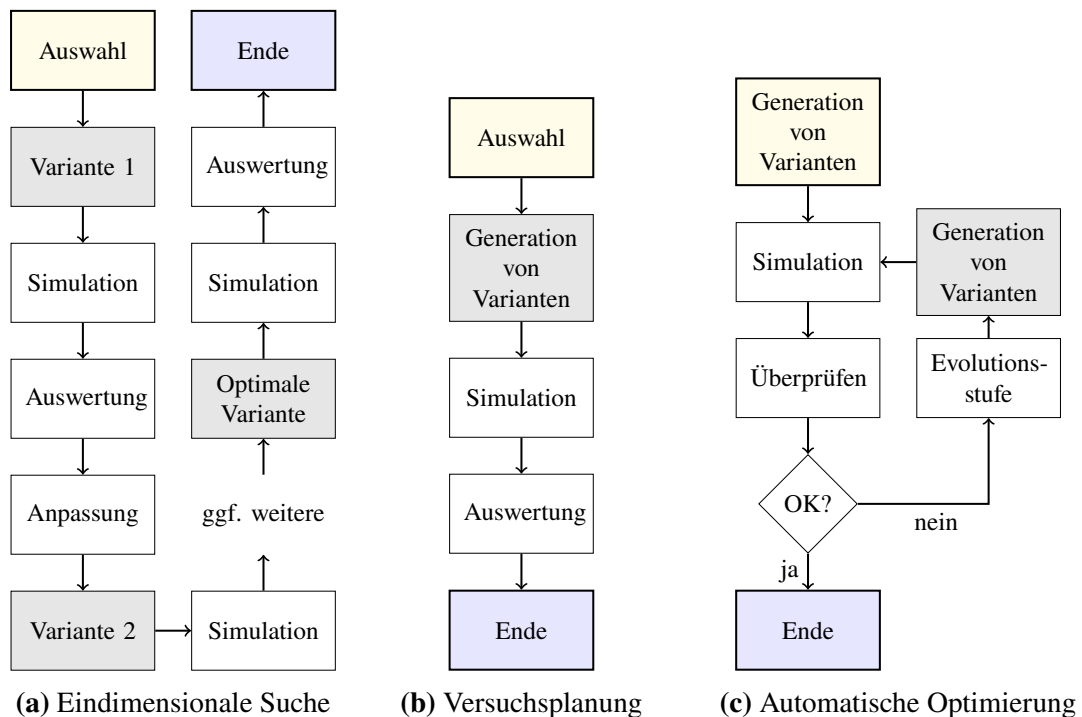


Abbildung 3.1: Ansätze der simulationsgestützten Gießereiprozessoptimierung nach Hahn und Sturm (2012). Nur bei der automatischen Optimierung werden Varianten gebildet und variiert, bis ein Zielkriterium erfüllt ist.

begrenzt und eine Konvergenz zum Optimum nicht gewährleistet. Bei der *Versuchsplanung* wird der Ansatz verfolgt, reale Experimente durch Simulationen zu ersetzen. Ziel ist es, die Auswirkung einzelner Prozessparameter auf den Gießprozess zu bewerten, um das Verhalten später möglichst genau vorhersagen zu können. Es gibt eine Mindestzahl durchzuführender virtueller Abgüsse, um Wirkmuster darstellen zu können. Der Nutzer gibt einen festen Bereich vor, in dem die Parameter variiert werden, und wertet die Ergebnisse nach Fertigstellung der Simulation aus. Bei der *automatischen rechnergestützten Optimierung* lassen sich nicht nur Freiheitsgrade der Prozessparameter, sondern auch Optimierungsziele festlegen. Diese können zueinander im Widerspruch stehen. Es werden eigenständig Generationen von Varianten definiert, Simulationen mit ihnen durchgeführt und anhand der Ziele ausgewertet. Die Erzeugung neuer Varianten erfolgt durch die Nutzung genetischer Algorithmen. Das Verfahren endet, sobald keine Verbesserungen erzielt werden können.

Die Grenzen dieser Art der Prozessoptimierung liegen einerseits in der Beschränkung der Zahl durchführbarer Versuche. Sturm und Pavlak (2014) führen beispielsweise eine statistische Versuchsplanung mit 16 unterschiedlichen Design durch, um das Auftreten von Oxideinschlüssen zu untersuchen. Obwohl keine Angaben über die genaue Rechenzeit gemacht werden, können derartige Simulationen mehrere Tage dauern. Sie liefern dafür detaillierte Einblicke in der Verfahren, eignen sich jedoch nicht für eine Online-Optimierung. Darüber hinaus können einige selten praktizierte Gießverfahren wie der Schleuderguss analytisch nicht adäquat abgebildet werden. Dies liegt unter anderem an der Rotationsbewegung der Kokille und Verfahrensspezifika, die sich je nach Prozess stark unterscheiden und für die eine Neuentwicklung der Simulation nicht wirtschaftlich wäre.

3.2 Anomalieerkennung für industrielle Prozessdaten

Eine umfangreiche Übersicht der aktuell genutzten Datenvorverarbeitungsmethoden stellen Xu et al. (2015) zusammen. Die Grundpfeiler der Datenvorverarbeitung bestehen aus Missing Data, Ausreißererkennung, Rauschreduktion und Zeitversatz. Qin (2009) erörtert unterschiedliche Ansätze der datengetriebenen Fehlerdiagnose in der Prozessindustrie. Die Anwendbarkeit der Verfahren unterliegt der Annahme, dass sich Anomalien in Prozessen in einer deutlichen Veränderung einzelner Parameter oder ihrer Kombinationen abzeichnen. Aus den Prozessdaten wird mit der Singulärwertzerlegung von \mathbf{X} die empirische Kovarianzmatrix

$$\text{cov}(\mathbf{X}) \approx \frac{1}{m-1} \mathbf{X}^T \mathbf{X} = \frac{1}{m-1} \mathbf{V} \mathbf{\Sigma}^2 \mathbf{V}^T \quad (3.1)$$

von \mathbf{X} gebildet. Mit $\mathbf{S}^2 = \mathbf{\Lambda}$ und $\frac{1}{\sqrt{m-1}} \mathbf{V} = \mathbf{P}$ ergibt sich

$$\text{cov}(\mathbf{X}) \approx \mathbf{P} \mathbf{\Lambda} \mathbf{P}^T. \quad (3.2)$$

Werden durch einen Threshold die ersten r Spalten von \mathbf{P} zu $\tilde{\mathbf{P}}$ und die verbleibenden zu \mathbf{P}_R zusammengefasst, lässt sich $\text{cov}(\mathbf{X})$ zu

$$\text{cov}(\mathbf{X}) \approx \begin{bmatrix} \tilde{\mathbf{P}} & \mathbf{P}_R \end{bmatrix} \mathbf{\Lambda} \begin{bmatrix} \tilde{\mathbf{P}} & \mathbf{P}_R \end{bmatrix}^T \quad (3.3)$$

umschreiben. $\tilde{\mathbf{P}}$ wird genutzt, um den *Squared Prediction Error* bzw. die *Q-Statistik*

$$Q(\mathbf{x}) = \|(\mathbf{I} - \tilde{\mathbf{P}} \tilde{\mathbf{P}}^T) \mathbf{x}\|_2^2 \quad (3.4)$$

zu bilden. Sie bezeichnet die Projektion einer Stichprobe \mathbf{x} des Prozesses auf den Residuenunterraum und eignet sich zur Prozessüberwachung und Anomalieerkennung (Chiang et al. 2000). Dazu wird ein Threshold unter Berücksichtigung eines Konfidenzintervalls gewählt. Als Fallbeispiele dienen Anwendungen aus der chemischen Industrie, wo aufgrund hoher Qualitätsanforderungen eine durchgängige Datenerfassung gewährleistet ist.

3.3 Dimensionsreduktion und Messraumverdichtung

Scherl et al. (2020) nutzen die robuste Hauptkomponentenanalyse zur Extraktion kohärenter Strukturen in Zylinderströmungen. Dadurch wird die Datenqualität der turbulenten Strömungsfelder verbessert. Datengrundlage sind numerische Simulationen und künstliches Rauschen. Messfehler und fehlende Messungen handhabt die RPCA effizient. Am Ende wird die Performance anhand der *Dynamic Mode Decomposition* geprüft, da diese empfindlich gegenüber Ausreißern ist. Champion et al. (2019) nutzen Deep-Autoencoder zur Generierung von Modellen reduzierter Ordnung aus hochdimensionalen Datenmengen, in denen eine niedrigrangige Dynamik vermutet wird. Die Validierung erfolgt anhand des komplexen Lorenzsystems und einer datengetriebenen nichtlinearen Systemidentifikation (SINDy).

Methoden der Messraumverdichtung erlauben es, unterabgetastete Signale mit ℓ_1 -Minimierung nahezu verlustfrei zu rekonstruieren und damit unter bestimmten Umständen das Nyquist-Shannon-Abtasttheorem zu umgehen (Brunton et al. 2015). Eine noch

effektivere Umsetzung ist die Extraktion relevanter Merkmale und die Nutzung des daraus gewonnenen Wissens über das System oder den Prozess. Manohar et al. (2018) demonstrieren die optimale Platzierung von Sensorik anhand von Gesichtserkennung und Strömungsfeldern aus der Fluidodynamik.

Witt (2018) entwickelt ein Verfahren zum Online-Zustandstracking anhand datengetriebener Prozessmodelle. Die Datengrundlage bildet eine nichtlineare Finite-Elemente-Simulation, aus denen mit maschinellem Lernen eine Darstellung reduzierter Ordnung extrahiert werden soll. Eine Reduktion der Zustandsvariablen erfolgt mit neuronalen Netzen und die anschließende Modellierung mit symbolischer Regression. Die Online-Abbildung nicht messbarer Verfahrenskennwerte wird an Wärmeleit- und Tiefziehprozessen validiert.

3.4 Industrielle Data-Mining-Standards

Auf Basis erfolgreicher Data-Mining-Projekte in der Industrie wurde aus einem Konsortium von Unternehmen aus dem Bereich Fertigung und Software sowie der Wissenschaft der *CRoss Industry Standard Process for Data Mining* (CRISP-DM) entwickelt. Er ist heutzutage immer noch einer der am weitesten verbreiteten Prozesse für Data Mining und Projekte mit maschinellem Lernen in der Industrie (Chapman et al. 2000). Es liegen sechs unterschiedliche Phasen vor, die iterativ und interaktiv bearbeitet werden.

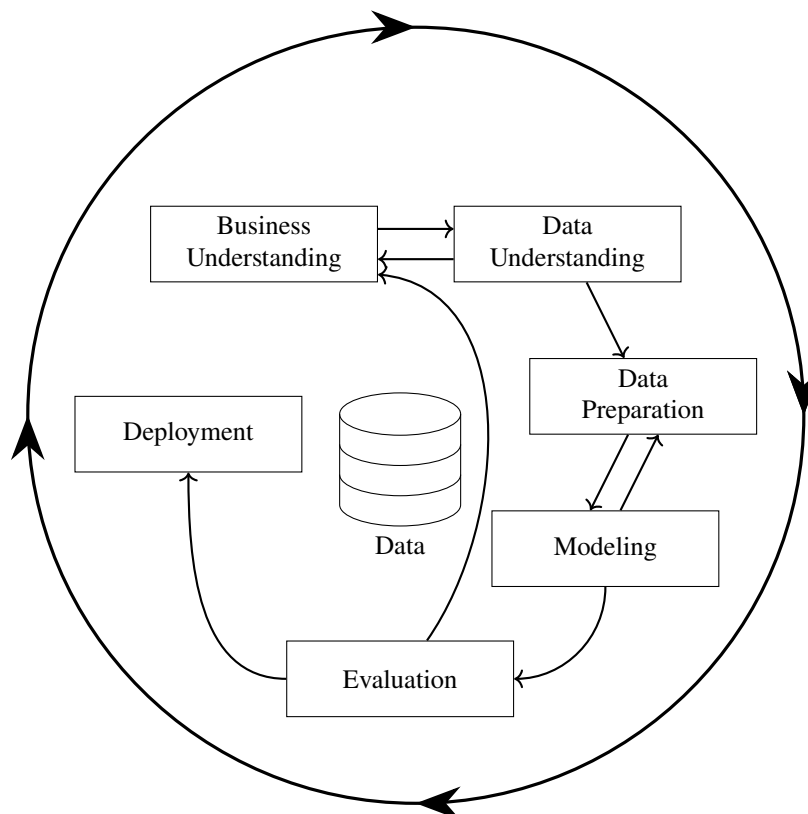


Abbildung 3.2: Schematischer Ablauf des CRISP-DM in Anlehnung an Chapman et al. (2000), eigene Darstellung.

In der ersten Phase werden Projektziele und Anforderungen festgelegt und Problem-

stellungen für das Data Mining formuliert. Daraus wird eine Aufgabenstellung und ein Projektplan erarbeitet. Das Projektmanagement spielt eine entscheidende Rolle. In der zweiten Phase werden die vorhandenen Daten zusammengetragen und durch eine explorative Analyse ein erstes Verständnis der Datenlage konsolidiert. Es lassen sich durch einfache statistische Untersuchungen bereits Erkenntnisse über kausale Zusammenhänge und einfache Korrelationen gewinnen, die sich zu Arbeitshypothesen zusammenfassen lassen. In der dritten Phase werden die Daten in eine konsistente Form überführt, die für das Training der Modelle benötigt wird. Hierzu werden eine Säuberung und eine Korrektur vorgenommen. Durch eine geeignete Datentransformation können verschiedene Datenquellen zusammengefügt werden. Bei der Modellierung (Modeling) wird für die aufbereiteten Daten ein Modell erstellt, welches beschreibt, wie die Daten einzuordnen oder zu behandeln sind. Das Modell kann aus verschiedenen Formen von Regeln oder Beschreibungen von Clustern bestehen. Abhängig von der Aufgabe wird ein oder mehrere geeignete Verfahren ausgewählt. Sie können später noch durch modellspezifische Hyperparameter optimal eingestellt und verfeinert werden. Anschließend erfolgt die Evaluierung und Validierung der trainierten Modelle im Hinblick auf die in Phase 1 festgelegten Ziele und Erfolgskriterien. Der wirtschaftliche Nutzen durch die Analyseergebnisse ist zu untersuchen. Weiterhin müssen die Modelle auf ihre Plausibilität überprüft werden. Daraus resultiert eine Entscheidung zur weiteren Anwendung. Die letzte Phase beinhaltet die Anwendung der Modelle auf die unternehmerischen Entscheidungsprozesse und Abläufe.

Fayyad et al. (1996) haben unter dem Namen Knowledge Discovery in Databases (KDD) eine der ersten systematischen Vorgehensweisen entwickelt, um Wissen aus Daten zu extrahieren. Der Prozess beinhaltet die Schritte Auswahl, Aufbereitung, Transformation, Analyse und Interpretation. Es existieren Gemeinsamkeiten und Unterschiede zwischen dem KDD-Workflow für Datenanalysen und dem CRISP-DM als Industriestandard. Die Punkte 1, 2 und 6 des CRISP-DM sind sehr projekt- und anwendungsspezifisch und werden im KDD nicht explizit angegeben. Dahingegen enthält Phase 3 des CRISP-Modells die ersten drei Schritte des Fayyad-Modells. Beide Vorgehensweisen erfolgen iterativ und interaktiv. CRISP eignet sich daher für die Umsetzung von Data-Mining-Projekten und infrastrukturbezogene Aufgaben in Industrieunternehmen. KDD setzt den Schwerpunkt auf den Ablauf der Datenanalyse und die Zieldatenformate.

3.5 Verwandte Forschungsarbeiten

Deike (2020) entwickelt ein Konzept für die Nutzung von maschinellem Lernen zur kontinuierlichen Qualitätsverbesserung in Fertigungsprozessen. Er unterscheidet drei verschiedene Anwendungsszenarien. Eine Verifizierung der Verfahren erfolgt im Prozess der Langstahlproduktion. Der Schwerpunkt der Entwicklung liegt in der Optimierungsschleife, die modulare Nutzbarkeit der Ansätze wird als Nebenanforderung aufgeführt.

Ein Konzept zur mehrkriteriellen Prozessoptimierung legen Zumsande et al. (2019) vor. Die zu optimierenden Zielgrößen des vorliegenden Projektes umfassen den Energieverbrauch, die Anlagenleistung und die mechanische Belastung eines Roboters. Den dahinterstehenden physikalischen Prozess beschreibt eine analytische Modellbildung. Die Aufgabenstellung wird zu einem einkriteriellen Optimierungsproblem überführt, indem eine gewichtete Summe gebildet wird.

Gröger (2015) entwickelte eine Plattform zur datengetriebenen Optimierung von Fer-

tigungsprozessen mit dem Fokus auf Business-Intelligence-Anwendungen. Die Herangehensweise untergliedert sich in die Datenbereitstellung, die Prozessanalyse und die Prozessoptimierung. Die Schwerpunkte der Arbeit liegen in der Ableitung von Modellen aus Prozessabläufen, der Prüfung ihrer Konformität und schließlich der kontinuierlichen Verbesserung der Modelle mit neuen Daten. Der Benutzerzugriff wird durch ein mobiles Prozess-Dashboard realisiert. Die Modularität der Ansätze und die Einbindung fortgeschrittener Analysemethoden werden nicht vordergründig diskutiert. Die Prozessoptimierung und anschließende Generierung von Handlungsempfehlungen erfolgt mit Hilfe von Entscheidungsbäumen und einer Pfadanalyse. Sie beschränkt sich auf Klassifikationsprobleme und erlaubt keine modellagnostischen Ansätze. Eine systematische Optimierung auf Grundlage der Prognosefunktion und gradientenfreien Methoden findet nicht statt. Ein ähnliches dreischichtiges Konzept mit dem Namen *Deep Business Optimization* wurde von Niedermann (2015) zur kontinuierlichen Verbesserung von Geschäftsprozessen implementiert. Der Schwerpunkt der Arbeit liegt in der Kombination der Fachexpertisen von Prozessexperten und Analysten sowie einer Ausführungsstrategie datengetriebener Optimierungsverfahren.

Zusammenfassung

In diesem Kapitel wurde zunächst der Stand der Technik hinsichtlich der Prozessoptimierung in Gießereien zusammengefasst. Die Forschungsgruppen um AZTERLAN nutzen maschinelles Lernen zur Vorhersage kritischer Qualitätskennwerte und zur automatischen Erkennung von Oberflächenfehlern. Während sie hohe Prognosegenauigkeiten für die einmalig erhobenen Daten nachweisen können, führen sie kein Konzept auf, die erarbeiteten Wirkzusammenhänge für eine nachhaltige Prozessverbesserung einzusetzen. Die simulationsgestützte Gießprozessoptimierung unter Nutzung von MAGMASOFT[®] verwendet eine analytische Systembeschreibung und beschränkt sich auf genetische Algorithmen zur Bestimmung einer optimalen Prozessvariante. Sie ist unabdingbar zur virtuellen Versuchs- und Bauteilplanung, kann jedoch nicht echtzeitgerecht die gleiche Variantenvielfalt und die Prozess- sowie Umgebungsspezifika abbilden wie datengetriebene Verfahren.

Der Stand der Forschung umfasst die Nutzung unüberwachter Lernverfahren für Ingenieuranwendungen und zur industriellen Prozessüberwachung, die in Gießereien noch nicht angewandt werden. In den aufgeführten Beispielen wird eine analytische Beschreibbarkeit des Prozesses bzw. des Systems vorausgesetzt. In keiner der verwandten Forschungsarbeiten konnte ein vereinheitlichtes Konzept zur modularen Integration unüberwachter Lernverfahren wie RPCA oder Autoencodern in eine Pipeline zur gradientenfreien Optimierung synthetisierter Prozessvarianten gefunden werden.

Kapitel 4

Entwicklung der Optimierungskette

Dieses Kapitel stellt das Konzept der datengetriebenen Optimierungskette vor, beginnend mit einer Bestimmung der zu erfüllenden Anforderungen, die es später in der Fallstudie zu verifizieren gilt (4.1). Als Nächstes wird das Datenvorverarbeitungsmodul beschrieben (4.2). Es beinhaltet eine robuste Störgrößenkompensation und Ansätze zur Findung einer optimalen Rangreduktion. Das Modellierungsmodul beschreibt die Nutzung der Verfahren des maschinellen Lernens zur Abbildung eines Prozesses und ihre Interpretation (4.3). Schließlich wird beim Prozessoptimierungsmodul auf die Nutzung der Modelle zur Generierung von Handlungsempfehlungen eingegangen (4.4). Es umfasst die gradientenfreie Optimierung und den Entwurf einer Kostenfunktion.

4.1 Rahmenbedingungen und Lösungsweg

Die in diesem Kapitel vorgestellte Methodik zur Modellierung und Optimierung des Ein- und Ausgangsverhaltens wird aus Gründen der Übersichtlichkeit und in Bezug auf das später gezeigte Anwendungsbeispiel nur für einen Teilprozess entwickelt. Das Konzept lässt sich aber unter Berücksichtigung der Echtzeitanforderungen und der verfügbaren Rechenkapazität auf beliebig viele Teil- und Unterprozesse ausweiten. Für die Anwendbarkeit der Analyseverfahren muss angenommen werden, dass eine eindeutige Zuordnung zwischen Prozessparametern und vorherzusagenden Qualitätskennwerten vorliegt. Ist dies nicht der Fall, können die in Ge et al. (2017) genannten Methoden für disparate Zeitreihen herangezogen werden. In manchen Fällen ist eine analytische Betrachtung des Prozesses sinnvoll. Beispielhafte Maßnahmen werden in Abschnitt 4.3 vorgestellt. Weiterhin können schleichende Prozessveränderungen und zeitvariante Zusammenhänge der Prozessgrößen nicht abgebildet werden. So kann sich eine Parameterkombination nur auf die Qualität des nächsten produzierten Teils auswirken. Vergangene Werte der Prozessparameter und deterministische Ereignisse, die sich auf die Bauteilqualität auswirken, müssen anhand gesonderter Features beschrieben werden.

4.1.1 Anforderungen

Es folgt eine Übersicht der Anforderungen an die zu entwickelnde Optimierungskette.

Gesamtsystem

Für das Gesamtsystem werden folgende Anforderungen definiert:

- A₁: Modulare Nutzbarkeit der Ansätze:* Die Schnittstellen innerhalb der Optimierungskette werden so entworfen, dass einzelne Teile auf andere Fertigungsprozesse übertragen werden können.
- A₂: Modus für Offline- und Onlinebetrieb:* Die Verfahren innerhalb der Optimierungskette müssen so ausgelegt sein, dass sie sowohl eine einmalige als auch eine kontinuierliche Prozessoptimierung ermöglichen.

Eine modulare Nutzbarkeit der Bestandteile (A_1) soll wie in Gröger (2018) und Deike (2020, S. 9–10) gewährleistet werden. Eine Wiederverwendbarkeit soll einerseits für verschiedene Gießereiprozesse gelten, aber auch dann Bestand haben, wenn einzelne Module nicht anwendbar sind oder eine Anwendung überflüssig ist. Letzteres ist beispielsweise dann der Fall, wenn die Datenvorverarbeitung an einer anderen Stelle vorgenommen wird oder datengetriebene Prozessmodelle aus einer externen Quelle entstammen, sofern tabellarische Datenstrukturen und keine proprietären Formate vorliegen. Unterschiedliche Modi der Optimierungskette für den Offline- und Onlinebetrieb (A_2) stellen sicher, dass verschiedene Verfahren genutzt werden, je nachdem, ob eine Gesamtdatenbasis zur Analyse vorliegt oder einzelne Instanzen des Prozesses verarbeitet werden sollen. So können einige Vorverarbeitungsmethoden lediglich ganze Datenmatrizen, jedoch keine alleinstehenden Parameterdatensätze handhaben.

Datenvorverarbeitung

Für die Datenvorverarbeitung werden folgende Anforderungen definiert:

- A₃: Featureextraktion:* Transformierte Metadaten müssen dem Prozessexperten immer in ihrem ursprünglichen Format zugänglich sein.
- A₄: Störgrößenkompensation:* Kompensation von Störungen, die nicht normalverteilt sind oder mit dem Prozess korrelieren.

Datentransformationen und Featureextraktionen (A_3) im Rahmen der Datenvorverarbeitung dienen dem Informationsgewinn, der Störgrößenkompensation und dem Modell als Eingang. Für jegliche Interpretation, Plausibilitätsprüfung und Visualisierung der Wirkzusammenhänge sind die Metadaten jedoch in ihre ursprüngliche Form zurückzuführen, um die physikalische Interpretierbarkeit zu gewährleisten. Es ist darüber hinaus eine Störgrößenkompensation (A_4) zu finden, die möglichst robust auf äußere Einflüsse reagiert und auch dann eine bereinigte Datenbasis bereitstellt, wenn Störungen prozessbedingt auftreten, mit Prozessgrößen korrelieren oder deterministischer Natur und nicht normalverteilt sind.

Modellierung

Für die Modellierung werden folgende Anforderungen definiert:

- A₅: *Modellagnostische Verwertungsansätze*: Aufgrund der unbekanntenen Eignung spezifischer Verfahren und Umsetzungen von maschinellem Lernen für gegebene Prozesse und Problemstellungen soll ein Ansatz verfolgt werden, der es zulässt, Modelle flexibel und anforderungsgerecht zu substituieren.
- A₆: *Vorgehen nach etabliertem Industriestandard*: Um die entwickelte Methode in Digitalisierungsprojekten zugänglich zu machen, soll sie sich in ihrer Ausführung an gängigen Industriestandards orientieren.
- A₇: *Modellierung und Modellinterpretation für Mehrgrößenprognosen*: Die Modellierung und Visualisierung der Wirkzusammenhänge zwecks Plausibilitätsprüfung wird auf Mehrgrößenprognosen ausgeweitet.

Aufgrund der unbekanntenen Eignung spezifischer Verfahren und Umsetzungen von maschinellem Lernen für gegebene Prozesse und Problemstellungen soll ein modellagnostischer Ansatz (A₅) verfolgt werden, der es zulässt, maschinelle Lernverfahren flexibel und anforderungsgerecht zu substituieren. Folglich sind die Schnittstellen der Glieder der Optimierungskette so zu implementieren, dass sowohl die Modellinterpretation zur Plausibilitätsprüfung als auch die gradientenfreie Optimierung der Prozessvarianten Zugriff auf die Modellergebnisse haben. Im Interesse einer erleichterten Verwertung der Ergebnisse und eines branchenübergreifenden Einsatzes kommt die Entwicklung der Methoden etablierten Industriestandards wie dem CRISP-DM oder der KDD (A₆) nach. Jeder Schritt der Optimierungskette lässt sich dabei eindeutig einer Projektphase zuordnen. Weiterhin werden die Methoden der Modellierung und der Modellinterpretation auf vektorwertige Zielgrößen ausgeweitet (A₇). So sollen Wirkzusammenhänge mehrdimensional sichtbar gemacht werden.

Prozessoptimierung

Für die Prozessoptimierung werden folgende Anforderungen definiert:

- A₈: *Adaptive Arbeitsweise*: Das datengetriebene Assistenzsystem muss lokale Fertigungsbedingungen bei der Generierung von Handlungsempfehlungen berücksichtigen können. Die Komplexität der der Anweisung muss steuerbar sein.
- A₉: *Berücksichtigung des zeitlichen Prozessablaufs*: Die Generierung der Handlungsempfehlung erfolgt dynamisch und datengetrieben zu einem Zeitpunkt, in dem sich noch Änderungen am Prozess vornehmen lassen. Es wird eine Maximallaufzeit für die Optimierung vorgegeben.
- A₁₀: *Nachvollziehbarkeit und Individualisierbarkeit*: Die Darstellung der Handlungsempfehlung muss an die Belange des Nutzers angepasst werden. Auswirkungen auf den Prozess müssen verständlich visualisiert und in einem absehbaren Zeitrahmen verfügbar gemacht werden.
- A₁₁: *Fehlertoleranz*: Physikalisch und verfahrensbedingt nicht realisierbare Prozessvarianten sowie unplausible Prozessergebnisse dürfen vom Assistenzsystem nicht umgesetzt werden. Dies gilt sowohl für synthetisierte Parameterkombinationen als auch für Nutzereingaben.

Nach Bauernhansl et al. (2015) liegen die praktischen Herausforderungen in der effektiven Datenverarbeitung bei einer großen Anzahl von Sensoren. Der richtige Zeitpunkt und Ort für die Manipulation der Daten muss gefunden werden, damit der Prozess auch auf diskontinuierliche Einflüsse reagieren kann. Anforderungen an die Dialoggestaltung im Rahmen eines Assistenzsystems zur Generierung von Handlungsempfehlungen werden in der DIN EN ISO 9241-110 (2020) festgelegt. Die Aufgabenangemessenheit eines solchen Systems soll sicherstellen, dass der Schwerpunkt bei der Interaktion in der Arbeitsaufgabe liegt und dem Nutzer nur relevante Informationen präsentiert werden. Eine Darstellung der Prozessdaten ist so zu implementieren, dass kein zusätzliches Material zum Verständnis notwendig ist.

Zu diesem Zweck ist die Arbeitsweise der datengetriebenen Prozessoptimierung adaptiv zu gestalten (A₈). So muss auch bei vorherrschenden Fertigungsbedingungen, die kein optimales Prozessergebnis mehr zulassen, eine anforderungsgerechte Maßnahme entwickelt werden können, die die Auswirkungen auf die Produktion begrenzt. Weiterhin muss die Komplexität der generierten Handlungsempfehlung dynamisch an die vorliegende Prozesssituation anzupassen und vom Nutzer steuerbar und bewertbar sein. Um die Kausalität der Prognose zu gewährleisten, müssen Eingriffe zu einem Zeitpunkt vorgenommen werden, an dem einerseits alle notwendigen Informationen für die Vorhersage vorliegen und andererseits noch wirkungsvoll auf das Prozessergebnis Einfluss genommen werden kann (A₉). Hieraus ergeben sich Anforderungen an die Rechenlaufzeit und allgemeine Performance der datengetriebenen Optimierung. Um die Nachvollziehbarkeit und die Individualisierbarkeit der Entscheidungsfindung zu gewährleisten, müssen dem Nutzer wichtige Informationen zur praktischen Umsetzung intuitiv bereitgestellt werden (A₁₀). Hierzu gehört neben der Visualisierung der relevantesten Wirkzusammenhänge die Möglichkeit, dem System eigene Prozesseingriffe zu übermitteln und ihre Auswirkungen mit den Maßnahmen der automatischen Optimierung abzugleichen. Das Assistenzsystem und die dahinterliegende Optimierungsmethodik sollen sich durch eine hohe Fehlertoleranz und die Fähigkeit, diese zu erkennen und ihnen entgegenzuwirken, auszeichnen (A₁₁). Die potenziellen Fehlerquellen liegen in der Nutzereingabe, der Generierung verfahrenstechnisch nicht umsetzbarer Prozessvarianten und unerwarteten Prozessergebnissen.

4.1.2 Entwicklungsansatz

Nachfolgend wird ein Entwicklungsansatz vorgestellt, der die Zielsetzung, die theoretischen Grundlagen und die Anforderungen dieser Arbeit zusammenführt. Der Aufbau der Optimierungskette erfolgt zunächst für den Offlineanwendungsfall und ist schematisch in Abbildung 4.1 gezeigt. Er erweitert die in Abbildung 1.1 skizzierten Aspekte der datengetriebenen Optimierung und markiert die Handlungsfelder sowie die Schnittstellen der zu erarbeitenden Module. Die Datenaufnahme und –verknüpfung werden dem Datenaufbereitungsmodul zugeordnet, das Management und die Bereitstellung der Prozessdaten werden in Abschnitt 4.1.3 gesondert erläutert. Die Verknüpfung datengetriebener und analytischer Modelle einschließlich der empirischen Modellinterpretation ist Aufgabe des Modellierungsmoduls. Die datengetriebene Optimierung unter Berücksichtigung einer Kostenfunktion und die darauffolgende Generierung einer Handlungsempfehlung fallen unter das Prozessoptimierungsmodul.

Der Fertigungsprozess als Ausgangspunkt kann die Prozessparameter, die später als

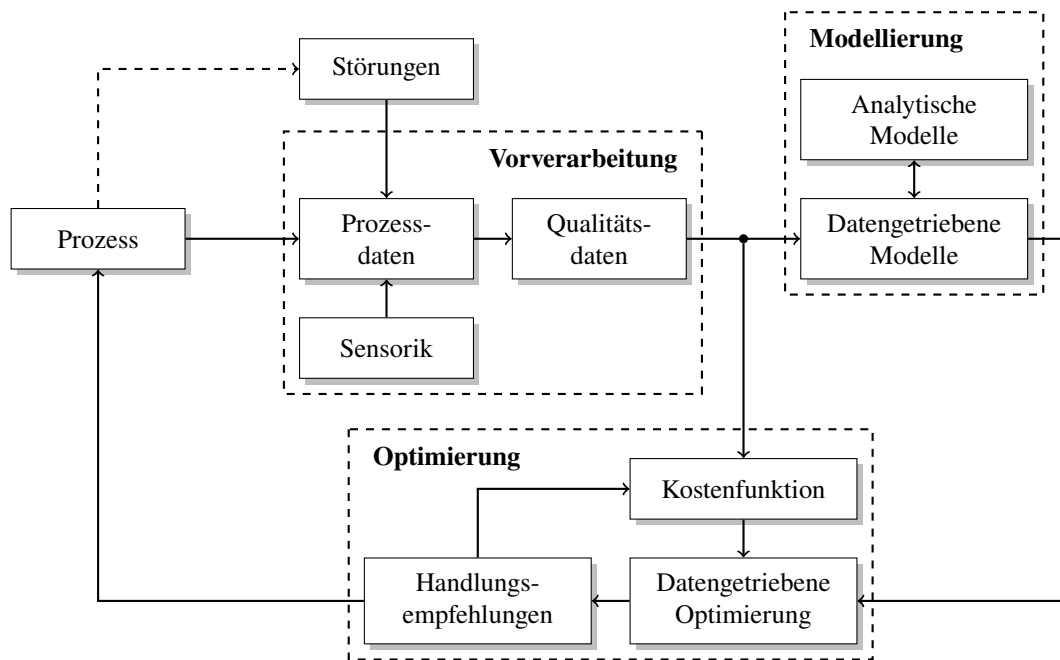


Abbildung 4.1: Entwicklungsansatz für den Offlinefall. Das Datenaufbereitungsmodul und das Datenmanagement liefern eine ganzheitliche Datenbasis. Analytische und datengetriebene Modelle lassen sich bei Bedarf miteinander kombinieren. Die Schleife aus Kostenfunktion, datengetriebener Optimierung und generierter Handlungsempfehlung bildet das Prozessoptimierungsmodul.

funktional unabhängige Daten dienen sollen, direkt aus den Aufzeichnungen der Anlage oder der Sensorik über geeignete Schnittstellen zur Verfügung stellen. Hinzu kommen händisch angefertigte Dokumente, Informationen über Einsatzmaterial und betriebliche Aufzeichnungen wie Schichtwechsel oder Stillstandzeiten. Für die modulare Modellierung des Produktionssystems müssen die Abläufe und die Datenzusammensetzung jedes einzelnen Fertigungsschrittes eindeutig beschrieben werden. Abbildung 4.2 zeigt die Einteilung der prozessbeschreibenden Einflüsse in Eingangsgrößen, Störungen und Steuerungsgrößen. Eingangsgrößen werden als fest gegeben angenommen und umfassen unter anderem aus vorherigen Fertigungsschritten bereitgestellte Einsatzstoffe, Materialeigenschaften und messbare Umgebungsbedingungen. Steuerungsgrößen können sowohl gemessen als auch aktiv beeinflusst werden. Externe Störungen, etwa bedingt durch die Umgebung, beeinflussen direkt die Prozessdatenaufzeichnung oder bewirken, dass die Sensoren falsche Informationen liefern. Innere Störungen können an die Dynamik des Prozesses gebunden sein. Sie müssen bei der Datenvorverarbeitung berücksichtigt werden, können aber bei der Modellierung nicht explizit abgebildet werden. Modelle müssen sich ihnen gegenüber jedoch robust verhalten. Es wird angenommen, dass sie deterministischer Natur und nicht normalverteilt sind und mit den Prozessgrößen korrelieren können.

Die Daten der QM-Klassifizierung sind nach der Erhebung durch ERP-Systeme, Kundenrückmeldungen, Sensorik oder menschliche Bewertung mit den Prozessparametern zu verknüpfen. Bei der Zuordnung können Zeitstempel oder Chargennummern hilfreich sein, wenn durch den Prozess keine eindeutige Rückverfolgung gewährleistet ist. Die sich

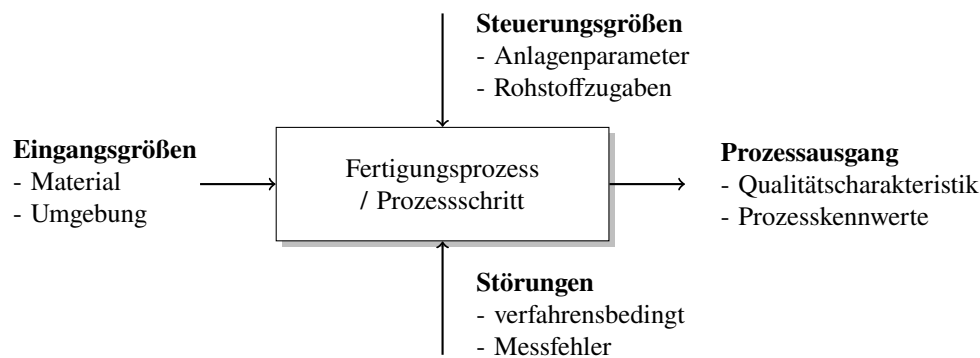


Abbildung 4.2: Verfügbare Daten in einem Prozessschritt oder für einen ganzen Prozess in Anlehnung an Herrmann et al. (2013, S. 253). Es wird zwischen vorgegebenen Eingangsgrößen, externen Störungen und steuerbaren Einflussgrößen unterschieden.

hieraus ergebende ganzheitliche Datenbasis kann entweder auf Einzelteilbetrachtung beruhen, Bauteile in Chargen zusammenfassen oder Zeiträume gemittelt darstellen, wenn die Zuordnung nicht im Detail realisierbar ist. Der Ausgang des Prozesses bzw. des Modells ist die Qualitätscharakteristik eines Produktes oder des Gesamtprozesses (Montgomery 2013).

Liegt eine ganzheitliche und hinreichend große Datenbasis vor, lassen sich mit ihr datengetriebene Prozessmodelle mit Hilfe von maschinellem Lernen entwickeln. Die Modelle sind schließlich in der Lage eine unmittelbare Beziehung zwischen den Fertigungsparametern und der Produktqualität herzustellen. Analytische Modelle können im Vorhinein und während der Entwicklungsphase das Aufgabenverständnis und den Lernprozess unterstützen. Typischerweise dienen sie

- als Referenzprozess für das Antrainieren von Modellen des maschinellen Lernens, wenn nicht genügend Prozessdaten vorhanden sind,
- als Anschauungsobjekt bei der Betrachtung des Prozessaufbaus und der Datenquellen, wenn die Zusammensetzung und die Wechselwirkungen der Prozessparameter unklar sind,
- sowie zur beispielhaften Verifizierung der Wirkzusammenhänge.

Eine Plausibilitätsprüfung kann auch mit Methoden der empirischen Modellinterpretation erfolgen. Am Ende des Lernprozesses steht eine Prognosefunktion, die ein Qualitätskriterien oder mehrere mit hinreichender Genauigkeit vorhersagen kann. Sie ist sowohl von Analysten als auch Prozessexperten hinsichtlich der Erfüllung des Verwendungszwecks abzunehmen.

Die Grundlage einer jeden Optimierung ist die Gestaltung einer anforderungsgerechten Ziel- bzw. Kostenfunktion. Aus ihr ergibt sich ein numerisches Gütekriterium, das die Erfüllung der Optimierungsziele, die Einhaltung der Rahmenbedingungen und andere Forderungen an die Prozessoptimierung quantifiziert. Jedes Zwischenergebnis, das bei einer Optimierung entsteht, kann als synthetisierte Prozessvariante verstanden werden, die

es anhand der Kostenfunktion zu evaluieren und gegebenenfalls zur Verbesserung zurückzuführen gilt. Eine entscheidende Eigenschaft der Optimierungskette ist die Fähigkeit, die generierten Prozessvarianten sowohl hinsichtlich des Modellausgangs bzw. der Prognose als auch der zugrundeliegenden Parameterkombination zu beurteilen. Für jede prozessbedingte Anforderung lässt sich mindestens ein Term in der Kostenfunktion formulieren. Weiterhin ist die Art und die Komplexität des empfohlenen Eingriffs zu bewerten. Dies muss innerhalb der Optimierungsschleife mathematisch erfolgen, nach Bereitstellung an den Maschinenbediener müssen sie aber abschließend mit vorhandenem Prozesswissen abgleichbar sein. An dieser Stelle fließen Informationen aus der Datenvorverarbeitung wie Anomaliekriterien und Charakteristika der Verteilung ein. Die datengetriebene Optimierung erfolgt mit substituierbaren gradientenfreien Algorithmen.

Die Handlungsempfehlungen werden im Einklang mit den Vorgaben der DIN EN ISO 9241-110 (2020) zur technischen Dialoggestaltung bereitgestellt und visualisiert. Es erfolgt ein Ranking der Prozessvarianten, in dem sie nach Optimalität und Komplexität der Anweisung abgestuft werden. Während der Zusammenstellung wird sichergestellt, dass die aufgelisteten Anweisungen sich hinreichend voneinander unterscheiden. Über die vom System automatisch generierten Maßnahmen hinaus lassen sich Eingaben manuell vom Mitarbeiter vornehmen.

4.1.3 Datenmanagement und Schnittstellen

Die datengetriebene Optimierungskette baut auf dem unternehmenseigenen Datenmanagement auf. Vorliegende Schnittstellen und IT-Systeme müssen bei der praktischen Implementierung berücksichtigt werden. Abbildung 4.3 zeigt die Ergänzung bestehender Prozessdatenstrukturen in Form der Automatisierungspyramide nach Babel (2021). Zuerst muss eine Vorverdichtung erfolgen, um eine Dimensionalität zu gewährleisten, die den Anforderungen der späteren Analyse entspricht. Sie kann bereits vor dem Abspeichern in die Prozessdatenbank vorgenommen werden, wenn eine komplette Übernahme wirtschaftlich und technisch nicht sinnvoll ist. Mittels eines ETL-Prozesses werden aus der bereits bestehenden digitalisierten Prozesskette aus unter anderem Ofenparametern, Formherstellung und Abguss die relevanten Teildatenmengen extrahiert. Gesondert neben dem Prozess erhobene Daten umfassen die thermische Analyse und Laborwerte. Sie werden meist in einer anderen Periodizität ermittelt. Hinzu kommen Daten aus ERP-Systemen, die Informationen über Produktqualitäten und bereitgestelltes Einsatzmaterial von Lieferanten enthalten. Im Rahmen der Offlineanalyse können auch Kundenrückmeldungen hinzugezogen werden, die unter Umständen erst Monate nach der eigentlichen Produktion anfallen. Die Einbringung unstrukturierter Daten und Dokumente aus weiteren Quellen wird von Gröger (2015, S. 13) beschrieben, für die methodische Ausarbeitung aber nicht benötigt.

Die gewonnenen Daten speichert eine auf IoT-Anwendungsfälle spezialisierte Prozessdatenbank, die für große Datenvolumina ausgelegt ist und auf die sowohl Prozessexperten als auch Analysten als Datawarehouse zugreifen können. In der Phase der Datenintegration sollten aus Gründen der Kompatibilität und der späteren Analyse möglichst tabellarische Strukturen im Rahmen von relationalen Datenbanksystemen (RDBMS) umgesetzt werden. Falls notwendig, können Einzeldatenbestände jedoch auch als NoSQL dokumentenbasiert hinterlegt werden. Letzteres eignet sich besonders für prototypisch

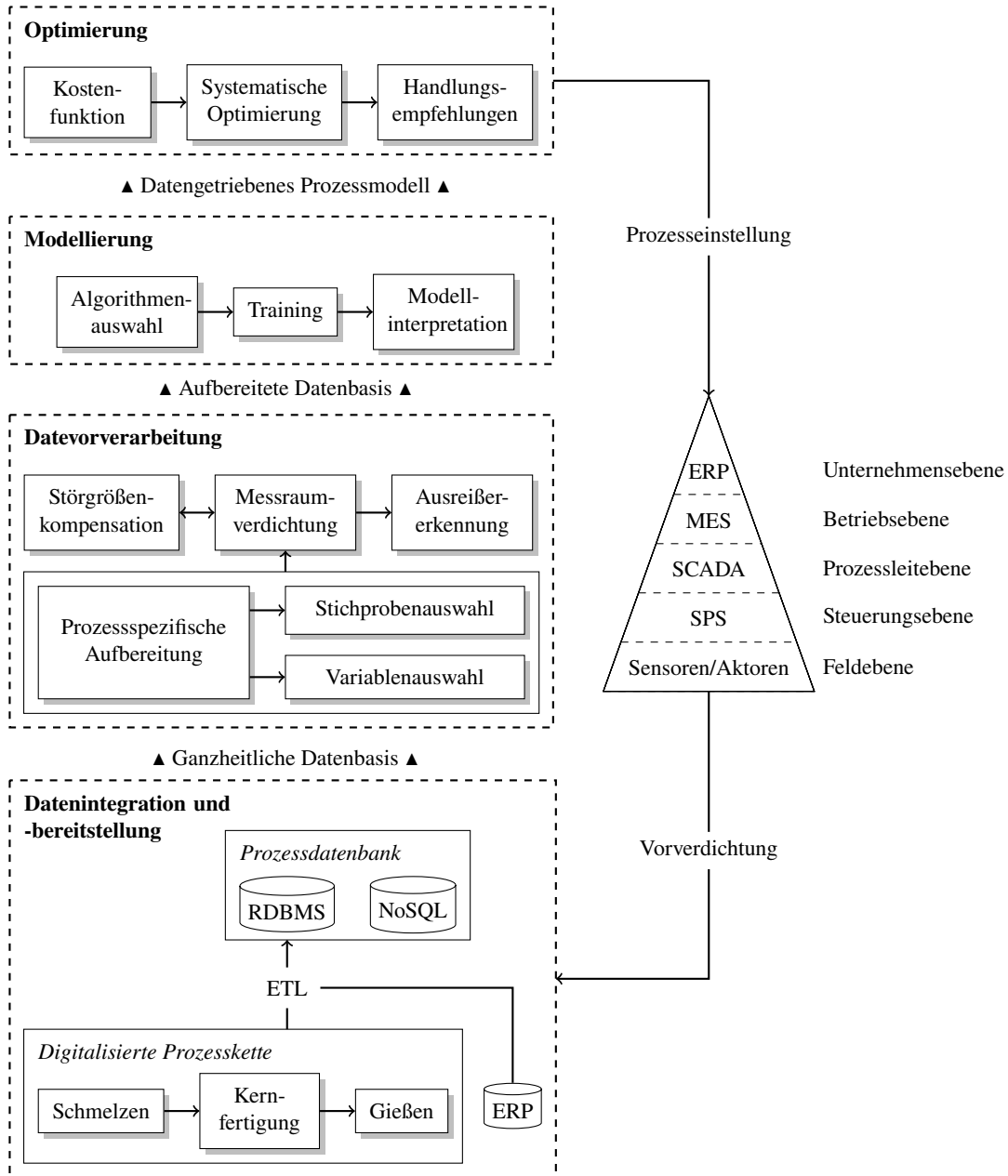


Abbildung 4.3: Datenmanagement und Schnittstellen bei Nutzung der Optimierungskette unter Berücksichtigung der firmenspezifischen IT-Infrastruktur in Anlehnung an Gröger et al. (2012), eigene Darstellung.

installierte IoT-Sensorik in Kombination mit der Verwendung schlanker Datenströme. Zur Bereitstellung einer ganzheitlichen Datenbasis als Endprodukt der Datenintegration und –bereitstellung müssen die durch den ETL-Prozess gewonnenen Daten an einem gemeinsamen Ort vorzufinden sein, können aber noch in den spezifischen Tabellenstrukturen vorliegen, in denen sie erhoben wurden, siehe Abbildung 4.3 unten.

Anschließend sind die Daten im Rahmen einer prozessspezifischen Aufbereitung zweckorientiert zuzuordnen, was sich in vielen Gießereien als Herausforderung darstellt und eine präzise Einzelfallbetrachtung erfordert. In Abschnitt 4.2.1 werden mehrere Anwendungsmöglichkeiten beispielhaft vorgestellt. Die Aufgabe ist, ein adäquates Datenmodell zu entwickeln, das die Teildatenbestände aus den heterogenen Quellen des Prozesses mit Hilfe von Domänenwissen miteinander vereint. Das Ergebnis der prozessspezifischen Aufbereitung ist eine Datenmenge, die sowohl in puncto Dimensionalität als auch Format zur Analyse durch maschinelles Lernen genutzt werden kann. Im Anschluss werden eine Stichprobenauswahl zur Eingrenzung der Analysezeiträume und eine Variablenselektion für das Modell durchgeführt. Die Selektion wird bis zu diesem Zeitpunkt durch den Nutzer veranlasst. Eine automatisch durchgeführte Erkennung von Ausreißern, eine Messraumverdichtung zur Extraktion relevanter Features und die Kompensation externer Störgrößen produzieren als Resultat eine aufbereitete Datenbasis in tabellarischer Form, die zur Modellierung genutzt werden kann. Nach der Auswahl der Modellierungsalgorithmen, dem Trainingsvorgang und der anschließenden Evaluierung steht ein datengetriebenes Prozessmodell bereit, das plattformunabhängig und modular Prognosen erstellen kann. Die Optimierungsschleife, bestehend aus Kostenfunktion, systematischer Optimierung und generierten Handlungsempfehlungen, liefert nach Abschluss eine Prozesseinstellung, die sich innerhalb der unternehmerischen Dateninfrastruktur weiterverarbeiten lässt.

4.2 Datenvorverarbeitungsmodul

Die Datengüte der in Abschnitt 4.1.2 identifizierten prozessrelevanten Einflussgrößen ist nun zu bewerten. Sie drückt sich im Rauschen, in Ausreißern in Multikollinearität und in Aufnahmeintervallen aus. Hieraus lassen sich erste Anpassungen an die Erhebungsmethodik ableiten, um die qualitätsbestimmenden Eigenschaften jedes Qualitätsschrittes zu ermitteln. Die prozessspezifische Aufbereitung und die Störgrößenkompensation sind nicht erst für die Analyse von Bedeutung, sondern können auch schon bei Dashboardlösungen zum Tragen kommen, wenn Daten beispielsweise anforderungsgerecht visualisiert werden müssen.

4.2.1 Prozessspezifische Aufbereitung

Die prozessspezifische Aufbereitung der ganzheitlichen Datenbasis ist eine anwendungsabhängige Aufgabenstellung und der Heterogenität der Datenquellen geschuldet. Optimale Ergebnisse bei der Nutzung von maschinellem Lernen können nur dann gewährleistet werden, wenn die Rückverfolgbarkeit von Prozesszuständen zu jedem Zeitpunkt gegeben ist. Nur so lassen sich im Nachhinein Ursache-Wirkung-Zusammenhänge herstellen. Im Kontext der Gießerei entfällt ein Großteil der Arbeit nicht auf die Anwendung der eigentlichen Optimierungsverfahren, sondern auf die Aufbereitung hin zu einem zusammenhängenden analysierbaren Zieldatenbestand. Eine verallgemeinerte Lösungsstrategie lässt sich hierfür

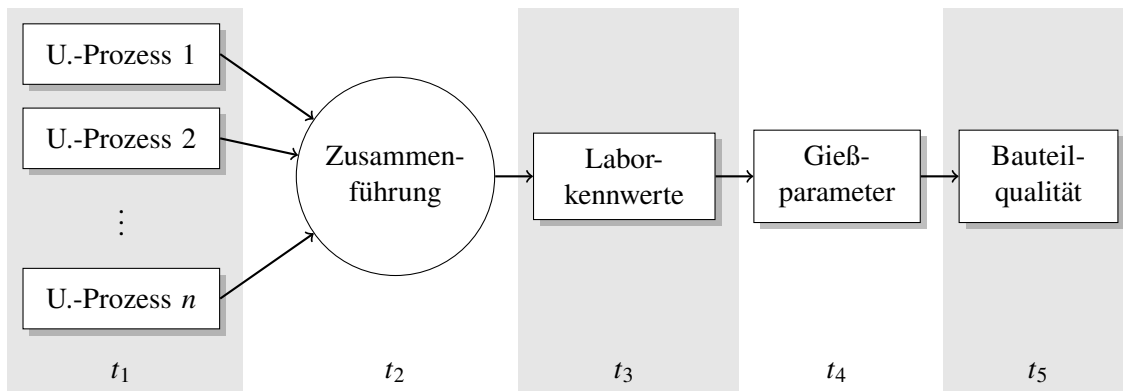


Abbildung 4.4: Beispielprozess zur Datenerhebung im Rahmen der prozessspezifischen Aufbereitung. Als Referenz dient eine Fertigung nach dem Kernpaketverfahren. Daten der einzelnen Prozessschritte fallen zu den Zeitpunkten t_1, \dots, t_5 an. Eigene Darstellung.

nicht formulieren. In diesem Abschnitt werden ausgewählte Ansätze vorgestellt, die genutzt werden können, um eine brauchbare Zieldatenmenge zu erhalten und mit der Analyse fortzufahren. Die Zusammenführung der Zieldatenmenge erweist sich häufig insofern als problematisch, als sich bestimmte Kenngrößen nicht eindeutig einer Instanz des Prozesses zuordnen lassen. Dies ist beispielsweise der Fall, wenn Messungen mit unterschiedlicher Periodizität vorgenommen werden. Bei der Zusammenführung ist es dann unumgänglich, Mehrfachzuordnungen vorzunehmen oder Instanzen zu entfernen, da die Dimensionen der Datenmengen nicht übereinstimmen. Beide Vorgehensweisen beeinflussen die Datenqualität negativ. Der Informations- und Varianzverlust schlägt sich schließlich in einer verschlechterten Prognosegenauigkeit nieder.

Ein weiteres Problem, das sich in der Prozessindustrie abzeichnet, sind fehlende Zuweisungsalgorithmen zur Berücksichtigung von zeitlichem Versatz in Datenmengen. So lassen sich Produktionsparameter und Messdaten, wenn sie mit gleicher Periodizität, aber an unterschiedlichen Messstellen erhoben werden, nicht miteinander kombinieren, da die Zeitdifferenz zwischen ihnen nicht bestimmbar ist. Zeitversatz innerhalb der Datenbasis verhindert die Identifikation wichtiger relationaler Zusammenhänge. Aus Sicht des Prozesses lassen sich durch zeitlich disparate Messwerterhebungen Wirkungsketten nicht mehr nachvollziehen, sodass auch Domänenwissen nur beschränkt nutzbar ist. Datengetriebene Verfahren zur Wissensgenerierung setzen ein durchgängig einheitliches Koordinatensystem voraus.

A posteriori angewandte datengestützte Methoden der Zusammenführung disparater Zeitreihen lösen das Problem allenfalls behelfsmäßig. Nachweislich zuverlässiger erweisen sich Markierungen von Bauteilen zur digitalen Rückverfolgung. Diese können beispielsweise in den Formstoff eingebracht und nach dem Gießen ausgelesen werden (Sandt et al. 2020). Alternativen sind die Anbringung einer Vorrichtung in der Kokille zur Abbildung eines Datamatrix-Codes am Gussteil (Meissner und König 2011) oder das Eingießen eines gekapselten RFID-Transponders in das Bauteil (Pille und Rahn 2017). Technische Maßnahmen zur Bauteilrückverfolgung gewähren bei korrektem Einsatz eine hundertprozentige korrekte Zusammenführung, sind jedoch mit Kosten bei der Installation und der Nutzung verbunden.

Abbildung 4.4 zeigt eine stark vereinfachte Fertigungslinie, die das Aufkommen disparater Dimensionalität und versetzter Zeitreihen veranschaulicht. In dem Beispiel handelt es sich um eine fiktive Fertigung nach dem Kernpaketverfahren, beim dem Einzelkerne zu einer Gießform zusammengesetzt werden. Nähere Informationen zum Prozess sind Langer und Dunnivant (2011, S. 290–298) und Brown (1999, S. 135–148) zu entnehmen. Die Grundproblematik lässt sich auch auf andere Prozesse übertragen. Der Schmelze werden zu bestimmten Zeitpunkten Proben entnommen, die im Labor auf ihre chemischen Komponenten analysiert werden. Dies wird in der Regel mehrfach pro Schicht, nicht jedoch einzelteilbezogen durchgeführt. Hierdurch liegen deutlich weniger chemische Analysen als andere Prozessgrößen vor. Die beschreibenden Parameter des Gießvorgangs werden in einem späteren Schritt mit enger zeitlicher Taktung oder bauteilspezifisch erhoben. Abhängig von den Qualitätsanforderungen werden die Gussteile im Anschluss zu 100 % manuell oder durch Röntgenuntersuchung inspiziert. Als Alternative können stichprobenartige Untersuchungen durchgeführt werden. Unter Umständen liegen Rückmeldungen zur Qualität erst lange Zeit nach der Fertigung vor, wenn Fehler beispielsweise während der mechanischen Nachbearbeitung in Folgeprozessen bei Kunden in Erscheinung treten.

Das Ziel ist es, die Bauteilqualität zum Zeitpunkt t_5 in Relation zu den Fertigungsparametern $\mathbf{x}_i(t_j)$ zu setzen. Die prozessspezifisch aufbereitete Datenmatrix des in Abbildung 4.4 gezeigten Beispiels

$$\mathbf{X}(t_5) = \begin{bmatrix} | & & | & & | & & | \\ \mathbf{x}_1(t_1) & \dots & \mathbf{x}_{n_2}(t_2) & \dots & \mathbf{x}_{n_3}(t_3) & \dots & \mathbf{x}_n(t_4) \\ | & & | & & | & & | \end{bmatrix} \quad (4.1)$$

mit

$$t_1 < t_2 < t_3 < t_4 \quad (4.2)$$

verknüpft die Parameter $\mathbf{x}_i(t_j)$, die zum Zeitpunkt t_j erhoben werden, miteinander. Liegt zwischen den zwei Teilprozessen zu den Zeitpunkten t_i und t_{i+1} keine eindeutige Zuordnung vor, kann versucht werden, relevante Features der beiden Zeitreihen zu extrahieren und darüber eine Verknüpfung herzustellen. Mathematisch lässt sich dies über *Dynamic Time Warping* (DTW) realisieren (Keogh und Ratanamahatana 2005), das bis dato größtenteils in der Verarbeitung von Audiosignalen Verwendung findet (Müller et al. 2006). Eine wichtige Voraussetzung ist, dass das Aufkommen der Merkmale, die zur Orientierung dienen, zeitlich miteinander korreliert. Hierfür können im Regelfall keine Sollwerte der Maschinen herangezogen werden, da diese häufig über mehrere Prozesszyklen hinweg konstant gehalten werden. Das Ergebnis einer erfolgreichen Anwendung von DTW wäre eine einzelteilbezogene Darstellung, bei der die Dimension am Ende jedoch nicht mehr stimmt, da Punkte einer Zeitreihe dupliziert werden müssen. Als Alternative ließe sich die Annahme einer konstanten Durchlaufzeit für die Zuordnung nutzen. Die Differenz wird von t_{i+1} abgezogen und dann der am nächsten liegende Zeitpunkt t_i gesucht.

Im Falle der disparaten Dimensionalität kann das Beispiel der Ermittlung einer chemischen Zusammensetzung anhand einer Probe im Labor zur Veranschaulichung herangezogen werden. Die ermittelten Kennwerte lassen sich im Normalfall nicht auf einzelne Gussteile beziehen, da sie lediglich den Zustand der Schmelze zu einem gegebenen Zeitpunkt beschreiben. Letzterer kann jedoch als Ausgangspunkt für eine Vorwärtssuche fungieren. So werden alle Abgüsse, die danach erfolgen, dieser chemischen Analyse

zugeordnet, bis eine neue durchgeführt wird. Der Nachteil dieser einzelteilbasierten Darstellung liegt darin, dass die zugeordneten Kennwerte für mehrere Teile identisch sind, was die Ermittlung der Wirkzusammenhänge im Nachgang erschwert. Aus diesem Grund können die betreffenden Zeiträume einer Analyse alternativ in Form von Mittelwerten zusammengefasst werden.

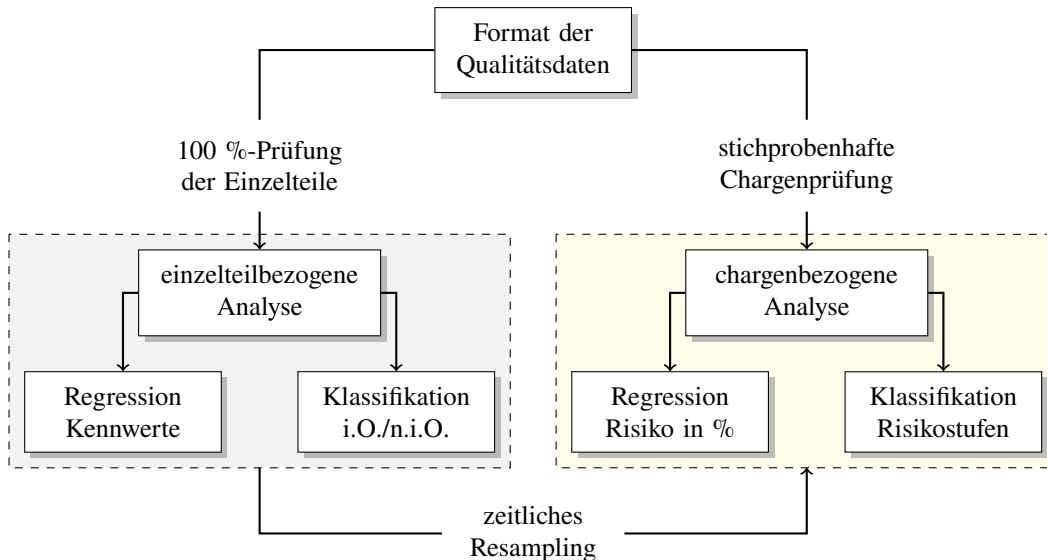


Abbildung 4.5: Möglichkeiten für die Aufbereitung der Qualitätsdaten.

Für die Art und Weise der Datenzusammenführung ist nicht zuletzt die Erhebung der Qualitätsdaten ausschlaggebend. Abbildung 4.5 zeigt mehrere Beispiele hierfür. Werden gefertigte Gussteile aufgrund personeller Kapazitäten oder anderer Rahmenbedingungen nur stichprobenartig geprüft, muss von den Ergebnissen auf eine gesamte Charge zurückgeschlossen werden. Eine Analyse erfolgt dann auf dieser Datenbasis und hat den prozentualen Ausschuss der Charge als Zielgröße und die restlichen Prozessparameter müssen ihr kumuliert zugeordnet werden. Ein weiterer Fall, der bei einer einzelteilbezogenen Analyse auftreten kann, sind über einen Zeitraum konstante Fertigungsparameter bei sich änderndem Prozessergebnis. Eine Vorhersage mit einer solchen Datenbasis ist aus mathematischer Sicht nicht zielführend, da sich aus ihr keine kausalen Zusammenhänge ableiten lassen. Die Daten sollten auf festgelegte Zeitintervalle hinsichtlich ihrer Mittelwerte zusammengefasst und eine Prognose basierend auf der Ausfallwahrscheinlichkeit der Einzelteile durchgeführt werden.

4.2.2 Datentransformation

Eine Transformation der Prozessdaten hat verschiedene Vorteile. Sie kann Rauschen entfernen oder die Datenmenge so verkleinern, dass ein Modell sie effizienter verarbeiten kann. Die Transformation und Rücktransformation erfolgt vor bzw. nach der Prognose und kann sowohl auf die unabhängigen als auch auf die abhängigen Variablen angewendet werden. Abbildung 4.6 zeigt eine generalisierte Darstellung der Datentransformation in Verbindung mit einer Prognosefunktion. In dieser Arbeit werden vorwiegend Dimensions-

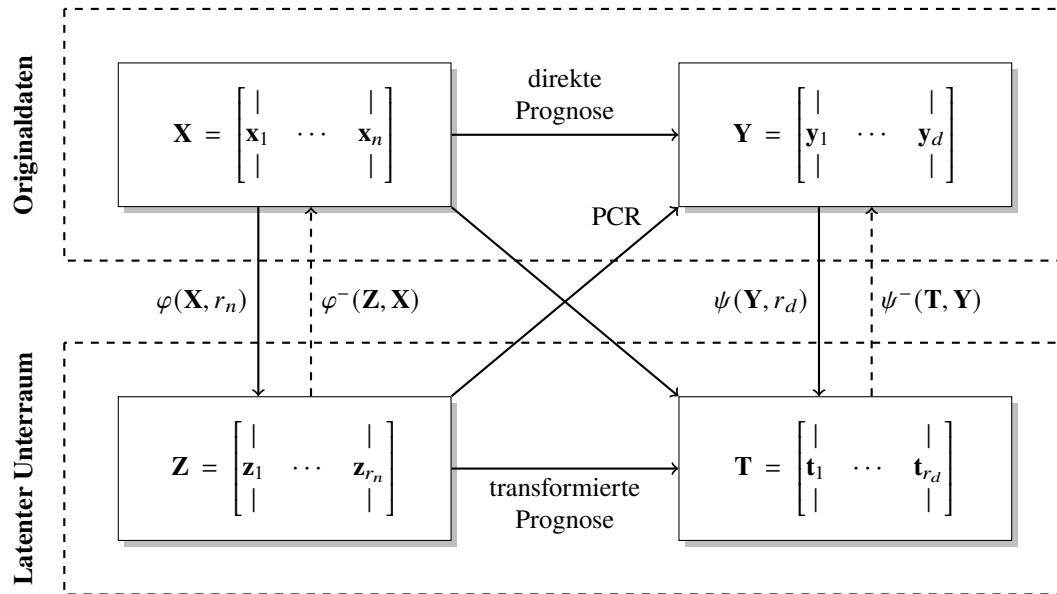


Abbildung 4.6: Anwendung der Dimensionsreduktion auf Prognosefunktionen. Die Dimensionsreduktion kann als Datentransformation für funktional unabhängige und abhängige Variablen gleichermaßen dienen. Eine Rücktransformation mittels φ^- bzw. ψ^- kann nur unter Informationsverlust durchgeführt werden.

und Rangreduktionen als Transformation durchgeführt. Die Umwandlung von \mathbf{X} in \mathbf{Z} erfolgt durch die Funktion φ , die wiederum von \mathbf{X} selbst und einem festgelegten Rang r_n abhängt. Anschließend besitzt \mathbf{Z} die Dimensionalität r_n . Um die Originaldaten zurückzuerhalten, kann die Rücktransformation φ^- genutzt werden. Sie ist jedoch nur unter Informationsverlust nutzbar, da die gekürzten Ränge bei der erneuten Projektion in den Originaldatenraum nicht wiederhergestellt werden. Die Umwandlung der abhängigen Variablen \mathbf{Y} erfolgt analog. Eine häufig verwendete Methode in diesem Zusammenhang ist die unter *Hauptkomponentenregression* bekannte Prognose der funktional abhängigen Daten durch transformierte unabhängige Daten.

Die Dimensions- und Rangreduktion kann auf unterschiedliche Weise genutzt werden, um eine prognosebasierte Fertigungsoptimierung zu unterstützen. Abbildung 4.7 zeigt verschiedene Anwendungsszenarien einer Rang- oder einer Dimensionsreduktion. Für die Prozessdatenanalyse werden weitreichende Mehrwerte durch Ermittlung einer latenten Basis mit einem intrinsischen Rang r generiert, der durch optimales Thresholding ermittelt wird (Brunton und Kutz 2019, S. 37). *Thresholding* bezeichnet in diesem Fall das Abschneiden bzw. Kürzen kleiner Singulärwerte, indem sie zu Null gesetzt werden. Das Vorgehen lässt sich jedoch auch auf Fourier-Koeffizienten oder Gewichte eines Autoencoders ausweiten. Es wird zwischen einer Messraumreduktion und einer Rangreduktion bei gleichbleibender Dimension unterschieden. Für beides ist die Grundlage die latente Basis, dabei spielt es keine Rolle, mit welchem Verfahren sie ermittelt wird.

Bei der FFT kann der Threshold in Form der anteilig größten Fourier-Koeffizienten festgelegt werden. Sie erlauben eine direkte Steuerung der Verdichtungsgrades. Die Bestimmung des reduzierten Ranges der SVD erfolgt analog nach festen Kriterien. Auch sie erlaubt, den Grad der Komprimierung explizit anzugeben. Bei der RPCA hingegen

lässt sich der reduzierte Rang nur implizit durch die Stoppbedingung der iterativen Bestimmung der \mathbf{L} -Matrix vorgeben. Im *Eigenfaces*-Beispiel von Brunton und Kutz (2019, S. 109) wird er so angesetzt, dass die Berechnung stoppt, sobald die Differenz $\mathbf{X} - \mathbf{L} - \mathbf{S}$ ein Vielfaches der Frobenius-Norm der Datenmatrix \mathbf{X} unterschritten hat. Der Faktor wird in diesem Fall mit 10^{-7} angegeben, ist aber für jeden Anwendungsfall neu anzupassen. Bei Autoencoder-Strukturen hängt das Thresholding maßgeblich von der Netztopologie ab, da der Verdichtungsgrad auf die Zahl der Neuronen in der mittleren Schicht, dem *Bottleneck*, zurückzuführen ist. Darüber hinaus entscheiden die Aktivierungsfunktionen, die Solver und die Initialbedingungen über ermittelten Kantengewichte und damit das Verhalten des Netzes.

4.2.3 Thresholding der Prozessdaten

Im Vorfeld der eigentlichen Beschreibung des Prozesses können mittels Messraumverdichtung bereits wichtige Informationen über ihn gewonnen werden. Brunton et al. (2016) führten eine Reduktion der Messpunkte bei der Gesichtsdatenverarbeitung durch und konnten feststellen, dass die PCA eine Platzierung der Sensorik an den Stellen wie Nase und Augen vorschlägt, die auch der menschlichen Wahrnehmung als relevant erscheinen. Auf diese Weise lassen sich die Varianzen einzelner Prozessparameter für sich oder ihre Relevanz hinsichtlich der Vorhersage einer Zielgröße beurteilen. Die Messraumverdichtung dient also zuallererst dem Informationsgewinn bzw. der Featureextraktion. Einer effizienteren Speicherung von Prozessdaten durch Verdichtung wird vor dem Hintergrund immer stärker steigender Datenvolumina eine hohe Bedeutung beigemessen. Da sich jedoch zahlreiche herstellereinspezifische Standards für die zeitliche Komprimierung gesammelter Daten durchgesetzt haben, ist dieser Aspekt zunächst kein Untersuchungsgegenstand dieser Arbeit. Performancevorteile ergeben sich vorwiegend bei der Optimierung der Prozessvarianten. Dies gilt sowohl für die Eingangs- als auch die Ausgangsseite der Prozessdaten: Je weniger Features zu verarbeiten sind, desto einfacher und schneller lassen sich Modelle trainieren und Prognosen durchführen. Die Zahl der zu berücksichtigten Ausgangsgrößen entscheidet darüber hinaus über den Rechenaufwand durch die auszuführenden Prognosen in einem Optimierungsproblem. Dies unterliegt der Annahme, dass für jede Zielgröße die Prognosefunktion einmal aufgerufen werden muss.

Für die Anomalieerkennung auf Grundlage einer rangreduzierten Darstellung des Prozesses sind der Rang r und der Threshold der Fehlerdetektion zu bestimmen. Die Wahl des Thresholds wird anwendungsbezogen oder unter Nutzung des χ^2 -Konfidenzintervalls vorgenommen (Qin 2009). Der optimale intrinsische Rang des Systems r wird meist als bekannt vorausgesetzt, kann sich aber dennoch von dem der Messraumverdichtung unterscheiden. Je nach Charakteristik der Anomalien und der Wichtigkeit einzelner Instanzen der Datenmenge sind beide Thresholds anzupassen. Letztlich bilden die rangreduzierten Prozessdaten auch die Basis für eine optimale robuste Störgrößenkompensation. Wird der Rang zu klein gewählt, sind die Daten zu stark verdichtet und vom Original verschieden. Die Prognosen werden dadurch besser, da weniger Rauschen modelliert werden muss, sie verlieren jedoch an Aussagekraft für den Prozess. Ist der Rang zu groß, ähnelt die Darstellung zu sehr den Originaldaten, sodass außer der Komprimierung kein Mehrwert erzielt wurde.

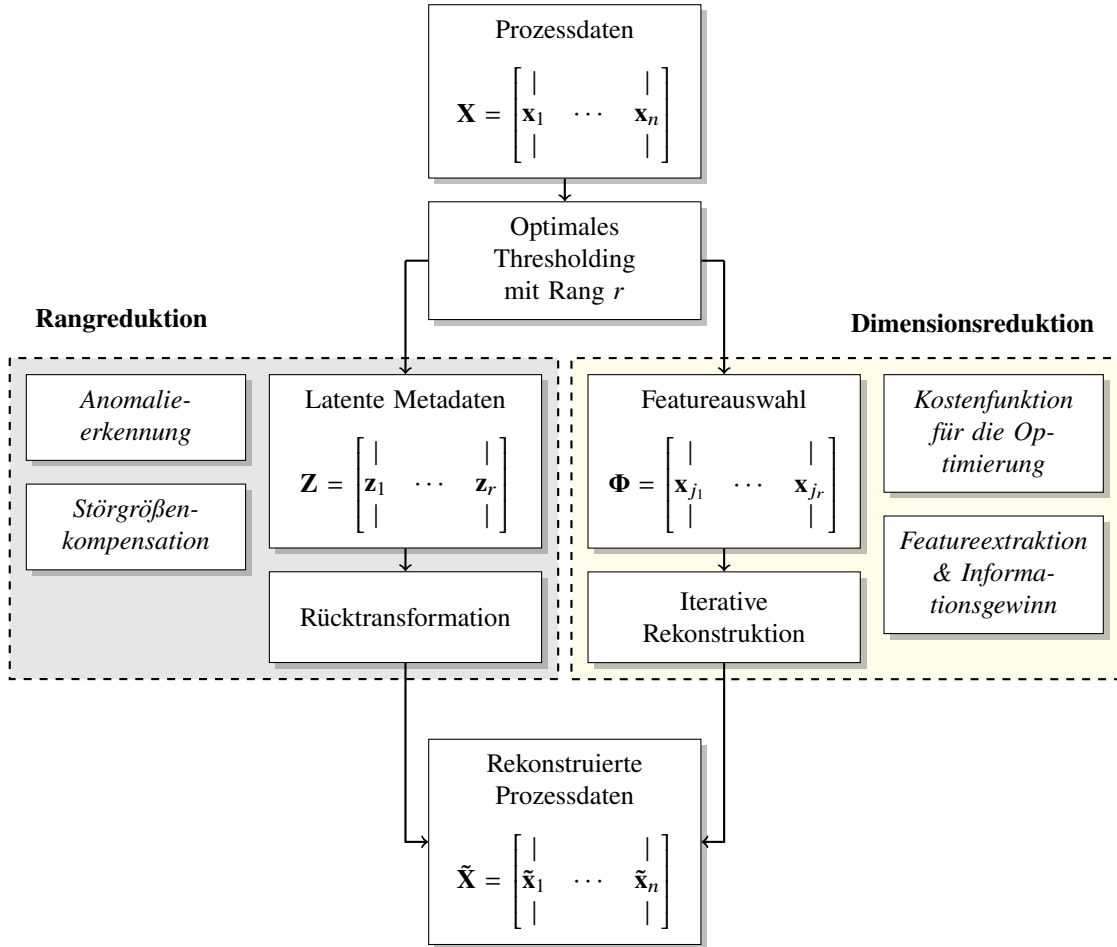


Abbildung 4.7: Nutzung der Prozessdaten zur Rang- und Dimensionsreduktion. Ausgangspunkt für beide ist die Festlegung des intrinsischen Rangs r . Anschließend können r latente Metavariablen gebildet oder r einzelne Spalten von \mathbf{X} zur Verdichtung extrahiert werden.

Thresholdbasierte Messraumverdichtung mit SVD

Eine Reduktion des Variablenraumes kann durch die Identifikation linear unabhängiger Spalten einer Datenmatrix durchgeführt werden. So können aus einer Matrix mit numerischem Rang r nach dem Thresholding die r relevantesten Features extrahiert werden. Zunächst wird von $\mathbf{X} \in \mathbb{R}^{m \times n}$ die Singulärwertzerlegung

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (4.3)$$

gebildet. Zur Rangreduktion wird in der Matrix \mathbf{S} der kleinste Singulärwert durch 0 ersetzt. Die resultierende Matrix \mathbf{S}_1 wird zur Berechnung der rangreduzierten Matrix

$$\mathbf{R}_1 = \mathbf{U}\mathbf{S}_1\mathbf{V}^T \quad (4.4)$$

genutzt. Sie hat die gleichen Dimensionen wie \mathbf{X} , jedoch den Rang $r - 1$. Anschließend wird der maximale Abstand der originalen Datenmatrix \mathbf{X} mit dem der rangreduzierten Matrix \mathbf{R}_1

$$D_{max}(\mathbf{X} - \mathbf{R}_1) := \max\{|x_{ij} - r_{1ij}|\} \quad (4.5)$$

für $1 \leq i \leq m$ und $1 \leq j \leq n$ berechnet. Bei einem ausreichend kleinen Abstand kann die Matrix \mathbf{S}_2 durch das Ersetzen des zweitkleinsten Singulärwerts durch 0 bestimmt und damit

$$\mathbf{R}_2 = \mathbf{U}\mathbf{S}_2\mathbf{V}^T \quad (4.6)$$

analog zur Gleichung (4.4) ermittelt werden. Das Verfahren wird so lange fortgesetzt, bis der Abstand zur Originalmatrix \mathbf{X} zu groß wird. Sollte der numerische Rang r bereits bekannt oder mit einer der Verfahren aus Abschnitt 2.4.4 ermittelt worden sein, kann \mathbf{R} damit direkt berechnet werden.

Mit Hilfe der rangreduzierten Matrix \mathbf{R} lässt sich schrittweise feststellen, welche Spalten linear unabhängig sind. Zunächst wird nur die erste Spalte betrachtet, die den Rang 1 hat. Dann wird die zweite Spalte hinzugenommen. Besteht eine lineare Abhängigkeit zwischen den beiden, bleibt der Rang 1, ansonsten wird er zu 2. Mit allen weiteren Spalten wird nach der gleichen Vorgehensweise verfahren. Das Ergebnis wird zu einem k -dünnbesetzten Vektor $\boldsymbol{\rho} \in \mathbb{R}^n$ mit

$$\|\boldsymbol{\rho}\|_0 = r \quad (4.7)$$

verarbeitet, der die r linear unabhängigen Spalten bzw. Variablen der rangreduzierten Datenmatrix markiert.

ℓ_2 -optimale Rekonstruktion der Messdaten

Das Ziel von *Compressed Sensing* ist ursprünglich, mit hoher Genauigkeit ein Signal $\mathbf{x} \in \mathbb{R}^n$ zu rekonstruieren, das niedrigrangige kohärente Strukturen hinsichtlich einer bekannten Basis aufweist. Die dafür vorgenommenen Messungen

$$\boldsymbol{\xi} = \mathbf{C}\mathbf{x} \quad (4.8)$$

mit $\boldsymbol{\xi} \in \mathbb{R}^r$ und $\mathbf{C} \in \mathbb{R}^{r \times n}$ können deutlich geringer ausfallen als die Länge des Signals, solange sie dennoch hinreichend häufig und nach dem Zufallsprinzip durchgeführt werden, sie also inkohärent sind (Brunton und Kutz 2019, S. 97). Gute Messmatrizen \mathbf{C} , die ein Signal stichprobenhaft abtasten, können demnach durch normalverteilte Einträge konstruiert werden, die keinem klaren Muster folgen. Als Alternativen werden zufällige Durchmischungen von Einsen und Nullen genutzt, wobei dünnbesetzte Messmatrizen nur wenige Einsen aufweisen. Sie bewirken, dass Elemente der Datenmatrix \mathbf{X} einzeln extrahiert werden.

Der im letzten Abschnitt vorgestellte Ansatz zur Isolation relevanter Features auf Basis der SVD liefert jedoch eine Lösung $\Phi \in \mathbb{R}^{m \times r}$ mit

$$\underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{x}_{j_1} & \mathbf{x}_{j_2} & \dots & \mathbf{x}_{j_r} \\ | & | & & | \end{bmatrix}}_{=\Phi} = \mathbf{C} \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ | & | & & | \end{bmatrix}^T, \quad (4.9)$$

die deterministisch r einzelne Spalten aus \mathbf{X} herausnimmt. Die Messmatrix \mathbf{C} steht somit im Vorhinein fest und ist konstant. Den Autoren zufolge stellt dies eine schlechte Wahl

für \mathbf{C} dar, da zahlreiche Spalten für die Messung außer Acht gelassen werden und somit Informationen verloren gehen. Eine Randomisierung von \mathbf{C} ist jedoch aus Sicht der Modellierung und Optimierung nicht hilfreich, da die Bezugsgrößen hierfür fixiert werden müssen. Aus diesem Grund wird im Folgenden eine ℓ_2 -optimale Rekonstruktion der Originaldaten auf Basis der thresholdbasierten Messraumverdichtung beschrieben.

Zur Berechnung von \mathbf{C} wird zunächst aus ϱ eine Diagonalmatrix gebildet. Aus ihr werden alle Zeilen entfernt, die nur Nullen enthalten. Damit aus \mathbf{X} die Spalten extrahiert werden können, muss sie transponiert werden. Die dimensionsreduzierte Prozessdatenmatrix Φ berechnet sich durch

$$\Phi = \mathbf{C}\mathbf{X}^T \quad (4.10)$$

Die ℓ_2 -optimale Rekonstruktion $\tilde{\mathbf{X}}$ der Originaldaten \mathbf{X} mit Hilfe von Φ zu

$$\Phi^T \Theta = \tilde{\mathbf{X}} \quad (4.11)$$

erfolgt durch elementweise Bestimmung der Koeffizienten θ_i aus

$$\underbrace{\Phi^T \begin{bmatrix} | & | & \dots & | \\ \theta_1 & \theta_2 & \dots & \theta_n \\ | & | & & | \end{bmatrix}}_{=\Theta} = \underbrace{\begin{bmatrix} | & | & \dots & | \\ \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \dots & \tilde{\mathbf{x}}_n \\ | & | & & | \end{bmatrix}}_{=\tilde{\mathbf{X}}}. \quad (4.12)$$

Hierzu sind die n Vektoren mit Regressionskoeffizienten θ_i für $i = 1, \dots, n$, die sich aus der ℓ_2 -Minimierung

$$\theta_i = \arg \min_{\theta'} \|\Phi^T \theta' - \mathbf{x}_i\|_2 \quad (4.13)$$

ergeben, mit Hilfe der Pseudoinversen $(\Phi^T)^\dagger$ zu bestimmen.

4.3 Modellierungsmodul

Die vorangegangenen Unterkapitel 4.1 und 4.2 befassten sich mit den Rahmenbedingungen der Optimierungskette und der Schaffung einer ganzheitlichen Datengrundlage. Diese wird nun für prädiktive Analysen genutzt. Das Modellierungsmodul hat die Funktion, ein Abbild des realen Prozesses anhand der spezifisch aufbereiteten Datenbasis zu erstellen. Es gilt zunächst, automatisiert eine optimale Featuremenge aus dem Datenbestand auszuwählen, die einen Kompromiss aus vertretbarem Verarbeitungsaufwand und hinreichender Prognosegüte darstellt. Weiterhin sind die Hyperparameter des Modells oder des Modellensembles für eine optimale Performance und Generalisierbarkeit systematisch zu bestimmen. Es sei angemerkt, dass zur domänenunspezifischen Featureauswahl und zum Hyperparametertuning zahlreiche hervorragend dokumentierte Softwarebibliotheken zur Verfügung stehen, so beispielsweise *TPOT* (Olson et al. 2016) oder *Auto-Sklearn* (Feurer et al. 2021). Sie führen umfangreiche ressourcenoptimierte Suchen innerhalb der Modellbibliotheken durch und stellen dann ein optimiertes Metamodell bereit. In diesem Abschnitt wird ein ganzheitlicher Ansatz zur Modellentwicklung und -optimierung sowie zur anschließenden empirischen Auswertung zwecks Plausibilitätsprüfung vorgestellt.

4.3.1 Datengetriebene Modellentwicklung und -optimierung

Die Featureauswahl sollte fundiert sowohl anhand von Expertenwissen als auch basierend auf Verfahren in Zusammenarbeit mit den trainierten Modellen durchgeführt werden. Aufbauend auf den in Abschnitt 2.2.4 skizzierten Methoden, können Ansätze zur Selektion herangezogen werden, die sich in der statistischen Modellierung bewährt haben. Univariate Verfahren wie *Select-k-best* ermitteln die k relevantesten Features aufgrund der Korrelation zwischen Ein- und Ausgangsgrößen und erfordern kein vorliegendes Modell. Beim *rekursiven Eliminieren* wird ein Modell zunächst mit der gesamten Featuremenge trainiert. Anschließend werden die Features mit der niedrigsten Relevanz einzeln entfernt und der Trainingsvorgang so lange wiederholt, bis die vorher festgelegte Zahl von Features erreicht ist (Guyon et al. 2002). Bei der sequenziellen Auswahl werden Features dem Modell vorwärts hinzugefügt oder rückwärts weggenommen. Im Gegensatz zur rekursiven Eliminierung wird keine Metrik für die Relevanz wie die PFI benötigt, da lediglich die Modellgüte für die Beurteilung genutzt wird, allerdings ist ein erhöhter Rechenaufwand zu erwarten (Ferri et al. 1994). Bei *modellbasierten Verfahren* werden alle Features gleichzeitig betrachtet und solche, deren Relevanz einen festgelegten Threshold unterschreitet, entfernt (Raschka und Mirjalili 2021, S. 166). Zur Optimierung der Datenzusammensetzung wird zudem häufig eine Kreuzvalidierung eingesetzt, was den Rechenaufwand steigen lässt.

Abbildung 4.8 zeigt schematisch die Vorgehensweise bei der Modellentwicklung unter Berücksichtigung der Featureauswahl und des Trainingsvorgangs. Nachdem die prozessspezifisch aufbereiteten Daten eingegrenzt wurden, werden sie nach einem festen Schlüssel in Lern- und Testdaten aufgeteilt. Mit dem Trainingsdatenbestand erfolgt die Anpassung der modellspezifischen Hyperparameter. Die Vorgehensweise beim Hyperparameter-Tuning besteht aus der Wahl eines untrainierten Prognosemodells aus einer Klassenbibliothek, einem abzusuchenden Parameterraum als Raster und einer Methode zur Auswahl der optimalen Hyperparameterkombination. Um letztere anhand des Modells zu bewerten, wird meist eine Kreuzvalidierung unter Einbezug eines Bewertungskriteriums für die Modellperformance durchgeführt. Als Suchmethoden stehen eine *Grid Search* aller Parameterkombinationen, eine *Random Search* zur Entnahme zufälliger Stichproben im Parameterraum (Bergstra und Bengio 2012) oder *Successive Halving* (Jamieson und Talwalkar 2015) zur Verfügung. Bei Letzterem werden den Hyperparametervarianten anfangs nur eine geringe Anzahl von Stichproben für das Training zur Verfügung gestellt und dann schrittweise mehr zugewiesen, sodass viele Varianten bei der Optimierung frühzeitig ausscheiden. Die Bewertung innerhalb der Kreuzvalidierung erfolgt mit angepassten Gütekriterien, je nachdem, ob es sich um ein Klassifikations- oder ein Regressionsproblem handelt. Mit der ermittelten optimalen Hyperparameterkombination wird das Modell schließlich trainiert. Eine endgültige Bewertung erfolgt mit den bis dahin zurückgehaltenen Testdaten. Auf dieser Grundlage kann folglich entschieden werden, welches Modell sich am besten für die vorliegende Problemstellung eignet und welche gegebenenfalls zu einem Ensemblemodell fusioniert werden sollten.

4.3.2 Prozessspezifische Modellinterpretation

Nachdem ein Vorhersagemodell trainiert wurde, das den vorgegebenen Gütekriterien hinsichtlich Prognosegenauigkeit und Generalisierbarkeit genügt, sind die erlernten Wirk-

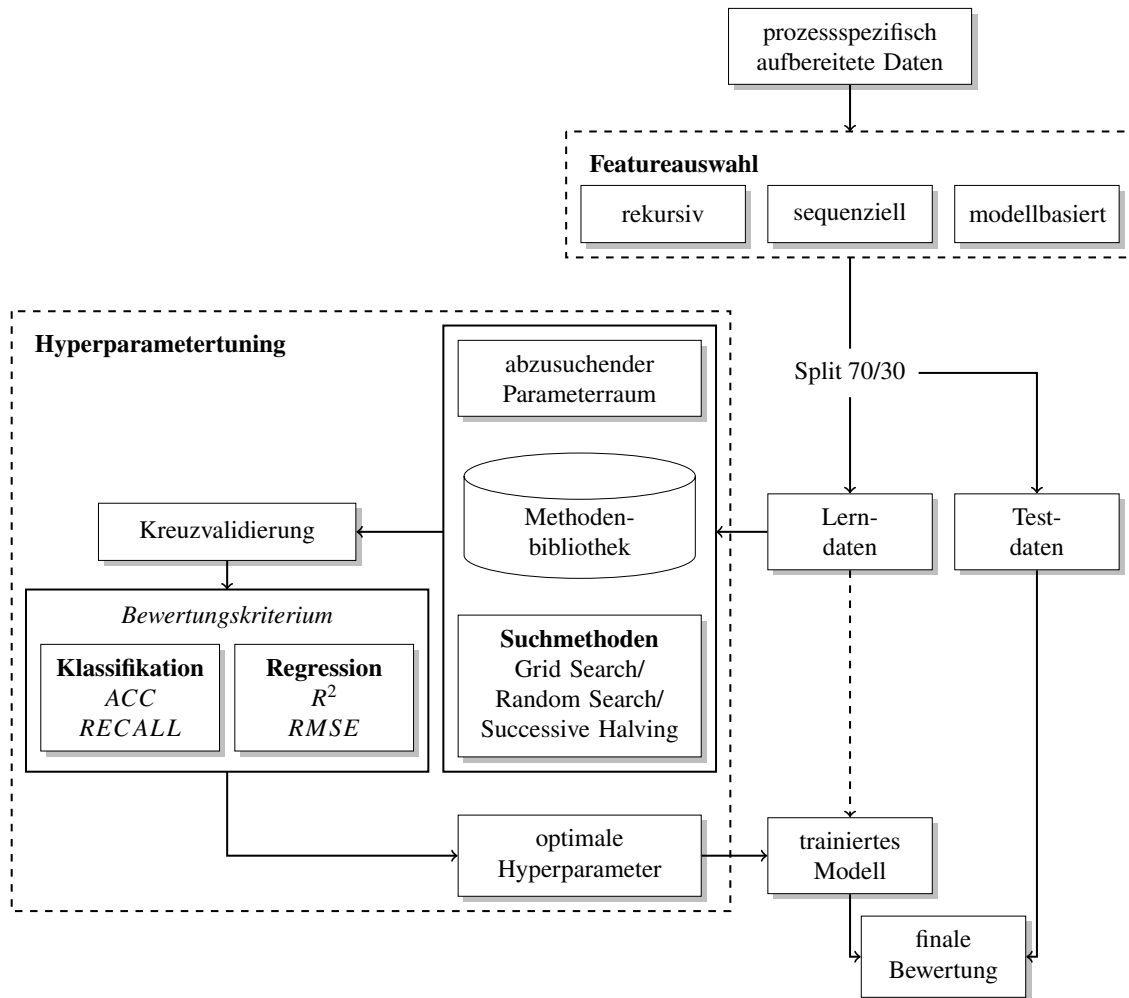


Abbildung 4.8: Prinzip der datengetriebenen Modellentwicklung und -optimierung. Zunächst erfolgt die Featureauswahl mit den aufgeführten Methoden. Anschließend werden die Hyperparameter mit einer Suchmethode unter Nutzung der Kreuzvalidierung und einer Bewertungsmetrik eingestellt. Die Prognosegüte des neu trainierten Modells kann dann zur endgültigen Bewertung mittels Testdaten bestimmt werden.

zusammenhänge auf ihre Plausibilität zu überprüfen. Ein wichtiger Untersuchungsgegenstand ist die Einhaltung der Kausalkette bei Mehrgrößenprognosen. Hierzu wird im Folgenden davon ausgegangen, dass für jede abhängige Zielgröße ein Einzelmodell trainiert wird, da dies die analytische Handhabung der Modellinterpretation erleichtert. Mit agglomerierten Ausgangsvariablen oder transformierten Hauptkomponenten kann gleichermaßen verfahren werden. Entsprechend der vom englischen Statistiker Austin Bradford Hill (1965) aufgestellten Kriterien soll Vorzeitigkeit als notwendige Bedingung der Kausalität gelten (Bradford-Hill-Kriterium Nr. 4). Deshalb dürfen in die Prognose der Zielgrößen einer Prozessinstanz, die zu einem gegebenen Zeitpunkt gemessen werden, nur diejenigen Variablen eingehen, die zu diesem Zeitpunkt bereits wirken bzw. messbar zur Verfügung stehen und die aus diesem Grund als funktional unabhängig anzusehen sind. Ein ähnliches Kriterium gilt für die analytische Modellierung linearer zeitinvarianter

(LTI) Systeme (Unbehauen 2002, S. 10–12).

Für jedes $y_j(t_F)$ kann die zugehörige Prognosefunktion

$$\hat{y}_j = f_j(\mathbf{x}_i) \quad (4.14)$$

für $i = 1, \dots, n$ und $j = 1, \dots, d$ nur mit Prozessgrößen $\mathbf{x}_i(t_m)$ mit dem Zeitpunkt der Messung $t_m < t_F$ trainiert werden. Im Fall der Offlineanalyse im Beispielprozess aus Abschnitt 4.2.1 kann als Forderung formuliert werden, dass einzelteilbezogene Kennwerte, die sich aus der chemischen Analyse ergeben, nicht von Gießparametern abhängen dürfen. Dies ist jedoch insbesondere bei eng getakteten und verknüpften Teilprozessen nicht immer direkt ersichtlich. Eine Möglichkeit, die Kausalität zumindest seitens der Datenzusammensetzung zu erzwingen, ist, die relevanten Features für jede Zielgröße manuell festzulegen. Dies ist jedoch zeitlich aufwändig und birgt die Gefahr der subjektiven Bewertung der Wirkzusammenhänge (Deike 2020, S. 68). Wird jedoch zunächst ein weites Spektrum von Einflussgrößen genutzt, kann die Featurerelevanz nützliche Erkenntnisse über Scheinkorrelationen und unplausible Zusammenhänge liefern.

Zu diesem Zweck ist als erstes die Relevanzmatrix $\text{rel}(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^{d \times n}$ durch

$$\text{rel}(\mathbf{X}, \mathbf{Y}) = \begin{bmatrix} | & | & & | \\ \mathbf{FI}(\mathbf{x}_1, \mathbf{Y}) & \mathbf{FI}(\mathbf{x}_2, \mathbf{Y}) & \dots & \mathbf{FI}(\mathbf{x}_n, \mathbf{Y}) \\ | & | & & | \end{bmatrix} \quad (4.15)$$

mit den Elementen

$$\mathbf{FI}(\mathbf{x}_i, \mathbf{Y}) = \begin{bmatrix} FI(\mathbf{x}_i, \mathbf{y}_1) & FI(\mathbf{x}_i, \mathbf{y}_2) & \dots & FI(\mathbf{x}_i, \mathbf{y}_d) \end{bmatrix} \quad (4.16)$$

zu bestimmen. Dabei steht $FI(\mathbf{x}_i, \mathbf{y}_j) \in \mathbb{R}$ für die Relevanz des Features \mathbf{x}_i für die Vorhersage von \mathbf{y}_j . Die Matrix $\text{rel}(\mathbf{X}, \mathbf{Y})$ kann thresholdbasiert abgesucht werden, um systematische Einflüsse aufzudecken, die über reines Rauschen hinausgehen. So sollte für jede Parametergruppe ein Mindestindex $1 < d_{\min} < d$ definiert werden, ab welchem ein Wirkzusammenhang plausibel ist. Übersteigt die Featurerelevanz $FI(\mathbf{x}_i, \mathbf{y}_j)$ für $j < d_{\min}$ einen Threshold, kann \mathbf{x}_i für die Prognose künftig verworfen oder nach einer Ursache für die Scheinkorrelation gesucht werden. Zur genaueren Untersuchung der Wirkzusammenhänge kann eine Sensitivitätsmatrix $\text{pdp}(\mathbf{x}_i, \mathbf{Y})$ mit

$$\text{pdp}(\mathbf{x}_i, \mathbf{Y}) = \begin{bmatrix} | & | & & | \\ f_s(\mathbf{x}_i, \mathbf{y}_1) & f_s(\mathbf{x}_i, \mathbf{y}_2) & \dots & f_s(\mathbf{x}_i, \mathbf{y}_d) \\ | & | & & | \end{bmatrix} \quad (4.17)$$

für ausgewählte Einflussgrößen \mathbf{x}_i berechnet werden. Sie besteht aus den Sensitivitätsanalysen, beispielsweise der partiellen Abhängigkeit, $f_s(\mathbf{x}_i, \mathbf{y}_j)$ der Zielgröße \mathbf{y}_j von \mathbf{x}_i . Willkürliche Schwankungen und Unstetigkeiten geben hier Aufschluss über die statistische Absicherung des Modells, allerdings sind die typischen Fehlschlüsse zu beachten, die aus modellagnostischen Interpretationsmethoden hervorgehen können (Molnar et al. 2021).

Die soeben aufgeführten Ansätze können bereits insofern als statistische Versuchsplannung aufgefasst werden, als gezielt Parametervariationen gebildet wurden, um das Verhalten des Modells zu untersuchen. Im Gegensatz zur analytischen Modellierung erfüllt

sie bei der datengetriebenen Herangehensweise den Zweck, bereits bekannte Zusammenhänge quantitativ zu validieren und die Richtigkeit der Wirkungsbeziehungen mit Hilfe von Expertenwissen zu bestätigen. Sofern die Zielgrößen nicht nur Einzelkennwerte eines Prozesses beschreiben, sondern Informationen über die Geometrie des Endproduktes enthalten wie in Kapitel 5, lassen sich mit der datengetriebenen Modellierung ganze Bauteile durch eine virtuelle DoE planen. Die genaue Vorgehensweise zur Variation der Parameter wird im nächsten Abschnitt vorgestellt. Eine Sensitivitätsanalyse im Rahmen der empirischen Modellinterpretation verfolgt das Ziel, relative Veränderungen im Ausgang des Modells zu veranschaulichen. Mit Hilfe der datengetriebenen virtuellen DoE lässt sich durch Parametervariation eine Darstellung erzeugen, in der mehrere Bauteile direkt miteinander verglichen werden können. Im Rahmen der Einzel-DoE wird ein Parameter x_i der Prozessinstanz \mathbf{x} zu x_s verändert. Die Auswirkungen von

$$x_{DoE,j} = \begin{cases} x_s, & \text{wenn } i = j \\ x_i, & \text{sonst} \end{cases} \quad (4.18)$$

auf das Prozessergebnis können durch

$$\mathbf{y}_{DoE} = \mathbf{f}(\mathbf{x}_{DoE}) \quad (4.19)$$

visualisiert werden. So lässt sich die durch die trainierten und getesteten Modelle hervorgerufene Prozesstransparenz nicht nur anhand spezifischer Wirkzusammenhänge, sondern am Endprodukt zeigen.

4.4 Optimierungsmodul

Das Prozessoptimierungsmodul hat die Funktion, durch systematische Variation eine Parametereinstellung zu finden, die das Prozessergebnis zu einem Optimum überführt. Seine Ergebnisse werden dem Prozessexperten als Handlungsempfehlungen bereitgestellt. Der Kern dieses Abschnitts ist die systematische datengetriebene Optimierung. Hierfür werden Begrifflichkeiten und grundlegende Konzepte aus der analytischen Prozess- und Topologieoptimierung (vgl. beispielsweise Herrmann et al. 2013, S. 51) implementiert. Die Optimierung stützt sich auf eine datengetriebene Prognose und eine anforderungsgerecht definierte Kostenfunktion. Darauf folgend werden Ansätze vorgestellt, die Kostenfunktion so zu manipulieren, dass ℓ_1 - bzw. ℓ_2 -optimale Prozessvarianten entstehen, die sich den Gegebenheiten der Fertigung anpassen. Weiterhin werden Kriterien vorgestellt, mit denen die physikalische Plausibilität der generierten Varianten während der Optimierungsschleife gewährleistet wird. Schließlich wird ein Konzept zur Bereitstellung der Handlungsempfehlung für Maschinenbediener und den Prozessleitstand vorgestellt. Im Fokus stehen dabei die Mensch-Maschine-Interaktion und ein Ranking der ermittelten Prozessvarianten.

4.4.1 Prinzip der systematischen Optimierung

Grundsätzlich gibt es für Prognosefunktionen auf Basis maschinellen Lernens keine Umkehrfunktion, die es ermöglichen würde, bei einem vorgegebenen Prozessziel bzw. dem Wert einer funktional abhängigen Variablen die dazugehörigen Prozessparameter unmittelbar zu bestimmen. Dies lässt sich mit der fehlenden Injektivität der Funktion begründen;

aus technischer Sicht führen viele Parameterkombinationen, die teilweise weit auseinanderliegen, zu sehr ähnlichen Prozessergebnissen. Eine mögliche Strategie zur Gattierungsplanung und damit verbundenen Kostenoptimierung ist die Rückwärtsanalyse durch die Erstellung von Lookup-Tabellen (Gottschling et al. 2020). Anstelle der Planung der Rohstoffzusammensetzung in Echtzeit werden die Parameterkombinationen offline generiert und die dazugehörigen Prognosefunktionswerte einmalig berechnet. Die aufwändig ermittelten Prozessvarianten können im laufenden Betrieb in einer Rastersuche abgerufen werden, um stabile Prozessfenster zu erhalten, in denen die Zielgröße den Vorgaben entspricht.

In Anbetracht der Fallstudie aus Kapitel 5 und der darin zugrundeliegenden Datenmenge wird im Folgenden ein generischer Ansatz verfolgt, der es ermöglicht, in großen Parameterräumen, in denen eine Rastersuche rechnerisch nicht mehr realisierbar ist, echtzeitgetreu Lösungen zu finden, die einer Zielvorgabe entsprechen. Die Inline-Optimierung erfolgt zu einem Zeitpunkt, an dem noch wirkungsvoll in den Prozess eingegriffen werden kann und alle eingangsseitig benötigten Daten für die Prognosefunktion vorliegen. Grundlegende Lösungsansätze für Klassifikationsprobleme wurden bereits von Gröger (2015, S. 164) und Deike (2020, S. 76) durch Maximierung der Wahrscheinlichkeit zur gewünschten Qualitätsklasse beschrieben. Nachfolgend werden Regressionsansätze gesondert berücksichtigt. Als Grundvoraussetzung für die Verlässlichkeit und die Nutzbarkeit einer durch datengetriebene Optimierung bestimmten Handlungsempfehlung ist die sowohl von Prozessexperten als auch Datenanalysten abgesicherte Prognosegüte und Plausibilität der Modelle.

Abbildung 4.9 zeigt die schematische Übersicht der systematischen Optimierung mit dem Ziel, aus dem gegebenen Prozesszustand heraus auf Grundlage der Prognosefunktion das bestmögliche Prozessergebnis iterativ zu bestimmen. Die in Abschnitt 4.1.2 (vgl. Abbildung 4.2) erfolgte Definition der Ziel-, Steuerungs- und Eingangsgrößen wird als momentgetreues Abbild des Prozesses zugrunde gelegt. Fertigungsparameter, die sich auf abgeschlossene Prozessabschnitte beziehen, müssen zum Zeitpunkt der Optimierung als Eingangsgrößen deklariert werden, auch wenn sie vorher noch beeinflussbar waren.

Für die Ziel- und Steuerungsgrößen werden Rahmenbedingungen festgelegt, innerhalb der die generierten Prozessvarianten verändert werden können und sich das Endergebnis der Optimierung befinden muss. Die Eingangsgrößen bleiben von den Nebenbedingungen unberührt, da sie direkt dem Prozess entnommen werden und eine gesonderte Prüfung durchlaufen. Die vorab definierte Kostenfunktion wertet sowohl die Zielgrößen in Form von Prognosen als auch die zur Generierung von Handlungsempfehlungen genutzten Steuerungsgrößen aus. Dahingegen werden die Eingangsgrößen nur indirekt während der Prognose hinzugezogen. Die Spezifizierung der Kostenfunktion folgt in Abschnitt 4.4.2. Vor der Optimierungsschleife ist die DoE-Sequenz festzulegen. Sie bestimmt den Ablauf der Optimierung, indem angegeben wird, wie viele Iterationen maximal durchlaufen werden und wann ein Optimum erreicht ist. Bei genetischen Algorithmen können hier beispielsweise die Zahl der Generationen und der Individuen, bei der Partikelschwarmoptimierung die Größe des Schwarms übergeben werden. Die Startwerte zur Initialisierung der ersten Prozessvariante werden aus der vorliegenden Instanz des Prozesses übernommen und dem Optimierungsalgorithmus übergeben. Das trainierte Modell errechnet mit dieser Parameterkombination die Zielgrößen. Mit Hilfe der Kostenfunktion werden anschließend sowohl die Zielgrößen als auch die dazu variierten Steuerungsgrößen hinsichtlich der Erfüllung

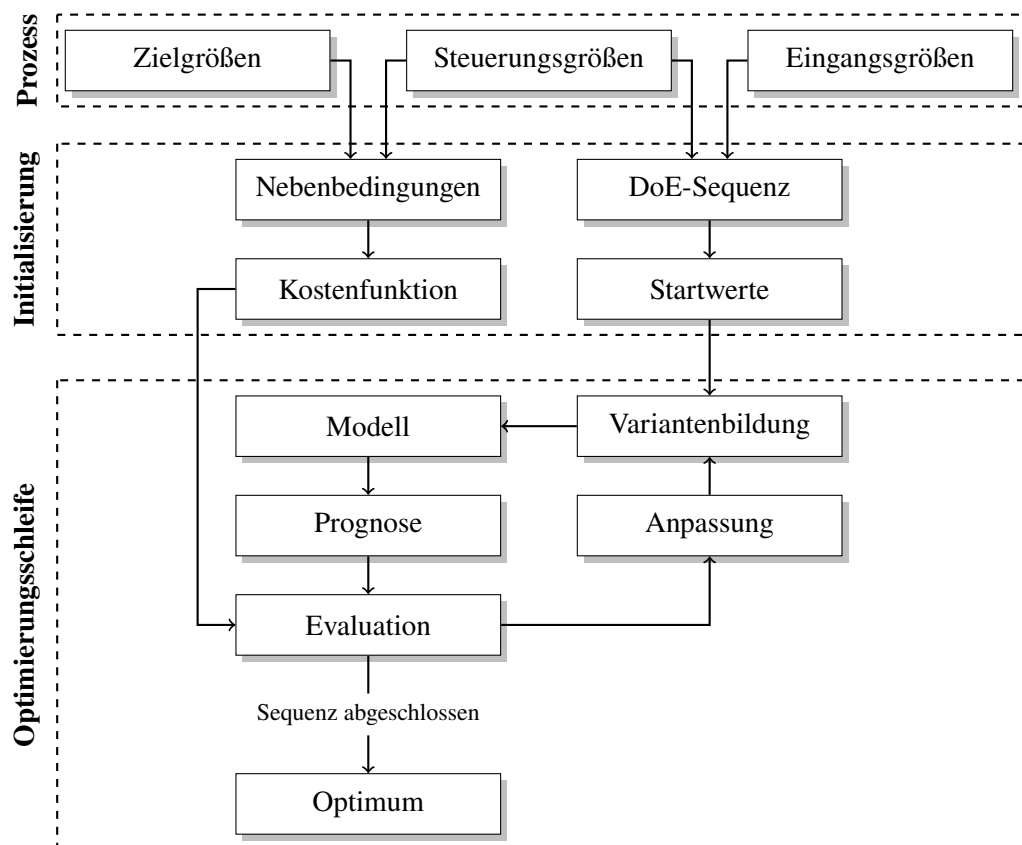


Abbildung 4.9: Prinzip der systematischen Optimierung auf Grundlage datengetriebener Prozessmodelle, eigene Darstellung. Die Steuerungsgrößen und die Kostenfunktion müssen im Vorhinein festgelegt werden. Während der Optimierungsschleife synthetisierte Prozessvarianten werden laufend bewertet und angepasst bis ein Optimum gefunden wurde.

der Qualitätsvorgabe und der technischen Umsetzbarkeit evaluiert. Die Prozessvariante wird schließlich vom Algorithmus angepasst und die Schleife erneut durchlaufen.

So wird bei jedem Optimierungsschritt auch eine mögliche Handlungsempfehlung generiert. Sie wird in diesem Kontext als Differenz der Steuerungsgrößen zum Ist-Zustand vor der Optimierung definiert. Das Ergebnis wird unter den gegebenen Eingangsgrößen als lokal optimal angenommen, wenn die Prognose- und Kostenfunktionen zu einer Lösung konvergiert sind, die hinsichtlich der Vorgaben nicht mehr verbessert werden kann oder die DoE-Sequenz abgeschlossen ist. Die zum Optimum zugehörige Prozesseinstellung wird schließlich als Handlungsempfehlung ausgegeben. Je nach Komplexität des Prozesses und der Bewertung kann es sinnvoll sein, mehrere Varianten nach einem Ranking bereitzustellen.

Der Vorteil bei der Nutzung datengetriebener Modelle für die Prozessoptimierung liegt in der Vielzahl der Varianten, die in einem gegebenen Zeitraum synthetisiert und simuliert werden können. Aufwändiges Hyperparameter-Tuning kann offline vorgenommen werden. Die meisten trainierten Prognosefunktionen liefern ein Ergebnis im Bruchteil einer Sekunde. Eine genauere Untersuchung der Performanceaspekte erfolgt später. Zudem können im Gegensatz zu analytischen Modellen theoretisch nahezu alle Zusammenhänge abgebildet

Tabelle 4.1: Komponenten der Kostenfunktion und ihre Bedeutung für die Optimierung

Faktor	Beschreibung
Γ	Optimalitätskriterium der Zielgröße
Π	Nebenbedingungen der Zielgröße
ζ	Verteilungskriterium der funktional unabhängigen Variablen
Υ	Verteilungskriterium der funktional abhängigen Variablen
Ξ	ℓ_k -Abstand zum Startwert \mathbf{x}_0

werden, solange die Daten dazu in hinreichender Qualität vorliegen. Nachteile einer rein mathematisch empirischen Modellierung sind die fehlende Prüfung der physikalischen Plausibilität und der verfahrenstechnischen Umsetzbarkeit einer Prozessvariante. Extrapolation gelingt bei maschinellem Lernen aufgrund der Größe des Parameterraums meist nur unzureichend und in der Regel unberechenbar. Der Performance-Bottleneck bei der Optimierungsschleife ist der Aufruf der Prognosefunktion innerhalb der Kostenfunktion. Somit skaliert der Rechenaufwand für die Auswertung einer Prozessvarianten mit der Zahl der durchzuführenden Prognosen und ihrer Dauer. Durch eine effektive Messraumverdichtung lässt sich die Zahl der Zielgrößen, die erforderlich sind, um ein Qualitätskriterium adäquat abzubilden, reduzieren.

4.4.2 Definition einer Kostenfunktion

Die Kostenfunktion muss den Echtzeitbedingungen der Optimierung und der Beschreibungsgenauigkeit der Prozessvorgaben Rechnung tragen. Bei der Optimierung mit datengetriebenen Modellen ist zu erwarten, dass einerseits Varianten erzeugt werden, die hinsichtlich ihrer praktischen Umsetzbarkeit nicht plausibel sind, und andererseits unvorhergesehene Prozessergebnisse entstehen. Beide Phänomene können, müssen aber nicht gleichzeitig auftreten. Bei der Variantensynthese muss daher die Grundverteilung sowohl der funktional unabhängigen als der abhängigen Variablen berücksichtigt werden.

Die Kostenfunktion $J(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{x}_0)$ einer generierten Prozessvariante \mathbf{x} und der daraus prognostizierten Zielgrößen $\hat{\mathbf{y}} = f(\mathbf{x})$ wird in ihrer grundlegenden Form durch

$$J(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{x}_0) = \alpha_1 \Gamma(\hat{\mathbf{y}}) + \alpha_2 \Pi(\hat{\mathbf{y}}) + \alpha_3 \zeta(\mathbf{x}, \mathbf{X}) + \alpha_4 \Upsilon(\hat{\mathbf{y}}, \mathbf{Y}) + \alpha_5 \Xi(\mathbf{x} - \mathbf{x}_0) \quad (4.20)$$

beschrieben. Tabelle 4.1 zeigt die wesentlichen Komponenten der Kostenfunktion.

$\Gamma(\mathbf{y})$ beschreibt das Optimierungsziel und hängt einzig von der Prognose der Zielgrößen ab. Die genaue analytische Formulierung muss für jeden Anwendungsfall anforderungsgerecht neu festgelegt werden. Für einkriterielle Zielgrößen ist dies in der Regel die Abweichung zum Sollwert bei Regressionsproblemen oder die Zugehörigkeitswahrscheinlichkeit zur n.i.O.-Klasse bei einer Klassifikation. So kann die Optimierungssequenz abgeschlossen werden, wenn die Differenz zur Prozessvorgabe oder die Wahrscheinlichkeit der Vorhersage eines Bauteilfehlers hinreichend klein ist. Werden mehrkriterielle

Probleme in eine skalare Qualitätskennzahl überführt, kann eine Linearkombination

$$\Gamma(\mathbf{y}) = \sum_{i=1}^k \alpha_i y_k \quad (4.21)$$

oder eine andere an den Anwendungskontext angepasste nichtlineare mathematische Formulierung $\Gamma(\mathbf{y})$ zur Bewertung dienen.

$\Pi(\mathbf{y})$ beinhaltet die d_Π problemspezifisch zu definierenden Nebenbedingungen $g_i(y) \leq 0$ (vgl. Aragón et al. 2019, S. 253) für die Zielgrößen, beispielsweise ein Überschreiten oder Unterschreiten der oberen oder unteren Grenzwerte. Sie können innerhalb der Kostenfunktion als additive Konstanten C_i definiert werden, für den Fall, dass g_i verletzt wird (Smith et al. 1998). Alternativ kann der Strafterm

$$\Pi(\mathbf{y}) = \sum_{i \in d_\Pi} \max(0, g_i(y_i)) \quad (4.22)$$

auch mit dem Grad der Über- oder Unterschreitung linear anwachsen (Chong und Žak 2013, S. 565). Hierzu ist die Abwägung erforderlich, ob Verletzungen der g_i pauschal zur Verwerfung der Prozessvariante führen sollen oder in einem gewissen Ausmaß toleriert werden können. Letzteres kommt zum Tragen, wenn angenommen werden muss, dass das Prozessziel unter gegebenen Eingangsparametern durch Anpassung der Steuerungsgrößen nicht erreicht werden kann. In diesem Fall ist die Kostenfunktion dennoch so weit wie möglich zu minimieren, um in nachfolgenden Prozessschritten gegebenenfalls die Weiterverarbeitung zu erleichtern. Ein direkter Faktor für Werteüberschreitungen von \mathbf{x} muss nicht zwingend definiert werden, da bei der DoE-Sequenz die oberen und unteren Grenzen angegeben werden, innerhalb derer die \mathbf{x}_i variiert werden können.

ζ und Υ sind Distanzmetriken, die den Abstand der generierten Prozessvariante zur Grundverteilung der Daten von jeweils \mathbf{X} oder \mathbf{Y} messen. Nicht zulässige Einzelwerte der Prozessparameter und Zielgrößen können meist durch Expertenkenntnisse explizit angegeben werden. Informationen über die Extrapolationsbereiche der gelernten Modelle, in denen Prognosen mit einer größeren Unsicherheit behaftet sind, liegen ad hoc meist jedoch nicht vor. Hinzu kommt die Tatsache, dass in stark erfahrungsgesteuerten Prozessen selbst den Experten nur implizite Kenntnisse über die verfahrenstechnische Umsetzbarkeit einer Prozessvariante vorliegen. Darüber hinaus treten unter Umständen bei mehrdimensionalen Ausgangsgrößen trotz unauffälliger Parametereinstellungen unerwartete Prozessergebnisse auf, was eine ausgangsseitige Prüfung ebenfalls erforderlich macht. Anhaltspunkte für die physikalische Plausibilität können PCA-basierte oder andere Anomalieindizes wie

$$\zeta(\mathbf{x}, \mathbf{X}) = \left\| \left(\mathbf{I} - \tilde{\mathbf{P}}_X \tilde{\mathbf{P}}_X^T \right) \mathbf{x} \right\|_2^2 \quad (4.23)$$

für \mathbf{x} bzw.

$$\Upsilon(\mathbf{y}, \mathbf{Y}) = \left\| \left(\mathbf{I} - \tilde{\mathbf{P}}_Y \tilde{\mathbf{P}}_Y^T \right) \mathbf{y} \right\|_2^2 \quad (4.24)$$

für \mathbf{y} sein. Sie basieren auf der Basis der SVD von \mathbf{X} bzw. \mathbf{Y} analog zur Definition von Qin (2009). Welcher Anomalieindex die Verteilungen am besten abbildet, ist im Einzelfall empirisch zu prüfen.

Die Abweichung zwischen synthetisierter Variante und Startvektor kann weiterhin hinsichtlich einer ℓ_k -Vektornorm minimiert werden, um Lösungen zu erzeugen, die bezüglich der daraus resultierenden Handlungsanweisung bestimmte Eigenschaften aufweisen:

$$\Xi(\mathbf{x} - \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_k \quad (4.25)$$

So kann die ℓ_2 -Norm als Korrektur analog zur Ridge-Regression gewählt werden, wenn Prozessumstellungen gefordert werden, die auf alle Einstellparameter weitestgehend gleich verteilt sind. Ein möglicher Anwendungsfall ist die Vermeidung drastischer Prozesseingriffe hinsichtlich einzelner Stellgrößen. Alternativ wird die ℓ_1 -Norm wie bei der LASSO-Regression herangezogen, falls auf möglichst wenige Stellgrößen gleichzeitig eingewirkt werden soll. Ein mögliches Szenario für ℓ_1 -optimale schlanke Handlungsempfehlungen ist ein beschränktes Zeitfenster, in dem nur eine begrenzte Zahl von Stellgrößen vom Bediener manuell angepasst werden kann. Darüber hinaus kann die Nachvollziehbarkeit der Prozesshandlung durch den Bediener bewahrt werden, wenn dieser die Auswirkung des vom System vorgeschlagenen Eingriffs weiterhin abschätzen soll. Die Folgen der Einführung der Norm für die für die Konvergenz der datengetriebenen Optimierung müssen einzelfallspezifisch untersucht und bewertet werden.

Sollte im Vorhinein keine Normierung der Prozessparameter vorgenommen worden sein, sind die Elemente des Differenzvektors von $\mathbf{x} = [x_1, \dots, x_n]$ und $\mathbf{x}_0 = [x_{0,1}, \dots, x_{0,n}]$ mit Skalierungsfaktoren ϑ_i gemäß

$$\Xi(\mathbf{x} - \mathbf{x}_0) = \sqrt[k]{\sum_{i=1}^n (\vartheta_i(x_i - x_{0,i}))^k} \quad (4.26)$$

zu gewichten, um einen gleichmäßigen Einfluss auf den Strafterm $\Xi(\mathbf{x} - \mathbf{x}_0)$ zu gewährleisten. Für den Fall $k = 0$ ist mit Problemen bei der Konvergenz zu rechnen, da die Differenz hinsichtlich der Pseudonorm keine konvexe Optimierung zulässt. Hinzu kommen Unstetigkeiten bei der Kostenfunktion. Um dem entgegenzuwirken, kann die direkte Nutzung der Pseudonorm zu

$$\Xi(\mathbf{x} - \mathbf{x}_0) = 1 - \frac{1}{1 + \mu \|\mathbf{x} - \mathbf{x}_0\|_1^2} \quad (4.27)$$

abgewandelt werden. Dies erwirkt eine Glättung des Strafterms bei Änderungen in \mathbf{x} und stellt sicher, dass sich die Werte nur zwischen 0 und 1 bewegen. μ ist dabei ein spezifisch anzupassender Skalierungsfaktor.

Die vorliegende Kostenfunktion liefert einen skalaren zur iterativen Bewertung der Zielgröße. Ihre Komponenten können anhand der Koeffizienten α_i gewichtet werden, um einzelne Kriterien zu priorisieren. Ein mehrkriterieller Ansatz würde statt des Optimums, das die Kostenfunktion minimiert, eine Menge pareto-optimaler Lösungen bereitstellen. Dies erfordert die gleichzeitige Minimierung der in diesem Abschnitt vorgestellten Teilkomponenten der Kostenfunktion. Nach Ermittlung der Paretofront wäre die Gewichtung der Faktoren durch den Nutzer selbst vorzunehmen.

4.4.3 Bereitstellung der Handlungsempfehlungen

Es wird im Folgenden davon ausgegangen, dass die Optimierung und die Generierung der Handlungsempfehlung vor jedem Prozessdurchlauf durchgeführt werden. So erfolgt

die Berechnung für jedes Bauteil oder jede Charge, die durch eine Prozessvariante beschrieben werden kann, einzeln. Die Voraussetzung hierfür ist ein ausreichend großes Zeitfenster zwischen Prozessstart und -ende. Alternativ können die Parameter einmalig eingestellt und nur dann verändert werden, wenn aufgrund der Prognose eine hinreichend starke Abweichung vom vorgegebenen Prozessergebnis zu erwarten ist. Die aus der datengetriebenen Optimierung hervorgehenden Handlungsempfehlungen stellen die zweite Stufe der Interaktion der Optimierungskette zwischen Maschinenbedienern und Prozessingenieuren dar. Es gilt, eine Auswahl möglicher Prozesseingriffe für jede Zielgruppe anforderungskonform bereitzustellen. Dabei wird zwischen der manuellen Empfehlungsverarbeitung zur Unterstützung und der automatischen Empfehlungsverarbeitung zur Entscheidungsautomation unterschieden (Gröger 2015, S. 173). Im Folgenden wird von einem dashboardbasierten Assistenzsystem für Maschinenbediener ausgegangen, das weiterhin die personenunabhängige Nachverfolgung im Leitstand zulässt. Eine autonome Steuerung der Prozesse wird nicht angestrebt und die installierten Systeme sind nicht invasiv. Das Assistenzsystem kann Informationen zur Produktqualität und zu den Sekundärkriterien Materialverfügbarkeit, Kosten der Rohstoffe und prozessübergreifendem CO₂-Ausstoß bereitstellen. Diese eignen sich vorwiegend für eine Darstellung im Leitstand und der nachgelagerten Analyse im MES.

Sekundärziele wurden in der Kostenfunktion zur Erstellung des Rankings bislang nicht betrachtet. Für die Auflistung möglicher Handlungsempfehlungen können sie im Nachhinein zur weiteren Bewertung herangezogen werden. Zur einfacheren mathematischen Handhabung werden sie in Form eines Vektors \mathbf{x}_c durch elementweise Multiplikation mit den optimalen Prozessvarianten zu einem skalaren Qualitätsindex

$$S_i = \mathbf{x}_c^T \mathbf{x}_{opt}^{(i)} \quad (4.28)$$

verarbeitet. Zur Bestimmung von \mathbf{x}_c unter Einbezug der Rohstoffverfügbarkeit kann über eine geeignete Schnittstelle auf das ERP-System zugegriffen werden. Kosten für Werkstoffe im Falle einer Gattierungsplanung lassen sich beispielsweise über eine API in Echtzeit abrufen und dann mit den Spaltennamen der Prozessdaten abgleichen. Informationen über den CO₂-Ausstoß können, sofern sie sich mittels der Prozesseinstellung aufgrund komplexer Wechselwirkungen adäquat beschreiben lassen, im Nachgang zur Bewertung dienen, eignen sich aber nur bedingt zur Online-Steuerung. Die Prozessvarianten, die hinsichtlich eines Qualitätskriteriums S_i ein Optimum darstellen, kann mit einem entsprechenden Kommentar versehen werden. Da es sich bei diesen Aspekten zum Zeitpunkt der Erstellung der Arbeit noch um aktuelle Forschungsgegenstände handelt, werden nachfolgend nur noch die Produktqualität und die Komplexität der Handlungsempfehlung als Bewertungskriterien genutzt.

Das Ranking der q Prozessvarianten erfolgt zunächst nach dem zugehörigen Wert der Kostenfunktion. Da durch die mathematische Optimierung erfahrungsgemäß zahlreiche Lösungen entstehen, die sich nur geringfügig unterscheiden, muss ein Minimum beim euklidischen Abstand der Varianten

$$\left\| \mathbf{x}_{opt}^{(i)} - \mathbf{x}_{opt}^{(j)} \right\|_2 > \varepsilon \quad (4.29)$$

für $i, j = 1, \dots, q$ zueinander gewährleistet werden. Des Weiteren muss eine Mindestanzahl an Empfehlungen bereitstehen, die dem üblichen Umfang der Prozessveränderungen des

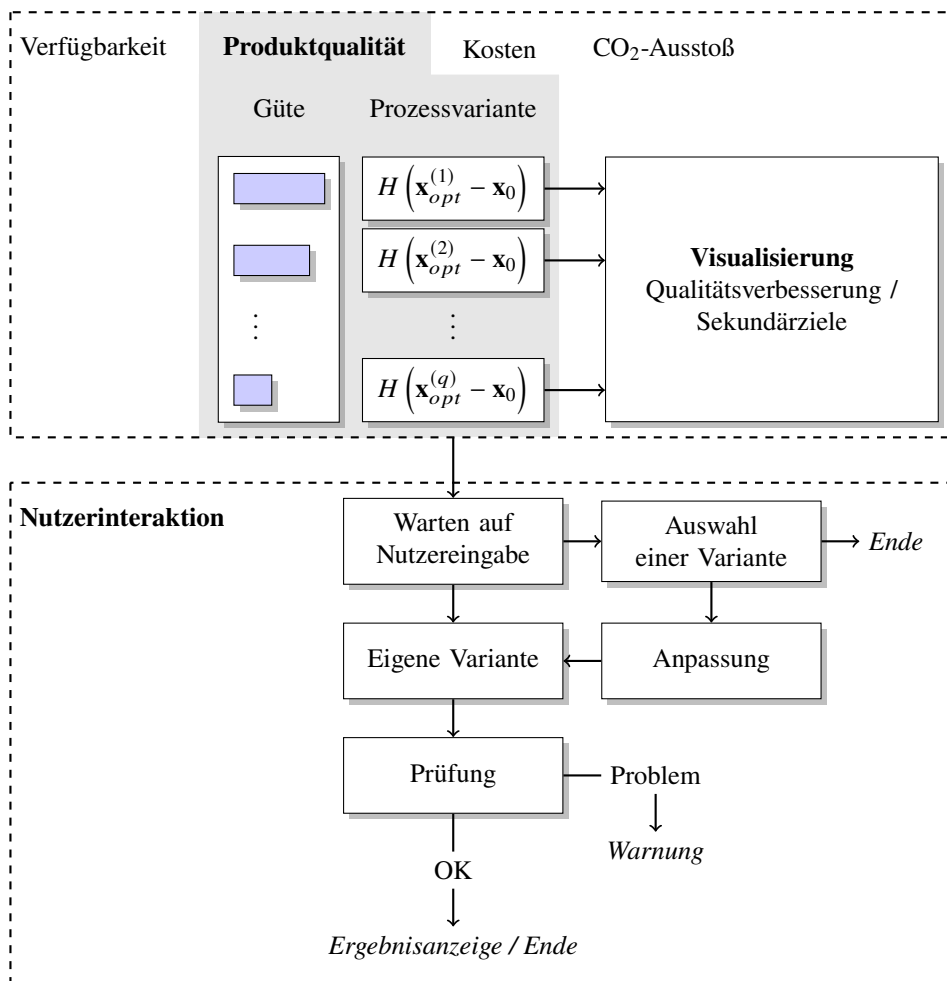


Abbildung 4.10: Struktur eines Assistenzsystems, bestehend aus einem Ranking optimaler Handlungsempfehlungen und dem Ablauf der Nutzerinteraktion. Die Kriterien für das Ranking sowie Nebenbedingungen können ausgewählt werden. Eingaben und Rückmeldungen orientieren sich formell an der DIN EN ISO 9241-110 (2020).

Nutzers entspricht. Darüber hinaus sollte ein Hinweis mit Begründung erfolgen, wenn eine Qualitätsvorgabe nicht mehr erreicht werden kann.

Abbildung 4.10 zeigt den schematischen Aufbau eines dashboardbasierten Assistenzsystems sowie die mögliche Nutzerinteraktion. Als Hauptbewertungskriterium ist die Produktqualität ausgewählt. Die sich aus Ist-Einstellung \mathbf{x}_0 und synthetisierter optimaler Prozessvariante $\mathbf{x}_{opt}^{(i)}$ ergebenden Handlungsempfehlungen $H(\mathbf{x}_{opt}^{(i)} - \mathbf{x}_0)$ werden vertikal aufgelistet. Sie können als konkrete Maßnahmen textuell, z.B. *Drehzahl 3: 50 → 60*, oder in Form von Balkendiagrammen mit den relativen Veränderungen vorgeschlagen werden. H ist dabei keine mathematische Operation, sondern kann als Visualisierung der Anweisung verstanden werden. Daneben befindet sich ein Indikator der Güte hinsichtlich der Kostenfunktion, der numerisch oder rein visuell dargestellt werden kann. In einer gemeinsamen oder wahlweise variantenspezifischen Visualisierung lassen sich die vorgeschlagenen Maßnahmen und Zielvorgaben grafisch darstellen.

Die Nutzerinteraktion wird an die Anforderungen zur Dialoggestaltung im Rahmen der DIN EN ISO 9241-110 (2020) ausgerichtet. Zur Aufgabenangemessenheit wird die Anzeige auf die soeben beschriebenen Informationen beschränkt. Der Nutzer hat im Anschluss die Möglichkeit, eine Variante direkt auszuwählen und zu bestätigen. Eine Anpassung lässt sich, falls erforderlich, zu diesem Zeitpunkt noch vornehmen und wird umgehend visualisiert. Hierfür ist eine Gesamtprognose aller Zielgrößen erforderlich. Lange Berechnungszeiten auf einem Server oder hohe Latenzen bei der Kommunikation mit dem Endgerät können gemäß der Erwartungskonformität nicht in Kauf genommen werden. Als Alternative besteht die Möglichkeit zur Eingabe einer eigenen Prozesseinstellung. In beiden Fällen wird eine Prüfung durch die Nebenbedingungen und die Verteilungskriterien der Kostenfunktion durchgeführt. Sollten diesbezüglich Probleme entstehen, erfolgt eine Warnung, jedoch keine Blockade oder Meldung an den Leitstand.

4.5 Implementierungsaspekte

In diesem Abschnitt werden die Bedingungen für eine erfolgreiche prototypische Implementierung der Optimierungskette in den laufenden Betrieb beschrieben. Zunächst wird die Herangehensweise für eine Bewertung der Module sowohl einzeln als auch im Zusammenschluss aufgezeigt. Als nächstes werden die Teilaufgaben den Phasen des CRISP-DM zugeordnet und eine schrittweise Integration in die Produktion in Form eines mehrstufigen Assistenzsystems erläutert.

4.5.1 Modulübergreifende Bewertung

Eine abschließende Bewertung der gesamten Analyse-Pipeline, also der Hintereinanderschaltung von Datenvorverarbeitung, Modellierung und Optimierung, entscheidet darüber, ob sie in der Produktion eingesetzt werden kann. Die Performanceeinordnung der Module kann einzeln mit den jeweiligen Gütekriterien erfolgen, für die Wahl der optimalen Pipeline müssen jedoch im Regelfall Kompromisse aus Gründen der Umsetzbarkeit eingegangen werden. Strukturell orientiert sich die modulübergreifende Bewertung am intelligenten Analysemanager der Softwareumgebung *EIDominer* (Saleem et al. 2015), bei dem nur Verfahren ausgewählt werden, die eine Mindestgüte erreichen. Sie wird in diesem Rahmen um die gradientenfreie Optimierung erweitert.

Abbildung 4.11 zeigt die modulübergreifende Bewertung schematisch. Die Wirkung der Rausch- und Störgrößenkompensation lässt sich bei einer Regression nur dann effektiv, beispielsweise durch den RMSE, quantifizieren, wenn die echten Bezugsgrößen im Rahmen eines Experiments vorliegen und nicht zu stark verändert wurden (Halim et al. 2012). So werden für stark geglättete Daten häufig hohe Modellgenauigkeiten erreicht, die Erkenntnisse sind in der Praxis jedoch meist unbrauchbar. Für Klassifikationsprobleme kann die Genauigkeit der auf den verarbeiteten Daten basierenden Vorhersagemodelle herangezogen werden, wenn die Klassenvariable selbst dadurch nicht verändert wird. Ein Beispiel hierfür ist die Klassifikation von Röntgenaufnahmen zur Erkennung von Lunkern (Li et al. 2006). In diesem Fall entstehen unabhängige Variablen und Labels erst durch die Featureextraktion im Rahmen der Datenaufbereitung. Eine schlechte Klassifikationsperformance kann somit auch auf eine falsche Vorverarbeitung der Daten zurückgeführt werden und darf nicht als alleiniges Bewertungskriterium dienen. Wie später in Kapitel 5 verdeutlicht,

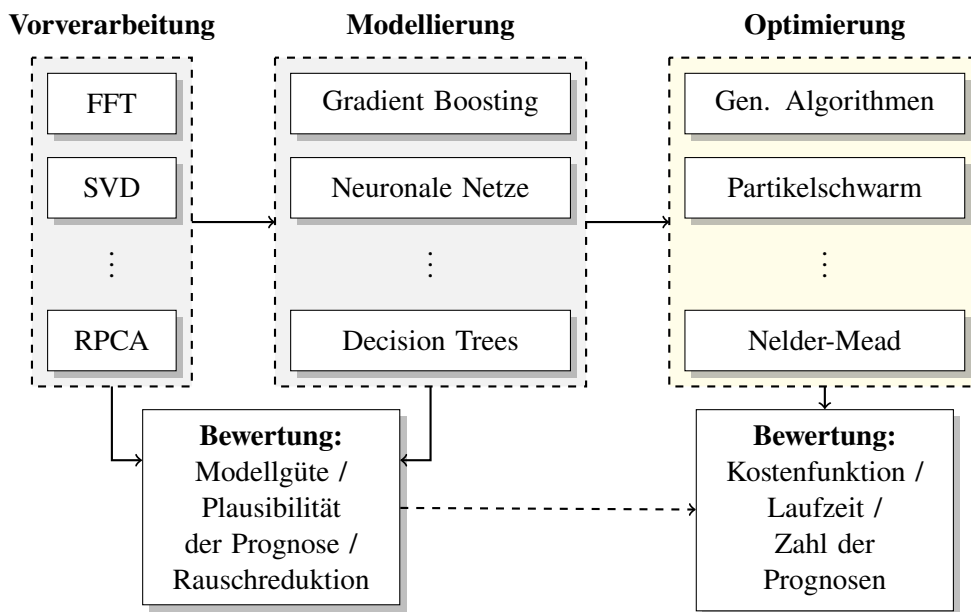


Abbildung 4.11: Modulare Bewertung der Optimierungskette in Anlehnung an den intelligenten Analysemanager (Saleem et al. 2015). Datenvorverarbeitung und Modellierung können gemeinsam hinsichtlich der Modellgenauigkeit bewertet werden. Bei der Optimierung werden neben der Kostenfunktion noch die Rechenlaufzeit und die Messraumverdichtung berücksichtigt.

können durch einen falsch gewählten Threshold wichtige Merkmale in den Daten entfernt werden. Eine objektive Bewertung ist daher letztlich nur durch Expertenwissen möglich. Die Algorithmenauswahl bei der Optimierung ist so zu treffen, dass die beste Methodenkombination die Vorhersage der Kenngrößen innerhalb der Kostenfunktion vornimmt. Für die Laufzeit ist die Berechnungsdauer einer Einzelprognose und die etwaige Vorverarbeitung der zugehörigen Prozessinstanz von Bedeutung. Darüber hinaus entscheidet die Messraumverdichtung, wie viele Prognosen für einen aussagekräftigen Qualitätskennwert notwendig sind.

4.5.2 Bereitstellung der Module

Für die Nutzung der Optimierungskette innerhalb der unternehmensspezifischen Dateninfrastruktur werden die Module den Phasen des CRISP-DM nach Chapman et al. (2000) zugeordnet. Darüber hinaus sind Aspekte des Projektmanagements und die Anforderungsspezifikation von großer Bedeutung. Die Implementierung und Integration der datengetriebenen Optimierungslösung in den Prozess können schrittweise erfolgen und je nach genutzten Modulen in unterschiedlicher Ausprägung die Produktion unterstützen. Das Datenverständnis und die Prozessstrukturierung ist dem Data und Business Understanding zuzuordnen. Es beschreibt die Vorprojektphase und umfasst die Festlegung der Optimierungsziele sowie der relevanten Teilprozesse. In Zusammenarbeit zwischen Prozessexperten und Datenanalysten erfolgt die Datenkonsolidierung, eine Spezifikation der Formate und die Zugänglichkeit für alle Beteiligten. Die in dieser Phase festgelegten

Rahmenbedingungen gelten in der Regel für den restlichen Projektablauf.

Obwohl im CRISP-DM lediglich als Vorstufe der Modellierung genutzt, können die Ergebnisse des Datenvorverarbeitungsmoduls (vgl. Data Preparation) bereits im Rahmen eines ersten Assistenzsystems als Dashboard-Lösung den Prozess unterstützen. Eine anforderungsgerechte Visualisierung prozessrelevanter Kennwerte stellt den ersten Berührungspunkt zwischen Feldebene und Analyseergebnissen dar. Mit robusten Kompensationsverfahren wie der RPCA lassen sich echtzeitnah einerseits Störgrößen herausfiltern, andererseits können mit der erlernten Verteilung Messdaten bei Sensorausfällen rekonstruiert werden. Die Verkleinerung des Datenraumes mit SVD erlaubt eine Komplexitätsreduktion und eine einfachere Inline-Bewertung der Performance-Indikatoren des Prozesses.

Die Ergebnisse der datengetriebenen Modellierung (vgl. Modeling) eignen sich vorwiegend für die Offline-Analyse und als Diskussionsgrundlage für die Beschreibung (vgl. Evaluation) der Modelle der Wirkzusammenhänge im Prozess. Zwar können Prognosen in die Dashboard-Visualisierung einfließen, sie sollten jedoch direkt mit den Handlungsempfehlungen kombiniert werden. Mit Hilfe der Bewertung durch den Supervisor kann die beste Analyse-Pipeline, einschließlich der Offline-Optimierung, ausgewählt werden. Zu diesem Zeitpunkt müssen Analysten und Prozessexperten gemeinsam entscheiden, ob die Güte und Aussagekraft der Modelle ausreichend sind.

Die Bereitstellung der Online-Optimierung und der daraus folgenden Handlungsempfehlungen (vgl. Deployment) erfolgt als zweite Stufe des Assistenzsystems. Zur mittel- und langfristigen Auswertung der im Rahmen der Optimierungskette erarbeiteten Verfahren sind Benchmarking-Methoden anzuwenden, die einen Vorher-Nachher-Vergleich der Performance-Indikatoren des Prozesses zulassen. Im Mittelpunkt steht auch die über einen längeren Zeitraum aufgenommene Differenz zwischen vorhergesagten und tatsächlich vorliegenden Kennwerten als Bewertungskriterium für die Modellgüte.

Datensicherheit

Produktionsdaten kann inzwischen ein wirtschaftlicher Wert beigemessen werden. Sie zu schützen dient einerseits dem Erhalt des damit verknüpften Prozesswissens, andererseits können sie in den falschen Händen enormen Schaden anrichten. Zu ihrer Abschirmung werden daher weitreichende Maßnahmen unternommen, eine vollkommene Sicherheit besteht jedoch nicht. Die in diesem Kapitel entwickelten Verfahren bieten zum Teil tiefgehende Einblicke in die Funktionsweise eines Fertigungsprozesses und können unter Umständen seine Schwächen aufdecken. Auch wenn die generierten Informationen nicht unmittelbar auf eine andere Produktion übertragen werden können, wären Wettbewerber in der Lage, ihr eigenes Portfolio entsprechend anzupassen. Weiterhin ist nicht auszuschließen, dass durch indirekte Schlussfolgerungen ein Personenbezug hergestellt wird.

Die im Rahmen dieser Arbeit entwickelte Methodik lässt sich ausnahmslos unter der Nutzung von Open-Source-Bibliotheken umsetzen. Bei der Ausführung im laufenden Betrieb kann die bestehende IT-Sicherheitsinfrastruktur die Datenflüsse abschirmen. Ein zusätzliches Sicherheitsrisiko bestünde somit nicht. Darüber hinaus kann der Personenkreis, dem Zugriff auf die Prozessdaten und -analysen gewährt wird, eingeschränkt werden.

Zusammenfassung

In diesem Kapitel wurde das Konzept der datengetriebenen Optimierungskette entwickelt. Es wurden Anforderungen für die Datenvorverarbeitung, die Modellierung und die Optimierung sowie für das Gesamtkonzept formuliert. Der Entwicklungsansatz arbeitet die Relevanz der drei Module und ihre Verknüpfung zueinander heraus. Im Rahmen des Datenvorverarbeitungsmoduls wird zunächst die prozessspezifische Aufbereitung aufgegriffen. Es zeigt Möglichkeiten auf, Prozessdaten aus heterogenen Quellen für eine Analyse nutzbar zu machen. Anschließend wird die thresholdbasierte Verdichtung vorgestellt und ihr Potenzial für die Störgrößenkompensation sowie die Beschleunigung der Optimierung aufgezeigt. Das Modellierungsmodul greift auf bewährte Methoden zurück Prognose-Pipelines zu optimieren. Darüber hinaus werden Ansätze der prozessspezifischen empirischen Modellinterpretation unter besonderer Berücksichtigung mehrdimensionaler Zielgrößen erläutert. Als nächstes erfolgte die Konzipierung des Optimierungsmoduls. In die Optimierungsschleife fließen Komponenten der virtuellen Versuchsplanung mit analytischen Modellen ein. Die Kostenfunktion wurde für mehrdimensionale Zielgrößen ausgerichtet und enthält Bestandteile, die neben dem Prozessergebnis auch die Charakteristik der generierten Variante selbst berücksichtigt. Zum Abschluss wurde eine schrittweise Bereitstellung der Module in Anlehnung an den CRISP-DM illustriert.

Kapitel 5

Anwendung der Entwicklungsmethode

In diesem Kapitel wird die erarbeitete Entwicklungsmethode an einem konkreten Anwendungsfall zur datengetriebenen Optimierung verifiziert. Hierzu werden Prozessdaten einer Schleudergießerei genutzt. Mit ihnen werden Wandstärkenprofile duktiler Gusseisenrohre vorhergesagt. Zunächst werden die für das Fallbeispiel relevanten gießtechnischen Grundlagen zusammengefasst (5.1). Es folgt eine Beschreibung des zugrundeliegenden Projektes und der dazugehörigen Datenbasis. Im Rahmen einer korrelationsbasierten Dimensionsreduktion werden Anwendungsfälle für die Störgrößenkompensation und die Messraumverdichtung aufgeführt (5.2). Die datengetriebene Modellierung mit überwachtem maschinellem Lernen (5.3) einschließlich der Analyse der Wirkzusammenhänge (5.4) stellt die empirische Prozessbeschreibung dar. Abschließend erfolgt die Entwicklung einer methodischen Grundlage für die Inline-Optimierung der Wandstärkenverteilungen auf Basis datengetriebener Handlungsempfehlungen (5.5).

5.1 Grundlagen der Fallstudie

In diesem Abschnitt werden die für das Projekt relevanten gießtechnischen Grundlagen zusammengefasst. Es werden die Klassifikation von Form- und Gießverfahren, die wichtigsten Eisengusswerkstoffe und ausgewählte Beispiele für Gießverfahren beschrieben.

5.1.1 Gießtechnische Grundlagen

Gießen bezeichnet laut DIN 8580 (2003) das Urformen aus dem flüssigen, breiigen oder pastenförmigen Zustand. Bei Gießverfahren wird zunächst zwischen der Herstellung von Halbzeugen zur Weitererarbeitung im Gegensatz zu Formteilen unterschieden. Der nachfolgende Abschnitt beschränkt sich auf Formteile. Für die Einteilung der Verfahren ist die Art der Modelle, also der Positive der Gussteile, und der Formen, also der Negative, relevant. Es wird zwischen drei Verfahrensgruppen unterschieden (Klocke 2018, S. 2):

- Verfahren mit verlorenen Formen und Dauermodellen
- Verfahren mit verlorenen Formen und verlorenen Modellen
- Verfahren mit Dauerformen

Lässt sich das Modell mehrmals verwenden, wird es Dauermodell genannt, wird es bei der Verwendung zerstört, handelt es sich um ein verlorenes Modell. Analog können die

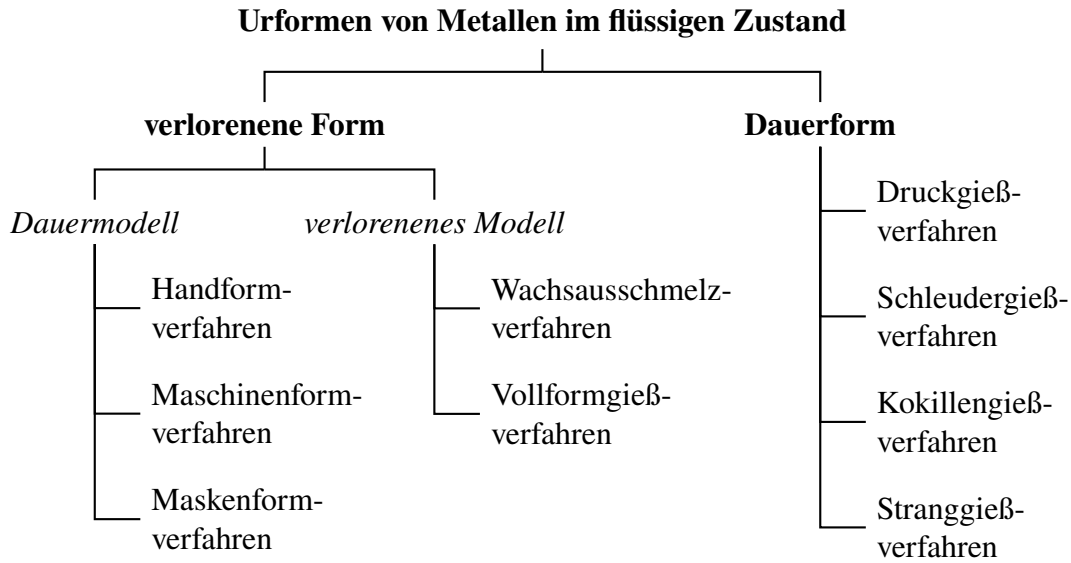


Abbildung 5.1: Übersicht über wichtige Gießverfahren aus Riße (2012, S. 14), eigene Darstellung.

Formen in Dauerformen und verlorene Formen unterteilt werden. Alle in der Praxis angewandten Gießverfahren lassen sich in diese Kategorien einordnen, wobei manche Kombinationen physikalisch und verfahrenstechnisch nicht sinnvoll sind. Abbildung 5.1 zeigt eine Übersicht der am häufigsten genutzten Gießverfahren.

Gusswerkstoffe

Bei Gusswerkstoffen wird eine grundsätzliche Unterscheidung zwischen Eisen- und Nicht-eisen-Werkstoffen getroffen. Die Produktionszahlen beider Werkstoffgruppen sind vor allem durch die Nutzung im Automobilbau geprägt. Bei Eisenlegierungen werden zumeist solche mit einem hohem Kohlenstoffgehalt von 2,4 bis 4,3 % vergossen. Dieser wird durch die Zugabe von 2 bis 3 % Silizium bewerkstelligt. Der Einsatz von Silizium verschiebt den eutektischen Punkt im Eisen-Kohlenstoff-Diagramm nach links, sodass dieser bei niedrigerem Kohlenstoffgehalt erreicht wird. Dies senkt die Schmelztemperatur, was zu einer geringeren Belastung des Formstoffes, einem guten Fließvermögen und einer höheren Maßgenauigkeit führt. *Gusseisen mit Lamellengrafit* (GJL) ist der älteste und hinsichtlich seiner jährlichen Tonnage am meisten verwendete Eisengusswerkstoff. Er wird durch lamellare, blättchenförmige Ausscheidungen des Grafiten charakterisiert, die eine innere Kerbwirkung erzeugen und die Eisengrundmasse unterbrechen. Die entstehenden Gussteile haben eine geringere Zugfestigkeit, die für die meisten Anwendungen jedoch ausreicht. *Gusseisen mit Kugelgrafit* (GJS), nach geschützter Bezeichnung auch *Sphäroguss* genannt, entsteht durch die Behandlung der Schmelze mit Magnesium. Hierdurch bilden sich kugelförmige Grafitausscheidungen, die die innere Kerbwirkung und die Volumenschrumpfung verringern und Festigkeits- und Dehnwerte erhöhen. Das Resultat ist ein duktiler Gusswerkstoff, der beispielsweise im Schleuderguss eingesetzt wird und sich für Bauteile eignet, die einer hohen dynamischen Belastung ausgesetzt sind (Fritz 2018, S. 30–37). *Gusseisen mit Vermikulargraft* (GJV) liegt hinsichtlich seiner mechanischen Eigenschaften zwischen GJL und GJS. Es zeichnet sich durch das in wurmartiger Form

verwendete entartete Graphit aus (Bührig-Polaczek 2014, S. 58).

Beispiele für Gießverfahren

Beim *Druckguss* werden vorwiegend Nichteisen-Werkstoffe wie Aluminium vergossen. Die Schmelze mit 150 bis 1200 bar unter hoher Geschwindigkeit in eine Dauerform gespritzt, die meist aus Stahl besteht. Hierdurch gefertigte Gussteile sind maßgenau und können schnell und in hoher Stückzahl produziert werden. Neben dem Abtrennen von Eingusssystemen ist in den meisten Fällen keine Nachbearbeitung erforderlich. Beim Warmkammverfahren bilden Warmhalteofen und die Füllkammer eine Einheit, während sie beim Kaltkammverfahren getrennt sind. Der Druckguss eignet sich zur Anwendung in der Massenfertigung; je nach mittlerer Wanddicke und Legierung der Gussteile beträgt die Zykluszeit zwischen 50 und 120 Millisekunden (Bührig-Polaczek 2014, S. 297–311).

Der *Kokillenguss* ist ein weiteres Verfahren mit einer Dauerform. Sie besteht zumeist aus Stahl oder Gusseisen. Es werden vorwiegend Gussteile aus Aluminium-, Magnesiumumlegierungen oder Messing hergestellt, da deren Schmelzpunkte deutlich unter dem des Kokillenmaterials liegen. Verwendete Kerne aus Sand werden bei der Entnahme zerstört, während Metallkerne so angeordnet werden müssen, dass sie nach der Erstarrung des Gussteils durch Drehen oder Verschieben entnommen werden können (Klocke 2018, S. 22).

Schleudergießen ist ein Verfahren zur Herstellung von Hohlkörpern mit zylindrischer Innenfläche wie Rohre oder ringähnliche Außenformen. Die Schmelze wird in eine sich um die Mittelachse rotierende Form gegossen. Durch die Zentrifugalkraft wird sie an die Forminnenwand gepresst, wo optional ein Profil eingebracht werden kann, und erstarrt dort. Je nach Anlagenaufbau lässt sich eine horizontale, vertikale oder geneigte Drehachse realisieren. Kokillen, die der Forminnengebung dienen, bestehen aus Metallwerkstoffen, können mit keramischen Materialien ausgekleidet und je nach Schmelztemperatur gekühlt werden. Optional werden, abhängig vom geforderten Innenprofil, Sandkerne eingesetzt (Bührig-Polaczek 2014, S. 329–333). Die mittlere Wandstärke wird durch die zugegebene Schmelzemenge bestimmt. Auf den Prozess lässt sich über die Gießgeschwindigkeit, die Drehzahl der Kokille und das Abmessen oder Abwiegen der Metallmenge Einfluss nehmen. Der Zentrifugalkraft wird eine reinigende Wirkung zugesprochen, da Verunreinigungen wie Schlacken, Lunker und Gasblasen an die Innenwand des Gussteils gedrückt werden. Weitere Vorteile sind die gute Formfüllung, nicht erforderliche Anschnittsysteme und eine hohe Festigkeit aufgrund des dichteren Gefüges. Nachteilig sind der begrenzte Wandstärkenbereich, eine notwendige Mindeststückzahl sowie die verringerte Flexibilität der Fertigungsabläufe (Riße 2012, S. 41–42).

5.1.2 Auslegung der Fallstudie

Ein entscheidendes Qualitätsmerkmal von durch Schleudergießen gefertigten Rohren ist ein möglichst homogener Verlauf der Wandstärke parallel zur Rotationsachse. Eingriffe in den Prozessablauf und die damit verbundene Wanddickenausprägung obliegen nach wie vor der Intuition erfahrener Mitarbeiter. Eine Kontrolle erfolgt für gewöhnlich erst nach der Fertigstellung. Prozessbedingte Störgrößen, die sich auf die Messung auswirken, sind unter anderem die Ovalität der Kokille und die Zugabe von Kokillenspuder sowie raue

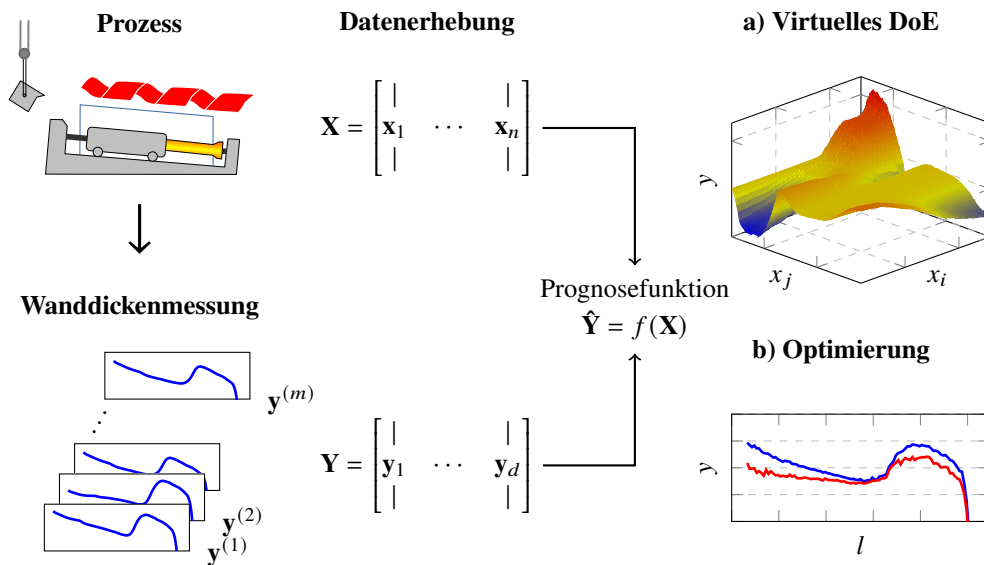


Abbildung 5.2: Projektziele der datengetriebenen Optimierung im horizontalen Schmelzerguss. Darstellung in Anlehnung an Kutz et al. (2016, S. 6).

Gussteiloberflächen und innere Ungenzen. Diese erschweren die Erfassung der Produktqualität und damit auch ihre Zuordnung zu Prozessparametern. Den Wandstärkenverlauf mit Hilfe modellbasierter Simulation zu optimieren, stellt aufgrund der hohen Dimensionalität und der hochgradig nichtlinearen Zusammenhänge und Interaktionen mit Millionen von Freiheitsgraden eine enorme Herausforderung dar; eine Lösung des Optimierungsproblems in geschlossener Form lässt sich wegen unbekannter analytischer Systemkenntnisse nicht ermitteln. Verfahren zur Simulation des Prozesses stehen bis dato nicht in kommerzieller Form zur Verfügung. Die konventionelle Prozessoptimierung zur Kostensenkung und Qualitätssteigerung stößt in diesem Fall an ihre Grenzen.

Insbesondere der Wanddickenverlauf am Spitzende des Rohrs ist nach wie vor nur schwer beherrschbar. Eine analytische Simulation des Verfahrens mit MAGMASOFT® ist aufgrund der sich drehenden Kokille nur unter großem Entwicklungsaufwand umsetzbar. Es lassen sich keine Rückschlüsse über die vorliegenden Wirkzusammenhänge ziehen, da das Wanddickenprofil darüber hinaus an dieser Stelle von zu vielen Parametern abhängt, die in der Simulation nicht abgebildet werden können. Abbildung 5.2 zeigt die Zielsetzung des Projektes. Es sollen Muster zwischen Prozessparametern und Wanddicken durch maschinelles Lernen sichtbar gemacht werden. Das Ziel ist ein grundlegendes Verständnis der Verfahrensabläufe beim Gießen, insbesondere am Spitzende. Zu prüfen ist die Anwendbarkeit der datengetriebenen Prozessmodelle zur Prognose und Identifikation der Wirkzusammenhänge. Schließlich sollen sie zu einer Optimierung des Wandstärkenverlaufs durch eine Minimierung der Wandstärkenunterschiede beitragen.

5.1.3 Beschreibung des Prozesses

Der Gesamtprozess der Rohrfertigung ist schematisch in Abbildung 5.3 aufgezeigt. Einsatzstoffe wie Schrotte sowie Zuschlagstoffe werden in einem Kupolofen zu flüssigem Gusseisen erschmolzen. Zur Warmhaltung der Schmelze dient ein Induktionsrinnenofen.

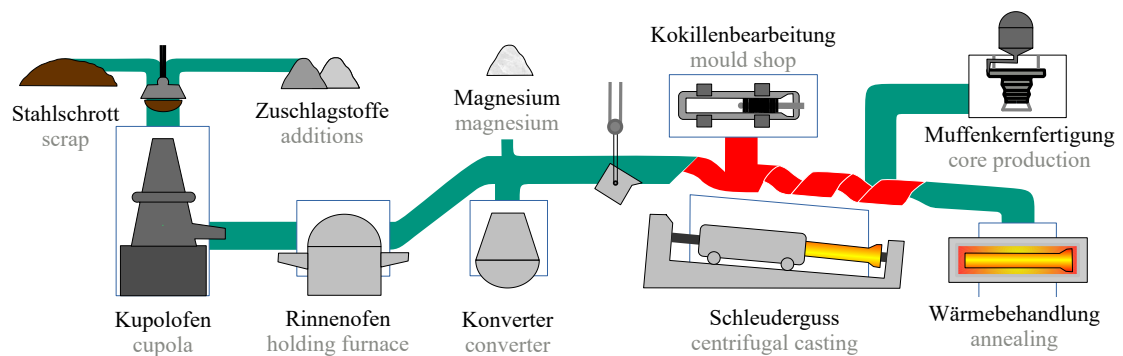


Abbildung 5.3: Übersicht des Prozesses. Quelle: Tiroler Rohre GmbH.

Zur Herstellung von Gusseisen mit Kugelgraphit wird die Vorlegierung im Konverter mit Magnesium behandelt. Der finale Werkstoff ist EN-GJS-450-10 nach DIN EN 1563 (2019). Eine Pfanne und ein Kipper leiten das Flüssigeisen zunächst in eine Rinne und dann in die drehbare Kokille. Durch die Rotations- und Translationsbewegung erfolgt der Aufbau des Rohrs spiralförmig. Sandkerne bilden das Muffenprofil und die Innenkontur des Rohrs und dichten die Kokille während des Gießvorgangs nach außen ab. Aufgrund der hohen Abkühlgeschwindigkeit entsteht ein sprödes Eisengefüge. Eine abschließende Wärmebehandlung „glüht“ den Werkstoff wieder weich und baut innere Spannungen ab.

Abbildung 5.4 zeigt die beispielhafte Fertigung eines Rohres im Detail. Der Kipper wird für jedes gegossene Rohr neu aus der Pfanne befüllt und abgeschlackt. Nach dem automatisierten Einsetzen des Kernes wird der Schleudergießprozess vom Mitarbeiter initialisiert. Durch die Neigung des Kippers wird das Flüssigeisen über Bogen und Rinne in die fahrbar und drehbar gelagerte Kokille geleitet. Während die Maschine auf ihrer Anfangsposition steht, befüllt sie die Muffe mit Schmelze. Sobald der Vorgang abgeschlossen ist, fährt die Maschine los und legt dabei unterschiedliche Fahrbereiche zurück. Währenddessen liefert der Kipper durch eine kontinuierliche Bewegung gleichmäßig Flüssigeisen nach. Die Erstarrung erfolgt indes an der durch konstante Kühlwasserzufuhr gekühlten Kokillenwand. Noch während der Fahrt wird der Kipper wieder zurückgeneigt, sodass die Schmelzezufuhr gestoppt wird. Die Maschine legt dann die Reststrecke zurück und das in der Rinne verbleibende Flüssigeisen wird aufgefangen. In der Zeit, in der das Rohr abkühlt, werden Rinne und Bogen gewechselt. Nachdem die Maschine auf ihre Startposition zurückgekehrt ist, wird das Rohr entnommen und der Prozess kann erneut starten.

Die Wandstärke der Rohre entlang der x-Achse wird im heißen Zustand aufgenommen. Im Folgenden werden gängige Verfahren zur zerstörungsfreien Wanddickenprüfung vorgestellt. Die gepulste Wirbelstromprüfung (engl. *Pulsed Eddy Current Testing*, PECT) kann unter Verwendung zylindrischer Spulen neben der Wanddicke auch die Leitfähigkeit und die Permeabilität von Rohrkörpern messen (Chen et al. 2020). Dies funktioniert auch durch Kunststoffisolierungen hindurch und wird insbesondere für die Korrosionsprüfung trinkwasserführender Gusseisenrohre genutzt (Nguyen und Miro 2020). Darüber hinaus lassen sich Wandstärken über die *Laufzeitermittlung von Ultraschall* messen, da die Fortschrittggeschwindigkeit der Wellen für ein Material als konstant gelten kann (Lebowitz und Brown 1993). Der Zusammenhang zwischen Messung und tatsächlichen Werten wird

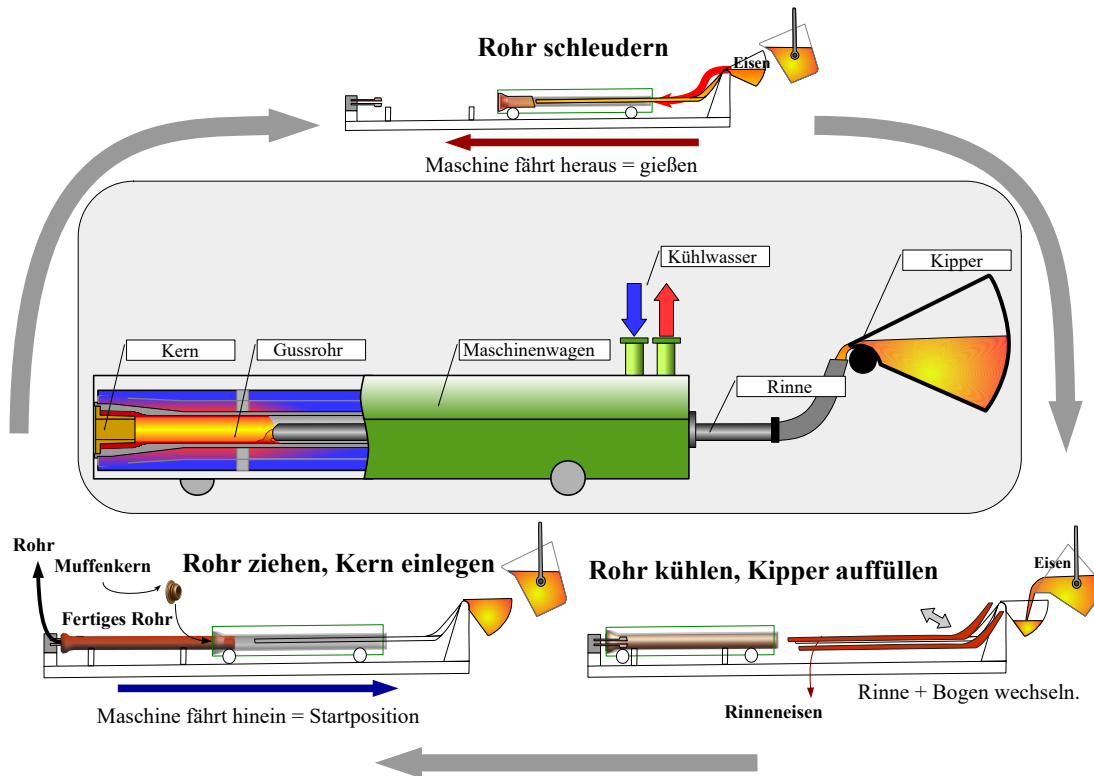


Abbildung 5.4: Verfahrensablauf beim Schleudergießen. Quelle: Tiroler Rohre GmbH.

empirisch ermittelt. Bekannte Störgrößen sind Rost, Materialschwund durch Korrosion, Schmutzablagerungen, Oberflächenrauigkeit und aufgetragener Lack (Adamowski et al. 2013). Bei schleudergegossenen Rohren kommen hierzu eine mit der Zeit wachsende Ovalität der Kokille, Oxid- und Kernsandeinschlüsse sowie Kokillenpulver. Die genauen Anforderungen und technischen Rahmenbedingungen, unter denen die Wanddickenprüfung duktiler Gusseisenrohre erfolgt, legt die DIN EN 545 (2011) fest.

5.1.4 Beschreibung der Datenbasis

Für die Analysen in diesem Kapitel wurden insgesamt 22.729 Rohre und Pfähle betrachtet, die im Zeitraum von Februar 2020 bis September 2021 gegossen wurden. Eine Klassifizierung erfolgt anhand des Verhältnisses von Soll-Wanddicke und Außendurchmesser¹. Die folgenden Ausführungen beziehen sich nur noch auf Rohre. Es werden je nach Kundenanforderung unterschiedliche Produktklassen gefertigt, die jeweils durch einen Nenndurchmesser und eine Solldicke bzw. Mindestwandstärke gekennzeichnet sind. Die Konzepte zur Datenvorverarbeitung und Störgrößenkompensation wurden mit den bereinigten Gesamtdaten durchgeführt, während für die Modellierung und die Optimierung eine Rohrklasse isoliert betrachtet wurde. Dies liegt darin begründet, dass Schwankungen der Wandstärke auf die Prozessparameter und nicht auf die Produktgruppe zurückgeführt werden sollen. Für die Analysen liegen nach der Filterung und Bereinigung, auf die im nächsten Abschnitt näher eingegangen wird, Daten über 3.490 Rohre vor. Die hierfür zu-

¹https://trm.at/fileadmin/user_upload/VP_Pfahlfolder_sRGB_DE_FINAL.pdf

grundlegenden Prozessdaten umfassen den Konverter, die Magnesiumbehandlung und den Schleudergießprozess selbst. Für sie ist eine einzelteilbezogene Datenzuordnung gewährleistet. Informationen über Schrottzugaben, Sekundärrohstoffe und den Kupolofen liegen dem Unternehmen vor. Sie spielen bei der Wandstärkenausbildung jedoch nur eine untergeordnete Rolle und lassen sich auch nicht anforderungsgerecht im Vorfeld steuern.

Den Fertigungsprozess beschreiben insgesamt 192 Kenngrößen beschrieben. Darin sind 100 Wandstärkenmessungen enthalten, die entlang der Längsachse aufgenommen werden. Die verbleibenden 92 Prozessparameter umfassen neben der Gießzeit und der Maschinenummer zur eindeutigen Kennzeichnung unter anderem Folgendes:

- Sekundärrohstoffzugaben (z.B. Kokillenpulver und Impfmittel)
- Daten der Schmelze (z.B. Flüssigeisentemperatur)
- Verfahrensparameter (z.B. Drehzahlen, Fahrgeschwindigkeiten etc.)

Die Verknüpfung der Anlagen- und Schmelzedaten ist gewährleistet. Sie wird innerhalb einer Prozessdatenbank vorgenommen, auf die zu Analysezwecken zugegriffen werden kann.

5.2 Datenvorverarbeitung und Dimensionsreduktion

Die benachbarten Wandstärkenmessungen in den vorliegenden Daten sind – unabhängig von den Prozessdaten – paarweise miteinander stark korreliert. Dies legt einerseits die Möglichkeit nahe, anhand einer Dimensionsreduktion einen kleineren Unterraum der funktional abhängigen Variablen zu ermitteln, sodass weniger Messungen durchgeführt werden müssen. Andererseits lassen sich mit Hilfe der vorhandenen Korrelationen auch wichtige strukturelle Informationen extrahieren, die genutzt werden können, um den Einfluss von Messrauschen und externen Störgrößen zu reduzieren.

Ein Missing-Data-Problem lag nicht vor. Die standardmäßige Datenvorverarbeitung sowohl für die Prozessparameter als auch die Wandstärken umfasste

- die Entfernung konstanter Spalten,
- die Entfernung von Zeilen, bei denen der Z-Score als Indikator für Ausreißer

$$z_i = \frac{x_i - \bar{x}_i}{\text{std}(\mathbf{x})} \quad (5.1)$$

einer Spalte mehr als drei beträgt,

- die Festlegung physikalisch plausibler Wertebereiche für einzelne Variablen und
- die Entfernung von Zeilen mittels der PCA-basierten Q-Statistik.

Weiterhin wurden zur Einhaltung der Kausalkette Variablen entfernt, die erst nach der Fertigstellung aufgenommen werden. Dies betrifft beispielsweise eine nachträgliche Untersuchung der Oberflächenqualität und das Gewicht des Rohres.

Im weiteren Abschnitt werden nun Verfahren beschrieben, die prozessspezifische Störgrößen und Sensorausfälle kompensieren, die keine stochastische Charakteristik aufweisen. Eine Nutzung der standardmäßigen Verfahren wäre hier nicht ohne weiteres möglich. Es werden hierzu mehrere Verfahren des unüberwachten maschinellen Lernens angewendet und miteinander verglichen. Schließlich werden die gewonnenen Erkenntnisse dazu

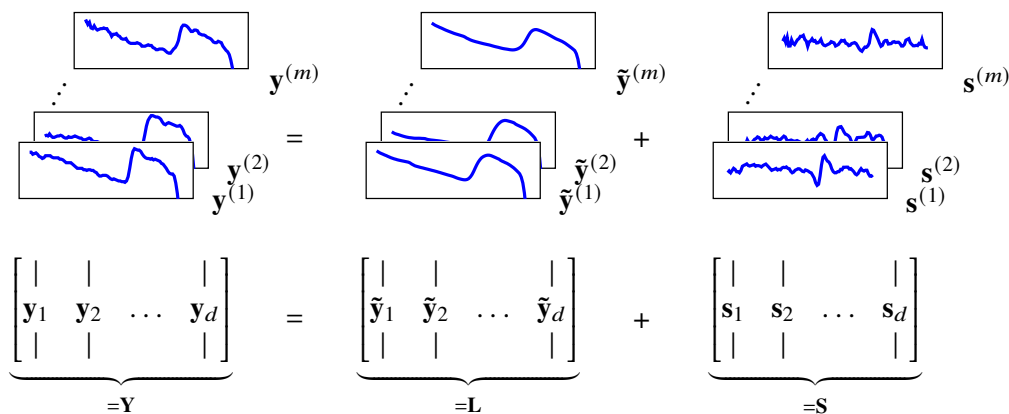


Abbildung 5.5: Prinzip der Störgrößenkompensation in Anlehnung an Scherl et al. (2020). Jedes Rohr wird als Zeile von Y dargestellt. Es kann in seine wahre Struktur L und das Rauschen S zerlegt werden.

eingesetzt, aus einer Untermenge der Messungen die gesamte Wandstärkenverteilung zu rekonstruieren. Weiterhin wird das Konzept so ausgeweitet, dass Sensorausfällen auch in Echtzeit instanzbasiert entgegengewirkt werden kann. Hierdurch wird der Einsatz der Methodik für Dashboard-Lösungen ermöglicht.

5.2.1 Störgrößenkompensation und Rauschreduktion

Die in Abschnitt 5.1.3 aufgeführten Störgrößen wirken sich auf die Wandstärkenmessung aus. Sie führen dazu, dass der dargestellte Verlauf verfälscht wird und das darunter liegende Profil nicht korrekt wiedergibt. Es gilt, diese externen Einflüsse von den eigentlich zu untersuchenden Wanddickenschwankungen zu unterscheiden. Abbildung 5.5 zeigt die Ziele einer robusten und möglichst verteilungsunabhängigen Störgrößenkompensation. Aus der Messdatenmatrix Y werden die kohärenten Strukturen L extrahiert, die die bestmögliche Näherung für den tatsächlichen Wandstärkenverlauf repräsentieren. Die sich additiv auswirkenden Störungen S werden herausgefiltert. Während die L -Matrix so berechnet wird, dass sie einen niedrigen Rang hat, enthält S isolierte Ausreißer und Rauschen, die nicht zwingend einer Normalverteilung folgen. Die Herangehensweise kann nicht nur für die RPCA verwendet werden, sondern lässt sich auf alle thresholdbasierten Dimensionsreduktionsverfahren übertragen.

Durch die Rotationsbewegung der Kokille und ihre gleichzeitige translative Bewegung entlang der x -Achse des Rohres bei der Messung entstehen sinusförmige Überlagerungen im aufgezeichneten Wandstärkenprofil. Sie motivieren einen domänenunspezifischen Ansatz, der in der Lage ist, dominante Frequenzen aus Zeitreihen zu extrahieren und andere auszublenden. Dieser wird durch die FFT realisiert. Unter Umständen treten Schmutzablagerungen auf der Kokille und andere innere Ungängen auf, die sich datentechnisch und analytisch nicht erfassen lassen. Aus diesem Grund werden die SVD und die RPCA als domänenspezifische unüberwachte Ansätze herangezogen. Innerhalb der Dynamik der Wandstärkenschwankungen zeigten sich auch nichtlineare Zusammenhänge. Zu diesem Zweck werden Denoising-Autoencoder zur Dimensionsreduktion genutzt. Die Auswirkungen der Störgrößenkompensation auf die Prognosegüte mit überwachtem maschinellem

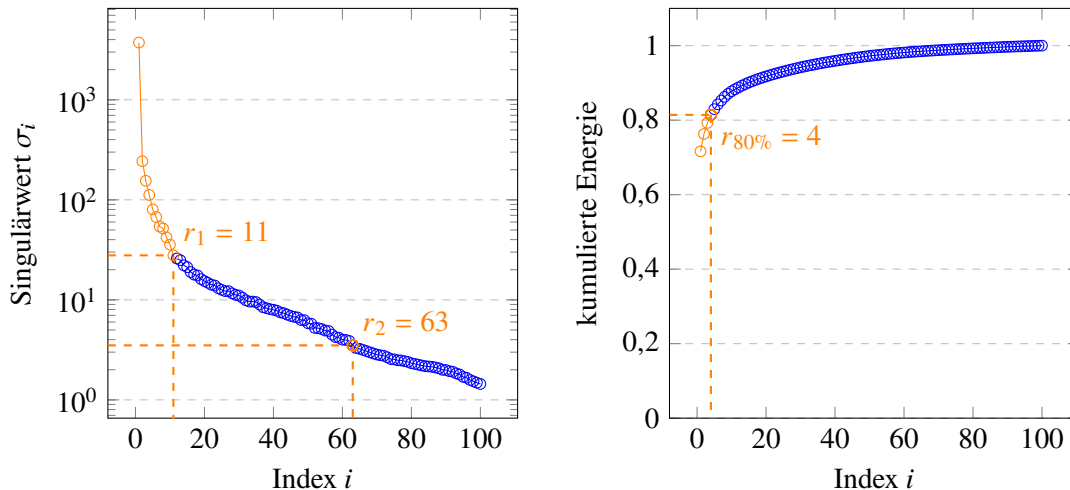


Abbildung 5.6: Logarithmischer Verlauf der Singulärwerte (links) und ihre kumulierte Energie (rechts). Sie zeigen die starke Verdichtbarkeit der Datenmenge. Es sind zwei Knickstellen zu erkennen. Die Position der Ersten legt nahe, eine Dimensionsreduktion mit 11 Singulärwerten vorzunehmen.

Lernen werden in Abschnitt 5.3 untersucht.

Dimensionsreduktion mit SVD und RPCA

Vor der Anwendung der SVD besteht die Aufgabe zunächst darin, sowohl anhand von Domänenkenntnissen als auch mittels statistischer Varianz der Hauptkomponenten den Grad der Dimensionsreduktion bzw. den numerischen Rang r der Matrix \mathbf{Y} zu ermitteln. Die Zahl der Singulärwerte, die beibehalten werden, ist ausschlaggebend. Wird sie zu hoch gewählt, also nahe der Zahl der tatsächlich vorhandenen Spalten, so ergibt sich eine Darstellung, die den Originaldaten ähnelt und zu viel Rauschen abbildet. Werden zu wenig Singulärwerte gewählt, gehen zu viele Informationen verloren und die Daten werden zu sehr verfälscht. Es wird zunächst die Singulärwertzerlegung von \mathbf{Y}

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (5.2)$$

gebildet und die 100 Diagonalelemente von $\mathbf{\Sigma}$ untersucht. Abbildung 5.6 zeigt den logarithmischen Verlauf der Singulärwerte und ihre kumulierte Energie als Partialsummen ihrer anteiligen Varianz. Erwartungsgemäß ist aufgrund der Kollinearität benachbarter Wandstärken ein starker Abfall der Singulärwerte zu beobachten. Wie im Regelfall bei Prozessdaten, die nicht künstlich generiert wurden oder einer eindeutigen Normalverteilung entstammen, kann jedoch keine unmittelbare Knickstelle erkannt werden, die es gemäß Scree-Test zulassen würde, den numerischen Rang r direkt abzulesen. Vielmehr lassen sich geringfügige Änderungen in der Steigung feststellen, sodass $r_1 = 11$ einen möglichen Threshold darstellt. Ein weiterer leichter Knick ist bei $r_2 = 63$ zu erkennen, allerdings werden so bereits 98,50 % der Gesamtvarianz erklärt. Eine anforderungsgerechte Rauschreduktion oder eine Komprimierung des Datenraumes wäre damit nicht realisierbar. Wird nach dem 80 %-Threshold vorgegangen, so lässt sich die geforderte Gesamtvarianz schon mit vier Modi erklären.

Die Datenmenge lässt sich in Bezug auf die Informationssättigung somit in drei Bereiche unterteilen, von denen der letzte kaum noch relevant ist. Dies gilt jedoch nur aus einer statistischen Sicht und berücksichtigt nicht die physikalischen Eigenschaften des Prozesses. Letztere wurden in der ersten Projektphase mit Domänenwissen abgeglichen und es konnte $r = 8$ als optimaler Threshold bestimmt werden. Mit der Kenntnis des numerischen Ranges r werden alle Singulärwerte $\sigma_{r+1}, \dots, \sigma_d$ in Σ zu Null gesetzt und die rangreduzierte Matrix

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^T \quad (5.3)$$

berechnet. Sie hat das gleiche Format wie die Originaldaten, enthält aber weniger Information. Die SVD kann als Methode des unüberwachten Lernens interpretiert werden, bei der im Trainingsvorgang die Koeffizienten der Transformationsmatrix \mathbf{V} ermittelt werden. Sie projiziert durch eine lineare Operation die Spalten von \mathbf{Y} in den Hauptkomponentenraum, wo sie durch $\mathbf{T} = \mathbf{U}\Sigma$ charakterisiert werden (vgl. Abbildung 4.6). Die gekürzte Matrix $\tilde{\mathbf{V}}$ ist allerdings nicht mehr unitär, sodass $\tilde{\mathbf{V}}^T\tilde{\mathbf{V}} = \mathbf{I}_{r \times r}$, jedoch $\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T \neq \mathbf{I}_{d \times d}$. So kann, wenn \mathbf{V} einmal bestimmt ist, eine Transformation und Rücktransformation für beliebige Messdatenmatrizen $\mathbf{Y} \in \mathbb{R}^{m \times d}$ oder ein einzelnes neu vermessenes Rohr $\mathbf{y}^{(m+i)} \in \mathbb{R}^d$ mit

$$\tilde{\mathbf{y}}^{(m+i)} = \mathbf{y}^{(m+i)}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^T \quad (5.4)$$

durchgeführt werden. Das ermöglicht eine Online-Anwendung der SVD.

Zur Nutzung der robusten Hauptkomponentenanalyse (RPCA) in Bezug auf die Rohrdaten wird auf den von Candès et al. (2011) entwickelten Algorithmus zurückgegriffen. Der Hintergrund für die Anwendung ist die Dünnbesetztheit und deterministische Natur einiger Störungen, die Ausreißercharakter besitzen und von der RPCA bedient werden können (Scherl et al. 2020). Die Matrizen \mathbf{L} und \mathbf{S} sind hierfür so zu bestimmen, dass

$$\mathbf{Y} = \mathbf{L} + \mathbf{S} \quad (5.5)$$

Das Optimierungsproblem

$$\min_{\mathbf{L}, \mathbf{S}} \text{rank}(\mathbf{L}) + \|\mathbf{S}\|_0, \quad \text{sodass } \mathbf{L} + \mathbf{S} = \mathbf{Y} \quad (5.6)$$

wird mit dem Ansatz der alternierenden Richtungen wie durch Brunton und Kutz (2019, S. 108) iterativ gelöst. Der Grad der Rangreduktion lässt sich hierbei nicht explizit, sondern nur indirekt über den Threshold K_{th} im Algorithmus beeinflussen. Als Skalierungsfaktor

$$\varepsilon_r := \frac{K_{th}}{\|\mathbf{Y}\|_F} \quad (5.7)$$

wurde $\varepsilon_r = 0,1$ empirisch ermittelt. Er bewirkt, dass die \mathbf{L} -Matrix einen vergleichbaren Rang wie $\tilde{\mathbf{Y}}$ der SVD besitzt, um die Resultate miteinander vergleichen zu können.

Die Auswirkung der Rangreduktionen von SVD und RPCA wird in Abbildung 5.7 beispielhaft an einem Rohr, also einer Zeile von $\tilde{\mathbf{Y}}$ bzw. \mathbf{Y} , als Vergleich zwischen tatsächlicher Messung und reduzierter Darstellung aufgezeigt. Es werden für beide Verfahren Rangreduktionen von $r = 8$ demonstriert. Die Ergebnisse ähneln sich in weiten Teilen. Es kann jedoch festgestellt werden, dass die mit RPCA reduzierten Daten beim gleichen Rang stärkere Abweichungen zu den Originaldatenpunkten aufweisen. Somit wird der

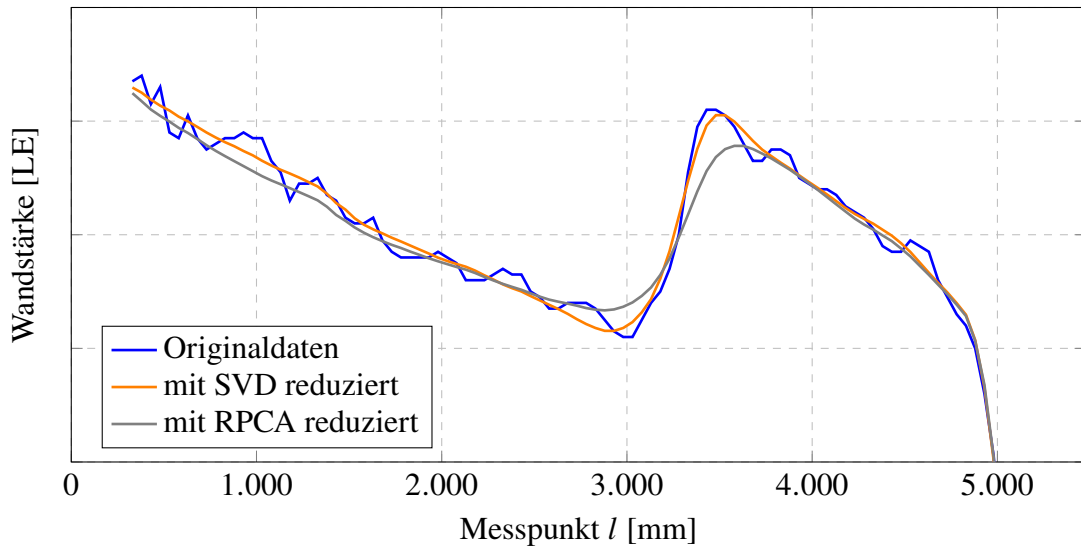


Abbildung 5.7: Der mit der RPCA gefilterte Wandstärkenverlauf weicht von den Originaldaten stärker ab als der der SVD (Rang der SVD $r = 10$, Threshold der RPCA: 0,1). Die \mathbf{L} -Matrix der RPCA hat in diesem Fall den gleichen Rang wie $\tilde{\mathbf{Y}}$ der SVD.

Wandstärkenverlauf² bei gleichem Informationsgehalt zwar schlechter rekonstruiert. Die Tatsache, dass ausgeprägte Berge und Täler im Wandstärkenprofil von der RPCA nicht abgebildet werden, zeugt jedoch von ihrer Robustheit und der zugrundeliegenden ℓ_1 -Optimierung. Obwohl im Nachhinein nicht eindeutig belegbar, kann angenommen werden, dass sich an diesen Messpunkten mehrere Störungen überlagert haben. Eine endgültige Bewertung der Rangreduktion lässt jedoch erst die Untersuchung der Prognosequalität auf Basis der unterschiedlichen Darstellungen zu.

Im Interesse einer robusteren Bestimmung des Thresholds und um Aufschluss über den Grad der Verdichtung zu erhalten, werden im Folgenden die Abstände der rekonstruierten Matrizen $\tilde{\mathbf{Y}}$ der SVD bzw. \mathbf{L} der RPCA miteinander verglichen. Als Abstandsmetrik wurde die Frobenius-Norm

$$\|\mathbf{Y}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m y_{ij}^2} \quad (5.8)$$

gewählt. Der numerische Rang r der SVD wird explizit vorgegeben, während er bei der RPCA rückwirkend in Bezug auf die \mathbf{L} -Matrix bestimmt wird. Hierzu wurde der Skalierungsfaktor ε_r für den Threshold von 0,1 bis 0,001 mit einer Schrittweite von 0,001 variiert und dann der zugehörige Rang von \mathbf{L} berechnet. Dabei war 0,1 der größtmögliche Wert, bei dem ein plausibler Wandstärkenverlauf ermittelt werden konnte. Werte unter 0,001 führen für die vorliegenden Daten aufgrund der wachsenden Zahl der Iterationen zu einem erheblichen zeitlichen Rechenaufwand für die Konvergenz der Lösung. Für $r = \text{rank}(\mathbf{L})$ konnten Werte zwischen 9 und 43 gefunden werden, die jedoch trotz der konstanten Schrittweite nicht äquidistant verteilt sind. In Abbildung 5.8 ist der Abstand

²Darstellung in Längeneinheiten (LE) aus betrieblichen Gründen.

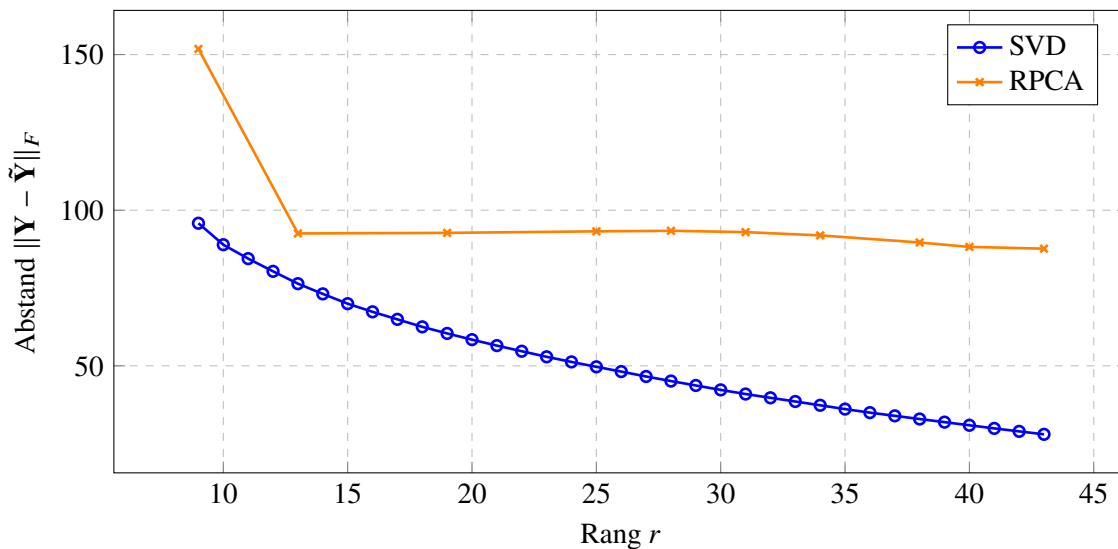


Abbildung 5.8: Vergleich der Differenzen in der Frobenius-Norm zwischen reduzierter und Originalmatrix für SVD und RPCA. In dieser Darstellung entspricht $\tilde{\mathbf{Y}}$ der \mathbf{L} -Matrix der RPCA.

der reduzierten und vollständigen Datenmatrix gegenüber dem Grad der Verdichtung für beide Verfahren aufgetragen. Erwartungsgemäß konvergiert der Abstand der SVD bei ansteigendem Rang r zu Null. Er ist weiterhin stets kleiner und bestätigt so den Satz von Eckart und Young (1936), wonach die SVD die beste Rang- r -Näherung für die Matrix \mathbf{Y} hinsichtlich der Frobenius-Norm liefert. Beim Abstand der RPCA hingegen stellt sich ab einem Rang von $r = 13$ eine Sättigung ein, sodass er sich auch für höhere Ränge nur noch geringfügig verändert. Es lässt sich darüber hinaus eine klar erkennbare Knickstelle verorten, die nahelegt, $r = 9$ bzw. $\varepsilon_r = 0,1$ als optimalen Threshold zu wählen. Dieser Wert ähnelt dem durch Expertenwissen abgeschätzten von $r = 8$ und liegt niedriger als $r = 11$, der mittels der Knickstelle des logarithmischen Verlaufs der Singulärwerte abgelesen wurde. Weitere Ränge für \mathbf{L} zwischen 9 und 13 ließen sich auch bei einer Schrittweite kleiner als 0,001 nicht finden.

Dimensionsreduktion mit FFT

Da es sich bei der Überlagerung bei der Messung der Wandstärke um eine drehzahlabhängige und somit periodische Störgröße handelt, liegt ein frequenzbasierter Ansatz nahe. Mit der FFT lassen sich dominante Schwingungen isolieren und somit Rauschen kompensieren. Zur Anwendung der Fourier-Transformation auf die Rohrdaten \mathbf{Y} wird die zweidimensionale FFT analog zur Bildverarbeitung und Komprimierung verwendet. Hierzu wird die eindimensionale FFT zunächst auf die Spalten und dann auf die Zeilen von \mathbf{Y} angewendet, wobei die Reihenfolge keine Rolle spielt. Eine Vertauschung der Operationen würde zum gleichen Ergebnis führen. Die resultierende Matrix der Koeffizienten kann nun gekürzt werden, indem hinreichend kleine Koeffizienten zu Null gesetzt werden. In diesem Beispiel werden 5 und 10 % der größten Koeffizienten beibehalten. Mit der gekürzten Matrix kann die inverse Fourier-Transformation durchgeführt werden, um rauschreduzierte Daten zu erzeugen.

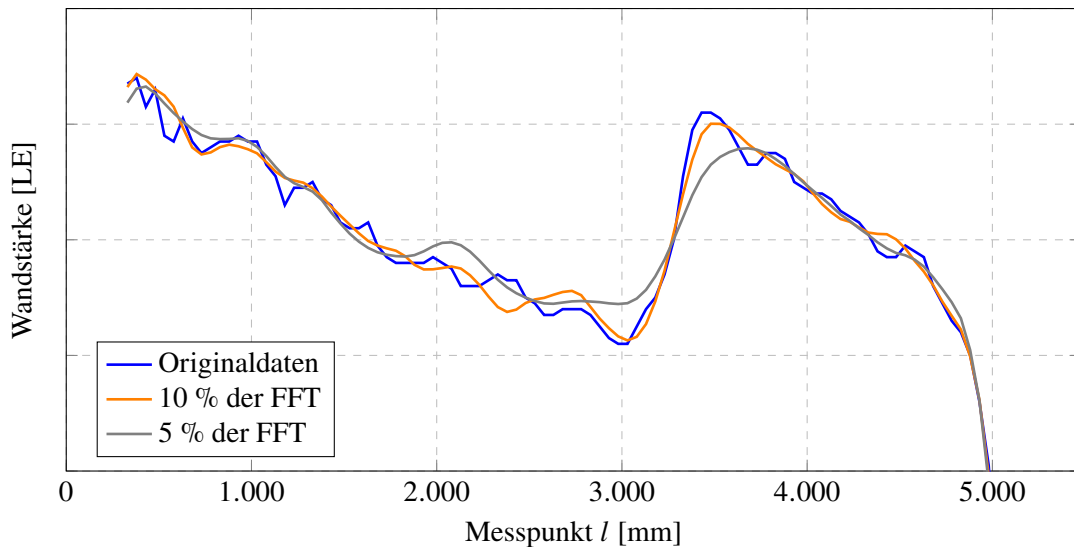


Abbildung 5.9: Rauschreduktion mit 10 und 5 % der größten Koeffizienten der FFT.

Abbildung 5.9 zeigt das Ergebnis der FFT für beide Thresholds. Werden 10 % der wichtigsten Fourier-Koeffizienten beibehalten, ähnelt der resultierende Wandstärkenverlauf am Spitzende des Rohres dem der SVD. Sowohl Berge als auch Täler werden erfasst. Im Bereich des Schafts und an der Muffe ist jedoch eine Überanpassung zu erkennen, da die sinusförmigen Störgrößen zum Teil abgebildet sind. Lediglich einige Schmutzablagerungen in Form von lokalen Ausreißern werden ausgeblendet. Bei einer Reduktion auf 5 % der Fourier-Koeffizienten werden prozessrelevante Schwankungen im Bereich 3000 mm teils nicht mehr korrekt erfasst, es ist also zu viel Information verloren gegangen. Darüber hinaus lässt sich im rechten Teil des Schafts stellenweise eine zu hohe Wandstärke erkennen, während die Kokillenovalität nicht korrigiert werden kann. Die Diskrepanzen sind einerseits in der Tatsache begründet, dass die für die sinusförmigen Überlagerungen verantwortliche Drehzahl der Kokille sich während des Prozessablaufs ändert. Andererseits ist während des Abkühlvorgangs mit einer Schrumpfung des Rohrs zu rechnen, was wiederum Disparitäten entlang der x -Achse hervorruft. Abschließend ist anzumerken, dass die FFT bei gleichem Verdichtungsgrad eine bessere quantitative Näherung der Originaldaten liefert als die SVD und deshalb für eine möglichst genaue Rekonstruktion zu bevorzugen ist.

Dimensionsreduktion mit Autoencodern

Nichtlineare Wechselwirkungen können bei den Wandstärkenmessungen nicht ausgeschlossen werden. Für ihrer Handhabung werden Autoencoderstrukturen zum unüberwachten Lernen genutzt. Es werden mehrere Varianten erprobt und die Qualität der Rauschreduktion ermittelt. Als Standard-Autoencoder wird aus einer Eingabeschicht und einer Ausgabeschicht mit 100 Neuronen aufgebaut. Der Bottleneck besteht aus $r = 8$ Einheiten und erzeugt eine Dimensionsreduktion nach einem Muster, das dem der SVD ähnelt. Als Aktivierungsfunktion der Eingabe- und Zwischenschicht dient die *Rectified*

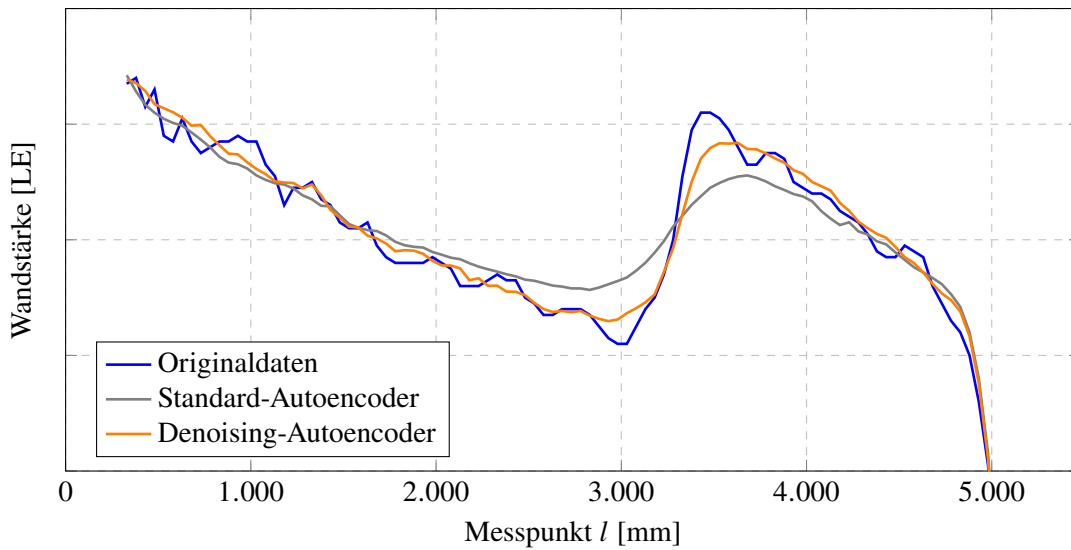


Abbildung 5.10: Rauschreduktion mit einem Denoising-Autoencoder. Beim Encoder und Decoder sind je drei Schichten mit 80, 60 und 40 Neuronen in absteigender bzw. aufsteigender Reihenfolge zwischengeschaltet.

Linear Unit (ReLU) mit

$$f_a(x) = \max(0, x). \quad (5.9)$$

Die Ausgabeschicht nutzt lineare Aktivierungsfunktionen. Die Optimierung nutzt den Adam-Algorithmus (Kingma und Ba 2014) und den MSE als Metrik der Verlustfunktion. Den Denoising-Autoencoder implementiert die Platzierung dreier zusätzlicher Schichten mit je 80, 60 und 40 Neuronen in absteigender Reihenfolge als Encoder nach der Eingabeschicht und in aufsteigender Reihenfolge vor der Ausgabeschicht als Decoder. Beide Autoencoder werden mit 100 Epochen trainiert, während das stochastische Gradientenabstiegsverfahren mit Batches von 10 Stichproben abläuft. Die Validierung in jeder Epoche erfolgt mit 10 % der Gesamtdaten.

Abbildung 5.10 zeigt das Ergebnis beider Autoencoderstrukturen für ein Rohr beispielhaft. Im Bereich der Muffe und am linken Teil des Schafts unterscheiden sich die Ergebnisse nur geringfügig, der Denoising-Autoencoder neigt jedoch zu Overfitting, wie bei $l \approx 1300$ mm erkannt werden kann. Am rechten Teil des Schafts bildet sich der gleiche Versatz wie bei der FFT. Der dünnste Teil des Rohrs und die Wandstärkenspitze werden nicht korrekt abgebildet. Dahingegen führt der DAE an diesen Stellen eine adäquate Störgrößenkompensation durch. Es ist anzumerken, dass die Ergebnisse beider Verfahren je nach zufälliger Initialisierung der Gewichte des Netzes teils starken Schwankungen unterliegen. Die Startwerte lassen sich zwecks Reproduzierbarkeit fixieren, haben aber dennoch einen Einfluss.

5.2.2 Reduktion und Rekonstruktion der Messungen

Das Volumen der vorliegenden Datenmengen stellt noch keine Big-Data-Situation dar. Daher spielen Fragen der Performance für die Modellierung und Datenkomprimierung

zwecks Speicherplatzeinsparung nur eine untergeordnete Rolle. Dennoch liefert eine komprimierte Darstellung wichtige Erkenntnisse zur Dynamik der Wanddickenschwankung. Darüber hinaus können Rechenzeiteinsparungen bei der Online-Optimierung erzielt werden, wenn für die Kostenfunktion weniger Prognosen durchgeführt werden müssen und die Wandstärken stattdessen durch lineare Operationen rechnerisch ermittelt werden.

Die Rang- r -Darstellung der Rohrdaten ist geeignet, r linear unkorrelierte Messwerte zu ermitteln, mit denen sich die Originaldaten ℓ_2 -optimal rekonstruieren lassen. Hierzu wird die Vorgehensweise aus Abschnitt 4.2.3 verwendet. Es wird $r = 10$ gewählt, da so eine hinreichend genaue Rekonstruktion gewährleistet werden kann. Der Vektor \mathbf{a} , der die Positionen der Messpunkte bestimmt, und die dazugehörige reduzierte Messdatenmatrix Φ_{SVD} sowie die Koeffizientenmatrix zur Rekonstruktion Θ_{SVD} werden einmal mit der SVD berechnet. Anschließend erfolgt die Definition von \mathbf{a} durch eine äquidistante Verteilung der Messungen entlang der x -Achse des Rohrs mit den entsprechenden Matrizen Φ_c und Θ_c . Für den vorliegenden Grad der Reduktion gilt

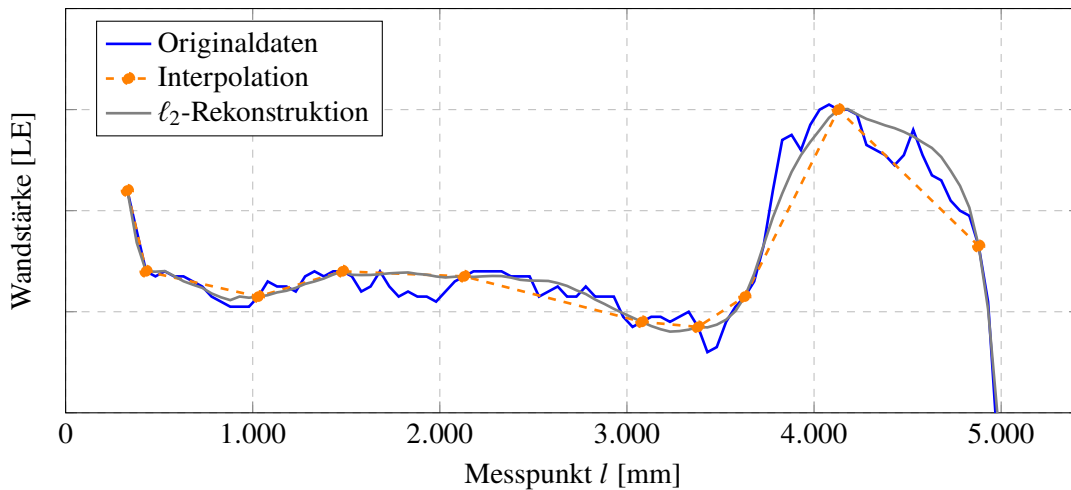
$$\|\mathbf{Y} - \Phi_{SVD}^T \Theta_{SVD}\|_F < \|\mathbf{Y} - \Phi_c^T \Theta_c\|_F. \quad (5.10)$$

Abbildung 5.11 zeigt beispielhaft das Ergebnis beider Platzierungen. Die Punkte sind lediglich zur Veranschaulichung miteinander verbunden. Eine Rekonstruktion durch lineare Interpolation wird aufgrund des hohen Informationsverlustes nicht realisiert, sondern soll die Originaldaten einbeziehen. Mit der SVD werden Messungen nahe der Muffe und bei $l \approx 1500$ mm gesetzt, wo die Pudermittelzugabe erfolgt. Letztere wirkt als Störfaktor auf die Dynamik der Wandstärkenschwankung und erschwert eine rechnerische Rekonstruktion anhand benachbarter Werte. Weitere Messungen sind am dünnsten Teil des Rohrs und beim Wandstärkenmaximum nötig. Der restliche Verlauf kann unter Berücksichtigung der Störgrößen adäquat abgebildet werden. Bei der homogenen Verteilung der Messungen lassen sich aufgrund mehrerer Fehlplatzierungen Aliasingeffekte erkennen. Sie führen zu einer Verfälschung der Originaldaten.

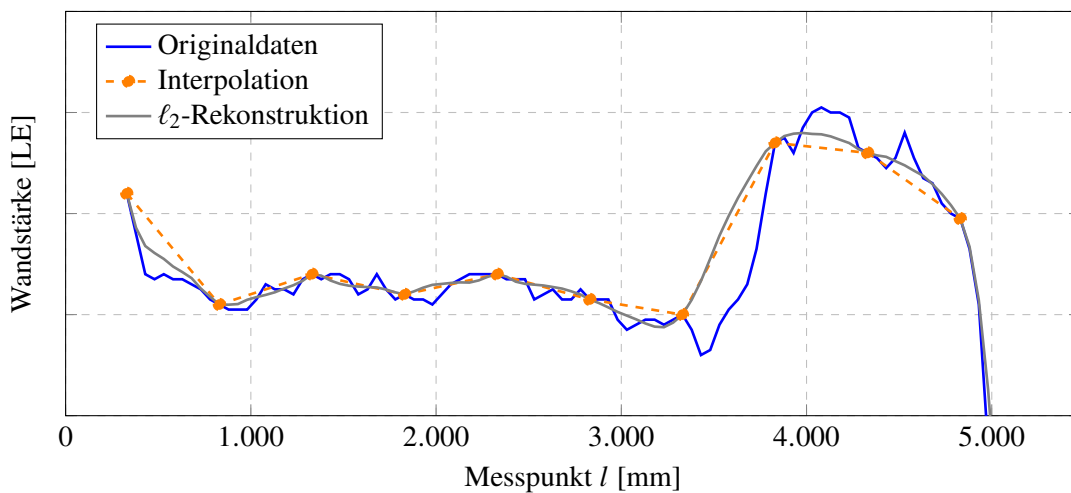
5.2.3 Handhabung von Sensorausfällen

Ausfälle der Sensorik führen dazu, dass abschnittsweise Wandstärken nahe Null angezeigt werden. Liegen diese bereits in der Prozessdatenbank, können sie vor der Analyse herausgefiltert werden. Dazu eignet sich die PCA-basierte Q-Statistik, da der Verlauf an den betroffenen Stellen stark von dem der rangreduzierten Darstellung abweicht. Durch eine Projektion der Instanz in den Residuenraum kann ein Sensorausfall so problemlos festgestellt werden. Falls die betroffenen Rohre nicht entfernt werden sollen, kann alternativ eine Rekonstruktion mit Hilfe der RPCA durchgeführt werden. Zum Erlernen der Teilmatrizen \mathbf{L} und \mathbf{S} sollten die fehlerhaften Daten jedoch nicht verwendet werden. Treten die Sensorfehler während der Online-Visualisierung auf, müssen sie in Echtzeit korrigiert werden können, um dem Maschinenbediener einen unterbrechungsfreien Überblick über das Prozessgeschehen zu verschaffen.

Wie in Abschnitt 5.2.1 beschrieben, können SVD und Autoencoder, wenn sie einmal trainiert sind, auf einzelne Prozessinstanzen angewendet werden. Mit der RPCA ist dies nicht unmittelbar möglich, eine ganzheitliche Berechnung der rauschreduzierten Datenmatrix benötigt jedoch weniger als eine Sekunde Rechenzeit, sodass sie für diesen Prozess



(a) Messpunkte mit SVD platziert.

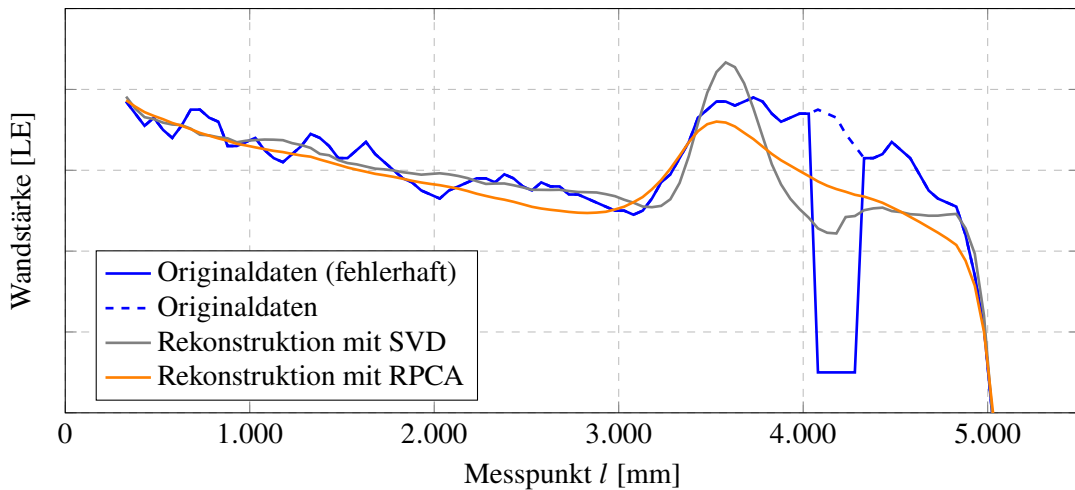
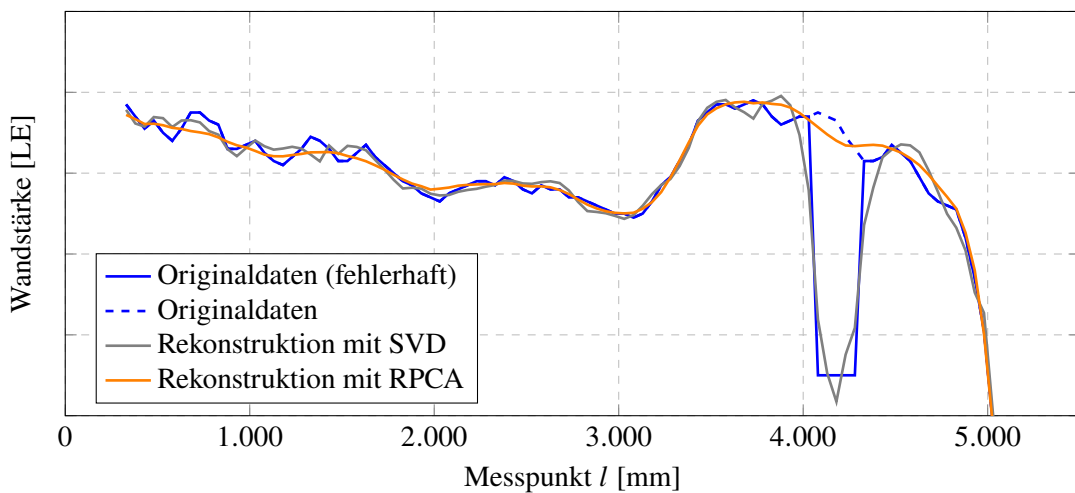
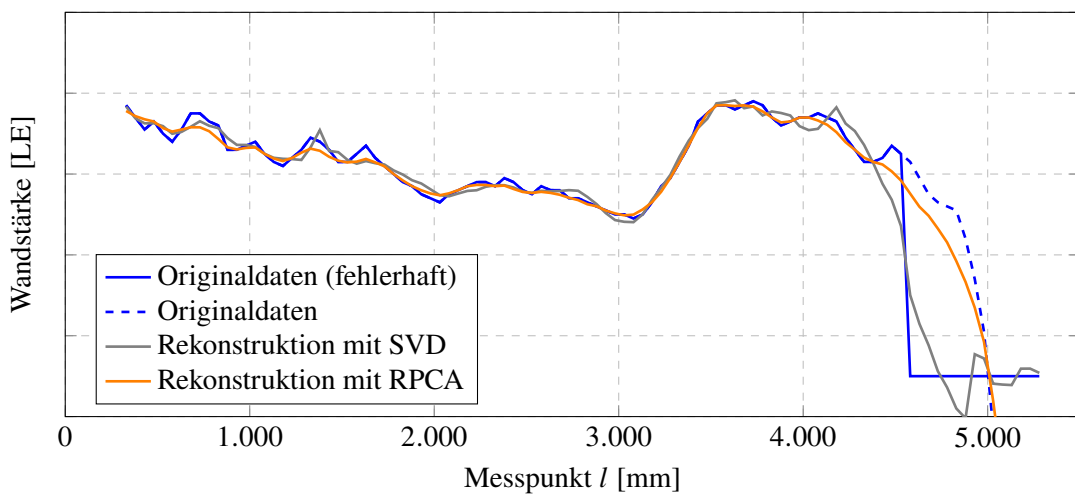


(b) Messpunkte äquidistant verteilt.

Abbildung 5.11: Beispielhafte Verdichtung der Messung bei $r = 10$. Die mit der SVD gewählten Messpunkte erzielen eine bessere ℓ_2 -Rekonstruktion. Die Interpolation der Messungen ist nur aus Gründen der Anschaulichkeit eingezeichnet.

für jedes Rohr neu initialisiert werden kann. Im nachfolgenden Beispiel werden Sensorausfälle am Spitzende des Rohrs simuliert. Es werden jeweils fünf aufeinanderfolgende Messungen auf einen Wert gesetzt, der deutlich unter dem der erwarteten Wandstärke an der entsprechenden x-Position liegt. Fehlerhaft aufgezeichnete Rohre in der Prozessdatenbank waren nur im Bereich des Schafts betroffen, der für die Endqualität eine weniger kritische Rolle spielt.

Abbildung 5.12 zeigt die Ergebnisse der Rekonstruktion für SVD und RPCA. Wie in Abschnitt 4.2.3 angedeutet, sind für diese Art der Datenaufbereitung andere Thresholds zu nutzen. Im oberen Teil wurden $r = 10$ und $\varepsilon_r = 0,1$ gewählt. Die Rekonstruktion ist mit diesen Parametern jedoch unzureichend: Der SVD-Verlauf schwingt am Anfang des Spitzendes über und fällt danach schnell ab, während die RPCA die Wandstärken

(a) $r = 10, \varepsilon_r = 0,1$: Underfitting beider Verfahren.(b) $r = 20, \varepsilon_r = 0,01$: Overfitting der SVD.(c) $r = 20, \varepsilon_r = 0,01$ für einen Sensorausfall am Rohrende: Overfitting der SVD.**Abbildung 5.12:** Simulation von Sensorausfällen am Spitzende des Rohrs mit unterschiedlichen Parametern der SVD und RPCA.

qualitativ besser, aber durchgängig zu niedrig wiedergibt. Im nächsten Schritt werden die Parameter zu $r = 20$ bzw. $\varepsilon_r = 0,01$ angepasst und die Rekonstruktion erneut durchgeführt. Erwartungsgemäß ist eine Überanpassung in Form einer periodischen Überlagerung am Schaft zu erkennen, die bei der SVD stärker ausgeprägt ist. Letztere bildet auch den Sensorausfall fälschlicherweise ab, während die RPCA den Verlauf beständig repliziert. Sie zeigt sich weiterhin in der Lage, Ausfälle von 15 Messwerten am äußersten Ende des Rohrs zu kompensieren, wo keine Möglichkeit zur Interpolation besteht. Die Qualität der Projektion nimmt jedoch mit zunehmender Zahl fehlender Messungen ab.

5.3 Wandstärkenmodellierung

Im folgenden Abschnitt werden Prognosen des Wandstärkenverlaufs mit überwachtem maschinellen Lernen angestellt. Die Formulierung der Zielvariablen und der späteren Optimierung erfolgt als Regressionsproblem. Mathematisch werden die 100 Wandstärken getrennt betrachtet, um einerseits Einflussgrößen besser abzugrenzen und andererseits die Reaktion der Einzelmodelle auf eine Prozessvariante getrennt betrachten zu können, ohne dass Korrelationseffekte die Interpretation verfälschen. Die Einzelmodelle werden schließlich zu einer Gesamtprognose eines Rohrs auf Grundlage der Prozessdaten kombiniert. Im Anschluss erfolgt die empirische Modellvalidierung durch eine virtuelle Versuchsplanung zur Identifikation des Einflusses wichtiger Steuergrößen. Sie beinhaltet die Sensitivitätsanalyse durch eine zweidimensionale partielle Abhängigkeit und die Untersuchung von Scheinkorrelationen.

5.3.1 Einzelprognosen

Die Evaluation der maschinellen Lernmodelle erfolgt nach der von Chollet (2021, S. 133–136) beschriebenen Methode. Es wird eine zufällige Aufspaltung der Lern- und Testdaten mit einem Verhältnis von 70/30 vorgenommen. Die Hyperparameter tuning nutzt ausschließlich die Lerndaten und eine Kreuzvalidierung mit $k = 10$. Die optimalen Hyperparameter und die gesamte Lerndatenmenge werden schließlich für die Bestimmung des finalen Modells genutzt (vgl. hierzu Abbildung 4.8). Die nachfolgenden Erkenntnisse und Metriken beziehen sich auf die Modellvorhersagen hinsichtlich der zurückgehaltenen Testdatenmenge. Als Bewertungskriterien dienen das Bestimmtheitsmaß für den linearen Zusammenhang zwischen Prognose und wahren Werten, der *RRMSE* mit

$$RRMSE = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (f(\mathbf{x}^{(i)}) - y_i)^2}}{\bar{y}} \quad (5.11)$$

als Indikator der Abweichung und die Steigung β der linearen Regressionsgeraden nach

$$(\alpha, \beta) = \arg \min_{\alpha, \beta} \sum_{i=1}^N (\hat{y}_i - \alpha - \beta y_i)^2. \quad (5.12)$$

Als erster Schritt werden Einzelmodelle für die Wandstärke an ausgewählten Messpunkten trainiert, um die Qualität der Vorhersage lokal zu untersuchen. Der Vergleich der

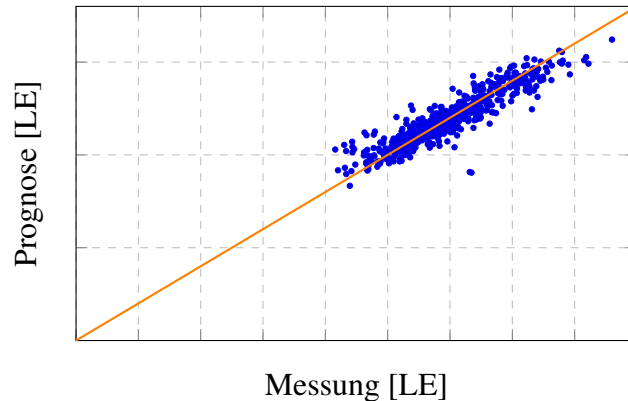


Abbildung 5.13: Gegenüberstellung von Messungen und Prognosen der Wandstärke bei $l = 580$ mm mit RPCA als Vorverarbeitung unter Betrachtung der gesamten Achse. $R^2 = 0,8138$, $RRMSE = 0,0575$ (bei Testdaten).

Prognoseperformance verschiedener Modelle, das Hyperparameter tuning und die Ermittlung eines geeigneten Metamodells schließen sich an. Abbildung 5.13 zeigt das Ergebnis der Einzelprognose der Wandstärke bei $l = 580$ mm unter Betrachtung der vollständigen x- und y-Achsen zur Einordnung der Gesamtschwankung. Die nachfolgenden Darstellungen beschränken sich nur noch auf den relevanten Achsenbereich. Anhand der Wanddickenvorhersage bei $l = 2080$ mm werden die Auswirkungen der unterschiedlichen Dimensionsreduktionsmethoden miteinander verglichen. Der vorliegende Messpunkt eignet sich für die Bewertung, da die Wirkgrößen für die Wandstärke hier weitestgehend bekannt sind und somit eine hohe Prognosegüte erreicht werden kann. Für das Einzelmodell wird zunächst ein CatBoost-Regressor mit Standardeinstellungen verwendet.

Abbildung 5.14 zeigt die Gegenüberstellung von Prognosen und Messungen für die Rohdaten sowie die störgrößenkompensierten Daten jeweils mit FFT, SVD und RPCA. Das Modell weist für die Rohdaten mit $R^2 = 0,8851$ und $RRMSE = 0,0340$ bereits eine akzeptable Prognosequalität auf. Die Steigung der Regressionsgeraden weicht mit $\beta = 0,8713$ jedoch deutlich von der Identität ab und deutet auf einen Bias bei der Prognose hin. In den Randgebieten besteht die Tendenz, niedrige Werte zu hoch und hohe Werte zu niedrig zu schätzen. Das Phänomen tritt bei Regressionsproblemen häufig auf und lässt sich teilweise auf die Varianz der Daten zurückführen, die durch das Modell nicht erklärbar ist. Im vorliegenden Beispiel ist eine begründete Hypothese, dass die Störgrößen aus statistischer Betrachtung zufällig auftreten und somit vom Modell nicht abgebildet werden können. So führen periodische Überlagerungen, beispielsweise durch die Ovalität der Kokille, zu Abweichungen der Prognose nach oben und nach unten für das gesamte Wandstärkenspektrum. Ausreißer, die beispielsweise durch Schmutzablagerungen entstehen und meist isoliert auftreten, täuschen hohe Wandstärken vor, die vom Modell zu tief geschätzt werden und systematische Abweichungen nach unten ergeben. Sie sind im oberen rechten Teil des Streudiagramms zu erkennen und erklären den uneinheitlichen Verlauf der Regressionsgeraden.

Die RPCA zeigt die beste Eignung zur Vorverarbeitung der Datenbasis. Neben der Korrektur einzelner Ausreißer bewirkt sie eine Verdichtung der Schätzungen zur Identitätsgeraden. Dies äußert sich in einem verbesserten Bestimmtheitsmaß von $R^2 = 0,9562$ und einem um 42 % verringerten relativen Fehler von $RRMSE = 0,0196$, wobei anzu-

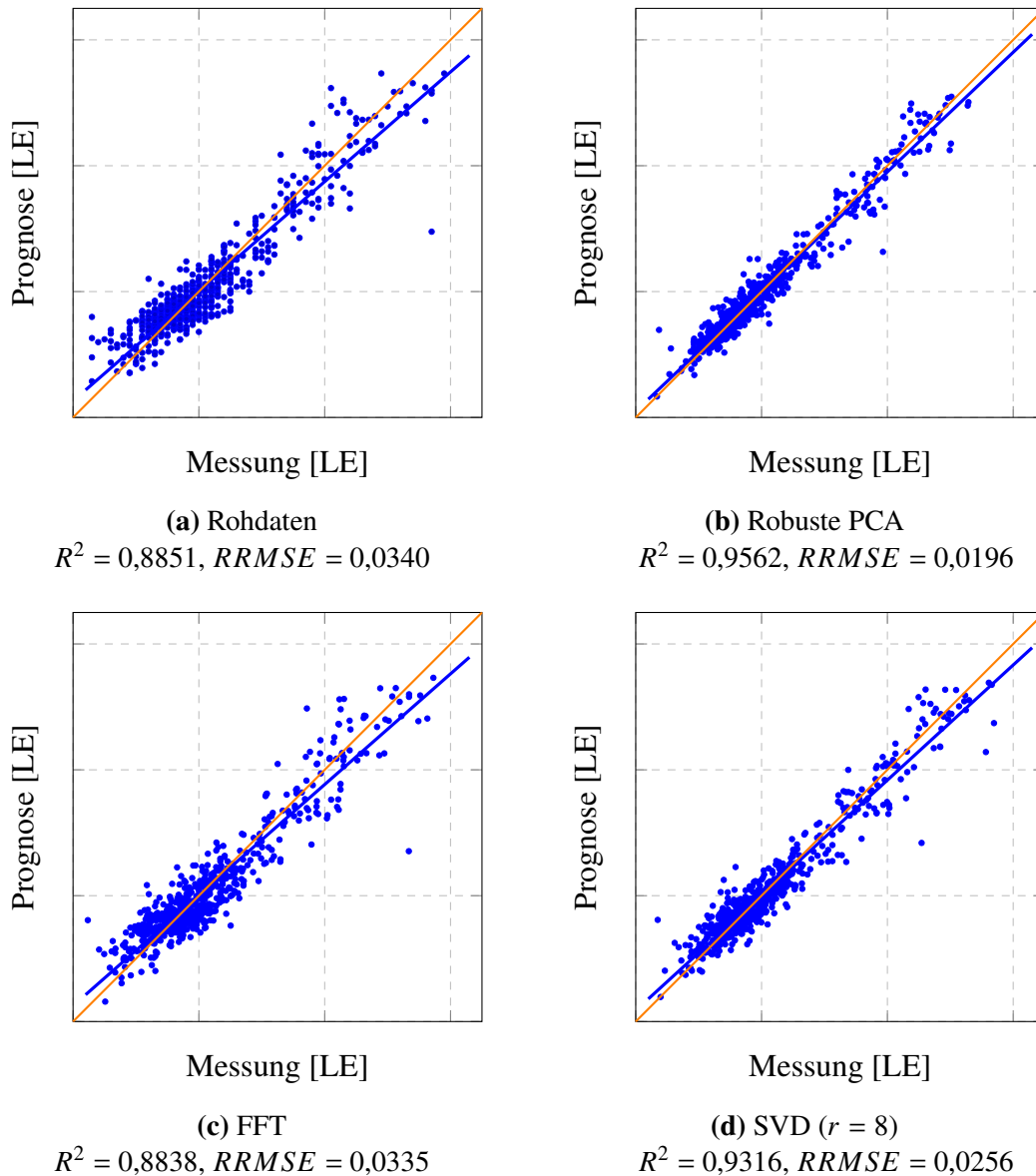


Abbildung 5.14: Gegenüberstellung von Messungen und Prognosen der Wandstärke bei $l = 2080$ mm mit Rohdaten, RPCA, SVD und FFT.

merken ist, dass beide Gütekriterien auf der ℓ_2 -Norm beruhen und somit erheblich von Ausreißern beeinflusst werden. Die Vorverarbeitung der Daten mit SVD erzielt ebenfalls eine Verbesserung der Prognose mit $R^2 = 0,9316$ und eine Korrektur der Ausreißer. Die wahre Stärke der RPCA liegt jedoch in der Kompensation des Prognosebias, die eine höhere Steigung der Regressionsgeraden von $\beta = 0,9506$ hervorbringt. Dies verdeutlicht die Wirksamkeit der ℓ_1 -optimalen robusten Störgrößenkompensation. Eine Vorverarbeitung mittels FFT bewirkt eine Verschlechterung der Prognoseperformance hinsichtlich R^2 wie auch $RRMSE$.

Die Auswirkung der RPCA auf den Prognosebias drückt sich auch in der Häufigkeitsverteilung der Vorhersagen aus. Der unterste Teil von Abbildung 5.14 zeigt ein Histogramm der Residuen als Differenz aus Prognosen und wahren Werten auf Grundlage der Rohdaten und der RPCA. Wird nach Bangert (2021, S. 29) der Mittelwert der Verteilung als

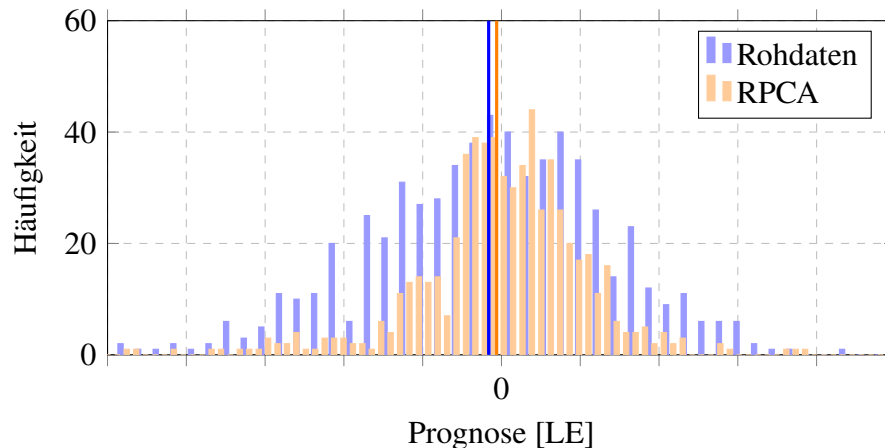


Abbildung 5.15: Histogramm der Prognoseresiduen. Es ist eine Verschiebung des Mittelwerts der Residuen (vertikale Linien) in Richtung Null erkennbar.

Qualitätskriterium der Prognose herangezogen, erkennt man, dass die Datenaufbereitung eine Verschiebung des Erwartungswertes der Residuen gen Null bewirkt. Die Schätzung wird damit erwartungstreu, darüber hinaus attestiert die Abbildung die Verdichtung der Prognosen um den Mittelwert bzw. eine Reduktion der Streuung.

Je nach Messposition l entlang der Rohrachse variieren sowohl Einflussparameter als auch Prognosequalität. Dies ist vor allem der Messbarkeit der Prozessparameter aufgrund der Umgebungsbedingungen geschuldet. Wirken an einer Rohrposition Prozessgrößen, die zwar einen entscheidenden Einfluss haben, sich aber quantitativ nur schwer ermitteln lassen, leidet die Prognosequalität und es kommt zu einer erhöhten Streuung. Abbildung 5.16 verdeutlicht dies am Beispiel der Wandstärkenmessung bei $l = 4280$ mm, einer Position nahe dem Spitzende des Rohres. Die Fließeigenschaften der Schmelze wie ihre Viskosität und die Temperatur haben dort einen Einfluss. Auch in diesem Fall bewirkt die Störgrößenkompensation eine signifikante Verbesserung der Prognosequalität. Das Bestimmtheitsmaß steigt $R^2 = 0,7939$ auf $R^2 = 0,8362$, sodass der CatBoost-Regressor nach wie vor in der Lage ist, einen Großteil der Wandstärkenschwankungen zu erklären. Die Verzerrung durch Ausreißer kann in weiten Teilen abgeschwächt werden, gerade im statistisch schlecht abgesicherten unteren Bereich bleibt sie jedoch in dieser Form bestehen. Sowohl die Prognoseunsicherheit als auch die Verteilung der Messwerte werden bei der Synthetisierung der Prozessvarianten später im Interesse plausibler Ergebnisse berücksichtigt.

5.3.2 Performanceanalyse der Modelle

Um die Eignung der Einzelmodelle miteinander zu vergleichen, werden diese hinsichtlich der Wandstärke an ausgewählten Messpositionen trainiert und mittels Bestimmtheitsmaß und $RMSE$ bewertet. Repräsentative Messpunkte sind für den Bereich der Muffe 580 mm, für den Schaft 2080 mm und für das Spitzende bzw. den Punkt des Wandstärkenmaximums 4280 mm. Obwohl für baumbasierte Regressionsverfahren im Normalfall nicht notwendig, wird die Datenbasis vor dem Training auf einen Mittelwert von Null und eine Einheitsvarianz normalisiert. Dies verbessert die Handhabbarkeit stark heterogen

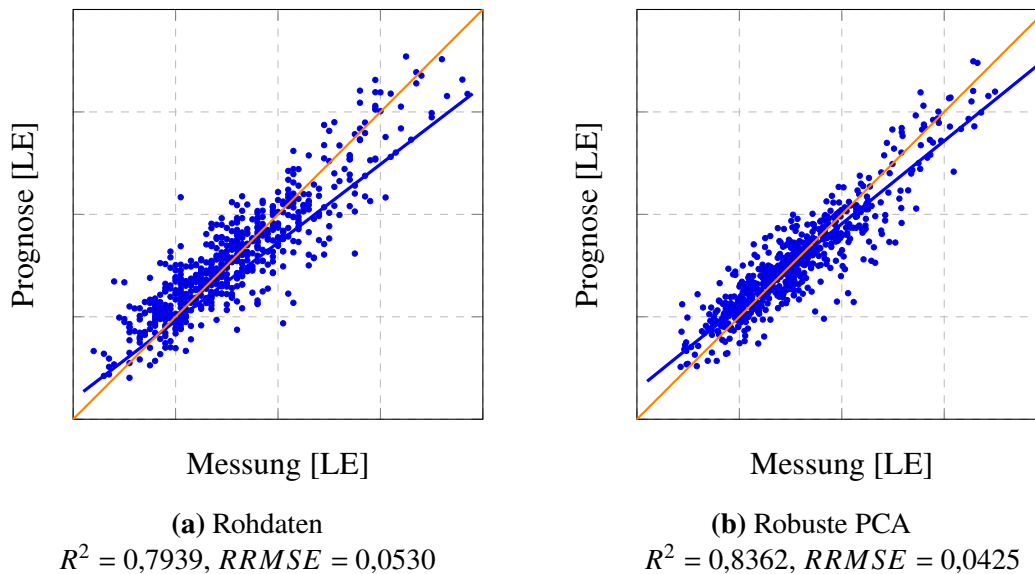


Abbildung 5.16: Gegenüberstellung von Messungen und Prognosen der Wandstärke bei $l = 4280$ mm mit Rohdaten und RPCA.

skalierender Variablen wie beispielsweise der Kombination aus Flüssigeisentemperatur und Impfmittelzugabe. Insbesondere das Multilayer Perceptron und die Support Vector Regression profitieren hiervon.

Tabelle 5.1 zeigt eine Übersicht der Prognoseperformance ausgewählter Regressionsverfahren. Sie wurden mit ihren Standardparametern trainiert und anhand der zurückgehaltenen Testdaten bewertet. Der RRMSE erlaubt den Verzicht auf die Einheit der Wandstärke. Die höchste Prognosegenauigkeit bei Einzelmodellen erzielen der CatBoost-Regressor und der Light Gradient Boosting Machine. Andere Verfahren, die auf Boosted Trees beruhen, erweisen sich als ähnlich performant wie der in Scikit-Learn nativ unterstützte Random Forest. Support-Vector-Regression (SVR), neuronale Netze (MLP) und die instanzbasierten k-Nearest-Neighbors (kNN) bieten ebenfalls eine hinreichend genaue Prognose. Ein lineares Regressionsmodell und seine Abwandlungen LASSO und Elastic Net können in ihrer Ursprungsform ohne Hyperparameteranpassungen die Zusammenhänge nicht adäquat abbilden.

Pipeline-Optimierung mit TPOT

Zur Ermittlung eines Meta-Regressors wird auf die TPOT-Umgebung (Olson et al. 2016) zurückgegriffen. Sie implementiert eine auf genetischen Algorithmen beruhende stochastische Suche nach einer optimalen Prognose-Pipeline für die gegebenen Daten. Obwohl der Einsatz zur Featureauswahl, der Vorverarbeitung und der Erzeugung von Metavariablen unterstützt wird, erfolgt die Nutzung in diesem Fall lediglich zur Modellauswahl und zur Hyperparameteroptimierung. Zur automatischen Optimierung ist neben den spezifischen Kennwerten der genetischen Algorithmen wie der Zahl der Generationen und der Populationsgröße eine Kreuzvalidierungsstrategie anzugeben. Es werden darüber hinaus sämtliche Modelle, die in Scikit-Learn enthalten sind, neuronale Netze im Rahmen von PyTorch und XGBoost berücksichtigt. CatBoost und LightGBM werden nicht unterstützt.

Tabelle 5.1: Performancevergleich der Einzelmodelle angewandt auf die Testdaten hinsichtlich Bestimmtheitsmaß und RRMSE bei den Messpunkten 580 mm, 2080 mm und 4280 mm. Vorverarbeitung mit RPCA.

	Bestimmtheitsmaß			RRMSE		
	580 mm	2080 mm	4280 mm	580 mm	2080 mm	4280 mm
CatBoost	0,8144	0,9562	0,8362	0,0580	0,0196	0,0425
XGBoost	0,7379	0,9352	0,7937	0,0696	0,0239	0,0477
LGBMRegressor	0,7719	0,9443	0,7913	0,0643	0,0220	0,0479
Optimierte Pipeline	0,7780	0,9469	0,7881	0,0654	0,0220	0,0636
Entscheidungsbäume	0,6351	0,8458	0,5299	0,0890	0,0379	0,0777
Gradient Boosting	0,7412	0,9293	0,7379	0,0685	0,0251	0,0541
Random Forest	0,7894	0,9425	0,7788	0,0618	0,0225	0,0497
AdaBoost	0,5668	0,8619	0,5755	0,0911	0,0398	0,0689
Lineare Regression	0,6092	0,7116	0,5969	0,0845	0,0502	0,0666
SGDRegressor	0,6022	0,6936	0,5905	0,0851	0,0518	0,0671
Elastic Net	0,6117	0,7090	0,5971	0,0842	0,0504	0,0666
LASSO	0,6135	0,7102	0,5969	0,0839	0,0503	0,0666
Bayesian Ridge	0,6109	0,7072	0,5970	0,0841	0,0506	0,0666
SVR	0,7627	0,9096	0,7695	0,0658	0,0286	0,0508
MLP	0,7503	0,8860	0,7636	0,0688	0,0318	0,0520
kNN	0,6774	0,8939	0,6770	0,0768	0,0308	0,0600

Die automatische Optimierung beginnt beispielhaft mit der Prognose bei $l = 2080$ mm. Zur Validierung wird eine 10-fache Kreuzvalidierung mit drei Wiederholungen genutzt. Die Optimierung anhand genetischer Algorithmen zur Findung des Meta-Regressors erfolgt mit 100 Individuen und fünf Generationen. Als Gütekriterien werden sowohl Bestimmtheitsmaß als auch RMSE verwendet. Das beste Ergebnis für beide Qualitätskriterien wird bei einer Optimierung hinsichtlich des RMSE erzielt. Für alle drei Prognosemodelle wird eine zweistufige Pipeline erzeugt:

- $l = 580$ mm: Zwei Gradient Boosted Trees mit je 100 Bäumen, einer Lernrate von 0,1 und einer maximalen Tiefe von je neun und zwei.
- $l = 2080$ mm: Als erste Stufe ein Elastic Net mit einer anteiligen Gewichtung des ℓ_1 -Strafterms von 0,8. Als zweite Stufe ein Extra Trees Regressor mit 100 Bäumen.

- $l = 4280$ mm: Zwei Gradient Boosted Trees mit je 100 Bäumen, einer Lernrate von 0,1 und einer maximalen Tiefe von je neun und zwei.

Die von TPOT entworfene Pipeline enthält einen *Stacking Estimator*. Dieser prognostiziert die abhängige Variable auf Grundlage der Lerndaten mit dem ersten Modell und fügt die Vorhersage als weitere Spalte den unabhängigen Variablen hinzu. Die neue Datenmenge dient anschließend als Eingang für das zweite Modell. Seine Prognose fungiert als Ausgang des Meta-Regressors. Die Ergebnisse der optimierten Pipeline enthält Tabelle 5.1. Sie übertreffen zwar stets die der Einzelmodelle, liegen aber dennoch unterhalb des CatBoost-Verfahrens mit Standardparametern. Für keine der 100 Wandstärken konnte bei den gewählten Einstellungen die Performance übertroffen werden.

5.3.3 Modellierung des Gesamtprofils

Für die nachfolgenden Untersuchungen wird zunächst der CatBoost-Regressor aus dem letzten Abschnitt genutzt. Es wird für jede der 100 Wandstärkenmessungen unabhängig voneinander je ein Modell trainiert. Die Modelle fußen auf den gleichen Eingangsvariablen und werden mit den gleichen Daten trainiert und ausgewertet. Zur Untersuchung der positionsabhängigen Prognosequalität wird der *RMSE* jedes Modells hinsichtlich der Testdaten mit der Schwankung der Prognosevariablen in Form der Standardabweichung verglichen. Abbildung 5.17 zeigt für den gemittelten Wandstärkenverlauf die Prognoseunsicherheit und die Standardabweichung. Der *RMSE* ist sowohl für positive als auch für negative Abweichungen aufgetragen, da er die gleiche Einheit wie die berechnete Wandstärke besitzt. Somit beinhaltet der grau eingefärbte Bereich ca. 68 % aller Prognosen. Der gleiche Anteil ist für die Messungen gestrichelt hinterlegt.

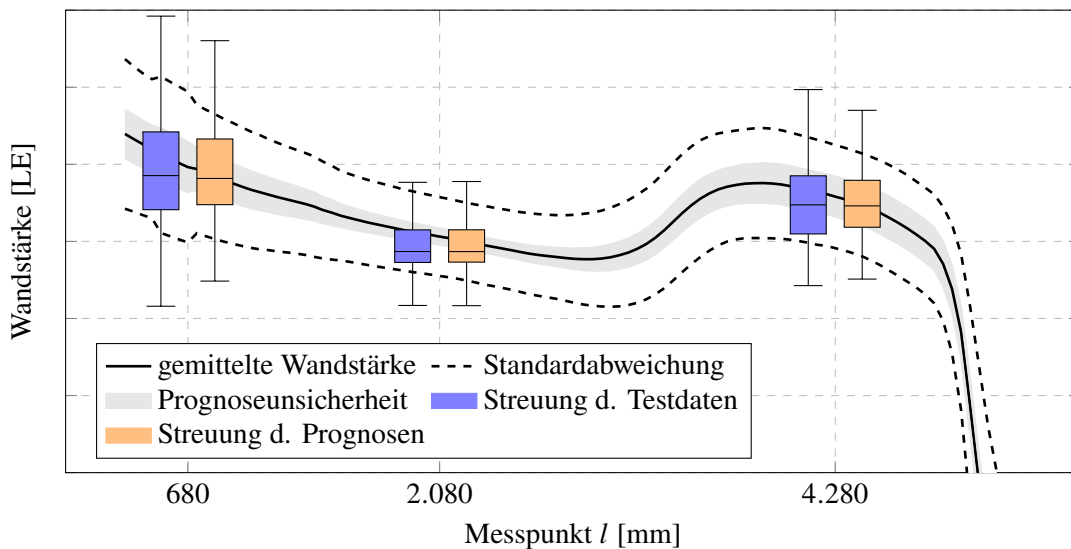


Abbildung 5.17: Prognoseunsicherheit als *RMSE* in Bezug auf den gesamten Wandstärkenverlauf im Vergleich zur Schwankung als Standardabweichung der Messdaten.

Ein Großteil der nicht erklärbaren Varianz der Modelle ist für die Prognosen am Spitzende des Rohrs zu verorten. Hier liegt der Modellfehler in der Größenordnung von fast

einem Drittel der regulären Schwankungen innerhalb der Daten. Die Abweichungen sind vorwiegend auf die hier einwirkenden Einflussgrößen zurückzuführen. Wie gleich später noch gezeigt, hat die Flüssigeisentemperatur beim Kokilleneintritt einen maßgeblichen Einfluss auf die Erstarrung und die Wandstärkenausbildung am Spitzende. Eine Messung während des Gießprozesses ist jedoch zum derzeitigen Stand der Technik nur mit begrenzter Genauigkeit möglich. Die Aufnahme der Temperaturdaten und ihre einzelteilbezogene digitale Verfügbarkeit sind jedoch essentiell für die Modellierung. So ist, wie in Abbildung 5.16 erkennbar, bei der Prognose der Wandstärke zwar mit einer erhöhten Streuung zu rechnen, hohe Schwankungen lassen sich aber dennoch weitestgehend auf die Prozessgrößen zurückführen. Darüber hinaus lassen sich Wandstärkenprognosen am Schaft mit hoher Genauigkeit vornehmen. Der relative Fehler entspricht dort weniger als einem Fünftel der normalen Schwankung. Im Bereich der Muffe steigt hingegen die Ungenauigkeit, was sich teils mit einer erhöhten Varianz der Originaldaten begründen lässt. Weitere Faktoren sind Störgrößen und externe Einflüsse wie Kernsandrückstände.

5.4 Sensitivitätsanalysen

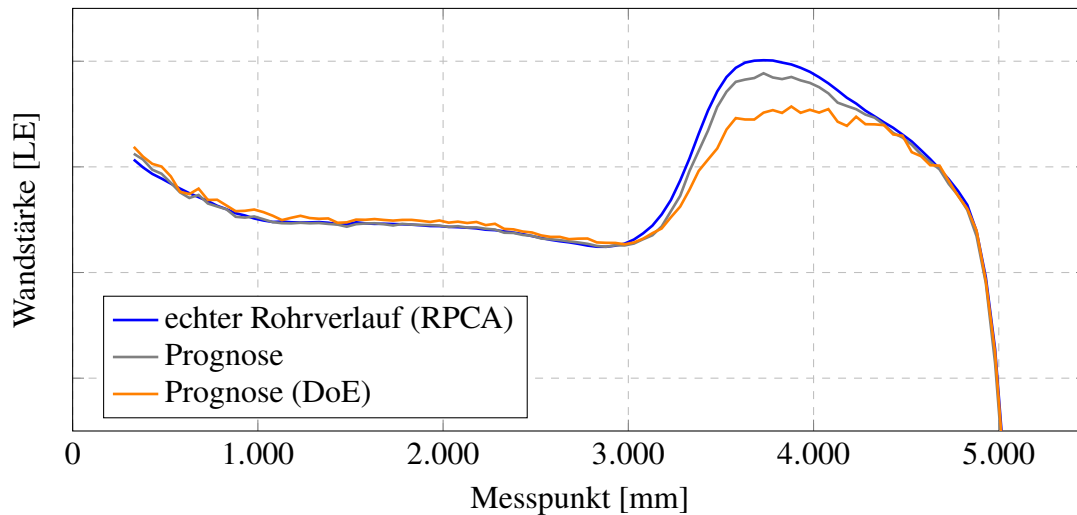
Zur Veranschaulichung und zur Verifizierung der Modellergebnisse kommen Sensitivitätsanalysen zum Einsatz. So können ermittelte Wirkzusammenhänge auf ihre physikalische Plausibilität überprüft werden. Der Einfluss einzelner Prozessgrößen wird zunächst für ausgewählte Rohre und dann ganzheitlich mit der partiellen Abhängigkeit untersucht. Unter Berücksichtigung der statistischen Absicherung werden abschließend quantitative Einflüsse analysiert und ihre Nutzbarkeit für die Optimierung bewertet.

5.4.1 Plausibilitätsprüfung mit virtuellen Parametervariationen

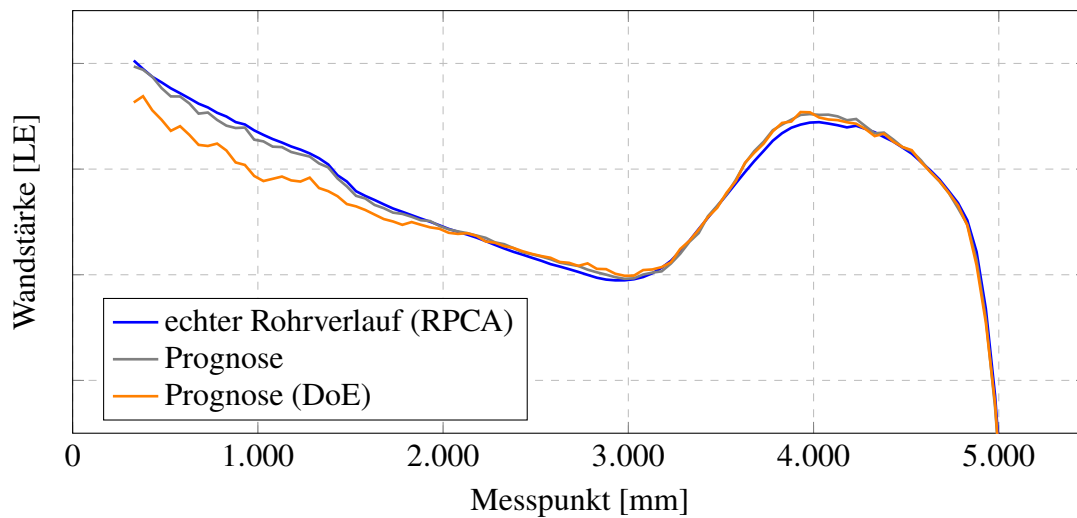
Im Rahmen mehrerer virtueller Einzel-DoEs werden die Auswirkungen gezielter Parametervariationen auf den Wandstärkenverlauf untersucht. Die dabei betrachteten Wirkzusammenhänge sind bereits bekannt und lassen sich physikalisch leicht nachvollziehen. Dazu wird nur jeweils eine Prozessgröße verändert und alle anderen konstant gehalten, um die Effekte möglichst isoliert zu betrachten. Etwaige Wechselwirkungen und Kreuzkorrelationen können dabei dennoch nicht ausgeschlossen werden. Eine Verifizierung wird nachfolgend für ausgewählte Prozessinstanzen vorgenommen. Folgende Einflussgrößen werden variiert:

- Die *Restwegstrecke* der fahrbaren Kokille nach Zurückfahren des Kippers beschreibt die x-Koordinate, ab der kein neues Eisen mehr in die Rinne fließt.
- Die *Kippergeschwindigkeit* vor dem Gießen bestimmt das Tempo, mit dem das Flüssigeisen zum ersten Mal in die Rinne eintritt.

Für die stichprobenhafte Untersuchung des Modellverhaltens werden alle 100 Prognosen zur Ermittlung des Wandstärkenverlaufs durchgeführt. Vor dem Hintergrund der späteren Optimierung ist die Zahl der Berechnungen jedoch signifikant zu verringern, um die Laufzeit zu begrenzen. Abbildung 5.18 zeigt beispielhaft zwei Rohre, die den Testdaten entnommen wurden und deren Wandstärken den Modellen noch nicht zum Training



(a) Erhöhung der Rückfahrposition des Kippers.



(b) Verringerung der Kippergeschwindigkeit vor dem Gießen.

Abbildung 5.18: Einzel-DoEs zur Validierung von erlernten Wirkzusammenhängen der Modelle. Abgebildet sind Veränderungen einzelner Parameter und ihre Auswirkungen auf den Wandstärkenverlauf ausgewählter Rohre.

vorlagen. Als wahre Verläufe werden die mit RPCA störgrößenkompensierten Daten zugrunde gelegt. Aufgrund der Unabhängigkeit der Einzelmodelle voneinander reagieren sie bei der Prognose und auf Parameteränderungen unterschiedlich empfindlich. Es kommt ferner zu einem vermehrten Grundrauschen bei der Wandstärkenverteilung, je weiter die Parameterkombinationen von der ursprünglichen Verteilung abweichen.

Abbildung 5.18 zeigt im oberen Teil den Einfluss der Abstellposition des Kippers. Durch eine Erhöhung der Restwegstrecke der Kokille kann die Wandstärke am Spitzende des Rohrs ab ca. 3200 mm stark verringert werden. Andere Bereiche bleiben davon nahezu unberührt, da die Erstarrung dort in den meisten Fällen bereits fortgeschritten ist und eine Veränderung des Fließverhaltens ohne Auswirkungen bliebe. Ein Zurückfahren des Kippers ging in der Praxis jedoch zwangsläufig mit einer geringeren verarbeiteten Ge-

samteisenmenge einher. Letztere wurde bei der virtuellen Parametervariation aber nicht verändert. Eine begründete Annahme für das Modellverhalten ist das Auffangen des Resteizens in der Rinne, was eine Umverteilung des Einsatzmaterials nicht erforderlich macht.

Der untere Teil von Abbildung 5.18 zeigt die Auswirkung der Kippergeschwindigkeit vor dem Gießen. Durch sie lässt sich die Wandstärke im Bereich der Muffe und im vorderen Teil des Rohrs reduzieren. Mit zunehmender Länge sinkt der Einfluss jedoch, sodass die Kippergeschwindigkeit sich für eine Steuerung des Prozesses in diesem Rohrabschnitt eignet. Der Aspekt des Modellverhaltens hinsichtlich der Verteilung der Gesamteisenmenge ist in diesem Fall jedoch analog zur Kipperendposition zu beachten. Der Effekt der Anpassung der zugegebenen Impfmittelmenge auf die Rissneigung an der dünnsten Stelle des Rohrs konnte in wenigen Fällen gezeigt werden. Ein systematischer Zusammenhang konnte jedoch nicht hergestellt werden, sodass diese Prozessgröße nicht für die Steuerung berücksichtigt wird.

5.4.2 Systematische Evaluation von Einflussgrößen

Die systematische Interpretation der Einflussgrößen in den Einzelmodellen erfolgt mit Hilfe der Permutationsrelevanz und der SHAP-Werte für die jeweilige Wandstärke. Sie hat das Ziel, einerseits Wirkzusammenhänge quantitativ darzustellen und andererseits die grundlegenden Steuerungsparameter für eine Prozessoptimierung zu identifizieren. Beide Interpretationsverfahren werden auf den trainierten CatBoost-Regressor aus dem letzten Abschnitt angewendet. Als Datenbasis dient der dort genutzte Lernbestand.

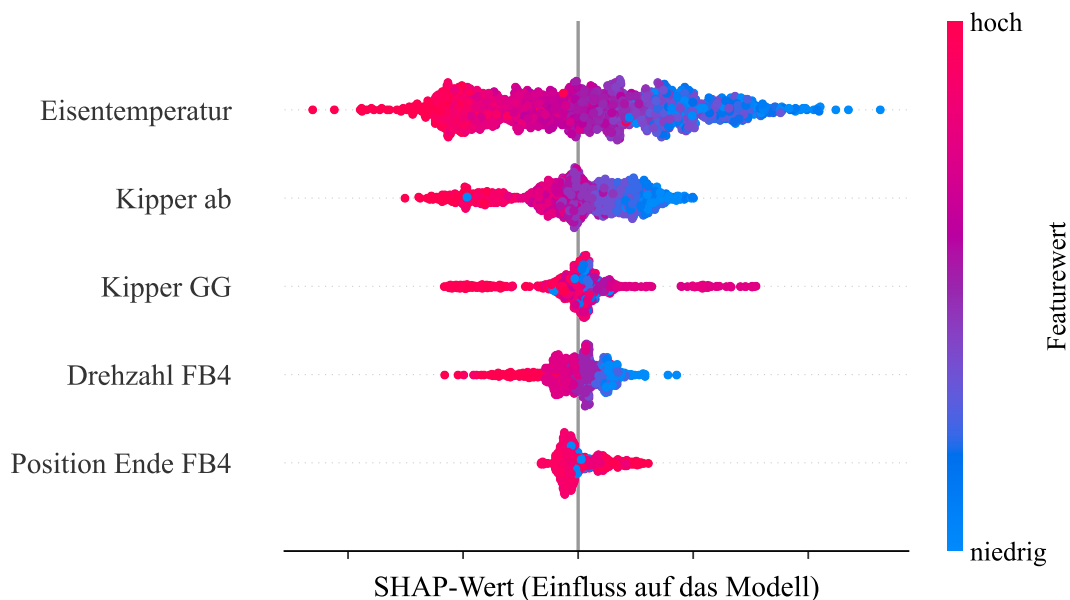


Abbildung 5.19: SHAP-Werte für die Prognose bei $l = 4080$ mm.

Abbildung 5.19 zeigt die SHAP-Werte³ für einen Auszug der sechs wichtigsten von insgesamt 57 verwendeten Prozessparametern. Sie sind der Relevanz nach absteigend angeordnet, wobei diese mittels der Summe ihrer SHAP-Werte über alle Lerninstanzen

³Darstellung ohne Einheiten aus betrieblichen Gründen.

hinweg ermittelt wird. Der relative Einfluss auf das Modell ist auf der x-Achse aufgetragen, während der Wert der jeweiligen Features mittels der rechten Farbskala gekennzeichnet wird. Die Linie in der Mitte repräsentiert den Mittelwert der Prognose. Variablen, deren SHAP-Werte nahe dieser Nullachse kumulieren, haben somit keinen systematischen Einfluss. Linksseitig wird demnach eine mögliche Reduktion der Wandstärke dargestellt. Isolierte Werte an beiden Rändern deuten auf Ausreißer hin und sollten für die Wirkzusammenhänge nicht in Betracht gezogen werden.

Die Schwankungsbreite und die Verteilung der Punkte impliziert, dass die Flüssig-eisentemperatur den höchsten systematischen Einfluss aufweist. Für eine Absenkung des Wandstärkenmaximums kann ein ideales Temperaturfenster angestrebt werden, wobei sie sich nicht direkt steuern lässt. Aufgrund der Gegenläufigkeit der Zielgrößen und zur Sicherstellung der Wanddickenhomogenität sind die anderen Modelle hierfür jedoch auch zu berücksichtigen. Eine Forderung nach einer möglichst hohen Eisentemperatur würde in diesem Fall nicht ausreichen. Eine positionsabhängige Untersuchung des Einflusses folgt im nächsten Abschnitt. Den zweithöchsten Einfluss an dieser Messposition bildet die Rückfahrposition des Kippers (Kipper ab), deren Effekt bei der Einzelparametervariation bereits nachgewiesen wurde. Der Gießgeschwindigkeit des Kippers (Kipper GG) wird zwar hohe Bedeutung seitens des Modells beigemessen, die Punkte sind jedoch ungleich verteilt und deuten auf keinen systematischen Zusammenhang hin. Die Drehzahl der Kokille in Fahrbereich 4 wirkt sich geringfügig, aber in plausibler Weise aus. Bei der Endposition des Kokillenwagens in Fahrbereich 4 handelt es sich um einen Scheineffekt, dessen Ausmaß im Rahmen der Datenvorverarbeitung bereits reduziert wurde. Genauere Ausführungen zur Vorgehensweise finden sich am Ende des Abschnitts.

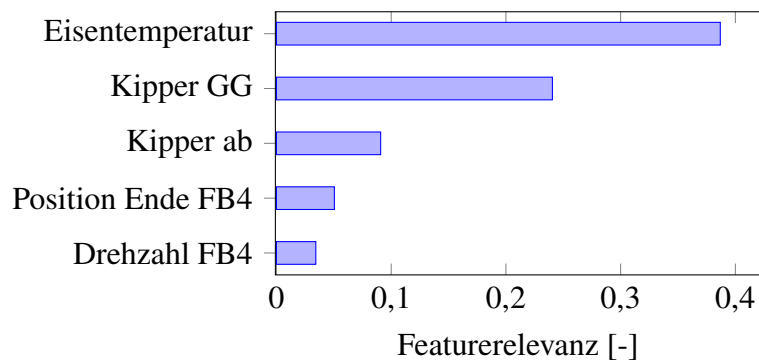


Abbildung 5.20: Featurerelevanz für die Prognose bei $l = 4080$ mm.

Die in Abbildung 5.20 gezeigten Ergebnisse für die Permutationsfeaturerelevanz ähneln den SHAP-Werten qualitativ. Es wurden für jede Variable 10 Permutationen durchgeführt und die gemittelten Relevanzen in absteigender Reihenfolge aufgetragen. Obwohl mittels der Abnahme der Modellgüte berechnet, korrelieren die PFIs weitestgehend mit der Gesamtschwankungsbreite der SHAP-Werte. Bei Letzteren werden sowohl die Summe der Werte als auch ihre Verteilung beim Erstellen des Rankings der Variablen berücksichtigt. So ergibt sich eine unterschiedliche Reihenfolge bei der Abstufung, da nur solche Prozessparameter für signifikant befunden werden, deren Einfluss durch eine breitere Verteilung der SHAP-Werte statistisch abgesichert ist. Auf diese Weise wird Ausreißern am Rand weniger Bedeutung beigemessen.

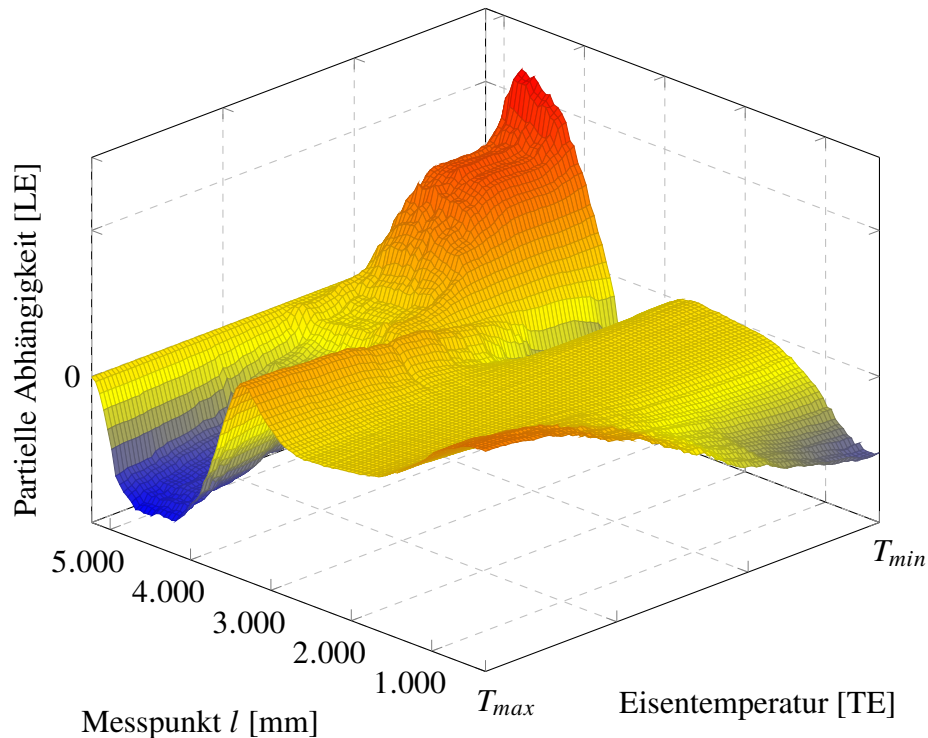


Abbildung 5.21: Sensitivitätsmatrix der Eisentemperatur als zweidimensionale partielle Abhängigkeit. Achsen aus Gründen der Anschaulichkeit gedreht.

Sensitivitätsanalysen mit partieller Abhängigkeit

Die Erkennung positionsabhängiger Wirkzusammenhänge wird anhand der Visualisierung einer zweidimensionalen partiellen Abhängigkeit realisiert. Hierfür wird statt einer zweiten Prozessvariablen der Messpunkt l entlang der Längsachse des Rohrs auf der y -Achse aufgetragen. Die z -Achse beschreibt nach wie vor die relativen Auswirkungen auf die Wanddicke. Es ist dabei anzumerken, dass für die Vielzahl der Prozessparameter die Veränderungen der Wandstärke deutlich geringer ausfallen als solche hinsichtlich der Messposition. Um die grafische Interpretierbarkeit zu gewährleisten, werden die Verläufe der partiellen Abhängigkeit um die mittlere Wandstärke am jeweiligen Punkt zu

$$\overline{pd}_S(x_S) := pd_S(x_S) - \bar{y}_i \quad (5.13)$$

korrigiert. Zur Ermittlung der Sensitivitätsmatrix wird die partielle Abhängigkeit anhand des jeweiligen Modells für jeden Messpunkt berechnet.

Abbildung 5.21 zeigt die Sensitivitätsmatrix für die Flüssigeisentemperatur⁴ in den 100 Einzelmodellen. Aus Gründen der Anschaulichkeit wurden die x - und y -Achse gedreht. Die Temperaturwerte wurden zwischen ihrem 0,05- und dem 0,95-Quantil mit linearem Abstand variiert. Als Auflösung bzw. Zahl der Auswertungen der Abhängigkeitsfunktion $pd_S(x_S)$ dient standardmäßig ein Raster von 100 Punkten. Ein klarer Zusammenhang mit der Wandstärkenausbildung ist im Bereich des Spitzendes erkennbar. So führen hohe Eisentemperaturen zu einer signifikanten Reduktion der Wanddicke, was im Einklang mit der Evaluation der SHAP-Werte steht. Obwohl sich die Temperatur der Schmelze

⁴Darstellung in Temperatureinheiten (TE) aus betrieblichen Gründen.

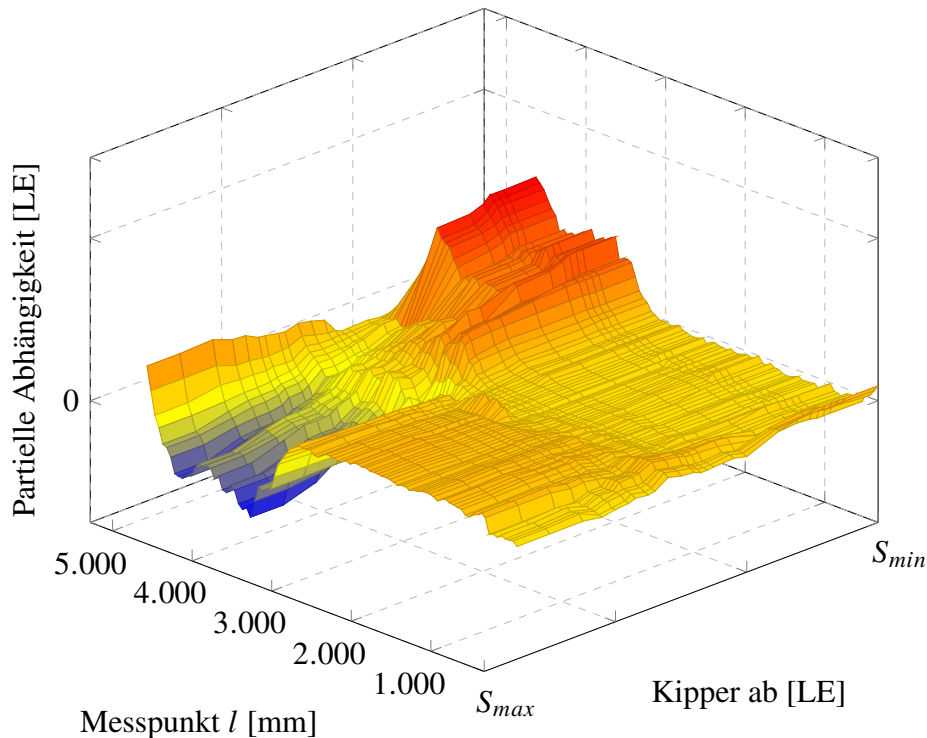


Abbildung 5.22: Sensitivitätsmatrix der partiellen Abhängigkeit der Kipperrückfahrposition. Achsen aus Gründen der Anschaulichkeit gedreht.

während der Fertigung nicht direkt anpassen lässt, ist es auf diese Weise möglich, ideale Prozessfenster zu erschließen. Hohe Temperaturen wiederum haben im Bereich der Muffe einen gegenteiligen Effekt: Sie führen zu einem leichten Anstieg der Wanddicke und verschärfen in Anbetracht der konträren Zielgrößen die Forderung nach einer automatischen Optimierung.

Abbildung 5.22 zeigt die Sensitivitätsmatrix für die Rückfahrposition des Kippers⁵. Da es sich um eine verfahrensbedingte Stellgröße handelt, ist die Auflösung der partiellen Abhängigkeit durch die Zahl der Variationen des Parameters begrenzt. Alle anderen Kennwerte wurden für die Berechnung übernommen. Es wurde weiterhin die gleiche z-Achsenkalierung gewählt. Im Vergleich zur Eisentemperatur lässt sich ein erhöhtes Grundrauschen durch Abweichungen der Prognosen benachbarter Einzelmodelle feststellen. Die unterschiedlichen Modellreaktionen lassen sich insbesondere durch den schwächeren direkten Zusammenhang mit der Wandstärke begründen. Dennoch zeigt sich ein klarer Einfluss im Bereich des Spitzendes. Er setzt bei ca. 3500 mm sprunghaft ein. Im Bereich des Schafts und der Muffe ist kein Wirkungsmuster ersichtlich, das über statistisches Rauschen hinausgeht. Die erhöhten Wandstärken am äußersten Ende des Rohrs können als statistische Artefakte eingestuft werden. Eine Abhängigkeit sollte in diesem Abschnitt nicht bestehen, da die Dicke dort messungsbedingt stets Null beträgt. Die äußersten Messwerte werden bei der Optimierung jedoch ohnehin nicht berücksichtigt.

⁵Darstellung in Längeneinheiten (LE) aus betrieblichen Gründen.

Beispiel einer Scheinkorrelation

Ein weiterer Effekt der Datenvorverarbeitung, den erst die Analyse der Einflussgrößen aufdeckt, ist beispielhaft an der Maschinenendposition im vierten Fahrbereich zu beobachten. Die nachfolgenden Untersuchungen dienen der beispielhaften Veranschaulichung und beruhen auf einer Datenbasis, von der nur in diesem Unterabschnitt Gebrauch gemacht wird. Sie enthält das Eisengewicht als Einflussgröße, die für gewöhnlich erst nach Fertigstellung des Rohres gemessen wird. Der Effekt lässt sich hierdurch besser darstellen. Für die restlichen Analysen werden die nachfolgend abgeleiteten Maßnahmen der Vorverarbeitung angewandt.

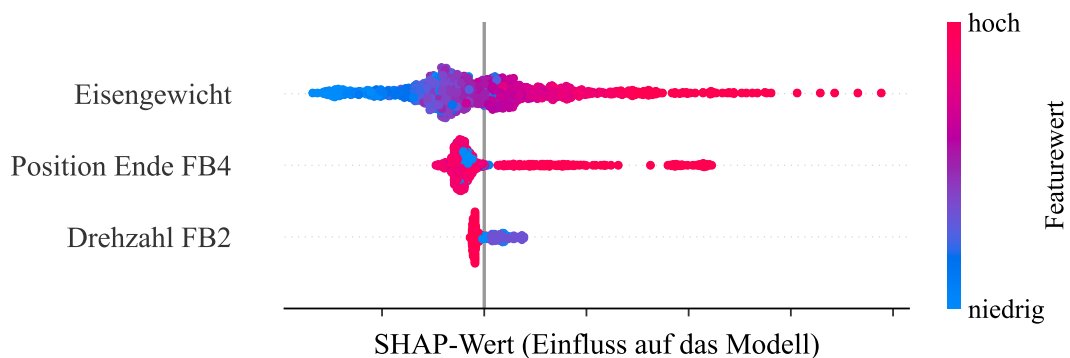


Abbildung 5.23: SHAP-Werte für die Prognose bei $l = 2080$ mm.

Abbildung 5.23 zeigt die SHAP-Werte der drei wichtigsten Features der CatBoost-Prognosefunktion für die Wandstärke bei $l = 2080$ mm, einer Messstelle am Schaft des Rohrs. Den höchsten Einfluss hat das abgeessene Eisengewicht, während eine Erhöhung der Drehzahl im zweiten Fahrbereich eine geringfügige Wanddickenreduktion hervorruft. Für die Maschinenendposition im vierten Fahrbereich lässt sich eine hohe Schwankungsbreite, jedoch aufgrund der Zahl der vorgenommenen Prognosen kein systematischer Einfluss feststellen. Ein Zusammenhang an diesem Punkt des Rohres ist weiterhin physikalisch unplausibel, da der Fahrbereich erst weiter hinten anfängt. Die Permutationsrelevanz würde dem Feature aufgrund der Schwankungsbreite jedoch eine hohe Bedeutung beimessen.

Der Effekt konnte weitestgehend auf einen verfahrensbedingten Offset bei der Wandstärkenmessung zurückgeführt werden, dessen genaue Entstehung an dieser Stelle nicht weiter erläutert wird. Er führt in einigen Fällen zu einem konstanten Versatz des Profils. Eine Korrektur erfolgte durch die Normierung der letzten Wandstärke bei $l = 5280$ mm auf null. Als Alternative zur Erkennung des Versatzes erweist sich die Hauptkomponentenanalyse von \mathbf{Y} als nützlich. Abbildung 5.24 zeigt eine Gegenüberstellung der Komponenten mit der höchsten Varianz \mathbf{z}_1 und \mathbf{z}_2 . Es kann eine zweite Gesamtheit mit ihrer eigenen Verteilung erkannt werden. Die Wandstärkenkorrektur bewirkt eine Überführung des rechten Clusters in das linke. Sollen die betroffenen Daten vor der Analyse entfernt werden, ist eine automatische Erkennung mit einem *Gaussian-Mixture-Clustering* möglich.

Abbildung 5.25 zeigt die Sensitivitätsmatrix hinsichtlich der Maschinenendposition⁶ ohne und mit Wandstärkenkorrektur. Der Einfluss ist für weite Teile des Rohrs und Werte

⁶Darstellung ohne Einheiten aus betrieblichen Gründen.

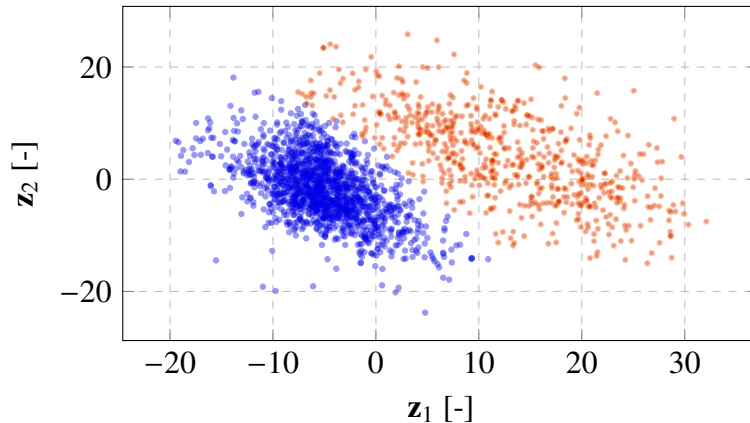


Abbildung 5.24: Gegenüberstellung der Hauptkomponenten z_1 und z_2 . Die in Orange markierten Datenpunkte können mit einem Gaussian-Mixture-Clustering erkannt und herausgefiltert werden.

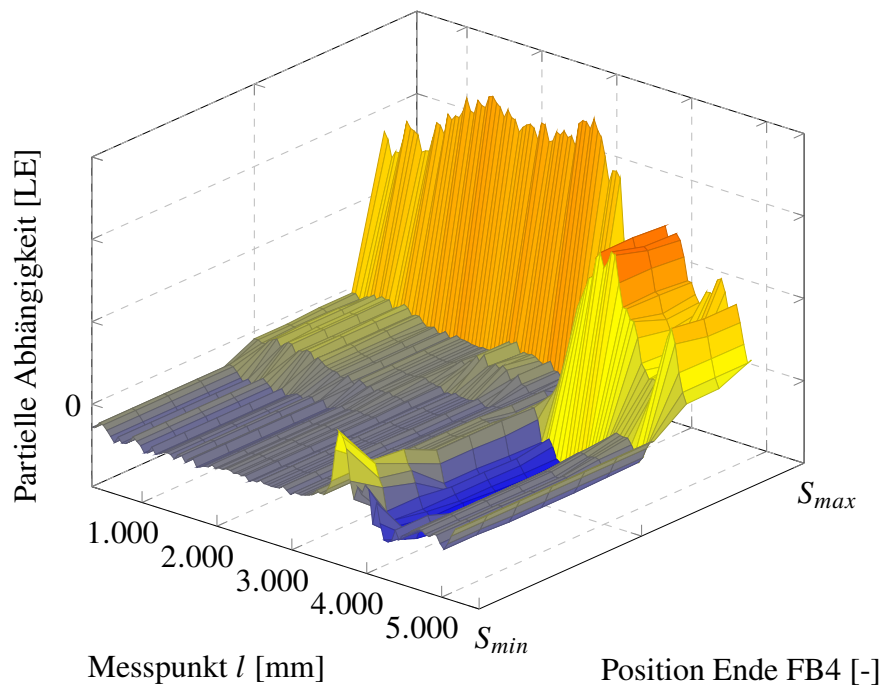
der Variablen selbst sehr niedrig. Beim höchsten Wert der Endposition nimmt die Wandstärke jedoch unabhängig der x-Position sprunghaft zu. Die erkennbaren Unstetigkeiten sind physikalisch unplausibel und deuten auf Modellfehler hin. Es ist zu vermuten, dass der Parameter in Verbindung zu Messabweichungen steht, die vom Modell wiederum abgebildet werden. Dies lässt jedoch keinen Rückschluss auf einen tatsächlichen Zusammenhang zu. Die Wandstärkenkorrektur mindert den systematischen Einfluss deutlich. Die Relevanz des Modellparameters fällt nun wesentlich geringer aus. Abhängigkeiten treten ca. zwischen 3000 und 4000 mm auf, wo ein tatsächlicher Zusammenhang bestätigt werden kann. Das vorliegende Beispiel zeigt die Wichtigkeit einer effektiven Datenvorverarbeitung für die Modellinterpretation.

5.5 Systematische Optimierung des Wandstärkenverlaufs

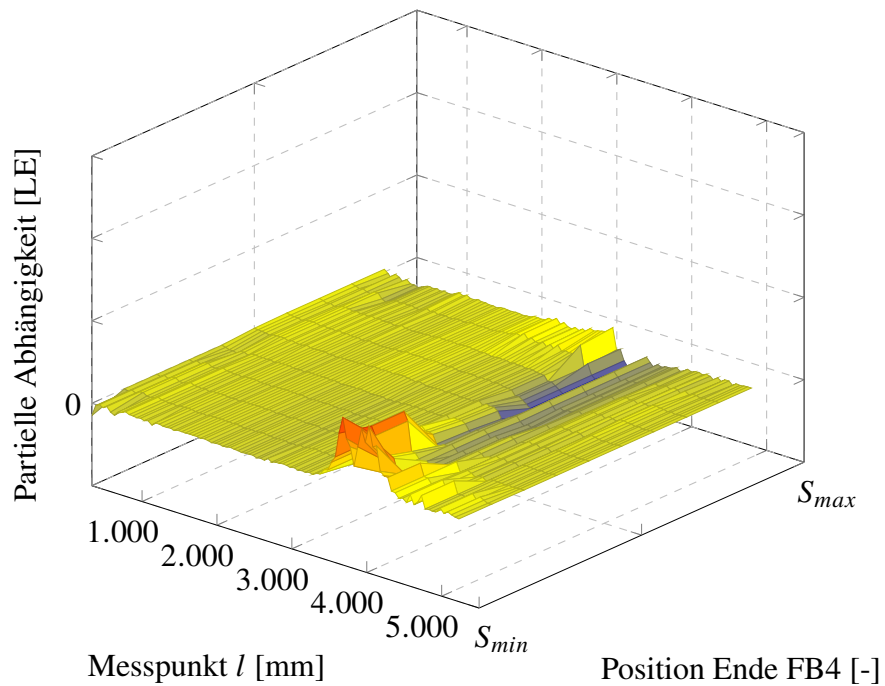
In diesem Abschnitt werden die gewonnenen Erkenntnisse aus der Datenvorverarbeitung und der Modellierung genutzt, um Parametereinstellungen für optimale Wandstärkenverläufe zu finden. Die störgrößenkompensierten Waddickendaten werden in Kombination mit den Prozessparametern zu einer optimierten Modellpipeline zusammengeführt, die synthetische Wandstärkenprofile liefert. Zwecks Optimierung werden die Komponenten der Kostenfunktion an die gegebenen Anforderungen angepasst. Die Bewertung der Optimierungslösung erfolgt anhand der Kostenfunktion, der Laufzeit und der Komplexität der generierten Handlungsempfehlung.

5.5.1 Optimierungsstrategie

Abbildung 5.26 zeigt den schematischen Aufbau der Optimierungspipeline zur Ermittlung der besten Prozessvarianten. Die aus den bereinigten und störgrößenkompensierten \mathbf{X} und \mathbf{Y} ermittelten Prognosefunktionen werden aus Abschnitt 5.3.2 übernommen. Sie bestimmen den Wandstärkenvektor $\hat{\mathbf{y}} = \begin{bmatrix} y_1 & \dots & y_d \end{bmatrix}$, der aus einer synthetisierten Prozessvariante resultiert. Aus der Waddickenmatrix \mathbf{Y} werden mit Hilfe der SVD die relevanten



(a) Sensitivitätsanalyse ohne Wandstärkenkorrektur



(b) Sensitivitätsanalyse mit Wandstärkenkorrektur

Abbildung 5.25: Auswirkung der Wandstärkenkorrektur in den Rohdaten auf die Sensitivitätsanalyse. Die Unstetigkeiten, die eine Scheinabhängigkeit im Randgebiet verursachen, können hierdurch fast vollständig entfernt werden. Aus Gründen der Anschaulichkeit sind die Abweichungen in z-Richtung relativ zu ihrem Mittelwert aufgetragen.

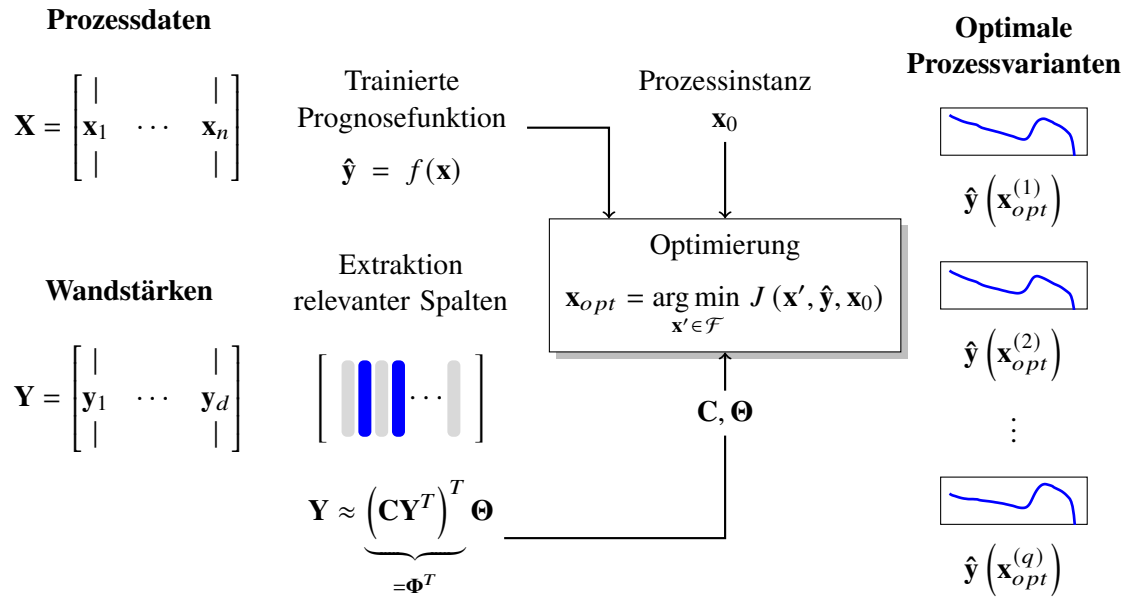


Abbildung 5.26: Optimierungsstrategie.

Spalten extrahiert. Die so entstehende dimensionsreduzierte Matrix Φ ausgewählter Messpunkte und die Matrix Θ der Regressionskoeffizienten können zur ℓ_2 -optimalen Rekonstruktion des Wandstärkenprofils genutzt werden. Für die Auswertung der Kostenfunktion werden daher die Messmatrix \mathbf{C} und Θ benötigt. Die Berechnung von

$$\mathbf{x}_{opt} = \arg \min_{\mathbf{x}' \in \mathcal{F}} J(\mathbf{x}', \hat{\mathbf{y}}, \mathbf{x}_0) \quad (5.14)$$

erfordert darüber hinaus einen Startvektor \mathbf{x}_0 , der der zu optimierenden Prozessinstanz entnommen wird, und Grenzen für \mathbf{x} in Form von \mathcal{F} , die dem Optimierungsalgorithmus übergeben werden. Die q besten Prozessvarianten werden schließlich ausgegeben.

Anpassung der Kostenfunktion

Im Folgenden werden die einzelnen Summanden der Kostenfunktion

$$J(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{x}_0) = \alpha_1 \Gamma(\hat{\mathbf{y}}) + \alpha_2 \Pi(\hat{\mathbf{y}}) + \alpha_3 \zeta(\mathbf{x}, \mathbf{X}) + \alpha_4 \Upsilon(\hat{\mathbf{y}}, \mathbf{Y}) + \alpha_5 \Xi(\mathbf{x} - \mathbf{x}_0) \quad (5.15)$$

an die Anforderungen der Optimierung angepasst. Um die Homogenität des Wandstärkenverlaufes zu gewährleisten, wird die ℓ_2 -Norm der Abweichung der Einzelwandstärken

$$\Gamma(\mathbf{y}) = \sqrt{\sum_{i=1}^d (y_i - Y_{soll})^2} \quad (5.16)$$

zum Sollwert Y_{soll} als Optimalitätskriterium herangezogen. Es wird bewusst eine Empfindlichkeit gegenüber Ausreißern eingeführt, da so Sprünge im Profil vermieden werden können.

Die Einhaltung der oberen und unteren Nebenbedingungen wird durch einen Strafterm

$$\Pi(\mathbf{y}) = \sum_{i=1}^d \max(0, g_l(y_i)) + \max(0, g_u(y_i)) \quad (5.17)$$

sichergestellt, der bei Überschreitungen der Höchstwandstärke Y_{max} bzw. Unterschreitungen der Mindestwandstärke Y_{min} linear anwächst. Dazu werden die Beschränkungsfunktionen

$$g_l(y) = \lambda_l y - Y_{min} \quad \text{und} \quad g_u(y) = \lambda_u y - Y_{max} \quad (5.18)$$

genutzt. Sie sind qualitativ in Abbildung 5.27 gezeigt.

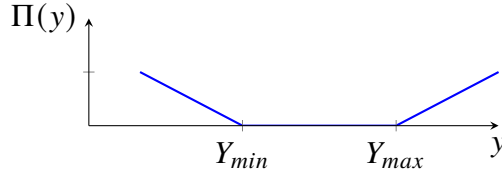


Abbildung 5.27: Beschränkungsfunktion für die Nebenbedingungen bestehend aus $g_l(y)$ und $g_u(y)$.

Die Terme $\zeta(\mathbf{x}, \mathbf{X})$ und $\Upsilon(\mathbf{y}, \mathbf{Y})$ geben den Abstand einer Prozessvariante zur Grundverteilung der Prozessdaten bzw. der Wandstärkenmessungen an. Sie haben das Ziel, anhand der Daten die physikalische Plausibilität und die verfahrenstechnische Umsetzbarkeit einer Optimierungslösung zu bewerten. Da sie sich im Rahmen der Datenaufbereitung bereits bewährt hat, wird erneut auf die Q-Statistik zurückgegriffen. Sie erfordert die Berechnung der empirischen Kovarianzmatrix und ihrer Singulärwertzerlegung, siehe dazu Gleichung (3.4). Für \mathbf{X} und \mathbf{Y} werden geeignete Thresholds gewählt, sodass $\tilde{\mathbf{P}}_X$ und $\tilde{\mathbf{P}}_Y$ zur Anomalieüberwachung genutzt werden können. Die Prüfung der Prozessvariante selbst erfolgt durch

$$\zeta(\mathbf{x}, \mathbf{X}) = \left\| \left(\mathbf{I} - \tilde{\mathbf{P}}_X \tilde{\mathbf{P}}_X^T \right) \mathbf{x} \right\|_2^2 \quad (5.19)$$

und einer Rangreduktion mit $r = 15$. Letztere wurde aus der Vorverarbeitung der Prozessdaten übernommen. Sie führte zur einer optimalen Prognoseperformance und begrenzte die Zahl der für die Analyse ausgeschlossenen Dateninstanzen. Analog dazu wird

$$\Upsilon(\mathbf{y}, \mathbf{Y}) = \left\| \left(\mathbf{I} - \tilde{\mathbf{P}}_Y \tilde{\mathbf{P}}_Y^T \right) \mathbf{y} \right\|_2^2 \quad (5.20)$$

mit $r = 10$ berechnet. Auf diese Weise konnten alle tatsächlich aufgetretenen und künstlich generierten Sensorausfälle erkannt werden. Die von Qin (2009) angeführten Verfahren zur Ermittlung eines optimalen Thresholds für die Überwachung brachten keine nutzbaren Ergebnisse. Ein möglicher Grund hierfür ist die Annahme der Normalverteilung, die bei den Prozessvariablen nicht erfüllt ist.

Die Wahl der ℓ_k -Abstandsmetrik für $\Xi(\mathbf{x} - \mathbf{x}_0)$ wird in einem späteren Abschnitt im Rahmen der Generierung der Handlungsempfehlungen gesondert untersucht.

Rahmenbedingungen für die Optimierung

Die Rahmenbedingungen der Optimierung werden nach der Vorgehensweise in Abschnitt 4.4.1 definiert. Das Wandstärkenprofil wird als Zielgröße definiert und anhand der skalaren Kostenfunktion bewertet. Als Steuerungsgrößen werden alle vor dem Schleudergießvorgang einstellbaren Parameter herangezogen. Sie umfassen:

- die Parametrierung der horizontalen Kokillenbewegung,
- die abschnittsweise eingestellte Drehzahl der Kokille
- und die Parametrierung der Kipperbewegung.

Alle weiteren gemessenen Prozessparameter werden dem Optimierungsalgorithmus als Eingangsgrößen übergeben und konstant gehalten. Die Nebenbedingungen der Zielgröße werden durch die im letzten Abschnitt beschriebenen Strafterme kontrolliert. Bei der Optimierung erfolgt jedoch keine explizite Prüfung, um Prozessvarianten zuzulassen, bei denen die vorgegebenen Wandstärkenbeschränkungen geringfügig überschritten werden. Die Nebenbedingungen der Steuerungsgrößen werden durch \mathcal{F} festgelegt und vom Optimierungsalgorithmus berücksichtigt. Falls nicht anders angegeben, dienen hierfür analog zur partiellen Abhängigkeit das 0,05- und das 0,95-Quantil jeder Prozessvariablen. Die Kostenfunktion zieht für die Bewertung des Wandstärkenverlaufs vier Messungen heran.

Alle Berechnungen wurden auf einem Desktop-Computer unter Windows 10 (64-bit) mit folgender Spezifikation durchgeführt:

- CPU: Intel® Core™ i7-4790 @3,60 GHz (8 Kerne)
- Arbeitsspeicher: 32 GB DDR4-RAM
- Grafikkarte: NVIDIA GeForce GTX 970 mit 4 GB RAM

Es wurde keine GPU-Unterstützung mittels CUDA genutzt. Obwohl im Rahmen des Projektes ein GPU-Cluster zur Verfügung stand, wurden alle Berechnungen zur Übertragbarkeit auf die unternehmensspezifische IT-Infrastruktur lokal ausgeführt.

5.5.2 Ergebnisse der Optimierung

Nachfolgend werden die Ergebnisse der datengetriebenen Optimierung vorgestellt. Als Verfahren wurden Nelder-Mead, Partikelschwarmoptimierung und genetische Algorithmen genutzt. Es wird auf die Qualität der ermittelten Lösung, die Konvergenzeigenschaften des jeweiligen Verfahrens und ihre Laufzeit eingegangen.

Optimierung nach Nelder-Mead

Zunächst wird die Downhill-Simplex-Optimierung nach Nelder-Mead im Rahmen des Scipy-Pakets in Python genutzt. Zur Betrachtung der Topographie der Kostenfunktion und der Konvergenzeigenschaften des Verfahrens wird der Optimierungsprozess anhand einer zweidimensionalen Problemstellung demonstriert. Hierfür werden beispielhaft die beiden wichtigsten steuerbaren Einflussgrößen der Wandstärke am Spitzende systematisch variiert und der Verlauf der besten Parameterkombination in jeder Stufe des Algorithmus visualisiert. Als Referenz dient ein Raster aller Kostenfunktionswerte innerhalb von \mathcal{F} . Hierzu wurden mit einer Exhaustive Search 100 Variationen jeder Steuergröße in einem linearen Abstand zueinander gebildet.

Abbildung 5.28 zeigt das Ergebnis der zweidimensionalen Optimierung. Die 10.000-fache Berechnung der Kostenfunktion benötigt 25,8 Sekunden. Es lassen sich zwei lokale Minima im unteren rechten Bereich finden. Das Rechte ist am kleinsten mit einem Funktionswert von 1,1949. Weiterhin sind die Optimierungspfade von Nelder-Mead und der Powell-Variante einschließlich ihrer Start- und Endpunkte eingezeichnet. Dabei wurde jede

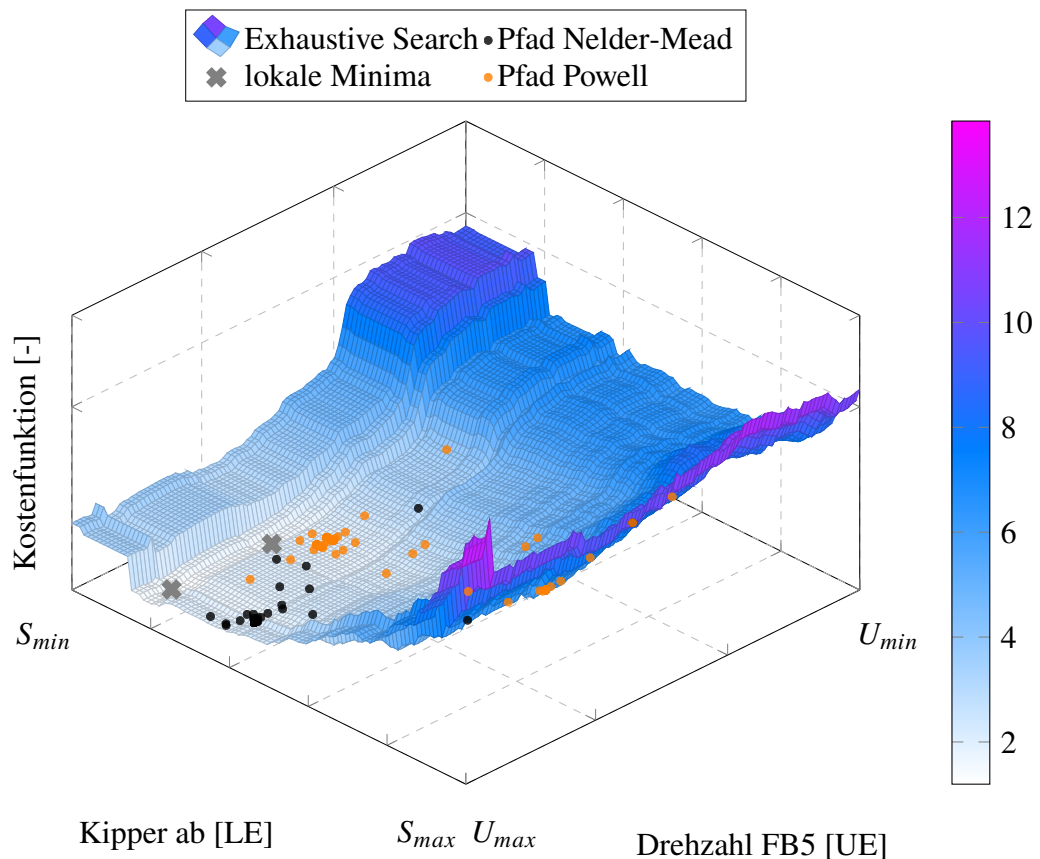


Abbildung 5.28: Optimierungstopographie für Nelder-Mead. Es wurden nur zwei Prozessgrößen variiert, um das Konvergenzverhalten zu untersuchen. Abbildung aus Gründen der Anschaulichkeit gedreht.

Parametervariante berücksichtigt, die anhand der Kostenfunktion ausgewertet wurde, da die einzelnen Simplexes zum Zeitpunkt der Erstellung der Arbeit nicht explizit ausgegeben werden konnten. Beide Verfahren wurden an der gleichen Position initialisiert. Die Nelder-Mead-Optimierung benötigt 30 Iterationen und insgesamt 86 Funktionsauswertungen. Sie findet das linke lokale Minimum und ermittelt einen Funktionswert von 1,6767 innerhalb von 1,2 Sekunden. Zwischenzeitlich kommt es zu Überschreitungen des gültigen Wertebereichs, da in dieser Implementierung des Verfahrens keine Beschränkungen angegeben werden können. Die Powell-Methode benötigt insgesamt 150 Funktionsauswertungen und überschreitet die in \mathcal{F} definierten Grenzen trotz Vorgabe mehrmals. Das Verfahren konvergiert in Richtung des Minimums, endet jedoch bei einem Wert der Kostenfunktion von 1,3730 nach einer Rechenzeit von 1,7 Sekunden.

Es sei angemerkt, dass die Konvergenzeigenschaften und das gefundene Optimum teils stark mit der Startposition variiert. Das gezeigte Beispiel verdeutlicht die Problematik anhand zweier lokaler Minima im \mathbb{R}^2 . Für höherdimensionale Optimierungen lässt sie sich nicht mehr geeignet veranschaulichen. Die Konvergenz des Algorithmus zu lokalen Nebenoptima ist wohlbekannt. Es existieren Lösungsansätze, die dies teilweise verhindern, sie erfordern jedoch eine detaillierte Kenntnis der Optimierungstopographie, die für jedes Rohr neu zu bestimmen wäre.

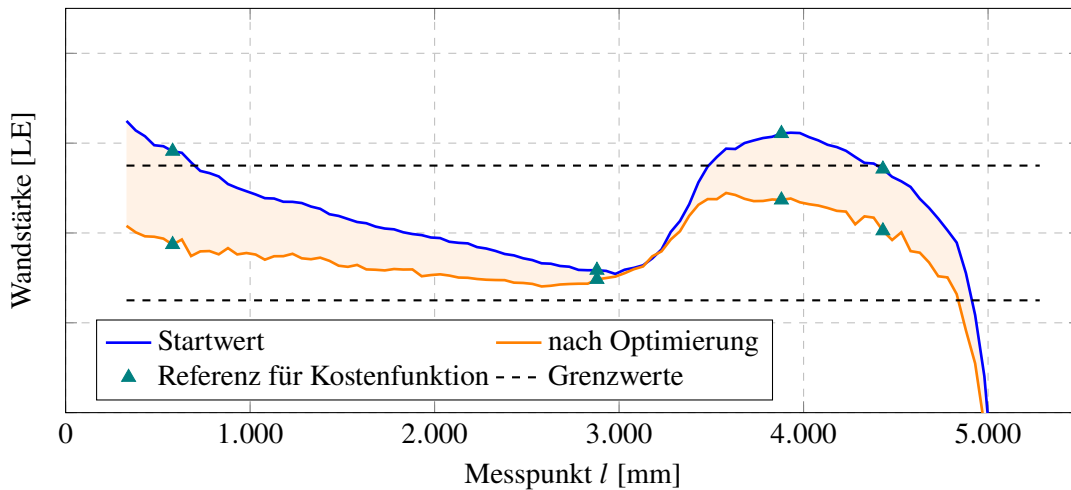


Abbildung 5.29: Optimierungsergebnis für Nelder-Mead bei der Variation aller Steuerungsgrößen.

Es werden nun alle 15 Steuerungsgrößen für die Optimierung berücksichtigt. Die Kostenfunktion bewertet den Wandstärkenverlauf weiterhin anhand von vier automatisch ausgewählten Messpunkten. Abbildung 5.29 zeigt das Ergebnis der Nelder-Mead-Optimierung für ein ausgewähltes Rohr. Sie benötigt 481 Funktionsauswertungen bei 118 Iterationen und einer Rechenzeit von 2,5 Sekunden. In diesem Beispiel können alle Wandstärken in einen unkritischen Bereich überführt werden, der vorher von zwei Referenzpunkten geringfügig überschritten wurde. Die Reduktion wird unter anderem durch ein früheres Zurückfahren des Kippers und eine Anpassung der Bewegung vor dem Gießvorgang realisiert. Sie führt weiterhin zu einer geringeren vergossenen Gesamteisenmenge, die in den Daten jedoch nicht mehr abgebildet ist, da sie erst nach dem Gießvorgang gemessen wird.

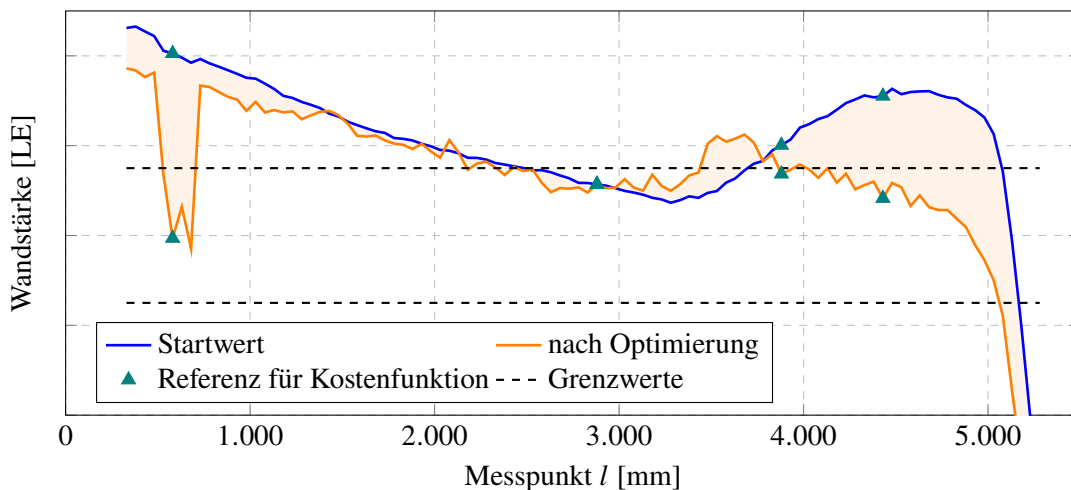


Abbildung 5.30: Optimierungsergebnis für Nelder-Mead bei der Variation aller Steuerungsgrößen ohne Verteilungskriterien in der Kostenfunktion.

Abbildung 5.30 zeigt anhand eines Beispiels eine physikalisch unplausible Prozessvariante als Optimierungsergebnis. Es wurde mit dem Nelder-Mead-Verfahren und ohne die

Einbeziehung der Verteilungskriterien $\zeta(\mathbf{x}, \mathbf{X})$ und $Y(\mathbf{y}, \mathbf{Y})$ in der Kostenfunktion ermittelt. Eine Anomalieerkennung wurde somit weder für die synthetisierten Prozessvarianten noch für die generierten Wandstärkenverläufe durchgeführt. Hinsichtlich der formellen Optimierungskriterien, nämlich des ℓ_2 -Abstandes zum Sollwert und der Überschreitung der maximalen Wandstärke, kann eine Verbesserung erreicht werden. Während die Referenzpunkte in den unkritischen Bereich überführt werden, verschiebt sich das Wanddickenmaximum nach links, sodass es durch die Rekonstruktion des Verlaufes nicht mehr vollständig erfasst wird.

Bei Betrachtung des Profils selbst lässt sich zunächst ein erhöhtes Grundrauschen bei den Wanddickenprognosen entlang des gesamten Rohres erkennen. Die vorliegende Prozessvariante liegt so weit außerhalb des gewohnten Arbeitsbereiches, dass die Einzelmodelle mit ihrer ohnehin begrenzten Interpolationsfähigkeit sehr unterschiedlich darauf reagieren. Die Charakteristik des Rauschens deutet auf einen stochastischen Effekt hin. Weiterhin erfolgt im unmittelbaren Bereich hinter der Muffe ein starker lokaler Abfall der Wanddicke. Die Platzierung des Referenzpunktes der Kostenfunktion an dieser Position resultiert in einer Fehlinterpretation und ist höchstwahrscheinlich der Grund für die Erfüllung der Optimierungskriterien. Als begründete Annahme hierfür dient die Neigung des Nelder-Mead-Verfahrens, in lokalen Minima zu verharren, die von hohen Steigungen umgeben sind (Kelley 1999). Eine erhöhte Prognoseunsicherheit in diesem Bereich des Rohrs konnte bereits in Abschnitt 5.3.3 attestiert werden, siehe auch Abbildung 5.29. Durch die Nutzung der Q-Statistik wäre diese Prozessinstanz im Rahmen der Datenvorverarbeitung bereits herausgefiltert worden.

Partikelschwarmoptimierung

Die Partikelschwarmoptimierung erfolgt mit Hilfe der PySwarms-Umgebung (Miranda 2018) in Python. Der größte Unterschied in der Implementierung liegt darin, dass sich keine expliziten Startwerte für den Algorithmus vorgeben lassen. Die festgeschriebene Zahl der Partikel wird zufällig innerhalb von \mathcal{F} platziert und konvergiert anschließend nach den vorgegebenen Kriterien. Somit unterliegt die Konvergenz des Verfahrens einer weiteren Zufallskomponente. Für diese Arbeit wurde die globale Optimierungsstrategie gewählt, bei der jeder Partikel seine Performance mit dem gesamten Schwarm vergleicht. Die Optimierungssequenz wird durch die Zahl der Partikel und die der maximal durchzuführenden Iterationen bestimmt, während die Bewegung der Parameter durch die Gewichtungen der kognitiven Komponente c_1 , der sozialen Komponente c_2 und des Trägheitsterms w beeinflusst werden kann.

Das Konvergenzverhalten wird anhand des zweidimensionalen Beispiels aus dem letzten Abschnitt demonstriert. Die Rückfahrposition des Kippers und die Drehzahl im fünften Fahrbereich werden systematisch variiert, sodass die gleiche Optimierungstopographie wie in Abbildung 5.28 entsteht. Mit den in der Dokumentation angegebenen Standardeinstellungen $c_1 = 0,5$, $c_2 = 0,3$ und $w = 0,9$ ist bereits nach ca. 10 Iterationen ein starker Abfall der Kostenfunktion zu verzeichnen. Dies lässt sich mit dem begrenzten Variationsbereich der Eingangsgrößen erklären, der mit einer hohen Wahrscheinlichkeit dazu führt, dass einer der 100 eingesetzten Partikel bei der Initialisierung in unmittelbarer Nähe zum Optimum platziert wird. Für das vorliegende Beispiel wurde eine Random Search zur Bestimmung der optimalen Hyperparameter durchgeführt. Eine Konvergenz des Schwarms

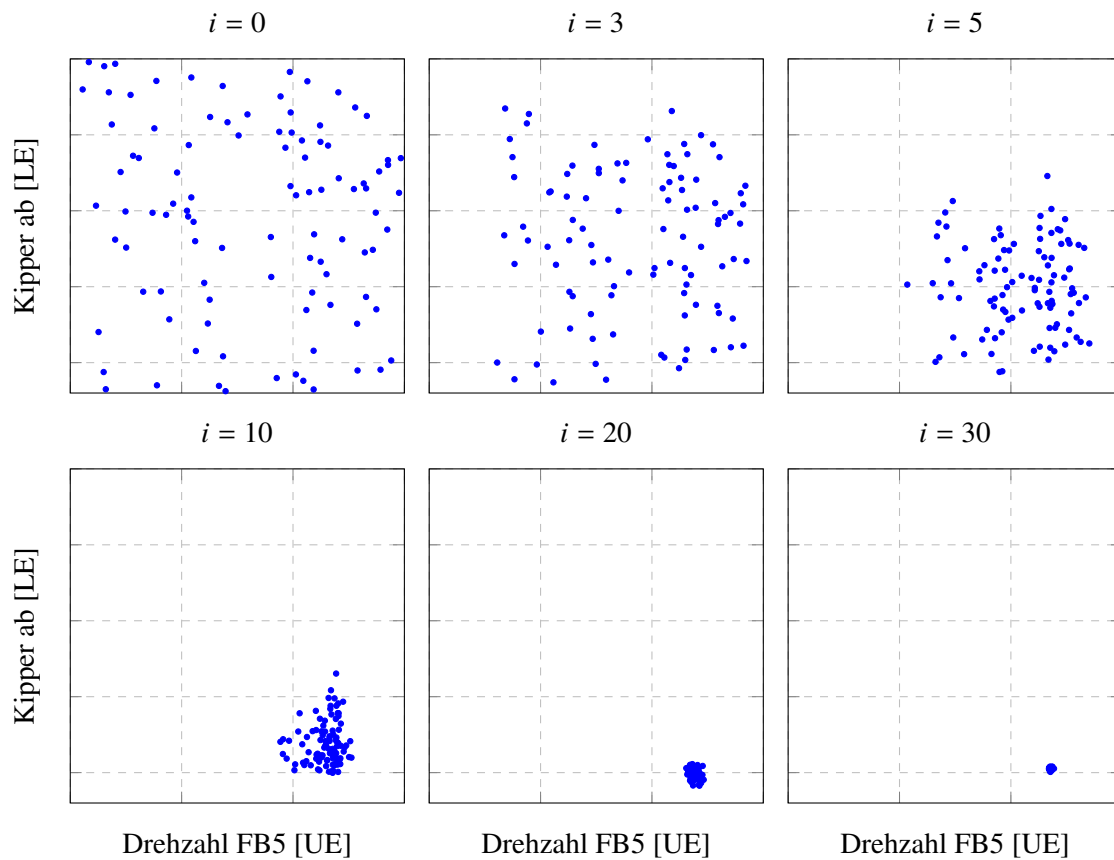


Abbildung 5.31: Konvergenz der Partikelschwarmoptimierung bei der Variation zweier Steuerungsgrößen. Es wird die Position des Schwarms zum Iterationsschritt i gezeigt.

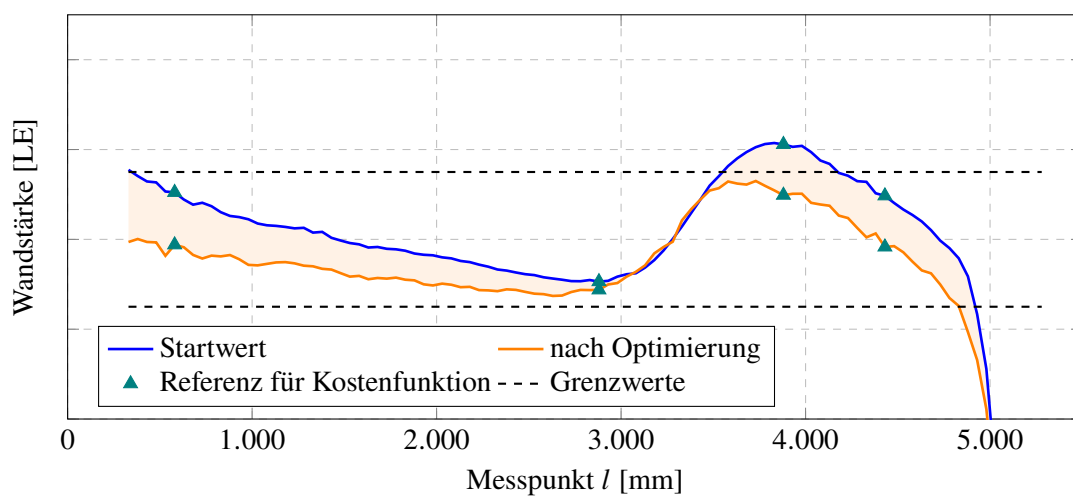


Abbildung 5.32: Optimierungsergebnis des Partikelschwarmalgorithmus.

kann mit $c_1 = c_2 = 0,2$ und $w = 0,5$ erreicht werden. Abbildung 5.31 zeigt das Ergebnis für ausgewählte Iterationsschritte. Eine deutliche Absenkung der kognitiven Komponente und der Trägheit gewährleistet, dass die Bewegung zum Optimum die der Eigendynamik übersteigt. Auf diese Weise kann die Konvergenz reproduzierbar sichergestellt werden.

Abbildung 5.32 zeigt das Ergebnis der Partikelschwarmoptimierung für die Variation aller Steuerungsgrößen. Es wurden die für das zweidimensionale Problem optimierten Hyperparameter verwendet. Eine Überführung der Wandstärken in den unkritischen Bereich ist auch hier gewährleistet. Aufgrund der ermittelten Konvergenzeigenschaften wurde die Zahl der maximalen Iterationen von 100 auf 30 herabgesetzt. Die Optimierung benötigt damit 12,5 Sekunden.

Optimierung mit genetischen Algorithmen

Für die Nutzung genetischer Algorithmen wird auf die Python-Bibliothek PyGAD (Gad 2021) zurückgegriffen. Sie unterstützt verschiedene Implementierungen der Kreuzung, der Mutation und der Replikation. Die Optimierung erfolgt anhand einer Fitnessfunktion f_f , die es zu maximieren gilt. Aus diesem Grund wird die Kostenfunktion J zu

$$f_f(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{x}_0) = \frac{1}{J(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{x}_0)} \quad (5.21)$$

umformuliert. Das Verfahren wird zunächst mit einer Populationsgröße von 100 und 100 Generationen gestartet. Folgende Hyperparameter werden weiterhin festgelegt:

- Mutationswahrscheinlichkeit = 0,1: Wahrscheinlichkeit, dass das Gen (Feature) einer Lösung durch einen zufälligen Wert ersetzt wird.
- Elitenanteil = 0,01: Aufgrund der Populationsgröße wird eine Elite (entspricht 1 %) in jeder Generation übernommen.
- Kreuzungswahrscheinlichkeit = 0,5. Wahrscheinlichkeit, ein Individuum für die Kreuzung auszuwählen.
- Elternanteil = 0,3: Teil der Population, der mittels Replikation gefüllt wird.

Die Rand- und Startbedingungen werden analog zu den anderen Optimierungsverfahren gewählt.

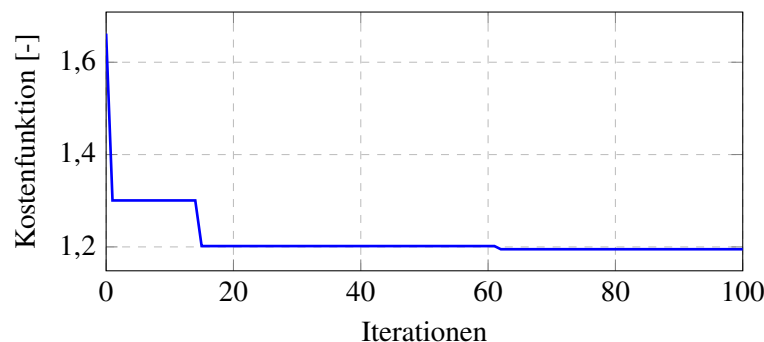


Abbildung 5.33: Verlauf der Kostenfunktion des genetischen Algorithmus.

Abbildung 5.33 zeigt den Verlauf der Kostenfunktion für 100 Berechnungsschritte. Sie wurde aus Gründen der Anschaulichkeit aus den Werten der Fitnessfunktion bestimmt. Das Verfahren benötigt dafür eine Gesamtzeit von 29,8 Sekunden, während nach 15 Iterationen jedoch keine merkliche Verbesserung mehr festzustellen ist. Das endgültige Minimum wird in der 62. Generation gefunden. In nahezu allen untersuchten Fällen konnte die Konvergenz nach weniger als 30 Iterationen und einer Berechnungsdauer von 9,9 Sekunden sichergestellt werden.

Bewertung der Optimierungsverfahren

Zur abschließenden Bewertung der Optimierungsperformance der in diesem Abschnitt betrachteten Verfahren werden sie auf eine Stichprobe der Lerndatenmenge angewendet, die aus $N = 500$ zufällig ausgewählten Rohren besteht. Es wird jeweils untersucht, inwieweit eine Verbesserung des Wandstärkenverlaufs im Bereich der Muffe, am Schaft und am Spitzende des Rohrs erzielt werden kann. Hierfür werden die Messpunkte $l_1 = 580$ mm, $l_2 = 2080$ mm und $l_3 = 3880$ mm herangezogen.

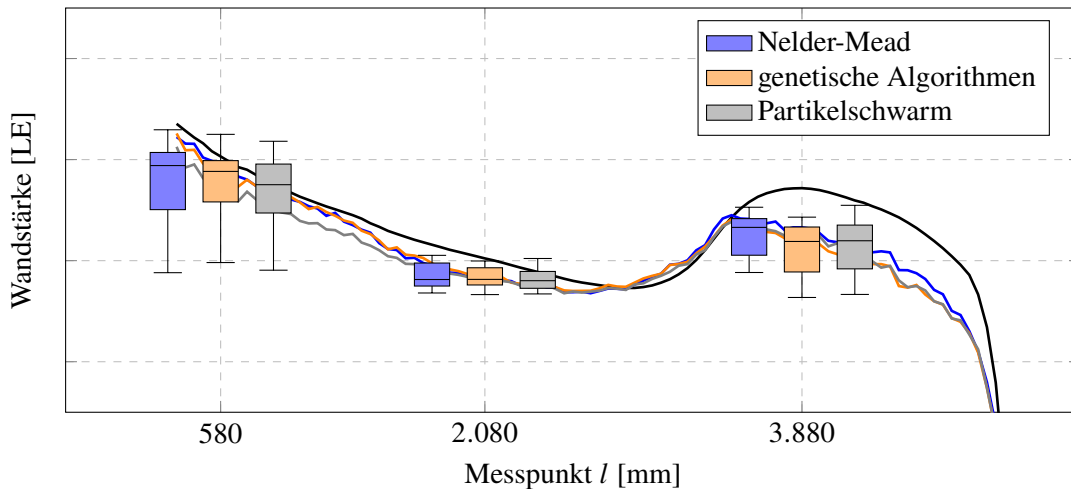


Abbildung 5.34: Vergleich der Optimierungsperformance der drei genutzten Verfahren. Dargestellt sind die durch die Stichprobe ermittelten Verteilungseigenschaften der Wanddickenreduktion an drei Messpunkten.

Für jede Prozessinstanz $\mathbf{x}^{(i)}$, $i = 1, \dots, N$ wird die mit der Prognosefunktion f_j vorhergesagte Wandstärke $f_j(\mathbf{x}^{(i)})$, $j = 1, \dots, d$ mit der der optimierten Prozessvariante $f_j(\mathbf{x}_{opt}^{(i)})$ verglichen. Die Matrix der Wandstärkendifferenzen

$$\begin{bmatrix} \Delta \hat{\mathbf{y}}^{(1)} \\ \vdots \\ \Delta \hat{\mathbf{y}}^{(N)} \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}_{opt}^{(1)}) - f_1(\mathbf{x}^{(1)}) & \dots & f_d(\mathbf{x}_{opt}^{(1)}) - f_d(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ f_1(\mathbf{x}_{opt}^{(N)}) - f_1(\mathbf{x}^{(N)}) & \dots & f_d(\mathbf{x}_{opt}^{(N)}) - f_d(\mathbf{x}^{(N)}) \end{bmatrix} \quad (5.22)$$

wird schließlich für die ausgewählten Messpunkte auf ihre Streuungsmaße untersucht. Der quantitative Vergleich der Optimierungsperformance erfolgt somit zunächst nur hinsichtlich des prognostizierten Wandstärkenprofils. So lässt sich die Handhabungsfähigkeit der

gradientenfreien Verfahren bezüglich der durch die Kostenfunktion entstehenden Optimierungstopographie isoliert betrachten.

Abbildung 5.34 zeigt die kumulierten Optimierungsergebnisse der Stichprobe als Boxplots. Die Streuungseigenschaften der Wandstärkenveränderung wurden an den gekennzeichneten Messpunkten aus Gründen der Anschaulichkeit in Dreiergruppen nebeneinander aufgetragen, sie beziehen sich jedoch alle auf die gleiche Position l_i . Das im Hintergrund dargestellte Profil ist der Mittelwert der Stichprobe und dient als Referenz. Im Bereich hinter der Muffe bei $l_1 = 580$ mm bewirkt die Partikelschwarmoptimierung die höchste mittlere Wanddickenreduktion, während das Nelder-Mead-Verfahren die breiteste Streuung aufweist. Die sehr tief gelegenen unteren Whisker können auf die Entfernung zum Sollwert zurückgeführt werden. In wenigen Fällen, zu erkennen am oberen Whisker, erfolgt sogar ein Anstieg der Wanddicke. Ein Beispiel einer solchen Variante wird im Rahmen der ℓ_0 -Regularisierung im nächsten Abschnitt gezeigt. Für die Position $l_2 = 2080$ mm sind allenfalls minimale Unterschiede zwischen den drei Verfahren zu erkennen. Für die Position $l_3 = 3880$ mm am Spitzende produziert der genetische Algorithmus das beste Resultat, wenngleich die Abweichung zur PSO gering ausfällt. Lediglich Nelder-Mead erzielt ein merklich schlechteres Ergebnis.

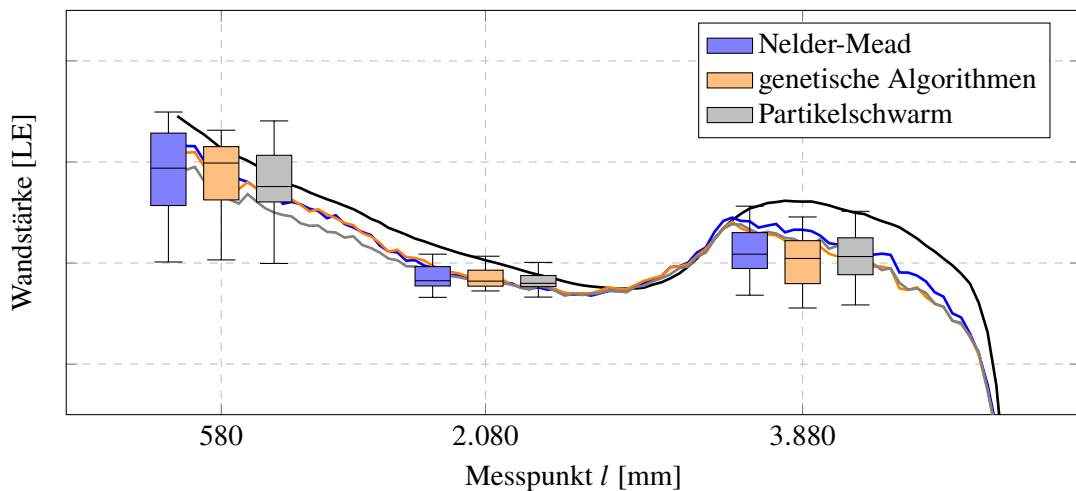


Abbildung 5.35: Vergleich der Optimierungsergebnisse der drei genutzten Verfahren im Vergleich mit den wahren Werten von \mathbf{Y} .

Es sei dabei angemerkt, dass die Differenzen in der Performance die Prognoseunsicherheit des Modells noch nicht berücksichtigen und letztere für $l_3 = 3880$ mm deutlich höher ausfällt. Dem steht jedoch eine deutlich geringere benötigte Rechenzeit bei Nelder-Mead gegenüber. Um die Prognoseabweichung einzubeziehen, erfolgt der Vergleich der Optimierungslösung stattdessen mit den tatsächlichen Werten $\mathbf{y}^{(i)}$. Die Matrix der Wandstärkendifferenzen wird damit zu

$$\begin{bmatrix} \Delta \mathbf{y}^{(1)} \\ \vdots \\ \Delta \mathbf{y}^{(N)} \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}_{opt}^{(1)}) - y_1^{(1)} & \cdots & f_d(\mathbf{x}_{opt}^{(1)}) - y_d^{(1)} \\ \vdots & \ddots & \vdots \\ f_1(\mathbf{x}_{opt}^{(N)}) - y_1^{(N)} & \cdots & f_d(\mathbf{x}_{opt}^{(N)}) - y_d^{(N)} \end{bmatrix} \quad (5.23)$$

und kann auf die gleiche Weise analysiert werden.

Abbildung 5.35 zeigt die Boxplots zu Gleichung (5.23). Bei l_3 ist eine Verschiebung der Mediane und der Quartile nach unten zu erkennen. Dies suggeriert eine Verbesserung der Optimierungsperformance. Bei l_1 hingegen bleiben die Mediane weitestgehend unverändert, während die oberen Quartile ansteigen. Beide Phänomene lassen sich teilweise mit der Prognoseunsicherheit der Modelle erklären. So tendieren diese dazu, am Spitzende besonders hohe Werte zu niedrig vorherzusagen, was die Wanddickenreduktion insgesamt ansteigen lässt. Bei l_1 erfährt die Prognose einen gegenteiligen Effekt: insbesondere kleine Wandstärken werden zu hoch geschätzt.

5.5.3 Generierung von Handlungsempfehlungen

Die Handlungsempfehlungen zur Prozessverbesserung werden mittels einer Optimierungslösung berechnet. In diesem Abschnitt werden mehrere Ansätze erprobt, die Charakteristik des Prozesseingriffs durch Anpassungen in der Kostenfunktion aktiv zu steuern. Die dafür genutzte Datenbasis wurde gesondert bereitgestellt und enthält anonymisierte Steuerungsgrößen. Zum Zweck der Vergleichbarkeit werden die Handlungsempfehlungen an einem Rohr erprobt.

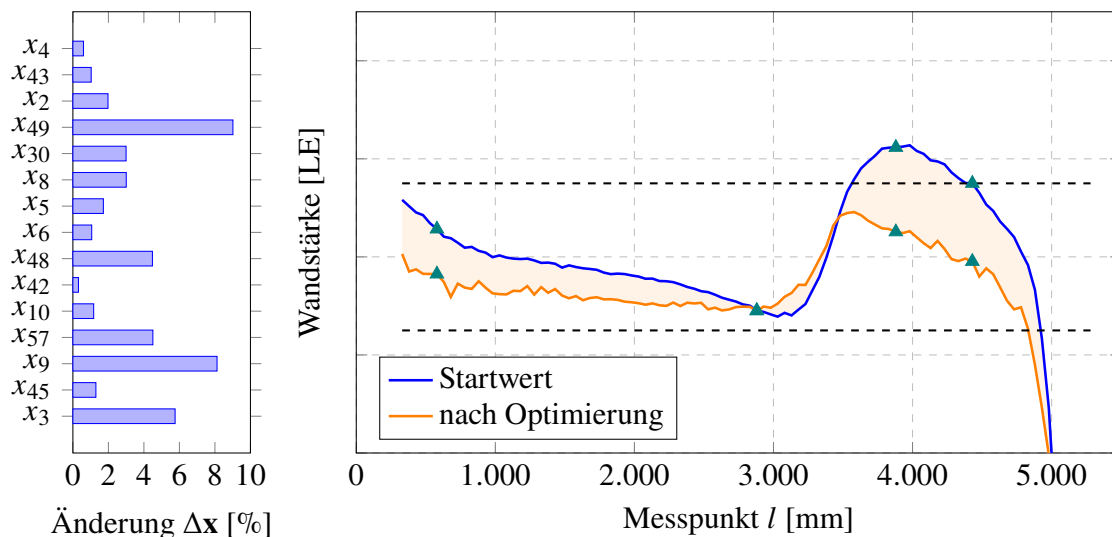


Abbildung 5.36: Handlungsempfehlung ohne l_k -Beschränkung der generierten Lösung.

Der Strafterm

$$\Xi(\mathbf{x} - \mathbf{x}_0) = \|\boldsymbol{\vartheta} \odot (\mathbf{x} - \mathbf{x}_0)\|_k = \sqrt[k]{\sum_{i=1}^n (\vartheta_i (x_i - x_{0,i}))^k} \quad (5.24)$$

wird mit unterschiedlichen l_k -Normen der Kostenfunktion hinzugegeben. Der Skalierungsvektor

$$\boldsymbol{\vartheta} = \left[\frac{1}{x_1} \quad \dots \quad \frac{1}{x_n} \right] \quad (5.25)$$

enthält die Mittelwerte der zu variierenden Steuerungsgrößen. Er bewirkt eine Normierung mittels elementweiser Multiplikation. Durch die weiteren Beschränkungen sind Kompromisse beim Prozessergebnis und Laufzeitveränderungen durch eine andere Topographie der Optimierung zu erwarten.

Die Auswirkungen der unterschiedlichen Kostenfunktionen werden anhand eines beispielhaft ausgewählten Rohres als Referenz demonstriert. Abbildung 5.36 zeigt seinen optimierten Wandstärkenverlauf. Es wurden neben den Verteilungskriterien von \mathbf{x} keine direkten Beschränkungen der Handlungsempfehlung festgelegt. Die Veränderungen der Prozessparameter werden zum Zweck der visuellen Darstellung als Absolutwerte der prozentualen Abweichungen zum Startwert

$$\Delta x_i = \frac{x_i - x_{i,0}}{x_i} \quad (5.26)$$

dargestellt. Es werden die bereits bekannten Verbesserungen im Wandstärkenprofil erzielt. Hierfür sind alle Steuerungsgrößen zu variieren. Die Rechenzeit beträgt 12,5 Sekunden.

Handlungsempfehlungen mit ℓ_2 -Regularisierung

Zur Generierung ℓ_2 -optimaler Handlungsempfehlungen wird ein Strafterm

$$\Xi(\mathbf{x} - \mathbf{x}_0) = \|\boldsymbol{\vartheta} \odot (\mathbf{x} - \mathbf{x}_0)\|_2 \quad (5.27)$$

hinzugenommen. Er gewährt bei einer korrekten Gewichtung innerhalb der Kostenfunktion näherungsweise eine Gleichverteilung der Parameteränderungen und verhindert, dass einzelne Stellgrößen zu stark angepasst werden müssen. Die Optimierungstopographie ändert sich insoweit, als J mit dem euklidischen Abstand der Lösung vom Ausgangspunkt linear anwächst. Es ist weiterhin ein geringerer zusätzlicher Rechenaufwand als mit den anderen Korrekturfaktoren zu erwarten, da der ℓ_2 -Term selbst konvex optimiert werden kann. Da die Gesamtheit der Lösungsvariablen mit der ℓ_2 -Regularisierung klein gehalten, jedoch nicht zu null gesetzt werden soll, eignet sie sich für eine automatische Empfehlungsverarbeitung. Neben der verringerten Rechenzeit ist der zusätzliche Aufwand bei der manuellen Anpassung zu berücksichtigen.

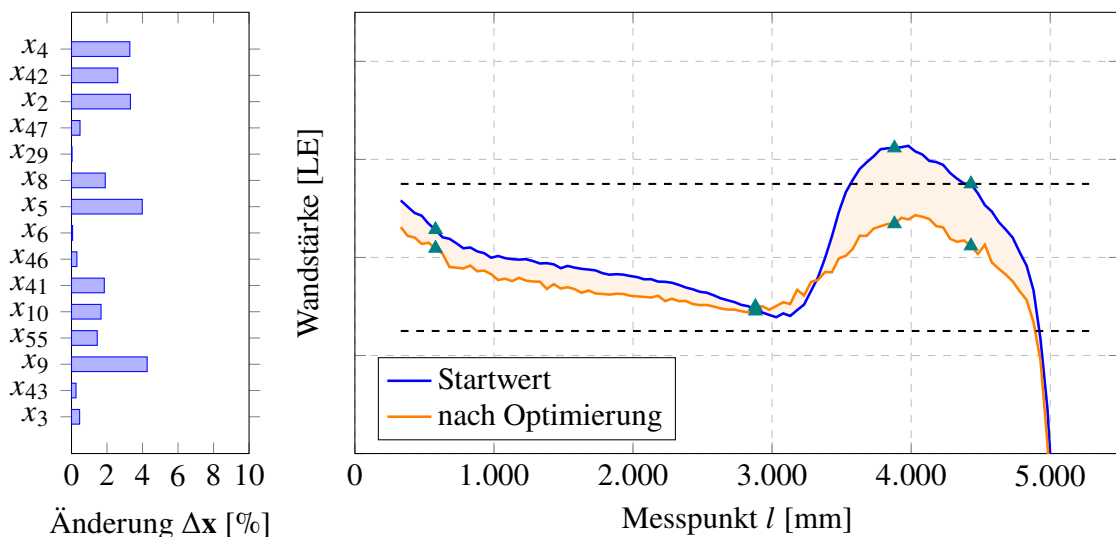


Abbildung 5.37: Handlungsempfehlung mit ℓ_2 -Regularisierung. Die gezeigten Parameteränderungen sind teils klein, jedoch nicht Null.

Abbildung 5.37 zeigt die Auswirkung der ℓ_2 -Regularisierung bei der Wandstärkenoptimierung des gleichen Rohrs. Es wurde $\alpha_5 = 0,12$ gewählt. Die relativen Änderungen der

Prozessparameter sind unter dieser Einstellung bei ca. 4 % begrenzt. Erwartungsgemäß sinkt der Einfluss einzelner Steuerungsgrößen teils sehr stark, er ist jedoch nie gleich null. Durch die Vielzahl der Parameteranpassungen kann in nahezu allen Wandstärkenbereichen eine Verbesserung erzielt werden, wenngleich sie geringer als die der uneingeschränkten Optimierung ausfallen. Insbesondere die Steuerungsgrößen zur Wandstärkenkorrektur im Bereich der Muffe werden hierbei stärker vernachlässigt. Die Rechenlaufzeit verringert sich auf 12,2 Sekunden, unterliegt aufgrund der zufälligen Initialisierung des Schwarms jedoch Schwankungen.

Handlungsempfehlungen mit ℓ_1 -Regularisierung

Die ℓ_1 -Regularisierung der Handlungsempfehlung wird durch

$$\Xi(\mathbf{x} - \mathbf{x}_0) = \|\boldsymbol{\vartheta} \odot (\mathbf{x} - \mathbf{x}_0)\|_1 \quad (5.28)$$

realisiert. Sie bestreitet den Mittelweg zwischen dünnbesetzten und gleichmäßig verteilten Parameteränderungen. Die bei der LASSO-Regression gewährte Robustheit des ℓ_1 -Strafterms, die bei Ausreißern Abhilfe schafft, wirkt bei der Optimierung insoweit, als stärkere Variationen einzelner Parameter toleriert werden. Es ist darüber hinaus ein höherer Rechenaufwand zu erwarten, da die zugegebenen Beschränkungen in der Optimierungstopographie nicht differenzierbar sind. ℓ_1 -optimale Lösungen tendieren weiterhin dazu, dünner besetzt zu sein, also weniger von Null verschiedene Einträge zu besitzen. Im Gegensatz zur ℓ_2 -Optimierung wird der Einfluss unwichtiger Features für die Handlungsempfehlung vollständig herausgenommen und nicht nur verringert. Der Strafterm fungiert somit auch als Featureselektion.

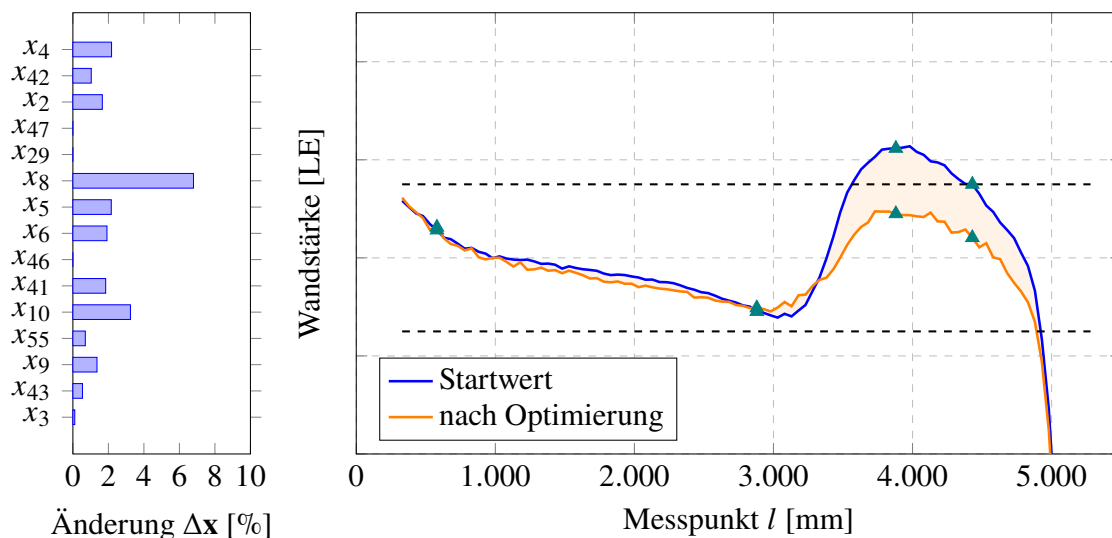


Abbildung 5.38: Handlungsempfehlung mit ℓ_1 -Regularisierung. Der Einfluss dreier Parameter wird vollständig abgeschaltet. Rechenzeit: 11,6 s.

Abbildung 5.38 zeigt das Ergebnis der Optimierung mit $\alpha_5 = 0,1$. Die Auswirkungen des ℓ_1 -Strafterms lassen sich in der generierten Handlungsempfehlung erkennen. Drei Variablen werden bei der Lösungsfindung nicht mehr berücksichtigt. Der Einfluss anderer Steuerungsgrößen fällt geringer aus als bei der ℓ_2 -optimalen Prozessvariante. Die einzige

Ausnahme bildet x_8 : Sie wird um ca. 7 % erhöht. Der zugehörige Wandstärkenverlauf lässt sich hiermit nur noch im Bereich des Spitzendes verbessern. Ihm wird durch diese Variante eine höhere Bedeutung beigemessen, da die anderen Referenzpunkte bereits hinreichend nah am Sollwert liegen. Die Rechenzeit beträgt 11,6 Sekunden und fällt damit geringer aus als die der Optimierung ohne Beschränkung.

Dünnbesetzte Handlungsempfehlungen

Schlussendlich erfolgt die gezielte Generierung dünnbesetzter Handlungsempfehlungen mit Hilfe der Pseudonorm. Der Strafterm

$$\Xi(\mathbf{x} - \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_0 \quad (5.29)$$

beschreibt die Anzahl von Null verschiedener Elemente des Differenzvektors. Die Implementierung einer derartigen Kostenfunktion führt zu einer nicht konvexen Optimierung und zu langen Rechenzeiten, wie bereits bei der RPCA in Abschnitt 2.4.4 gezeigt wurde. Für ihre Minimierung liegen keine verlässlichen Lösungsansätze vor, da sie kombinatorisch schwer ist. Aus diesem Grund wird sie in der Praxis häufig mit der ℓ_1 -Norm substituirt.

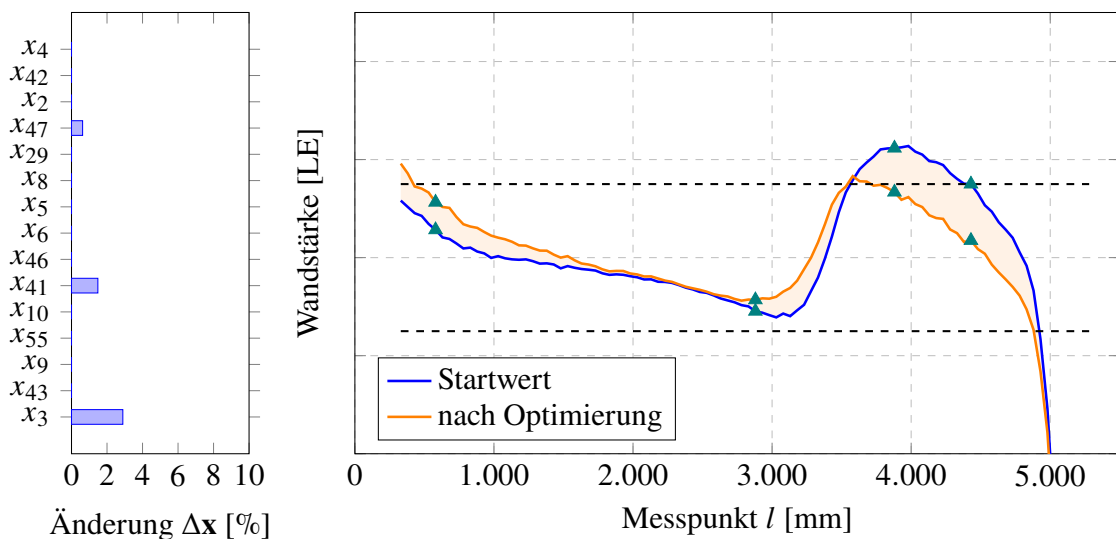


Abbildung 5.39: Handlungsempfehlung mit ℓ_0 -Regularisierung. Die Pseudonorm bewirkt, dass nur noch drei Prozessparameter für die Optimierung genutzt werden.

Abbildung 5.39 zeigt das Ergebnis der Optimierung. Es verbleiben drei Steuerungsgrößen, die im Vergleich zu den anderen ℓ_k -Regularisierungen nur geringfügig angepasst werden. Eine Reduktion der Wandstärke am Spitzende wird auf Kosten jener an der Muffe erzielt. Das Ergebnis verdeutlicht die Problematik bei der Optimierung konträrer Zielgrößen. Durch die Beschränkung der Zahl veränderbarer Steuerungsgrößen können die wechselseitigen Effekte nicht mehr ausgeglichen werden. Obwohl sich das Wanddickenmaximum verschiebt, können die Werte weitestgehend in den unkritischen Bereich überführt werden. Die dargestellten Ergebnisse wurden mit $\alpha_5 = 0,17$ erzeugt. Der Effekt des ℓ_0 -Strafterms stellt sich jedoch sprunghaft ein, sodass sich ab einem Schwellwert von $\alpha_5 \approx$

0,165 keine signifikanten Änderungen in der Prozessvariante erkennen lassen. Für kleinere Gewichtungen von Ξ ähneln die Lösungen jenen der unbeschränkten Optimierung.

Bewertung der Handlungsempfehlungen

Es konnte bislang gezeigt werden, dass eine zusätzliche Beschränkung der Kostenfunktion zur Steuerung der Handlungsempfehlung zu einem schlechteren Prozessergebnis und zusätzlichem Rechenaufwand führt. Die Bewertung der ℓ_k -Regularisierung erfolgt abschließend anhand der Veränderungen in der Optimierungsperformance. Hierfür wird die Zunahme der Kostenfunktion durch den ℓ_k -Strafterm für 30 Variationen des Gewichtungsfaktors α_5 untersucht.

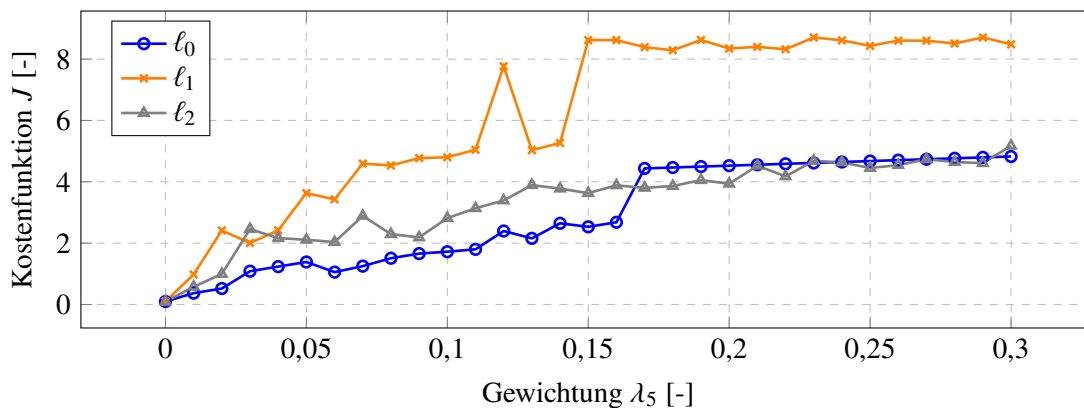


Abbildung 5.40: Auswirkung der ℓ_k -Regularisierung auf die Kostenfunktion für unterschiedliche Gewichtungen α_5 .

Abbildung 5.40 zeigt die Veränderung der Kostenfunktion für Werte von α_5 von 0 bis 0,3 mit einer Schrittweite von 0,01. Sie beziehen sich auf das gleiche bisher betrachtete Rohr. Für alle drei Normen lässt sich zunächst ein nahezu linearer Anstieg der Kostenfunktion erkennen. Ab ca. $\alpha_5 = 0,15$ stellt sich für die ℓ_1 -Norm ein Sättigungseffekt ein, sodass keine systematischen Veränderungen mehr erkennbar sind. Sie hat weiterhin die größte Auswirkung auf die Verschlechterung des Optimierungsergebnisses. Lokale Schwankungen im Verlauf können auf Zufallseffekte zurückgeführt werden und sind durch Spezifika in der Topographie zu begründen. Dahingegen erzeugen die ℓ_0 - und ℓ_2 -Strafterme nur noch einen geringen Zuwachs, da sie das gefundene Optimum nicht mehr modifizieren, sondern lediglich den Wert der Kostenfunktion additiv beeinflussen. Insgesamt ist bei der Anwendung der Regularisierung eine erhebliche Zunahme letzterer hinsichtlich des Startwerts von 0,0944 zu erwarten. Die drei bereits beispielhaft gezeigten Prozessvarianten genügen außerdem der in Gleichung (4.29) gestellten Bedingung, wonach sie sich merklich voneinander unterscheiden sollen, sodass sie sich sinnvoll in einem Ranking darstellen lassen.

Eine Übersicht der bislang gezeigten Optimierungsergebnisse ist in Tabelle 5.2 aufgeführt. Ihr liegen die stichprobenhaften Untersuchungen mit $N = 500$ aus der Lerndatenmenge für die drei verwendeten Verfahren zugrunde. Es sind die durchschnittliche Rechenzeit und die Wandstärkenhomogenität $\Gamma(\mathbf{y})$. Letztere wird als Bewertungskriterium für die tatsächliche Produktqualität statt der Kostenfunktion herangezogen. Das beste Ergebnis erzielt die Partikelschwarmoptimierung. Der Wert wird als Referenz genommen

Tabelle 5.2: Übersicht der Optimierungsergebnisse für eine Stichprobe von 500 Rohren.

Verfahren	Zeit [s]	$\Gamma(y_{opt})$ [%]	Regularisierung
Partikelschwarm	10,51	100,00	-
Nelder-Mead	2,32	102,63	-
Genetische Algorithmen	9,75	101,21	-
Partikelschwarm	11,37	119,23	ℓ_0
Partikelschwarm	11,54	114,98	ℓ_1
Partikelschwarm	6,69	112,15	ℓ_2

und die Resultate der anderen Verfahren als prozentuale Abweichung zu ihm angegeben. Die PSO benötigt allerdings die meiste Rechenzeit. Das Nelder-Mead-Verfahren ist mit 2,3 Sekunden am schnellsten und erzielt nur geringfügig schlechtere Ergebnisse.

Unter Berücksichtigung der gewählten Gewichtungssparameter muss bei Nutzung der ℓ_0 -Regularisierung für die Generierung der Handlungsempfehlung die größte Verschlechterung des Wandstärkenverlaufes hingenommen werden. Zwar wirkt sich der Strafterm weniger stark auf die Kostenfunktion aus wie die ℓ_1 -Norm, jedoch genügen die gefundenen Lösungen kaum noch den Prozessvorgaben. Eine ℓ_2 -Regularisierung beschleunigt die Konvergenz der PSO und liefert im Vergleich mit den anderen Beschränkungen die beste Lösung. Eine Reduktion der zu verändernden Steuerungsgrößen kann mit ihr jedoch nicht erreicht werden.

Zusammenfassung

In diesem Kapitel wurden die erarbeiteten Konzepte der datengetriebenen Optimierungskette an einem Beispiel aus der Gießereiindustrie verifiziert. Es erfolgte zunächst eine Einführung in die relevanten gießtechnischen Grundlagen. Anschließend wurden der vorliegende Schleudergießprozess und die für die Analysen genutzte Datenbasis beschrieben. Hierfür dient eine Auswahl von 3.490 Rohren, denen die Prozessparameter und Wandstärkenprofile einzelteilbasiert zugeordnet werden können. Eine effektive Kompensation der prozessspezifischen Störgrößen konnte mit der robusten Hauptkomponentenanalyse realisiert werden. Sie wurde ebenfalls genutzt, um einen optimalen Threshold zur Verdichtung der Daten zu finden und Wandstärkenverläufe bei Sensorausfällen zu rekonstruieren.

Zur Wandstärkenmodellierung dienen Regressionsverfahren, die für alle 100 Messpunkte getrennt trainiert wurden. Bedingt durch die Messbarkeit der Einflussgrößen an der jeweiligen Rohrposition variiert dabei die Prognosegenauigkeit. Die beste Vorhersage liefert der CatBoost-Regressor unter Verwendung der RPCA zur Vorverarbeitung. Letztere verbessert nachweislich die Prognose, mindert Ausreißer und reduziert den Bias. Hinsichtlich der für die Endqualität ausschlaggebenden Zielgrößen konnten die Eisentemperatur und die Kipperbewegung als Haupteinflussgrößen durch empirische Modellinterpretation identifiziert werden.

Im Rahmen der Optimierungsstrategie wurde eine zu minimierende Kostenfunktion entworfen, die den ℓ_2 -Abstand der Wandstärken zum Sollwert und Überschreitungen des Minimal- und Maximalwerts quantitativ erfasst. Weiterhin wurden Verteilungskriterien

für Prozessparameter und Zielgrößen eingeführt, die auf der Q-Statistik beruhen und sicherstellen, dass generierte Prozessvarianten keine Interpolation der Modelle hervorrufen. Die Optimierung erfolgte schließlich mit einem Partikelschwarm, da er zuverlässige Konvergenzeigenschaften vorwies und eine Berechnungszeit von weniger als 15 Sekunden benötigte, die wiederum für die Echtzeitnutzung erforderlich ist. Anpassungen der Kostenfunktion wurden mit ℓ_0 -, ℓ_1 - und ℓ_2 -Regularisierungen erprobt. Sie wirkten sich auf die Charakteristik der Handlungsempfehlung aus, zogen aber auch eine Verschlechterung des Prozessergebnisses mit sich. Teile der Ergebnisse der Fallstudie wurden bereits durch Salentin und Hartmann (2021) veröffentlicht.

Kapitel 6

Zusammenfassung und Ausblick

Dieses Kapitel enthält ein Resümee der gewonnenen Erkenntnisse und einen Ausblick auf zukünftige Entwicklungen.

6.1 Zusammenfassung

Vor dem Hintergrund eines wachsenden globalen Wettbewerbs und der digitalen Transformation müssen fertigende KMU, insbesondere in der Gießereiindustrie, das Potenzial der effektiven Prozessdatennutzung erschließen. Dies birgt Chancen und Herausforderungen zugleich. Das Ziel der vorliegenden Dissertation war die Entwicklung einer modularen datengetriebenen Optimierungskette als methodische Grundlage. Sie umfasst drei anwendungsorientierte Hauptbestandteile: Datenvorverarbeitung und -verdichtung, Modellierung sowie Optimierung. Die Module wurden weitestgehend unabhängig voneinander entwickelt, profitieren aber dennoch voneinander. Sie werden schließlich am Beispiel einer Schleudergießerei prototypisch verifiziert.

Gegenwärtig wird überwachtes maschinelles Lernen im Rahmen von Forschungsprojekten zur Erkennung von Gussfehlern mit Erfolg eingesetzt. Generische Ansätze zur Nutzung der Modelle für die Qualitätsverbesserung existieren jedoch kaum. Eine systematische Optimierung durch virtuelle Versuchsplanung erfolgt für gewöhnlich mit analytischen Simulationen, ist rechenintensiv und kann nur ein vorgegebenes Spektrum von Prozessgrößen abbilden. Fortgeschrittene datengetriebene Modellierungsverfahren finden meist auf Basis künstlich generierter Daten Anwendung.

Die vorgestellte Optimierungskette kombiniert die aufgeführten Ansätze miteinander und überführt sie in ein generalisiertes Anwendungskonzept. Sie bietet zunächst eine thresholdbasierte Störgrößenkompensation unüberwachter Lernverfahren, die kohärente Strukturen aus Prozessdaten extrahiert und externe Einflüsse isoliert. Sie ermöglicht weiterhin eine effiziente Verdichtung der Daten zum Zweck der Extraktion relevanter Features. Hierfür werden die Singulärwertzerlegung, die robuste PCA, die FFT und Denoising-Autoencoder genutzt. Die datengetriebene Modellierung erfolgt mit den standardmäßig eingesetzten Methoden zur Featureselektion und zur Optimierung von Prognosepipelines. Zur empirischen Modellinterpretation werden modellagnostische Verfahren herangezogen, die sich mit jedem überwachten Modell realisieren lassen. Diese umfassen unter anderem SHAP-Werte und die partielle Abhängigkeit zur quantitativen Untersuchung der Prozesseinflussgrößen. Sie dienen der Validierung und der Plausibilitätsprüfung der vom

Modell erlernten Zusammenhänge. Die datengetriebene Optimierung unter Nutzung einer trainierten Prognosefunktion baut, unabhängig vom verwendeten Algorithmus, auf dem Prinzip des virtuellen DoEs auf. Es werden nur die Steuerungsgrößen bei der Optimierung berücksichtigt, die sich zeitgerecht vor Prozessbeginn noch anpassen lassen. Alle anderen Prozessparameter gehen in die Prognosefunktion ein, werden jedoch nicht verändert. Die Optimierung erfolgt bis Abschluss der DoE-Sequenz oder bis keine Verbesserung der Zielgrößen mehr erzielt werden kann. Letztere wird anhand einer fünfteiligen Kostenfunktion quantifiziert. Neben den Kriterien zur Erfüllung des vorgegebenen Prozessziels und der Überschreitung von Nebenbedingungen werden Strafterme definiert, die die Entfernung einer synthetisierten Prozessvariante zur ursprünglichen Verteilung messen und so die Extrapolation der Modelle vermeiden. Darüber hinaus werden ℓ_k -Regularisierungsterme eingeführt, die die Charakteristik der generierten Handlungsempfehlung steuern.

Die prototypische Anwendung der Entwicklungsmethode erfolgt anhand der Daten eines Schleudergießprozesses. Das Ziel der Fallstudie ist die Erzeugung möglichst homogener Wandstärkenprofile durch die Variation der Prozessparameter. Es konnte gezeigt werden, dass sich mit Hilfe der RPCA und SVD prozessspezifische Störungen in der Wanddickenmessung bereinigen lassen, wobei die RPCA aufgrund ihrer ℓ_1 -Optimierung insbesondere starke Ausreißer zuverlässig kompensiert. Die Robustheit der Datenvorverarbeitung wird anhand künstlich generierter Sensorausfälle demonstriert, bei denen sich die Wandstärkendaten erfolgreich rekonstruieren lassen. Eine Reduktion des Datenraumes der abhängigen Variablen wird aufgrund der Kollinearität der Messungen vorgenommen. Zur ℓ_2 -optimalen Rekonstruktion der Wanddickenprofile werden mit der SVD platzierte Messpunkte herangezogen. Die datengetriebene Modellierung des Prozesses erfolgt mit einer Auswahl von 3.490 Rohren auf Grundlage aufgenommener Prozess- und Verfahrensparameter. Mit einem CatBoost-Regressor kann eine maximale Prognosegenauigkeit von bis zu $R^2 = 0,96$ erreicht werden, wobei diese in Abhängigkeit von der Messposition variiert. Die qualitätsrelevanten Einflussgrößen, die mittels empirischer Modellinterpretation identifiziert wurden, umfassen die Temperatur der Schmelze und spezifische Parameter der Kokillenbewegung. Letztere werden als Basis für die Entwicklung der Optimierungsstrategie verwendet. Hierfür wurden insgesamt 15 Verfahrensparameter variiert und die Ergebnisse für das Downhill-Simplex-Verfahren nach Nelder-Mead, die Partikelschwarmoptimierung und genetische Algorithmen verglichen. Den besten Kompromiss aus Optimierungsergebnis und Konvergenzeigenschaften lieferte die PSO. Im Rahmen der Anpassung der generierten Handlungsempfehlungen wurden der Kostenfunktion Strafterme zur ℓ_k -Regularisierung mit unterschiedlichen Gewichtungen hinzugefügt. Die aus der Literatur bekannten Effekte der Regularisierung konnten bei den generierten Prozessvarianten erneut nachgewiesen werden, es sind jedoch teils erhebliche Einschränkungen im Ergebnis der Optimierung hinzunehmen. Eine Optimierung hinsichtlich der ℓ_1 -Norm der Handlungsempfehlung wirkt sich bei gleicher Gewichtung des Strafterms am stärksten auf das Prozessergebnis aus, während die Zahl der zu verändernden Steuerungsgrößen um lediglich drei verringert wird. Bei der ℓ_0 -Regularisierung hingegen verbleiben drei anzupassende Prozessparameter, die in einer unwesentlichen Verbesserung des Wanddickenverlaufes resultieren. Durch keine der durchgeführten Maßnahmen ließ sich der Umfang des Prozesseingriffes direkt steuern.

6.2 Ausblick

Die im Rahmen der Arbeit durchgeführten Untersuchungen stellen ein Proof of Concept dar und beruhen weitestgehend auf Offlineanalysen. Eine Verifizierung der trainierten Modelle erfolgte sowohl mit zurückgehaltenen Testdaten als auch mit solchen, die zu einem späteren Zeitpunkt des Projektes erhoben wurden und somit hinsichtlich ihrer Charakteristik den Modellen völlig unbekannt waren. Während die entwickelten Module zur Datenvorverarbeitung und Modellierung bereits zwecks betrieblicher Nutzung ausgeliefert wurden, steht eine langfristige Bewertung der tatsächlich erzielten Optimierungsergebnisse noch aus. Nur durch die Umsetzung und Dokumentation der generierten Handlungsempfehlungen lassen sich die Performance und die Interaktion der Module innerhalb der Optimierungskette beurteilen.

Für einen nachhaltigen Einsatz datengetriebener Assistenzsysteme ist neben der stetigen Anpassung der Verfahren auch der Mitarbeiterakzeptanz Sorge zu tragen. Als Endanwender müssen sie von digitalen Entscheidungshilfen profitieren und einen direkten Mehrwert in ihnen erkennen. Hierzu gehört, Maschinenbediener in den Bereitstellungsprozess der Handlungsempfehlungen einzubeziehen und ihre Herangehensweise an die Prozesssteuerung zu berücksichtigen. Dies kann durch das Angebot von In-House-Qualifikationsmaßnahmen oder durch spezielle Schulungen in Form von Gamification erfolgen. Alle komplexen Vorgänge wie Prozessabläufe, Eingriffssignale und Handlungsanweisungen müssen dem Nutzer verständlich und transparent dargelegt werden.

Die entwickelten Konzepte lassen sich auch auf andere Gießereien und Fertigungsprozesse anwenden. Bislang beschränkte sich die Betrachtung nur auf einen Teil der gesamten Gießprozesskette. Es standen verfahrenstechnische Fragestellungen im Vordergrund, während die vorgelagerten Prozessschritte nur bedingt Berücksichtigung fanden. Insbesondere die energetische Betrachtung der Fertigung erfolgte nur indirekt: So müssen Rohre, die den Qualitätsanforderungen entsprechen, nicht wieder im Kupolofen eingeschmolzen werden, was mit einer CO₂-Einsparung einhergeht. Weiterhin lassen sich durch die Realisierung eines gleichmäßigeren Wandstärkenverlaufes Rohre insgesamt dünnwandiger fertigen und so Material einsparen. Das verringerte Volumen führt anschließend zu einer niedrigeren Durchwärmzeit im Ofen bei der Nachbehandlung. Vor dem Hintergrund steigender Rohstoff- und Energiepreise ließe sich die Gattierung optimieren, indem der Einsatz von Schrotten und Zuschlagstoffen so geplant wird, dass die vorgegebenen Materialeigenschaften möglichst kosten- und energieeffizient erreicht werden. Darüber hinaus könnten prädiktive Algorithmen zur Minimierung von Stillstandszeiten in die Optimierung der Belegungsplanung eingespannt werden.

Im Hinblick auf die Übertragbarkeit der Optimierungskette auf andere Gießverfahren erwachsen manche für diese Prozesse spezifische Problemstellungen, die in dieser Arbeit nur eine untergeordnete Rolle gespielt haben. So ist bei nahezu kontinuierlichen Fertigungsverfahren wie dem Stranggießen eine instanz- bzw. einzelteilbasierte Darstellung des Prozesses nicht mehr sinnvoll. Es müssten stattdessen auf Zeitreihen beruhende Modelle zur positionsgenauen Vorhersage einer Aufkommenswahrscheinlichkeit für beispielsweise Risse und Anhaftungen entworfen werden. Hier können Verfahren der korrelationsbasierten oder der ℓ_1 -optimalen Störgrößenkompensation helfen, relevante Fehlstellen zu erkennen.

Literatur

- Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu und Xiaoqiang Zheng (2016). „TensorFlow: A system for large-scale machine learning“. In: DOI: 10.48550/ARXIV.1605.08695.
- Adamowski, Julio C., Flavio Buiocchi, Marcos Tsuzuki, Nicolas Perez, Claudio S. Camerini und Carlos Patusco (Juli 2013). „Ultrasonic measurement of micrometric wall-thickness loss due to corrosion inside pipes“. In: *2013 IEEE International Ultrasonics Symposium (IUS)*. Prague, Czech Republic: IEEE, S. 1881–1884. ISBN: 978-1-4673-5686-2. DOI: 10.1109/ULTSYM.2013.0479.
- Aggarwal, Charu C. (2015). *Data mining*. Cham: Springer International Publishing. ISBN: 978-3-319-14141-1. DOI: 10.1007/978-3-319-14142-8.
- Altmann, André, Laura Toloşi, Oliver Sander und Thomas Lengauer (Mai 2010). „Permutation importance: a corrected feature importance measure“. In: *Bioinformatics* 26.10, S. 1340–1347. ISSN: 1460-2059, 1367-4803. DOI: 10.1093/bioinformatics/btq134.
- Andelfinger, Volker P. und Till Hänisch (2017). *Industrie 4.0*. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-15556-8. DOI: 10.1007/978-3-658-15557-5.
- Aragón, Francisco J., Miguel A. Goberna, Marco A. López und Margarita M.L. Rodríguez (2019). *Nonlinear Optimization*. Springer Undergraduate Texts in Mathematics and Technology. Cham: Springer International Publishing. ISBN: 978-3-030-11183-0. DOI: 10.1007/978-3-030-11184-7.
- Babel, Wolfgang (2021). *Industrie 4.0, China 2025, IoT: Der Hype um die Welt der Automatisierung*. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-34717-8. DOI: 10.1007/978-3-658-34718-5.
- Bangert, Patrick (2012). *Optimization for Industrial Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-24973-0. DOI: 10.1007/978-3-642-24974-7.
- Bangert, Patrick, Hrsg. (2021). *Machine Learning and Data Science in the Oil and Gas Industry Best Practices, Tools, and Case Studies*. Cambridge, MA Oxford: Gulf Professional Publishing. ISBN: 978-0-12-820714-7. DOI: 10.1016/C2019-0-02033-X.
- Bauernhansl, Thomas, Bernhard Diegner, Johannes Diemer, Mathias Dümmler, Claudia Eckert, Werner Herfs, Holger Heyn, Claus Hilger, Michael Ten Hompel, Johannes Kalhoff et al. (2015). „Industrie 4.0. Whitepaper FuE-Themen der Plattform Industrie 4.0“. In: *Berlin: Bundesministerium für Wirtschaft und Energie-Plattform Industrie 4.0*.

- Becker, Hermann, Ralf Otte, Oliver Bachmann und Christian Ressel (2011). „Reduzierung von Ausschussraten durch gezielten Einsatz von DataMining-Verfahren“. In: *GIESSE-REI* 98.4, S. 52–55. ISSN: 0016-9765.
- Becker, Wolfgang, Patrick Ulrich und Tim Botzkowski (2017). *Industrie 4.0 im Mittelstand*. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-15655-8. DOI: 10.1007/978-3-658-15656-5.
- Bellman, Richard (1984). *Dynamic programming*. Princeton, NJ: Princeton Univ. Pr. ISBN: 978-0-691-07951-6.
- Bergstra, James und Yoshua Bengio (Feb. 2012). „Random Search for Hyper-Parameter Optimization“. In: *Journal of Machine Learning Research* 13.null, S. 281–305. ISSN: 1532-4435.
- Bhadeshia, H. K. D. H. (2009). „Neural networks and information in materials science“. In: *Statistical Analysis and Data Mining* 1.5, S. 296–305. ISSN: 19321864. DOI: 10.1002/sam.10018.
- Bishop, Christopher M. (2006). *Pattern recognition and machine learning*. Information science and statistics. New York: Springer. ISBN: 978-0-387-31073-2.
- Blank, Julian und Kalyanmoy Deb (2020). „Pymoo: Multi-Objective Optimization in Python“. In: *IEEE Access* 8, S. 89497–89509. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2990567.
- Blondheim, David (Juni 2021). „Improving Manufacturing Applications of Machine Learning by Understanding Defect Classification and the Critical Error Threshold“. In: *International Journal of Metalcasting*. ISSN: 1939-5981, 2163-3193. DOI: 10.1007/s40962-021-00637-0.
- Blondheim, David und Alex Monroe (Jan. 2022). „Macro Porosity Formation: A Study in High Pressure Die Casting“. In: *International Journal of Metalcasting* 16.1, S. 330–341. ISSN: 1939-5981, 2163-3193. DOI: 10.1007/s40962-021-00602-x.
- Borchani, Hanen, Gherardo Varando, Concha Bielza und Pedro Larrañaga (2015). „A survey on multi-output regression“. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.5, S. 216–233. ISSN: 19424787. DOI: 10.1002/widm.1157.
- Bourlard, H. und Y. Kamp (Sep. 1988). „Auto-association by multilayer perceptrons and singular value decomposition“. In: *Biological Cybernetics* 59.4-5, S. 291–294. ISSN: 0340-1200, 1432-0770. DOI: 10.1007/BF00332918.
- Branke, Jürgen, Kalyanmoy Deb, Kaisa Miettinen und Roman Słowiński, Hrsg. (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Bd. 5252. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-88907-6. DOI: 10.1007/978-3-540-88908-3.
- Braunholz, Christopher (2021). *Integration von Sensitivitätsanalysemethoden in den Entwicklungsprozess für Fahrwerkregelsysteme*. Wissenschaftliche Reihe Fahrzeugtechnik Universität Stuttgart. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-33358-4. DOI: 10.1007/978-3-658-33359-1.
- Breiman, Leo (Aug. 1996). „Bagging predictors“. In: *Machine Learning* 24.2, S. 123–140. ISSN: 0885-6125. DOI: 10.1007/BF00058655.
- Breiman, Leo, Hrsg. (1998). *Classification and regression trees*. 1. CRC Press repr. Boca Raton, Fla.: Chapman & Hall/CRC. ISBN: 978-0-412-04841-8.

- Breiman, Leo (2001). „Random forests“. In: *Machine Learning* 45.1, S. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324.
- Brown, John R., Hrsg. (1999). *Foseco non-ferrous foundryman's handbook*. 11th ed. Oxford, England; Boston: Butterworth-Heinemann. ISBN: 978-0-7506-4286-6. DOI: 10.1016/B978-0-7506-4286-6.X5000-1.
- Brunton, Bingni W., Steven L. Brunton, Joshua L. Proctor und J. Nathan Kutz (Jan. 2016). „Sparse Sensor Placement Optimization for Classification“. In: *SIAM Journal on Applied Mathematics* 76.5, S. 2099–2122. ISSN: 0036-1399, 1095-712X. DOI: 10.1137/15M1036713.
- Brunton, Steven L. und J. Nathan Kutz (2019). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge, United Kingdom und New York, NY: Cambridge University Press. ISBN: 978-1-108-38069-0. DOI: 10.1017/9781108380690.
- Brunton, Steven L., Bernd R. Noack und Petros Koumoutsakos (2020). „Machine learning for fluid mechanics“. In: *Annual Review of Fluid Mechanics* 52.1, S. 477–508. ISSN: 0066-4189. DOI: 10.1146/annurev-fluid-010719-060214.
- Brunton, Steven L., Joshua L. Proctor, Jonathan H. Tu und J. Nathan Kutz (2015). „Compressed sensing and dynamic mode decomposition“. In: *Journal of Computational Dynamics* 2.2, S. 165. DOI: 10.3934/jcd.2015002.
- Bührig-Polaczek, Andreas, Hrsg. (2014). *Handbuch Urformen*. Edition Handbuch der Fertigungstechnik. München: Hanser. ISBN: 978-3-446-42035-9.
- Bührig-Polaczek, Andreas, Max Rudack, Felix Gandha Salentin, Marvin Sandt, Dierk Hartmann, Maximilian Brait, Eduard Koppensteiner, Gerhard Schindelbacher und Peter Schumacher (2021). „Digitalisierung in der Gießereitechnik, Teil 1: Herausforderungen und Chancen“. In: *GIESSEREI* 108.6, S. 37–43. ISSN: 0016-9765.
- Candès, Emmanuel J., Xiaodong Li, Yi Ma und John Wright (2011). „Robust principal component analysis?“ In: *Journal of the ACM* 58.3, S. 1–37. ISSN: 0004-5411. DOI: 10.1145/1970392.1970395.
- Cattell, R. B. (1966). „The scree test for the number of factors“. In: *Multivariate behavioral research* 1.2, S. 245–276. ISSN: 0027-3171. DOI: 10.1207/s15327906mbr0102_10.
- Cawley, Gavin C. und Nicola L. C. Talbot (2010). „On over-fitting in model selection and subsequent selection bias in performance evaluation“. In: *The Journal of Machine Learning Research* 11, S. 2079–2107. ISSN: 1532-4435.
- Champion, Kathleen, Bethany Lusch, J. Nathan Kutz und Steven L. Brunton (2019). „Data-driven discovery of coordinates and governing equations“. In: *Proceedings of the National Academy of Sciences of the United States of America* 116.45, S. 22445–22451. DOI: 10.1073/pnas.1906995116.
- Chapman, Pete, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer und Rudiger Wirth (2000). „CRISP-DM 1.0: Step-by-step data mining guide“. In: *SPSS inc* 9.
- Chen, Tianqi und Carlos Guestrin (Aug. 2016). „XGBoost: A Scalable Tree Boosting System“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, S. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785.
- Chen, Xingle, Jingwen Li und Zhaohui Wang (Dez. 2020). „Inversion Method in Pulsed Eddy Current Testing for Wall Thickness of Ferromagnetic Pipes“. In: *IEEE Transac-*

- tions on Instrumentation and Measurement* 69.12, S. 9766–9773. ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2020.3005114.
- Cherkassky, Vladimir und Filip Mulier (2007). *Learning from data: Concepts, theory, and methods*. Second edition. Hoboken, New Jersey: IEEE Press Wiley-Interscience John Wiley & Sons Inc. Publication / IEEE Xplore (Online service). ISBN: 978-0-470-14052-9. DOI: 10.1002/9780470140529.
- Chiang, Leo H, Evan L Russell und Richard D Braatz (März 2000). „Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis“. In: *Chemometrics and Intelligent Laboratory Systems* 50.2, S. 243–252. ISSN: 01697439. DOI: 10.1016/S0169-7439(99)00061-1.
- Chollet, François (2021). *Deep learning with Python*. Second edition. Shelter Island: Manning Publications. ISBN: 978-1-61729-686-4.
- Chong, Edwin Kah Pin und Stanislaw H. Żak (2013). *An Introduction to Optimization*. Fourth edition. Wiley series in discrete mathematics and optimization. Hoboken, New Jersey: Wiley. ISBN: 978-1-118-27901-4.
- Cleve, Jürgen und Uwe Lämmel (2016). *Data mining*. 2nd ed. De gruyter studium. Berlin: De Gruyter. ISBN: 978-3-11-045677-6. DOI: 10.1515/9783110456776.
- Cohen, Jacob (1960). „A coefficient of agreement for nominal scales“. In: *Educational and Psychological Measurement* 20.1, S. 37–46. ISSN: 0013-1644. DOI: 10.1177/001316446002000104.
- Cooley, James W. und John W. Tukey (1965). „An algorithm for the machine calculation of complex Fourier series“. In: *Mathematics of Computation* 19.90, S. 297–301. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-1965-0178586-1.
- Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal und T. Meyarivan (Apr. 2002). „A fast and elitist multiobjective genetic algorithm: NSGA-II“. In: *IEEE Transactions on Evolutionary Computation* 6.2, S. 182–197. ISSN: 1089778X. DOI: 10.1109/4235.996017.
- Deike, Linus (Apr. 2020). „Kontinuierliche Qualitätsoptimierung von Produktionsprozessen durch maschinelle Lernverfahren“. Diss. Duisburg: Universität Duisburg-Essen. DOI: 10.17185/DUEPUBLICO/71560.
- DIN 8580 (2003). *DIN 8580:2003-09 - Fertigungsverfahren: Begriffe, Einteilung*. DOI: 10.31030/9500683.
- DIN EN 1563 (Apr. 2019). *DIN EN 1563:2019-04 Gießereiwesen - Gusseisen mit Kugelgraphit; Deutsche Fassung EN 1563:2018*. Techn. Ber. Beuth Verlag GmbH. DOI: 10.31030/2792445.
- DIN EN 545 (Sep. 2011). *DIN EN 545:2011-09, Rohre, Formstücke, Zubehörteile aus duktilem Gusseisen und ihre Verbindungen für Wasserleitungen - Anforderungen und Prüfverfahren; Deutsche Fassung EN 545:2010*. Techn. Ber. Beuth Verlag GmbH. DOI: 10.31030/1813629.
- DIN EN ISO 9241-110 (2020). *Ergonomie der Mensch-System-Interaktion – Teil 110: Interaktionsprinzipien (ISO 9241-110:2020); Deutsche Fassung EN ISO 9241-110:2020*.
- Duriez, Thomas, Steven L. Brunton und Bernd R. Noack (2017). *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence*. Bd. 116. Fluid Mechanics and Its Applications. Cham: Springer International Publishing. ISBN: 978-3-319-40623-7. DOI: 10.1007/978-3-319-40624-4.

- Eckart, Carl und Gale Young (1936). „The approximation of one matrix by another of lower rank“. In: *Psychometrika* 1.3, S. 211–218. ISSN: 0033-3123. DOI: 10.1007/BF02288367.
- Ehrgott, Matthias (2005). *Multicriteria optimization*. 2nd ed. Berlin ; New York: Springer. ISBN: 978-3-540-21398-7.
- Engelbrecht, Andries P. (Okt. 2007). *Computational Intelligence*. Chichester, UK: John Wiley & Sons, Ltd. ISBN: 978-0-470-51251-7. DOI: 10.1002/9780470512517.
- Escobar, Carlos A., Megan E. McGovern und Ruben Morales-Menendez (Dez. 2021). „Quality 4.0: a review of big data challenges in manufacturing“. In: *Journal of Intelligent Manufacturing* 32.8, S. 2319–2334. ISSN: 0956-5515, 1572-8145. DOI: 10.1007/s10845-021-01765-4.
- Fahrmeir, Ludwig, Christian Heumann, Rita Künstler, Iris Pigeot und Gerhard Tutz (2016). *Statistik*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-50371-3. DOI: 10.1007/978-3-662-50372-0.
- Fawcett, Tom (2006). „An introduction to ROC analysis“. In: *Pattern Recognition Letters* 27.8, S. 861–874. ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010.
- Fayyad, Usama, Gregory Piatetsky-Shapiro und Padhraic Smyth (1996). „From data mining to knowledge discovery in databases“. In: *AI magazine* 17.3, S. 37–54. ISSN: 2371-9621.
- Ferri, Francesc J., Pavel Pudil, Mohamad Hatef und Josef Kittler (1994). „Comparative study of techniques for large-scale feature selection“. In: *Machine Intelligence and Pattern Recognition*. Bd. 16. Elsevier, S. 403–413. ISBN: 978-0-444-81892-8. DOI: 10.1016/B978-0-444-81892-8.50040-7.
- Feurer, Matthias, Katharina Eggenberger, Stefan Falkner, Marius Lindauer und Frank Hutter (Sep. 2021). *Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning*. arXiv: 2007.04074 [cs, stat].
- Freund, Yoav und Robert E Schapire (Aug. 1997). „A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting“. In: *Journal of Computer and System Sciences* 55.1, S. 119–139. ISSN: 00220000. DOI: 10.1006/jcss.1997.1504.
- Friedman, Jerome H. (2001). „Greedy Function Approximation: A Gradient Boosting Machine“. In: *Annals of statistics*, S. 1189–1232. ISSN: 0090-5364.
- Fritz, Alfred Herbert (2018). *Fertigungstechnik*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-56534-6. DOI: 10.1007/978-3-662-56535-3.
- Gad, Ahmed Fawzy (2021). „PyGAD: An Intuitive Genetic Algorithm Python Library“. In: DOI: 10.48550/ARXIV.2106.06158.
- Gao, Fuchang und Lixing Han (Jan. 2012). „Implementing the Nelder-Mead simplex algorithm with adaptive parameters“. In: *Computational Optimization and Applications* 51.1, S. 259–277. ISSN: 0926-6003, 1573-2894. DOI: 10.1007/s10589-010-9329-3.
- Gavish, Matan und David L. Donoho (2014). „The optimal hard threshold for singular values is $4/\sqrt{3}$ “. In: *IEEE Transactions on Information Theory* 60.8, S. 5040–5053. ISSN: 0018-9448. DOI: 10.1109/TIT.2014.2323359.
- Ge, Zhiqiang, Zhihuan Song, Steven X. Ding und Biao Huang (2017). „Data mining and analytics in the process industry: The role of machine learning“. In: *IEEE Access* 5, S. 20590–20616. DOI: 10.1109/ACCESS.2017.2756872.

- Géron, Aurélien (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. Second edition. Beijing u. a.: O'Reilly. ISBN: 978-1-4920-3264-9.
- Goldstein, Alex, Adam Kapelner, Justin Bleich und Emil Pitkin (2014). „Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation“. In: *Journal of Computational and Graphical Statistics* 24.1, S. 44–65. DOI: 10.1080/10618600.2014.907095.
- Golub, Gene H. und Charles F. van Loan (2013). *Matrix computations*. 4. ed. Johns Hopkins studies in mathematical sciences. Baltimore, Md.: Johns Hopkins Univ. Press. ISBN: 978-1-4214-0859-0.
- Gottschling, Johannes, Caroline Biedermann, Annika Tonnius und Dierk Hartmann (2020). „Cloudbasierter Rohstoffhandel – datengetriebene Optimierung von Metallschmelzen“. In: *GIESSEREI* 107.7, S. 34–39. ISSN: 0016-9765.
- Greenwell, Brandon M. (2017). „pdp: An R Package for Constructing Partial Dependence Plots“. In: *The R Journal* 9.1, S. 421.
- Gröger, Christoph (2015). *Advanced Manufacturing Analytics: datengetriebene Optimierung von Fertigungsprozessen*. 1. Aufl. Lohmar Köln: Eul-Verl. ISBN: 978-3-8441-0420-2.
- Gröger, Christoph (März 2018). „Building an Industry 4.0 Analytics Platform: Practical Challenges, Approaches and Future Research Directions“. In: *Datenbank-Spektrum* 18.1, S. 5–14. ISSN: 1618-2162, 1610-1995. DOI: 10.1007/s13222-018-0273-1.
- Gröger, Christoph, Florian Niedermann und Bernhard Mitschang (2012). „Data Mining-driven Manufacturing Process Optimization“. In: *Proceedings of the world congress on engineering*. Bd. 3. London. ISBN: 978-988-19252-2-0.
- Grus, Joel (2020). *Einführung in Data Science: Grundprinzipien der Datenanalyse mit Python*. Übers. von Kristian Rother und Thomas Demmig. 2. Auflage. Heidelberg: O'Reilly. ISBN: 978-3-96010-336-3.
- Gutiérrez, José Manuel, Amaia Natxiondo, Javier Nieves, Argoitz Zabala und Jon Sertucha (Apr. 2017). „Statistical Study to Evaluate the Effect of Processing Variables on Shrinkage Incidence During Solidification of Nodular Cast Irons“. In: *Metallurgical and Materials Transactions A* 48.4, S. 1827–1840. ISSN: 1073-5623, 1543-1940. DOI: 10.1007/s11661-017-3969-1.
- Guyon, Isabelle, Jason Weston, Stephen Barnhill und Vladimir Vapnik (2002). „Gene Selection for Cancer Classification using Support Vector Machines“. In: *Machine Learning* 46.1/3, S. 389–422. ISSN: 08856125. DOI: 10.1023/A:1012487302797.
- Hahn, Ingo und Jörg Sturm (2012). „Automatische Optimierung in der Gießprozesssimulation“. In: *GIESSEREI* 99.4, S. 56–67. ISSN: 0016-9765.
- Halim, Suhaila Abd, M. Z. Puteri Zirwatul Nadila, Arsmah Ibrahim und Yupiter H P Manurung (Apr. 2012). „Performance of noise removal methods with image quality parameter on μ -focused digital radiographic image“. In: *2012 IEEE Business, Engineering & Industrial Applications Colloquium (BEIAC)*. Kuala Lumpur, Malaysia: IEEE, S. 313–317. ISBN: 978-1-4673-0426-9. DOI: 10.1109/BEIAC.2012.6226073.
- Han, Jiawei und Micheline Kamber (2012). *Data mining: concepts and techniques*. 3rd ed. Burlington, MA: Elsevier. ISBN: 978-0-12-381479-1.
- Hastie, Trevor, Robert Tibshirani und Jerome H. Friedman (2017). *The elements of statistical learning: Data mining, inference, and prediction*. Second edition, corrected

- at 12th printing 2017. Springer series in statistics. New York, NY: Springer. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7.
- Herrmann, Christoph, Helge Pries und Götz Hartmann, Hrsg. (2013). *Energie- und ressourceneffiziente Produktion von Aluminiumdruckguss*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-39852-0. DOI: 10.1007/978-3-642-39853-7.
- Hill, Austin Bradford (Mai 1965). „The Environment and Disease: Association or Causation?“ In: *Proceedings of the Royal Society of Medicine* 58.5, S. 295–300. ISSN: 0035-9157. DOI: 10.1177/003591576505800503.
- Hinton, G. E. und R. R. Salakhutdinov (Juli 2006). „Reducing the Dimensionality of Data with Neural Networks“. In: *Science* 313.5786, S. 504–507. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1127647.
- Hoerl, Arthur E. und Robert W. Kennard (Feb. 1970). „Ridge Regression: Biased Estimation for Nonorthogonal Problems“. In: *Technometrics* 12.1, S. 55–67. ISSN: 0040-1706. DOI: 10.1080/00401706.1970.10488634.
- Horn, John L. (1965). „A Rationale and Test for the Number of Factors in Factor Analysis“. In: *Psychometrika* 30, S. 179–185. ISSN: 0033-3123. DOI: 10.1007/BF02289447.
- Hotelling, Harold (1933). „Analysis of a complex of statistical variables into principal components.“ In: *Journal of Educational Psychology* 24.6, S. 417–441. ISSN: 0022-0663. DOI: 10.1037/h0071325.
- Hyndman, Rob J. und Anne B. Koehler (2006). „Another look at measures of forecast accuracy“. In: *International Journal of Forecasting* 22.4, S. 679–688. ISSN: 01692070. DOI: 10.1016/j.ijforecast.2006.03.001.
- Isermann, Rolf und Marco Münchhof (2011). *Identification of dynamic systems*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-78878-2. DOI: 10.1007/978-3-540-78879-9.
- James, Gareth, Daniela Witten, Trevor Hastie und Robert Tibshirani (2021). *An introduction to statistical learning: with applications in R*. Second edition. Springer texts in statistics. New York: Springer. ISBN: 978-1-07-161417-4.
- Jamieson, Kevin und Ameet Talwalkar (Feb. 2015). *Non-stochastic Best Arm Identification and Hyperparameter Optimization*. arXiv: 1502.07943 [cs, stat].
- Jarre, Florian und Josef Stoer (2019). *Optimierung: Einführung in mathematische Theorie und Methoden*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-58854-3. DOI: 10.1007/978-3-662-58855-0.
- Kallien, Lothar, Volkan Görgün und Christian Wilhelm (2018). „Einfluss der Elektromobilität auf die Gussproduktion in der deutschen Gießerei-Industrie“. In: *GIESSEREI* 105.4, S. 84–92. ISSN: 0016-9765.
- Kaufmann, Tim, Dierk Hartmann, Christoph Jonas und Friedrich Lohmann-Voß (2022). „Datenbasierte Verfahren zur Steigerung der Energieeffizienz“. In: *GIESSEREI* 109.2, S. 24–31. ISSN: 0016-9765.
- Kaufmann, Tim und Annika Tonnius (2021). „Rohstoff- und Energieeffizienz im induktiven Schmelzprozess steigern“. In: *GIESSEREI* 108.4, S. 30–41. ISSN: 0016-9765.
- Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye und Tie-Yan Liu (2017). „LightGBM: A Highly Efficient Gradient Boosting Decision Tree“. In: *Advances in Neural Information Processing Systems*. Hrsg. von I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan und R. Garnett. Bd. 30. Curran Associates, Inc.

- Kelley, Carl T. (Jan. 1999). „Detection and Remediation of Stagnation in the Nelder-Mead Algorithm Using a Sufficient Decrease Condition“. In: *SIAM Journal on Optimization* 10.1, S. 43–55. ISSN: 1052-6234, 1095-7189. DOI: 10.1137/S1052623497315203.
- Kennedy, James F. und Russell C. Eberhart (1995). „Particle Swarm Optimization“. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Bd. 4. Perth, WA, Australia: IEEE, S. 1942–1948. ISBN: 978-0-7803-2768-9. DOI: 10.1109/ICNN.1995.488968.
- Kennedy, James F., Russell C. Eberhart und Yuhui Shi (2001). *Swarm intelligence*. The Morgan Kaufmann series in evolutionary computation. San Francisco: Morgan Kaufmann Publishers. ISBN: 978-1-55860-595-4.
- Keogh, Eamonn J. und Chotirat Ann Ratanamahatana (2005). „Exact indexing of dynamic time warping“. In: *Knowledge and Information Systems* 7.3, S. 358–386. ISSN: 0219-1377. DOI: 10.1007/s10115-004-0154-9.
- Kerst, Philine und Johannes Gottschling (2020). „Datengetriebene Modelle – Aus Daten Prozesswissen generieren“. In: *GIESSEREI Special* 107.1, S. 24–29. ISSN: 2512-3157.
- Kerst, Philine, Johannes Gottschling und Dierk Hartmann (2018a). „Analyse von Gießereidaten mit Methoden des maschinellen Lernens - Teil 1“. In: *Gießerei-Praxis* 69.10, S. 11–17. ISSN: 0016-9781.
- Kerst, Philine, Johannes Gottschling und Dierk Hartmann (2018b). „Analyse von Gießereidaten mit Methoden des maschinellen Lernens - Teil 2“. In: *Gießerei-Praxis* 69.12, S. 9–15. ISSN: 0016-9781.
- Kingma, Diederik P. und Jimmy Ba (2014). „Adam: A Method for Stochastic Optimization“. In: *arXiv preprint arXiv:1412.6980*. San Diego: arXiv. DOI: 10.48550/ARXIV.1412.6980. arXiv: 1412.6980.
- Kingma, Diederik P. und Max Welling (2013). „Auto-Encoding Variational Bayes“. In: DOI: 10.48550/ARXIV.1312.6114.
- Klocke, Fritz (2018). *Fertigungsverfahren 5: Gießen und Pulvermetallurgie*. VDI-Buch. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-54727-4. DOI: 10.1007/978-3-662-54728-1.
- Kochenderfer, Mykel J. und Tim A. Wheeler (2019). *Algorithms for optimization*. Cambridge, Massachusetts: The MIT Press. ISBN: 978-0-262-03942-0.
- Kohavi, Ron (1995). „A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection“. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann, S. 1137–1143.
- Konrad, Benedikt, Daniel Lieber und Jochen Deuse (2013). „Striving for Zero Defect Production: Intelligent Manufacturing Control Through Data Mining in Continuous Rolling Mill Processes“. In: *Robust Manufacturing Control*. Hrsg. von Katja Windt. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 215–229. ISBN: 978-3-642-30748-5. DOI: 10.1007/978-3-642-30749-2_16.
- Kuhn, Max und Kjell Johnson (2013). *Applied Predictive Modeling*. New York, NY: Springer New York. ISBN: 978-1-4614-6848-6. DOI: 10.1007/978-1-4614-6849-3.
- Kutz, J. Nathan, Steven L. Brunton, Bingni W. Brunton und Joshua L. Proctor (Nov. 2016). *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, PA: Society for Industrial und Applied Mathematics. ISBN: 978-1-61197-449-2. DOI: 10.1137/1.9781611974508.

- Langer, Heimo J. und William R. Dunnivant (Apr. 2011). „Foundry Resins“. In: *Encyclopedia of Polymer Science and Technology*. Hrsg. von John Wiley & Sons, Inc. Hoboken, NJ, USA: John Wiley & Sons, Inc., S. 290–298. ISBN: 978-0-471-44026-0. DOI: 10.1002/0471440264.pst544.
- Lebowitz, Carol A. und Lawrence M. Brown (1993). „Ultrasonic Measurement of Pipe Thickness“. In: *Review of Progress in Quantitative Nondestructive Evaluation*. Hrsg. von Donald O. Thompson und Dale E. Chimenti. Boston, MA: Springer US, S. 1987–1994. ISBN: 978-1-4613-6233-3. DOI: 10.1007/978-1-4615-2848-7_255.
- Li, Xiaoli, S. K. Tso, Xin-Ping Guan und Qian Huang (Dez. 2006). „Improving Automatic Detection of Defects in Castings by Applying Wavelet Technique“. In: *IEEE Transactions on Industrial Electronics* 53.6, S. 1927–1934. ISSN: 0278-0046. DOI: 10.1109/TIE.2006.885448.
- Lundberg, Scott M und Su-In Lee (2017). „A Unified Approach to Interpreting Model Predictions“. In: *Advances in Neural Information Processing Systems*. Hrsg. von I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan und R. Garnett. Bd. 30. Curran Associates, Inc.
- Manohar, Krithika, Bingni W. Brunton, J. Nathan Kutz und Steven L. Brunton (2018). „Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns“. In: *IEEE Control Systems Magazine* 38.3, S. 63–86. DOI: 10.1109/MCS.2018.2810460.
- Meissner, Knut und Burckhardt König (Juni 2011). „Die neue Gießuhr – vollflexibles Markiersystem für Gussteile“. In: *GIESSEREI* 98.6, S. 108–113. ISSN: 0016-9765.
- Miettinen, Kaisa (1998). *Nonlinear Multiobjective Optimization*. Hrsg. von Frederick S. Hillier. Bd. 12. International Series in Operations Research & Management Science. Boston, MA: Springer US. ISBN: 978-1-4613-7544-9. DOI: 10.1007/978-1-4615-5563-6.
- Miranda, Lester J. (2018). „PySwarms: a research toolkit for Particle Swarm Optimization in Python“. In: *The Journal of Open Source Software* 3.21, S. 433. ISSN: 2475-9066. DOI: 10.21105/joss.00433.
- Mittag, Hans-Joachim (2017). *Statistik*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-55320-6. DOI: 10.1007/978-3-662-55321-3.
- Molnar, Christoph, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A. Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup und Bernd Bischl (Aug. 2021). „General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models“. In: *arXiv:2007.04131 [cs, stat]*. arXiv: 2007.04131 [cs, stat].
- Montgomery, Douglas C. (2013). *Introduction to Statistical Quality Control*. 7th ed. Hoboken, NJ: Wiley. ISBN: 978-1-118-14681-1.
- Montgomery, Douglas C. (2017). *Design and analysis of experiments*. Ninth edition. Hoboken, NJ: John Wiley & Sons, Inc. ISBN: 978-1-119-11347-8.
- Müller, Meinard, Henning Mattes und Frank Kurth (2006). „An efficient multiscale approach to audio synchronization“. In: *ISMIR*. Bd. 546, S. 192–197.
- Nelder, J. A. und R. Mead (Jan. 1965). „A Simplex Method for Function Minimization“. In: *The Computer Journal* 7.4, S. 308–313. ISSN: 0010-4620, 1460-2067. DOI: 10.1093/comjnl/7.4.308.

- Ng, Andrew und Michael Jordan (2001). „On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes“. In: *Advances in neural information processing systems* 14.
- Ngatchou, P., A. Zarei und A. El-Sharkawi (2005). „Pareto Multi Objective Optimization“. In: *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. Arlington, Virginia, USA: IEEE, S. 84–91. ISBN: 978-1-59975-174-0. DOI: 10.1109/ISAP.2005.1599245.
- Nguyen, Linh und Jaime Valls Miro (Dez. 2020). „Efficient Evaluation of Remaining Wall Thickness in Corroded Water Pipes Using Pulsed Eddy Current Data“. In: *IEEE Sensors Journal* 20.23, S. 14465–14473. ISSN: 1530-437X, 1558-1748, 2379-9153. DOI: 10.1109/JSEN.2020.3007868.
- Niedermann, Florian (2015). „Deep Business Optimization: concepts and architecture for an analytical business process optimization platform“. Diss. Universität Stuttgart. DOI: 10.18419/opus-8810.
- Nieves, Javier, Igor Santos und Pablo G. Bringas (2012). „Combination of machine-learning algorithms for fault prediction in high-precision foundries“. In: *International conference on database and expert systems applications*, S. 56–70.
- Nieves, Javier, Igor Santos, Pablo G. Bringas, Argoitz Zabala und Jon Sertucha (2014). „Enhancing the stationary state prediction in Model Predictive Control systems to avoid Dross defect in heavy-section foundries“. In: *IEEE 9th conference on industrial electronics and applications (ICIEA), 2014*. Piscataway, NJ: IEEE. ISBN: 978-1-4799-4315-9. DOI: 10.1109/iciea.2014.6931200.
- Nocedal, Jorge und Stephen J. Wright (2006). *Numerical optimization*. 2nd ed. Springer series in operations research. New York: Springer. ISBN: 978-0-387-30303-1.
- Olson, Randal S., Nathan Bartley, Ryan J. Urbanowicz und Jason H. Moore (Juli 2016). „Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science“. In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. Denver Colorado USA: ACM, S. 485–492. ISBN: 978-1-4503-4206-3. DOI: 10.1145/2908812.2908918.
- Pastor-López, Iker, José Gaviria de la Puerta, Borja Sanz, Aitor Goti und Pablo G. Bringas (Okt. 2019a). „How IoT and computer vision could improve the casting quality“. In: *Proceedings of the 9th International Conference on the Internet of Things*. Bilbao Spain: ACM, S. 1–8. ISBN: 978-1-4503-7207-7. DOI: 10.1145/3365871.3365878.
- Pastor-López, Iker, Jorge de-la-Pena-Sordo, Igor Santos und Pablo G. Bringas (2015). „Surface Defect Categorization of Imperfections in High Precision Automotive Iron Foundries using Best Crossing Line Profile“. In: *2015 IEEE 10th conference on industrial electronics and applications (ICIEA)*. Piscataway, NJ: IEEE. ISBN: 978-1-4799-8389-6. DOI: 10.1109/iciea.2015.7334136.
- Pastor-López, Iker, Borja Sanz, José Gaviria de la Puerta und Pablo G. Bringas (2019b). „Surface Defect Modelling Using Co-occurrence Matrix and Fast Fourier Transformation“. In: *Hybrid Artificial Intelligent Systems*. Hrsg. von Hilde Pérez García, Lidia Sánchez González, Manuel Castejón Limas, Héctor Quintián Pardo und Emilio Corchado Rodríguez. Bd. 11734. Cham: Springer International Publishing, S. 745–757. ISBN: 978-3-030-29858-6. DOI: 10.1007/978-3-030-29859-3_63.
- Pastor-López, Iker, Borja Sanz, Alberto Tellaeche, Giuseppe Psaila, José Gaviria de la Puerta und Pablo G. Bringas (Okt. 2021). „Quality assessment methodology based on

- machine learning with small datasets: Industrial castings defects“. In: *Neurocomputing* 456, S. 622–628. ISSN: 09252312. DOI: 10.1016/j.neucom.2020.08.094.
- Pearson, Karl (Nov. 1901). „On lines and planes of closest fit to systems of points in space“. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, S. 559–572. ISSN: 1941-5982. DOI: 10.1080/14786440109462720.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot und E. Duchesnay (2011). „Scikit-learn: Machine learning in python“. In: *Journal of Machine Learning Research* 12, S. 2825–2830.
- Penya, Yoseba K., Pablo G. Bringas und Argoitz Zabala (2008a). „Advanced fault prediction in high-precision foundry production“. In: *6th IEEE international conference on industrial informatics, 2008*. Piscataway, NJ: IEEE Service Center. ISBN: 978-1-4244-2170-1. DOI: 10.1109/indin.2008.4618372.
- Penya, Yoseba K., Pablo G. Bringas und Argoitz Zabala (2008b). „Efficient failure-free foundry production“. In: *2008 IEEE conference on emerging technologies & factory automation*. Piscataway, NJ: IEEE. ISBN: 978-1-4244-1505-2. DOI: 10.1109/etfa.2008.4638399.
- Pieper, Martin (2017). *Mathematische Optimierung: eine Einführung in die kontinuierliche Optimierung mit Beispielen*. essentials. Wiesbaden [Heidelberg]: Springer Fachmedien Wiesbaden GmbH. ISBN: 978-3-658-16975-6. DOI: 10.1007/978-3-658-16975-6.
- Pille, Christoph und Thomas Rahn (2017). „RFID bringt Gießereien in die Industrie 4.0“. In: *RFID im Blick* 3, S. 30–39. ISSN: 1860-5907.
- Poli, Riccardo (Feb. 2008). „Analysis of the Publications on the Applications of Particle Swarm Optimisation“. In: *Journal of Artificial Evolution and Applications* 2008, S. 1–10. ISSN: 1687-6229. DOI: 10.1155/2008/685175.
- Powell, M. J. D. (Jan. 1998). „Direct search algorithms for optimization calculations“. In: *Acta Numerica* 7, S. 287–336. ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S0962492900002841.
- Prokhorenkova, Liudmila, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush und Andrey Gulin (Jan. 2019). „CatBoost: unbiased boosting with categorical features“. In: *arXiv:1706.09516 [cs]*. arXiv: 1706.09516 [cs].
- Qin, S. Joe (2009). „Data-driven fault detection and diagnosis for complex industrial processes“. In: *IFAC Proceedings Volumes* 42.8, S. 1115–1125. ISSN: 14746670. DOI: 10.3182/20090630-4-ES-2003.00184.
- Quinlan, J. Ross (1993). *C4.5: programs for machine learning*. The Morgan Kaufmann series in machine learning. San Mateo, Calif: Morgan Kaufmann Publishers. ISBN: 978-1-55860-238-0.
- Ranzato, Marc’Aurelio, Christopher Poultney, Sumit Chopra und Yann Cun (2006). „Efficient learning of sparse representations with an energy-based model“. In: *Advances in neural information processing systems* 19.
- Raschka, Sebastian und Vahid Mirjalili (2021). *Machine learning mit Python und Keras, TensorFlow 2 und Scikit-learn: das umfassende Praxis-Handbuch für data science, deep learning und predictive analytics*. Übers. von Knut Lorenzen. 3., aktualisierte und erweiterte Auflage. Frechen: mitp. ISBN: 978-3-7475-0214-3.

- Read, Jesse, Bernhard Pfahringer, Geoff Holmes und Eibe Frank (Dez. 2011). „Classifier chains for multi-label classification“. In: *Machine Learning* 85.3, S. 333–359. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/s10994-011-5256-5.
- Reinheimer, Stefan (2017). *Industrie 4.0*. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-18164-2. DOI: 10.1007/978-3-658-18165-9.
- Richter, Stefan (2019). *Statistisches und maschinelles Lernen: Gängige Verfahren im Überblick*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-59353-0. DOI: 10.1007/978-3-662-59354-7.
- Riße, Andreas (2012). *Fertigungsverfahren der Mechatronik, Feinwerk- und Präzisionsgerätetechnik*. Wiesbaden: Springer Vieweg. ISBN: 978-3-8348-1519-4.
- Runkler, Thomas A. (2015). *Data Mining*. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-8348-1694-8. DOI: 10.1007/978-3-8348-2171-3.
- Russell, Stuart J. und Peter Norvig (2016). *Artificial intelligence: A modern approach*. Third edition, Global edition. Always learning. Boston, Columbus und Indianapolis: Pearson. ISBN: 978-0-13-604259-4.
- Saleem, Muhammad, Dierk Hartmann, Saadia Malik, Heiko Gemming und Johannes Gottschling (2015). „Intelligente Prozesssteuerung in Gießereien“. In: *Gießerei Rundschau* 62.5, S. 106–121.
- Salentin, Felix Gandha und Dierk Hartmann (2021). „Datengetriebene Optimierung von Wandstärkenverteilungen duktiler Gusseisenrohre“. In: *GIESSEREI Special* 108.2, S. 36–42. ISSN: 2512-3157.
- Sandt, Marvin, Michael Beck, Fabian Linkerhägner, Dierk Hartmann, Michael Layh und Bernd R. Pinzer (2020). „CastCode – Gussteilrückverfolgbarkeit an automatischen Formanlagen“. In: *GIESSEREI Special* 107.1, S. 30–39. ISSN: 2512-3157.
- Santos, Igor, Javier Nieves, Pablo G. Bringas, Argoitz Zabala und Jon Sertucha (Juni 2013). „Supervised learning classification for dross prediction in ductile iron casting production“. In: *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*. Melbourne, VIC: IEEE, S. 1749–1754. ISBN: 978-1-4673-6322-8. DOI: 10.1109/ICIEA.2013.6566651.
- Santos, Igor, Javier Nieves, Yoseba K. Peña und Pablo G. Bringas (2009a). „Machine-learning-based mechanical properties prediction in foundry production“. In: *2009 iccas-sice*, S. 4536–4541.
- Santos, Igor, Javier Nieves, Yoseba K. Peña und Pablo G. Bringas (2009b). „Optimising machine-learning-based fault prediction in foundry production“. In: *Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living*. Hrsg. von Sigeru Omatu. Bd. 5518. Lecture notes in computer science. Berlin: Springer, S. 554–561. ISBN: 978-3-642-02480-1. DOI: 10.1007/978-3-642-02481-8_80.
- Scherl, Isabel, Benjamin Strom, Jessica K. Shang, Owen Williams, Brian L. Polagye und Steven L. Brunton (2020). „Robust principal component analysis for modal decomposition of corrupt fluid flows“. In: *Physical Review Fluids* 5.5. DOI: 10.1103/PhysRevFluids.5.054401.
- Schmitt, Berthold Immanuel (2015). *Convergence Analysis for Particle Swarm Optimization*. FAU Forschungen Reihe B, Medizin, Naturwissenschaft, Technik Band 3. Erlangen: FAU University Press. ISBN: 978-3-944057-30-9.

- Scholz, Daniel (2018). *Optimierung interaktiv: Grundlagen verstehen, Modelle erforschen und Verfahren anwenden*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-57952-7. DOI: 10.1007/978-3-662-57953-4.
- Shapley, Lloyd S. (Dez. 1953). „A Value for n-Person Games“. In: *Contributions to the Theory of Games (AM-28), Volume II*. Hrsg. von Harold William Kuhn und Albert William Tucker. Princeton University Press, S. 307–318. ISBN: 978-1-4008-8197-0. DOI: 10.1515/9781400881970-018.
- Smith, Alice, David Coit, Thomas Bäck, David Fogel und Zbigniew Michalewicz (Juli 1998). „Constraint-Handling Techniques - Penalty Functions“. In: *Handbook of Evolutionary Computation*. Bristol, U.K.: Institute of Physics Publishing und Oxford University Press.
- Strang, Gilbert (2010). *Wissenschaftliches Rechnen*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-78494-4. DOI: 10.1007/978-3-540-78495-1.
- Sturm, Jörg und Lubos Pavlak (2014). „Reduzierung von Oxideinschlüssen in Aluminium-Zylinderköpfen durch virtuelle Versuchsplanung“. In: *Gießerei Rundschau* 61.9, S. 283–291.
- Tibshirani, Robert (Jan. 1996). „Regression Shrinkage and Selection Via the Lasso“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, S. 267–288. ISSN: 00359246. DOI: 10.1111/j.2517-6161.1996.tb02080.x.
- Tonn, Babette, Ludwig Tonn, Julius Gogolin, Nils S. Tiedje, Giovanni di Muoio, Phillip Lechner, Christoph Hartmann, Patricia Erhard, Wolfram Volk, Adam P. Fros und Martin Fehlbier (2021). „Anwendungsbeispiele zur Prozessoptimierung“. In: *GIESSEREI* 108.7, S. 34–42. ISSN: 0016-9765.
- Unbehauen, Rolf (2002). *Systemtheorie. 1: Allgemeine Grundlagen, Signale und lineare Systeme im Zeit- und Frequenzbereich*. 8., korrig. Aufl. München Wien: Oldenbourg. ISBN: 978-3-486-25999-5.
- Velazquez Blandino, Mauricio, Ingo Wagner und Jörg Sturm (2019). „Virtuelle statistische Versuchsplanung und Optimierung der Kernherstellung“. In: *Gießerei-Praxis* 70.3, S. 17–25. ISSN: 0016-9781.
- Vincent, Pascal, Hugo Larochelle, Yoshua Bengio und Pierre-Antoine Manzagol (2008). „Extracting and composing robust features with denoising autoencoders“. In: *Proceedings of the 25th international conference on Machine learning - ICML '08*. Helsinki, Finland: ACM Press, S. 1096–1103. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390294.
- Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa und Paul van Mulbregt (2020). „SciPy 1.0: fundamental algorithms for scientific computing in Python“. In: *Nature Methods* 17.3, S. 261–272. ISSN: 1548-7091. DOI: 10.1038/s41592-019-0686-2.
- Wagner, Ingo und Jörg Sturm (2019). „Virtuelle Optimierung in der Kernfertigung – Innovative Methodik zur systematischen Werkzeugauslegung und Prozessoptimierung“. In: *GIESSEREI Special* 106.1, S. 70–77. ISSN: 2512-3157.

- Witt, Susanne (2018). „Online-Zustandstracking mit datengetriebenen Prozessmodellen“. Diss. Karlsruher Institut für Technologie (KIT). DOI: 10.5445/IR/1000087161.
- Witten, Ian H., Christopher J. Pal, Eibe Frank und Mark A. Hall (2017). *Data mining: Practical machine learning tools and techniques*. Fourth edition. Cambridge, MA: Morgan Kaufmann. ISBN: 978-0-12-804291-5.
- Wolpert, David H. (Jan. 1992). „Stacked generalization“. In: *Neural Networks* 5.2, S. 241–259. ISSN: 08936080. DOI: 10.1016/S0893-6080(05)80023-1.
- Wüller, Eike (2015). *Einfluss von Legierungselementen auf die Gefügeausbildung und mechanische Eigenschaften von Gusseisen mit Kugelgraphit*. Aachen: Giesserei Institut der RWTH Aachen. ISBN: 978-3-944601-06-9.
- Xu, Shu, Bo Lu, Michael Baldea, Thomas F. Edgar, Willy Wojsznis, Terrence Blevins und Mark Nixon (Jan. 2015). „Data cleaning in the process industries“. In: *Reviews in Chemical Engineering* 31.5. ISSN: 2191-0235, 0167-8299. DOI: 10.1515/revce-2015-0022.
- Zou, Hui und Trevor Hastie (Apr. 2005). „Regularization and variable selection via the elastic net“. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, S. 301–320. ISSN: 1369-7412, 1467-9868. DOI: 10.1111/j.1467-9868.2005.00503.x.
- Zumsande, Johannes, Karl-Philipp Kortmann, Mark Wielitzka, Christian Hansen und Tobias Ortmaier (2019). „Mehrkriterielle Prozessoptimierung in der digitalisierten Produktion“. In: *Fachtagung Mechatronik 2019*. DOI: 10.17619/UNIPB/1-785.

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub | universitäts
bibliothek

Diese Dissertation wird via DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt und liegt auch als Print-Version vor.

DOI: 10.17185/duepublico/78268

URN: urn:nbn:de:hbz:465-20230420-080858-0



Dieses Werk kann unter einer Creative Commons Namensnennung - Keine Bearbeitungen 4.0 Lizenz (CC BY-ND 4.0) genutzt werden.