STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG

Can Kaymakci

»Gestaltung eines auf dem föderierten Lernen basierenden Systems zur energetischen Anomalieerkennung in Produktionsprozessen«

Fraunhofer IPA

Universität Stuttgart

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG BAND 189

Can Kaymakci

»Gestaltung eines auf dem föderierten Lernen basierenden Systems zur energetischen Anomalieerkennung in Produktionsprozessen«

Herausgeber

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl[1,2]
Univ.-Prof. Dr.-Ing. Dipl.-Kfm. Alexander Sauer[1,3]
Univ.-Prof. Dr.-Ing. Kai Peter Birke[4]
Univ.-Prof. Dr.-Ing. Marco Huber[1,2]

[1] Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
[2] Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart
[3] Institut für Energieeffizienz in der Produktion (EEP) der Universität Stuttgart
[4] Institut für Photovoltaik (ipv) der Universität Stuttgart

Kontaktadresse:

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA
Nobelstr. 12
70569 Stuttgart
Telefon 0711 970-1100
info@ipa.fraunhofer.de
www.ipa.fraunhofer.de

Bibliographische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliographie; detaillierte bibliografische Daten sind im Internet über
http://dnb.de abrufbar.

Zugl.: Stuttgart, Univ., Diss., 2025

D 93

2025

Druck und Weiterverarbeitung:

Fraunhofer-Druckerei, Stuttgart
Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

Dieses Werk steht, soweit nicht gesondert gekennzeichnet,
unter folgender Creative-Commons-Lizenz:
Namensnennung – Nicht kommerziell – Keine Bearbeitungen
International 4.0 (CC BY-NC-ND 4.0).

Gestaltung eines auf dem föderierten Lernen basierenden Systems zur energetischen Anomalieerkennung in Produktionsprozessen

Von der Fakultät Energie-, Verfahrens- und Biotechnik der Universität Stuttgart zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Can Kaymakci
aus Bietigheim-Bissingen

Hauptberichter:	Prof. Univ.-Prof. Dr.-Ing. Dipl.-Kfm. Alexander Sauer
Mitberichter:	Prof. Univ.-Prof. Dr.-Ing. Matthias Weigold
Tag der mündlichen Prüfung:	24.03.2025

Institut für Energieeffizienz in der Produktion
der Universität Stuttgart
2025

Vorwort des Autors

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA und dem Institut für energieeffiziente Produktion (EEP) der Universität Stuttgart. Besonderer Dank gilt Herrn Univ.-Prof. Dr.-Ing. Dipl.-Kfm. Alexander Sauer für die Unterstützung meiner wissenschaftlichen Arbeit und die Übernahme des Hauptberichts. Herrn Univ.-Prof. Dr.-Ing. Matthias Weigold danke ich für die Übernahme des Mitberichts zu dieser Arbeit. Ebenfalls danken möchte ich meinen Kollegen am Fraunhofer IPA, insbesondere Dr.-Ing. Dennis Bauer für die vielen interessanten und wegweisenden Diskussionen zu der Themenstellung dieser Arbeit sowie meinen Abteilungsleitern Dr.-Ing. Sebastian Weckmann und Dr.-Ing. Timm Kuhlmann für das Vertrauen und die fortwährende Unterstützung. Ein Großteil dieser Arbeit entstand in dem Forschungsprojekt DigiWeld und ich danke vor allem Stefan Rohleder für den interessanten fachlichen Austausch und den Lehrtagen im Schweißlabor. Außerdem danken möchte ich Dr. Simon Wenninger für seine hilfreichen Anmerkungen und die vielen Gespräche und Veröffentlichungen während der Erstellung der Arbeit. Zudem danke ich allen Kolleginnen und Kollegen der Abteilung industrielle Energiesysteme für die großartige und spaßige Zusammenarbeit. Weiterhin bedanke ich mich bei Lukas Baur, Philipp Pelger und bei allen Studentinnen und Studenten, die mich während meiner Zeit am Fraunhofer IPA unterstützt haben. Großer Dank gilt meinen Eltern, die mich immer gefördert und unterstützt haben sowie meinen Freunden, die mich immer motiviert haben. Carina danke ich für ihre mentale Unterstützung, ihr Verständnis und die vielen schönen Momente neben der Arbeit.

Inhaltsverzeichnis

Abkürzungsverzeichnis V

Symbolverzeichnis VII

Abbildungsverzeichnis XI

Tabellenverzeichnis XVII

Formelverzeichnis XIX

Kurzfassung XXI

Abstract XXIII

1 Einleitung 1

1.1 Ausgangssituation 1

1.2 Praktische und theoretische Problemstellung 3

1.3 Zielsetzung und Forschungsfrage 4

1.4 Forschungsmethode und wissenschaftstheoretische Einordnung 6

1.5 Aufbau der Arbeit 9

2 Grundlagen und Begriffsdefinitionen 11

2.1 Produktion und Fertigung 11

2.1.1 Begriffsdefinition 11

2.1.2 Maschinen- und Betriebsdaten 14

2.1.3 Energiedatenanalyse in Produktionsprozessen 15

2.2 Energetische Anomalieerkennung 16

2.2.1 Begriffsdefinition 16

2.2.2 Szenarien 18

2.2.3 Verfahren 19

2.2.4 Verfahrensauswahl ... 22
2.3 Föderiertes Lernen ... 25
2.3.1 Begriffsdefinition ... 25
2.3.2 Föderierte Lernsysteme ... 27
2.3.3 Referenzarchitekturen ... 30
2.4 Zwischenfazit ... 32
3 Stand der Technik und Wissenschaft ... 33
3.1 Vorgehen der systematischen Literaturanalyse ... 33
3.2 Energetische Anomalieerkennung in Produktionsprozessen ... 34
3.3 Föderiertes Lernen zur Anomalieerkennung in Produktionsprozessen ... 39
3.4 Zwischenfazit ... 44
4 Anforderungen und Handlungsbedarf ... 45
4.1 Definition und Ableitung der Anforderungen ... 45
4.2 Einordnung der bestehenden Ansätze ... 49
4.2.1 Energetische Anomalieerkennung ... 49
4.2.2 Föderiertes Lernen zur Erkennung von Anomalien ... 53
4.3 Zwischenfazit und Handlungsbedarf ... 56
5 Systemgestaltung ... 59
5.1 Methodisches Vorgehen ... 59
5.2 Architektur ... 61
5.2.1 Anomaliedetektor ... 61
5.2.2 FL-basiertes System zur energetischen Anomalieerkennung ... 65
5.3 Methode ... 70
5.3.1 Entwicklung des Modells ... 73

5.3.2 Implementierung des Anomaliedetektors ... 96
5.3.3 Parameteroptimierung des föderierten Lernens ... 105
5.4 Zwischenfazit ... 119
6 Fallbeispiele ... 123
6.1 Erkennung von Bedienfehlern beim Lichtbogenschweißen ... 123
6.1.1 Anomalien beim Lichtbogenschweißen ... 124
6.1.2 Entwicklung des Modells zur energetischen Anomalieerkennung ... 125
6.1.3 Implementierung des Anomaliedetektors ... 138
6.1.4 Parameteroptimierung des föderierten Lernens ... 145
6.2 Erkennung von Druckluftleckagen ... 160
6.2.1 Druckluftleckagen in Produktionsprozessen ... 160
6.2.2 Entwicklung des Modells zur energetischen Anomalieerkennung ... 162
6.2.3 Implementierung des Anomaliedetektors ... 172
6.2.4 Parameteroptimierung des föderierten Lernens ... 179
6.3 Zwischenfazit ... 193
7 Reflexion ... 199
7.1 Bewertung der Anforderungserfüllung ... 199
7.1.1 Anforderungen an die energetische Anomalieerkennung ... 199
7.1.2 Anforderungen an das föderierte Lernen ... 201
7.1.3 Anforderungen an die verteilte Systemarchitektur ... 202
7.2 Beantwortung der Forschungsfrage ... 203
8 Zusammenfassung und Ausblick ... 207
Literaturverzeichnis ... 209
Anhang ... 223

A.1 Detaillierte Beschreibung der Modelle für das Modelltraining 223

A.2 Detaillierte Beschreibung der Augmentationsmechanismen 224

A.3 Ergebnisse der Modellauswahl für ersten Anwendungsfall..... 225

A.4 Ergebnisse der Modellauswahl für zweiten Anwendungsfall 228

Abkürzungsverzeichnis

CNC	Rechnergestützte Numerische Steuerung
CNN	Faltendes Neuronales Netzwerk
CRISP-DM	*Cross Industry Standard Process for Data Mining*
DSRM	Design Science Research Methode
DVE	Datenverarbeitungseinheit
EDMS	Energiedatenmanagementsystem
EEG	Elektroenzephalografie
EN	Europäische Norm
ERP	Enterprise Resource Planning
FCN	Voll Gefaltenes Neuronales Netzwerk
FedAvg	Federated Averaging
FedOpt	Federated Optimization
FLEA	FL-System zur energetischen Anomalieerkennung
FL-Systeme	*Föderierte Lernsysteme*
FTL	Föderiertes Transfer-Lernen
GRU	Gated Recurrent Unit
HDF5	Hierarchical Data Format 5
ID	Identifikator
IEEE	Institute of Electrical and Electronics Engineers
ISO	Internationale Organisation für Normung
IT	Informationstechnologie
JSON	JavaScript Object Notation
KI	Künstliche Intelligenz
KNN	Künstliche Neuronale Netze
kW	Kilowatt
LEI	Leistungseffizienzindex

LSTM	Long Short-Term Memory
ML-Modell	Maschinelles Lernmodell
ONNX	Open Neural Network Exchange
OPC	Open Platform Communications
PEMD	Performance, Environment, Measure, Data
PTE	*Parzen Tree Estimator*
QFedAvg	Quantile Federated Averaging
RFID	Radiofrequenz-Identifikation
SPS	Speicherprogrammierbare Steuerung
SSAE	Gestapelte Spärliche Autoencoder
u. a.	unter anderem
UA	Unified Architecture
VDI	Verein Deutscher Ingenieure e.V.
z. B.	zum Beispiel

Symbolverzeichnis

A	Menge der zu untersuchenden Abtastraten
a	Abtastrate
$\mathbf{B}$	Kontextbasierte Betriebsdaten
$\boldsymbol{b}_i$	Betriebsdatensatz des i-ten Prozessschritts
$\boldsymbol{b}_t$	Betriebsdaten zum Zeitpunkt t
$\mathbf{D}_{train}$	Trainingsdatensatz
$\mathbf{D}_{test}$	Testdatensatz
E	Evaluationsmetrik
$E_{dezentral}$	Durchschnittliche Evaluationsmetrik von lokalen Modellen
E_i	Evaluationsmetrik in der i-ten Runde
E_{komb}	Kombinierte Evaluationsmetrik aus Trainingszeit und $F1\text{-}Score$
$\overline{E_R}$	Arithmetisches Mittel der Evaluationsmetrik nach R Runden
E_S	Standardisierte Evaluationsmetrik
e_w	Endzeitpunkt des w-ten Zeitfensters
$\mathbf{EZ}$	Energiezeitreihe
$\mathbf{ez}_j$	Univariate Energiezeitreihe des j-ten Sensors
$ez_{j,t}$	Messwert des j-ten Sensors zum Zeitpunkt t
$\boldsymbol{ez}_{w,s}$	Energiezeitreihe des s-ten Sensors im w-ten Fenster
$\mathbf{EZG}$	Energiezeitreihengruppe
$\mathbf{EZG}_{fenster}$	Fensterbasierte Energiezeitreihengruppe

$\mathbf{EZG}_{kontext}$	Kontextbasierte Energiezeitreihengruppe
f	Funktion zur Merkmalsextraktion
$Fair_i$	Fairness in der i-ten Runde
$\overline{Fair_R}$	Arithmetisches Mittel der Fairness nach R Runden
ID_w	Eindeutige Identifikation des w-ten Zeitfensters
I_{training}	Eindeutige Zeitfenster innerhalb der Trainingsgruppe
I_{test}	Eindeutige Zeitfenster innerhalb der Testgruppe
k	Zeitfenster innerhalb der Energiezeitreihengruppe
K	Menge aller Zeitfenster
$\mathbf{KV}$	Matrix der Kreuzvalidierungsergebnisse
kv_j	j-ter Durchlauf der Kreuzvalidierung
LEI	Leistungseffizienzindex
LEI_{max}	Maximaler Leistungseffizienzindex
LEI_{max}	Minimaler Leistungseffizienzindex
$LEI_{a,m}$	Leistungseffizienzindex des Modells m mit der Abtastrate a
M_i	Trainiertes Modell aus der i-ten Modellarchitektur
M_{max}	Modell mit dem höchsten LEI
$\mathbf{MT}$	Merkmalstabelle
$\boldsymbol{m}_i$	Merkmale aus dem Betriebsdatensatz $\boldsymbol{b}_i$
p	Prozentualer Anteil des Testdatensatzes
r	Zufallszahl
S	Anzahl an Sensoren
$sLEI$	Skalierter Leistungseffizienzindex

st_w	Startzeitpunkt des w-ten Zeitfensters
T	Gesamtlänge der Zeitreihe
t	Zeitpunkt
t_S	Standardisierte Trainingszeit
$t_{\text{start},i}$	Startzeitpunkt für den i-ten Prozessschritt
$t_{\text{end},i}$	Endzeitpunkt für den i-ten Prozessschritt
w	Fenstergröße
α	Gewichtungsfaktor für den Einfluss der Evaluationsmetriken
β	Gewichtungsfaktor für den Einfluss der Trainingszeiten
μ_E	Arithmetisches Mittel der Evaluationsmetriken
σ_E	Standardabweichung der Evaluationsmetriken
μ_t	Arithmetisches Mittel der Trainingszeiten
σ_t	Standardabweichung der Trainingszeiten
θ_s	Konfigurationsparameter des Systems
θ_c	Konfigurationsparameter der Clients

Abbildungsverzeichnis

Abbildung 1-1 Umfrage zu Anwendungsfeldern von Methoden der künstlichen Intelligenz in der Produktion (VDI Zentrum Ressourceneffizienz 2021, S. 48) ... 2

Abbildung 1-2 Darstellung des Forschungsrahmens angelehnt an Ulrich et al. (1976, S.305) ... 7

Abbildung 1-3 Aufbau der Arbeit ... 10

Abbildung 2-1 Produktionssystem nach Westkämper (2006, S.195) ... 12

Abbildung 2-2 Kategorisierung von Anomalien nach Ruff et al. (2020) ... 17

Abbildung 2-3 Kategorisierung der Szenarien für die Anomalieerkennung ... 18

Abbildung 2-4 Prozess des föderierten Lernens, angelehnt an McMahan et al. (2016, S.5) ... 26

Abbildung 2-5 Grundlegende Systemarchitektur von FL nach McMahan et al. (2016) ... 28

Abbildung 2-6 Darstellung der Unterschiede zwischen horizontalem und vertikalem Lernen ... 29

Abbildung 2-7 Vereinfachte Darstellung der Referenzarchitektur aus der IEEE-Richtlinie 3652.1-2020 ... 31

Abbildung 4-1 Bewertung der Ansätze zur energetischen Anomalieerkennung in Produktionsprozessen ... 51

Abbildung 4-2 Bewertung der Ansätze zu FL für die Anomalieerkennung in Produktionsprozessen ... 55

Abbildung 5-1 Exemplarische Darstellung eines Anomaliedetektors ... 62

Abbildung 5-2 Beispielhafte Darstellung eines auf dem FL basierenden Systems zur energetischen Anomalieerkennung ... 67

Abbildung 5-3 Methode für die Auslegung eins FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen . 72

Abbildung 5-4 Phase I der Methode für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen 73

Abbildung 5-5 Ablaufdiagramm der Aufbereitung der kontextbasierten Energiezeitreihe 79

Abbildung 5-6 Schematischer Aufbau der kontextbasierten Energiezeitreihe 81

Abbildung 5-7 Darstellung der Aufteilung der kontinuierlichen Zeitreihe in identische Zeitfenster 83

Abbildung 5-8 Darstellung der Aufteilung der kontinuierlichen Zeitreihe in kontextbasierte Zeitfenster 84

Abbildung 5-9 Ablaufdiagramm zur Extraktion von Merkmalen aus der Energiezeitreihengruppe 85

Abbildung 5-10 Exemplarische merkmalsbasierte Sicht mit identischen Zeitfenstern 86

Abbildung 5-11 Ablaufdiagramm zur Verwendung der Augmentation im Trainingsprozess 91

Abbildung 5-12 Exemplarischer Vergleich fünf beispielhafter Modelle auf Grundlage von augmentierten Daten 93

Abbildung 5-13 Exemplarische Heatmap des sLEI der unterschiedlichen Modell-Abtastrate-Kombinationen 96

Abbildung 5-14 Phase II der Methode für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen 97

Abbildung 5-15 Ablaufdiagramm zur Entwicklung der Datenpipeline des Anomaliedetektors 98

Abbildung 5-16 Ablaufdiagramm zur Entwicklung von Agenten für den Anomaliedetektor 102

Abbildung 5-17 Ablaufdiagramm zur Bereitstellung des Anomaliedetektors in einer lauffähigen Umgebung 104

Abbildung 5-18 Phase III der Methode für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen 106

Abbildung 5-19 Ablaufdiagramm zur simulationsbasierten Optimierung der Konfigurationsparameter 110

Abbildung 5-20 Exemplarische Darstellung des Vergleichs der Evaluationsmetrik zwischen dezentralem Training, zentralem Training und FL... 114

Abbildung 5-21 Exemplarische durchschnittliche Entwicklung des F1-Scores und Fairness pro Runde..... 115

Abbildung 5-22 Exemplarische Verteilung des F1-Scores..... 116

Abbildung 5-23 Mittelwert der F1-Scores für jeden Aggregationsmechanismus 117

Abbildung 5-24 Exemplarische Darstellung der Aggregationsmechanismen in Relation zu F1-Score und Fairness..... 118

Abbildung 5-25 Exemplarische Korrelationsmatrix dargestellt als Heatmap der Konfigurationsparameter und der definierten Leistungsmetriken 119

Abbildung 6-1 Darstellung des Versuchsaufbaus des Laborexperiments 129

Abbildung 6-2 Visualisierung der gemessenen Größen einer normalen Schweißung..... 130

Abbildung 6-3 Visualisierung der gemessenen Größen einer fehlerhaften Schweißung..... 131

Abbildung 6-4 Vergleich der trainierten Modelle auf Grundlage von augmentierten Daten 135

Abbildung 6-5 Visualisierung der F1-Scores in Abhängigkeit der Abtastrate und Modelle 136

Abbildung 6-6 Heatmap des sLEI der unterschiedlichen Modell-Abtastrate-Kombinationen 137

Abbildung 6-7 Konzept des Anomaliedetektors für die Erkennung von Bedienfehlern beim Lichtbogenschweißen 139

Abbildung 6-8 Darstellung des Vergleichs der Evaluationsmetrik zwischen dezentralem, zentralem Training und FL 149

Abbildung 6-9 Durchschnittliche Entwicklung des F1-Scores und Fairness pro Runde 150

Abbildung 6-10 Verteilung des F1-Scores 151

Abbildung 6-11 Mittelwert der F1-Scores für jeden Aggregationsmechanismus 152

Abbildung 6-12 Darstellung der einzelnen Strategien in Relation zu F1-Score und Fairness 153

Abbildung 6-13 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit der Epochen 154

Abbildung 6-14 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit von der Batchgröße 156

Abbildung 6-15 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit des Anteils der Clients 157

Abbildung 6-16 Korrelationsmatrix dargestellt als Heatmap der Konfigurationsparameter und der definierten Leistungsmetriken 159

Abbildung 6-17 Darstellung des Versuchsaufbaus des Laborexperiments 165

Abbildung 6-18 Visualisierung des Massenstroms bei unterschiedlichen Lochgrößen 166

Abbildung 6-19 Visualisierung der F1-Scores in Abhängigkeit von der Abtastrate und der Modelle 170

Abbildung 6-20 Heatmap des LEI der unterschiedlichen Modell-Abtastrate-Kombinationen 171

Abbildung 6-21 Konzept des Anomaliedetektors für die Erkennung von Druckluftleckagen bei pneumatischen Aktorbewegungen..... 173

Abbildung 6-22 Darstellung des Vergleichs der Evaluationsmetrik zwischen dezentralem Training, zentralem Training und FL..... 182

Abbildung 6-23 Durchschnittliche Entwicklung von F1-Scores und Fairness pro Runde..... 183

Abbildung 6-24 Verteilung des F1-Scores 184

Abbildung 6-25 Durchschnittliche Entwicklung des F1-Scores für jeden Aggregationsmechanismus 186

Abbildung 6-26 Darstellung der einzelnen Strategien in Relation zu F1-Score und Fairness..... 187

Abbildung 6-27 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit von den Epochen..... 188

Abbildung 6-28 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit von der Batch-Größe 189

Abbildung 6-29 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit des Anteils der Clients 190

Abbildung 6-30 Korrelationsmatrix dargestellt als Heatmap der Konfigurationsparameter, F1-Score und Fairness 192

Tabellenverzeichnis

Tabelle 1 Exemplarische Darstellung des PEMD-Frameworks 74

Tabelle 2 Exemplarische Definition der Konfigurationsparameter für die Simulation von FL 109

Tabelle 3 Darstellung des PEMD-Frameworks für die Fallstudie 126

Tabelle 4 Definition der Konfigurationsparameter für die Simulation von FL 146

Tabelle 5 Ergebnis der bayesschen Optimierung mit dem Tree-Parzen Estimator 148

Tabelle 6 Darstellung des PEMD-Frameworks für die Fallstudie 163

Tabelle 7 Definition der Konfigurationsparameter für die Simulation von FL 180

Tabelle 8 Ergebnis der bayesschen Optimierung mit dem Tree-Parzen Schätzer 181

Tabelle 9 Ausführliche Darstellung der genutzten ML-Modelle für das Modelltraining 223

Tabelle 10 Auflistung der genutzten Augmentationsmechanismen zur Generierung von synthetischen Daten 224

Tabelle 11 Ergebnisse des Modelltrainings der jeweiligen Modelle (bestes Modell und Zeit) des ersten Anwendungsfalls zur Erkennung von Bedienfehlern beim Lichtbogenschweißen 225

Tabelle 12 Ergebnisse des Modelltrainings der jeweiligen Modelle (bestes Modell und Zeit) des zweiten Anwendungsfalls zur Erkennung von Druckluftleckagen bei pneumatischen Aktorbewegungen..... 228

Formelverzeichnis

Formel 2-1 Berechnung der Genauigkeit 23

Formel 2-2: Berechnung der Sensitivität 23

Formel 2-3 Berechnung der Präzision 23

Formel 2-4 Berechnung des F1-Scores 23

Formel 5-1 Univariate Energiezeitreihe eines Sensors 76

Formel 5-2 Multivariate Energiezeitreihe 77

Formel 5-3 Bestandteile eines kontextbasierten Betriebsdatensatzes von Prozessschritt i 77

Formel 5-4 Berechnung der Energiezeitreihengruppen 82

Formel 5-5 Berechnung der fensterbasierten Energiezeitreihengruppe 82

Formel 5-6 Berechnung der kontextbasierten Energiezeitreihengruppe ... 83

Formel 5-7 Berechnung der Merkmalstabelle 84

Formel 5-8 Extraktion der IDs 87

Formel 5-9 Aufteilen der IDs in eine Trainings- und Testgruppe 88

Formel 5-10 Trennung in Trainings- und Testdatensatz 88

Formel 5-11 Definition der Kreuzvalidierung anhand der unterschiedlichen Modelle und Abtastraten 90

Formel 5-12 Definition des Trainingsprozesses in jeder Kreuzvalidierungsrunde 90

Formel 5-13 Standardisierte Evaluationsmetrik 94

Formel 5-14 Standardisierte Trainingszeit 94

Formel 5-15 Gewichtete Berechnung des Leistungseffizienzindexes anhand der standardisierten Evaluationsmetrik und Trainingszeit 94

Formel 5-16 Identifikation des Modells mit dem höchsten sLEI 95

Formel 5-17 Berechnung der Fairness auf Grundlage der Evaluationsmetrik in einer bestimmten Trainingsrunde 112

Formel 5-18 Berechnung der durchschnittlichen Evaluationsmetrik für alle Clients in einer bestimmten Runde 112

Formel 5-19 Berechnung der durchschnittlichen Fairness in einer bestimmten Anzahl an Runden 112

Formel 5-20 Berechnung der durchschnittlichen Evaluationsmetrik in einer bestimmten Anzahl an Runden 112

Formel 5-21 Berechnung der kombinierten Metrik anhand der durchschnittlichen Evaluationsmetrik und der Fairness 112

Formel 5-22 Bayesscher Optimierungsprozess für die Identifikation der idealen Konfigurationsparameter 113

Formel 5-23 Berechnung der durchschnittlichen Evaluationsmetrik des dezentralen Trainings 113

Formel 5-24 Darstellung der Beziehung der unterschiedlichen Szenarien ... 114

Kurzfassung

Um die nationalen Effizienzziele zu erreichen, ist es unerlässlich, Energie und Material einzusparen. Die frühzeitige Erkennung von energetischen Anomalien in Produktionsprozessen kann hier mit Modellen aus dem Bereich des maschinellen Lernens erfolgen. Für das Training der Modelle werden von Anwendern von Maschinen und Anlagen innerhalb der Produktionsprozesse erfasste Daten benötigt, die viele Anwender in der Praxis nicht freigeben. Daher ist ein umfassendes Training und ein erfolgreiches Erkennen von Unregelmäßigkeiten aus der Perspektive des Maschinen- und Anlagenbauers nicht möglich. Diese Arbeit zielt darauf ab, ein auf dem föderierten Lernen basierendes System zur Erkennung von energetischen Anomalien in Produktionsprozessen zu entwickeln. Dieses System soll Produktionsunternehmen in die Lage versetzen energetische Anomalien zu erkennen, ohne dabei sensible Maschinen- und Betriebsdaten weiterzugeben. Daher adressiert diese Arbeit die Forschungsfrage, wie ein auf dem föderierten Lernen basierendes System zur energetischen Anomalieerkennung in Produktionsprozessen ausgelegt werden kann. Hierzu wird eine Architektur und eine Methode entwickelt, die sowohl die Identifikation der Datenverarbeitung, die Modellarchitektur als auch die Optimierung der Konfigurationsparameter des Systems umfasst. Die in der Arbeit definierten funktionalen Komponenten der Architektur können mit der Methode hergeleitet und instanziiert werden. Der vorgestellte Ansatz wird anhand zweier Anwendungsfälle validiert. Der erste Anwendungsfall befasst sich mit der Erkennung von Bedienfehlern beim Lichtbogenschweißen. Der zweite Anwendungsfall konzentriert sich auf die Erkennung von Druckluftleckagen, welche erhebliche Energieverluste verursachen können. In beiden Anwendungsfällen zeigt sich, dass eine Nutzung des auf dem föderierten Lernen basierenden Systems zur energetischen Anomalieerkennung eine Erkennung von Unregelmäßigkeiten in Produktionsprozessen ermöglicht, ohne dass sensible Maschinen- und Betriebsdaten zentral gespeichert werden.

Abstract

In order to achieve the national efficiency targets, it is essential to save energy and materials. The early detection of energy anomalies in production processes can be achieved using machine learning models. Training the models requires data recorded by users of machines and systems within the production processes, which many users do not release in practice. Therefore, comprehensive training and successful recognition of irregularities is not possible from the perspective of the industrial equipment and machinery manufacturer. This thesis aims to develop a method for designing a federated learning-based system for detecting energy anomalies in production processes. This system should enable manufacturing companies and users of industrial equipment and machinery to detect energy anomalies without disclosing sensitive machine and operational data. Therefore, this thesis addresses the research question of how a federated learning based system for energy anomaly detection in production processes can be designed. For this purpose, an architecture and a method are developed that includes the identification of the data processing, the model architecture and the optimization of the configuration parameters of the system. The instantiation of the functional components of the architecture defined in the thesis can thus be derived using the method. The approach presented is validated using two use cases. The first use case deals with the detection of operator errors during arc welding. The second use case focusses on the detection of compressed air leaks, which can cause considerable energy losses. Both use cases show that using the federated learning-based system for energy anomaly detection makes it possible to detect irregularities in production processes without storing sensitive machine and operating data centrally.

1 Einleitung

Das Kapitel führt in das Gebiet der energetischen Anomalieerkennung ein. Die Ausgangssituation beleuchtet die Herausforderung der Energieeffizienz in der Produktion, die durch die Erkennung von Fehlern und Qualitätsmängeln in Produktionsprozessen gesteigert werden kann. Anschließend wird die Zielsetzung und Forschungsfrage formuliert, wissenschaftlich eingeordnet und die verwendeten Forschungsmethoden vorgestellt.

1.1 Ausgangssituation

Mit einem Anteil von 25 % am gesamten Stromverbrauch in der EU bietet vor allem die produzierende Industrie ein großes Potenzial zur Reduktion der CO_2-Emissionen mittels Steigerung der Energieeffizienz (International Energy Agency 2023). Der nationale Anteil der produzierenden Industrie am Stromverbrauch liegt in Deutschland bei 44 % (BDEW 2023). Produzierende Unternehmen stehen daher vor großen Herausforderungen, denn sie müssen ressourcen- und energieeffizient herstellen, um ihre CO_2-Emissionen zu senken (Bauer et al. 2020, S. 105; Sauer et al. 2022, S. 148). Hier hat sich in den letzten Jahren u. a. der Einsatz von Methoden der künstlichen Intelligenz zur Steigerung der Energieeffizienz herauskristallisiert (Waltersmann et al. 2021, S. 19). Entsprechende Methoden können in unterschiedlichen Anwendungsbereichen der Produktion implementiert werden, um die Energieeffizienz zu steigern (siehe Abbildung 1-1).

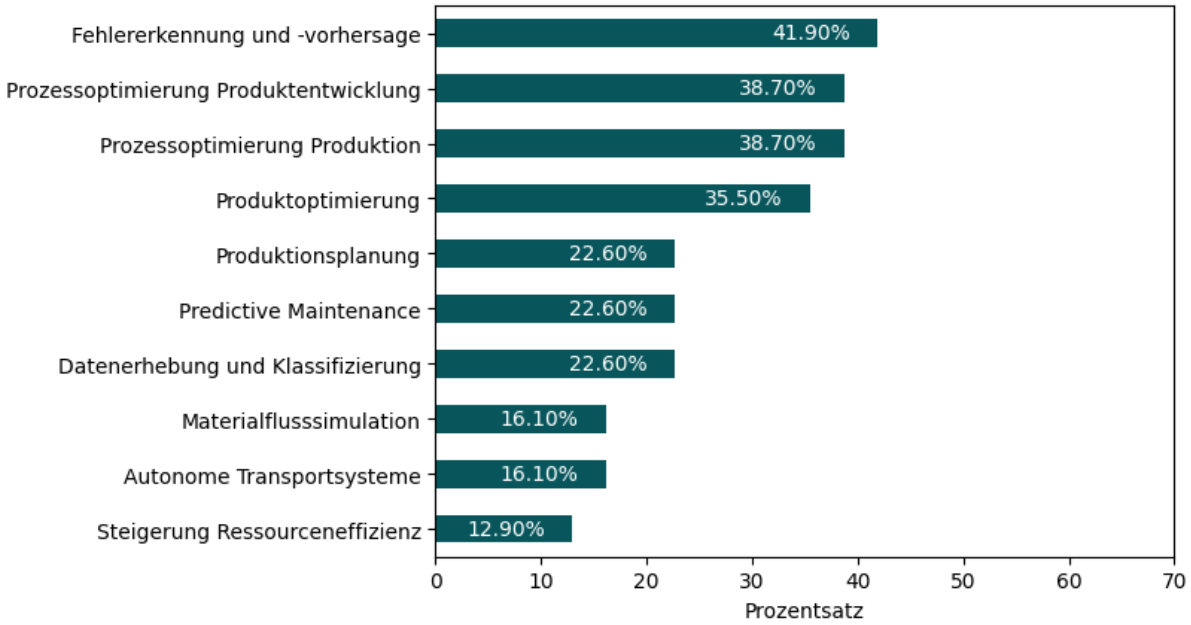


Abbildung 1-1 Umfrage zu Anwendungsfeldern von Methoden der künstlichen Intelligenz in der Produktion (Friedrich et al. 2021, S. 48)

Eine dieser Maßnahmen ist die Erkennung und Vorhersage von Fehlern im Energieverbrauch von Produktionsprozessen zur Steigerung der Ressourcen- und Energieeffizienz, sowie zur Reduktion der CO_2-Emissionen (Friedrich et al. 2021, S. 48). Im Forschungsbereich der künstlichen Intelligenz (KI) wird dies auch als *Anomalieerkennung* bezeichnet (Himeur et al. 2021). Anomalien zur Steigerung der Energieeffizienz, sowie die Erkennung von Anomalien in Produktionsprozessen mithilfe von Energiedaten werden in dieser Arbeit als *energetische Anomalien* bezeichnet. Der Einsatz datenbasierter Lernalgorithmen, insbesondere maschineller Lernmodelle (ML-Modelle), ermöglicht die Erfassung, Verarbeitung und Analyse von Energiedaten aus Produktionsprozessen. Dadurch können Unregelmäßigkeiten im Energieverbrauch sowie potenzielle Anomalien im Produktionsprozess identifiziert werden (Himeur et al. 2021, S. 2). Energetische Anomalien, die durch schlecht gewartete Anlagen oder auch falsche Parametrisierung von Produktionsprozessen verursacht werden und zu einem erhöhten Energieverbrauch führen, lassen sich dadurch identifizieren, um entsprechende Gegenmaßnahmen ergreifen zu können (Basora et al. 2019, S. 116).

Die unternehmensübergreifende Erkennung energetischer Anomalien ist insbesondere dann schwierig, wenn die Daten ausschließlich unternehmensintern verwaltet werden.

Dies liegt vor allem an der eingeschränkten Generalisierbarkeit solcher Daten (McMahan et al. 2016, S. 6). Lokale Datensätze sind zusätzlich in der Regel relativ klein und können daher nicht die Vielfalt und Menge abbilden, die für ein ML-Modell erforderlich sind (Maschler et al. 2019, S. 745). Einen Lösungsansatz bietet das in den letzten Jahren entwickelte Konzept des *föderierten Lernens* (FL) (McMahan et al. 2016): Statt Daten in einer zentralen Datenbank des Maschinen- und Anlagenbauers zu sammeln, können ML-Modelle direkt auf lokalen Maschinen- und Anlagen trainiert werden. Letztere nutzen hierfür lokal verfügbare Rechenkapazitäten, wobei nur die Modellparameter zentral gespeichert und aggregiert werden. Dies ermöglicht eine höhere Generalisierung und Robustheit der Modelle, ohne dass die Daten direkt mit dem Maschinen- und Anlagenbauer geteilt werden müssen.

1.2 Praktische und theoretische Problemstellung

Während unter Laborbedingungen die Erfassung, Verwaltung und Analyse von Daten sowie der Einsatz von ML-Modellen problemlos möglich ist, ergeben sich in der realen Produktionsumgebung vielfältige Herausforderungen, die bisher unzureichend berücksichtigt wurden. Eine kontinuierliche Übertragung der Maschinen- und Betriebsdaten sowie der Ergebnisse der energetischen Anomalieerkennung erfordern eine **ständige Konnektivität und niedrige Latenzzeiten**, die in den meisten Anwendungsfällen, z. B. in einer Produktionshalle ohne Mobilfunkempfang, nicht immer gegeben sind (Nayak et al. 2022, S. 19).

Die zweite wesentliche Herausforderung ist der **Schutz sensibler Daten der einzelnen Anwender**. Für einen Maschinen- und Anlagenbauer, der mithilfe der unternehmensübergreifend erfassten Maschinen- und Betriebsdaten den Energieverbrauch der Maschinen überwacht, um Fehler im Produktionsprozess zu erkennen, ist dies **ohne zentrale Verwaltung der Daten** bislang schwer umsetzbar. Energiedaten müssen hierfür dezentral durch Sensoren an der Maschine erfasst, aber zentral verarbeitet und analysiert werden (Nayak et al. 2022, S. 19).

Im Falle von Daten mit hoher Abtastrate und zentraler Verwaltung des Maschinen- und Anlagenbauers sind potenzielle Angreifer in der Lage, Informationen über Produktionsprozesse oder auch Probleme einzelner Unternehmen abzugreifen (Mohammed M. Alani & Mohamed Alloghani 2019, S. 130). Da dieses Risiko aus Sicht des Anwenders durch informationstechnischer Sicherheitsbedrohungen immer wahrscheinlicher wird, gilt es, neue Ansätze für industrielle Maschinen und Anlagen zu entwickeln, die eine Erkennung energetischer Anomalien in Produktionsprozessen ohne den Verlust von Informationen an Dritte ermöglichen (Mohammed M. Alani & Mohamed Alloghani 2019, S. 130).

In der Praxis liegt der Fokus stärker auf der initialen Anwendung von ML in der Produktion, während FL bislang nicht ausreichend berücksichtigt wird. Laut einer Studie des VDI Zentrums Ressourceneffizienz (Friedrich et al. 2021, S. 45) haben zwar 42 % der befragten Unternehmen Anwendungen mit ML implementiert, doch sehen viele den Implementierungsaufwand und fehlendes Fachwissen als Hemmnisse für eine erfolgreiche Anwendung. Da bereits die Implementierung von ML-Methoden in der Industrie als Herausforderung wahrgenommen wird, erscheint die Einführung von FL entsprechend komplexer. Ein Grund dafür ist das Fehlen wissenschaftlicher und methodischer Ansätze für die Einführung und Auslegung. Dies unterstreicht die Notwendigkeit, zielgerichtete Methoden für Unternehmen der produzierenden Industrie zu entwickeln, um die Potenziale der energetischen Anomalieerkennung mit Hilfe von FL auszuschöpfen.

1.3 Zielsetzung und Forschungsfrage

Im Anschluss an die Erörterung der Problemstellung erfolgt in diesem Abschnitt die Ableitung der Zielsetzung sowie der darauf aufbauenden Forschungsfragen der Arbeit. Zielsetzung dieser Arbeit ist die Entwicklung einer Methode zur Auslegung eines FL-basierten Systems zur Erkennung von energetischen Anomalien in Produktionsprozessen. Sie soll es Unternehmen ermöglichen, FL einzusetzen, um eine Erkennung energetischer Anomalien in Produktionsprozessen zu ermöglichen, ohne Kompromisse bei der

Weitergabe sensibler Daten eingehen zu müssen. Es kann in Teilziele aufgegliedert werden: Zunächst erfolgt eine **Analyse bestehender Ansätze** zur energetischen Anomalieerkennung und zum Einsatz von FL in Produktionsprozessen. Hierbei soll der Fokus auf der Identifikation der grundlegenden Komponenten wie auch Methoden der energetischen Anomalieerkennung und von FL in der Produktion liegen. Basierend auf den dabei gewonnenen Erkenntnissen sollen sodann **Anforderungen** an ein System zur energetischen Anomalieerkennung auf Basis von FL definiert werden. Der Schwerpunkt liegt auf der Systematisierung der identifizierten Anforderungen und der Bewertung des Standes der Technik im Hinblick auf die Erfüllung der Anforderungen. Für den Einsatz von FL und energetischer Anomalieerkennung in Produktionsprozessen sind je nach betrachtetem Prozess unterschiedliche Daten und Evaluationsmetriken, aber auch verschiedene Implementierungen notwendig. Daher soll eine **Methode zur Auslegung** eines FL-basierten Systems zur energetischen Anomalieerkennung entwickelt werden, die sich an den definierten Anforderungen orientiert. Dass das Ausrollen von Software auf Maschinen und Anlagen nicht trivial ist, macht eine Validierung des Nutzens des FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen erforderlich. Aufgrund dessen soll abschließend die entwickelte Methode **simulativ validiert** werden. Dabei werden verschiedene Szenarien und Parameterkonfigurationen berücksichtigt, um das System unter realen Bedingungen zu testen.

Vor dem Hintergrund der Problemstellung und der Zielsetzung lautet die zentrale Forschungsfrage dieser Arbeit wie folgt:

Wie kann ein auf dem föderierten Lernen basierendes System zur energetischen Anomalieerkennung in Produktionsprozessen ausgelegt werden?

Um die zentrale Forschungsfrage zielgerichtet aufzuarbeiten, empfiehlt sich eine Spezifizierung der zentral zu untersuchenden Aspekte. So kann die zentrale Forschungsfrage in Teilforschungsfragen untergliedert werden:

- Was sind die relevanten Bestandteile eines auf dem FL basierten Systems zur energetischen Anomalieerkennung eines Produktionsprozesses?
- Welche Anforderungen können für das FL zur energetischen Anomalieerkennung in Produktionsprozessen identifiziert werden?

- Wie kann das FL in die Auslegung eines Systems zur energetischen Anomalieerkennung in Produktionsprozessen integriert werden?
- Wie kann die simulative Validierung des entwickelten Systems zur energetischen Anomalieerkennung in Produktionsprozessen erfolgen?

1.4 Forschungsmethode und wissenschaftstheoretische Einordnung

Ulrich et al. (Ulrich & Hill 1976) kategorisieren die unterschiedlichen Disziplinen in der Wissenschaft zu Formal- und Realwissenschaften (siehe Abbildung 1-2). Die Definition und Herleitung der Begriffe wird wie folgt vorgenommen: *Formalwissenschaften* sind Wissenschaften, die sich mit der Erforschung formaler Systeme und deren Eigenschaften befassen. *Formale Systeme* sind Systeme, die auf formalen Sprachen und Logiken basieren, die sich durch bestimmte Regeln und Symbole auszeichnen. Die Formalwissenschaften umfassen Disziplinen wie Mathematik, Informatik und Logik. Sie entwickeln und nutzen abstrakte, formale Modelle der Realität, um allgemeingültige Aussagen über die Welt zu treffen. Die Formalwissenschaften nutzen somit Formalisierungen und Abstraktionen, um das Verständnis von realen Systemen und Phänomenen zu vertiefen. *Realwissenschaften* beziehen sich auf konkrete, empirische Beobachtungen und Messungen. Sie versuchen, allgemeingültige Theorien und Gesetze über die Natur der Welt und deren Funktionsweise zu entwickeln. Die Realwissenschaften werden zusätzlich in *Grundlagenwissenschaften* zur Erklärung empirischer Wirklichkeitsausschnitte (z. B. Naturwissenschaften) sowie in angewandte Handlungswissenschaften zur Analyse menschlicher Handlungsalternativen (z. B. Sozialwissenschaften) untergliedert. Die Ingenieurs- oder auch Technikwissenschaften werden an der Schnittstelle zwischen Grundlagenwissenschaften und angewandter Handlungswissenschaften angesiedelt. Sie dienen zur Erforschung und Entwicklung von Forschungsartefakten, die meist als technologische Lösung für ein spezifisches Forschungsproblem definiert sind. Da in dieser Arbeit eine Methode und Architektur zur Entwicklung einer technologischen Lösung für ein spezifisches Forschungsproblem

entwickelt wird, ist die Arbeit im Bereich der Technikwissenschaften anzusiedeln. Zusätzlich werden zur Adressierung des Forschungsproblems Methoden aus den Formalwissenschaften (z.B. Mathematik, Statistik und KI) herangezogen.

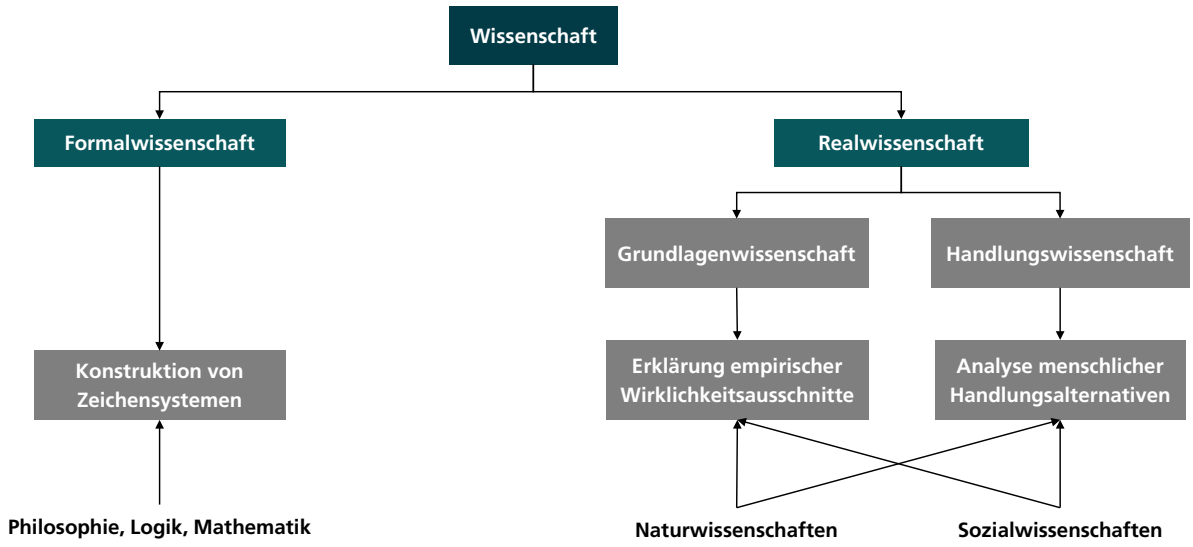


Abbildung 1-2 Darstellung des Forschungsrahmens angelehnt an Ulrich et al. (1976, S.305)

In Anlehnung an den *Design-Science-Research-Ansatz* der gestaltungsorientierten Wirtschaftsinformatik von Hevner et al. (Hevner et al. 2004) und dem Forschungszyklus nach Tomczak (Tomczak 1992) wird in dieser Arbeit eine Zielarchitektur und eine Methode als Forschungsartefakte anhand der iterativen Vorgehensweise in Form eines Gestaltungszyklus entwickelt. Eine Methode bezeichnet einen strukturierten Satz an Aktivitäten, die zur Erreichung eines Ziels eingesetzt wird. Die Beschreibung des Prozesses dient der Erläuterung, auf welche Weise ein bestimmtes Ergebnis erzielt werden kann (Offermann et al. 2010, S. 84). Der Begriff „Architektur" wird, anlehnend an Offermann et al. (2010, S.84) in der vorliegenden Arbeit in einem umfassenderen Sinne verwendet, nämlich als spezifische, strukturierte Lösung für ein System, das aus einer Reihe von Komponenten und deren Beziehungen besteht. Diese Architektur ist das konkrete Ergebnis, welches durch die Anwendung der Methode erreicht wird. Die in dieser Arbeit zu entwickelnde Zielarchitektur zeigt den praktischen Nutzen der Methode und dient als Referenz für ähnliche Implementierungen in der Industrie. Sie bietet eine Lösung, die von anderen Forschern und Praktiker nachvollzogen und auf andere Anwendungsfälle repliziert werden kann. Die Zyklen der Relevanz und der Stringenz komplementieren den

Gestaltungszyklus, indem sie anwendbares Wissen aus einer theoretischen Grundlage mit praktischen Bedürfnissen aus einer Realumgebung kombinieren und damit einen Beitrag zur Forschung leisten (Hevner et al. 2004).

Um den Forschungsprozess zu strukturieren, wird die *Design-Science-Research-Methode* (DSRM) übernommen, die von Peffers et al. eingeführt wurde und sechs Phasen umfasst: *Problemidentifikation*, *Zieldefinition*, *Design & Entwicklung*, *Demonstration*, *Evaluation* und *Kommunikation* (Peffers et al. 2007). Angelehnt an die DSRM wurde die in der Arbeit verwendeten Forschungsmethode auf den Forschungsprozess von Ulrich (1976) und Tomczak (1992) angepasst. Folglich wurden nicht alle Schritte strikt wie vorgeschlagen durchgeführt. Die erste Phase erfordert die Definition des spezifischen Forschungsproblems, das auf dem Anwendungsgebiet basiert. Wie in Abschnitt 1.1 beschrieben, bietet der Einsatz von ML zur Erkennung energetischer Anomalien in Produktionsprozessen für Maschinen- und Anlagenbauer großes Potenzial. Durch den Einsatz von FL können diese einen doppelten Mehrwert erzielen: Sie gewährleisten den Schutz der Daten vor möglichen Sicherheitsbedrohungen – insbesondere vor unberechtigtem Datenzugriff – und ermöglichen gleichzeitig die Optimierung von Produktionsprozessen, was zu einer effizienteren Nutzung von Ressourcen wie Energie und Material führt. Für die Praxis wird ein FL-basiertes System zur energetischen Anomalieerkennung in Produktionsprozessen benötigt. Diese Arbeit adressiert diese Lücke durch die Entwicklung einer Architektur und Methode und dient als Grundlage für die erfolgreiche Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen.

Die Design- und Entwicklungsphase umfasst sowohl die Bestimmung der Funktionalität der Artefakte in Form einer Architektur und einer Methode zur Entwicklung des Systems als auch die Erstellung des eigentlichen Artefakts (Peffers et al. 2007). Die Design- und Entwicklungsphase ist somit dem Gestaltungszyklus von Hevner et al. (2004) zuzuordnen. Nach der Entwicklung des Forschungsartefakts erfolgt eine Validierung anhand zweier Anwendungsfälle mit real gemessenen Energieverbrauchsdaten von Produktionsprozessen, um die Wirksamkeit der Methode zur Lösung des

zugrundeliegenden Problems der energetischen Anomalieerkennung in Produktionsprozessen ohne zentralen Datenaustausch zu demonstrieren.

Die Überführung der Artefakte in den Relevanzzyklus und damit an Forscher und Praktiker in Unternehmen des Maschinen- und Anlagenbaus zielt darauf ab, die bestehende theoretische Wissensbasis zu erweitern, um dadurch die Integration der entwickelten Artefakte in die praktische Realumgebung und den entsprechenden Anwendungsbereich hinein zu ermöglichen.

1.5 Aufbau der Arbeit

Im Folgenden wird der weitere Aufbau der Arbeit anhand der jeweiligen Kapitel und der vorgestellten Forschungsmethode aus Abschnitt 1.4 in Abbildung 1-3 dargestellt. Kapitel 2 umfasst die Konzepte und Grundlagen der Produktion, die Anomalieerkennung durch die ML-Methoden und die Grundlagen von FL. Darauf aufbauend werden in Kapitel 3 verwandte Arbeiten sowie der Stand der Technik hinsichtlich der energetischen Anomalieerkennung und des FL in der wissenschaftlichen Literatur analysiert. Kapitel 4 identifiziert und definiert die relevanten Anforderungen für das Forschungsartefakt und ordnet die bestehenden Arbeiten aus Kapitel 3 hinsichtlich der Anforderungen ein. Hierbei erfolgt die Identifikation von Schwachstellen bestehender Ansätze für die Abgrenzung dieser Arbeit. Gegenstand von Kapitel 5 ist die Entwicklung der Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen. In Kapitel 6 wird die FL-basierte energetische Anomalieerkennung anhand zweier praxisnaher Anwendungsfälle implementiert und validiert. Kapitel 7 überprüft die Erfüllung der in Kapitel 4 definierten Anforderungen des in Kapitel 5 erstellten Forschungsartefakts und beantwortet die eingangs formulierten Forschungsfragen. Abschließend fasst Kapitel 8 die Arbeit zusammen und gibt einen Ausblick für weitere Forschungsarbeiten in diesem Bereich.

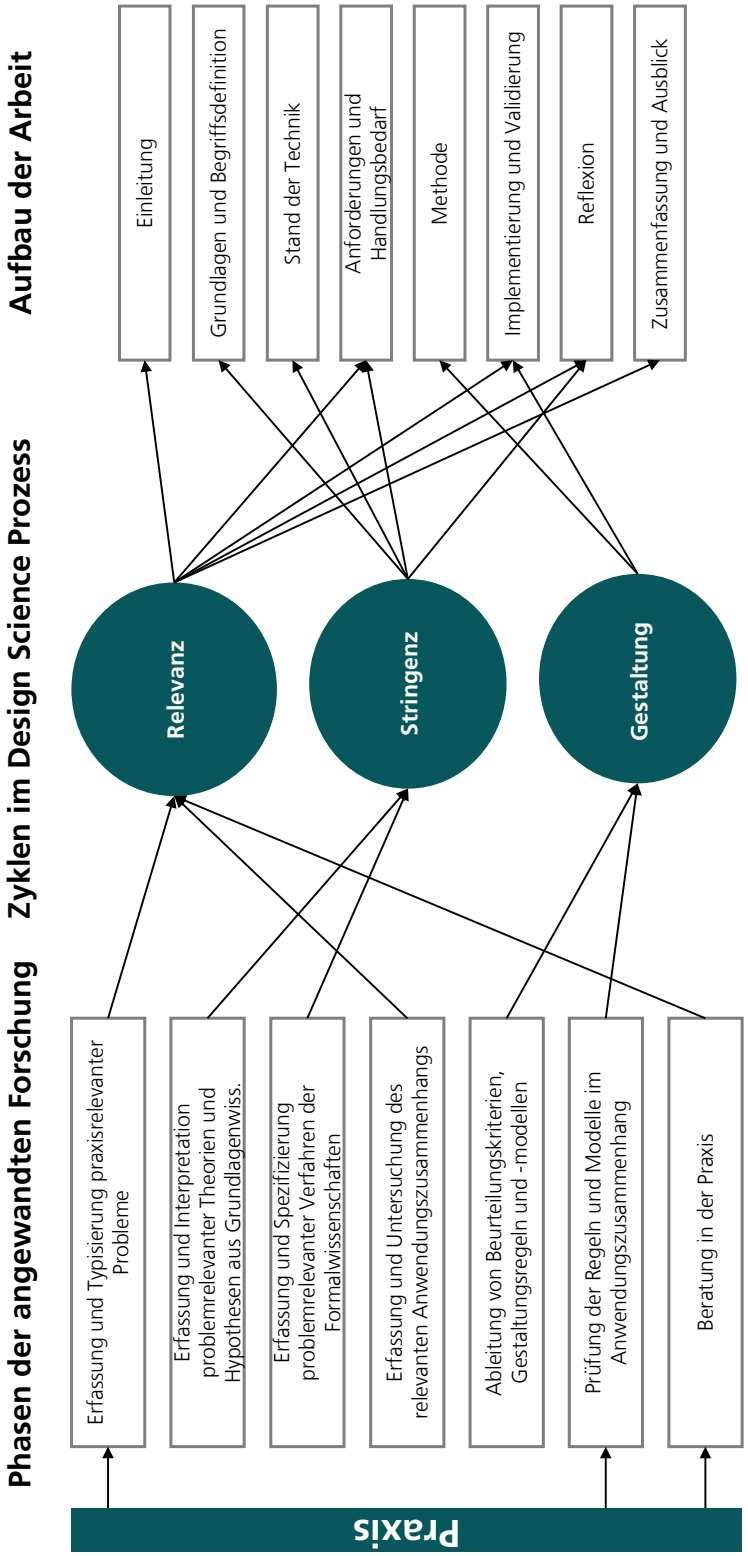


Abbildung 1-3 Aufbau der Arbeit

2 Grundlagen und Begriffsdefinitionen

Zur Beantwortung der Forschungsfragen hinsichtlich der Entwicklung einer Methode zur Auslegung eines auf dem föderierten Lernen basierenden Systems zur energetischen Anomalieerkennung in Produktionsprozessen sollen in diesem Kapitel die notwendigen Grundlagen erarbeitet und für das weitere Verständnis der Arbeit in den vorliegenden Kontext gesetzt werden. Im ersten Abschnitt werden die Grundlagen der Produktion erarbeitet und deren Hauptkomponenten definiert. Zusätzlich umfasst dieser die Grundlagen der Erkennung von Anomalien und den Ansatz des FL.

2.1 Produktion und Fertigung

Dieses Kapitel widmet sich dem Thema der Produktion im Zusammenhang mit der Erkennung energetischer Anomalien in Produktionsprozessen. Für ein klares Verständnis des Untersuchungsgegenstandes werden zunächst auf begrifflicher Ebene die relevanten Systemgrenzen definiert. Anschließend erfolgt eine Definition der Produktionsprozesse. Des Weiteren werden die Charakteristika der Datenerfassung und insbesondere der Energiedatenanalyse in Produktionsprozessen herausgearbeitet. Diese Informationen bilden die Grundlage für die Betrachtung energetischer Anomalien in Produktionsprozessen.

2.1.1 Begriffsdefinition

Die grundlegende Definition der Produktion nach Dyckhoff ist die zeitliche, qualitative oder räumliche Transformation von Objekten zur Befriedigung fremder Bedürfnisse (Dyckhoff & Spengler 2010, S. 5). Groover und Westkämper spezifizieren dies, indem das Ziel der Produktion als Transformation von Materialien zu Komponenten mit einem

höheren Wert definiert wird (Westkämper 2006, S. 34; Groover 2020, S. 3). Dies beinhaltet die Herstellung von Produkten in der richtigen Art und Menge zum richtigen Zeitpunkt in einer spezifizierten Qualität sowie zu akzeptablen Kosten. Die Produktion ist somit darauf ausgerichtet, die Kundennachfrage zu erfüllen, während gleichzeitig Effizienz und Rentabilität der Transformationsprozesse zur Nachfrage-Erfüllung optimiert werden (Westkämper 2006, S. 68).

Laut Westkämper (2006, S. 195) umfasst der Begriff *Produktion* alle erforderlichen Tätigkeiten, die dazu beitragen, Rohmaterial, Halbzeuge und Hilfsstoffe mithilfe von Boden, Energie, Informationen und Arbeit innerhalb wertschöpfender und assoziierter Prozesse zu transformieren, um Produkte, Fertigteile oder Abfälle zu erzeugen. Die Produktion ist somit u. a. der Prozess der Herstellung von Produkten, der darauf ausgerichtet ist, Mehrwert zu schaffen und die Nachfrage der Kunden zu erfüllen. Die Produktion umfasst alle Tätigkeiten von der Erfassung der Kundenbedarfe über die Materialbeschaffung und -verarbeitung bis hin zur Qualitätskontrolle und Verpackung der Produkte. Für die eigentliche Transformation, den wertschöpfenden Produktionsprozess, werden u. a. Fertigungsmittel eingesetzt, die Westkämper (2006, S. 161) in *Maschinen*, *Werkzeuge* und *Vorrichtungen* untergliedert. Abbildung 2-1 veranschaulicht das Produktionssystem nach Westkämper (2006, S. 195).

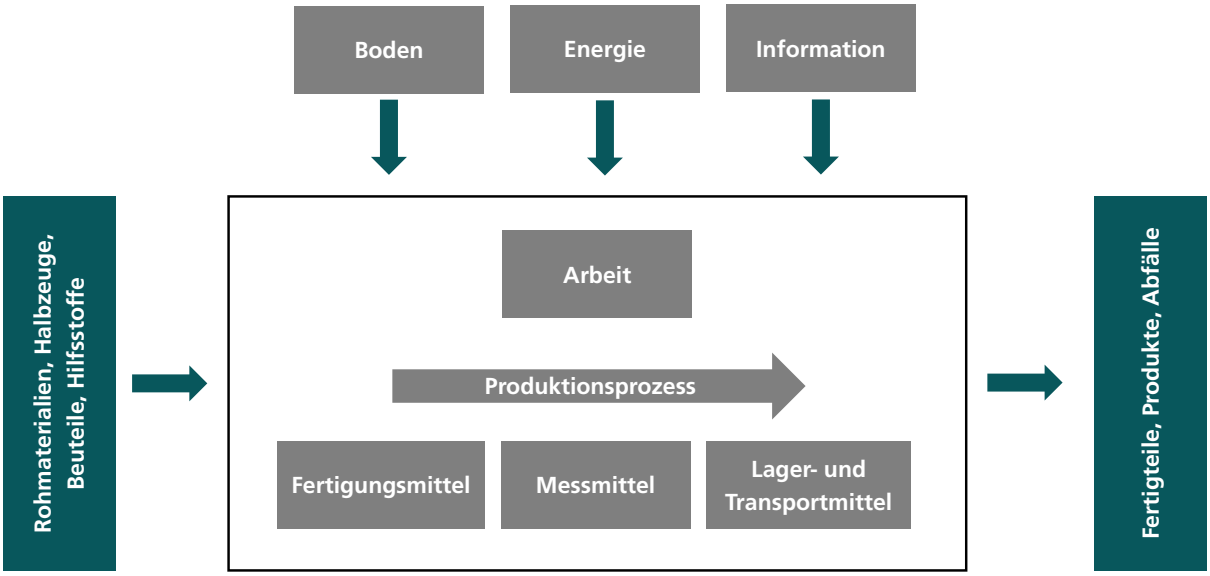


Abbildung 2-1 Produktionssystem nach Westkämper (2006, S.195)

Da in den meisten Branchen, wie der Automobilproduktion, der Energiebedarf der Produktionsprozesse bei 60 % liegt und Fertigungsmittel sowie Lager- und Transportmittel (z. B. Robotik oder Druckluft) signifikant zum Energieverbrauch beitragen, liegt der Fokus dieser Arbeit auf den Fertigungs-, Lager- und Transportmitteln innerhalb der Produktionsprozesse (Geckler et al. 2020, S. 306).

Eine weitere Definition für *Produktionsprozess* liefert Groover (2020, S. 10) als die Änderung der Geometrie, der Eigenschaften oder des Aussehens von Rohmaterialien und Halbzeugen mit dem Ziel der Erhöhung des Fertigstellungsgrades. Überträgt man dies auf das Produktionssystem von Westkämper (2006, S. 200), werden durch den Einbezug und die Nutzung von Fertigungsmitteln, Messmitteln sowie Lager- und Transportmitteln Produktionsprozesse durchgeführt, die in der(DIN-Norm 8580)DIN 8580:2022 als Fertigungsverfahren definiert sind. Im Folgenden soll für den Begriff *Fertigungsverfahren* weiterhin der Begriff *Produktionsprozess* genutzt werden.

Im sog. *Stuttgarter Unternehmensmodell* ist die Produktion eine sowohl vertikale als auch horizontale Anordnung von unterschiedlichen Systemebenen (Westkämper & Hummel 2009). Dabei ist die vorletzte Ebene, definiert als *Arbeitsplätze und Maschinen*, die Voraussetzung für die Durchführung der bereits vorgestellten Produktionsprozesse aus der DIN 8580:2022. Die *Maschinenrichtlinie der Europäischen Union* definiert eine Maschine als eine Zusammenfügung von verbundenen Teilen oder Vorrichtungen für meist einen Produktionsprozess (Europäisches Parlament 2006). Dabei muss mindestens ein Teil oder eine Vorrichtung beweglich sein. Zusätzlich gehören zur Definition der Maschine Betätigungsgeräte sowie Steuer- und Energiekreise, die für die Anwendung erforderlich sind. Relevante Informationen des Zustands der Systemebene *Arbeitsplätze und Maschinen* können durch die Ebene 1 (*Intelligente Geräte*) der produktionstechnischen Automatisierungspyramide nach DIN EN 62264 erfasst werden (DIN EN 62264-1). Eine weitere Definition ist Siepmann (Siepmann & Graef 2016) zu entnehmen, die Ebene 1 als „Feldebene“ bezeichnet. Hier werden durch Sensoren und Aktoren produktionsrelevante Informationen erfasst, verarbeitet und rückgekoppelt. In beiden Modellen sind die zuvor definierten Produktionsprozesse als *Ebene 0* definiert.

2.1.2 Maschinen- und Betriebsdaten

Um produktionsrelevante Informationen über die Maschinen und die zugehörigen Produktionsprozesse zu erhalten, werden Daten durch intelligente Geräte wie Sensoren auf Ebene 1 nach Siepmann (2016, S. 50) erfasst. VDI-Richtlinie 5600 definiert die Maschinen- und Betriebsdatenerfassung eingebettet in einem Fertigungsmanagementsystem (VDI-Richtlinie 5600 Blatt 1). Maschinen- und Betriebsdaten sind Daten, die über Schnittstellenverbindungen erfasst werden, um den Zustand der im Produktionsprozess eingesetzten Fertigungs-, Transport- und Lagermittel zu ermitteln. Zu den Maschinendaten gehören u. a. aktuelle Informationen über Maschinenparameter wie z. B. Temperatur, Geschwindigkeit, Energieverbrauch oder auch Betriebszustand. Daher sind die in dieser Arbeit relevanten Energiedaten der Definition von Maschinendaten zuzuordnen. Zum Erfassen der Parameter der Produktionsprozesse in digitale Signale werden Sensoren, industrielle Bussysteme oder auch Terminals genutzt (Kletti 2015).

Als *Betriebsdaten* werden Daten und Informationen der Arbeitsplätze bezeichnet (Kletti 2015, S. 20). Dazu gehören die Informationen der jeweiligen Mitarbeiter oder der dem Prozess zugehörigen Ressourcenverbräuche wie Material oder Energie. Je nach Bedarf können weitere Daten wie Durchlaufzeiten abhängig vom Produkt oder vom Mitarbeiter aufgezeichnet werden. Außerdem beinhaltet die Betriebsdatenerfassung eine Ausschussstatistik, um Ausschüsse so genau wie möglich zuzuordnen. Dabei können Letztere auch als *Anomalien* bezeichnet werden, die zur Produktionszeit erfasst und mit dem Zeitpunkt ihres Auftretens abgeglichen werden, um mögliche Ursachen näher zu lokalisieren.

Auf der Feldebene können Daten aus der Produktion entweder automatisiert oder manuell erfasst werden (Westkämper 2013, S. 118). Die automatisierte Datenerfassung erfolgt durch Schnittstellen an Fertigungs-, Transport- und Lagermitteln. In diesem Fall werden die Daten direkt an die übergeordneten Systeme oder Plattformen weitergeleitet. Automatisiert erfasste Daten aus Produktionsprozessen der Ebenen 0 oder 1 können bspw. Maschinentakte, Zählimpulse oder Messwerte von Sensoren sein (Siepmann &

Graef 2016, S. 50). Um Daten manuell zu dokumentieren, sind sog. *Erfassungsterminals* notwendig, die eine Erfassung, Speicherung und Verwaltung von Daten ermöglichen. Letztere werden über Barcode-Lesegeräte, Scanner oder auch RFID-Leser aufgezeichnet (Kletti 2015, S. 51). Ein Beispiel für manuell erfasste Daten sind Stempelzeiten der Produktionsmitarbeiter.

Um aus den erfassten Daten sinnvolle Informationen zu gewinnen, ist eine hinreichende Datenqualität erforderlich. Zu den Kriterien hoher Datenqualität gehören ausreichende Vollständigkeit und Plausibilität der Daten. Letztere kann bereits während der Erfassung überprüft werden (Kletti 2015, S. 53). Zusätzlich können fehlende Werte zu einem Informationsverlust führen, der nur bedingt durch Interpolationsverfahren vermieden werden kann. Ein weiterer Punkt für die Informationsgewinnung ist die Verfügbarkeit der Daten. Da keine IT-Infrastruktur eine vollständige Ausfallsicherheit gewährleisten kann, ist es notwendig, dass die erfassten Daten für einen gewissen Zeitraum auch offline verfügbar sind (Kletti 2015, S. 53). Trotzdem ist es erforderlich, die aus den Daten gewonnenen Informationen und Auswertungen nicht nur offline mittels Bedienterminals in den Produktionshallen, sondern auch – über einen Netzwerkanschluss – online zur Verfügung zu stellen. Mobile Endgeräte sind durch die Entwicklung spezifischer Applikationen zur Darstellung der Information und zur Steuerung von Produktionsprozessen durch den Produktionsmitarbeiter nutzbar (Kletti et al. 2015, S. 54).

2.1.3 Energiedatenanalyse in Produktionsprozessen

Nach DIN EN ISO 50001 wird Energie als die Fähigkeit definiert, eine externe Aktivität oder Arbeit zu verrichten (DIN EN ISO 50001, S. 16). Der Energieverbrauch beschreibt die Menge an Energie, die für eine bestimmte Aktivität benötigt wird (DIN EN ISO 50001, S. 16). In der vorliegenden Arbeit sollen alle Daten, die den Energieverbrauch beschreiben, als **Energiedaten** bezeichnet werden.

Ein Teilaspekt des EnMS befasst sich mit der Überwachung, Messung und Analyse von Energiedaten und wird *Energiedatenmanagementsystem* (EDMS) genannt. Das EDMS ist das zentrale Informationswerkzeug im EnMS. Es umfasst alle Schritte von der

Datenerfassung bis zur umfassenden Analyse und Auswertung der Energiedaten. Die Energiedatenanalyse dient als „Übergang von reinen Zahlen zu verwertbaren Ergebnissen“ und umfasst Aufgaben wie die Berechnung des Energiebedarfs, das Erstellen von Zeitreihenanalysen, die Entwicklung von Kennzahlen und den Vergleich von Datenreihen (Geilhausen et al. 2015, S. 295). Geilhausen et al. definieren zusätzlich den Begriff des Monitorings als komplementäre Funktion zur Energiedatenanalyse (2015, S. 295). Das Monitoring bezieht sich auf die regelmäßige Überwachung von Kenngrößen und das Erkennen von Unregelmäßigkeiten im Energieverbrauch. Darüber hinaus definiert die VDI das Energiemonitoring als eine Funktionalität zur Überwachung der Entwicklung des Energieeinsatzes von Produktionsanlagen. Dabei werden vorgegebene Grenzwerte überwacht (VDI-Richtlinie 5600 Blatt 6, S. 12).

2.2 Energetische Anomalieerkennung

Die Erkennung von Anomalien anhand von Energiedaten ist ein zentraler Aspekt dieser Arbeit und soll daher im Folgenden näher beleuchtet werden. Zunächst wird hierfür der Begriff der Anomalie definiert und die unterschiedlichen Ausprägungen von Anomalien werden analysiert. Daraufhin erfolgt die Beschreibung der Methoden zur Erkennung von Anomalien.

2.2.1 Begriffsdefinition

Eine *Anomalie* ist definiert als eine Beobachtung oder ein Datenpunkt eines Systems, das von dessen normalem Verhalten abweicht (2021, S. 1). Dabei wird das Konzept der Normalität eines Systems als Wahrscheinlichkeitsverteilung definiert, die das grundlegende normale Verhalten eines beobachteten Systems oder Objekts erfasst. Mithin stellt eine Anomalie eine Beobachtung dar, deren Wahrscheinlichkeit, innerhalb der Wahrscheinlichkeitsverteilung des Gesetzes des Normalverhaltens zu liegen, sehr gering ist (Ruff et al. 2021, S. 3).

In der Literatur wird zwischen punktuellen, kontextuellen und kollektiven Anomalien (siehe Abbildung 2-2) unterschieden (Gupta et al. 2014, S. 2254). Eine *Punktanomalie* ist definiert als ein eindeutiger Datenpunkt – z. B. eine Messung der Wirkleistung. Die Erkennung von Punktanomalien ist einer der typischen Anwendungsfälle in Industrie und Forschung (Himeur et al. 2021, S. 13). Eine *kontextuelle Anomalie* indes bezieht die gemessenen Daten auf einen bestimmten Kontext, bspw. die Zeit oder den Raum. Zusätzlich können andere Einflussgrößen in Form anderer Systeme miteinbezogen werden, um eine Anomalie zu definieren (Ruff et al. 2021, S. 3). Die momentane elektrische Leistungsaufnahme von 120 kW im Kontext eines elektrischen Heizelementes eines Schmelzofens etwa wird als normales Verhalten angesehen, während die momentane elektrische Leistungsaufnahme von 120 kW im Kontext einer einfachen Bohrmaschine als kontextuelle Anomalie betrachtet werden kann. Insbesondere die Erkennung von Anomalien in Zeitreihendaten ist häufig als kontextuelle Anomalie definiert (z. B., dass der Energieverbrauch an Wochenenden höher als unter der Woche ist). Daher ist das genannte Konzept der Normalität eine bedingte Verteilung, die von einer Dimension wie der Zeit oder dem Raum abhängt (Cook et al. 2020, S. 6485). Die kontextuelle wird durch die *kollektive Anomalie* erweitert (Ruff et al. 2021, S. 3): Mehrere Datenpunkte werden hierbei als Anomalie definiert, während die einzelnen Datenpunkte dem normalen Verhalten zugeordnet werden können.

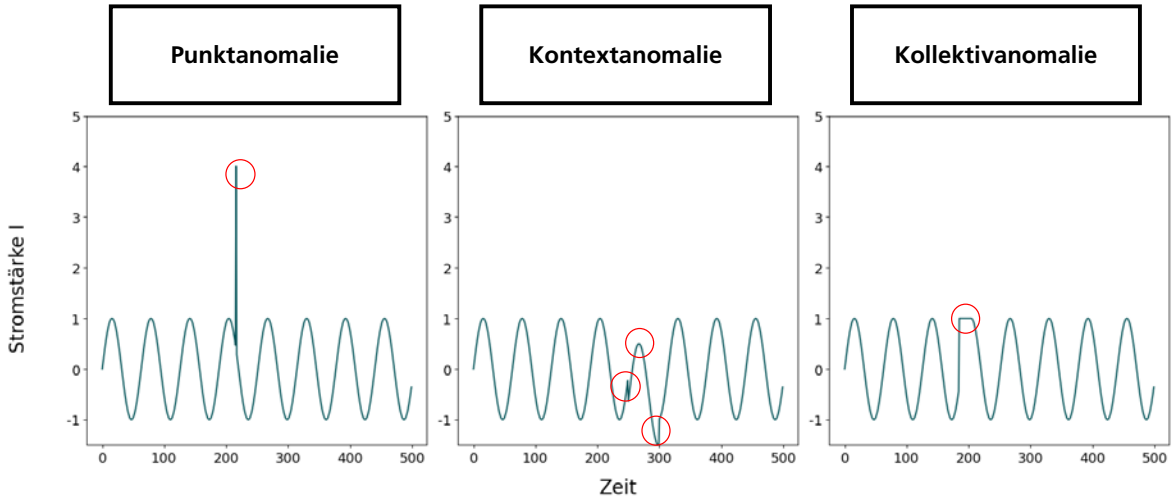


Abbildung 2-2 Kategorisierung von Anomalien nach Ruff et al. (2020)

2.2.2 Szenarien

Das Ziel der Anomalieerkennung ist das erfolgreiche Identifizieren der in Abschnitt 2.1.1 definierten Abweichungen zum Normalverhalten eines beobachteten Systems oder Objekts. Grundlegend können drei mögliche Szenarien zur Anomalieerkennung unterschieden werden: überwacht, unüberwacht und teilüberwacht (siehe Abbildung 2-3). Sie hängen hauptsächlich von den Eigenschaften der zur Verfügung stehenden Daten ab (Ruff et al. 2021).

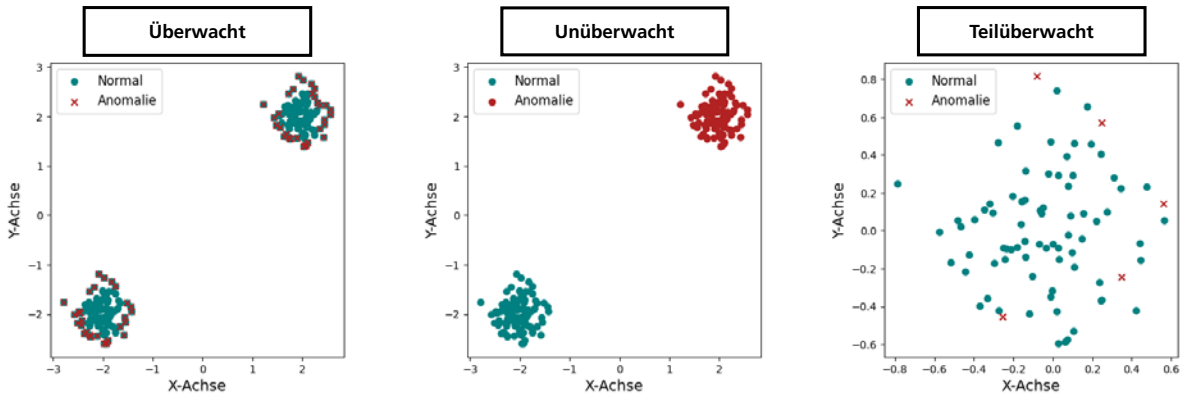


Abbildung 2-3 Kategorisierung der Szenarien für die Anomalieerkennung

Im *überwachten* Szenario gibt es für jede Anomalie annotierte Daten, die als positive Klasse annotiert sind, und Daten mit normalem Verhalten, die als negative Klasse definiert werden. Es handelt sich also um ein aus dem ML bekanntes Klassifikationsproblem mit zwei oder mehr Klassen, abhängig von der Kategorisierung der betrachteten Anomalien (Russell et al., 2021, S. 653). Bewährte statistische ML-Methoden wie die logistische Regressionsanalyse oder Entscheidungsbäume, aber auch Architekturen im Bereich der *künstlichen neuronalen Netze* (KNN) können angewandt werden, um die Daten mithilfe einer Funktion den jeweiligen Klassen zuzuordnen (Russell et al., 2021, S. 751). Jedoch sind in der Praxis für die meisten Anwendungsfälle keine oder wenige annotierte Daten vorhanden, da es in der Regel arbeitsintensiv und mühsam ist, Daten zu annotieren (Himeur et al. 2021, S. 14). Zusätzlich können in diesem Szenario keine Anomalien erkannt werden, die nicht zuvor als Anomalie definiert worden sind. Störungen im Betrieb eines Systems, die noch nicht aufgetreten sind, können daher mit Methoden aus dem Bereich des überwachten Szenarios nicht hinreichend erkannt werden.

Im *unüberwachten* Szenario sind die Anomalien eines Systems im beobachteten Datensatz nicht annotiert (Ruff et al. 2021, S. 6). Hinsichtlich des hohen Aufwands der Annotation sind die meisten Szenarien in der Realität diesem Szenario zuzuordnen (Himeur et al. 2021, S. 14). Bei der Beobachtung eines Systems und der Erfassung von Daten wird im unüberwachten Szenario von einer Messung des Normalverhaltens eines Systems ausgegangen. Allerdings tritt bei den meisten Messungen durch bspw. Sensoren ein Rauschen auf (Cook et al. 2020, S. 6484). Auch wenn die unüberwachten Systeme erfolgreich eingesetzt werden und niederschwellig zu implementieren sind, ist eine wissenschaftliche Validierung anhand korrekt annotierter Daten nicht möglich, da meist die erforderliche Datenbasis fehlt (Himeur et al. 2021, S. 14). Im Beispiel der Anomalien anhand der Energiedaten einer Maschine oder Anlage in Produktionsprozessen wäre ein mögliches Ergebnis, die Beobachtungen, die nicht in die bestehende Wahrscheinlichkeitsverteilung passen, als Anomalie zu betrachten.

In *teilüberwachten* Szenarien stehen sowohl annotierte als auch nicht annotierte Daten eines Systems oder Prozesses für die Erkennung von Anomalien zur Verfügung (Ruff et al. 2021, S. 6). Jedoch liegt die Anzahl der annotierten deutlich unter jener der nicht annotierten Daten, sodass Methoden und Verfahren aus dem überwachten Szenario nicht sinnvoll angewandt werden können (Ruff et al. 2021, S. 6). Trotzdem kann selbst ein kleiner Anteil an annotierten Daten die Erkennungsrate von Anomalien im teilüberwachten Szenario signifikant steigern (Kiran et al. 2018). In der wissenschaftlichen Literatur zu teilüberwachten Szenarien gibt es keine eindeutige Aussage dazu, ob in den vorhandenen Daten nur das Normalverhalten abgebildet wird (Chandola et al. 2009) oder auch bereits entdeckte Anomalien annotiert wurden (Ruff et al. 2021).

2.2.3 Verfahren

Bisherige Forschungen haben keine einheitliche Form der Kategorisierung unterschiedlicher Modelle zur Erkennung von Anomalien erbracht. Auf Grundlage bestehender Literaturanalysen im Bereich der Anomalieerkennung soll deshalb eine generalisierte Kategorisierung durchgeführt werden (Chandola et al. 2009; Gupta et al.

2014; Pang et al. 2021; Ruff et al. 2021). In dieser Arbeit werden die drei Verfahrensgruppen – Distanz, Rekonstruktion und Klassifikation – vorgestellt.

Distanzbasierte Verfahren transformieren eine gemessene Beobachtung in einen n-dimensionalen Raum, in dem auf Basis von sowohl Ähnlichkeits- als auch Distanzfunktionen Anomalien erkannt werden können. Es existieren unterschiedliche Distanzmaße, bspw. die euklidische oder die Mahalanobis-Distanz (Maesschalck et al. 2000). Bei distanzbasierten Verfahren wird zunächst ein Algorithmus verwendet, um Datenpunkte oder Bereiche einer Zeitreihe in einem multidimensionalen Raum in Form von Vektoren darzustellen und Bereiche mit einer hohen Datendichte zu identifizieren. Der einfachste Algorithmus wäre in diesem Fall die Evaluation der Distanz einer Beobachtung gegenüber allen anderen Punkten im multidimensionalen Datenraum (Russell et al. 2021, S. 738). Im Bereich der distanzbasierten Verfahren gibt es eine Vielzahl von Algorithmen, z. B. den *K-means*-Algorithmus und Distanzmaße, die für die Evaluation von Beobachtungen in einem Datenraum zur Erkennung von Anomalien in Betracht gezogen werden können (Hartigan & Wong 1979). Mithilfe der Distanzmaße können auch Gruppen von Beobachtungen gebildet werden, die innerhalb der Gruppe eine besonders niedrige, aber zwischen mehreren Gruppen eine hohe Varianz aufweisen. Im zweiten Schritt wird das gleiche Distanzmaß für die Zuordnung neuer Beobachtungen in unterschiedliche Gruppen verwendet, um ein Maß der Anomalie für die Beobachtung zu berechnen (Aggarwal 2017). Im Falle des K-means-Algorithmus wird bspw. die euklidische Distanz einer Beobachtung zur nächstgelegenen Gruppe verwendet (Hartigan & Wong 1979, S. 100).

Rekonstruktionsbasierte Verfahren zielen auf die Approximation eines meist statistischen Modells zur Rekonstruktion von Beobachtungen (Chandola et al. 2009, S. 19). Dabei gilt die Grundannahme, dass Beobachtungen, die durch das implizit approximierte Modell nicht ausreichend rekonstruiert sind, ab einem zu definierenden Schwellenwert als Anomalie annotiert werden können. Auch bei den rekonstruktionsbasierten Verfahren existieren Lösungen wie die Hauptkomponentenanalyse, aber auch komplexere Verfahren wie die Autoencoder-Architektur für KNN (Ruff et al. 2021, S. 14). Autoencoder sind

bspw. KNN mit dem Ziel der Repräsentation eines Datensatzes durch die Reduktion der Dimensionalität (Encoder) und der darauffolgenden Rekonstruktion der Daten anhand der generierten reduzierten Dimensionen (Decoder). Durch das Erlernen latenter Variablen in einer niedrigen Dimension soll eine Repräsentation der Daten erzeugt werden, die den gemessenen Daten möglichst ähnlich ist (Ruff et al. 2021, S. 14).

Klassifikationsbasierte Verfahren ermöglichen die Klassifikation in zwei oder mehr Klassen mithilfe eines ML-Modells. Die grundlegende Prämisse klassifikationsbasierter Verfahren basiert auf der Fähigkeit eines ML-Modells, Daten in zwei unterschiedliche Kategorien zu klassifizieren: *normales* Verhalten, repräsentiert durch die *Normalklasse*, und abweichendes oder *anomales* Verhalten, repräsentiert durch die *Anomalieklasse* (Chandola et al. 2009, S. 18). Dabei ist die häufig angewandte Form die binäre Klassifikation, die Zeitreihen als „keine Anomalie/Normalzustand" oder „Anomalie" klassifiziert (Himeur et al. 2021, S. 5). Es gibt jedoch auch Anwendungen, bei denen unterschiedliche Anomalien erkannt werden sollen, die als multinomiale Klassifikationsprobleme definiert werden (Russell et al. 2021, S. 696). Ähnlich den rekonstruktionsbasierten Verfahren ist das grundlegende Ziel hierbei die Approximation von Parametern einer Funktion durch die Optimierung einer Verlustfunktion anhand bestehender Daten (Chandola et al. 2009, S. 18). Die Approximation der Parameter wird oft auch als „induktives Lernen oder als Prozess der Identifikation einer Hypothese [bezeichnet], die mit den beobachteten Beispielen übereinstimmt" (Russell et al. 2021, S. 653). Dabei wird in der Literatur auch häufig vom *Trainieren eines ML-Modells durch bereits mit Anomalien annotierte Daten* geschrieben (Ruff et al. 2021, S. 8). Auch in der vorliegenden Arbeit soll für die Approximation der Parameter anhand historischer Daten der Begriff *Training* genutzt werden. Nach dem Training erfolgt das Testen des ML-Modells. Hierzu werden Daten genutzt, die das trainierte ML-Modell noch nicht klassifiziert hat (Russell et al. 2021, S. 666).

2.2.4 Verfahrensauswahl

Damit es möglich ist, unterschiedliche Verfahren für die energetische Anomalieerkennung miteinander zu vergleichen, können statistische Evaluationsmetriken genutzt werden, die die Leistungsfähigkeit der Verfahren bewerten. Um zufällige Ergebnisse der Evaluationsmetriken zu vermeiden, werden Kreuzvalidierungen durchgeführt (Runkler 2015, S. 78). Die gängigsten Metriken sind *Genauigkeit* bzw. *Accuracy*, *Recall*, *Präzision* und der *F1-Score*, die in diesem Abschnitt nach (Tharwat 2021) vorgestellt werden. Unter der Annahme eines binären Problems zur Anomalieerkennung können vier unterschiedliche Ergebnisse bei der Erkennung auftreten (Runkler 2015, S. 90):

Richtig Positiv (TP) – Anomalie wird als Anomalie erkannt

Richtig Negativ (TN) – keine Anomalie wird als keine Anomalie erkannt

Falsch Negativ (FN) – Anomalie wird als keine Anomalie erkannt (Fehler erster Art)

Falsch Positiv (FP) – keine Anomalie wird als Anomalie erkannt (Fehler zweiter Art)

Falls ein Zeitfenster in einem Produktionsprozess korrekt entweder als positiv (Anomalie) oder negativ (keine Anomalie) erkannt wird, kann man dieses als *richtig* bezeichnen, falls nicht als *falsch*. So sind *richtig positiv* und *richtig negativ* durch das Modell erfolgreich erkannte Anomalien oder keine Anomalien, die entweder eine Anomalie korrekt als 1 (TP) oder keine Anomalie als 0 (TN) erkennt. Wird das betrachtete Zeitfenster falsch vorhergesagt, entspricht dies FN, falls das Zeitfenster fälschlicherweise als *positiv* klassifiziert, oder FP, wenn das Zeitfenster eigentlich *positiv* ist, jedoch als *negativ* klassifiziert wurde.

Anhand der vier unterschiedlichen Ergebnisse können im nächsten Schritt Evaluationsmetriken berechnet werden. Die *Genauigkeit* ist ein Maß für die Gesamtleistung der Anomalieerkennung (Tharwat 2021, S. 4). Sie wird definiert als das Verhältnis der Anzahl der korrekten Erkennung von Zeitfenstern (sowohl wahre Positiv- als auch wahre Negativfälle) zur Gesamtzahl der betrachteten Zeitfenster. Mathematisch wird die Genauigkeit ausgedrückt als:

$$Genauigkeit = \frac{TP + TN}{TP + TN + FP + FN}$$

Formel 2-1 Berechnung der Genauigkeit

Die *Sensitivität* misst das Verhältnis der richtig erkannten positiven Zeitfenster (TP) zur Gesamtzahl der tatsächlichen positiven Zeitfenster (TP + FN). Diese Metrik ist für Anwendungen relevant, bei denen Anomalien schwerwiegende Folgen haben können und daher eine sichere Erkennung aller Anomalien erforderlich ist (Tharwat 2021, S. 2). Die Sensitivität wird wie folgt berechnet:

$$Sensitivität = \frac{TP}{TP + FN}$$

Formel 2-2: Berechnung der Sensitivität

Die *Präzision* gibt an, wie viele der als positiv erkannten Zeitfenster tatsächlich positiv sind. Sie ist definiert als das Verhältnis von TP zur Gesamtzahl der als positiv erkannten Fälle (TP + FP). Die Präzision ist besonders relevant in Anwendungsfällen, in denen die Maßnahmen nach Erkennung einer Anomalie sehr aufwendig sind (Tharwat 2021, S. 2). Die Berechnung der Präzision erfolgt durch:

$$Präzision = \frac{TP}{TP + FP}$$

Formel 2-3 Berechnung der Präzision

Da in den meisten Anwendungsfällen sowohl die Fähigkeit des Verfahrens, Anomalien korrekt zu erkennen (Sensitivität), als auch die Zuverlässigkeit der Erkennung (Präzision) eine übergeordnete Rolle spielen, wird in der Praxis häufig der *F1-Score* als Evaluationsmetrik eingesetzt (Tharwat 2021, S. 4). Dieser ist ein harmonischer Mittelwert aus Präzision und Sensitivität. Er ermöglicht eine Bewertung der Gesamtleistung des verwendeten Verfahrens:

$$F1\text{-}Score = 2 \times \frac{Präzision \cdot Sensitivität}{Präzision + Sensitivität}$$

Formel 2-4 Berechnung des F1-Scores

Nachdem die Evaluationsmetriken für klassische ML-Modelle hergeleitet wurden, sollen nun die unterschiedlichen Verfahren zur Evaluation der Hyperparameter beschrieben und erläutert werden. Die Hyperparameteroptimierung ist ein wichtiger Schritt bei der

Evaluierung und Parameterkonfiguration von ML-Modellen und anderen Optimierungsverfahren. Sie umfasst verschiedene Techniken zur systematischen Anpassung statischer Modellparameter, z. B. Epochen oder die Anzahl der Schichten in einem KNN, um zuvor definierte Evaluationsmetriken zu maximieren. In diesem Zusammenhang werden häufig Methoden wie randomisierte Suche, Gittersuche und bayessche Hyperparameteroptimierung verwendet.

Die *Gittersuche* ist ein umfassender Ansatz, bei dem jede mögliche Kombination von Hyperparametern innerhalb eines vordefinierten Gitters getestet wird. Dieses Verfahren ist gründlich, aber oft zeitaufwändig und rechenintensiv, insbesondere bei großen Suchräumen (Grimme & Bossek 2018, S. 162). Bei der *randomisierten Suche* werden zufällig ausgewählte Parameterkombinationen ausprobiert, um den Suchraum effizient zu durchsuchen (Grimme & Bossek 2018, S. 192). Dieser Ansatz kann schneller sein als die Gittersuche, da nicht alle Kombinationen durchprobiert werden müssen, aber er kann das Optimum verfehlen, da die Suche unsystematisch ist. Außerdem benötigt die randomisierte Suche weitaus mehr Versuche, um ein Optimum zu finden, wenn die Trefferwahrscheinlichkeit bei 90 % liegen soll (Grimme & Bossek 2018, S. 194). Der *Parzen-Tree Estimator* (PTE) ist ein *bayessches Optimierungsverfahren* und verwendet ein probabilistisches Modell, um den Suchraum für das Optimum effizient zu durchsuchen (Ozaki et al. 2022). Im Gegensatz zu anderen Methoden, die entweder zufällig oder vollständig systematisch suchen, nutzt die bayessche Optimierung Informationen aus früheren Iterationen, um die nächsten Parameter vorherzusagen, die am wahrscheinlichsten zu einer Verbesserung führen (Hutter et al. 2019, S. 9). Dieser Ansatz betrachtet die Hyperparameteroptimierung als ein sequenzielles Entscheidungsproblem, bei dem jede Auswahl auf den vorherigen Ergebnissen basiert. Die Effizienz der bayesschen Hyperparameteroptimierung zeigt sich in deren Fähigkeit, den Suchraum gezielt zu untersuchen, ohne den gesamten Raum systematisch abdecken zu müssen. Für praktische Anwendungen – insbesondere, wenn Rechenressourcen und Zeit begrenzt sind – bietet die bayessche Optimierung somit erhebliche Vorteile gegenüber anderen Verfahren (Yang & Shami 2020, S. 37). Dies ist besonders bei rechen- und zeitintensiven Optimierungsiterationen vorteilhaft. Yang et al. zeigen, dass für nicht komplexe Modelle

und Suchräume oft fünfzig Optimierungsiterationen ausreichen, um eine adäquate Parameterauswahl zu identifizieren (Yang & Shami 2020, S. 47).

2.3 Föderiertes Lernen

In diesem Kapitel sollen die grundlegenden Konzepte und Methoden aus dem Bereich FL vorgestellt werden. Im ersten Schritt wird die Notwendigkeit eines Paradigmenwechsels von zentraler hin zu lokaler Datenverarbeitung auf Endgeräten wie Maschinen oder Anlagen hergeleitet.

2.3.1 Begriffsdefinition

Das IEEE definiert *Federated Machine Learning* als ein System, das es mehreren Teilnehmern dieses Systems ermöglicht, gemeinsam mehrere anwenderspezifische ML-Modelle mit einer vorgegebenen Modellarchitektur zu initialisieren und zu trainieren, ohne sensible Daten der Teilnehmer preiszugeben und gleichzeitig eine hinreichend gute, vorab definierte Evaluationsmetrik zu erhalten (IEEE 3652, S. 14). Im Folgenden soll für den Begriff Federated Machine Learning weiterhin der Begriff FL genutzt werden. Somit ist FL eine Methode zum verteilten Trainieren von ML-Modellen, wobei die Verarbeitung der Daten nicht auf zentraler Instanz, sondern dezentral auf den oder in der Nähe der Datenquellen ausgeführt werden. In der Literatur werden Letztere oder die lokalen Endgeräte als *Client* bezeichnet (McMahan et al. 2016, S. 1).

Der Prozess von FL ist schematisch in Abbildung 2-4 dargestellt und angelehnt an das Vorgehen von McMahan et al. (2016, S. 5). Hierfür wird von einem Server ein globales Modell mit fest definierter Modellarchitektur als Ausgangsmodell initialisiert. Der Server überträgt das initiale globale Modell an die Clients, um mit dem Training zu beginnen. In der Regel wird eine bestimmte Anzahl von Clients ausgewählt, um das Training durchzuführen. Diese Auswahl kann zufällig oder nach bestimmten Kriterien erfolgen, z. B. aufgrund verfügbarer Datenmengen oder Rechenleistung. Das globale Modell wird

in Form von Gewichten oder Parametersätzen übertragen, die von den Clients für Training oder für die Inferenz verwendet werden.

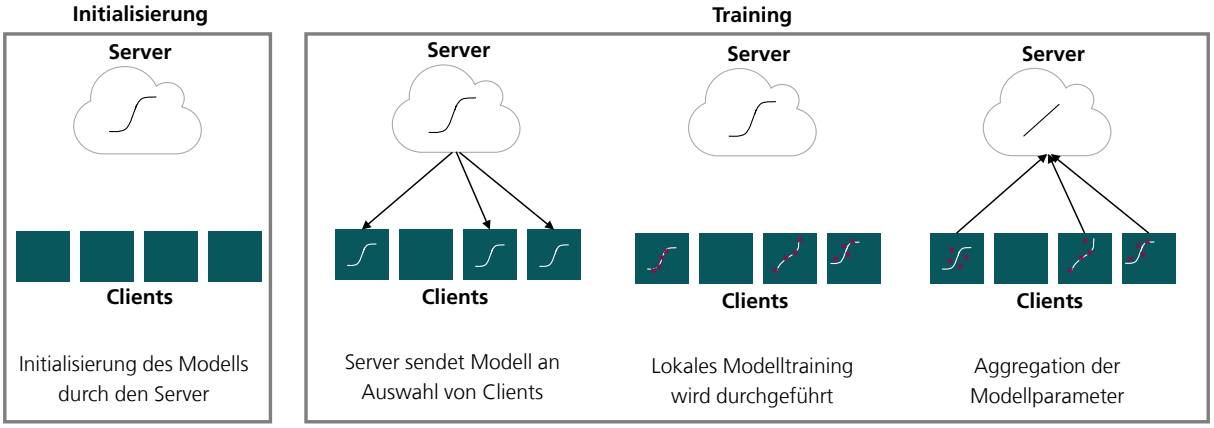


Abbildung 2-4 Prozess des föderierten Lernens, angelehnt an McMahan et al. (2016, S.5)

Nachdem das initiale globale ML-Modell von den Clients empfangen wurde, beginnt das Training der lokalen ML-Modelle. Die lokalen Daten werden vorbereitet, um sie für das Training zu nutzen. Dieser Schritt kann die Datenaufbereitung, Normalisierung, Anonymisierung oder andere Vorverarbeitungsschritte umfassen, um die lokalen Daten für das Modell vorzubereiten. Das lokale ML-Modell wird auf den vorbereiteten lokalen Daten trainiert. Dieser Schritt erfolgt bspw. bei KNN meist durch die Optimierung einer Verlustfunktion mithilfe des stochastischen Gradientenabstiegsverfahrens. Die Gewichte bzw. Parametersätze des ursprünglichen globalen Modells werden in diesem Fall als Startwerte verwendet und in Abhängigkeit vom Optimierungsansatz weiter approximiert.

Nachdem alle lokalen Modelle von den Clients übertragen wurden, beginnt der Server mit der Aggregation der lokalen Modelle. Zu diesem Zweck erhält er die Gewichte oder Parametersätze der lokalen Modelle von allen Clients, die an der Trainingsrunde teilnehmen. Der Kern von FL ist die Aggregation der unterschiedlichen Modellparameter der unterschiedlichen Clients hin zu einem aktualisierten globalen Modell, das in der nächsten Trainingsrunde aktualisiert werden kann. Die lokalen Modelle werden durch die Berechnung des (gewichteten) Mittelwerts oder andere Methoden aggregiert (McMahan et al. 2016). In dieser Arbeit wird das Verfahren zur Aggregation als Aggregationsmechanismus definiert. Der am häufigsten genutzte Aggregationsmechanismus ist FedAvg, der den Mittelwert in Abhängigkeit Anzahl der

lokal genutzten Trainingsdaten berechnet (McMahan et al. 2016). Fault-Tolerant FedAvg erweitert diese Methode, indem es Mindestabschlussraten einführt, um sicherzustellen, dass ausreichend viele Clients erfolgreich abgeschlossen haben, bevor eine Aggregation erfolgt . Quantile Federated Averaging (QFedAvg) führt eine faire Ressourcenzuweisung ein, indem es die Modell-Updates der Clients basierend auf deren Leistung gewichtet und skalierte Updates aggregiert (Li et al. 2019). Des Weiteren existieren adaptive Optimierungsmethoden wie Adam, die im klassischen Training für ML-Modelle eingesetzt werden (Kingma & Ba 2014). Die adaptiven Optimierungsmethoden werden in den Rahmen von FL übertragen und als FedOpt definiert (Reddi et al. 2020). Der Aggregationsmechanismus FedOpt nutzt unterschiedliche Regularisierungsterme, die eine zu hohe Abweichung zwischen dem lokalen und dem globalen Modell minimieren. Eine spezifische Implementierung von FedOpt ist FedAdam, welche den Adam-Optimierer verwendet und dabei die ersten und zweiten Momente der Gradienten berücksichtigt, um eine stabilere Konvergenz zu gewährleisten (Reddi et al. 2020). Nach der Anwendung des Aggregationsmechanimus der individuellen Modellparameter sendet die zentrale Instanz das neu parametrisierte globale Modell an alle Clients, um eine neue Runde des lokalen Trainings zu starten. Dieses Trainingsverfahren wird so lange wiederholt, bis ein bestimmtes Kriterium – meist die Konvergenz der Verlustfunktion – als erfüllt gilt.

2.3.2 Föderierte Lernsysteme

Systeme, die für eine bestimmte Anwendung Methoden von FL nutzen, sollen nach Bonawitz et al. (2019) *föderierte Lernsysteme* (FL-Systeme) genannt werden. FL-Systeme sind in der Regel auf Client-Server-Architekturen aufgebaut, bei denen mehrere Clients über einen Server miteinander kommunizieren (siehe Abbildung 2-5) (Bonawitz et al. 2019, S. 383). Der grundlegende Aufbau der Architektur eines FL-Systems soll anhand der grundlegenden Veröffentlichungen erläutert werden (Bonawitz et al. 2019, S. 376; Yang et al. 2019, S. 8):

Der Server fungiert als zentraler Knotenpunkt und koordiniert die Aktivitäten der Clients. Es empfängt die lokalen Modelle von den Clients, aggregiert diese und stellt das globale

Modell bereit. Der Server kann auch Informationen an die Clients senden, um diese über das globale Modell und die Ergebnisse von FL zu informieren. Die Clients sind typischerweise über eine Netzwerkverbindung mit dem Server verbundene Endgeräte, die ihre lokalen Gewichte oder Parametersätze nach erfolgreichem Training an den Server senden. Die Kommunikation der Clients erfolgt in der Regel nicht bilateral, sondern immer zentral über den Server. Dadurch werden die lokalen Daten der Clients innerhalb des Unternehmens oder der Organisation des jeweiligen Clients erfasst, gespeichert und verwaltet. Dies ist insbesondere wichtig, wenn es um vertrauliche Daten aus dem Produktionsprozess geht, die nicht an Dritte weitergegeben werden dürfen.

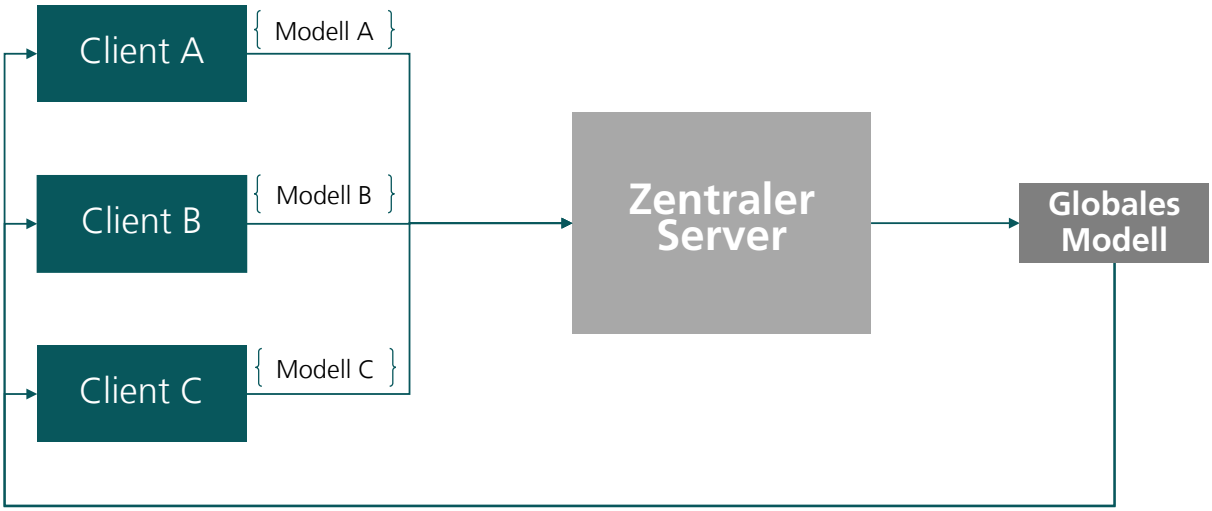
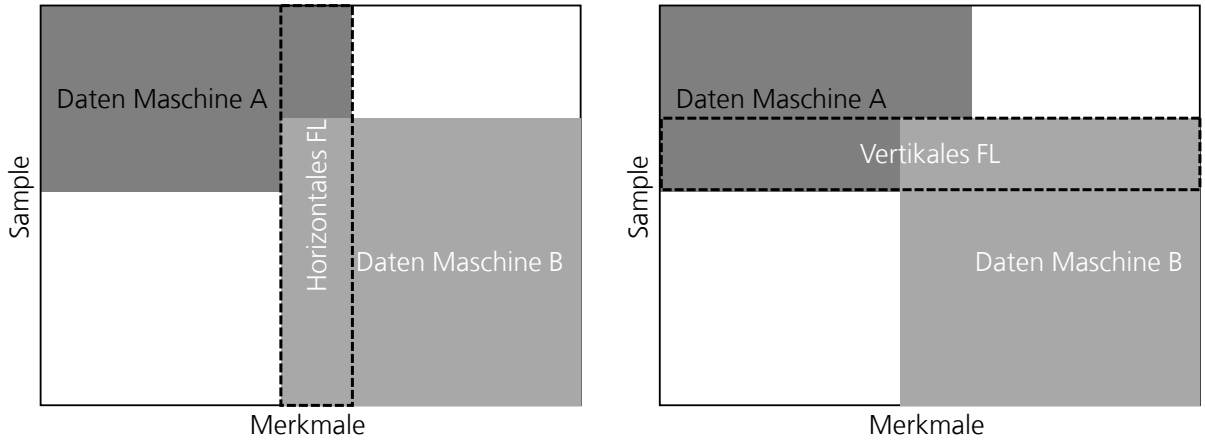


Abbildung 2-5 Grundlegende Systemarchitektur von FL nach McMahan et al. (2016)

FL-Systeme werden in drei Kategorien eingeteilt: horizontales Lernen, vertikales Lernen und föderiertes Transferlernen (Aledhari et al. 2020, S. 140704). Beim *horizontalen* FL-Ansatz (siehe Abbildung 2-6a) handelt es sich um ein System, bei dem dieselben Daten hinsichtlich der Merkmale einer Datenquelle verarbeitet werden, aber die Stichprobe gegenüber anderen Datenquellen unterschiedlich ist (McMahan et al. 2016). Ein typisches Beispiel für horizontale FL in der Produktion ist eine Maschine, die dieselben Energiedaten (z. B. Spannung oder Strom) erfasst, aber in verschiedenen Unternehmen oder für unterschiedliche Anwendungen eingesetzt wird. Die erste horizontale FL-Lösung wurde von McMahan et al. (2016) veröffentlicht, bei der ein einzelner Benutzer die Modellparameter zur Vorhersage des nächsten Wortes lokal auf einem lokalen Endgerät aktualisiert. Die aktualisierten Modellparameter werden an den zentralen Server

übermittelt, in dem ein zentralisiertes Modell mit den Modellparametern mehrerer Datenquellen trainiert wird. *Vertikale* oder merkmalsbasierte FL-Systeme (siehe Abbildung 2-6b) sind Anwendungen, die unterschiedliche Merkmale oder Features der Datenquellen verarbeiten, jedoch denselben Stichprobenraum besitzen (Yang et al. 2019, S. 7). Unter Berücksichtigung von Energiedaten in Produktionsprozessen (siehe Abschnitt 2.1.3) und deren entsprechenden Merkmalsräumen kann vertikale FL für Produktionssysteme verwendet werden, die den gleichen Stichprobenraum (z. B. Zeitstempel), aber verschiedene Maschinen haben, die unterschiedliche Daten sammeln, um Anomalien des gesamten Produktionsprozesses zu erkennen. Einer der nicht weit verbreiteten Ansätze ist die Kombination von FL und Transfer-Lernen, die als *Föderiertes Transfer-Lernen* (FTL) bezeichnet wird (Yang et al. 2019, S. 7). Ziel von FTL ist es, das implizite Wissen im ML-Modell aus einem bestehenden Bereich auf einen neuen zu übertragen. Daher haben FTL-Anwendungen weder dieselben Merkmale noch die gleiche Stichprobe. Eine der ersten Anwendungen von FTL ist *FedHealth* (Chen et al. 2020), das die Erkennung von Aktivitäten ermöglicht, die von tragbaren Geräten gemessen werden. In (Ju et al. 2020) wird ein FTL-Framework für die Klassifizierung verschiedener Quellen von *Elektroenzephalografie*(EEG)-Daten aus Gehirn-Computer-Schnittstellen vorgeschlagen.



a) Horizontales föderiertes Lernen b) Vertikales föderiertes Lernen

Abbildung 2-6 Darstellung der Unterschiede zwischen horizontalem und vertikalem Lernen

2.3.3 Referenzarchitekturen

Für den Einsatz des beschriebenen FL-Ansatzes in Software-Anwendungen wurden Architekturentwurfsmuster entwickelt, die beim Aufbau eines produktiven FL-Systems zu berücksichtigen sind (Lo et al. 2022). Die Muster werden in vier verschiedene Kategorien unterteilt: Endgerätmanagement, Modellverwaltung, Modelltraining und -aggregation. Die Kategorie *Endgerätmanagement* besteht aus Mustern, die die Prozesse und Komponenten für das Datenmanagement, die Endgerätauswahl und das Gruppieren von Endgeräten in Benutzergruppen (z. B. Verwendung einer Werkzeugmaschine in verschiedenen Branchen) definieren. Die *Modellverwaltung* leitet sich aus der Modellregistrierung ab. Hauptbestandteile der Modellregistrierung sind die Speicherung und Verwaltung der Modellparameter sowie der verschiedenen Versionen des Modells (Bonawitz et al. 2019, S. 381). Darüber hinaus sind ein Auslöser für die Ersetzung von Modellen und ein Einsatzsektor erforderlich, um Modelle zu ändern, wenn sich die Leistung verschlechtert. Zusätzlich wird der Modellkompressor als wichtiger Schritt eingeführt, bevor die Modellparameter an die Clients gesendet werden. Die Kategorien *Modelltraining* und *-aggregation* bestehen aus einer Modelltrainerkomponente, einer Registry sowie verschiedenen Aggregationskomponenten, z. B. sicheren, asynchronen und dezentralen Mustern.

Abbildung 2-7 zeigt die Referenzarchitektur der Richtlinie IEEE für FL-Systeme (IEEE 3652). Diese Referenzarchitektur stellt die grundlegenden Funktionalitäten von FL-Systemen in Modulen dar, die in unterschiedlichen Schichten kategorisiert sind. Zusätzlich können Module in verschiedenen Schichten miteinander über Querschnittsfunktionen interagieren. Hervorzuheben ist die Modularität der Referenzarchitektur, die eine nicht vollständige Abdeckung aller Aktivitäten und Funktionalitäten erfordert (IEEE 3652, S. 15). Im Folgenden sollen die unterschiedlichen Schichten kurz vorgestellt werden.

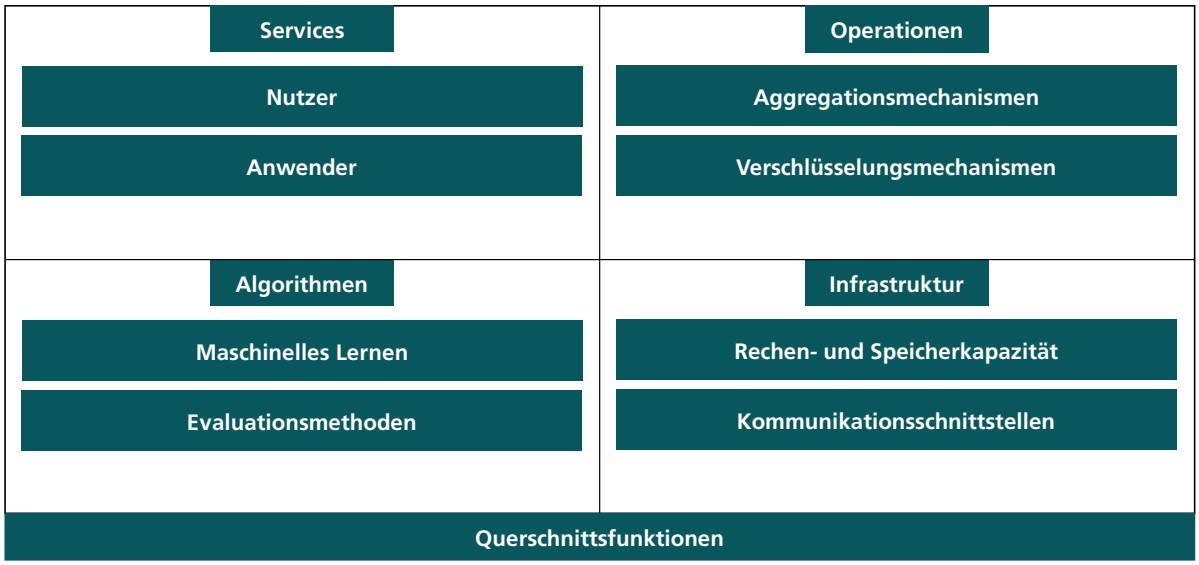


Abbildung 2-7 Vereinfachte Darstellung der Referenzarchitektur aus der IEEE-Richtlinie 3652.1-2020

Die *Serviceschicht* implementiert die Geschäftslogik auf Grundlage der Serviceanforderungen und bietet den Nutzern des FL-Systems unterschiedliche Services. Das Nutzerservicemodul ist bspw. für die Bereitstellung einer Schnittstelle für die Bedienung des Systems verantwortlich. Dies kann eine Schnittstelle mit grafischer Oberfläche, aber auch eine programmierbare Schnittstelle sein. Auch ein Modul zur Koordination der Clients für das Clientmanagement, das Sicherheitsmanagement oder das Monitoring werden in der Serviceschicht abgebildet. Die *Algorithmenschicht* beinhaltet alle Module und Funktionalitäten, die das klassische Vorgehen bei der Anwendung von ML-Methoden unterstützt. Dazu gehören Funktionalitäten zur Merkmalsextraktion sowie unterschiedliche Algorithmen und Modellarchitekturen, aber auch die Implementierung von Evaluationsmetriken. Letztere können jedoch hinsichtlich FL erweitert werden. Die *Operationsschicht* stellt elementare Operationen für FL-Systeme bereit, die unabhängig von den ausgewählten Algorithmen genutzt werden können. Eines der grundlegenden Module in der Operationsschicht ist das Aggregatormodul – eine Sammlung der implementierten und nutzbaren Methoden für die Aggregation der lokalen Gewichte oder Parametersätze. Ein weiteres Modul ist das Rechenmodul, das Verschlüsselungsmechanismen für die lokalen Gewichte und Parametersätze bereitstellt. Die *Infrastrukturschicht* kann als Fundament des FL-Systems betrachtet werden; es soll die Möglichkeiten der Berechnung, Speicherung und Kommunikation von langen Bytes an

Dateneinheiten (z. B. 1024 Bit) bieten. Es handelt sich um die Grundmodule für eine ausreichende Infrastruktur. Module und Komponenten wie bspw. ein Modulkatalog oder Hilfsfunktionen sind schichtenübergreifend in den Querschnittsfunktionen eingebettet.

2.4 Zwischenfazit

Für die Gestaltung einer Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen wurden in diesem Kapitel die Grundlagen hinsichtlich des Bezugsraums der Produktion, Anomalieerkennung und des FL erarbeitet.

Zur Definition des Gegenstandbereichs wurden zunächst der Begriff *Produktion* sowie die relevanten Eingangs- und Ausgangsgrößen eingeführt. Für die Durchführung der Produktionsprozesse werden Fertigungs-, Mess-, Transport- und Lagermittel benötigt, die als Maschinen, Vorrichtungen oder Werkzeuge definiert sind. Mithilfe von Sensoren und Erfassungsterminals können sog. *Maschinen- und Betriebsdaten* erfasst, weiterverarbeitet und analysiert werden. Zu den Maschinendaten gehören entsprechend auch Energiedaten der Produktionsprozesse, die mithilfe eines Energiedatenmanagements erfasst, verwaltet und analysiert werden. Die Energiedatenanalyse bildet die Basis für die Erfassung, Verwaltung, Analyse und Verwertung von Energiedaten innerhalb von Unternehmen. Im zweiten Abschnitt wurden die Grundlagen der Erkennung von Anomalien hergeleitet. Zunächst wurde der Begriff *Anomalie* definiert. Sodann wurden Grundlagen der methodischen Herangehensweise zur Erkennung von Anomalien hergeleitet und ML-Methoden zur Erkennung von Anomalien vorgestellt. Im letzten Unterabschnitt wird das Konzept von FL vorgestellt. Es wurden der prinzipielle Ablauf von FL, die Kategorisierung und der Aufbau von FL-Systemen sowie die Möglichkeit des Schutzes sensibler Daten durch FL näher beleuchtet.

Die in diesem Kapitel erarbeiteten Grundlagen adressieren zugleich die erste Teilforschungsfrage nach den Bestandteilen eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen.

3 Stand der Technik und Wissenschaft

Die Entwicklung von Algorithmen und neuen Ansätzen zur energetischen Anomalieerkennung aus Energiedaten in Produktionsprozessen hat sowohl in der wissenschaftlichen Literatur als auch in der Praxis hohe Relevanz (Himeur et al. 2021). Auch die Weiterentwicklung und Nutzung von FL spielen seit dem Jahr 2020 eine immer wichtigere Rolle, um sensible und unternehmenskritische Daten aus Produktionsprozessen im Unternehmen zu sichern sowie vor externen Angriffen zu schützen.

Im Folgenden sollen der Stand der Technik im Bereich von FL für die Anomalieerkennung in Produktionsprozessen und die energetische Anomalieerkennung in Produktionsprozessen untersucht werden. Zunächst wird hierfür das methodische Vorgehen vorgestellt und die Ansätze werden kurz beschrieben sowie erläutert. Die Beschreibung und Erläuterung der wissenschaftlichen Arbeiten dienen in Kapitel 4 als Grundlage zur Bewertung der Ansätze.

3.1 Vorgehen der systematischen Literaturanalyse

Für die Ableitung der Anforderungen und die Identifizierung der Forschungslücke ist ein Überblick zum Stand der Wissenschaft im Kontext FL-basierter Systeme zur energetischen Anomalieerkennung in Produktionsprozessen notwendig. Hierzu wird in dieser Arbeit die Methode der *systematischen Literaturanalyse* angewandt (vom Brocke et al. 2009). Vom Brocke et al. (2009) gliedern die systematische Literaturanalyse in fünf aufeinanderfolgende Schritte. Im ersten wird das Thema eingegrenzt und konzeptualisiert. Dadurch können im zweiten Schritt relevante Suchwörter identifiziert und für eine Recherche in Datenbanken für wissenschaftliche Literatur genutzt werden. Im nächsten Schritt erfolgt die Vorwärts- und Rückwärtssuche nicht identifizierter Veröffentlichungen innerhalb der schon untersuchten Veröffentlichungen. Diese werden

daraufhin dem Katalog hinzugefügt, um die Literaturbasis der ersten Recherche zu erweitern. Im letzten Schritt kann der Forschungsbedarf identifiziert werden.

Die Auswahl der vier wissenschaftlichen Datenbanken – ScienceDirect, ACM Digital Library, IEEE Xplore und Springer – erfolgt aufgrund ihrer umfangreichen Sammlung von wissenschaftlichen Veröffentlichungen aus Fachzeitschriften und Konferenzen. Die ACM Digital Library beinhaltet eine umfangreiche Sammlung von Publikationen, die sich mit Fragestellungen der Informatik sowie der Künstlichen Intelligenz befassen. Für die Auswahl der wissenschaftlichen Literatur werden drei Kriterien festgelegt: Der Artikel ist (1) in einer von Experten begutachteten akademischen Forschungszeitschrift sowie (2) zwischen 2010 und 2023 veröffentlicht und (3) im Volltext in mindestens einer der vier Datenbanken verfügbar. Nach einer ersten Sichtung der Datenbanken werden Duplikate entfernt und Titel sowie Kurzzusammenfassungen analysiert. Zusätzlich wird ein Wissensgraph genutzt, der die Vernetzung der Veröffentlichungen anhand der Autoren darstellt, um Vorwärts- und Rückwärtssuchen durchzuführen.

3.2 Energetische Anomalieerkennung in Produktionsprozessen

Die Erkennung von Anomalien in Produktionsprozessen anhand von Energiedaten von Fertigungsmitteln ist eine Möglichkeit der Erkennung von Qualitätsmängeln und ineffizienten Prozessen in der Produktion. In der wissenschaftlichen Literatur werden verschiedene Verfahren zur energetischen Anomalieerkennung eingesetzt (siehe Abschnitt 2.2.3), darunter distanz-, klassifikations- und rekonstruktionsbasierte Verfahren. Durch das Trainieren eines Modells mit historischen Maschinen- und Betriebsdaten eines Produktionsprozesses können Parameter einer Funktion approximiert werden, um bei parametrischen Modellen (z. B. rekonstruktions- oder klassifikationsbasierte Verfahren) die optimalen Parameter zu identifizieren. Nichtparametrische Modelle hingegen werden durch algorithmische Verfahren optimiert.

Insbesondere die Nutzung von Energiedaten bietet großes Potenzial zur Anomalieerkennung, da diese Daten aufschlussreiche Informationen über den Zustand von Maschinen und Anlagen liefern. In diesem Abschnitt werden Ansätze zur energetischen Anomalieerkennung in Produktionsprozessen mit Energiedaten untersucht und bewertet.

Von *Chen et al.* wird ein Ansatz zur Erkennung von Anomalien in industriellen Maschinen anhand des gemessenen Energieverbrauchs vorgeschlagen. Es werden sowohl ein rein statistischer, auf der Mahalanobis-Distanz basierender als auch ein datengetriebener Ansatz mit KNN genutzt. Echte und synthetische Energiedaten werden für die Analyse, das Training, Modellierung und Erkennung von Anomalien verwendet. Laut den Autoren sind hinsichtlich der Erkennung von Anomalien anhand des Energieverbrauchs datengetriebene Modelle für komplexe Produktionsprozessen vielversprechender als statistische Verfahren. Neben dem Energieverbrauch werden auch andere Arten Daten, z. B. Vibration und Temperatur, einbezogen, um die Erkennung von Anomalien zu verbessern (Chen et al. 2014).

Faltinski et al. stellen einen neuartigen modellbasierten und überwachten Ansatz für die Vorhersage des Energieverbrauchs in Produktionsanlagen mit dem Ziel vor, energetische Anomalien zu erkennen. Für das System wird ein Modell verwendet., das auf einem hybriden Zeitautomaten basiert und teilweise auf Expertenwissen fußt. Dabei sind Anomalien als Abweichungen des realen Energieverbrauchs einer Anlage von der Vorhersage des Energieverbrauchs definiert. Zusätzlich wird ein Modell für den Energieverbrauch eines gesamten Produktionssystems entwickelt. Auf Grundlage dieses Modells können Anomalien in der momentanen elektrischen Leistungsaufnahme von Anlagen erkannt werden. Dies unterstützt eine erweiterte Zustandsüberwachung und hilft dabei, abnormales Verhalten bestimmter Komponenten automatisch zu erkennen (Faltinski et al. 2012).

Flick et al. entwickeln ein zweistufiges Verfahren zur Erkennung energetischer Anomalien, indem ein Standardlastprofil des Produktionsprozesses generiert wird sowie Betriebsmodi erkannt und prognostiziert werden. Weichen diese Prognosen der Betriebsmodi vom Standardlastprofil ab, werden die betreffenden Zeitintervalle als *Anomalie* klassifiziert. Die

Klassifikation erfolgt durch den Vergleich mit historischen Mittelwerten. Zusätzlich werden künstliche neuronale Netze genutzt, um weitere Faktoren wie Wetterdaten mit einzubeziehen (Flick et al. 2020).

Kroll et al. stellen einen Ansatz vor, der zeitgesteuerte hybride Automaten des normalen Verhaltens von Maschinen für eine vorausschauende Wartung von Industrieanlagen verwendet. Dieses theoriebasierte Systemmodell reduziert diskrete und kontinuierliche Energiedaten auf einzelne Zustände, die sich auf den aktuellen Zustand der Maschinen beziehen. Dies ermöglicht eine Erkennung von Anomalien durch die Implementierung eines kombinierten Datenerfassungs- und Anomalieerkennungsansatzes und den Ausblick auf andere Anwendungen, wie z. B. eine vorausschauende Wartungsplanung (Kroll et al. 2014).

Liang et al. stellen ein zweistufiges Verfahren zur energetischen Anomalieerkennung im Prozess der Aluminiumextrusion vor. Im ersten Schritt wird eine konventionelle Lastprognose mittels historischer Energiedaten des Prozesses durchgeführt. Anhand der Abweichung zwischen der Prognose und dem tatsächlichen Lastgang können Anomalien erkannt werden. Zusätzlich entwickeln die Autoren einen Ansatz für neue Extrudermaschinen ohne historische Daten (Liang et al. 2018).

Lorenti et al. stellen in ihrer Arbeit den Ansatz CUAD-Mo vor, der eine unüberwachte und interpretierbare Anomalieerkennung aus CNC-Daten und der Energiedaten vorschlägt. Dabei werden während des Prozesses Merkmale aus den Zeitreihen extrahiert, um Anomalien zu erkennen und Post-hoc-Erklärungen für die erkannten Anomalien zu liefern. Hierzu wird der Isolation-Forest-Algorithmus verwendet, um implizit die Wichtigkeit der verschiedenen Merkmale zu bewerten. Zur Validierung wurden zusätzlich Evaluationsmetriken verwendet, die sich nicht nur auf das Modell, sondern auch auf die Qualität des Endproduktes beziehen. Somit können direkte Maßnahmen zur Qualitätsverbesserung abgeleitet werden (Lorenti et al. 2022).

Parvathy et al. stellen in ihrer Arbeit einen methodischen Ansatz zur Erkennung energetischer Anomalien der angeschlossenen elektrischen Verbraucher in der betrachteten Produktionseinheit durch die Analyse von Energiedaten vor. Zur Erkennung

energetischer Anomalien werden unterschiedliche Modelle wie klassifikationsbasierte Verfahren, aber auch Segmentierungsverfahren genutzt. Aus der Analyse und Anomalieerkennung werden wirksame Schlussfolgerungen gezogen und wirtschaftliche Empfehlungen ausgesprochen, die der Betriebsleitung mitgeteilt werden (Parvathy et al. 2020).

Petladwala et al. nutzen eine korrelationsbasierte Methode zur Merkmalsextraktion für die Erkennung energetischer Anomalien in elektrischen Maschinen. Dabei werden die ungeraden und geraden Abtastrateoberschwingungen der elektrischen Stromsignale aufgespaltet und mithilfe der kanonischen Korrelationsanalyse transformiert. Auswertungen zeigen, dass der vorgeschlagene Ansatz die Erkennungsrate von Anomalien anhand realer Daten gegenüber konventionellen unüberwachten Methoden zur Merkmalsextraktion übertrifft (Petladwala et al. 2019).

Soni et al. beschreiben ein Konzept zur Erkennung energetischer und betrieblicher Anomalien in energieintensiven Prozessen, die zu Energieverschwendung, Gerätefehlfunktionen und Verlusten oder minderwertigen Produktionsergebnissen führen können. Durch die Installation einer großen Anzahl von Stromzählern auf allen Ebenen sollen historische Daten für eine bessere Erkennung von Anomalien genutzt werden. Zu diesem Zweck werden Stützvektormaschinen eingesetzt (Soni et al. 2016).

Theumer et al. nutzen einen GRU-Autoencoder für die rein unüberwachte Erkennung energetischer Anomalien in Produktionsprozessen. Dabei wird der mittlere absolute skalierte Fehler für die Rekonstruktionsfähigkeit des Autoencoders als Metrik zur Erkennung der energetischen Anomalien genutzt. Der Ansatz wird auf künstlich erstellten Zeitreihendaten und synthetisch injizierten Anomalien validiert (Theumer et al. 2021).

Wen et al. entwickeln ein Verfahren für die Anomalieerkennung von Industrierobotern. Dafür wird eine Long-Short-Term-Memory(LSTM)-Autoencoder-Architektur genutzt, die eine unüberwachte Erkennung von Anomalien für Industrieroboterprozesse (z. B. Schleifen, Schweißen und Montieren) ermöglicht. Zusätzlich werden industrielle Kommunikationsprotokolle wie z. B. OPC UA genutzt und mit ML-Methoden kombiniert. Da Industrieroboter für unterschiedliche Produktionsprozesse genutzt werden können, ist

auch die Anomalieerkennung in heterogenen Produktionssystemen möglich (Wen & Chen 2020).

In der Arbeit von *Windmann et al.* wird die Erkennung von Anomalien in Energiedaten in industriellen Produktionssystemen untersucht. Der Ansatz basiert auf der Annahme von mehreren Betriebsmodi, d. h. Phasen mit kontinuierlichem oder binärem Systemverhalten. Der Schwerpunkt dieser Arbeit liegt auf der physikalischen Modellierung des Energieverbrauchs in den einzelnen Systemmodi. Dazu werden Sequenzen von stochastischen Zustandsraummodellen verwendet. Der vorgeschlagene Ansatz wird in einer kleinen Modellfabrik evaluiert (Windmann et al. 2013).

Xu et al. erkennen Anomalien anhand von Energiedaten in der Garnherstellung, um Störungen im Produktionsprozess zu vermeiden. Dabei nutzen die Autoren Modelle aus ähnlichen Anwendungen der Anomalieerkennung zum Approximieren des eigentlichen Modells. Zusätzlich wird eine Adaptionsebene eingeführt, die durch das unüberwachte Segmentieren von Daten in unterschiedlichen Gruppe eine gezieltere Modellierung ermöglichen (Xu et al. 2021).

Yang et al. entwickeln eine Methode zur Erkennung von Anomalien hinsichtlich der Energieeffizienz im Reifenhärtungsprozess. Dafür werden die unterschiedlichen Datenquellen integriert und generische Prognosemodelle genutzt. Durch den Miteinbezug externer Einflussfaktoren wie etwa Wetter oder Produktionsarbeiter können energetische Anomalien erkannt und die Robustheit der Lösung kann gesteigert werden (Yang et al. 2012).

Zhang et al. erweitern das Konzept der cyberphysischen Systeme hinsichtlich eines energiesensitiven Moduls. Dabei wird ein ganzheitliches Datenverarbeitungsmodul für Energiedaten mit unterschiedlichen Teilschritten vorgestellt. Das Modul beinhaltet die Datenakquise bis hin zur eigentlichen Analyse und Modellierung der energetischen Anomalieerkennung anhand des Energieverbrauchs. Die vorgeschlagene Methode baut auf dem Konzept der generativen Modelle auf und wird als *Deep Belief Network* bezeichnet (Zhang et al. 2020).

Zhang et al. stellen einen Ansatz zur Erkennung von Anomalien und energieeffizienten Produktionsentscheidungen vor. Zunächst wird hierbei eine Architektur für die Erkennung von Anomalien und eine energieeffiziente Umplanung vorgestellt. Daraufhin wird ein Algorithmus zur Vorverarbeitung von Energiedaten entwickelt. Zur Modellierung der Anomalien wird eine LSTM-Architektur genutzt. Bei einer auftretenden Anomalie wird diese nicht nur erkannt, sondern es werden auch Gegenmaßnahmen wie die Anpassung von Wartungsintervallen eingeleitet (Zhang & Ji 2020).

3.3 Föderiertes Lernen zur Anomalieerkennung in Produktionsprozessen

Der Ausdruck *Föderiertes Lernen* beschreibt einen Ansatz, der es mehreren Clients ermöglicht, ein globales Modell für ML zu trainieren und zu nutzen, ohne die sensiblen Daten der Clients offenzulegen und gleichzeitig eine den Anforderungen entsprechende Leistung in Abhängigkeit einer Evaluationsmetrik zu erzielen (siehe Abschnitt 2.3). Ziel ist es, ein gemeinsames Modell für die unterschiedlichen Clients zu trainieren, um eine Erkennung von Anomalien aller Clients zu ermöglichen. Das FL eignet sich deshalb insbesondere, wenn Daten unternehmensübergreifend zur energetischen Anomalieerkennung genutzt werden sollen. Im Gegensatz zu einer zentralen Speicherung und Analyse der Daten werden mit FL die erfassten Daten anhand der Produktionsprozesse lokal verarbeitet bzw. werden nur die aggregierten Modellparameter an einen zentralen Server gesendet. Dadurch bleiben die eigentlichen sensiblen, dem Produktionsprozess zugeordneten Daten auf den lokalen Geräten, was zu höherem Datenschutz führt und insbesondere in Produktionsunternehmen mit sensiblen Maschinen- bzw. Betriebsdaten von Vorteil sein kann. Nachfolgend werden verschiedene Ansätze zur Anwendung von FL zur Anomalieerkennung in der Produktion untersucht.

Ahn et al. haben einen Ansatz für die föderierte vorausschauende Wartung in Produktionsprozessen entwickelt, der insbesondere die Herausforderungen berücksichtigt, die sich aus der Verteilung der Fabriken und der Heterogenität der Daten zwischen den einzelnen Pumpen ergeben. Sie schlagen ein Modell vor, das faltende

neuronales Netzwerke (CNN) und LSTM miteinander kombiniert, um Merkmale aus Zeitreihendaten effizient zu extrahieren und Anomalien zu erkennen. Die Evaluierung anhand des öffentlichen Pumpendatensatzes zeigt, dass der vorgeschlagene Ansatz eine hohe Genauigkeit aufweist (Ahn et al. 2023).

Der Schwerpunkt von *Bagheri et al.* liegt auf der Entwicklung eines Frameworks für die Verarbeitung von Daten aus unterschiedlichen Unternehmen zur Erkennung von Anomalien mithilfe geteilter Prognosemodelle. Dabei sollen Maschinenausfälle mithilfe von FL erkannt werden. Die Autoren entwickeln ein Architekturkonzept, das die Blockchain und die homomorphe Verschlüsselung für eine sichere Datenübertragung nutzt (Bagheri et al. 2020).

Da Silveira Dib et al. entwickeln eine Methode zur Vorhersage von Anomalien in Blech- und Umformprozessen. Daten unterschiedlicher lokaler Geräte werden verwendet, um ein zentrales ML-Modell zu trainieren. Dabei wird FL zum lokalen und simultanen Training auf unterschiedlichen Endgeräten genutzt, um die Verlustfunktion durch den Austausch der Modellparameter zu minimieren. Auf diese Weise haben die unterschiedlichen Teilnehmer keinen Zugriff auf die lokalen Daten der anderen Partner. Die erzielten Metriken zur Anomalieerkennung erweisen sich als zum herkömmlichen zentralen Training geeignet (da Silveira Dib et al. 2021).

Ge et al. entwickeln eine Architektur für die Vorhersage von produktspezifischen Störungen in Fertigungslinien. Während in anderen Arbeiten meist KNN für FL-Systeme genutzt werden, konzipieren die Autoren föderierte Stützvektormaschinen und einen föderierten Entscheidungsbaum. Durch die Validierung mit Realdaten einer Fertigungslinie zeigen sie, dass zwischen zentralen Verfahren und den FL-Systemen kein signifikanter Unterschied existiert (Ge et al. 2021).

Die Arbeit von *Geng et al.* baut auf den Arbeiten der Autoren hinsichtlich der Anomalieerkennung bei Wälzlagern auf. Dabei liegt der Fokus auf der Entwicklung eines effizienteren Modells in Abhängigkeit von der Leistung – in diesem Fall der Präzision. Abhängig von der Modellleistung erfolgt eine gewichtete Aggregation der Modellparameter, um vor allem leistungsstarke lokale Modelle zu belohnen und lokale

Modelle mit schwacher Leistung weniger zu berücksichtigen. Gegenüber dem standardmäßigen Verfahren zur Aggregation konnten die Trainingszyklen und die Leistung gesteigert werden (Geng et al. 2022).

Kanagavelu et al. entwickeln ein FL-Framework für die Vorhersage von Zielgrößen auf Basis industrieller Sensordaten. Das vorgeschlagene Framework ermöglicht mehreren Unternehmen die Implementierung von FL durch ML-Modelle ohne den Austausch von sensiblen Daten. Der Fokus der Arbeit liegt auf überwachten, klassifikationsbasierten Verfahren. Die Evaluation zeigt, dass das FL-Framework eine vergleichbare Vorhersagegenauigkeit im Vergleich zum zentralen Training aufweist (Kanagavelu et al. 2021).

Parto et al. Entwickeln eine generische Architektur zur Nutzung von ML-Methoden für industrielle Anwendungen. Der Fokus liegt auf einer eventbasierten Architektur, die das Konzept von FL umsetzt. Dabei besteht die Architektur aus drei unterschiedlichen Ebenen: Edge, Fog und Cloud. Für die Kommunikation zwischen den Ebenen werden standardisierte Kommunikationsprotokolle wie MTConnect oder OPC UA genutzt. Der Fokus dieser Arbeit liegt vor allem auf der Kommunikation und der Architektur, weniger aber auf der eigentlichen Modellierung. Der Ansatz kann auch für andere Anwendungen wie die Anomalieerkennung genutzt werden (Parto et al. 2020).

Tran et al. nutzen zur Anomalieerkennung in Produktionsprozessen einen LSTM-Autoencoder. Dabei wird für FL ein Clustering-Ansatz verwendet, der ähnliche Clients segmentiert, um die Leistung von FL zu verbessern. Hierzu verwenden die Autoren 53 Sensoren einer Industriepumpe. Die Genauigkeit, die Trefferquote und der F1-Score übertreffen den zentralen Ansatz. Zusätzlich kann mit dem Clustering-Ansatz die Zahl der Trainingsrunden reduziert werden. Indes wurde die Methode nicht kreuzvalidiert (Tran et al. 2022).

Das kollaborative *Deep Learning Framework* von *Wang et al.* bietet die Möglichkeit, FL in verteilten komplexen Systemen zu nutzen. Dabei wurden ML-Methoden hinsichtlich der Konsensbildung in verteilten Systemen erweitert. Der Fokus der Arbeit liegt grundsätzlich auf der Modellierung eines LSTM-Modells sowie dessen Anpassung auf FL (Wang, Haoxiang et al. 2021).

Wang et al. erweitern den Ansatz für die Erkennung von Anomalien hinsichtlich einer asynchronen Optimierung, um die Effizienz der Datenverarbeitung und Kommunikation zu steigern. Eine weitere Kerninnovation der Arbeit ist die Entwicklung eines Zweigmodells auf Grundlage von KNN. Jeder einzelne Zweig ist genau einer Maschine oder Anlage zugeordnet. Die Maschine oder Anlage approximiert daher genau die Modellparameter des zugehörigen Zweigs. Gegenüber konventionellen Modellarchitekturen für FL weist das Zweigmodell weniger Volatilität beim Training, jedoch keinen signifikanten Unterschied zur eigentlichen Leistung auf (Wang, Qizhao et al. 2021).

Zellinger et al. entwickeln eine modular aufgebaute Plattform für FL mithilfe des Konzepts des transferierenden Lernens. Dabei liegt der Fokus vor allem auf den heterogenen Produktionsumgebungen und dem transferierenden Lernen. Die Autoren versuchen, alle Anwendungen von FL in Probleme des transferierenden Lernens zu übertragen und diese mit Sicherheitsmechanismen zu erweitern. Eine Validierung der Plattform wird nicht durchgeführt (Zellinger et al. 2021).

Zhang et al. entwerfen ein blockchain- und FL-basiertes System für Ausfälle in Produktionsprozessen. Der Fokus der Autoren liegt zum einen auf dem Algorithmus zur Aggregation von Modellparametern, der die Distanz der einzelnen Geräte berechnet, um somit die Heterogenität der Geräte effizienter abzubilden, und zum anderen auf Anreizmechanismen zur Teilnahme an der Berechnung von Modellparametern für FL (Zhang, Weishan et al. 2021).

Für die Anomalieerkennung bei Wälzlagern entwickeln *Zhang et al.* ein CNN, das sowohl in einer zentralen als auch in einer dezentralen Umgebung mit unterschiedlichen Umgebungen validiert wird. Für die Anwendung von FL wird ein adaptiver Optimierungsalgorithmus weiterentwickelt, der aus ML übertragen worden ist. Das Modell und der adaptive Optimierungsalgorithmus werden mit einem statischen Datensatz validiert (Zhang, Zehui et al. 2021).

Zhang et al. entwickeln ein FL-Framework für die Anomalieerkennung in Produktionsprozessen. Das Framework *FedAnomaly* ist ein Konzept für FL zur

Anomalieerkennung und soll mithilfe generativer Verfahren auch nicht annotierte Anomalien erkennen. Der Fokus der Arbeit liegt auf der Entwicklung einer neuen Modellarchitektur zur Anomalieerkennung. Dieser Ansatz kombiniert verschiedene Architekturen generativer Modelle, um zeitliche und nicht annotierte Daten in Form latenter Variablen abzubilden. Die Implementierung von FL wird als einfaches Modul erweitert und ist nicht weiter spezifiziert (Zhang, Kai et al. 2021).

In der Arbeit von *Zhang et al.* wird ein *Few-Shot*-Lernansatz für die Anomalieerkennung in Produktionsprozessen vorgeschlagen. Dabei werden Daten von Industriecomputern und Geräten mit *speicherprogrammierbarer Steuerung* (SPS) genutzt. Hierbei besteht das Modell aus gestapelten spärlichen Autoencodern (SSAEs), die mithilfe von FL trainiert werden. Die SSAEs sollen Merkmale extrahieren, um verborgene Informationen in Bezug auf den Produktionsprozess aufzudecken und die Komplexität durch die Identifikation latenter Variablen zu reduzieren. Für die Validierung wird ein Permanentmagnet-Synchronmotor genutzt. Die Autoren zeigen, dass die Nutzung von SSAEs die Genauigkeit der Anomalieerkennung steigert und FL die Genauigkeit in Bezug zu den herkömmlichen Methoden nicht beeinträchtigt (Zhang, Jinglin et al. 2021).

In der Arbeit von *Zhang et al.* wird der Ansatz von FL und Anomalieerkennung für die Fehlerdiagnose von Maschinendaten vorgeschlagen. Das Modelltraining wird lokal auf jedem Client implementiert. Dabei werden teilüberwachte Verfahren bei limitierter Datenmenge für das Training genutzt. Der Server aggregiert die lokal aktualisierten Modelle in jeder Trainingsrunde und erstellt ein globales Fehlerdiagnosemodell durch die Aggregation der lokalen Modellparameter. Die Experimente mit Daten von Drehmaschinen zeigen jedoch einen signifikanten Unterschied in der Vorhersagequalität des vorgestellten Fehlerdiagnosemodells (Zhang, Wei et al. 2021).

Zhang et al. erweitern FL durch den Einbezug von A-priori-Verteilung der einzelnen Maschinen oder Anlagen. Somit können unterschiedliche heterogene Verteilungen durch diverse Produktionsökosysteme innerhalb des Systems abgebildet werden. Modelle werden mithilfe annotierter Daten und eines KNN trainiert. Die lokalen Modelle werden unter Einsatz des FedAvg-Algorithmus aggregiert und können mithilfe der A-priori-

Verteilungen in andere Umgebungen transferiert werden. Konkrete Methoden zur Verschlüsselung der Datenkommunikation werden nicht genannt (Zhang & Li 2022).

3.4 Zwischenfazit

Nach der systematischen Literaturanalyse nach vom Brocke et al. (2009) des Standes der Wissenschaft und Technik in diesem Kapitel konnten inhaltlich relevante Ansätze identifiziert und beschrieben werden. In Bezug auf die in dieser Arbeit definierte Forschungsfrage zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen wurden zwei Forschungsgebiete näher betrachtet. Zum einen wurden die verwendeten Methoden und Ansätze zur energetischen Anomalieerkennung in Produktionsprozessen untersucht. Zum anderen wurden bereits existierende Ansätze untersucht, die ein FL-basiertes System in Produktionsprozessen einsetzen.

4 Anforderungen und Handlungsbedarf

Dieses Kapitel dient dazu, die Anforderungen an den Einsatz eines Systems zur energetischen Anomalieerkennung in Produktionsprozessen auf Basis von FL zu definieren. Dazu wird unter Berücksichtigung der bereits dargestellten Grundlagen der Stand von Wissenschaft und Technik analysiert. Im zweiten Schritt werden diese Ansätze den definierten Anforderungen gegenübergestellt und es wird der gebotene Handlungsbedarf abgeleitet. Dieses Kapitel bildet somit die Grundlage für die Entwicklung eines eigenen Lösungsansatzes in Kapitel 5.

4.1 Definition und Ableitung der Anforderungen

Aufbauend auf den Grundlagen in Kapitel 2, der systematischen Literaturanalyse in Kapitel 3 und den bisherigen Veröffentlichungen des Autors werden im vorliegenden Abschnitt Anforderungen abgeleitet, dargestellt und beschrieben (Kaymakci et al. 2021a). In der Vorveröffentlichung des Autors wurden bereits erste Anforderungen für den Einsatz von FL in verschiedenen Anwendungsbereichen analysiert und bewertet (Karnebogen et al. 2023). Darüber hinaus wurde eine Taxonomie für den Einsatz von FL entworfen und hierfür Dimensionen mit unterschiedlichen Ausprägungen entwickelt. Darauf aufbauend, sollen in diesem Abschnitt die Anforderungen an ein FL-basiertes System zur energetischen Anomalieerkennung in Produktionsprozessen konkretisiert werden.

Die Anforderungen lassen sich in drei Kategorien einteilen. Erstens bestehen spezifische Anforderungen an die eigentliche energetische Anomalieerkennung in Produktionsprozessen, da bei der Erkennung und Klassifikation von Anomalien aus Maschinen- und Betriebsdaten, insbesondere Energiedaten, technische Restriktionen berücksichtigt werden müssen. Zusätzlich werden in Anlehnung an Karnebogen et al. (2023) spezifische Anforderungen an FL definiert. Da FL hinsichtlich seiner

Anwendungsgebiete vor allem im Konsumentenbereich weit verbreitet ist, soll der Einsatz von FL für die energetische Anomalieerkennung in Produktionsprozessen durch die Anforderungen eingegrenzt werden. Darüber hinaus werden Anforderungen an eine verteilte Systemarchitektur definiert, die ohne zentrale Datenverwaltung, -speicherung und -analyse auskommt, um den Schutz sensibler Daten aus Produktionsprozessen zu gewährleisten.

Energetische Anomalieerkennung

In der Literatur existieren zahlreiche Modelle zur Anomalieerkennung, die unterschiedliche Datenformate wie Video, Audio oder strukturierte Daten als Eingangsdaten verwenden (Kiran et al. 2018). Für die Anomalieerkennung aus Energiedaten ist die Berücksichtigung von Zeitreihen notwendig, da Sensordaten von Maschinen und Anlagen eine zeitliche Abhängigkeit aufweisen. Bei der Auswahl von Modellen ist daher die ***Eignung für Zeitreihen*** (A1) zu prüfen. Bei der energetischen Anomalieerkennung sollen Störungen oder Ausfälle in Produktionsprozessen anhand von Energiedaten frühzeitig erkannt werden. Für die praxisorientierte Umsetzung ist zusätzlich die Definition und Ausführung von Gegenmaßnahmen relevant, die die auftretende Störung verhindern oder beseitigen. Mittels gezielter ***Maßnahmenintegration*** (A2) in die energetische Anomalieerkennung kann durch die Verarbeitung der Modellergebnisse trennscharf auf die Störung reagiert werden. Die meisten Anwendungen im Bereich der Anomalieerkennung in Produktionsprozessen nutzen zahlreiche Sensoren zur Erfassung von Betriebs- und Prozessdaten (Theumer et al. 2021). Um den Installationsaufwand zusätzlicher Sensoren zu vermeiden und den Rechenaufwand bei der Datenverarbeitung zu minimieren, sollen möglichst wenige Daten verwendet werden. Es konnte bereits in Vorarbeiten des Autors gezeigt werden, dass die Verwendung von ML-Modellen in der Industrie zu erhöhtem Energieverbrauch führt und somit nicht zwingend mit einem Effizienzgewinn einhergehen muss (Wenninger et al. 2022). Eine Möglichkeit zur Vermeidung eines erhöhten Verbrauchs an Ressourcen und Rechenkapazität ist die Reduktion der notwendigen Daten. Die ***Dateneffizienz*** (A3) erfordert daher eine Minimierung der für die

energetische Anomalieerkennung verwendeten Daten, ohne dass die Minimierung der Daten signifikante Auswirkungen auf die Evaluationsmetrik hat.

Föderiertes Lernen

Daten, die lokal durch Anwender von Maschinen und Anlagen erfasst werden, sollen von außen nicht zugänglich sein und nicht zentral gespeichert werden. Dennoch sollen Verfahren zur Approximation von Vorhersagemodellen anhand der lokal gespeicherten Daten funktionieren. Unter Einbezug von Energiedaten, die Produktionsprozessen zugeordnet werden, können Rückschlüsse auf diese gezogen werden. Infolge der Sensibilität von Maschinen- und Betriebsdaten ist der ***Schutz sensibler Maschinen- und Betriebsdaten*** (FL1) eine wichtige Anforderung an FL-basierte System zur energetischen Anomalieerkennung in Produktionsprozessen. Zusätzlich soll das global trainierte Modell die anwendungsspezifische Zahl der Erkennung energetischer Anomalien erhöhen. Hierfür sollen die Evalutationsmetriken durch die in dieser Arbeit entwickelte Methode im Vergleich zu ML-Modellen, die auf lokalen Energiedaten trainiert wurden, verbessert werden. Die ***anwenderspezifische Verbesserung*** (FL2) der Erkennung energetischer Anomalien ist somit eine grundlegende Voraussetzung für die Anwendung von FL. Angesichts der unterschiedlichen und heterogenen Daten der Maschinen und Anlagen ist auch eine ***Berücksichtigung lokaler Evaluationsmetriken*** (FL3) unerlässlich – nicht nur, um die Datenanalyse und -nutzung in den individuellen Produktionsprozessen zu optimieren, sondern auch für die Gewährleistung einer gleichbleibenden Erkennungsrate durch FL. Ein solcher Ansatz erfordert Mechanismen, die sicherstellen, dass alle Daten der Anwender, unabhängig von ihrer Größe oder Art, angemessen berücksichtigt werden. Dies sollte sich entsprechend in den individuellen Evaluationsmetriken der einzelnen Clients widerspiegeln.

Aufgrund der unterschiedlichen Nutzung von Maschinen und Anlagen in Produktionsprozessen sowie der Variation von Einstellparametern ergeben sich unterschiedliche Wahrscheinlichkeitsverteilungen anwendungsspezifischer Maschinen- und Betriebsdaten, insbesondere bei der Herstellung unterschiedlicher Produkte. Werden die erfassten Daten nur lokal und nicht zentral gespeichert, wie es in den meisten Fällen – insbesondere in der Feldebene – der Fall ist, können Synergieeffekte von Maschinen in

ähnlichen industriellen Prozessen nicht genutzt werden. Zusätzlich hängt die Anwendung von FL von der ***Parameterkonfiguration*** (FL4) ab. Insbesondere die Wahl des geeigneten Aggregationsmechanismus steht hierbei im Vordergrund, da verschiedene Mechanismen unterschiedliche Auswirkungen auf die definierten Evaluationsmetriken und die Modellkonvergenz haben können. Für eine praxisnahe Anwendung der energetischen Anomalieerkennung in Produktionsprozessen ist es essenziell, zu bestimmen, welche *Konfigurationsparameter* für den spezifischen Anwendungsfall der energetischen Anomalieerkennung in Produktionsprozessen geeignet sind. Die Variation dieser Parameter kann signifikante Auswirkungen auf die Evaluationsmetriken haben, weshalb eine systematische Untersuchung und Bestimmung der optimalen Werte erforderlich ist.

Verteilte Systemarchitektur

Eine Anforderung für den entwickelten Ansatz ist die ***Übertragbarkeit auf diverse Produktionsökosysteme*** (S1). Die Maschinen und Anlagen eines Herstellers können in verschiedenen Produktionsumgebungen eingesetzt werden. Eine Lösung sollte daher nicht auf einen spezifischen Anwendungsfall beschränkt, sondern flexibel und skalierbar genug sein, um in verschiedenen Anwendungsbereichen eingesetzt werden zu können. Eine umfassende Berücksichtigung der Vielfalt von Produktionsumgebungen und -prozessen ist daher von großer Bedeutung, um breite Anwendbarkeit und Flexibilität zu erreichen. Eine weitere Anforderung an die Systemarchitektur zur energetischen Anomalieerkennung in Produktionsprozessen ist die Berücksichtigung einer ***ganzheitlichen Datenverarbeitung*** (S2). Eine solche Architektur umfasst nicht nur die Entwicklung und Implementierung eines Modells, sondern auch die Entwicklung der Datenverarbeitungsschritte. Dabei müssen alle Aspekte der Datenverarbeitung vom Datenimport über die Datenverarbeitung bis hin zur Datenausgabe an weitere Teilsysteme in der Produktion berücksichtigt werden. Nur durch eine umfassende Betrachtung der gesamten Architektur kann die Nutzung des Systems gewährleistet werden. Eine ganzheitliche Datenverarbeitung ermöglicht eine nahtlose Integration der Datenverarbeitung sowie die Modellierung in bestehende Maschinen und Anlagen. Da in vielen Produktionssystemen bereit IT-Systeme vorhanden sind, ist eine ***Integration in***

bestehende Systeme (S3) eine große Herausforderung, um eine reibungslose Funktionalität zu ermöglichen. Entscheidend ist auch die Interoperabilität mit anderen Systemen und Plattformen. Per standardisierter Schnittstelle kann eine Integration in vorhandene Produktionsumgebungen gewährleistet werden. Durch den modularen und standardisierten Aufbau soll der entwickelte Ansatz auch in bestehende Systeme integriert werden und somit den Implementierungsaufwand reduzieren.

4.2 Einordnung der bestehenden Ansätze

In diesem Abschnitt erfolgt die Bewertung der vorgestellten Arbeiten hinsichtlich des Standes der Technik (siehe Kapitel 3) auf Grundlage der zuvor definierten Anforderungen, die für die Beantwortung der zweiten Forschungsfrage erforderlich sind. Die untersuchten Ansätze sind in zwei verschiedene Gruppen gegliedert. In Abschnitt 4.2.1 sind die Arbeiten mit einem Fokus auf energetischer Anomalieerkennung in Produktionsprozessen zu bewerten. Darauffolgend werden in Abschnitt 4.2.2 bestehende Ansätze bewertet, die FL zur Anomalieerkennung in der Produktion nutzen.

Zur Bewertung der Erfüllung spezifischer Anforderungen werden drei distinkte Kategorien des Erfüllungsgrades eingeführt. Diese umfassen die Kategorien „nicht erfüllt", „teilweise erfüllt" und „vollständig erfüllt". Die Kategorie „nicht erfüllt" wird vergeben, wenn ein Ansatz die definierten Anforderungen überhaupt nicht erfüllt. Dem gegenüber steht die Kategorie „vollständig erfüllt", die angibt, dass die Anforderung zur vollständig erfüllt wurden. Die mittlere Kategorie „teilweise erfüllt" dient der Kennzeichnung einer partiellen Erfüllung der Anforderungen, bei der zwar wesentliche Aspekte adressiert, jedoch nicht alle Kriterien der Anforderung komplett abgedeckt wurden.

4.2.1 Energetische Anomalieerkennung

In diesem Abschnitt werden die Ansätze zur energetischen Anomalieerkennung in Produktionsprozessen anhand der zuvor definierten Anforderungen bewertet. Die

Anforderungen in der Kategorie der energetischen Anomalieerkennung werden von den bestehenden Arbeiten nur teilweise erfüllt. Auffällig ist die größtenteils fehlende Integration und Einleitung von Gegenmaßnahmen bei Erkennung einer Anomalie. Eine Ausnahme ist die Arbeit von Zhang et al. (Zhang & Ji 2020), die auf Basis von Entscheidungsunterstützungssystemen ein System zur energetischen Anomalieerkennung in Produktionsprozessen entwickeln. Bei der Erkennung einer Anomalie anhand der Energiedaten wird bspw. die Gegenmaßnahme *Wartung umplanen* ausgeführt und über das Entscheidungsunterstützungssystem kommuniziert. Des Weiteren verfolgen einige Ansätze (Faltinski et al. 2012; Yang et al. 2012; Liang et al. 2018; Zhang & Ji 2020) die ausschließliche Nutzung von Energiedaten zur Erkennung von Anomalien als dateneffiziente Methode, um Kosten zu sparen und vorhandene Sensoren für die Anomalieerkennung aufwandsarm zu nutzen. Dennoch werden oft auch unterschiedliche Datenquellen mit angebunden, um die vorhandene Datenmenge zu erhöhen. Zusätzlich erfolgt in keiner der Arbeiten eine Analyse des Einflusses der Abtastraten auf die Erkennung von Anomalien. Auch eine Analyse zur Reduzierung der Abtastrate zur Messung des Energieverbrauchs einer Maschine oder Anlage könnte die Datenmenge reduzieren. Die vollständige Bewertung des Erfüllungsgrades der Anforderungen zeigt Abbildung 4-1.

Anforderungen / Autor	A1 - Eignung für Zeitreihen	A2 - Maßnahmenintegration	A3 - Dateneffizienz	FL1 - Schutz sensibler Maschinen- und Betriebsdaten	FL2 - Anwenderspezifische Verbesserung	FL3 - Berücksichtigung lokaler Evaluationsmetriken	FL4 - Parameterkonfiguration	S1 - Übertragbarkeit auf diverse Produktionsökosysteme	S2 - Ganzheitliche Datenverarbeitung	S3 - Integration in bestehende Systeme
Chen et al. (2014)	●	◐	◐	○	○	○	○	○	●	◐
Faltinski et al. (2012)	●	◐	●	○	○	○	○	○	◐	◐
Flick et al. (2020)	●	○	◐	○	○	○	○	○	○	○
Kroll et al. (2014)	●	○	◐	○	○	○	○	○	◐	◐
Liang et al. (2018)	●	○	●	○	○	◐	○	○	○	○
Lorenti eti al. (2022)	●	○	◐	○	○	○	○	○	◐	◐
Parvathy et al. (2020)	●	◐	◐	○	○	○	○	○	◐	○
Petladwala et al. (2019)	●	○	◐	○	○	○	○	○	○	○
Soni et al. (2016)	●	○	◐	○	○	○	○	○	○	○
Theumer et al. (2021)	◐	○	◐	○	○	○	○	○	●	○
Wen et al. (2020)	◐	○	○	○	○	◐	○	◐	●	●
Windmann et al. (2013)	●	○	●	○	○	○	○	○	○	○
Xu et al. (2021)	●	○	◐	○	○	◐	○	◐	○	○
Yang et al. (2012)	●	○	●	○	○	○	○	○	○	○
Zhang et al. (2020a)	●	◐	◐	○	○	○	○	◐	●	◐
Zhang et al. (2020b)	●	●	●	○	○	○	○	◐	◐	◐

Legende: ○ nicht erfüllt ◐ teilweise erfüllt ● erfüllt

Abbildung 4-1 Bewertung der Ansätze zur energetischen Anomalieerkennung in Produktionsprozessen

Die definierten Anforderungen an FL werden von keinem der Ansätze vollständig erfüllt, da die meisten Ansätze den Schwerpunkt der Arbeit auf die Modellierung und das Kombinieren unterschiedlicher Modelle richten. Damit werden die großen Herausforderungen im realen Produktionsumfeld, etwa die heterogene Datenbasis infolge der stark individuellen Nutzung von Maschinen und Anlagen, aber auch der unternehmensübergreifende Einsatz der energetischen Anomalieerkennung in Produktionsprozessen nicht berücksichtigt. Dies führt dazu, dass die Anforderungen zum Schutz von sensiblen Maschinen- und Betriebsdaten von keinem Ansatz erfüllt werden. Eine weitere Berücksichtigung der lokalen Parametrisierung von Modellen und die Heterogenität in der Datenbasis durch die unterschiedliche Anwendung der Maschinen und Anlagen in verschiedenen Prozessen erfolgt allein bei Xu et al. (Xu et al. 2021): Hier werden ML-Modelle zur energetischen Anomalieerkennung bei Spindeln von einem Anwender zu einem anderen übertragen; beide weisen unterschiedliche Wahrscheinlichkeitsverteilungen in Abhängigkeit von der Nutzung der Spindel auf. Das bereits parametrisierte Modell kann somit durch ein erneutes Training an den neuen Anwender angepasst werden. Daher ermöglicht dieser Ansatz die Nutzung der energetischen Anomalieerkennung auf Grundlage einer heterogenen Datenbasis.

Die Anforderungen an eine verteilte Systemarchitektur werden in den untersuchten Ansätzen zur energetischen Anomalieerkennung nur teilweise erfüllt. Zwar existieren Methoden, die eine Übertragbarkeit des Ansatzes auf verschiedene Produktionsökosysteme bei unterschiedlichen Anwendern ermöglichen, doch fehlt häufig eine ganzheitliche Betrachtung (Wen & Chen 2020; Zhang et al. 2020; Zhang & Ji 2020; Xu et al. 2021). Einige Ansätze konzentrieren sich auf die reine Modellierung, während andere eine ganzheitliche Architektur zur energetischen Anomalieerkennung vorstellen. In vielen Arbeiten wird die Integration in bestehende Systeme entweder gänzlich vernachlässigt oder nur oberflächlich behandelt.

4.2.2 Föderiertes Lernen zur Erkennung von Anomalien

In diesem Abschnitt werden die Ansätze zu FL für die Anomalieerkennung in Produktionsprozessen anhand der zuvor definierten Anforderungen bewertet. Die vollständige Bewertung des Erfüllungsgrades der Anforderungen zeigt Abbildung 4-2. Da die in Abschnitt 3.3 vorgestellten Arbeiten zahlreiche konzeptionelle, systematische und anwendungsunabhängige Entwicklungen hinsichtlich FL für die Anomalieerkennung in Produktionsprozessen beinhalten, werden auch andere Datenquellen wie Bilder oder Videos mit einbezogen. Dahingehend erfüllen nicht alle Ansätze die Anforderung der Eignung für Zeitreihen. Auch die Anforderung einer effizienten Nutzung der Daten wird nur von Zhang et al. (Zhang, Weishan et al. 2021) erfüllt, da anstelle der mechanischen Maschinen- und Betriebsdaten ausschließlich energetische Maschinendaten genutzt werden.

Auffallend ist auch, dass keine der Arbeiten nach der Erkennung einer Anomalie anhand der Daten geeignete Maßnahmen integriert. Die Anforderungen des Schutzes der Maschinen- und Betriebsdaten sowie die anwenderspezifische Verbesserung der Evaluationsmetriken werden von nahezu allen Ansätzen vollständig oder teilweise erfüllt, da der Fokus der Ansätze meist auch auf der Entwicklung der Algorithmen für FL liegt. Gleichwohl berücksichtigen einige Ansätze den Umgang mit einer heterogenen Datenbasis, da auch Aggregationsmechanismen oder heterogene Daten genutzt werden. Jedoch werden in keinem der Ansätze die individuellen Evaluationsmetriken der einzelnen Clients für die Auslegung des Systems berücksichtigt. Zusätzlich werden bei vielen Ansätzen, etwa von Zhang et al. (Zhang, Zehui et al. 2021) oder Tran et al. (Tran et al. 2022), unterschiedliche Aggregationsmechanismen miteinander verglichen. Allerdings sind diese nur auf den Vergleich der Aggregationsmechanismen beschränkt, sie beziehen die Auswahl und Optimierung der Konfigurationsparameter nicht mit ein. Auch die Anforderungen hinsichtlich der verteilten Systemarchitektur sind meist nicht oder nur teilweise erfüllt. Parto et al. (2020) ragen dahingehend heraus: Ihr Modell erfüllt die jeweiligen Anforderungen der verteilten Systemarchitektur und kann sinnvoll von den anderen Ansätzen abgegrenzt werden. Auch beschreiben die Arbeiten keine Methoden

oder Möglichkeiten der Integration ihres Systems oder Architektur an bereits bestehende Systeme innerhalb der Produktion resp. der jeweiligen Maschinen oder Anlagen. Zusätzlich fehlen in den meisten Arbeiten wichtige Komponenten und Schnittstellen, die einen Ende-zu-Ende-Ansatz ermöglichen.

Anforderungen / Autor	A1 - Eignung für Zeitreihen	A2 - Maßnahmenintegration	A3 - Dateneffizienz	FL1 - Schutz sensibler Maschinen- und Betriebsdaten	FL2 - Anwenderspezifische Verbesserung	FL3 - Berücksichtigung lokaler Evaluationsmetriken	FL4 - Parameterkonfiguration	S1 - Übertragbarkeit auf diverse Produktionsökosysteme	S2 - Ganzheitliche Datenverarbeitung	S3 - Integration in bestehende Systeme
Ahn et al. (2023)	●	○	○	●	●	●	○	◐	○	○
Bagheri et al. (2020)	○	○	○	◐	○	○	○	◐	●	◐
Da Silveira Dib et al. (2021)	○	○	○	●	◐	○	○	◐	◐	○
Ge et al. (2021)	◐	○	○	●	●	○	○	○	○	◐
Geng et al. (2021)	●	○	○	●	●	◐	○	◐	○	○
Kanagavelu et al. (2021)	●	○	○	●	●	◐	○	●	○	○
Parto et al. (2020)	◐	○	○	●	○	○	○	●	●	◐
Tran et al. (2022)	●	○	○	●	●	○	○	●	○	○
Wang et al. (2021a)	●	○	○	●	●	◐	○	◐	◐	○
Wang et al. (2021b)	●	○	○	●	●	◐	○	◐	○	○
Zellinger et al (2021)	○	○	○	●	○	○	○	◐	◐	○
Zhang et al. (2021a)	●	○	○	●	●	○	○	●	◐	○
Zhang et al. (2021b)	●	○	○	●	●	◐	○	◐	◐	○
Zhang et al. (2021c)	●	○	●	●	●	◐	○	●	○	○
Zhang et al. (2021d)	●	○	◐	●	●	○	○	◐	●	○
Zhang et al. (2021e)	●	○	◐	●	◐	○	○	◐	○	○
Zhang et al. (2022)	●	○	○	●	●	◐	○	●	○	○

Legende: ○ nicht erfüllt ◐ teilweise erfüllt ● erfüllt

Abbildung 4-2 Bewertung der Ansätze zu FL für die Anomalieerkennung in Produktionsprozessen

4.3 Zwischenfazit und Handlungsbedarf

In diesem Kapitel wurden in Abschnitt 4.1 zunächst die Anforderungen für die Auslegung eines FL-basierten Systems für die energetische Anomalieerkennung in Produktionsprozessen definiert. Dabei wurden die Anforderungen in drei unterschiedliche Kategorien strukturiert: Anomalieerkennung, FL und verteilte Systemarchitektur. Aufbauend auf der Anforderungsdefinition erfolgte in Abschnitt 4.2 die Bewertung und Analyse der identifizierten Ansätze aus Kapitel 3. Es wurde festgestellt, dass im Bereich der energetischen Anomalieerkennung in Produktionsprozessen etwaige Arbeiten den Fokus auf die Modellierung und die Evaluationsmetriken von ML legen. Zusätzlich wurden erste Ideen der lokalen Verarbeitung der Energiedaten ohne die Kommunikation mit einer zentralen Instanz gewonnen. Jedoch sind diese Ansätze betont rudimentär – sie vernachlässigen die Herausforderung der zu kleinen Datenmenge auf den einzelnen lokalen Instanzen, um hinreichend leistungsfähige Modelle zu entwickeln.

Bei den bestehenden Ansätzen zu FL für die Anomalieerkennung in Produktionsprozessen werden konzeptionelle Architekturen zur Anwendung von Methoden von FL in Produktionsprozessen entwickelt oder Algorithmen in diesem Bereich neu- bzw. weiterentwickelt. Jedoch werden keine Architekturen betrachtet, die eine **ganzheitliche Auslegung** von der **Modellierung** bis zur **Integration** in bereits bestehende Systeme ermöglichen würden. Ein wichtiger Punkt ist zudem die grundlegende Annahme der Arbeiten, dass die Modellierung und die Auswahl des Modells bei der Einführung eines Systems schon abgeschlossen sind. Dies ist in der Praxis auch hinsichtlich der Adaption von ML in der Produktion sowie im Maschinen- und Anlagenbau meist nicht der Fall (Friedrich et al. 2021): Allen Ansätzen zum FL für die Anomalieerkennung in Produktionsprozessen fehlt ein **strukturiertes Vorgehen für die Auswahl und die Implementierung des ML-Modells**. Ein Großteil der betrachteten Arbeiten fokussiert sich auf die grundlegende Architektur und die Algorithmen zur Aggregation der Modellparameter für FL, wobei insbesondere die Anforderungen an den Datenschutz und die anwenderspezifische Verbesserung der Evaluationsmetriken erfüllt werden. Es zeigt sich jedoch, dass viele dieser Ansätze die spezifischen Eigenschaften von

Produktionsprozessen vernachlässigen, z. B. den **Umgang mit heterogenen Daten**. Besonders hervorzuheben sind die fehlende Berücksichtigung **individueller Evaluationsmetriken** und die **Identifikation optimaler Konfigurationsparameter** bei der Auslegung der Systeme.

In allen Arbeiten wurden Untersuchungen zur **Dateneffizienz** vernachlässigt. Dies kann insbesondere bei zeitabhängigen Energiedaten zu einem Effizienzvorteil führen. Auch die Integration von Maßnahmen wird – außer bei Zhang et al. (2021a) – in keiner der Arbeiten vollständig behandelt.

In Summe zeigt sich anhand der Bewertung und Analyse in diesem Kapitel, dass eine methodische Lösung zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen von der Entwicklung eines Modells bis zur Parametrisierung von FL in der Forschungsliteratur nicht vollumfänglich behandelt wird.

5 Systemgestaltung

Dieses Kapitel basiert auf den zuvor definierten Anforderungen (siehe Abschnitt 4) und stellt den zentralen Teil der Arbeit dar, indem es die Gestaltung des FL-basierten Systems zur Erkennung energetischer Anomalien in Produktionsprozessen beschreibt. Der Entwicklungsprozess des FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen beginnt mit der Darstellung des methodischen Vorgehens (siehe Abschnitt 5.1). Darauf folgen die Definition der Architektur eines auf FL-basierten Systems zur Erkennung energetischer Anomalien in Produktionsprozessen (siehe Abschnitt 5.2) und die Methode zur Entwicklung des Systems (siehe Abschnitt 5.3). In Kapitel 6 erfolgt die Umsetzung für zwei praxisbezogene Anwendungsfälle.

5.1 Methodisches Vorgehen

In diesem Abschnitt soll das methodische Vorgehen zur Entwicklung einer Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen vorgestellt werden. Vor der Methodenentwicklung ist der Gegenstandsbereich, in dem die Methode angewendet werden soll, zu beschreiben und abzugrenzen. Dieses geschieht basierend auf den Grundlagen (siehe Kapitel 2) sowie dem Stand der Technik und Wissenschaft (siehe Kapitel 3 und 4), aber auch ergänzend zur Zielsetzung der Arbeit (siehe Abschnitt 1.3). Der **Gegenstandsbereich** dieser Forschungsarbeit umfasst Fertigungs-, Transport- und Lagermittel in Produktionsprozessen nach Westkämper et al. (2006, S. 195).

Nachdem der Gegenstandsbereich definiert worden ist, sollen nach der Design- und Entwicklungsphase die **Artefakte** konzipiert und entwickelt werden (Peffers et al. 2007). Der Fokus dieser Phasen ist, wie bereits in Abschnitt 1.4, an den Gestaltungszyklus angelehnt, in dem auf Basis des Relevanz- und des Stringenzzyklus Artefakte zur

Beantwortung der Forschungsfrage entwickelt werden (Hevner et al. 2004). Im Rahmen des Gestaltungszyklus werden die Artefakte zunächst definiert und sodann detailliert ausgearbeitet. Im Falle dieser Arbeit sind die beiden grundlegenden Artefakte die **Architektur** (siehe Abschnitt 5.2) des FL-basierten Systems zur energetischen Anomalieerkennung und die **Methode** (siehe Abschnitt 5.3) zur Auslegung des genannten Systems.

Die **Architektur** des FL-basierten Systems bezieht sich auf die strukturelle Gestaltung des Systems – einschließlich der erforderlichen funktionalen Komponenten, die für den Betrieb des Systems erforderlich sind. Diese Architektur legt die Funktionalitäten und Interaktionen der Komponenten fest. Sie bildet die Grundlage für den Betrieb des Systems und ermöglicht die Implementierung der spezifischen Funktionen zur energetischen Anomalieerkennung. Für die Auslegung und Konfiguration der einzelnen funktionalen Komponenten der Architektur des FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen eng wird die hier vorgestellte Methode genutzt. Iterative Änderungen oder Erweiterungen der Methode können Anpassungen der Architektur erfordern und umgekehrt.

Für den genannten Gegenstandsbereich wird ein anwendungsorientierter Ansatz zur Entwicklung einer **Methode** für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen gewählt. Dabei wird dies für genau ein spezifisches und baugleiches Fertigungs-, Transport- oder Lagermittel bspw. in Form einer Maschine oder Anlage definiert, die in derselben Instanziierung bei unterschiedlichen unternehmensübergreifenden Anwendern in ein oder mehreren Unternehmen genutzt wird, sodass kein direkter Datenaustausch zwischen den unterschiedlichen Maschinen oder Anlagen möglich ist. Die Methode definiert die Aktivitäten und Schritte zur Auslegung der funktionalen Komponenten der Architektur (Offermann et al. 2010, S. 84). Die Methode leitet die Konfiguration der Architektur an, indem sie festlegt, wie Daten für die energetische Anomalieerkennung erfasst, verarbeitet und aufbereitet werden. Sie bestimmt außerdem, welche Modelle, Parameter und

Verfahren zur Parametrisierung des FL-basierten Systems innerhalb der Architektur implementiert werden müssen.

5.2 Architektur

Im Folgenden soll die Architektur des FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen vorgestellt werden. Hauptbestandteil dieser Architektur ist der Anomaliedetektor, der in Vorveröffentlichungen des Autors entwickelt und angewandt worden ist (Kaymakci et al. 2021a, 2021b). Die Architektur des Anomaliedetektors wird hinsichtlich des FL auf Grundlage der Vorarbeiten des Autors erweitert (Kaymakci et al. 2021c).

5.2.1 Anomaliedetektor

Das Ziel des Anomaliedetektors besteht darin, Unregelmäßigkeiten in Produktionsprozessen anhand von Energiedaten zu identifizieren. Die Aufgabe besteht darin, erkannte Anomalien an die entsprechenden Teilsysteme innerhalb des Produktionssystems zu übermitteln, um eine Weiterverarbeitung und das Einleiten von Maßnahmen zur Beseitigung der Unregelmäßigkeit zu ermöglichen. Der modulare Aufbau des Anomaliedetektors umfasst zentrale Komponenten und deren zugehörige Funktionalitäten (siehe Abbildung 5-1). Zunächst sind physische Objekte wie Sensoren oder Informationssysteme wie ERP-Systeme als Datenquellen definiert, die Daten zu bzw. von anderen Datenquellen senden und empfangen können. Komponenten mit einer Dateneingabe-, Datenverarbeitungs- und Datenausgabefunktionalität werden als *Datenverarbeitungseinheit* (DVE) bezeichnet. Da auch ein Modell und die Funktionalität zur Einleitung einer Gegenmaßnahme als DVE definiert werden können, werden diese als zusätzliche Komponenten *Modell* und *Agent* spezifiziert.

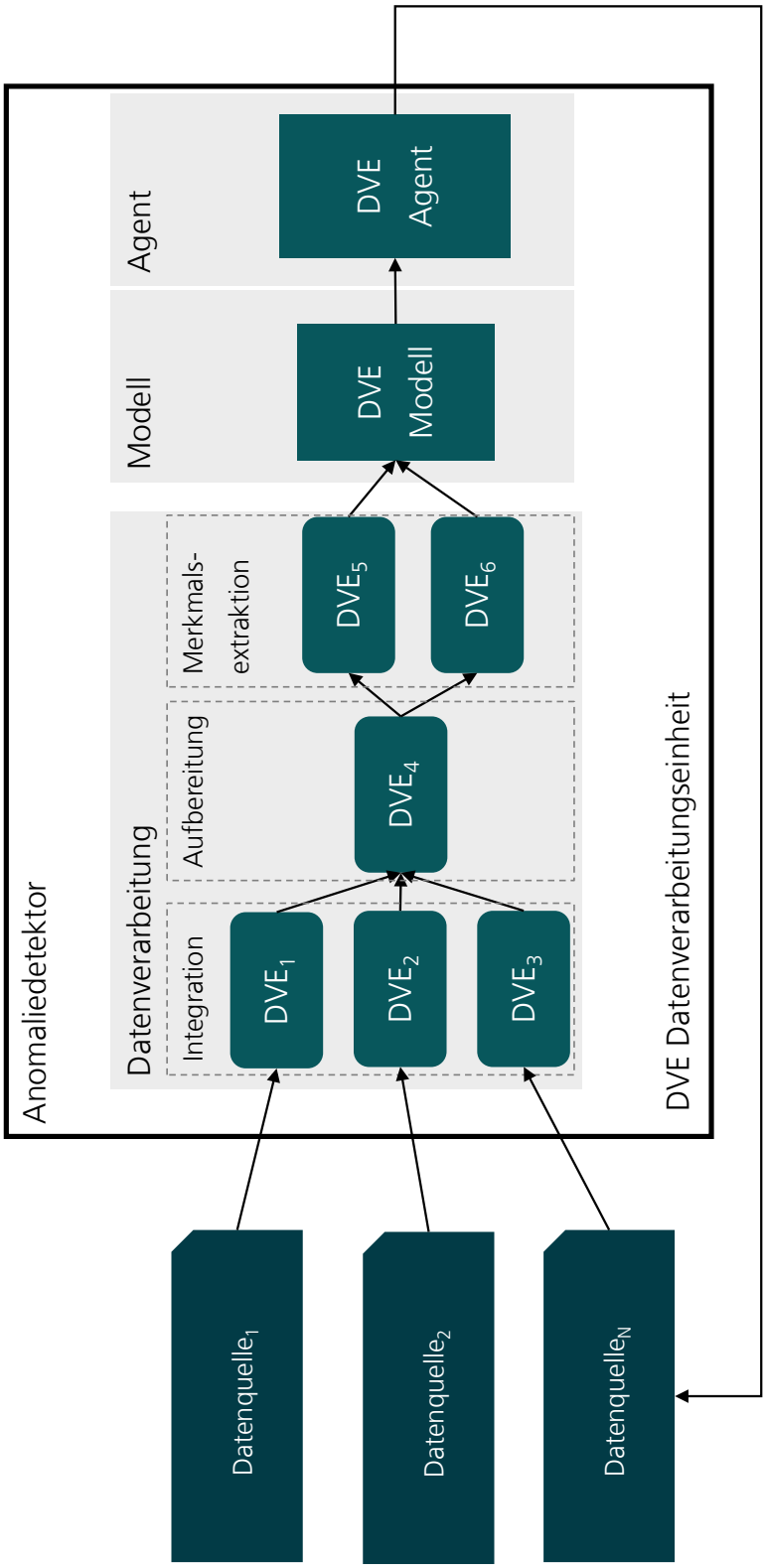


Abbildung 5-1 Exemplarische Darstellung eines Anomaliedetektors

Ein **Anomaliedetektor** besteht mindestens aus einem DVE, einem DVE-Modell und einem DVE-Agenten. Der Agent nimmt Vorhersagen des Modells als Input und verarbeitet diese zu einer Aktion für nachfolgende Systeme. Die Komponenten des Anomaliedetektors und deren Interaktionen werden in Abbildung 5-1 visualisiert. Basierend auf dem Konzept von (Russell & Norvig 2021) bieten Agenten einen spezifischen Nutzen, der sich als Aktion für das Produktionssystem und die zugehörigen Teilsysteme darstellt. Darüber hinaus können Agenten in verschiedenen Formen auftreten, um die Erkennung von Anomalien zu unterstützen und Gegenmaßnahmen einzuleiten. Ein Agent besteht nicht notwendigerweise aus einem Algorithmus oder einem technischen System, sondern kann auch eine menschliche Aufgabe beinhalten, die das Ergebnis eines Modells interpretiert und eine entsprechende Aktion auslöst. Die Ausgabe des Agenten erfolgt wiederum rückgekoppelt an bereits vorgelagerte, definierte oder weitere Teilsysteme.

Die erste Komponente der Architektur wird als **Datenquelle** bezeichnet und ist als ein System definiert, das eine oder mehrere Merkmale in Form von Sensordaten oder Maschinen- und Betriebsdaten in einem bestimmten Zeitintervall an ein anderes System oder Objekt transferiert. Datenquellen sind nicht nur als physikalische Objekte (z. B. Maschine) definiert, sondern auch als Informationssysteme oder andere technische Systeme in Produktionsprozessen. Eine Datenquelle ist geeignet, Daten sowohl zu senden als auch durch Agenten des Anomaliedetektors zu empfangen. Ein Beispiel ist die SPS, die Daten von Sensoren oder Aktoren durch Eingangssignale empfängt und diese als Ausgangssignal entweder an jeweilige Sensoren oder Aktoren sowie an nachfolgende Systeme sendet, etwa an ein ERP-System oder ein EnMS. Die Metadaten der einzelnen Attribute werden in der Datenquelle gespeichert (Datentyp, Format etc.). Für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen ist die Identifikation und Definition notwendiger Datenquellen der industriellen Anlage und Maschine erforderlich. Domänenexpertise und Erfahrungen aus anderen Systemen ermöglichen die Identifikation relevanter Datenquellen aus dem Umfeld, die das Ergebnis der energetischen Anomalieerkennung beeinflussen können. Für den individuellen und beispielhaften Prozess der Erkennung von Anomalien mithilfe von

Energiedaten müssen mögliche Datenquellen wie Sensoren zur Erfassung von Stromstärke, Spannung oder Massen- und Volumenstrom identifiziert werden.

Eine **DVE** ist als ein Basisfunktionsmodul innerhalb der Architektur definiert, die aus einem Dateneingabe-, Datentransformations- und Datenausgabemodul besteht. Das Transformationsmodul verarbeitet die Eingabe durch explizite und regelbasierte Anweisungen oder bestimmte mathematische Funktionen zu einer festgelegten Datenausgabe. Die Transformation einer Dateneingabe in eine Datenausgabe erfolgt mit verschiedenen Werkzeugen und Methoden. Die Anzahl der DVE kann je nach Bedarf variiert und an den Einzelfall angepasst werden. Die Modellierung und Implementierung von DVE lässt sich für Datenvorverarbeitungsprozesse und die Merkmalsextraktion verwenden. Eine spezifische Aufgabe einer DVE könnte das Löschen von Duplikaten, die Rekonstruktion fehlender Werte, die Skalierung von Eingabeparametern oder die Standardisierung von Werten sein.

Das **Modell** im Anomaliedetektor basiert auf einer DVE mit spezifischen Eigenschaften und berücksichtigt die Transformation der Energiedaten in eine Zielvariable. Bei einer binären Klassifikation wird die Zielvariable als binäre Variable definiert. Die Modellierung der Beziehung zwischen der Dateneingabe und der gewünschten Ausgabe erfolgt durch die dynamische Approximation der Modellparameter einer zuvor definierten Modellarchitektur, die auf Grundannahmen über deren Form basiert. Die Modellarchitekturen und Verfahren für die energetische Anomalieerkennung können wie in Abschnitt 2.2.3 beschrieben kategorisiert werden. Die Herausforderung besteht darin, das geeignete Modell in Abhängigkeit von den Anforderungen des Anwendungsfalls und den historischen Daten auszuwählen. Die Parameter werden approximiert, indem das Modell mit den historischen Energiedaten trainiert und somit eine Verlustfunktion durch die Anpassung der Modellparameter minimiert wird.

Ein **Agent** ist definiert als eine DVE, die die Ausgabe des Modells – das Ergebnis der Eingabe von Energiedaten in das trainierte Modell – verarbeitet und diese Ausgabe für weitere Anwendungen vorbereitet. Die Funktion des Agenten besteht in der Rolle einer Schnittstelle zwischen der Ausgabe des Modells des Anomaliedetektors und den

Teilsystemen. Die Anpassung eines Agenten an verschiedene Gegenmaßnahmen erfolgt durch die Bereitstellung spezifischer Verarbeitungslogiken der Modellergebnisse für unterschiedliche Arten von Anomalien sowie für die jeweiligen nachfolgenden Teilsysteme. Diese Flexibilität erlaubt eine Reaktion auf verschiedene Szenarien. Es sei an dieser Stelle hervorgehoben, dass der Agent nicht selbst die Gegenmaßnahmen initiiert, sondern als Vermittler zwischen dem Ergebnis des Modells und den Teilsystemen fungiert. Konkret verarbeitet der Agent die Modellergebnisse des Anomaliedetektors und bereitet sie in einer Weise auf, dass sie für die Teilsysteme nutzbar sind. Diese Teilsysteme können sowohl eine SPS als auch übergeordnete Systeme wie das MES sein. Neben den übergeordneten Fällen erfüllen Agenten verschiedene Funktionen, die in der Literatur umfangreich beschrieben sind. Hofmann et al. (2020) geben dazu einen Überblick und definieren sieben allgemeine Varianten von Agenten, die für den Anomaliedetektor relevant sind. In Abhängigkeit der durch den jeweiligen Anwendungsfall definierten Problemstellung können die Funktionalitäten genutzt werden, um den Agenten entsprechend zu gestalten.

5.2.2 FL-basiertes System zur energetischen Anomalieerkennung

In diesem Abschnitt werden, basierend auf den Vorarbeiten des Autors, die Komponenten der Architektur für das FL-basierte System als Erweiterung des Anomaliedetektors hin zu einem System für die energetische Anomalieerkennung, das auf dem Konzept von FL beruht (FLEA-Architektur), beschrieben und erläutert (Kaymakci et al. 2021c). Hierbei besteht die Erweiterung aus den drei funktionalen Hauptkomponenten *Anlagentrainingsadapter*, *globaler Modellkoordinator* und *FL-Modelldatenbank* (siehe Abbildung 5-2). Der Anlagentrainingsadapter ist eine funktionale Komponente auf einer in einem Produktionsprozess eingesetzten Maschine und Anlage. Das Hauptziel des Anlagentrainingsadapters ist das übergeordnete Management des Anomaliedetektors (vgl. Abschnitt 5.2.1) zur Nutzung von FL und somit die bilaterale Kommunikation mit dem globalen Modellkoordinator. Während der Anlagentrainingsadapter nahe an den

Fertigungs-, Transport- und Lagermitteln implementiert wird, aggregiert der globale Modellkoordinator die Modellparameter, die von mehr als einem Anlagentrainingsadapter erfasst werden. Die Historie und die verschiedenen Änderungen der Modellparameter werden in der FL-Modelldatenbank aufgezeichnet. Letztere nutzt einen Graphen, der die Beziehungen zwischen den einzeln trainierten Modellen und dem Trainingsprozess darstellt, was eine bessere Nachvollziehbarkeit des Prozesses ermöglicht. Durch die FL-Modelldatenbank können der globale Modellkoordinator und die Anlage mithilfe des Anlagentrainingsadapters synchronisiert werden. Die FL-Modelldatenbank dient somit zur Identifikation der besten Parameter für die nächste Trainingsiteration. Des Weiteren werden innerhalb der FLEA-Architektur zwei Rollen definiert. Anwender sind primär die Anwender, die die Fertigungs-, Transport- und Lagermittel einsetzen, um mithilfe des Anomaliedetektors energetische Anomalien in ihren Produktionsprozessen zu erkennen. Die Anwender nutzen die im Feld erzeugten Daten, um das Modell innerhalb des Anomaliedetektors zu trainieren und die Modellparameter durch den Anlagentrainingsadapter standardisiert an den Betreiber zu versenden. Die Betreiber der Architektur sind die Hersteller der Fertigungs-, Transport- und Lagermittel. Sie sind verantwortlich für den Betrieb der FLEA-Architektur und die Koordination von FL.

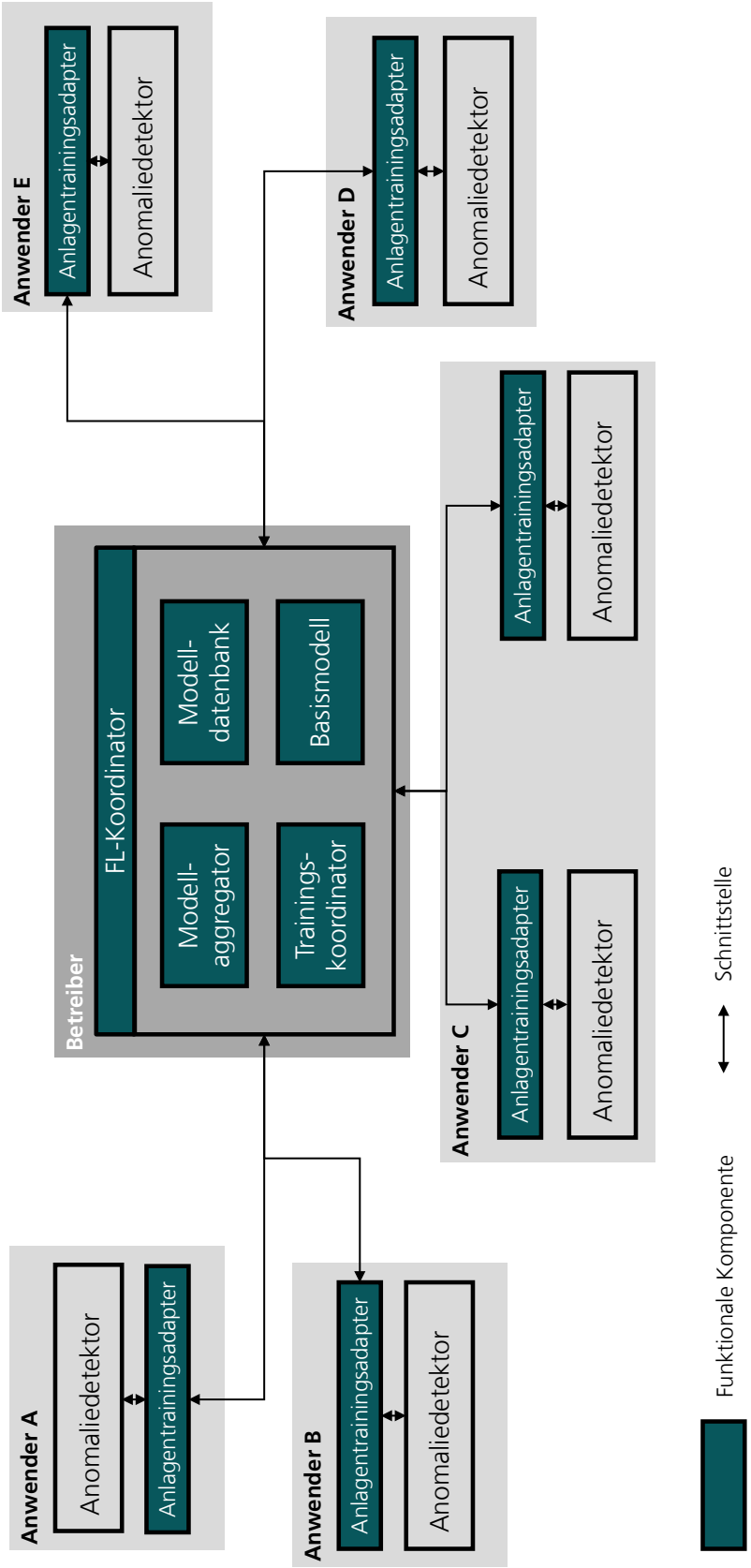


Abbildung 5-2 Beispielhafte Darstellung eines auf dem FL basierenden Systems zur energetischen Anomalieerkennung

Der **Anlagentrainingsadapter** ist die funktionale Komponente, die auf einem Client mit informationstechnischen Verarbeitungsmöglichkeiten wie z. B. Industriecomputern an Maschinen oder Anlagen lauffähig ist. Eine seiner drei Aufgaben ist die Kommunikation mit dem globalen Modellkoordinator, um Ereignisse wie die Aufforderung zum Start einer Trainingsiteration zu empfangen und über neue Modellaktualisierungen aus der FL-Modelldatenbank zu informieren. Der Anlagentrainingsadapter lädt aus Gründen der Flexibilität alle benötigten Modellspezifikationen herunter und nutzt einen Modellkompilierer, um Modelle für die Trainingsiteration zu generieren. Die Modellspezifikation ist ausschließlich deskriptiv – sie umfasst die spezifische Architektur des Modells sowie dessen Modellparameter wie etwa Gradienten oder Gewichte. Diese Parameter werden vom Modellkompilierer genutzt, um das Modell auf dem Client zu erstellen, ohne es von Grund auf neu zu trainieren. Soll eine Trainingsiteration beginnen, wird das mittels Modellkompilierer erstellte Modell verwendet und dessen Parameter werden weiter optimiert. Der in Abschnitt 5.2.1 vorgestellte Anomaliedetektor ermöglicht dabei die Integration von Daten aus verschiedenen Datenquellen, die Datenvorverarbeitung und das Training. Durch die lose Kopplung des Anlagentrainingsadapters und des Anomaliedetektors ist ein unabhängiges Training des Modells möglich.

Der **FL-Koordinator** ist der zentrale Bestandteil für die Verwaltung und Kommunikation der gesamten FLEA-Architektur. Alle Anfragen und die Initialisierung von FL werden von dieser Komponente gesendet oder empfangen. Der FL-Koordinator ermöglicht die Erstellung eines initialen *Basismodells*, das als Ausgangsmodell für die erste Trainingsiteration dient. Der *Trainingskoordinator* initiiert Trainingsiterationen, die als die Summe aller Trainings- und Kommunikationsiterationen auf den jeweiligen Anomaliedetektoren und dem zugehörigen Anlagetrainingsadapter definiert sind. Das Basismodell bildet die Grundlage für alle anderen lokalen Modelle, die während der Trainingsiterationen nacheinander erstellt werden. Der Trainingskoordinator ist zusätzlich für die Auswahl der Clients zuständig. Dabei muss der Anteil der am Training beteiligten Clients zuvor als Konfigurationsparameter definiert werden. Sobald das lokale Training beendet wurde und die Modellparameter für ein Modell über den Trainingskoordinator

in die Modelldatenbank eingegangen sind, beginnt die Aggregationsphase durch den *Modellaggregator*. Alle entsprechenden neuen Parameter der Modelle werden aus der Modelldatenbank heruntergeladen und mit Hilfe eines Aggregationsmechanismus wird ein kombiniertes Modell berechnet. Der Trainingskoordinator übernimmt nicht nur die Aufgabe, die Trainingsanfragen an die Clients zu versenden, sondern initiiert auch die Evaluation der Modelle. Dabei können nach einer Trainingsrunde auch Clients, die nicht am Training teilgenommen haben, an der Evaluation teilnehmen. Diese Clients evaluieren die lokalen Daten gegen das kombinierte Modell, um mithilfe von Evaluationsmetriken den Erfolg des Trainings und der Aggregation zu bewerten. Dazu können auch Evaluationsmetriken wie im klassischen ML verwendet werden. Bei heterogenen Produktionsprozessen ist diese verteilte Evaluierung besonders wichtig. Sie ermöglicht es, Probleme der Überanpassung zu überwinden und den Generalisierungsfehler zu reduzieren, indem die Modelle mit Daten von verschiedenen Maschinen und Anlagen evaluiert werden.

Die dokumentenorientierte *Modelldatenbank* verwaltet vier Parameter, die auch als unterschiedliche Tabellen mit gegenseitigen Verweisen betrachtet werden können: Parameter, Struktur, Metriken und Lerngraphen. Die Speicherung eines vorkompilierten Modells in einem bestimmten Datenformat in der Datenbank führt bei verschiedenen Anlagenspezifika, deren Betriebssysteme Unterschiede aufweisen, meist zu Kompatibilitätsproblemen (Faubel et al. 2023, S. 7). Aus diesem Grund werden die relevanten Informationen der Modellspezifikationen (Parameter und Struktur) nur deskriptiv gespeichert, um die Modelle mithilfe des Anlagentrainingsadapters vollständig neu zu kompilieren. Wann immer ein Modell aus der Modelldatenbank erforderlich ist, wird dessen serialisierte, deskriptive Dokumentation heruntergeladen und durch den Anlagentrainingsadapter als kompatible, anlagenabhängige Instanz kompiliert. Die Struktur des Modells wird als Referenz gespeichert, um die Struktur in der Strukturtabelle nachschlagen zu können. Die Metriken zur Evaluation werden direkt von den Clients hochgeladen und für jedes Modell-Client-Paar gespeichert, da ein Modell auf verschiedenen lokalen Anomaliedetektoren unterschiedliche Evaluationsmetriken aufweist. Durch die zentrale Speicherung kann der Evaluationsprozess vollständig

asynchron ablaufen. Da in jeder Trainingsrunde von FL ein Modell für jeden Client und ein kombiniertes Modell aus dem Aggregationsprozesses entstehen, wächst die Anzahl der Modelle binnen Kürze stark an. Um die Entwicklung der Modelle in Bezug auf Evaluationsmetriken und Verteilung zu verfolgen, wird ein gerichteter azyklischer Graph für die Modellverfolgung erstellt und jedes Mal aktualisiert, wenn ein neues Modell in die FL-Modelldatenbank kommuniziert wird.

5.3 Methode

Nachdem die grundlegende Architektur dargestellt und in den vorherigen Veröffentlichungen des Autors implementiert und validiert wurde, wird in diesem Abschnitt die Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen beschrieben (Kaymakci et al. 2021a; Kaymakci et al. 2021c). Sie beinhaltet drei Phasen, in denen die in 5.2.1 vorgestellte FLEA-Architektur und der Anomaliedetektor iterativ entwickelt werden (siehe Abbildung 5-3). Somit können Modelle auf den jeweiligen Maschinen und Anlagen für die energetische Anomalieerkennung ohne den Austausch sensibler Daten des Produktionsprozesses trainiert werden. Durch die Möglichkeit der iterativen Herangehensweise kann die Änderung einer Komponente einen neuen Iterationszyklus der Methode ab dem jeweiligen Startpunkt innerhalb der Methode initiieren. Dies ermöglicht zugleich eine Adaption zuvor entwickelter Architekturen aus anderen Anwendungsfällen.

Die erste Phase beinhaltet die Entwicklung eines Modells zur Erkennung energetischer Anomalien in Produktionsprozessen und ist an den aus der wissenschaftlichen Literatur bekannten Cross Industry Standard Process for Data Mining (CRISP-DM) Prozess zur Extraktion von Wissen anhand von Daten angelehnt (Wirth & Hipp 2000). Dies umfasst die Definition des Problems sowie der zu erkennenden Anomalien, Datenintegration und Vorverarbeitung, Extraktion relevanter Merkmale, Modellierung und Evaluation. Zentrale Bestandteile sind die Entwicklung des Modells und die in der Evaluation beinhaltete

Berücksichtigung der Dateneffizienz, die eine Auswertung der Abtastrate der Energiedaten zur Erkennung von Anomalien beinhaltet.

Nach Abschluss der Modellierungsphase erfolgt die Entwicklung des Anomaliedetektors (siehe Abschnitt 5.2.1) zur Erkennung von Anomalien an der Anlage. Die entwickelten Artefakte werden aus der Modellierungsphase in eine lauffähige energetische Anomalieerkennung in Produktionsprozessen sowohl zum Trainieren als auch für die Inferenz transformiert. Zusätzlich erfolgt die Gestaltung von anwendungsspezifischen Maßnahmen zur Kommunikation der erkannten Anomalien, die mit in den Anomaliedetektor integriert werden. Auf Grundlage der Softwareperipherie innerhalb der Anlage ist eine Bereitstellung des Anomaliedetektors erforderlich. In der letzten Phase werden die optimalen Konfigurationsparameter von FL definiert, um den Anomaliedetektor hin zum FL-basierten System zu erweitern. Der Fokus liegt auf der simulationsbasierten Konfiguration der Parameter zur Koordination und der Auswahl eines geeigneten modellabhängigen Aggregationsmechanismus.

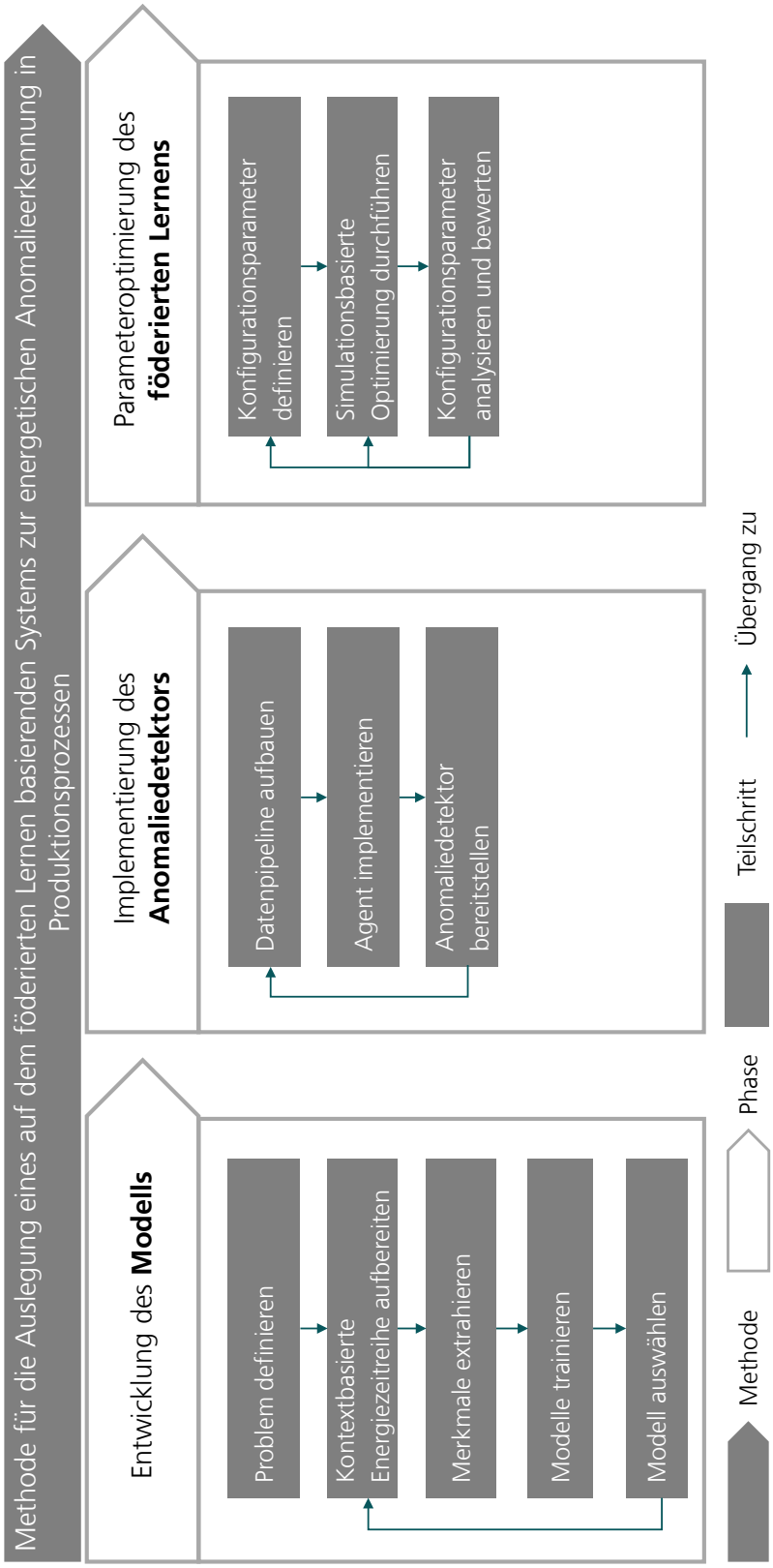


Abbildung 5-3 Methode für die Auslegung eins FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen

Die folgenden Unterabschnitte beschreiben die jeweiligen Phasen und die zugehörigen Teilschritte, die Abbildung 5-3 entnommen werden können.

5.3.1 Entwicklung des Modells

In diesem Abschnitt wird die erste Phase zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen beschrieben und in Abbildung 5-4 dargestellt. Dabei werden in der ersten Phase ein **Modell** (siehe Abschnitt 5.3.1.5), die zugehörige **Datensicht** (siehe Abschnitt 5.3.1.2) und die geeignete **Abtastrate** (siehe Abschnitt 5.3.1.5) zur Erkennung energetischer Anomalien identifiziert. Dies sind somit die **Ausgangsgrößen** der ersten Phase.

Im ersten Schritt erfolgen die Problemdefinition, angelehnt an Kaymakci et al. (2021a), und die Generierung eines Datensatzes für die weiteren Schritte. Nach der Datenaufbereitung zur kontextbasierten Energiezeitreihe und der Merkmalsextraktion folgen das Modelltraining und die Auswahl des geeigneten Modells.

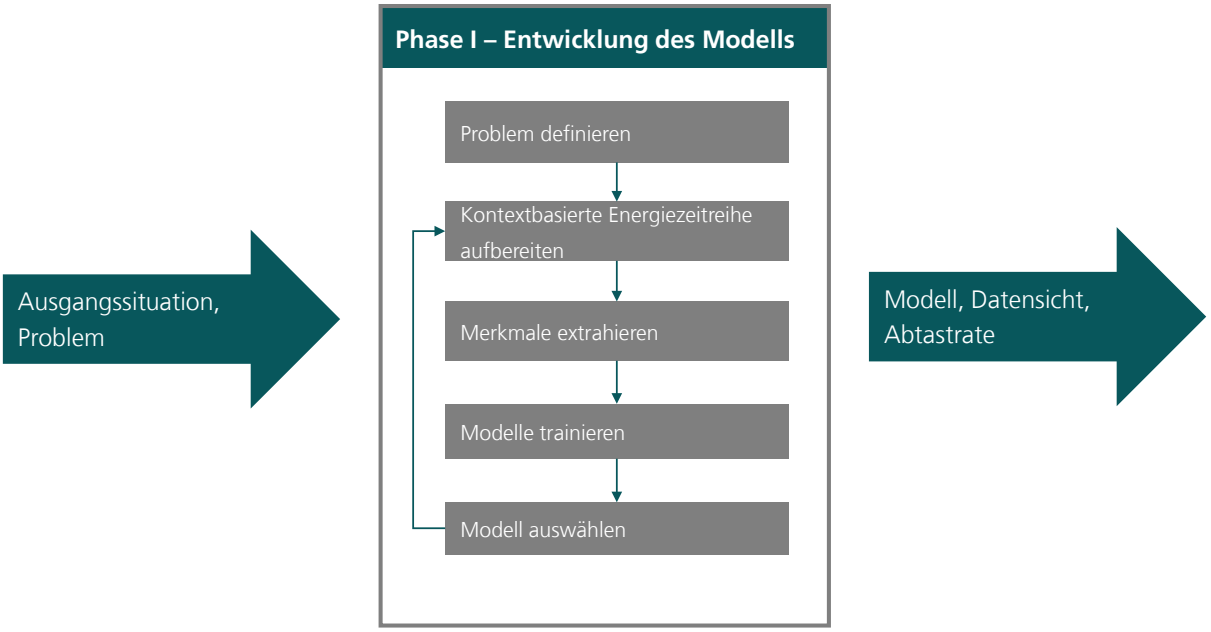


Abbildung 5-4 Phase I der Methode für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen

5.3.1.1 Problem definieren

In diesem Schritt wird auf Grundlage der **Ausgangssituation** und mithilfe des *Performance-Environment-Measures-Data*(PEMD)-Frameworks (siehe Tabelle 1), das auf der Arbeit von Russell et al. (Russell & Norvig 2021) aufbaut und in Kaymakci et al. (2021c) sukzessive für Anomaliedetektoren (siehe Abschnitt 5.2.1) weiterentwickelt worden ist, das **Problem** strukturiert definiert. Das PEMD-Framework besteht aus vier Dimensionen: Leistung, Umgebung, Maßnahmen und Daten. Die **Leistung** eines Anomaliedetektors wird durch die Evaluationsmetriken eines Modells definiert (siehe Abschnitt 2.2.4). Dabei gilt es zu beachten, dass die Evaluationsmetrik des Modells die Anforderungen des zu lösenden Problems berücksichtigt.

Tabelle 1 Exemplarische Darstellung des PEMD-Frameworks

PEMD-Framework	**Leistung**	Evaluationsmetriken: - Problem - Evaluationsmetrik
	Umgebung	- Maschinen und Anlagen - Sensoren und Aktoren - Menschen - Prozesse
	Maßnahmen	Maßnahmen, die auf Grundlage des Modells getroffen werden können
	Daten	Datenquellen: - Energiedaten - Informationssysteme - Maschinendaten - Betriebsdaten

Während die Evaluationsmetriken, z. B. für Klassifikations- oder Regressionsmodelle, hinreichend definiert sind, weisen die Zielgrößen in Abhängigkeit des Problems

Unterschiede auf. Zusätzlich ist die Beziehung zwischen den Evaluationsmetriken und den Zielgrößen oft unklar. Somit definiert die Dimension *Leistung* sowohl die Zielgrößen des Problems als auch die Evaluationsmetriken.

Eine Zielgröße für die energetische Anomalieerkennung kann aus dem identifizierten Problem und den definierten Anforderungen an den Anomaliedetektor abgeleitet werden. Beispiele für produktionsspezifische Zielgrößen sind die Minimierung von Maschinenausfällen, um so Kosten für Energie und Material zu reduzieren, oder auch die Steigerung der Produktqualität.

Um eine zuverlässige Erkennung von energetischen Anomalien sicherzustellen, ist es notwendig, nicht nur das direkte Problem, sondern auch seinen Kontext vollständig zu verstehen. Die Dimension **Umgebung** definiert die Informationssysteme, Prozesse und physischen Objekte, die mit dem Anomaliedetektor interagieren. Ein tiefes Verständnis der Umgebung und deren Prozesse wird im Bereich der Datenwissenschaften auch als *Domänenexpertise* oder *Expertenwissen* bezeichnet. Gerade in Produktionsprozessen werden zahlreiche Informationssysteme verwendet, die mit hoher Komplexität einhergehen und auch bei der Planung von Anomaliedetektoren berücksichtigt werden müssen.

Die Dimension **Maßnahmen** definiert die verschiedenen grundlegenden Aktionen, die ein oder mehrere Agenten des Anomaliedetektors durchführen sollten, um die definierte Zielgröße zur Bewertung der Anomalieerkennung zu erreichen. Die Maßnahmen für Agenten in industriellen Anwendungen sind von Hofmann et al. entnommen (Hofmann et al. 2020). Maßnahmen für die energetische Anomalieerkennung in Produktionsprozessen sind dabei die Erstellung von Berichten über erkannte Anomalien (Generierung), die Regulierung von Variablen wie z. B. der Motordrehzahl (Aktion) oder die Entscheidungsunterstützung bei der Produktionsplanung (Entscheidung). Relevante Maßnahmen werden abhängig von den anderen Dimensionen wie Leistung und Umgebung identifiziert. Eine Instanzierung der Agenten, der eine Maßnahme an die Teilsysteme kommuniziert, ist im Schritt der Problemdefinition nicht zu berücksichtigen, da die Entwicklung des Systems für Agenten in der zweiten Phase implementiert werden kann. Die Leistung der Maßnahmen ist an die Zielgröße gekoppelt. Daher kann die in der

Dimension *Leistung* definierte Zielgröße mit den verschiedenen in der Maßnahmendimension definierten Aktionen kombiniert werden.

Die letzte Dimension, **Daten**, definiert sowohl die schon vorhandenen Informations- und Datenquellen, die die Umgebung erfassen, als auch notwendige Daten für die Erkennung der zuvor definierten energetischen Anomalien. Im Falle der Erkennung energetischer Anomalien in Produktionsprozessen sind Daten als multivariate Energiezeitreihen definiert. Daher ist es notwendig, Datenquellen wie Informationssysteme, Sensoren und Aktoren sowie externe Daten wie Wetter-, Kunden- oder Marktdaten zu bestimmen, falls dies zu einer Verbesserung der Leistung führen könnte. Auch in dieser Dimension ist die anwendungsabhängige Domänenexpertise essenziell, um relevante Datenquellen für die Anomalieerkennung zu identifizieren.

Auf Grundlage der **Definition des Problems** mithilfe des PEMD-Frameworks kann anschließend die Datenakquise erfolgen, um einen historischen Datensatz für die weiteren Phasen und Schritte der Methode zu generieren. Ein historischer Datensatz besteht aus mindestens einer Zeitreihe eines gemessenen Merkmals. Eine univariate Energiezeitreihe $\mathbf{ez}_j$ wird als eine Sammlung von Beobachtungen definiert, die von einem Sensor j erfasst werden:

$$\mathbf{ez}_j = (ez_{j,1}, ez_{j,2}, \cdots, ez_{j,T})^T$$

Formel 5-1 Univariate Energiezeitreihe eines Sensors

Jede Beobachtung $ez_{j,t}$ wird zu einem definierten Zeitpunkt t erfasst, wobei t von 1 bis T läuft. Die Zeitpunkte sind als äquidistant definiert, was eine zeitliche Korrelation oder Abhängigkeit zwischen den einzelnen Messungen ergibt. Im Falle von Energiedaten in Produktionsprozessen werden diese Daten mithilfe physikalischer Sensoren erfasst. Die Daten eines einzelnen Sensors werden als univariate und die Daten mehrerer Sensoren als Matrix und multivariate Energiezeitreihe $\mathbf{EZ}$ definiert:

$$\mathbf{EZ} = \begin{bmatrix} ez_{1,1} & ez_{2,1} & \cdots & ez_{S,1} \\ ez_{1,2} & ez_{2,2} & \cdots & ez_{S,2} \\ \vdots & \vdots & \vdots & \vdots \\ ez_{1,T} & ez_{2,T} & \cdots & ez_{S,T} \end{bmatrix}$$

Formel 5-2 Multivariate Energiezeitreihe

- $ez_{j,t}$ die Beobachtung des j-ten Sensor zum Zeitpunkt t ist
- S die Anzahl der Sensoren und
- T die Gesamtlänge der Energiezeitreihe in Zeitpunkten darstellt.

$\mathbf{EZ}$ beschreibt eine geordnete Menge mehrdimensionaler Vektoren, von denen jeder zu einem bestimmten Zeitpunkt aufgezeichnet wird und reellwertige Beobachtungen enthält. Im Folgenden sollen sowohl univariate als auch multivariate Energiezeitreihen als *Energiezeitreihe* definiert werden.

Zusätzlich werden **kontextbasierte Betriebsdaten** $\mathbf{B}$ erfasst, wobei jeder Betriebsdatensatz $\boldsymbol{b}_i$ innerhalb der kontextbasierten Betriebsdaten spezifische Merkmale sowie zugehörige Start- und Endzeitpunkte eines Prozessschritts umfasst. Jeder Betriebsdatensatz $\boldsymbol{b}_i$ in $\boldsymbol{B}$ ist wie folgt definiert:

$$\boldsymbol{b}_i = (\boldsymbol{m}_i, t_{\text{start},i}, t_{\text{end},i})$$

Formel 5-3 Bestandteile eines kontextbasierten Betriebsdatensatzes von Prozessschritt i

- $\boldsymbol{m}_i$ ist als Vektor von Merkmalen definiert, die durch Sensoren von Maschinen und Anlagen oder Betriebsdatenterminals erfasst werden. Dies können beispielsweise Materialeigenschaften, Einstellparameter oder weitere kontextbasierte Informationen sein.
- $t_{\text{start},i}$ und $t_{\text{end},i}$ sind die Start- und Endzeitpunkte des Betriebsdatensatzes angegeben als Zeitindex innerhalb der Gesamtlänge T der Energiezeitreihe $\mathbf{EZ}$

Da $\mathbf{B}$ aus dem Produktionsprozess in einigen Fällen keine Annotation der Anomalien aufweisen und somit eine Validierung der Phasen I und III erschwert wird, ist es methodisch sinnvoll, $\mathbf{EZ}$ und $\mathbf{B}$ zunächst in einer kontrollierten Versuchs- und Laborumgebung zu erfassen. Die zu erkennenden Anomalien wie bspw. Fehler in der Bedienung einer Anlage können somit gezielt initiiert werden. Zusätzlich sind meist die Hardwarekomponenten einer sich in Betrieb befindenden Maschine oder Anlage nicht für

eine Messung von Energiedaten mit hohen Abtastraten (z. B. 10 Hz) ausgestattet. Da mithilfe des Modellauswahlverfahrens in Abschnitt 5.3.1.5 eine Untersuchung der benötigten Abtastrate und der Auswahl der notwendigen Merkmale erfolgt, ist eine höhere Abtastrate und eine Erfassung unterschiedlicher Merkmale essenziell. Weiterhin können statische und dynamische Prozessparameter bei der gezielten Aufnahme von Daten gezielt angepasst und variiert werden, damit unterschiedliche Szenarien und Distributionen des Produktionsprozesses dargestellt werden können. Dies ermöglicht eine gezielte Heterogenität innerhalb der aufgenommenen Daten und dient als Variation der dynamischen Einstellparameter des Prozesses.

5.3.1.2 Kontextbasierte Energiezeitreihe aufbereiten

Anhand von $\mathbf{EZ}$ und $\mathbf{B}$ wird in diesem Schritt die **kontextbasierte Energiezeitreihe** $\mathbf{EZ}_{\text{kontext}}$ erstellt und aufbereitet. Mit Hilfe von $\mathbf{EZ}_{\text{kontext}}$ können verschiedene Ansätze zur Erkennung der zuvor definierten energetischen Anomalien untersucht werden. Ausgangsgröße sind entsprechend die Datensichten, die für die Ansätze zur Erkennung der zuvor definierten energetischen Anomalien erzeugt werden (siehe Abbildung 5-5).

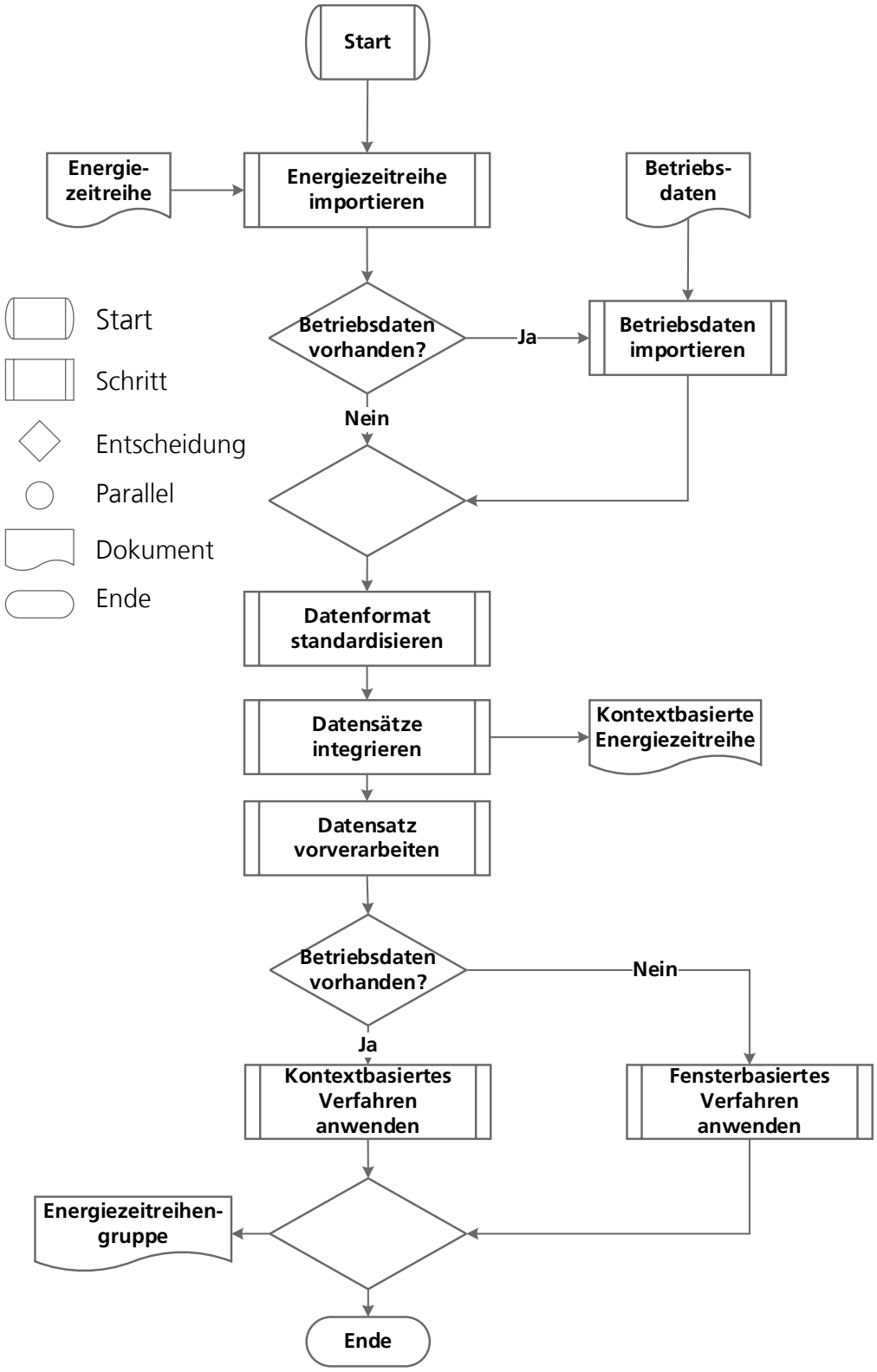


Abbildung 5-5 Ablaufdiagramm der Aufbereitung der kontextbasierten Energiezeitreihe

Es werden sowohl die Energiezeitreihe in Form der erfassten Energiedaten durch Sensoren als auch die kontextbasierten Betriebsdaten in das Zielsystem importiert und in ein standardisiertes Datenformat transformiert. Dabei können multidimensionale Arrays für die Speicherung und Verwaltung der Daten genutzt werden. Nach dem Import der Daten in das Zielsystem erfolgt die Integration der kontextbasierten Betriebsdaten $\mathbf{B}$ mit der Energiezeitreihe $\mathbf{EZ}$ zur kontextbasierten Energiezeitreihe $\mathbf{EZ}_{kontext}$ unter Berücksichtigung des Zeitpunkts t

$$\mathbf{EZ}_{\text{kontext}} = \{(\mathbf{EZ}_t, \mathbf{b}_t) | t = 1,2, \dots, T\}$$

wobei:

- $\mathbf{EZ}_t$ den Vektor der Energiezeitreihe zum Zeitpunkt t darstellt.
- $\mathbf{b}_t$ die zugehörigen kontextbasierten Betriebsdaten zum selben Zeitpunkt t aus $\mathbf{B}$ beinhaltet.
- T die Gesamtanzahl der Zeitpunkt in der Energiezeitreihe ist.

So können z. B. Produktionsaufträge oder bestimmte Produktionsprozesse und die zugehörigen Teilmengen der multivariaten Energiezeitreihe im Zusammenhang betrachtet werden (siehe Abbildung 5-6). Die Zusammenführung ermöglicht eine genaue Aufschlüsselung dessen, welcher Sensorwert zu welchem Prozessschritt gehört, aber auch, ob der Sensorwert innerhalb eines fehlerhaften Prozessschritts gemessen wurde oder nicht.

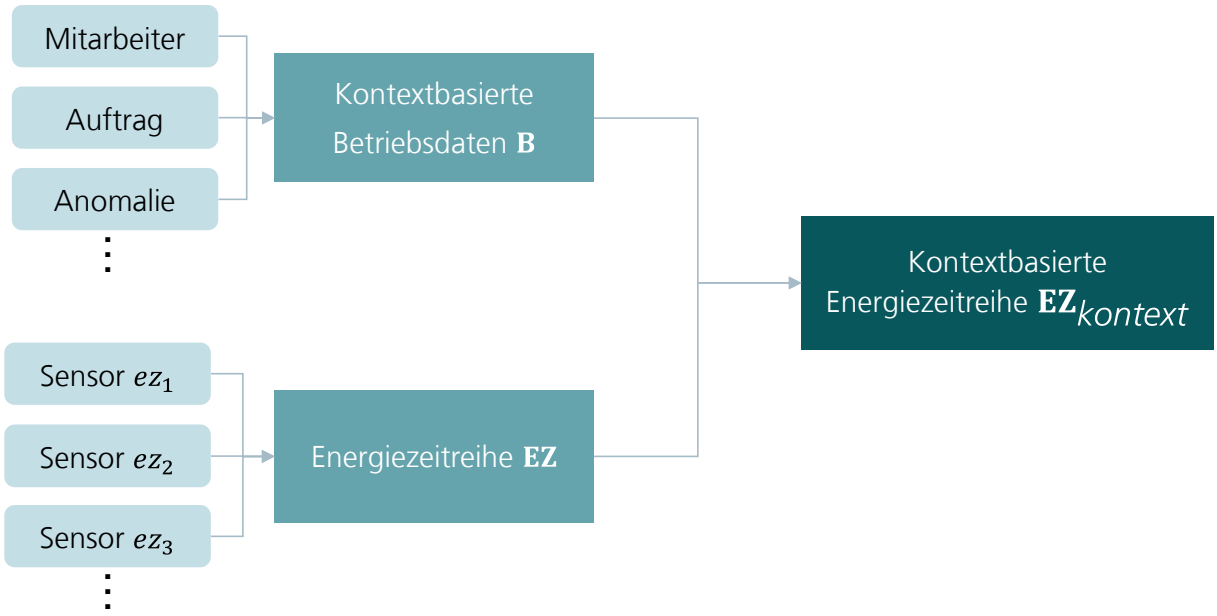


Abbildung 5-6 Schematischer Aufbau der kontextbasierten Energiezeitreihe

Ist die Integration abgeschlossen, werden die Schritte zur Datenvorverarbeitung von Zeitreihendaten durchgeführt. Zum einen sollen fehlende bzw. fehlerhafte Werte erkannt und entweder gelöscht oder durch richtige Werte ersetzt werden. In $\mathbf{EZ}_{kontext}$ können fehlerhafte Werte durch Sensor- oder Umrechnungsfehler des analogen physikalischen Signals auftreten. Dabei werden bspw. negative Werte für eine Eingangsgröße erfasst, obwohl ein negativer Wert physikalisch nicht möglich ist. Diese fehlerhaften Werte können durch Domänenexperten identifiziert und mit Interpolationsverfahren behoben werden. Des Weiteren erfolgt nach der Fehlerbereinigung die Skalierung der jeweiligen einzelnen univariaten Zeitreihen. Dabei wird die Normalisierung oder Standardisierung verwendet. Bei der Normalisierung werden die Daten aus dem ursprünglichen in einen neuen Bereich umskaliert, meist zwischen 0 und 1. Die Standardisierung erfolgt durch die Subtraktion eines bestimmten Wertes zu einem Zeitpunkt vom Mittelwert der dazugehörigen univariaten Zeitreihe und die Division des Ergebnisses durch die Standardabweichung der Zeitreihe. Dadurch werden die Daten so transformiert, dass sie einen Mittelwert von Null bzw. eine Zentrierung mit einer Standardabweichung von 1 aufweisen.

Nach der Vorverarbeitung der kontextbasierten Energiezeitreihe ist es erforderlich, die kontextbasierte Energiezeitreihe für die Merkmalsextraktion und Modellierung zu

transformieren und aufzubereiten. Dabei kann die kontextbasierte Energiezeitreihe aus unterschiedlichen Perspektiven betrachtet werden, die in dieser Arbeit als *Datensichten* definiert werden. Im ersten Schritt werden die Datensichten definiert und hergeleitet.

Die ersten beiden Datensichten werden als fenster- oder als kontextbasierte Energiezeitreihengruppe $\mathbf{EZG}$ definiert:

$$\mathbf{EZG} = \{ID_k, slice(\mathbf{EZ}, st_w, e_w) | w \in W\}$$

Formel 5-4 Berechnung der Energiezeitreihengruppen

wobei:

- ID_w die eindeutige Identifikation des w-ten Fenster innerhalb der Energiezeitreihengruppe darstellt.
- st_w und $e_w$die Start- und Endzeitpunkte des w-ten Fenster sind.
- W die Menge aller Indizes für die Fenster ist
- $slice(\mathbf{EZ}, st_w,\ e_w)$ eine Funktion ist, die die Teilmenge von $\mathbf{EZ}$ von Zeitpunkt st_w bis e_w extrahiert

Für die fensterbasierte Energiezeitreihengruppe (siehe Abbildung 5-7) ist die Funktion zur Trennung der Energiezeitreihe wie folgt definiert:

$$\mathbf{EZG}_{fenster} = \{ID_t, slice(\mathbf{EZ}, t, t + w - 1) | t = 1, w + 1, 2w + 1, \dots,\ T - w + 1\}$$

Formel 5-5 Berechnung der fensterbasierten Energiezeitreihengruppe

Die Trennung (slice) von $\mathbf{EZ}$ in unterschiedliche Zeitfenster ermöglicht eine klare Trennung der Daten, falls $\mathbf{B}$ nicht vorhanden ist. Dabei ist w die Fenstergröße, die die Anzahl der Zeitpunkte innerhalb eines Fensters definiert.

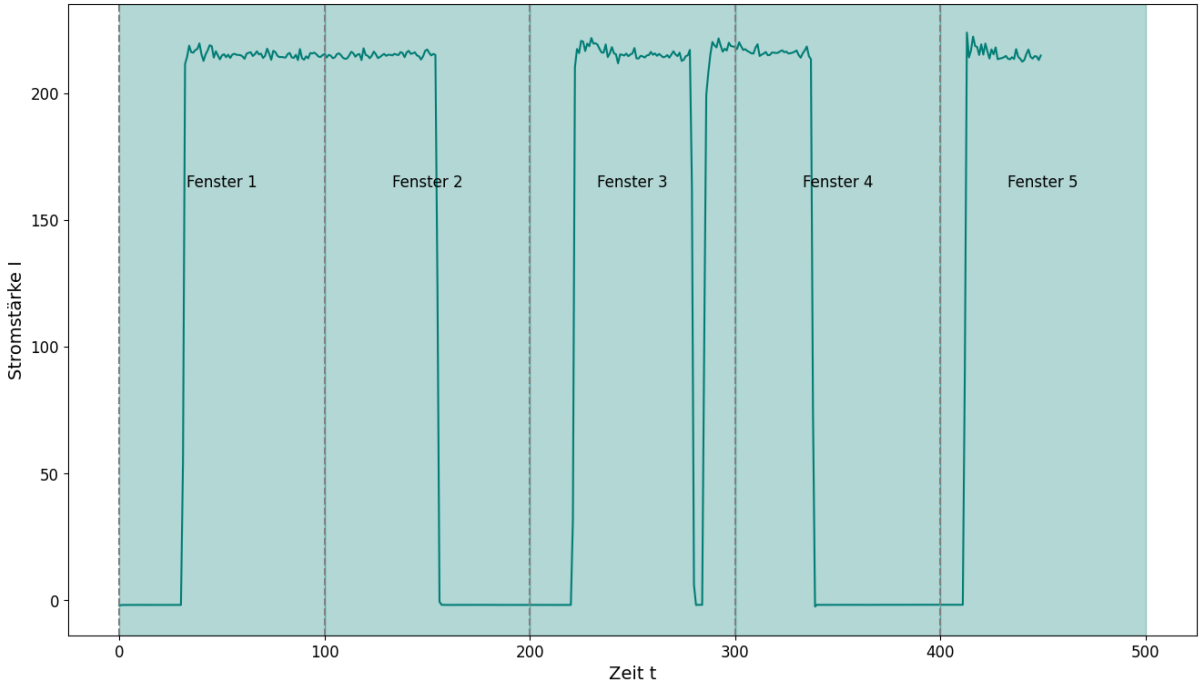


Abbildung 5-7 Darstellung der Aufteilung der kontinuierlichen Zeitreihe in identische Zeitfenster

Für die kontextbasierte Energiezeitreihengruppe $\mathbf{EZG}_{kontext}$ kann $\mathbf{EZ}$ wie folgt genutzt werden:

$$\mathbf{EZG}_{kontext} = \left\{ID_i, slice(\mathbf{EZ}, t_{\text{start},i}, t_{\text{end},i}) \middle| \forall \boldsymbol{b}_i \in \mathbf{B}\right\}$$

Formel 5-6 Berechnung der kontextbasierten Energiezeitreihengruppe

Die Integration von $\mathbf{B}$ ermöglicht somit eine Gruppierung der Energiezeitreihe in Abhängigkeit der Betriebsdaten. Werden bspw. Betriebsdaten wie Auftragsnummern oder SPS-Daten genutzt, ist eine Trennung der Zeitreihe anhand dieser Daten zu präferieren, da somit für jeden Zeitpunkt klar ist, welches Produkt gefertigt oder welche Bewegung eines Aktors durchgeführt wurde. Durch die Integration der Auftragsnummer kann die kontextbasierte Energiezeitreihe nach Aufträgen in Zeitreihen getrennt und somit separat betrachtet werden (siehe Abbildung 5-8). Dabei kann die Auftragsnummer vor allem als eindeutiger Identifikation (ID) verwendet werden, da diese in der Regel durch die Integration der kontextbasierten Betriebsdaten vorhanden ist.

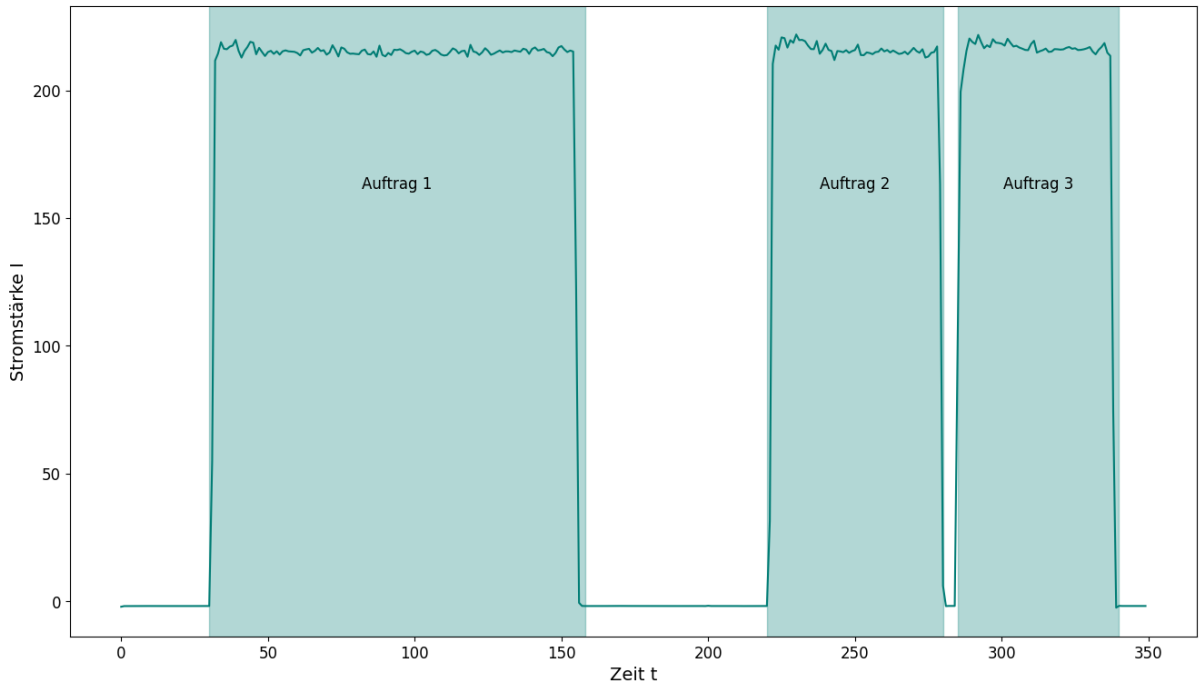


Abbildung 5-8 Darstellung der Aufteilung der kontinuierlichen Zeitreihe in kontextbasierte Zeitfenster

5.3.1.3 Merkmale extrahieren

Anhand der **EZG** kann im nächsten Schritt durch die Extraktion von sowohl statistischen als auch zeitreihenabhängigen Merkmalen eine Merkmalstabelle **MT** generiert werden (siehe Abbildung 5-9):

$$\mathbf{MT} = \bigcup_{w=1}^{W} \left\{ ID_w, \left\{ \left(f_1(\mathbf{ez}_{w,s}), f_2(\mathbf{ez}_{w,s}), \dots, f_m(\mathbf{ez}_{w,s}) \right) \middle| \boldsymbol{ez}_{w,s} \in \mathbf{EZG} \right\} \middle| s \in S \right\}$$

Formel 5-7 Berechnung der Merkmalstabelle

- ID_w ist die eindeutige Identifikation für jedes Fenster in **EZG**
- $f_1, f_2, \dots, f_m$ sind Funktionen zur Berechnung der statistischen Merkmale (z.B. Mittelwert oder die Standardabweichung) aus der univariaten Energiezeitreihe $\mathbf{ez}_{w,s}$ des s-ten Sensors innerhalb des w-ten Fensters.
- W ist die Anzahl der Fenster in **EZG.**
- S ist die Menge aller Sensoren, für die Merkmale berechnet werden.

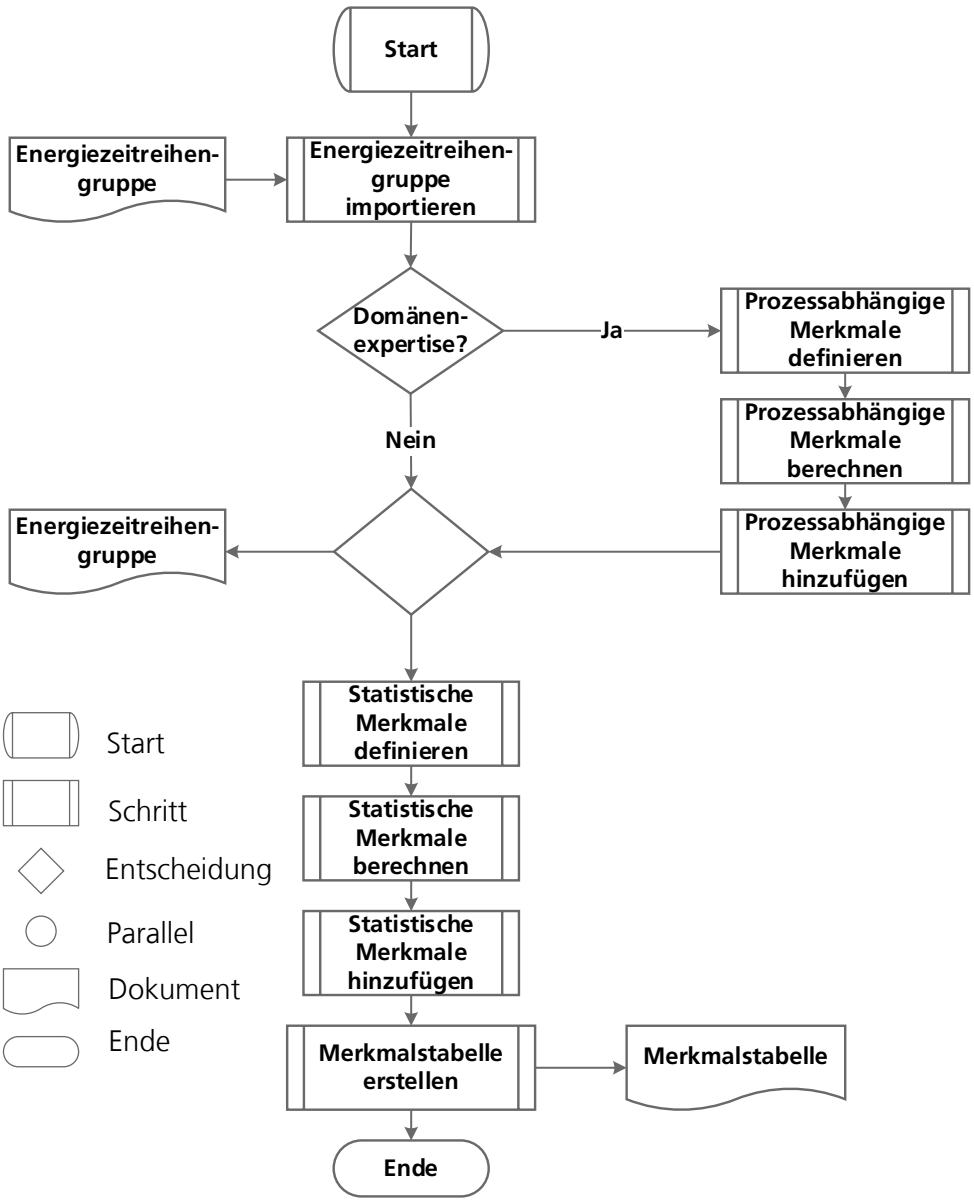


Abbildung 5-9 Ablaufdiagramm zur Extraktion von Merkmalen aus der Energiezeitreihengruppe

Dabei werden grundlegend zwei Merkmalskategorien unterschieden. Die erste beinhaltet alle prozessabhängigen Merkmale. Diese Merkmale werden anhand von ein oder mehreren Zeitreihen in $\mathbf{EZG}$ zu einem bestimmten Zeitpunkt t berechnet. Beispielsweise können prozessabhängige Merkmale wie die mittlere Kurzschlussfrequenz bei elektrischen Energiedaten berechnet werden. Auch die Effektivwerte der Spannung oder Stromstärke lassen sich in diesem Schritt berechnen.

Die zweite Kategorie beinhaltet zusätzliche statistische Merkmale (siehe Abschnitt 5.3.1.2), die für die einzelnen Sensoren der Energiezeitreihe extrahiert werden. Dabei wird

innerhalb der **EZG** eine bestimmte Anzahl an definierten Merkmalen für jede Zeitreihe $\mathbf{ez}_{w,s}$ berechnet. Die Extraktion statistischer Merkmale aus Zeitreihengruppen führt zur Eliminierung der inhärenten Zeitdimension. Dieser Schritt ermöglicht den Einsatz klassifikationsbasierter Verfahren zur energetischen Anomalieerkennung. Methoden dieser Kategorie (siehe Abschnitt 2.2.3) sind bei der direkten Anwendung auf Zeitreihen oft eingeschränkt, da sie die zeitlichen Abhängigkeiten zwischen aufeinanderfolgenden Datenpunkten nicht adäquat abbilden können. Durch die Eliminierung der zeitlichen Dimension kann diese Einschränkung umgangen werden, sodass sich klassifikationsbasierte Verfahren einsetzen lassen.

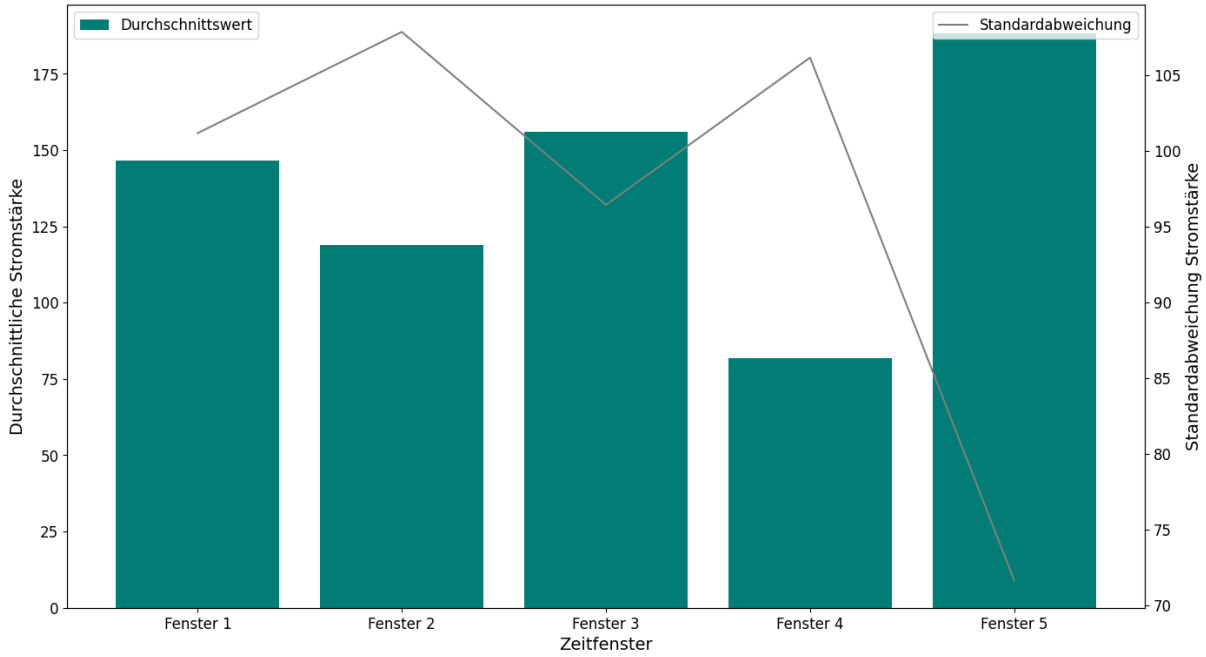


Abbildung 5-10 Exemplarische merkmalsbasierte Sicht mit identischen Zeitfenstern

Final sind anhand der kontextbasierten Energiezeitreihe, die aus der Integration der kontextbasierten Betriebsdaten mit den multivariaten Energiedaten resultieren, unterschiedliche Datensichten generiert. Einerseits kann die kontextbasierte Energiezeitreihe in kontext- oder fensterbasierte Energiezeitreihengruppen oder in Merkmalstabellen transformiert werden. Die Gruppierung der Zeitreihe kann entweder durch einen Kontext in Form kontextbasierter Betriebsdaten oder rein zeitfensterbasiert durchgeführt werden. Je nach Verfügbarkeit der kontextbasierten Betriebsdaten sollte deren Integration bevorzugt werden, um ein reales Abbild der Produktionsprozesse zu

gewährleisten. Je nachdem, welche Modellarchitektur im nächsten Abschnitt trainiert wird, erfolgt die Identifikation der für die jeweilige Modellarchitektur relevanten Datensichten.

5.3.1.4 Modelle trainieren

In diesem Abschnitt werden die in den beiden vorherigen Schritten erstellten Datensichten für das Training der unterschiedlichen Modellarchitekturen genutzt. Hierbei unterscheiden sich die Zielfunktionen und die Verfahren zur Parametrisierung der Zielfunktion in Abhängigkeit von den Modellarchitekturen. Die Ausgangsgröße des Modelltrainings ist eine definierte Menge an kreuzvalidierten Modellen, die mit den Daten der vorherigen Schritte trainiert werden.

Im ersten Schritt wird die zu betrachtende Datensicht in Trainings- und Testdaten unterteilt, damit die Leistung des Modells anhand der Evaluationsmetriken mit den für das Modell unbekannten Daten durchgeführt werden kann. Unter Einbezug der Trainingsdaten wird das Modelltraining durchgeführt. Dies wird auch in der in dieser Arbeit erstellten Vorgehensweise als Richtwert gesehen und für das Training aller Modelle festgelegt. Die eigentliche Trennung der Daten erfolgt jedoch nicht durch die zufällige Erzeugung zweier Datensätze, die jeweils dem Anteil der annotierten Daten entsprechen. Daher wird ein alternativer Ansatz entwickelt, der auf den eindeutigen ID von $\mathbf{EZG}$ oder $\mathbf{MT}$ basiert. Dies stellt sicher, dass die Aufteilung der Daten die Struktur und die Abhängigkeiten beibehält, und es verhindert, dass die gleiche ID sowohl im Trainings- als auch im Testdatensatz vorkommt.

Zuerst werden die IDs aus der zu verwenden Datensicht extrahiert:

$$I = \{ ID_w \mid w = 1,2, \dots, W \}$$

Formel 5-8 Extraktion der IDs

W repräsentiert die Anzahl der Zeitfenster. Dies erfolgt durch Zugriff auf die Datenquelle von $\mathbf{B}$, die die entsprechenden IDs enthält, und durch Anwendung einer geeigneten Funktion, um eine Liste der eindeutigen IDs zu erstellen.

Anschließend wird die Liste der eindeutigen ID in Trainings- und Testgruppen unterteilt

$$(I_{\text{training}}, I_{\text{test}}) = \mathbf{teile}(I, p, r)$$

Formel 5-9 Aufteilen der IDs in eine Trainings- und Testgruppe

- $\mathbf{teile}$ ist eine Funktion, die die IDs in Trainings- und Testdaten aufteilt. Die Funktion wählt in Abhängigkeit des prozentualen Anteils des Testdatensatzes zufällig IDs aus
- p ist der prozentuale Anteil des Testdatensatzes
- r ist eine zuvor festgelegte Zufallszahl zur Reproduzierbarkeit der Trennung

Nach der Aufteilung der ID in Trainings- und Testgruppen wird die Trennung des Datensatzes in Trainings- und Testdatensatz $\mathbf{D}_{train}$ und $\mathbf{D}_{test}$ ausgeführt.

$$\mathbf{D}_{train} = \bigcup_{ID_w \in I_{training}} \mathbf{EZG}_{ID_w}$$

$$\mathbf{D}_{test} = \bigcup_{ID_w \in I_{test}} \mathbf{EZG}_{ID_w}$$

Formel 5-10 Trennung in Trainings- und Testdatensatz

Nach der Trennung der Daten kann das eigentliche Modelltraining beginnen. Für die Entwicklung von Modellen zur energetischen Anomalieerkennung anhand den zuvor definierten Datensichten können folglich auch nur Verfahren genutzt werden, die die zeitabhängigen Charakteristika der Daten betrachten. Dennoch lassen sich auch Verfahren wie die klassifikationsbasierten Modelle nutzen, wenn die in Abschnitt 5.3.1.3 definierten fenster- oder kontextbasierten Merkmalstabellen anhand der multivariaten Energiezeitreihe erstellt worden sind. Anhand der erstellten Datensichten soll im Modellierungsschritt eine Menge an unterschiedlichen ML-Modellen zur energetischen Anomalieerkennung trainiert werden.

Für das Modelltraining werden initial zehn verschiedene Modellarchitekturen aus dem Stand der Technik genutzt (Wang et al. 2017; Zhao et al. 2017; Karim et al. 2019). Die Implementierung der Methode ist modular aufgebaut. Entsprechend können u. a. nachträglich andere Modellarchitekturen, aber auch neu entwickelte Modelle ausgewählt werden. Im Anhang (siehe Tabelle 9) sind die genutzten Modelle dargestellt. Jedes

verwendet unterschiedliche Schichten und Architekturen. Im Folgenden sollen die unterschiedlichen Modelle kurz beschrieben und die Architektur der einzelnen Modelle erläutert werden:

1. Das *ResNet*-Modell besteht aus zwei Gruppen von Convolutional-Schichten, wobei jede Gruppe vier Schichten mit 64 bzw. 128 Filtern enthält. Nach diesen Gruppen folgt eine Average-Pooling-Schicht.
2. Das *CNN*-Modell verwendet zwei Convolutional-Schichten, gefolgt von einer Average-Pooling-Schicht. Die erste Convolutional-Schicht hat sechs Filter, die zweite zwölf; CNN sind besonders effektiv bei der Vorhersage und Klassifizierung von Bilddaten. Dennoch können CNN-Modelle auch für zwei- oder eindimensionale Daten wie Zeitreihen genutzt werden.
3. Das *FCN*-Modell besteht aus drei Convolutional-Schichten mit 128, 256 und 128 Filtern, gefolgt von einer Average-Pooling-Schicht. Fully-Connected-Convolutional-Modelle können, ähnlich den CNN-Modellen, für die Klassifikation dreidimensionaler Daten, aber auch für zwei- oder eindimensionale Daten genutzt werden.
4. Das *LSTMFCN*-Modell kombiniert die Stärken von *Long Short-Term Memory* (LSTM) und *Fully Convolutional Networks* (FCN). Es beginnt mit einer LSTM-Schicht mit acht Neuronen, gefolgt von drei aufeinanderfolgenden Convolutional-Schichten mit 128, 256 und 128 Filtern. Schließlich wird eine Average-Pooling-Schicht verwendet. Diese Architektur ermöglicht es, sowohl zeitliche als auch merkmalsübergreifende Beziehungen der Daten in die Klassifikation mit einzubeziehen.
5. Das *MLP*-Modell verwendet drei Schichten mit jeweils 500 Neuronen; MLP sind standardmäßige neuronale Netze, die ohne besondere Architektur oder Komponente genutzt werden können.
6. Das *MLPKontext*$_1$-Modell verwendet eine einzige Dense-Schicht mit 100 Neuronen.
7. Das *MLPKontext*$_2$-Modell enthält fünf Dense-Schichten mit jeweils 256, 128, 64, 32 und 16 Neuronen.

8. Das *MLPKontext*$_3$-Modell hat vier Dense-Schichten mit 128, 64, 32 und 16 Neuronen.
9. Das *MLPKontext*$_4$-Modell besteht aus fünf Dense-Schichten mit 512, 256, 128, 64 und 32 Neuronen.
10. Das *MLPKontext*$_5$-Modell verwendet acht Dense-Schichten mit 128, 64, 32, 16, 16, 32, 64 und 128 Neuronen.

Nach der Definition der unterschiedlichen Modellarchitekturen kann im nächsten Schritt das Modelltraining erfolgen. Um die statistische Signifikanz des Vergleichs der verschiedenen Modelle zu gewährleisten, wird für jedes Modell und jede Datensicht eine Kreuzvalidierung durchgeführt, deren Ergebnis in der Matrix $\mathbf{KV}$ zusammengefasst wird:

$$\mathbf{KV} = \bigcup_{a \in A} \{ \bigcup_{\boldsymbol{i=1}}^{\boldsymbol{M}} \{ (M_i, a, kv_j | j = 1,2, \dots 10 \} \}$$

Formel 5-11 Definition der Kreuzvalidierung anhand der unterschiedlichen Modelle und Abtastraten

- A ist die Menge aller zu betrachtenden Abtastraten in Hz
- M_i ist definiert als das trainierte Modell aus der i-ten Modellarchitektur
- kv_j repräsentiert den j-ten Durchlauf der Kreuzvalidierung für jedes Modell M_i bei jeder Abtastrate a

Dabei wird das Training innerhalb der Kreuzvalidierung wie folgt definiert:

$$M_{i,a,kv_j} = training(\mathbf{D}_{train}, M)$$

Formel 5-12 Definition des Trainingsprozesses in jeder Kreuzvalidierungsrunde

- M_{i,a,kv_j} ist das trainierte Modell M_i für die Abtastrate a im j-ten Kreuzvalidierungsdurchlauf
- $training$ ist die Funktion, die das Modell mit den Trainingsdaten $\mathbf{D}_{train}$ unter Verwendung der zuvor definierten Modellarchitekturen M definiert

Zusätzlich wird für das Training geprüft, ob das Generieren synthetischer Daten, also eine Vervielfältigung der Energiezeitreihengruppen der vorhandenen Daten, einen Mehrwert für die energetische Anomalieerkennung darstellt (siehe Abbildung 5-11).

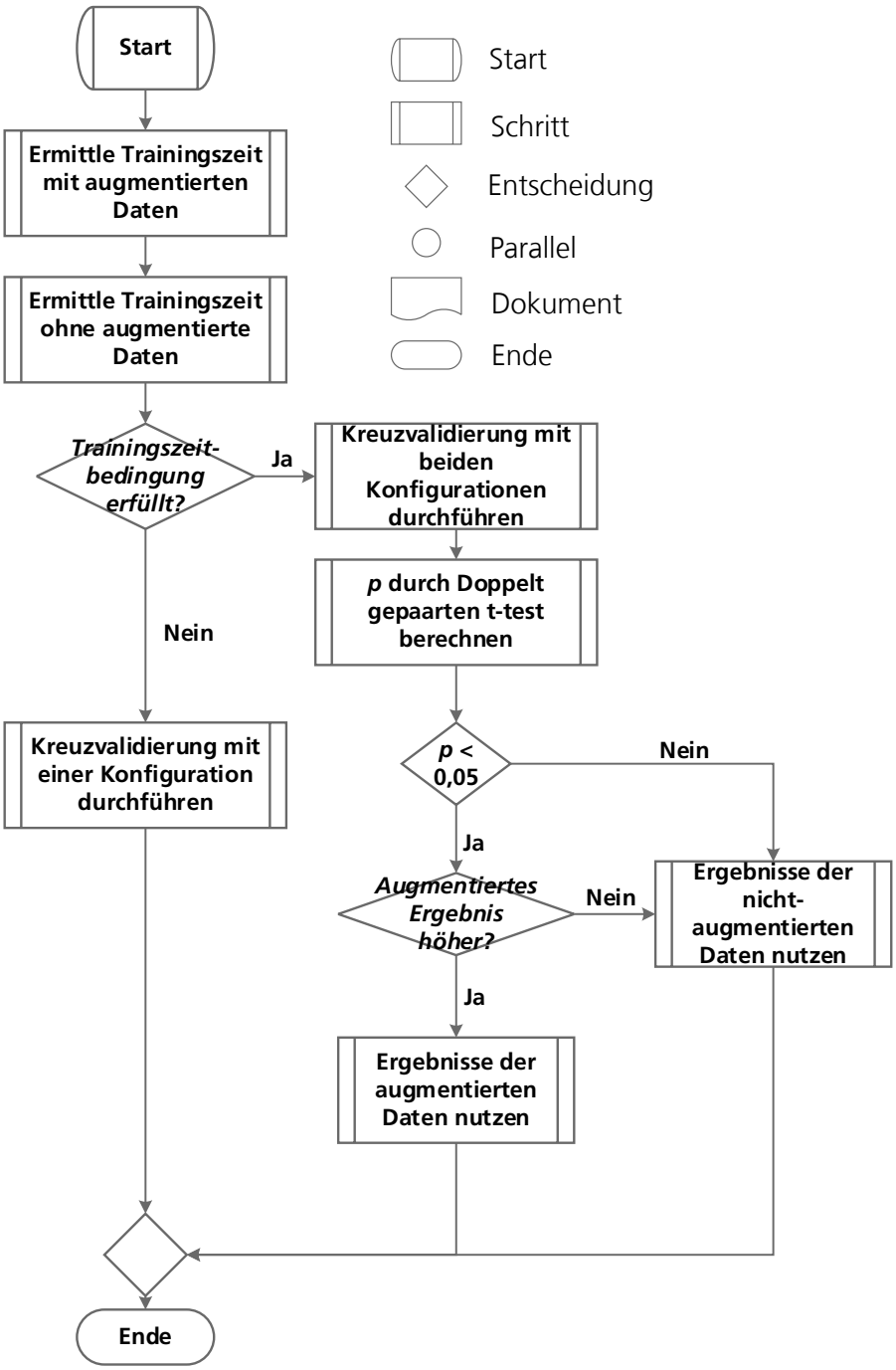


Abbildung 5-11 Ablaufdiagramm zur Verwendung der Augmentation im Trainingsprozess

Im Folgenden wird das Generieren synthetischer Daten als „Augmentation" bezeichnet. Bei großen Datensätzen kann die Augmentation weiterhin vorteilhaft sein, jedoch wird der zusätzliche Nutzen häufig durch den erhöhten Rechenaufwand relativiert. Sofern die Trainingszeit mit einem augmentierten Datensatz eine hinreichende Größe aufweist, kann

es als zweckmäßig erachtet werden, auf die Augmentation zu verzichten. Dies gilt insbesondere, wenn der ursprüngliche Datensatz bereits eine hinlängliche Größe aufweist und die definierte Evaluationsmetrik mit der ursprünglichen multivariaten Energiezeitreihe einen vorab definierten Schwellwert überschreitet.

Die Auswahl der Abtastraten erfolgt auf Grundlage der maximalen Abtastrate a_{max} und wird durch die niedrigste, maximale Abtastrate der eingesetzten Sensoren definiert. Alle zu evaluierenden Abtastraten liegen daher zwischen 1 Hz und a_{max}. Für jeden Anwendungsfall werden spezifische Abtastraten definiert, die den Anforderungen des jeweiligen Produktionsprozesses entsprechen. Die Auswahl der Abtastraten basiert somit auf Erfahrungen von Domänenexperten. Die Abtastraten werden so gewählt, dass sie eine repräsentative Abdeckung des gesamten Frequenzbereichs gewährleisten, wobei sowohl niedrige als auch hohe Frequenzen berücksichtigt werden.

Da die Kombinationsmöglichkeiten der Abtastrate der Energiezeitreihen und die Augmentation für Zeitreihen rechenintensiv sind, werden für das Training die Abtastrate variiert und der Einfluss der Augmentation evaluiert. Daher werden zwei getrennte Konfigurationen für das Modelltraining durchgeführt: eine mit und eine ohne Anwendung der Augmentationsmechanismen. Da die Anwendung Letzterer und somit die Anzahl an Daten proportional zur Trainingszeit ist, soll im ersten Schritt nur der Einfluss der Augmentationsmechanismen auf die niedrigste Abtastrate evaluiert werden. Ist der Einfluss der Augmentation auf die Evaluationsmetrik signifikant, kann die Augmentation der Daten für sinnvoll erachtet und fortgeführt werden. Die Signifikanz des Unterschieds beider Konfigurationen wird mit einem gepaarten t-Test ermittelt. Sofern der p-Wert unter $0{,}05$ liegt, also ein Unterschied als statistisch signifikant angenommen wird, und die Evaluationsmetriken der Experimente mit augmentierten Daten höher sind, kann die die Verwendung von augmentierten Daten für das Modelltraining rechtfertigt werden (siehe Abbildung 5-12).

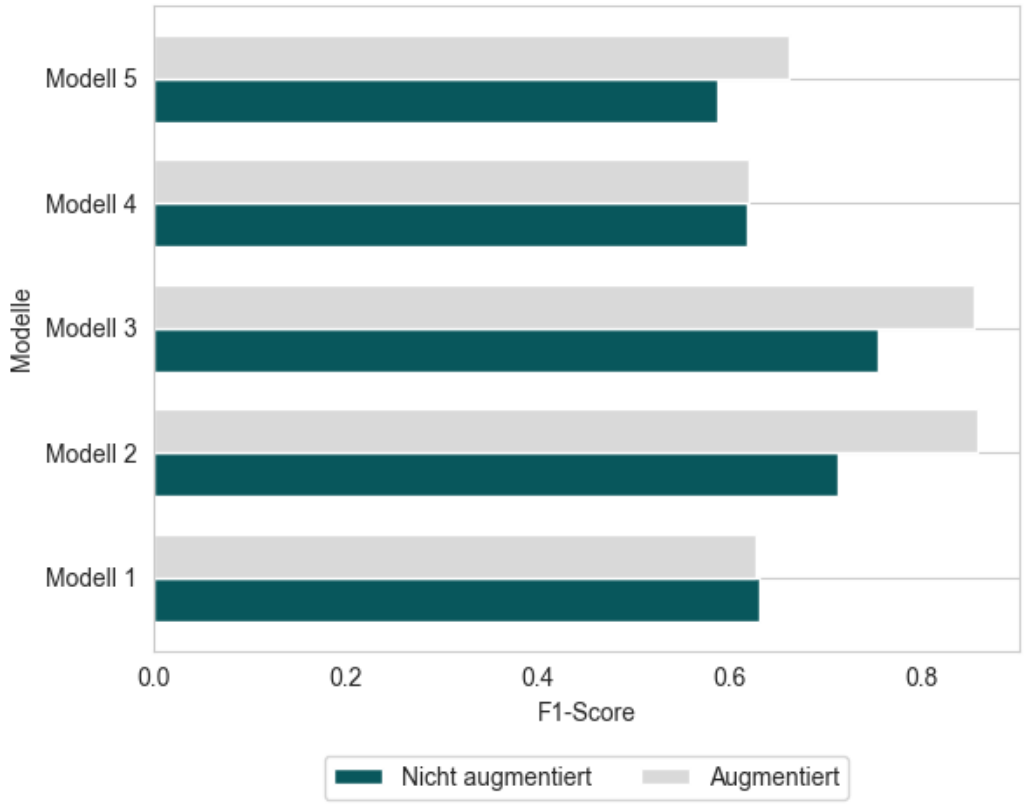


Abbildung 5-12 Exemplarischer Vergleich fünf beispielhafter Modelle auf Grundlage von augmentierten Daten

5.3.1.5 Modell auswählen

Für den Vergleich der Modelle wurde der *Leistungseffizienzindex* (LEI) entwickelt, um die Effizienz verschiedener Modelle zu bewerten. Der LEI berücksichtigt sowohl die Evaluationsmetrik (z. B. F1-Score) als auch die Trainingszeit der Modelle (siehe Formel 5-15). Beispielsweise können Maschinen und Anlagen aufgrund räumlicher Beschränkungen keine Hardware mit hoher Rechenleistung nutzen. Das ausgewählte ML-Modell sollte daher zum einen die definierte Evaluationsmetrik, aber auch die Trainingszeit miteinbeziehen. Die Trainingszeit ist ein direktes Maß für die Rechenkapazität, die zum Trainieren der Modelle benötigt wird, da sie konkret die Dauer für das Trainieren des Modells darstellt.

Um die Vergleichbarkeit und die Interpretation der Ergebnisse verständlich darzustellen, wird die Evaluationsmetrik für die Modelle und deren Trainingszeiten zunächst standardisiert. Dabei werden die Ergebnisse jeweils so skaliert, dass sie einen Mittelwert von 0 und eine Standardabweichung von 1 haben.

Anschließend wird ein gewichteter Wert berechnet, der sowohl die standardisierte Evaluationsmetrik E_S als auch den negativen standardisierten Zeitwert des Trainings t_S unter Verwendung vordefinierter Gewichtungsfaktoren α und β berücksichtigt.

$$E_S = \frac{E - \mu_E}{\sigma_E}$$

Formel 5-13 Standardisierte Evaluationsmetrik

Hierbei ist E die resultierende Evaluationsmetrik des Modells, μ_E der Mittelwert und σ_E die Standardabweichung der Evaluationsmetrik über alle Modelle. Der standardisierte Wert der Trainingszeit t_S wird analog berechnet:

$$t_S = \frac{t - \mu_t}{\sigma_t}$$

Formel 5-14 Standardisierte Trainingszeit

In dieser Formel steht t für die Trainingszeit des Modells, μ_t für den Mittelwert und σ_t für die Standardabweichung der Trainingszeit über alle Modelle. Durch die Berechnung des gewichteten Wertes für jedes Modell können die verschiedenen Modellarchitekturen direkt miteinander verglichen werden, wobei sowohl die Evaluationsmetrik als auch die Dateneffizienz (gemessen an der Trainingszeit) berücksichtigt werden. Die Modelle können dann anhand ihres gewichteten LEI bewertet und ausgewählt werden:

$$LEI = (\alpha \times E_S) - (\beta \times t_S)$$

Formel 5-15 Gewichtete Berechnung des Leistungseffizienzindexes anhand der standardisierten Evaluationsmetrik und Trainingszeit

Durch α und β können E_S und t_S je nach Anwendungsfall unterschiedlich gewichtet werden. Zusätzlich wird der LEI auf einen Wertebereich zwischen 0 und 1 skaliert, um eine vergleichbare Bewertung der Modelle zu erhalten.

$$sLEI = \frac{LEI - LEI_{min}}{LEI_{max} - LEI_{min}}$$

Modelle mit höherem $sLEI$ weisen eine bessere Kombination aus Evaluationsmetrik und Trainingszeit auf; sie gelten daher als dateneffizienter. Die Identifikation des Modells M_{max} erfolgt demnach:

$$M_{max} = arg \max_{(a,m) \in A \times M} sLEI_{a,m}$$

Formel 5-16 Identifikation des Modells mit dem höchsten $sLEI$

- M_{max} ist das Modell aus der Menge an trainierten Modellen M, das zusammen mit der Abtastrate a aus A den höchsten $sLEI$ aufweist
- $sLEI_{a,m}$ ist der Leistungseffizienzindex für das Modell m bei der Abtastrate a

Die in Abbildung 5-13 dargestellte Visualisierung stellt in den Zeilen die verschiedenen Modelle und in den Spalten die verschiedenen Abtastraten dar. Sie ist mit einer Farbskala hinterlegt, die von hellgrau (entspricht einem niedrigeren $sLEI$) bis dunkelgrün (entspricht einem höheren $sLEI$) reicht.

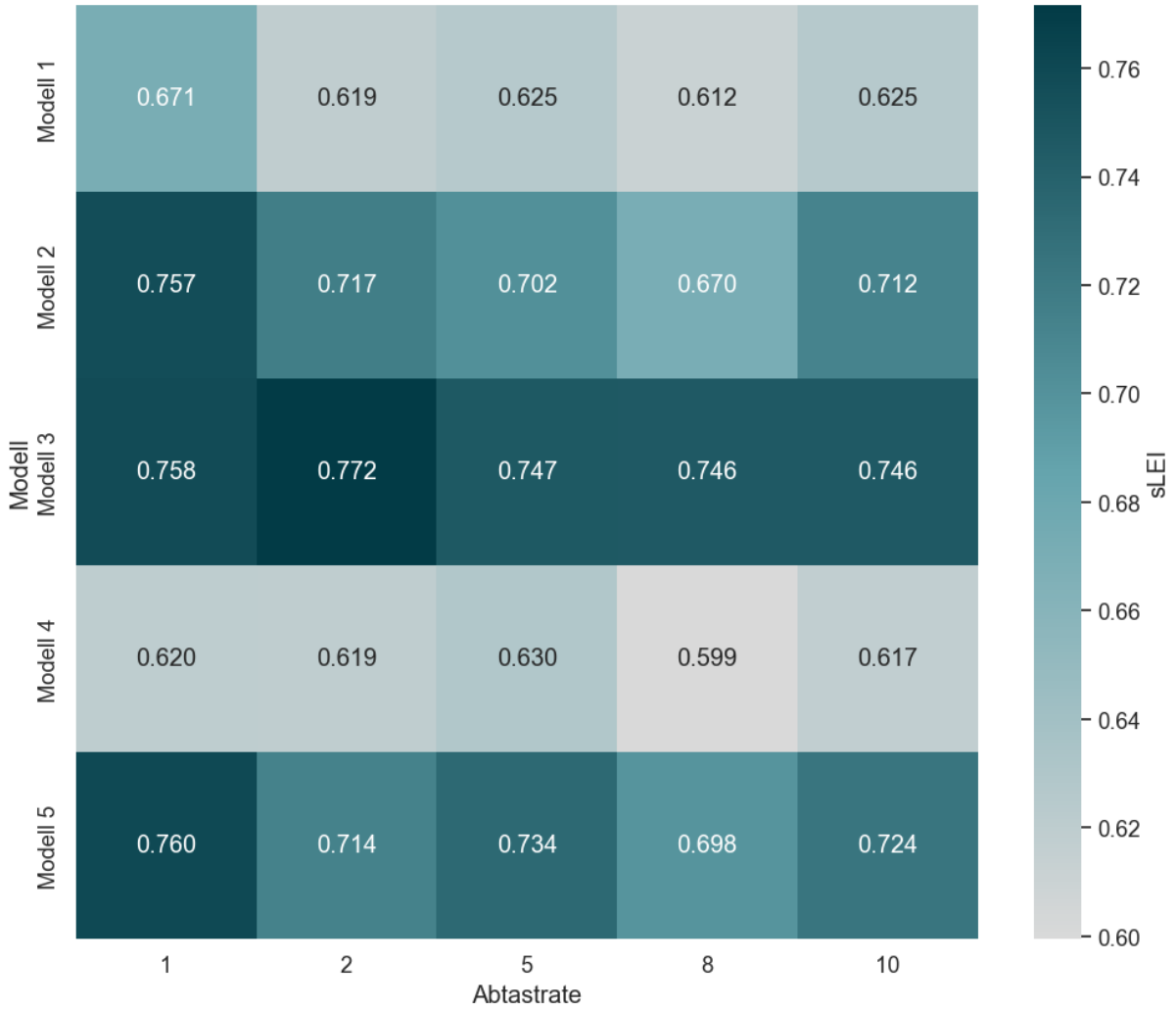


Abbildung 5-13 Exemplarische Heatmap des $sLEI$ der unterschiedlichen Modell-Abtastrate-Kombinationen

Nach Abschluss der Modellauswahl in diesem Abschnitt wird das ausgewählte Modell, die zugehörige Datensicht, die geeignete Abtastrate und mögliche synthetische Daten in die Phase II der Methode überführt .

5.3.2 Implementierung des Anomaliedetektors

In diesem Abschnitt wird die zweite Phase zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen beschrieben und in Abbildung 5-14 dargestellt. In dieser Phase der Methode werden die funktionalen

Komponenten des Anomaliedetektors aus der Architektur des FL-basierten Systems instanziiert und konfiguriert. Nachdem in der ersten Phase ein **Modell** der zugehörigen **Datensicht** und der geeigneten **Abtastrate** zur Erkennung energetischer Anomalien identifiziert worden ist, erfolgt die **Implementierung des Anomaliedetektors** für die anwendungsspezifische Zielarchitektur des FL-basierten Systems. Die Implementierung des Anomaliedetektors erfordert den **Aufbau der Datenpipeline** von der Datenquelle bis zur Inferenz, die **Entwicklung von Agenten** für das Einleiten einer Maßnahme bei einer erkannten Anomalie und die **Bereitstellung des Anomaliedetektors**. Diese einzelnen Teilschritte werden in den nächsten Abschnitten vorgestellt.

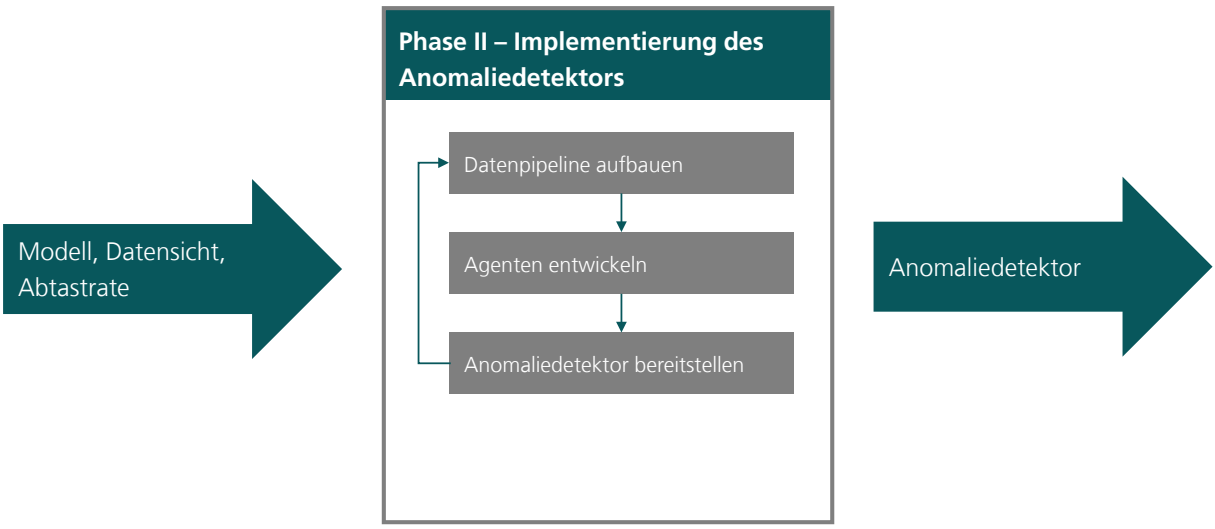


Abbildung 5-14 Phase II der Methode für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen

5.3.2.1 Datenpipeline aufbauen

In diesem Teilschritt wird die Datenpipeline bestehend aus funktionalen Komponenten und Schnittstellen für die Implementierung des Anomaliedetektors entwickelt (siehe Abbildung 5-15).

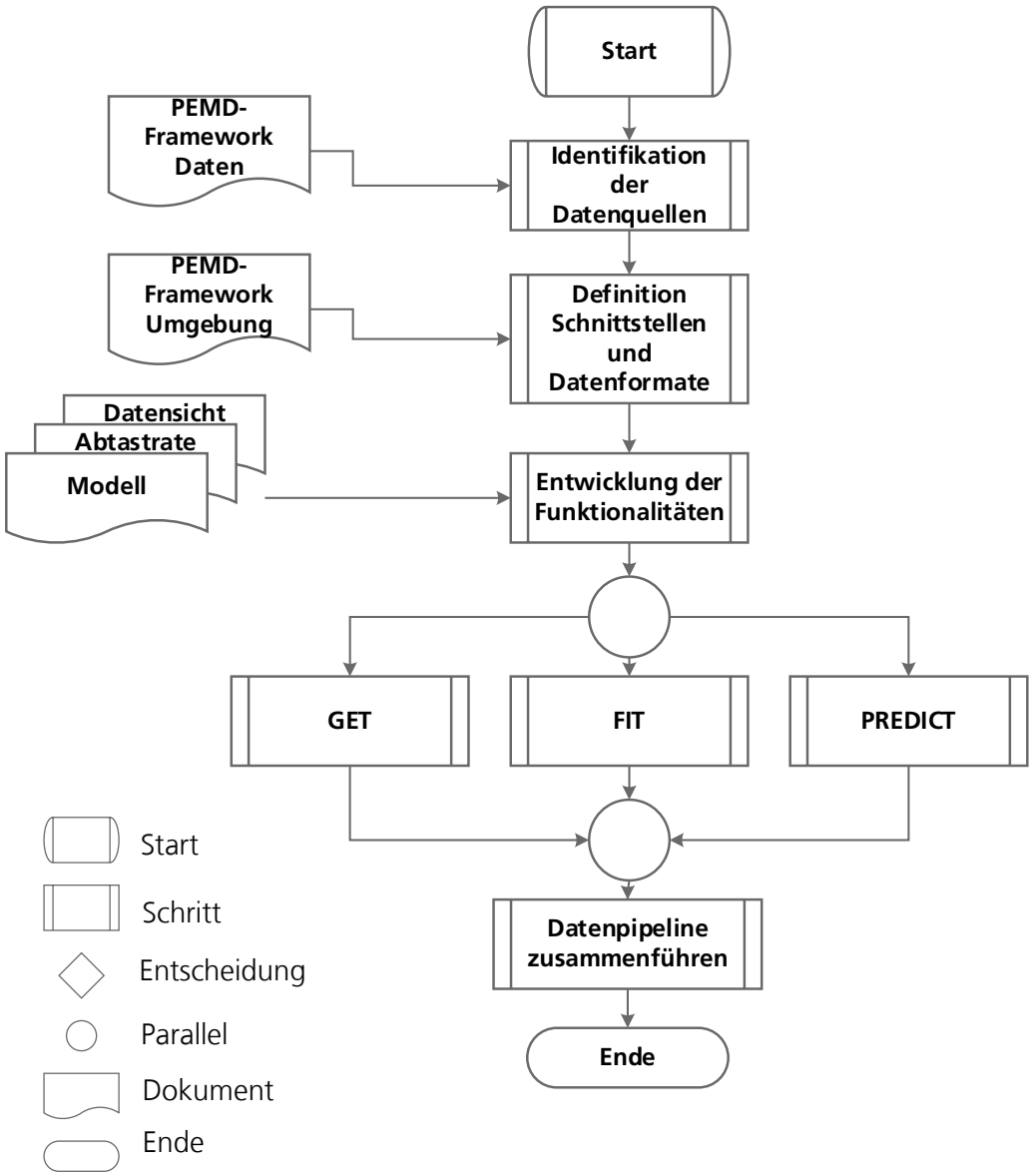


Abbildung 5-15 Ablaufdiagramm zur Entwicklung der Datenpipeline des Anomaliedetektors

Für die erfolgreiche Anbindung von Maschinen und Anlagen an den Anomaliedetektor gilt es, die notwendigen Datenquellen zu identifizieren, die für die Erkennung von Anomalien erforderlich sind. In diesem Abschnitt wird, anders als in Phase I, der Begriff *Daten* bzw. *Datenquelle* verwendet, da die Daten aus den unterschiedlichen Datenquellen nicht nur Energiezeitreihen, sondern auch kontextbasierte Betriebsdaten der Produktionsprozesse oder andere relevante Zeitreihen beinhalten können. Sind die einzelnen Datenquellen identifiziert, sollte der Zugriff der Daten mithilfe von

Kommunikationsschnittstellen betrachtet werden. Dabei sind für die weitere Implementierung des Anomaliedetektors zwei Aspekte von Bedeutung. Zum einen erfordert der Import der Daten aus der Datenquelle die Identifikation eines geeigneten Kommunikationsprotokolls; zum anderen ist es notwendig, das Datenformat zur Übertragung der Daten festzulegen. Beispielsweise werden in modernen Anwendungen häufig Daten im JSON-Format übertragen, das eine sowohl software- als auch hardware-agnostische Bereitstellung von Daten ermöglicht.

Nach der Identifikation der Datenquellen und deren Definition der Schnittstellenspezifikationen erfolgt der Aufbau der DVEs für die Datenpipeline. Letztere umfasst drei Hauptfunktionen: GET, FIT und PREDICT. Die GET-Funktionalität widmet sich dem Import und der Transformation der Daten. Auf Grundlage der vorherigen Phase werden die Daten vorverarbeitet und es werden zusätzliche Merkmale generiert. Es folgt die FIT-Funktionalität, in der das ML-Modell mit den vorverarbeiteten Daten trainiert, evaluiert und serialisiert wird. Abschließend bildet die PREDICT-Funktionalität die eigentliche Vorhersage einer Anomalie, indem sie die neu erfassten Daten und das trainierte ML-Modell nutzt, um Anomalien zu identifizieren und zu analysieren. Im Folgenden sollen die drei Funktionalitäten beschrieben und erläutert werden.

Die GET-Funktionalität besteht aus zwei grundsätzlichen DVEs, dem eigentlichen Import und der Datentransformation. Das Importmodul ist so zu implementieren, dass ein kontinuierliches Abfragen neuer Daten aus der Datenquelle ermöglicht wird. Die Daten werden mit dem zuvor definierten Datenformat übertragen und können in das für den Anomaliedetektor festgelegte Datenformat transformiert werden. Das Format der Daten zur Weiterverarbeitung im Anomaliedetektor ist entweder als tabellarische oder tabellenähnliche Form (siehe Formel 5-2) definiert. Es beinhaltet die einzelnen Energiezeitreihen oder anderen Datenquellen in einer Spalte und jede Ausprägung zu einem gewissen Zeitpunkt als Zeile. Die einzelnen Vorverarbeitungsschritte aus den Abschnitten 5.3.1.2 und 5.3.1.3 können von der Programmlogik als DVE übernommen werden. Dies beinhaltet die Aufbereitung und die Integration der Energiezeitreihen und kontextbasierten Betriebsdaten in der ausgewählten Abtastrate. Zusätzlich wird anhand der Modellauswahl in Abschnitt 5.3.1.5 der Algorithmus für die ausgewählte Datensicht

in die GET-Funktionalität überführt. Anhand der ausgewählten Datensicht **MT** oder **EZG** unterscheidet sich die Datenverarbeitungslogik zur Extraktion der Merkmale, da für **EZG** keine Extraktion von statistischen Merkmalen notwendig ist.

Nach der GET-Funktionalität erfolgt die Implementierung der FIT-Funktionalität. Dabei werden die vorverarbeiteten Daten für das Modelltraining des aus der ersten Phase ausgewählten Modells genutzt. Die Schritte des Trainings, die Bewertung des Modells anhand der Evaluationsmetrik und die Serialisierung des Modells erfolgen mit der Implementierung der FIT-Funktionalität. Zu diesem Zweck werden die Artefakte aus der Phase I übernommen. Die Implementierung des Modelltrainings ist vom eigentlichen ausgewählten Modell abhängig und erfolgt dahingehend in Abhängigkeit desselben. Dennoch ist die Implementierung des Trainingsmoduls so zu gestalten, dass eine abstrahierte Schnittstelle unabhängig des Modells realisiert werden kann. Nach dem Training des Modells wird dessen Leistung mit den in Abschnitt 5.3.1.1 definierten und in der ersten Phase bereits genutzten Evaluationsmetriken bewertet sowie mit vorherigen Trainingsrunden verglichen. Ist bspw. eine signifikante Abweichung der Evaluationsmetriken im Vergleich zu vorherigen Trainingsrunden oder den Ergebnissen aus der ersten Phase ersichtlich, kann eine Untersuchung der signifikanten Abweichung durch eine wiederholte Durchführung der Evaluation aus Phase I erfolgen.

Ist die Implementierung der FIT-Funktionalität abgeschlossen, kann das Serialisierungsmodul entwickelt werden. Die Serialisierung des Modells in ein geeignetes Datenformat ist notwendig, um die Portabilität und Interoperabilität des zuvor trainierten und evaluierten Modells zu gewährleisten. Hierbei eignen sich standardisierte Formate wie Pickle-Dateien, ONNX oder HDF5. Die Versionierung dieser Datenformate steht abhängig der Programmiersprache nicht im Vordergrund, da davon ausgegangen wird, dass die FIT-Funktionalitäten in der gleichen Umgebung und Programmiersprache wie die restlichen Funktionalitäten entwickelt wird. Das jeweils aktuelle serialisierte Modell wird lokal gesichert. Es kann dann von anderen Funktionalitäten und Modulen geladen bzw. genutzt werden.

Die letzte Funktionalität der Datenpipeline ist als PREDICT-Funktionalität definiert und beinhaltet die eigentliche Erkennung einer Anomalie. Dabei dienen die Daten aus der GET-Funktionalität und das Modell aus der FIT-Funktionalität als Input für die PREDICT-Funktionalität. Bei der Implementierung der PREDICT-Funktionalität ist darauf zu achten, dass die eingehenden Daten aus der Datenquelle die gleichen Transformations- und Vorverarbeitungsschritte wie in der GET-Funktionalität durchlaufen. Zusätzlich wird das serialisierte und lokal gesicherte Modell geladen. Die eingehenden Daten dienen somit als Input für das geladene Modell. Bei der Implementierung muss beachtet werden, dass innerhalb der PREDICT-Funktionalität meist nur eine Energiezeitreihe und keine Energiezeitreihengruppe oder Merkmalstabelle verarbeitet wird. Mithin ist keine Stapelverarbeitung wie in Phase I notwendig. Das Ergebnis des Modells kann in einer lokalen Datenbank für Evaluationszwecke gespeichert und/oder für die weitere Verarbeitung genutzt werden.

Im nächsten Schritt erfolgt die Implementierung des Agenten, der die unterschiedlichen Funktionalitäten aus diesem Schritt nutzt, um mithilfe von Agenten Gegenmaßnahmen im Zuge der Erkennung einer energetischen Anomalie einzuleiten.

5.3.2.2 Agenten entwickeln

In diesem Abschnitt werden die einzelnen Teilschritte beschrieben, die für die Implementierung des Agenten (siehe Abschnitt 5.2.1) notwendig sind (siehe Abbildung 5-16).

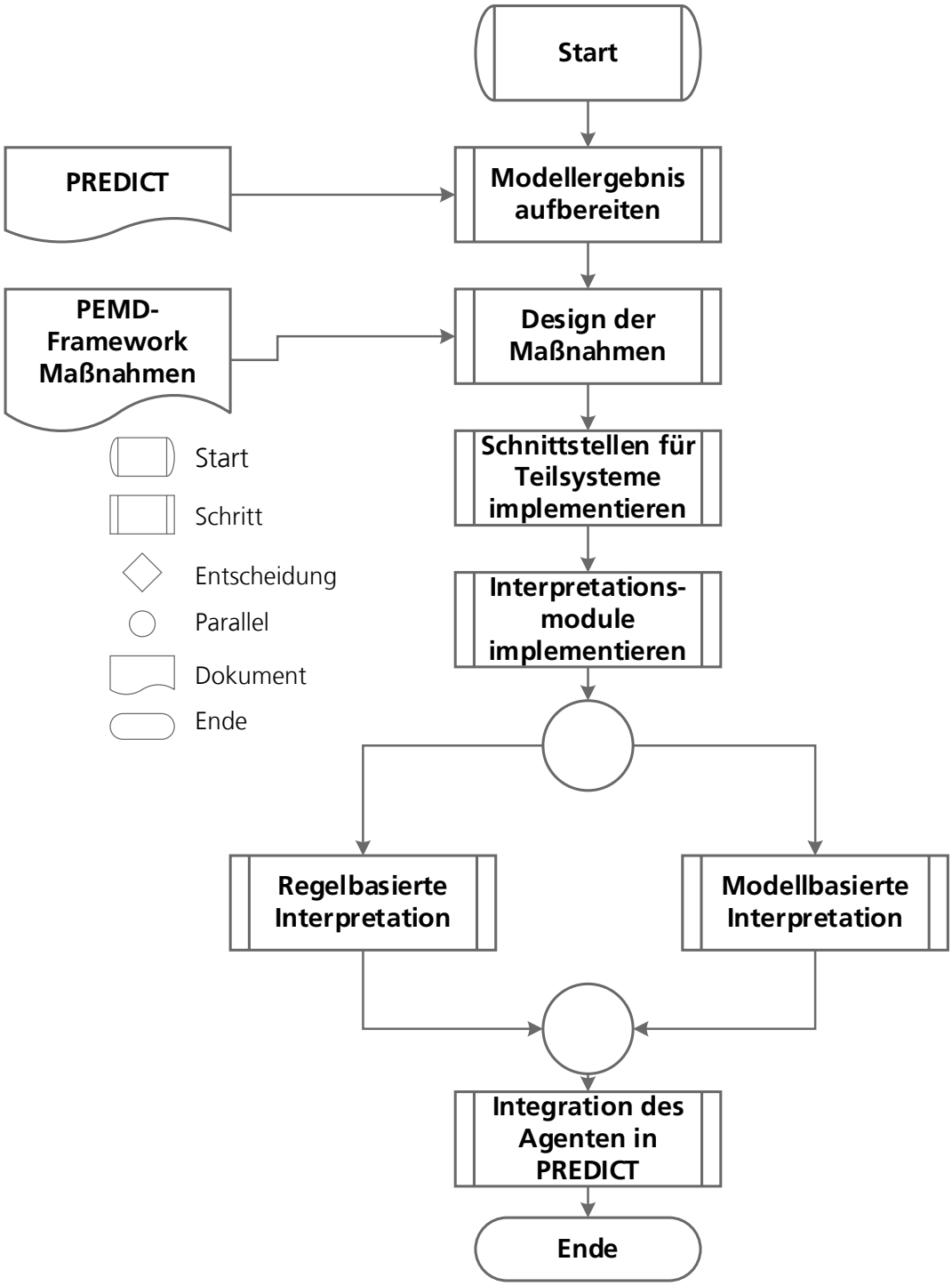


Abbildung 5-16 Ablaufdiagramm zur Entwicklung von Agenten für den Anomaliedetektor

Der Agent oder die Agenten beziehen die Modellergebnisse aus 5.3.2.1 ein, verarbeiten diese und schaffen die Schnittstelle zu den zuvor definierten Maßnahmen und deren zugehörigen Teilsystemen. Anhand des PEMD-Frameworks aus Abschnitt 5.3.1.1 erfolgt die Ableitung der Maßnahmen, die durch die Problemanalyse und die Spezifikation der

Anwendung gegeben sind. Während in Abschnitt 5.3.2.1 die Schnittstellen hin zu Datenquellen implementiert wurden, ist in diesem Abschnitt die Implementierung von Schnittstellen hin zu den nachgelagerten Systemen erforderlich, die mithilfe des Agenten notwendige Maßnahmen durchführen können. Dabei können jedoch die gleichen Kommunikationsschnittstellen und Datenformate genutzt werden. Ein weiterer Punkt ist die Integration der PREDICT-Funktionalität: Durch sie und ihre Resultate können die Modellergebnisse direkt abgerufen werden. Zusätzlich wird ein Modul zur Interpretation der Ergebnisse implementiert. Die Interpretation erfolgt entweder regel- oder modellbasiert. Eine rein regelbasierte Implementierung ist ein Agent, der z.B. bei einer bestimmten Anzahl energetischer Anomalien innerhalb eines Zeitfensters einen Alarm ausgibt oder bei einer höheren Anzahl an energetischen Anomalien die Maschine in den Standby-Modus schaltet. Diese Regeln können innerhalb des Agenten festgelegt werden. Eine andere Möglichkeit besteht darin, dass der Agent modellbasiert implementiert wird. In diesem Fall können datenbasierte Modelle auf Grundlage der Erkennung energetischer Anomalien Maßnahmen ergreifen. Eine beispielhafte Implementierung eines Anomaliedetektors mit einem zusätzlichen modellbasierten Agenten wurde in den Vorarbeiten des Autors dargestellt (Bauer et al. 2021).

5.3.2.3 Anomaliedetektor bereitstellen

Der letzte Schritt in Phase II beschreibt die Vorgehensweise zur Bereitstellung des entwickelten Anomaliedetektors (siehe Abbildung 5-17).

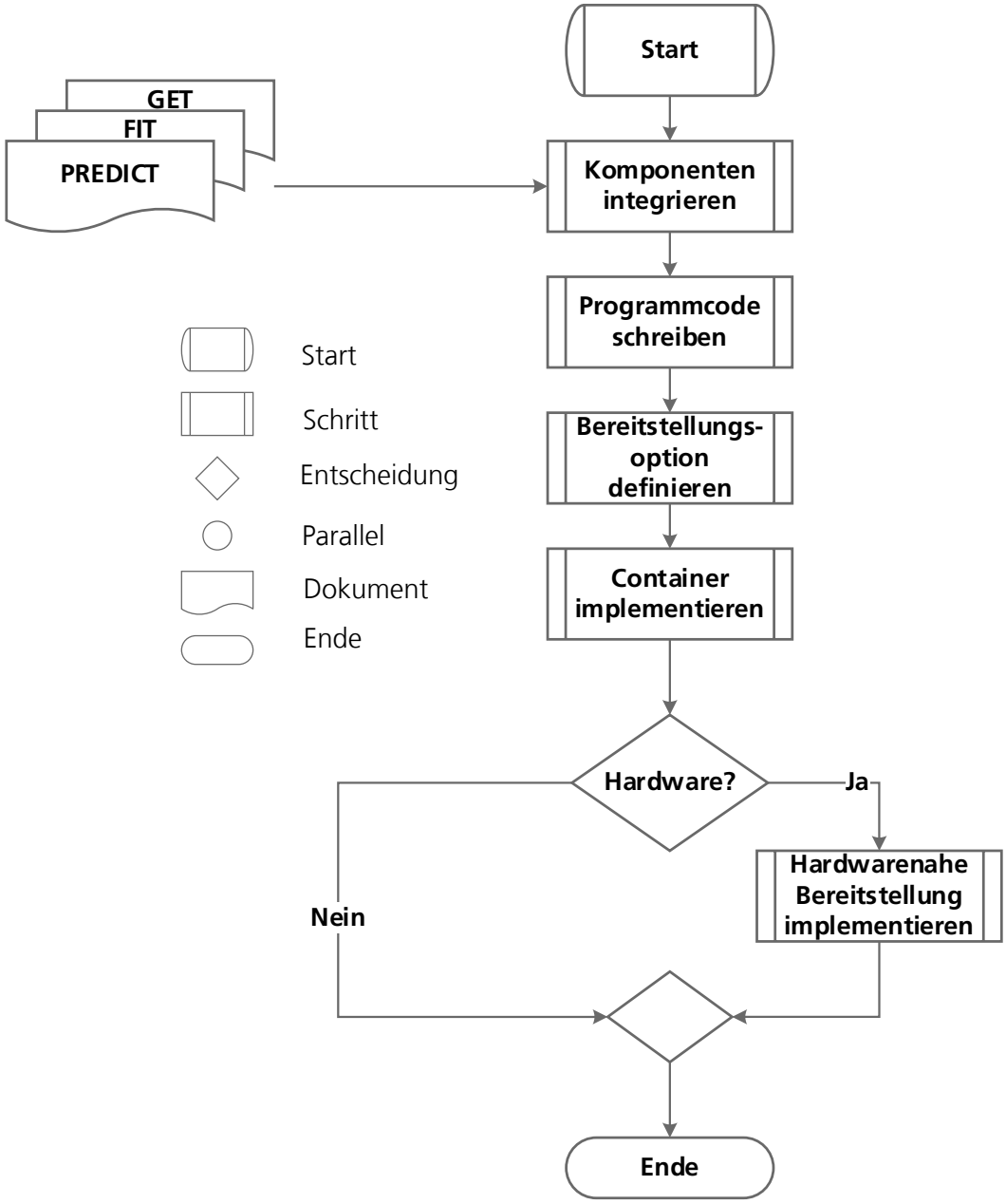


Abbildung 5-17 Ablaufdiagramm zur Bereitstellung des Anomaliedetektors in einer lauffähigen Umgebung

Der Prozess zur Bereitstellung des Anomaliedetektors beginnt mit der Integration der funktionalen Komponenten, zu denen die GET-, FIT- und PREDICT-Funktionalitäten

innerhalb der Datenpipeline gehören. Nach der Integration der Komponenten erfolgt das Schreiben des Programmcodes. Dies umfasst die Überführung des in Skripten entwickelten Codes in eine strukturierte Programmcodestruktur. Die Programmierung des Anomaliedetektors erfolgt in einer Entwicklungsumgebung, die an die spezifischen Anforderungen der Zielhardware angepasst ist. Der Code wird nach den Grundsätzen des Software-Engineering entwickelt, wobei der Schwerpunkt auf Modularität, Wiederverwendbarkeit und Wartbarkeit liegt. Nach der Implementierung des Programmcodes wird die Bereitstellungsoption definiert. Für die Bereitstellung des Anomaliedetektors werden zwei Hauptstrategien betrachtet: Containerbasierte Bereitstellung und hardwarenahe Bereitstellung. Die containerbasierte Bereitstellung bietet Vorteile hinsichtlich Portabilität und einfacher Skalierbarkeit und wird durch den Einsatz von Technologien wie Docker oder Kubernetes realisiert. Die Containerisierung ermöglicht eine hardwareagnostische Bereitstellung des Systems, was insbesondere für Umgebungen mit höherer Rechenleistung und der Möglichkeit, höhere Programmiersprachen zu verwenden, von Vorteil ist.

Zusätzlich kann eine hardwarenahe Bereitstellung untersucht werden, die eine direkte Implementierung auf SPS oder Industrie-PCs beinhaltet. Moderne speicherprogrammierbare Steuerungen oder auch Einplatinencomputer, wie sie im Maschinen- und Anlagenbau eingesetzt werden, können befähigt werden, den Anomaliedetektor maschinennah bereitzustellen und die DVE von der Datenquelle bis zum Abruf der Maßnahme zu gewährleisten. Dadurch kann die Latenzzeit verkürzt werden, was für kritische Produktionsprozesse entscheidend sein kann. Die Auswahl der geeigneten Bereitstellung basiert auf den spezifischen Rahmenbedingungen der Produktionsumgebung.

5.3.3 Parameteroptimierung des föderierten Lernens

Nachdem in der ersten Phase ein Modell, die zugehörige Datensicht und die geeignete Abtastrate identifiziert worden sind sowie in der zweiten der Anomaliedetektor für die energetische Anomalieerkennung entwickelt und bereitgestellt worden ist, erfolgt in der

dritten Phase die für die Auslegung des FL-basierten Systems notwendige Parameteroptimierung, um die Konfiguration für die in 5.2.2 definierte FLEA-Achitektur zu erhalten (siehe Abbildung 5-18). Im ersten Schritt werden dazu die möglichen Konfigurationsparameter definiert und initialisiert. Die Definition der Parameter erfolgt anhand der vorangegangenen wissenschaftlichen Literaturanalyse und den Vorveröffentlichungen des Autors (Karnebogen et al. 2023). Die Simulation von FL und der entsprechenden Auswahl der Parameter soll anschließend mit Hilfe der bayesschen Hyperparameteroptimierung durchgeführt werden (siehe Abschnitt 2.2.4). Kern der Simulation ist die Aggregation der einzelnen Modellgewichte und die Kommunikation zwischen Server und Clients, da so die Anzahl der Clients variiert werden kann, ohne dass die Anwendungen auf unterschiedlicher Hardware implementiert werden müssen. Im letzten Schritt sollen die Simulationsergebnisse umfassend evaluiert werden, um eine optimale Konfiguration des FL-basierten Systems zur energetischen Anomalieerkennung zu identifizieren.

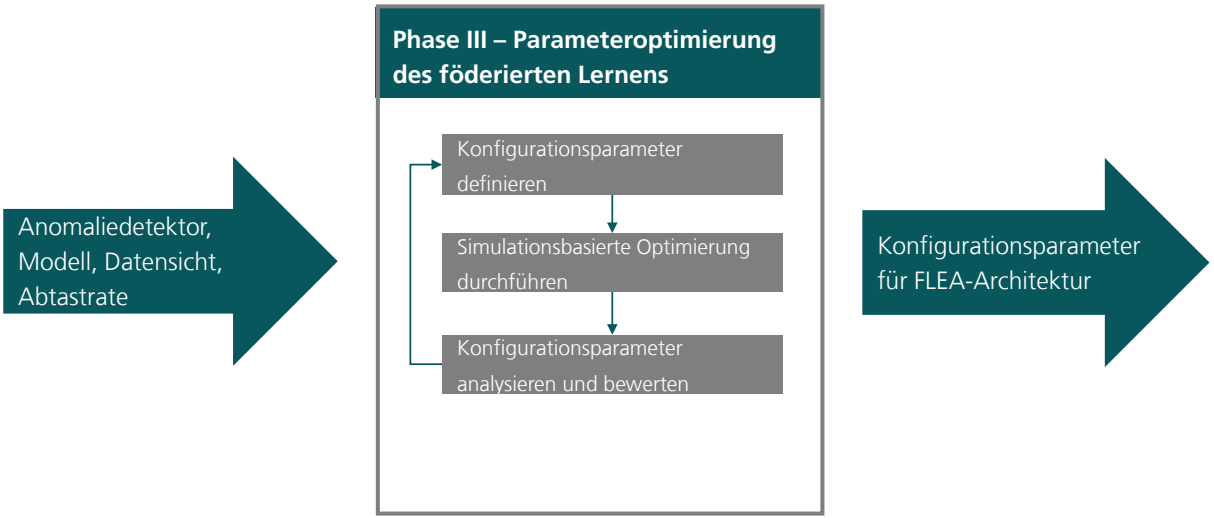


Abbildung 5-18 Phase III der Methode für die Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen

5.3.3.1 Konfigurationsparameter definieren

Für die erfolgreiche Auslegung des FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen sollen in diesem Methodenschritt die

Parameterräume von FL definiert werden, die Einfluss auf die in Abschnitt 5.3.1.1 definierte Evaluationsmetrik haben und mit in die Simulation von FL in Abschnitt 5.3.3.2 einfließen. Dieser Schritt kann nach der Evaluation der Simulationsergebnisse erneut durchlaufen werden, falls eine Anpassung der Konfigurationsparameter notwendig ist. Im Folgenden sollen die zu definierenden Konfigurationsparameter für FL vorgestellt und beschrieben werden. Die Definition des Parameterraums hängt jeweils vom Anwendungsfall ab. Sie sollte mithilfe von Domänenexperten und der explorativen Analyse der Daten durchgeführt werden. Dabei werden die Parameter in **Systemparameter** θ_s und **Client-Parameter** θ_c aufgeteilt. Durch die Entwicklung des Modells zur energetischen Anomalieerkennung in Abschnitt 5.3.1 konnten bereits zahlreiche Parameter wie die Modellarchitektur oder die Anzahl an Merkmalen optimiert werden, was den Optimierungsprozess in dieser Phase vereinfacht.

Systemparameter θ_s

Aggregationsmechanismus: Der Aggregationsmechanismus definiert die Strategie , wie die eingehenden Modellparameter der einzelnen Clients zu einem globalen Modell zusammengefasst werden (siehe Abschnitt 2.2.1). In der wissenschaftlichen Literatur und in der Praxis wurde der initiale Mechanismus, FedAvg, umfassend erweitert. Beispielsweise arbeiten Aggregationsmechanismen wie FedOpt mit mehreren Parametern wie der Datenmenge oder den Evaluationsmetriken der lokalen Modelle. Da sich die individuellen Parameter der einzelnen Aggregationsmechanismen stark voneinander unterscheiden können, werden sie in Kapitel 6 spezifisch je nach Anwendungsfall betrachtet.

Trainingsrunden: Wie bei der klassischen Approximation von Modellparametern mithilfe der Zielfunktion eines Modells erfolgt der Prozess von FL in Trainingsrunden, wobei eine Runde dem in Abschnitt 2.3.1 vorgestellten Verfahren entspricht. Eine Trainingsrunde entspricht somit der Übertragung des Modells vom Server an eine Auswahl von Clients, dem Training auf den unterschiedlichen Clients, dem Empfangen der Modellparameter durch den Server und der Aggregation dieser Parameter. Wie auch bei ML kann eine zu hohe Parametrisierung der Trainingsrunden zu einer Überanpassung führen.

Client-Anzahl: Da durch die Generierung eines Datensatzes (siehe Abschnitt 5.3.1.1) eine feste Menge an Daten erfasst wurde, muss ein geeigneter Parameter für die Aufteilung

der Daten in eine bestimmte Anzahl Clients gefunden werden. Dieser statische Parameter kann durch den Einbezug von Experten aus der Domäne definiert werden, ist jedoch auch von der Größe des Datensatzes abhängig. So sollten pro Client hinreichend Daten zur Verfügung stehen, um ein konvergierendes lokales Modelltraining zu ermöglichen und eine Unteranpassung zu vermeiden.

Client-Parameter θ_c

Epochen: Die Anzahl der lokalen Epochen auf dem Client definiert die Anzahl der Trainingsiterationen, bevor das Modell an den Server gesendet wird. Eine höhere Anzahl lokaler Epochen führt zu einer besseren Konvergenz des lokalen Modells, kann jedoch auch in einer Überanpassung an die lokalen Daten münden. So weichen bei einer höheren Anzahl lokaler Epochen die Parameter des lokalen Modells stärker von denen des repräsentativen globalen Modells ab, was zu einer schlechteren Generalisierung des globalen Modells führen kann. Aus diesem Grund ist es notwendig, die Anzahl der lokalen Epochen so zu parametrisieren, dass eine optimale Balance zwischen lokaler und globaler Modellanpassung erreicht wird.

Batch-Größe: Die Batch-Größe definiert die Anzahl an Daten, die pro Iteration im Optimierungsalgorithmus verarbeitet wird. Eine zu hoch gewählte Batch-Größe führt zu einer längeren Rechenzeit pro Iteration, da eine größere Anzahl Trainingsdaten für die Aktualisierung der Modellparameter verwendet wird. Dies ist insbesondere bei lokalen Endgeräten mit geringer Rechenleistung und Speicherkapazität ein wichtiger zu berücksichtigender Faktor. Eine zu geringe Batch-Größe kann zu einer langsamen Konvergenz des Modells führen. Im Extremfall wird lediglich ein Datenpunkt pro Iteration zur Optimierung des Modells einbezogen. Da weniger Informationen pro Iteration in das Modell einfließen, führt dies zu einer längeren Trainingszeit und der Möglichkeit, dass das Modell nicht konvergiert.

Nachdem die Parameter für die Simulation des FL-basierten Systems zur energetischen Anomalieerkennung definiert worden sind (beispielhaft in Tabelle 2), wird die Simulation vorbereitet und durchgeführt.

Tabelle 2 Exemplarische Definition der Konfigurationsparameter für die Simulation von FL

Parameter	Beschreibung
Aggregations-mechanismus	Bestimmt, wie die Modelle der Clients aggregiert werden
Anzahl Trainingsrunden	Anzahl der Durchführungen des föderierten Lernprozesses
Client-Anzahl	Anzahl der am Lernprozess beteiligten Clients
Lokale Epochen	Anzahl der Trainingsschleifen auf jedem Client
Batch-Größe	Anzahl der Datenpunkte, die in jedem Schritt verwendet werden

5.3.3.2 Simulationsbasierte Optimierung durchführen

In diesem Schritt wird die Simulation des FL-basierten Systems zur Erkennung energetischer Anomalien vorbereitet und durchgeführt (siehe Abbildung 5-19). Es werden die in Abschnitt 5.3.3.1 definierten Konfigurationsparameter θ_s und θ_c in die Simulation der FLEA-Architektur (siehe Abschnitt 5.2.2) initiiert.

Für die Simulation werden die Daten aufgeteilt und jede Teilmenge der Daten als virtueller Anomaliedetektor bezeichnet. Die virtuellen Anomaliedetektoren können über den Anlagetrainingsadapter mit dem zentralen Server und dem Modellkoordinator kommunizieren, um die approximierten Modellparameter an den Server zu senden. Der Server befindet sich innerhalb der Simulation nicht auf einer anderen Recheninstanz, sondern kann durch die virtuelle Entkopplung von Clients und Server auch auf einer einzigen Recheninstanz ausgeführt werden. Somit ist eine Evaluierung verschiedener Konfigurationen und Szenarien im Rahmen des Einsatzes von FL möglich.

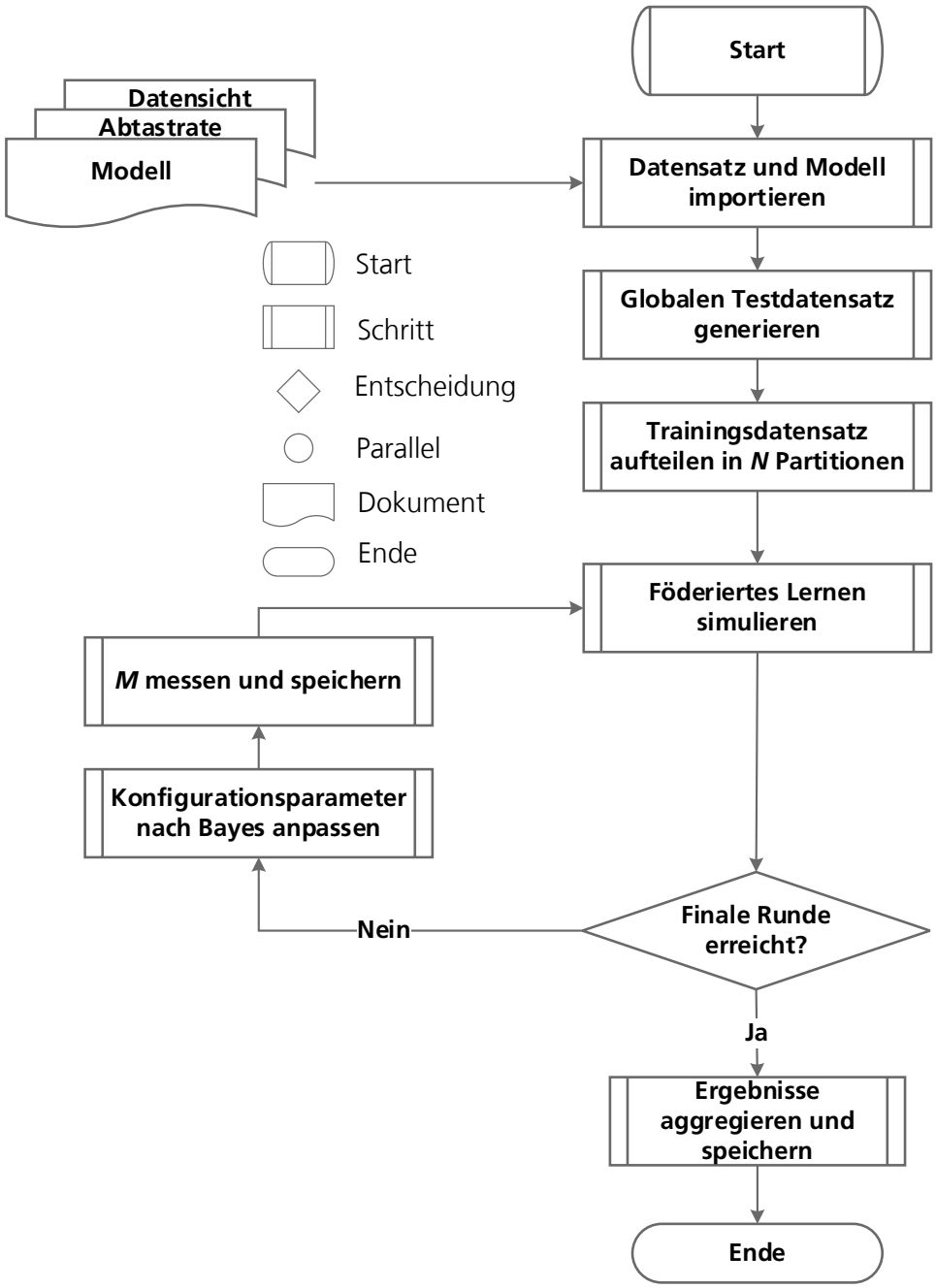


Abbildung 5-19 Ablaufdiagramm zur simulationsbasierten Optimierung der Konfigurationsparameter

Ist das grundlegende System der Simulation definiert, fließen die entwickelten Ergebnisse aus den vorherigen Phasen und die Konfigurationsparameter aus dem vorherigen Schritt (siehe Abschnitt 5.3.3.1) mit in die Vorbereitung der Simulation ein. Zum einen werden der in Abschnitt 5.3.1.1 erstellte Datensatz und die optimierten Parameter der Datenverarbeitung, z. B. Augmentationsmechanismen und die Abtastrate, genutzt, um

einen Datensatz für die Durchführung der Simulation zu erstellen. Dabei ist es notwendig, die Daten je nach Anzahl der in den Konfigurationsparametern definierten Clients aufzuteilen. Für eine zusätzliche Möglichkeit der Evaluation des globalen Modells wird ein sog. *globaler Testdatensatz* aus dem verarbeiteten Datensatz erstellt, der auch zum Testen der Clients genutzt werden kann. Dies garantiert, dass jedes lokale Modell des Clients mit den gleichen Daten evaluiert werden kann. Eine weitere wichtige Eingangsgröße ist die identifizierte Modellarchitektur (siehe Abschnitte 5.3.1.4 und 5.3.1.5). Diese Modellarchitektur kann abhängig von der Implementierung direkt aus den jeweiligen Ergebnissen übernommen und in die Simulation integriert werden.

Ein zentrales Element von FL ist die Verteilung der Daten auf mehrere Clients. Daher werden die Daten entsprechend der Anzahl der Clients partitioniert. Jeder Client erhält eine Teilmenge der Daten; die Verteilung erfolgt zufällig, aber gleichverteilt.

Nach der Initialisierung und Vorbereitung beginnt die eigentliche Simulation. In jeder Iteration trainiert jeder Client sein lokales Modell auf seiner Datenpartition. Nach Abschluss des lokalen Trainings werden die Modelle aller Clients zu einem globalen Modell aggregiert. Dieser Prozess wird für eine vordefinierte Anzahl Runden wiederholt. Parallel zum Trainingsprozess wird eine kontinuierliche Evaluation durchgeführt. Dabei wird das aggregierte globale Modell regelmäßig gegen einen Testdatensatz evaluiert.

Für die Auswahl der Konfigurationsparameter wird zusätzlich eine Metrik benötigt, mit der die Konfiguration bewertet werden kann. Ein klassischer Ansatz wäre die Einbeziehung bereits definierter Evaluationsmetriken, z. B. des F1-Scores. Neben den eigentlichen Evaluationsmetriken spielen jedoch auch Aspekte wie die Fairness zwischen den einzelnen Clients eine übergeordnete Rolle. Gerade in Anwendungsbereichen von FL, in denen die Datenquellen stark heterogen sein können, kann es zu erheblichen Unterschieden in der Evaluationsmetrik zwischen den Clients kommen. Die Einbeziehung der Fairness zwischen den einzelnen Clients in die Evaluationsmetrik stellt sicher, dass diese Unterschiede minimiert werden, während gleichzeitig eine hohe Gesamtleistung des Modells gewährleistet bleibt. Unter Berücksichtigung der Evaluationsmetrik und der Fairness wird die folgende kombinierte Metrik für die Optimierung der Konfigurationsparameter definiert.

Sei C die Anzahl der Clients im Training und $E_{i,j}$ die Evaluationsmetrik des j-ten Clients in der i-ten Runde im FL, dann ist die Fairness $Fair_i$ in der i-ten Runde definiert als:

$$Fair_i = \sqrt{\frac{1}{C}\sum_{j=1}^{C}(E_{i,j} - \bar{E}_i)^2}$$

Formel 5-17 Berechnung der Fairness auf Grundlage der Evaluationsmetrik in einer bestimmten Trainingsrunde

wobei $\bar{E}_i$ der Durchschnitt der Evaluationsmetriken für alle Clients in der i-ten Runde ist:

$$\bar{E}_i = \frac{1}{C}\sum_{j=1}^{C} E_{i,j}$$

Formel 5-18 Berechnung der durchschnittlichen Evaluationsmetrik für alle Clients in einer bestimmten Runde

Um die durchschnittliche Fairness über die letzten Runden R und bei einer maximalen Rundenanzahl T zu berechnen kann folgende Formel genutzt werden:

$$\overline{Fair_R} = \frac{1}{R}\sum_{i=T-R}^{T}\sqrt{\frac{1}{C}\sum_{j=1}^{C}(E_{i,j} - \bar{E}_i)^2}$$

Formel 5-19 Berechnung der durchschnittlichen Fairness in einer bestimmten Anzahl an Runden

Parallel dazu kann die durchschnittliche Evaluationsmetrik $\overline{E_R}$ innerhalb der letzten Runden R folgendermaßen berechnet werden:

$$\overline{E_R} = \frac{1}{R}\sum_{i=T-R}^{T} E_i$$

Formel 5-20 Berechnung der durchschnittlichen Evaluationsmetrik in einer bestimmten Anzahl an Runden

Mithilfe von $\overline{Fair_R}$ und $\overline{E_R}$ kann die kombinierte Metrik E_{komb} berechnet werden:

$$E_{komb} = \alpha \times \overline{E_R} - \beta \times \overline{Fair_R}$$

Formel 5-21 Berechnung der kombinierten Metrik anhand der durchschnittlichen Evaluationsmetrik und der Fairness

Um die Stabilität und Robustheit des Modells über die Zeit zu berücksichtigen, wird nicht nur die Evaluationsmetrik der letzten Trainingsrunde verwendet. Stattdessen wird das

arithmetische Mittel $\overline{E_R}$ und $\overline{Fair_R}$ der letzten 10 % der Runden von FL berechnet. Die Auswahl der letzten 10% der Trainingsrunden zur Berechnung von $\overline{E_R}$ und $\overline{Fair_R}$ beruht auf der Annahme, dass die Leistung eines Modells gegen Ende des Trainings stabil bleibt. Der Einbezug des arithmetischen Mittels der letzten 10 % der Runden reduziert das Rauschen und die Varianz der Metriken, die in früheren Phasen des Trainings auftreten können, und konzentriert sich auf den Zustand des Modells nach Abschluss des Trainings. Zudem werden Ausreißer durch die Mittelwertbildung der letzten Runden geglättet.

Der darauffolgende bayessche Optimierungsprozess zielt darauf ab, die Konfigurationsparameter θ_s und θ_c so zu wählen, dass E_{komb} maximiert wird:

$$(\theta_s^*, \theta_c^*) = \arg\max_{\theta_s, \theta_c} \mathbb{E}\left[E_{komb} \mid \theta_s, \theta_c\right]$$

Formel 5-22 Bayesscher Optimierungsprozess für die Identifikation der idealen Konfigurationsparameter

Dabei ist $\mathbb{E}[E_{komb} \mid \theta_s, \theta_c]$ als erwarteter Wert der Metrik E_{komb} definiert, gegeben der zuvor festgelegten Konfigurationsparameter θ_s und θ_c. E_{komb} bei θ_s^* und θ_c^* werden mit zwei definierten Szenarien verglichen (siehe Abbildung 5-20).

Zum einen werden die Daten in gleich große Client-Partitionen $D_1, D_1, \dots D_N$ aufgeteilt. Für jeden Client i wird ein Modell M_i auf Basis der identifizierten Modellarchitektur aus Phase I und dessen spezifischer Clientpartition D_i trainiert. Die Evaluationsmetriken dieser Modelle werden anhand eines globalen Testdatensatzes D_{test} ermittelt. Die dezentrale Evaluationsmetrik wird durch den Mittelwert der Evaluationsmetriken der lokalen Modelle ermittelt:

$$\mathrm{E}_{\mathrm{dezentral}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} E_i$$

Formel 5-23 Berechnung der durchschnittlichen Evaluationsmetrik des dezentralen Trainings

wobei E_i die Evaluationsmetrik des i-ten lokalen Modells M_i mit dem Testdatensatz D_{test} ist. Zum anderen werden nach dem dezentralen lokalen Training die Daten zentral auf der gleichen Modellarchitektur trainiert. Die zentrale Evaluationsmetrik $\mathrm{E}_{\mathrm{zentral}}$ kann angelehnt an Abschnitt 5.3.1.5 ermittelt werden.

Im Erfolgsfall sollte daher die Evaluationsmetrik von FL $\overline{E_R}$ zwischen dem dezentralen und dem zentralen Training liegen, wobei eine Annäherung an das zentrale Training zu bevorzugen ist:

$$\mathrm{E}_{\mathrm{dezentral}} \leq \overline{E_R} \leq \mathrm{E}_{\mathrm{zentral}}$$

Formel 5-24 Darstellung der Beziehung der unterschiedlichen Szenarien

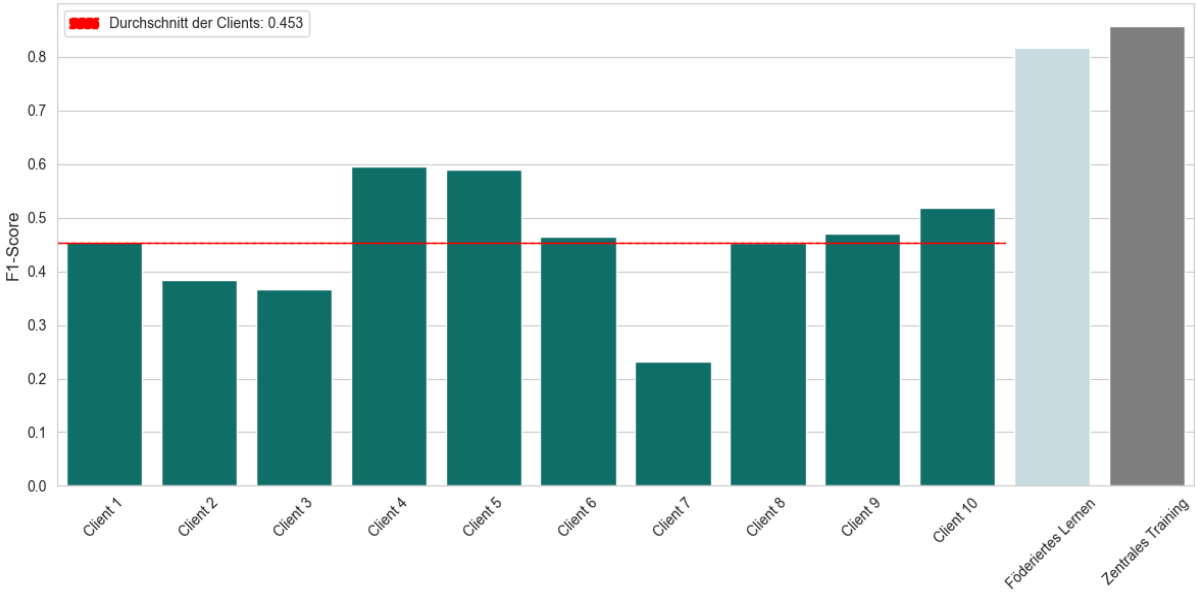


Abbildung 5-20 Exemplarische Darstellung des Vergleichs der Evaluationsmetrik zwischen dezentralem Training, zentralem Training und FL

Im nächsten Schritt wird der durchschnittliche Verlauf des Trainings von FL hinsichtlich der gewählten Evaluationsmetrik $\overline{E_R}$ und der Fairness $\overline{Fair_R}$, wobei $R = T$ ist, für jede Konfiguration betrachtet. Dabei wird nicht nur der absolute Wert dieser Metriken über die Trainingsrunden, sondern insbesondere auch deren Konvergenzverhalten betrachtet. Der Mittelwert der Evaluationsmetrik und der Fairness, berechnet über alle Konfigurationen pro Trainingsrunde, gibt Aufschluss über den allgemeinen Leistungstrend und das Stabilitätsverhalten der unterschiedlichen Konfigurationen. Dadurch können systematische Tendenzen oder Schwachstellen im FL frühzeitig erkannt werden (siehe Abbildung 5-21).

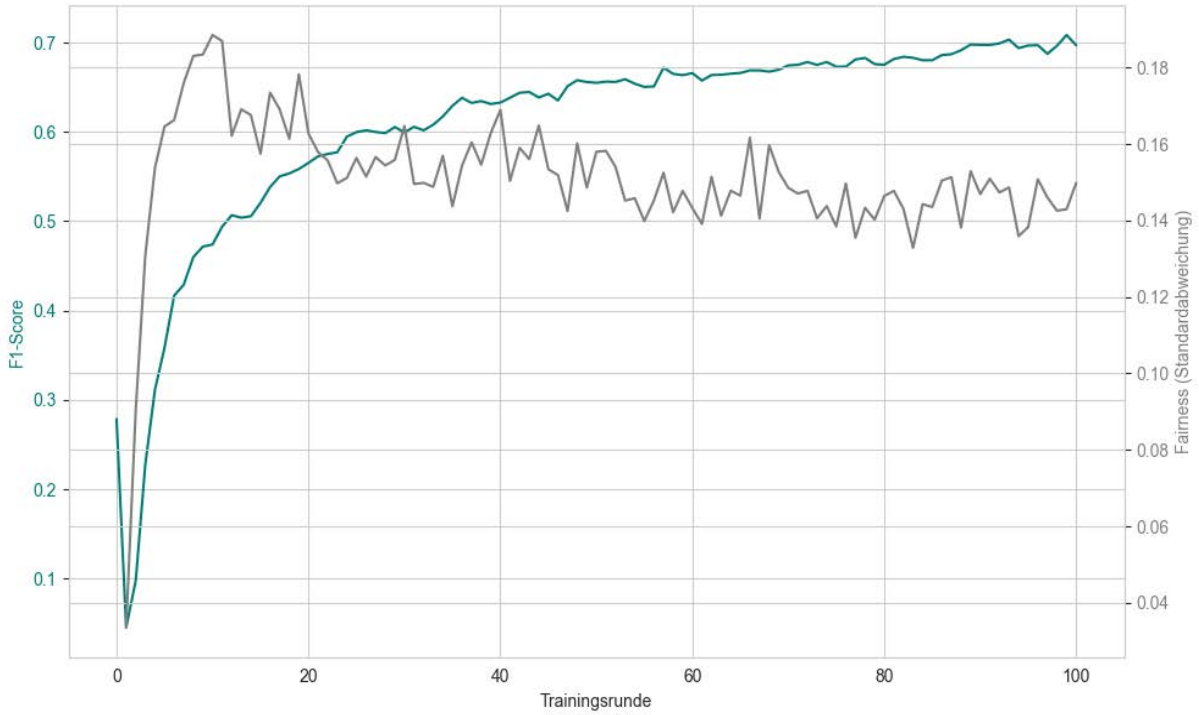


Abbildung 5-21 Exemplarische durchschnittliche Entwicklung des F1-Scores und Fairness pro Runde

Ein weiterer Indikator für eine zielführende Optimierung ist die Häufigkeit der für jede Konfiguration berechneten Evaluationsmetriken. Die Verteilung der jeweiligen Evaluationsmetriken kann aussagekräftig sein, da sie, abgesehen von gelegentlichen Ausreißern, überwiegend in einem bestimmten Bereich der Evaluationsmetrik konzentriert sein sollte. Durch die Darstellung der Häufigkeit der zuvor für E_{komb} berechneten Mittelwerte $\overline{E_R}$ für jede Konfiguration in einem Histogramm können Informationen über die Breite der Evaluationsmetrik und die Konsistenz der Konfigurationen erhalten werden (siehe Abbildung 5-22). Ausreißer könnten auf spezifische, suboptimale Kombinationen von Hyperparametern im Lernprozess hinweisen.

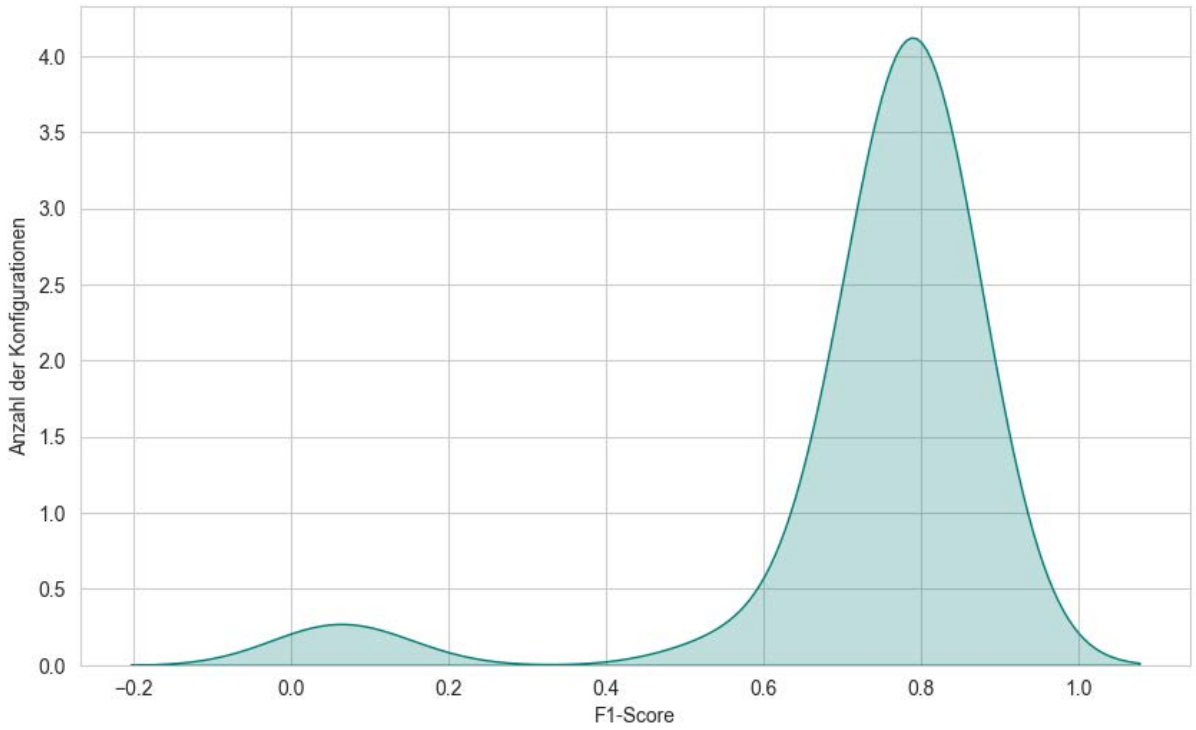


Abbildung 5-22 Exemplarische Verteilung des F1-Scores

Nach der Simulation des FL-basierten Systems zur energetischen Anomalieerkennung werden die Simulationsergebnisse evaluiert und interpretiert, um die optimale Konfiguration des Systems in die Architektur (siehe Abschnitt 5.2.2) zu überführen.

5.3.3.3 Konfigurationsparameter analysieren und bewerten

Die Evaluierung der Simulation des FL-basierten Systems zur energetischen Anomalieerkennung soll unter zwei Gesichtspunkten erfolgen. Einerseits werden die in der Simulation variierten Konfigurationsparameter hinsichtlich des Einflusses auf die Fairness und die Evaluationsmetrik untersucht. Andererseits kann mit der Kombination der Analyse und des Ergebnisses der bayesschen Hyperparameteroptimierung eine Plausibilitätsüberprüfung erfolgen.

Um den Einfluss der einzelnen Konfigurationsparameter auf die beiden Metriken zu bestimmen, kann der durchschnittliche Verlauf von FL pro Runde in Abhängigkeit des einzelnen Werts der zu untersuchenden Konfigurationsparameter analysiert werden.

Diese Analyse ermöglicht es, Muster zu erkennen, die spezifisch für bestimmte Parameterwerte sind. Beispielsweise könnte ein bestimmter Aggregationsmechanismus in den ersten Runden eine schnelle Konvergenz aufzeigen, während ein anderer über die Zeit hinweg stetigere Verbesserungen liefert. Eine beispielhafte Darstellung zeigt Abbildung 5-23. Bei nominalen Konfigurationsparametern sollte eine Analyse des Verlaufs aller Ausprägungen durchgeführt werden, da eine metrische Analyse und Berechnung des Pearson-Korrelationskoeffizienten nicht möglich ist.

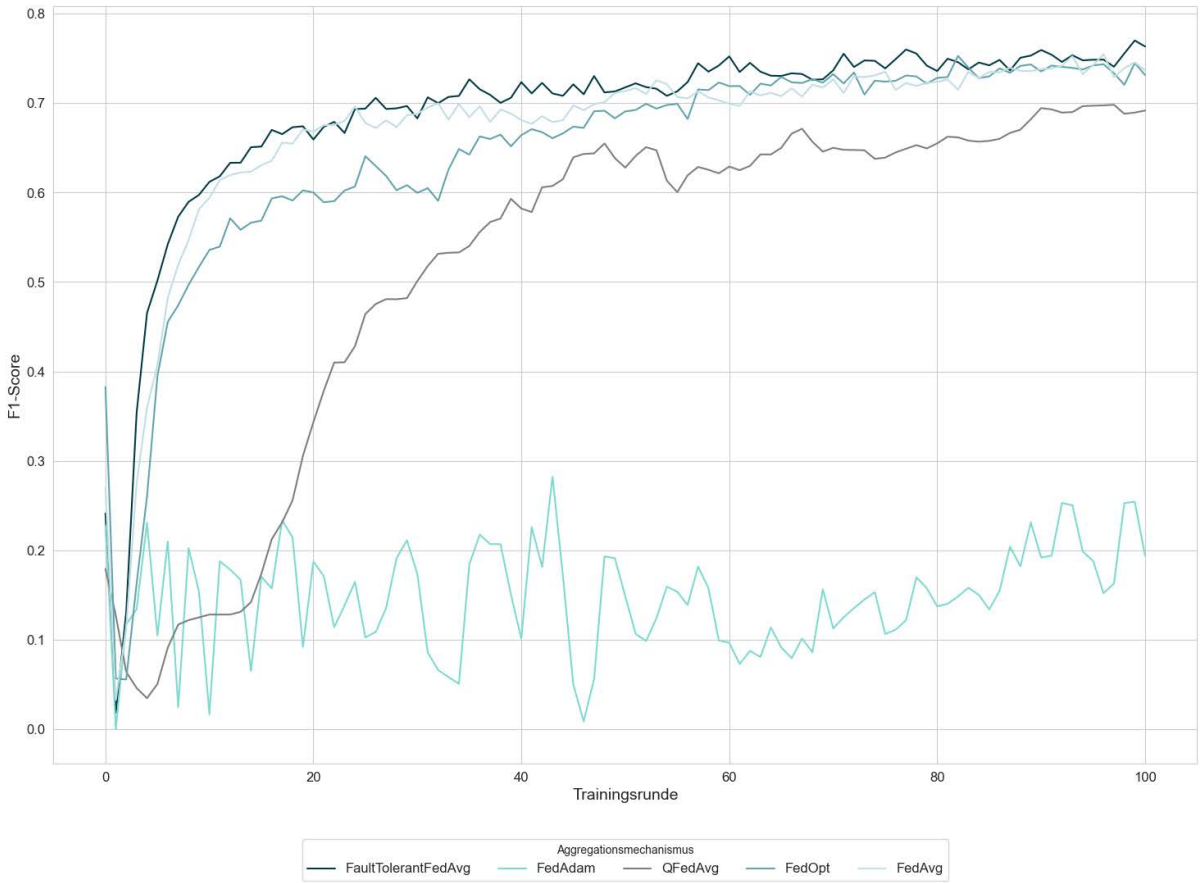


Abbildung 5-23 Mittelwert der F1-Scores für jeden Aggregationsmechanismus

Darüber hinaus ermöglicht eine Matrixanalyse, bei der die einzelnen Parameterwerte und die entsprechenden durchschnittlichen Werte für die Evaluationsmetrik und der Fairness gegenübergestellt werden, eine visuelle Analyse der Strategien, die im Durchschnitt sowohl hohe Fairness als auch hohe Evaluationsmetrik aufweisen. Bei der Erstellung eines Diagramms (siehe Abbildung 5-24) auf Grundlage der berechneten Matrix repräsentiert

jeder Punkt einen bestimmten Parameterwert (z. B. einen bestimmten Aggregationsmechanismus). Die Position des Punktes im Koordinatensystem gibt Aufschluss über die durchschnittliche Evaluationsmetrik und die Fairness: Punkte, die näher am linken oberen Rand liegen, weisen hohe Fairness und hohe Evaluationsmetrik auf. Sie sind daher im optimalen Bereich.

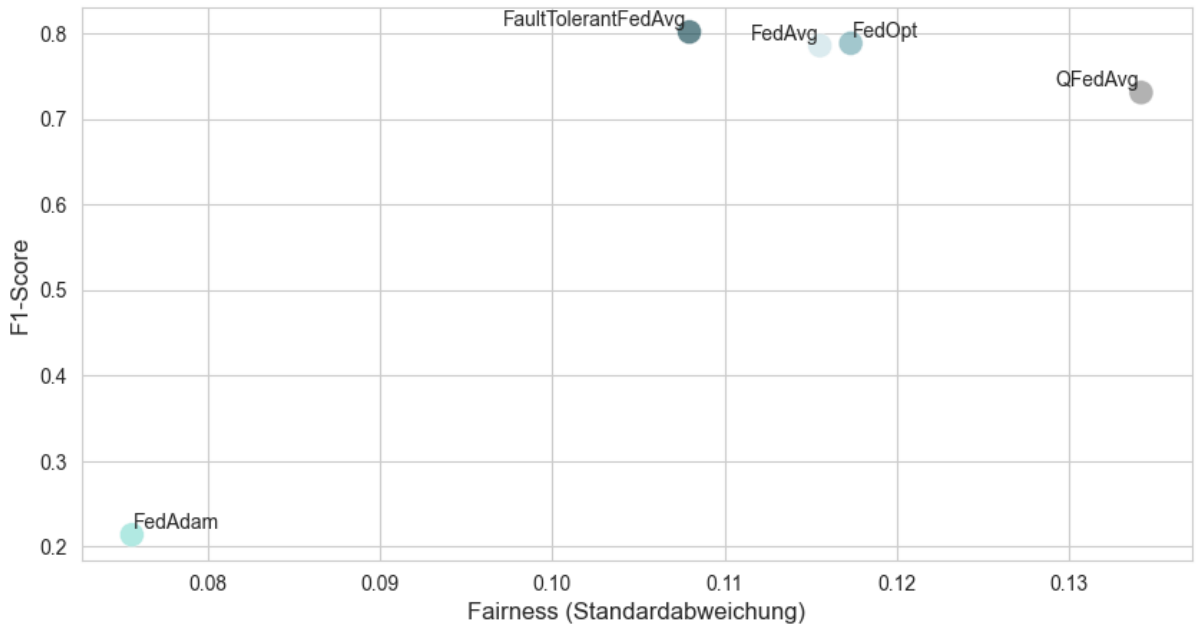


Abbildung 5-24 Exemplarische Darstellung der Aggregationsmechanismen in Relation zu F1-Score und Fairness

Um lineare Zusammenhänge zwischen den verschiedenen Konfigurationsparametern und den Metriken sowie zwischen den Metriken zu identifizieren, wird eine Korrelationsmatrix berechnet. Diese Matrix, beispielhaft dargestellt in Abbildung 5-25, zeigt die Pearson-Korrelationskoeffizienten zwischen allen Paaren von Parametern und Metriken. Werte nahe +1 oder -1 zeigen eine starke positive oder negative Korrelation an, während Werte nahe 0 eine schwache oder keine Korrelation indizieren.

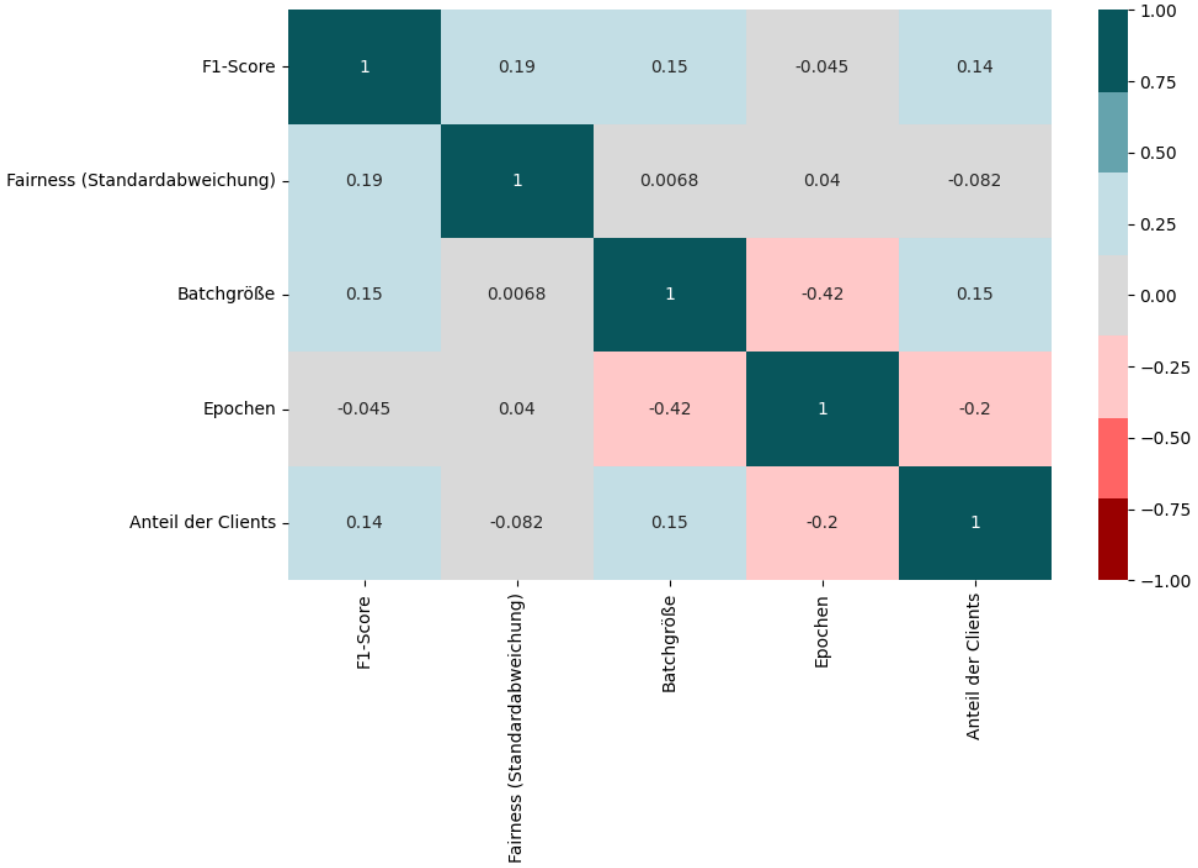


Abbildung 5-25 Exemplarische Korrelationsmatrix dargestellt als Heatmap der Konfigurationsparameter und der definierten Leistungsmetriken

Die vorgestellte systematische Analyse und Bewertung des Einflusses der Konfigurationsparameter auf die Evaluationsmetrik kann somit die bayessche Optimierung plausibilisieren. Darüber hinaus ermöglicht eine solche Analyse auch die Identifikation von Parametern, die keinen oder nur geringen Einfluss auf die Evaluationsmetrik haben. Dies ist insofern von Bedeutung, als auf diese Weise die Komplexität und die Dimensionalität des Optimierungsproblems reduziert werden können. In der Praxis kann dies die Optimierung beschleunigen und zu stabileren, konsistenteren Ergebnissen führen.

5.4 Zwischenfazit

Aufbauend auf dem Relevanz- und Stringenzzyklus wurden auf Grundlage der abgeleiteten Anforderungen und des Standes der Technik zwei Artefakte entwickelt. Im

ersten Schritt wurde in Abschnitt 5.2 die FLEA-Architektur für das FL-basierte System anhand der Vorveröffentlichungen des Autors hergeleitet und erweitert. Anschließend folgte die Entwicklung der Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen. Das Ziel der Methode ist die Entwicklung der funktionalen Komponenten und die Konfiguration der anwendungsbezogenen FLEA-Architektur. In den Abschnitten 5.3.1–5.3.3 werden die drei Phasen der Methode entwickelt.

In der ersten Phase erfolgt die Entwicklung eines Modells zur energetischen Anomalieerkennung. Mit der umfassenden Definition des Problems unter Berücksichtigung von Experten können Energiezeitreihen und kontextbasierte Betriebsdaten erfasst werden. Diese werden zu einer kontextbasierten Energiezeitreihe vorverarbeitet, wobei die Integration der Energiezeitreihen und der kontextbasierten Betriebsdaten im Vordergrund steht. Zusätzlich können unterschiedliche Datensichten generiert werden. Es erfolgt die Extraktion von zusätzlichen statistischen und prozessabhängigen Merkmalen, um rein merkmalsbasierte Datensichten zu erstellen. Durch das Modelltraining und die -auswahl kann die Modellarchitektur identifiziert werden, die die zuvor im PEMD-Framework definierte Evaluationsmetrik aus Trainingszeit und modellspezifischer Evaluationsmetrik für das ML-Modell maximiert.

Nachdem in der ersten Phase ein Modell, die zugehörige Datensicht, die Abtastrate und mögliche synthetische Daten identifiziert wurden, erfolgt die Implementierung der funktionalen Komponenten des Anomaliedetektors für die Architektur des FL-basierten Systems. Diese Implementierung erfordert den Aufbau der Datenpipeline von der Datenquelle bis zur Inferenz, die Entwicklung von Agenten für das Einleiten einer Gegenmaßnahme bei einer erkannten Anomalie und die eigentliche Bereitstellung des Anomaliedetektors.

In der letzten Phase folgt die für die Auslegung des FL-basierten Systems notwendige Parameteroptimierung, um die Konfiguration für die in Abschnitt 5.2.2 definierte FLEA-Architektur zu erhalten. Im ersten Schritt werden dazu die möglichen Konfigurationsparameter definiert und initialisiert. Die Definition erfolgt anhand der

vorangegangenen wissenschaftlichen Literaturanalyse und der Vorveröffentlichungen des Autors (Kaymakci et al. 2021c; Karnebogen et al. 2023). Daraufhin soll die Simulation des Trainingsprozesses mit einem zuvor definierten Parameterraum durchgeführt werden. Kern der Simulation ist die Kommunikation zwischen dem Server und den Clients, da hiermit die Anzahl der Clients ohne die Implementierung von Applikationen auf unterschiedlicher Hardware variiert werden kann. Im letzten Schritt sollen die Simulationsergebnisse umfassend evaluiert werden, um dadurch eine optimale Konfiguration des FL-basierten Systems zur energetischen Anomalieerkennung zu identifizieren.

Auf Grundlage der in diesem Kapitel entwickelten Methode und der Architektur sollen in Kapitel 6 zwei Anwendungsfälle betrachtet werden, bei denen die Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung angewandt wird.

6 Fallbeispiele

Die letzten Schritte des Gestaltungszyklus dienen der Implementierung und Validierung der zuvor entwickelten Artefakte. Dazu werden die einzelnen Phasen der Methode implementiert. Zusätzlich werden die jeweiligen Anwendungsgebiete der unterschiedlichen Anwendungsfälle definiert, um die implementierten Artefakte in den Anwendungsgebieten zu validieren. Dazu werden Anwendungsfälle zur energetischen Anomalieerkennung beim Lichtbogenschweißen sowie zur Leckage-Erkennung bei pneumatischen Aktorbewegungen über den Massenstrom näher betrachtet. Die Erkennung von Anomalien beim Lichtbogenschweißen ist insbesondere wichtig, um Energie- und Materialverschwendung zu vermeiden und die Qualität der Schweißnähte zu gewährleisten. Unerkannte Unregelmäßigkeiten im Schweißprozess können nicht nur die Energieeffizienz beeinträchtigen, sondern auch die Qualität der Endprodukte gefährden. Der zweite Anwendungsfall, die Leckage-Erkennung bei pneumatischen Aktorbewegungen, weist gleichfalls ein hohes Einsparpotenzial beim Energieverbrauch auf, denn in der Produktion können unbemerkte Leckagen zu einem erhöhten Energieverbrauch führen. Dies beeinträchtigt die Energie- und Kosteneffizienz der betroffenen Produktionssysteme, was die Bedeutung einer präzisen Leckage-Erkennung in pneumatischen Systemen unterstreicht.

6.1 Erkennung von Bedienfehlern beim Lichtbogenschweißen

Für das erste Fallbeispiel wurde die energetische Anomalieerkennung bei Schweißprozessen betrachtet, da die Anomalieerkennung vor allem anhand elektrischer Energiedaten erfolgen kann. Insbesondere das Lichtbogenschweißen, bei dem zwei der wichtigsten Einstellparameter die Stromstärke und die Spannung für einen korrekten Lichtbogen sind, wurde analysiert. Der Prozess des Lichtbogenschweißens ist sowohl beim

manuellen als auch beim automatisierten Lichtbogenschweißen immer noch durch manuelle Versuche mit einer Vielzahl von Schweißversuchen zur Ermittlung der optimalen Schweißparameter gekennzeichnet. Für die Schweißversuche und die Nacharbeit müssen erhebliche Ressourcen an Material, Energie und Arbeitskraft aufgewendet werden, um Schweißnahtfehler zu finden, zu beheben und eine Wiederholung auszuschließen. Es soll zukünftig erreicht werden, dass digitale Assistenzsysteme die manuelle Parametrisierung verkürzen und Schweißnahtfehler im Rahmen der Qualitätssicherung aufzeigen. Dadurch können Ressourcen wie Energie und Material eingespart werden.

6.1.1 Anomalien beim Lichtbogenschweißen

Das Schweißen ist ein Teil der Fertigungsgruppe des Fügens. Es handelt sich um ein stoffschlüssiges, unlösbares Verfahren, das in nahezu allen Industriezweigen eingesetzt wird. Dabei kann unter Einbringung von Wärme die Stoffverbindung mit oder ohne Schweißzusatzwerkstoffen erreicht werden. Zur Verbesserung der Schweißnaht werden auch Hilfsstoffe wie Schutzgase eingesetzt (Matthes & Schneider 2016). Bei vielen Schweißgeräten wird die Wärme über einen Lichtbogen erzeugt, der zwischen Werkstück und Werkzeug gezündet wird. Das Werkzeug ist dabei meist ein Schweißbrenner, bei dem durch Strom und die Spannung des Schweißgerätes ein Lichtbogen zwischen Elektrode und Werkstück gezündet wird.

Die Qualität einer Schweißnaht und deren Ausführung sind stark von der persönlichen Erfahrung des Schweißers sowie der verantwortlichen Schweißaufsichtsperson abhängig, die die Parametrierung und Qualitätskontrolle übernimmt. Durch das Aufzeigen von Schweißnahtdefekten, die durch falsche Parametrisierung des Schweißers oder andere personenspezifische Bedienfehler entstehen, können Unterschiede in den Fachkompetenzen der Schweißer minimiert werden. Das Erkennen von Änderungen der Prozessparameter, die von den als ‚normal' angesehenen Werten abweichen, kann ein erstes Anzeichen für einen Ausfall der Schweißanlage oder für Qualitätsprobleme sein. Die rechtzeitige Erkennung solcher Ereignisse in den Rohdaten, z. B. erhöhter

Energieverbrauch, der während der Prozess- oder Maschinenüberwachung aufgezeichnet wird, könnte eine Beschädigung der Maschine verhindern oder das Risiko von Ausfallzeiten während des Produktionsprozesses verringern. Dadurch können auch Personen mit geringerem schweißtechnischem Verständnis in Betriebe eingegliedert werden und die Schweißgeräte fachgerecht bedienen. Zusätzlich ermöglicht die automatisierte Erkennung der Bedienfehler eine Reduktion von manuellen Schweißnahtprüfungen und somit auch Material und Ressourcen, die für wertschöpfende Tätigkeiten genutzt werden können.

6.1.2 Entwicklung des Modells zur energetischen Anomalieerkennung

In diesem Abschnitt wird die erste Phase zur Auslegung des FL-basierten Systems zur energetischen Anomalieerkennung für die vorliegende Fallstudie auf Basis der in Abschnitt 5.3.1 entwickelten Vorgehensweise durchgeführt.

6.1.2.1 Problem definieren

Für die Problemdefinition in dieser Fallstudie wird das vorgestellte PEMD-Framework benutzt. Im Folgenden sollen die vier Dimensionen vorgestellt und erläutert werden (siehe Tabelle 3).

Tabelle 3 Darstellung des PEMD-Frameworks für die Fallstudie

	Dimension	**Ausprägung**
PEMD-Framework	***Leistung***	Reduktion der Bedienfehler beim Lichtbogenschweißen zur Vermeidung von Ausschuss und qualitativ minderwertigen Werkstückverbindungen F1-Score
	Umgebung	Schweißanlage, Stromquelle, Einplatinencomputer zur Datenerfassung, ausführender Schweißfachmann, mobile Betriebsdatenerfassung in Form einer mobilen Applikation für mobile Endgeräte
	Maßnahmen	Überwachungsfunktion am Einplatinencomputer, Visualisierung der fehlerhaften Schweißungen in einem Dashboard
	Daten	Sensordaten: Stromstärke, Spannung, Drahtvorschub, Betriebsdaten: Einstellparameter, Auftragsnummer, Schweißer, Fehlerklasse,

Das grundlegende Ziel und somit die Dimension **Leistung** des Schweißanlagenbauers ist die Reduktion von Bedienfehlern beim Lichtbogenschweißen zur Vermeidung von Ausschuss und qualitativ minderwertigen Werkstückverbindungen. So können durch die automatisierte und schnelle Erkennung von Bedienfehlern des Schweißers, etwa die falsche Auswahl des Schutzgases oder das Einlegen des falschen Drahtdurchmessers Gegenmaßnahmen für das korrekte Schweißen ergriffen werden. Im Kontext der Reduktion von Bedienfehlern beim Lichtbogenschweißen ist es wichtig, geeignete Evaluationsmetriken zu verwenden, um den Erfolg der angewendeten Maßnahmen zu messen. Da bei der Erkennung von Bedienfehlern im Schweißprozess sowohl eine Vermeidung von zu vielen fehlerhaften Alarmen als auch die Nichterkennung von Fehlern berücksichtigt werden muss, wird im Folgenden der F1-Score als harmonisches Mittel aus Präzision und Recall gewählt.

Die **Umgebung** der Fallstudie besteht aus der eigentlichen Stromquelle zur kontrollierten Energieversorgung des Schweißprozesses. Um eine Überwachung und Steuerung des Schweißprozesses zu ermöglichen, wird ein Einplatinencomputer zur Datenerfassung und -verarbeitung eingesetzt. Die Daten können mit der aktuellen Version des Einplatinencomputers zusätzlich in ein Cloudsystem geschrieben werden. Der ausführende Schweißfachmann ist für die Durchführung des Schweißprozesses verantwortlich. Er bedient die Stromquelle sowie den Brenner und stellt die Schweißparameter entsprechend den Anforderungen des Schweißprozesses ein. Zusätzlich wird eine mobile Betriebsdatenerfassung in Form einer Applikation für ein mobiles Endgerät oder eine webbasierte Benutzeroberfläche zur Verfügung gestellt. Dieses ermöglicht eine einfache Überwachung des Schweißprozesses und stellt eine nahtlose Integration von Menschen und Maschine sicher. Darüber hinaus können mit Hilfe dieser Applikation zusätzliche Informationen zu den einzelnen Schweißungen hinterlegt werden.

Mithilfe von Experteninterviews mit Anwendern, Anwendungstechnikern und Schweißfachingenieuren wurden zwei **Maßnahmen** identifiziert. Werden Bedienfehler erkannt, sollen zum einen die Schweißungen mit einer eindeutigen ID und die erkannten Bedienfehler in einer Log-Datei auf dem Einplatinencomputer gespeichert werden, damit der Schweißfachmann einen Überblick über die einzelnen Schweißungen erhält. Zusätzlich soll die Information auf der Applikation für mobile Endgeräte visualisiert werden, damit die Beseitigung des Fehlers direkt durchgeführt werden kann.

Die Stromquelle, die für die Durchführung des Schweißprozesses eingesetzt wird, erfasst mithilfe der mobilen Applikation und des Einplatinencomputers verschiedene **Daten**, die für die Modellierung von Bedienfehlern genutzt werden können. Dabei handelt es sich zum einen um Sensordaten, die direkt während des Schweißprozesses erfasst werden, und zum anderen um Betriebsdaten, die Informationen über den Schweißprozess und dessen Durchführung enthalten. Die Sensordaten umfassen die Stromstärke, die Spannung und den Drahtvorschub. Sie können somit zur Extraktion weiterer Merkmale und zur Identifikation von Bedienfehlern genutzt werden. Darüber hinaus stehen Betriebsdaten der mobilen Applikation zur Verfügung, die Informationen über die

einzelne Schweißung und deren Durchführung enthalten. Dazu gehören Einstellparameter wie das verwendete Material oder die Dicke des Bauteils, die Auftragsnummer, der Name des Schweißers und die mögliche Fehlerklasse. Auch diese Daten können in die Erkennung der Bedienfehler miteinfließen.

Auf Grundlage der Problemdefinition wurden im Schweißlabor des Schweißanlagenbauers zwei Versuchsreihen für 200 Schweißungen mit unlegiertem Stahl in unterschiedlichen Dicken durchgeführt. Der grundlegende Aufbau des Schweißlabors ist in Abbildung 6-1 dargestellt. Dabei wurde eine Kehlnaht (45 Grad) stechend geschweißt. Der Standarddurchmesser des Drahts wurde mit 1,2 mm definiert. Die Schweißungen wurden in fehlerfreien und fehlerbehafteten Varianten durchgeführt. Sie unterschieden sich hinsichtlich der verschiedenen Schweißverfahren, z. B. dem Pulslichtbogenschweißen. Es wurden folgende Fehler betrachtet: *Falscher Drahtdurchmesser eingespannt, falsche Einstellparameter von Gas und Material der Stromquelle, falsches Gas eingeschlossen* und *falsche Einstellparameter von Gas und Material der Stromquelle mit zusätzlicher Lichtbogenlängenkorrektur*.

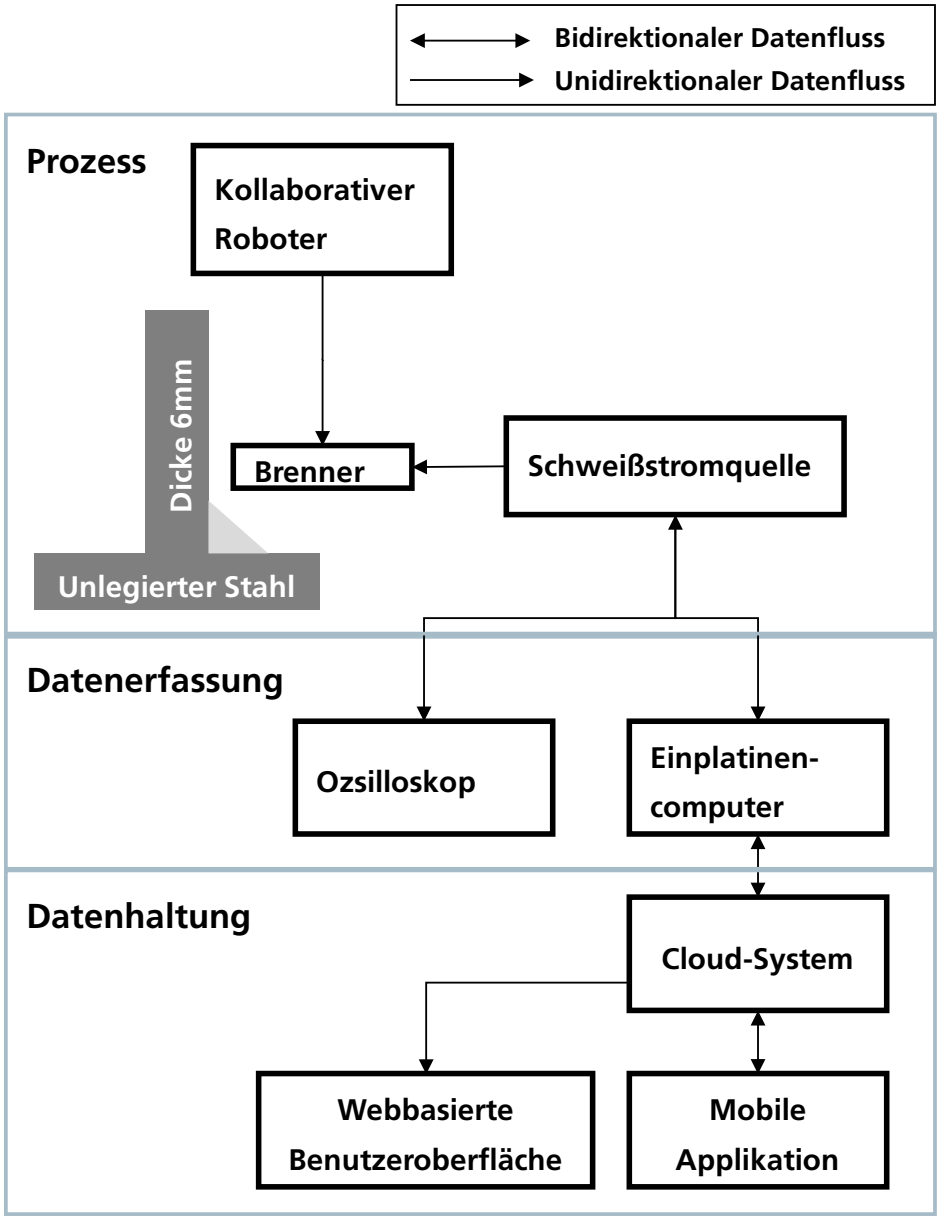


Abbildung 6-1 Darstellung des Versuchsaufbaus des Laborexperiments

Die Schweißungen eines kollaborierenden Roboters wurden mit konstanten, die Schweißungen des Handschweißers indes mit variablen Arbeitspunkten durchgeführt. Somit werden innerhalb des finalen Datensatzes unterschiedliche Anwendungsszenarien abgebildet. Die Datenerfassung erfolgte mit dem Einplatinencomputer und einem Oszilloskop als Kontrolleinheit. Um eine umfassende Auswertung der Schweißdaten zu ermöglichen, wurden die Schweißparameter mit unterschiedlichen Abtastraten (1–10 Hz) aufgezeichnet.

Nach einer erfolgreichen Schweißung wurde eine CSV-Datei mit den erfassten Daten angelegt und mit der Auftragsnummer beschriftet. Die geschweißten unlegierten Stähle wurden zusätzlich mit der Auftragsnummer beschriftet, um eine visuelle Inspektion und eine Nachverfolgung möglicher Fehler bei der Datenakquise zu ermöglichen. Die einzelnen Dateien wurden zentral gespeichert. Zusätzlich erfolgte ein Export der Betriebsdaten. In Abbildung 6-2 sind Stromstärke, Spannung, Drahtvorschub und Leistung einer fehlerfreien Schweißung dargestellt. Abbildung 6-3 zeigt exemplarisch eine fehlerhafte Schweißung.

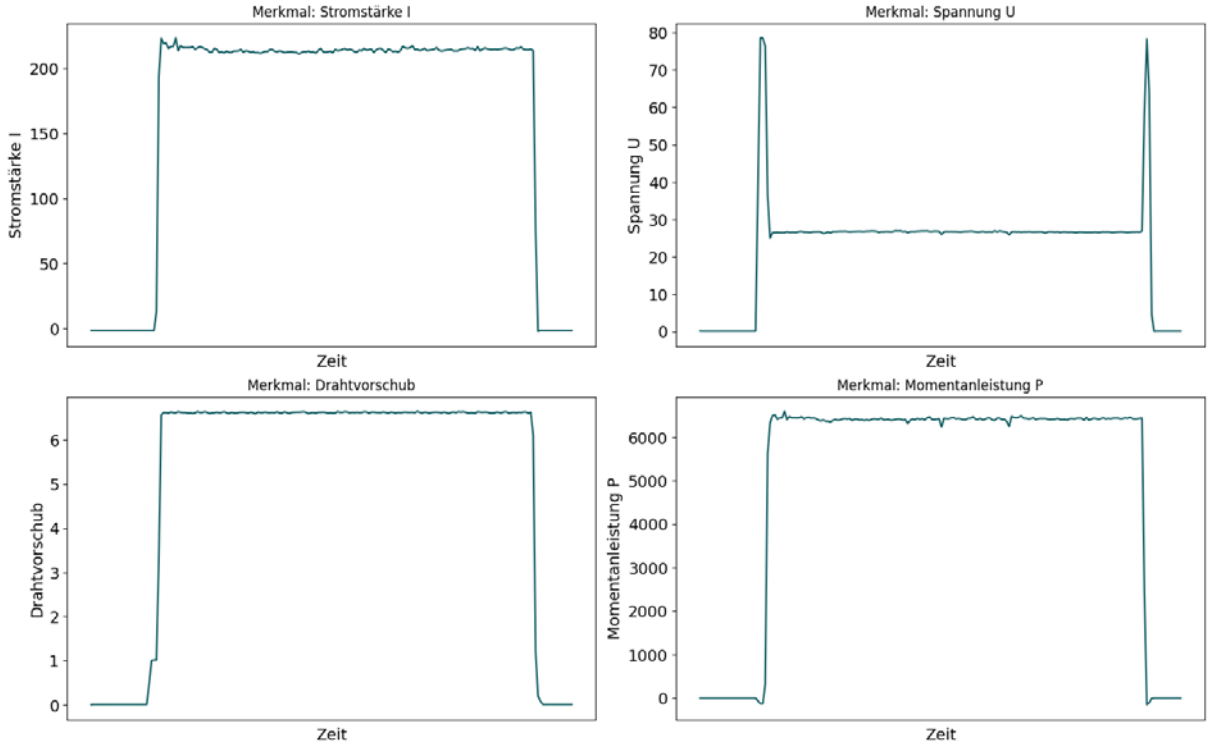


Abbildung 6-2 Visualisierung der gemessenen Größen einer normalen Schweißung

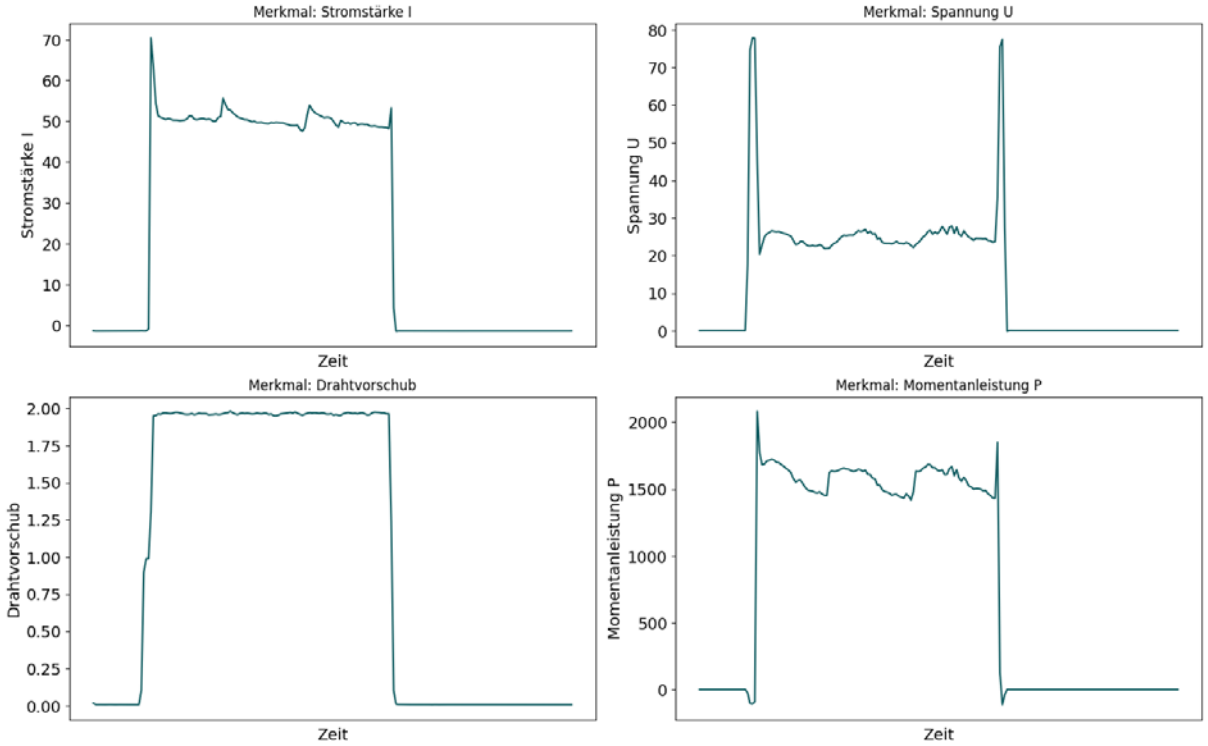


Abbildung 6-3 Visualisierung der gemessenen Größen einer fehlerhaften Schweißung

6.1.2.2 Kontextbasierte Energiezeitreihe aufbereiten

Im ersten Schritt der Datenvorverarbeitung werden die Daten aus unterschiedlichen Quellen integriert. Dazu wird in einer For-Schleife eine CSV-Datei der zugehörigen Schweißung und deren Auftragsnummer eingelesen. Die Startzeit des Schweißprozesses wird aus den Daten der mobilen Applikation abgerufen und als Zeitstempel gespeichert. Mithilfe der Schleife können alle Schweißungen zu einem Datensatz integriert werden. Zusätzlich können die weiteren kontextbasierten Betriebsdaten aus der mobilen Applikation integriert werden. Dabei erfolgt die Zusammenführung durch die eindeutige ID, die aus der Benennung der CSV-Dateien extrahiert werden kann. Die zusammengeführte Datei ist die kontextbasierte Energiezeitreihe als multivariate Energiezeitreihe inklusive der Kontextinformationen aus der mobilen Applikation. Die kontextbasierte Energiezeitreihe wird in einem geeigneten Datenformat (*Parquet*) für die weitere Verarbeitung gespeichert.

Im nächsten Schritt wird die kontextbasierte Energiezeitreihe auf fehlende Werte und signifikante Ausreißer pro Schweißung überprüft. Bei Schweißungen mit fehlenden Werten werden die Daten anhand der polynomialen Interpolation hinzugefügt. In diesem Anwendungsfall betrifft dies nur eine Schweißung, was somit keinen nennenswerten Einfluss auf die Datenmenge hat. Zusätzlich werden die Daten standardisiert, um eine schnellere Konvergenz der Approximation der Parameter während der Modellierung zu ermöglichen.

Auf Grundlage der aufbereiteten, kontextbasierten Energiezeitreihe erfolgt die Generierung der und daraufhin die Trennung des Datensatzes in einzelne Zeitreihen, um so die kontextbasierte Energiezeitreihengruppe zu generieren. In dieser Fallstudie wird aufgrund der möglichen Trennung der Schweißungen mithilfe der Auftragsnummern die rein kontextbasierte Sicht betrachtet. Die kontextbasierte Energiezeitreihe wird daher mithilfe der Auftragsnummer getrennt. Daraus resultiert die kontextbasierte Energiezeitreihengruppe. Anhand dieser können im nächsten Schritt Merkmale extrahiert werden.

6.1.2.3 Merkmale extrahieren

Wie in Abschnitt 5.3.1.3 beschrieben, werden in diesem Schritt die relevanten Merkmale der Schweißung extrahiert. Vorliegend wurde für die Extraktion von prozessabhängigen Merkmalen, bei der zusätzliche Zeitreihen erstellt werden, auf das Experten- und Domänenwissen der Schweißfachingenieure zurückgegriffen. Im Folgenden werden die zusätzlich erstellten Merkmale beschrieben, die als Zeitreihe in die kontextbasierte Energiezeitreihengruppe integriert wurden.

- Momentane Leistung: Produkt aus Strom und Spannung, das die tatsächlich abgegebene Wärmeleistung des Schweißprozesses widerspiegelt
- Quadratische Mittelwerte von Strom und Spannung

- Kurzschlussverhältnis: Verhältnis zwischen Kurzschlusszeit und Lichtbogenzeit; für nicht (absichtlich) kurzschlussbehaftete Schweißprozesse ist die Short Circuit Ratio ein aussagekräftiges Merkmal der Zeitdauer der Kurzschlussanteile im Prozess.
- Kurzschlussfrequenz: Reziprokwert der mittleren Zeitdauer zwischen aufeinanderfolgenden Kurzschlüssen
- Referenzwert-Kanal: normal (symmetrisch exponentiell) geglättete Lichtbogen-Spannung.
- Maximalwert-Hüllkurve: bildet Spannungsüberschwinger ab, die durch kratzende Kontakte, kurze Lichtbogenabrisse etc. hervorgerufen werden, und könnte Informationen über den Verschleißzustand des Stromkontaktrohres liefern
- Minimalwert-Hüllkurve: bildet Spannungseinbrüche ab, die durch Mikrokurzschlüsse hervorgerufen werden

Nach der Integration der Merkmale in den Datensatz erfolgt die Erstellung der kontextbasierten Merkmalstabelle. Für jede Schweißung werden statistische Merkmale aus der vorhandenen kontextbasierten Energiezeitreihengruppe berechnet. Dabei handelt es sich vor allem um allgemeingültige statistische Merkmale nach Christ et al. (2018), die aus einem Merkmal der jeweiligen Energiezeitreihe innerhalb der Energiezeitreihengruppe abgeleitet werden. Zu diesen Merkmalen gehören der Mittelwert, das Maximum oder das Minimum des Merkmals innerhalb der Zeitreihe. Da die Extraktion der Merkmale je Zeitreihe erfolgt, kann bei der Merkmalsextraktion eine Menge an Merkmalen entstehen, die mit in die Modellierung einfließen können. Nach der Extraktion sowohl der prozessabhängigen Merkmale als auch der aus der kontextbasierten Energiezeitreihe generierten kontextbasierten Merkmalstabelle kann die Modellierung für die zuvor erstellten Datensichten erfolgen. Zum einen wurde die kontextbasierte Energiezeitreihe erstellt. Sie umfasst 18 univariate Zeitreihen, die jeweils ein Merkmal des Schweißprozesses abbilden. Die kontextbasierte Merkmalstabelle mit einer Abtastrate von 10 Hz beschreibt die jeweiligen Schweißprozesse mit insgesamt 8249 Merkmalen, die aus der multivariaten Energiezeitreihe extrahiert wurden.

6.1.2.4 Modell trainieren

Für das Training werden die in Abschnitt 5.3.1.4 definierten Modelle initialisiert und parametrisiert, um sicherzustellen, dass die Modellkonfiguration dem jeweiligen Datensatz entspricht. Dabei werden die Daten wie in den Abschnitten 5.3.1.4 und 5.3.1.5 beschrieben in Trainings- und Testdaten getrennt. Das Training erfolgt mithilfe einer zehnfachen Kreuzvalidierung, um die statistische Signifikanz der Ergebnisse zu erhöhen.

Daraufhin werden verschiedene Experimentkonfigurationen erstellt. Die verwendeten Abtastraten wurden mit den Domänenexperten festgelegt, wobei der maximale Abtastrate bei 10 Hz liegt. Insgesamt wurden fünf Abtastraten definiert: 1, 2, 5, 8 und 10 Hz. Anschließend wird für jede Experimentkonfiguration ein Experiment durchgeführt, bei dem die Abtastrate des Datensatzes entsprechend angepasst wird. Ein spezielles Datensicht-Objekt wird erstellt, um den Datensatz für die anschließende Verarbeitung und Analyse vorzubereiten. Für jedes Experiment werden zwei verschiedene Datensichten verwendet – eine, die auf der kontextbasierten Merkmalstabelle basiert, und eine, die auf der kontextbasierten Energiezeitreihengruppe fußt. Die Kreuzvalidierung wird unter Berücksichtigung der jeweiligen Experimentkonfiguration durchgeführt, wobei die ausgewählten Augmentationsmechanismen und die Abtastrate berücksichtigt werden. Schließlich werden die Ergebnisse der verschiedenen Experimente in einer Ergebnistabelle (siehe Anhang: Tabelle 11) zusammengefasst und für die weitere Analyse wie auch Interpretation gespeichert.

Es wurden 1092 Modelle trainiert und evaluiert. Im Folgenden sollen die Ergebnisse vorgestellt werden, um das Modell mit dem höchsten F1-Score auszuwählen, das für die Entwicklung des Anomaliedetektor und für die Simulation von FL genutzt werden kann.

6.1.2.5 Modell auswählen

Im ersten Schritt wird die Anzahl der Schweißungen mit augmentierten Daten ergänzt. Die in diesem Fall genutzten Augmentationsmechanismen können dem Anhang (siehe

Tabelle 10) entnommen werden. Mit der Augmentation können anstelle von 200 Schweißungen nun 2200 Schweißungen für die Modellauswahl genutzt werden. Die Augmentation ermöglicht somit eine Erhöhung der Anzahl der Schweißungen. Dies trägt nicht nur zu einer genaueren Modellauswahl bei, sondern ermöglicht auch eine effektivere Simulation von FL.

Bei der Betrachtung der Evaluationsmetrik, gemessen am F1-Score, fällt auf, dass die Augmentation der Daten in den meisten Fällen zu einer Verbesserung der Evaluationsmetrik führt. Dies zeigt ein Vergleich der durchschnittlichen F1-Scores der augmentierten und nicht augmentierten Modelle (siehe Abbildung 6-4).

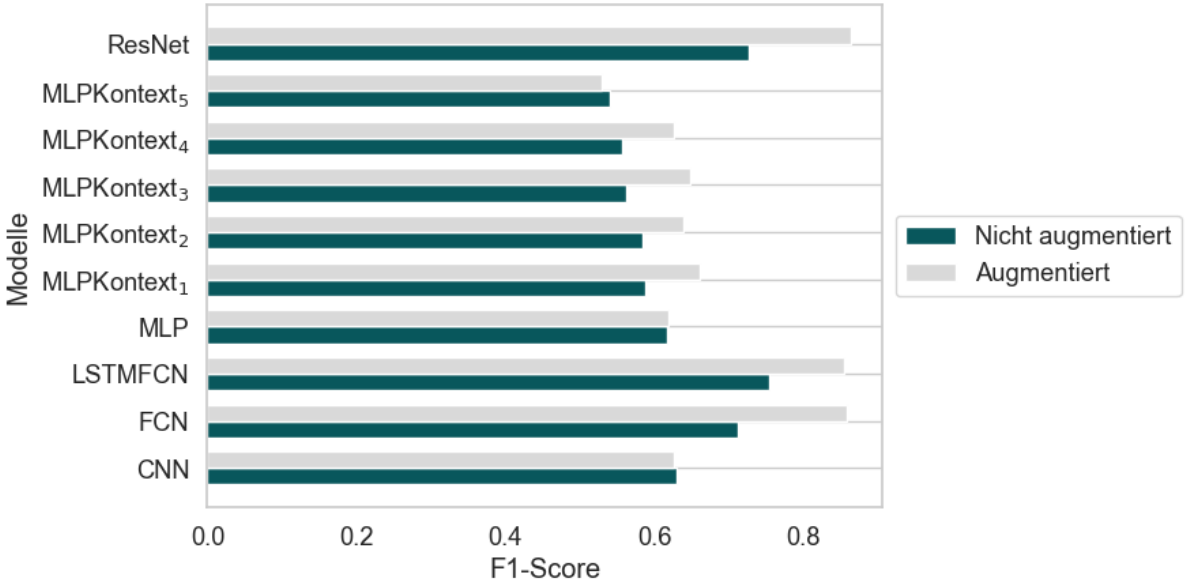


Abbildung 6-4 Vergleich der trainierten Modelle auf Grundlage von augmentierten Daten

Für Modelle wie FCN, LSTMFCN und ResNet zeigt die Augmentation eine erhebliche Verbesserung des F1-Scores. Beispielsweise verbessert sich der F1-Score des FCN von 0,71 (ohne Augmentation) auf 0,86 (mit Augmentation). Ähnlich deutliche Steigerungen sind für LSTMFCN und ResNet mit einer Verbesserung von 0,75 auf 0,85 bzw. von 0,73 auf 0,86 zu beobachten. Es gibt jedoch auch Modelle, bei denen die Augmentation zu keiner Verbesserung bzw. zu einer leichten Verschlechterung führt. So zeigt der CNN nach der Augmentation eine leichte Verschlechterung des F1-Scores von 0,63 auf 0,62. Auch das Modell $MLKontext_5$ zeigt einen Rückgang des F1-Scores von 0,54 auf 0,53. Einige der Modelle, insbesondere $MLKontext_1$, $MLKontext_2$, $MLKontext_3$ und $MLKontext_4$, profitieren

derweil von der Augmentation mit einem moderaten Anstieg des F1-Scores. Die Augmentation führt somit bei der Mehrheit der Modelle zu einem besseren F1-Score.

Zusätzlich zur visuellen Analyse des F1-Scores wurde ein statistischer Test durchgeführt, um festzustellen, ob die Unterschiede in den F1-Scores zwischen augmentierten und nicht augmentierten Daten signifikant sind. Ein gepaarter t-Test ergab einen p-Wert von 0,0049, was weit unter dem Signifikanzniveau von 0,05 liegt. Dies zeigt, dass der Unterschied zwischen den F1-Scores der beiden Gruppen statistisch signifikant ist. Insbesondere zeigt der Test, dass der durchschnittliche F1-Score für die augmentierten Daten signifikant höher ist als für die nicht augmentierten Daten.

Für die Analyse der Abtastraten zeigt Abbildung 6-5 auf der X-Achse die verschiedenen Abtastraten und auf der Y-Achse den durchschnittlichen F1-Score.

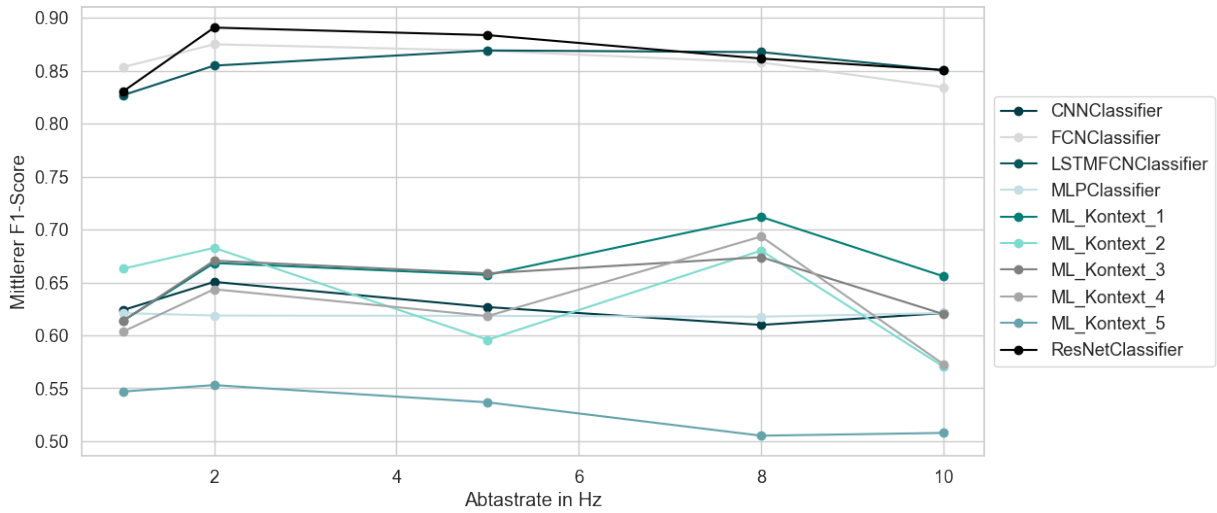


Abbildung 6-5 Visualisierung der F1-Scores in Abhängigkeit der Abtastrate und Modelle

Nach dem Modellauswahlverfahren auf Grundlage des $sLEI$ ist die beste Konfiguration, gemessen am höchsten $sLEI$, das Modell LSTMFCN mit einer Abtastrate von 2 Hz. Der $sLEI$ beträgt hier 0,834, was auf eine effiziente Kombination von F1-Score und Trainingszeit hinweist. Dieses Modell hat eine Trainingszeit von 174 s und einen F1-Score von 0,855.

Abbildung 6-6 zeigt, dass die FCN- und LSTMFCN-Modelle in den meisten Fällen höhere $sLEI$ als die anderen Modelle aufweisen. Insbesondere die FCN- und LSTMFCN-Modelle

mit niedrigeren Abtastraten (1 und 2 Hz) weisen eine höhere Effizienz als Modelle mit höheren Abtastraten auf.

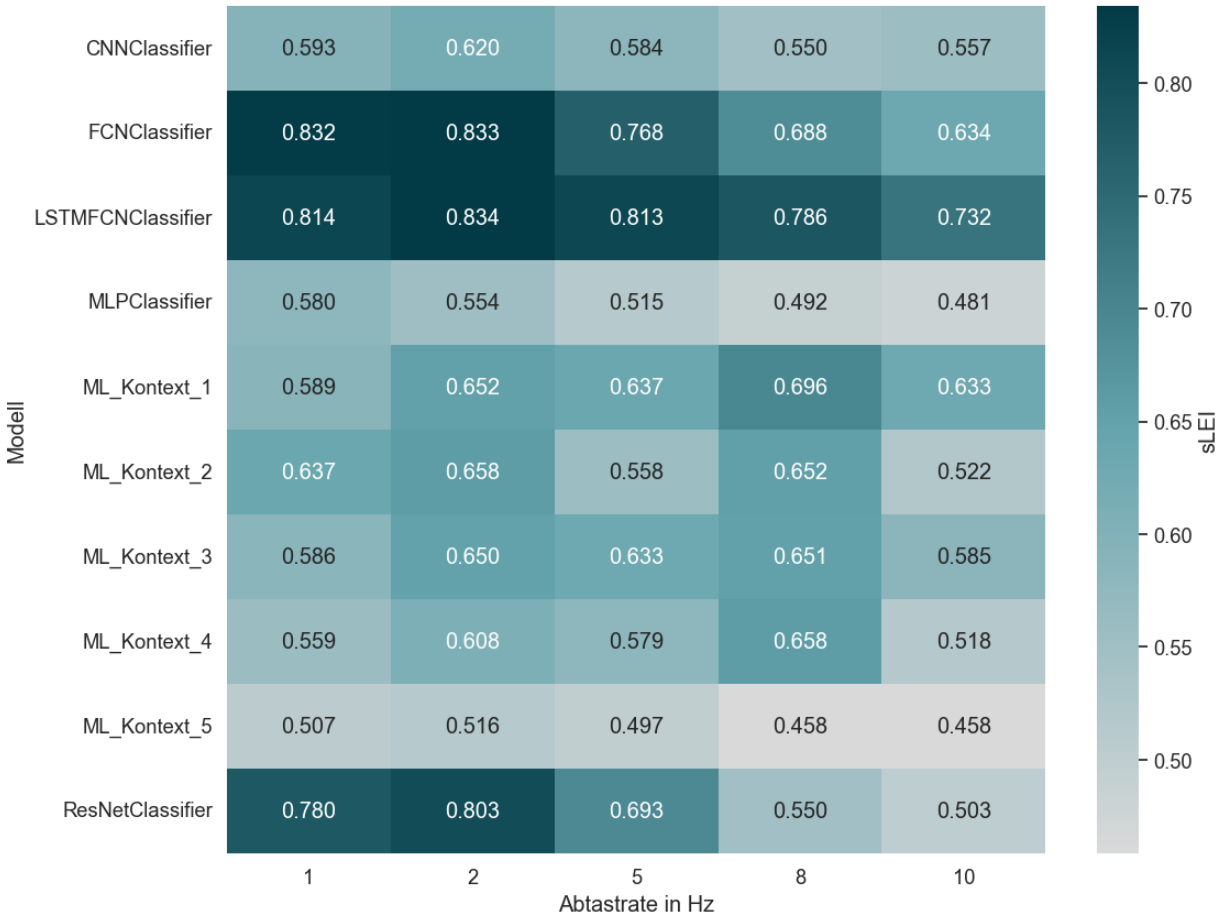


Abbildung 6-6 Heatmap des $sLEI$ der unterschiedlichen Modell-Abtastrate-Kombinationen

Die MLKontext-Modelle ($MLKontext_{1-5}$) zeigen in ihren $sLEI$ eine Variation. Insbesondere erreicht das Modell $MLKontext_2$ bei einer Abtastrate von 2 Hz den höchsten $sLEI$ von 0,658 innerhalb dieser Modellgruppe. Trotzdem liegt der $sLEI$ des Modells $MLKontext_2$ unter dem des Modells LSTMFCN.

Interessanterweise hat das Modell ResNet mit einer Abtastrate von 2 Hz den höchsten F1-Score (0,891) aller betrachteten Modelle. Allerdings hat es eine lange Trainingszeit von 544 s, sodass der $sLEI$ (0,803) niedriger ist als der des LSTMFCN-Modells mit einer Abtastrate von 2 Hz.

Insgesamt scheint somit das LSTMFCN-Modell mit einer Abtastrate von 2 Hz die dateneffizienteste Konfiguration zu sein, wenn $sLEI$ als Bewertungskriterium verwendet wird.

6.1.3 Implementierung des Anomaliedetektors

Im Folgenden wird die konzeptionelle Architektur des Anomaliedetektors für die vorliegende Fallstudie auf Basis der in Abschnitt 5.3.2 entwickelten Vorgehensweise abgeleitet. Dabei werden sowohl die Anforderungen als auch die Restriktionen der Fallstudie berücksichtigt.

6.1.3.1 Datenpipeline aufbauen

Im Folgenden wird die Datenpipeline mit dem Ziel aufgebaut, den durchgängigen Datenfluss von der Datenquelle bis zum Agenten innerhalb des Anomaliedetektors darzustellen (siehe Abbildung 6-7). Es wird auf die einzelnen funktionalen Komponenten des Anomaliedetektors eingegangen. Im ersten Schritt sollen die für diese Fallstudie identifizierten Datenquellen vorgestellt werden.

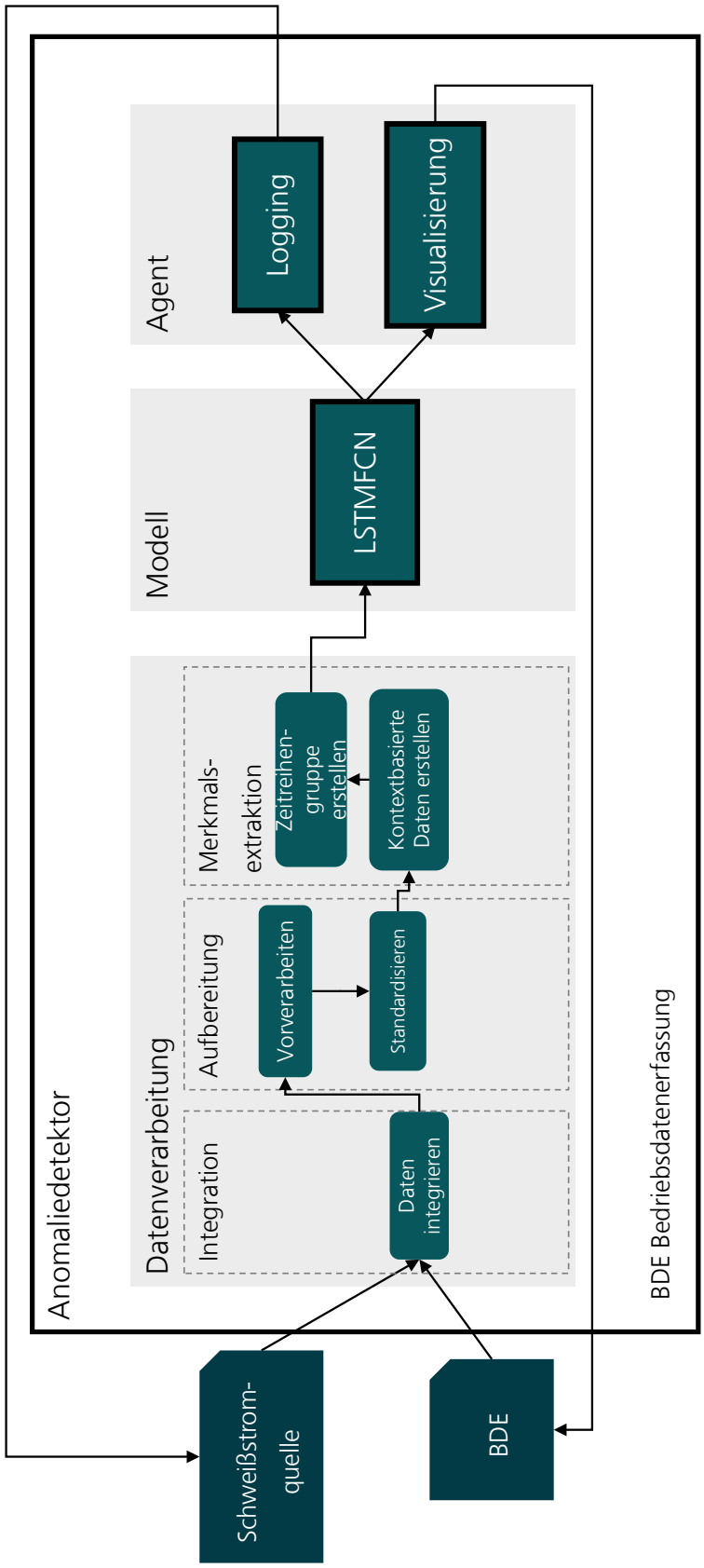


Abbildung 6-7 Konzept des Anomaliedetektors für die Erkennung von Bedienfehlern beim Lichtbogenschweißen

Wie beim PEMD-Framework in Abschnitt 6.1.2.1 beschrieben, werden zwei Datenquellen betrachtet: die Schweißstromquelle und die mobile Betriebsdatenerfassung in Form einer Smartphone-Applikation. Für beide wird die GET-Funktionalität beschrieben, die eine weitere Verarbeitung der Daten innerhalb des Anomaliedetektors ermöglicht.

Die Schweißstromquelle ermöglicht die Erfassung von Energiedaten wie Stromstärke und Spannung sowie die Messung des Drahtvorschubs. Die mit einer maximalen Abtastrate von 10 Hz erfassten Daten werden über ein CAN-Bus-System an einen Einplatinencomputer gesendet. Letzterer ermöglicht sowohl die Verarbeitung als auch die Analyse der Daten und übermittelt diese an ein zentrales Cloud-System des Schweißanlagenherstellers. Die Daten werden in einem vordefinierten JSON-Schema übertragen und können so mithilfe von Bibliotheken der Programmiersprache *Python* weiterverarbeitet werden. Die einzelnen Schweißungen in dieser Fallstudie werden als CSV-Datei gespeichert, da das Datenformat auch von anderen Programmiersprachen und Bibliotheken gelesen werden kann. Die Schweißstromquelle überträgt die Stromstärke, die Spannung und den Drahtvorschub als Zeitreihe. Der Anomaliedetektor kann diese Daten für die weiterführenden Prozesse zur Integration der unterschiedlichen Datenquellen nutzen

Die Komponente zum Import der Daten aus der Schweißstromquelle ist darauf ausgelegt, Daten aus einer CSV-Datei zu lesen. Diese Daten sind in einzelnen Zellen in einem bestimmten Format gespeichert, das in eine für die Zeitreihenanalyse geeignete Form umgewandelt werden muss.

Zunächst wird beim Initialisieren sichergestellt, dass die CSV-Datei existiert. Dann wird die Datei gelesen und bestimmte nicht benötigte Spalten werden gelöscht. Innerhalb der CSV-Datei können Daten für mehrere Maschinen vorhanden sein, die durch eine eindeutige Seriennummer gekennzeichnet sind. Der Code extrahiert eine Liste aller eindeutigen Maschinen.

Anschließend werden die Sensordaten für jede Schweißstromquelle in ein in Python verarbeitbares Datenformat transformiert. Dies geschieht, indem für jede Schweißstromanlage eine Teilmenge der Daten extrahiert und dabei jede Zeile der

Teilmenge in eine Zeitreihe umgewandelt wird. Diese Funktion extrahiert die Rohdaten aus der Zelle, teilt Erstere in separate Messungen auf und ordnet jeder Messung einen entsprechenden Zeitstempel zu.

Die mobile Betriebsdatenerfassung ermöglicht die Eingabe zusätzlicher Informationen, die einer Schweißung direkt zugeordnet werden können. Beispielhafte Informationen sind personenbezogene Daten des Schweißers, Bauteilinformationen, Material, Gasart, Drahtdurchmesser sowie weitere Betriebsdaten. Diese beinhalten auch die zuvor definierten Einstellparameter für den Schweißprozess. Die Daten werden über die mobile Betriebsdatenerfassung direkt in die lokale Datenbank des Anwenders geschrieben und dort gespeichert. Der Anomaliedetektor kann sie direkt über die Datenbank des Anwenders mit in die weiteren Komponenten zur Datenverarbeitung importieren bzw. dort einbeziehen. Die Übertragung der Daten aus der mobilen Betriebsdatenerfassung erfolgt durch das MQTT-Protokoll. Auch in diesem Fall werden die Daten im CSV-Format in den Anomaliedetektor integriert.

Die Daten aus beiden Datenquellen werden im ersten Schritt als zwei unterschiedliche Datensätze gespeichert. Diese, basierend auf multidimensionalen Arrays, ermöglichen eine einheitliche und niederschwellig bedienbare Verarbeitung der Daten.

Nach der Definition der Datenquellen sowie der Übertragung der beiden Datensätze aus der mobilen Betriebsdatenerfassung und der Schweißstromquelle erfolgt die Integration der Daten mithilfe der nächsten Datenverarbeitungseinheit, der *Datenintegration*. Da in beiden Datensätzen aus der Schweißstromquelle und der mobilen Betriebsdatenerfassung eine eindeutige ID vergeben wird, können die Daten mithilfe eines INNER JOIN integriert werden. Durch die Verwendung einer ID ist keine aufwendige Zeitsynchronisation wie bei der klassischen Signalverarbeitung notwendig. Die kontextbasierten Betriebsdaten und die multivariaten Energiezeitreihen, hier Stromstärke und Spannung, können somit integriert werden. Es wird in den integrierten Daten für jeden Zeitpunkt dargestellt, welche ID zu diesem Zeitpunkt geschweißt wurde. Dabei bezieht sich die ID auf die Schweißung und somit das gefertigte Produkt. Nach der Integration der Daten können die integrierten Daten einheitlich weiterverarbeitet werden. Dies erfolgt durch die Implementierung der Datenverarbeitungseinheiten der Datenaufbereitung.

Bei der Vorverarbeitung der Daten folgen zwei Schritte zur Bereinigung der integrierten Daten. Zunächst werden bestimmte Spalten entfernt, die ausschließlich NaN-Werte enthalten. Danach werden weitere spezifische Spalten entfernt, die einzigartige Werte enthalten und daher für die weitere Analyse irrelevant sind. Hierbei wird überprüft, welche Spalten der Daten keine Varianz aufweisen.

Da in Abschnitt 6.1.2.5 das Modell LSTMFCN ausgewählt wurde, wird keine Extraktion von statistischen Merkmalen benötigt, da für das ausgewählte Modell die Daten in Form von Zeitreihen erforderlich sind. Dennoch können mithilfe der bereitgestellten Skripte die prozessabhängigen Merkmale anhand der Daten aus der Schweißstromquelle erstellt werden. Mithilfe der eindeutigen ID lassen sich die einzelnen Schweißungen in die kontextbasierten Energiezeitreihengruppen transformieren. Diese kontextbasierten Energiezeitreihengruppen können nach der Merkmalsextraktion für das Training und somit für die FIT-Funktionalität genutzt werden.

Im Kontext der PREDICT-Funktionalität wird eine einzelne Schweißung anstelle der gesamten kontextbasierten Energiezeitreihengruppe betrachtet. Diese Schweißung durchläuft die gleiche Datenpipeline mit den gleichen Datenverarbeitungseinheiten, vergleichbar der FIT-Funktionalität. Entsprechend werden die Daten aus der mobilen Betriebsdatenerfassung und der Schweißstromquelle integriert und bereinigt. Dabei werden Spalten mit ausschließlich NaN-Werten sowie Spalten mit einer Varianz von 0 entfernt, da diese für die Vorhersage nicht relevant sind. Die bereinigte und vorverarbeitete Einzelschweißung wird dann zur Inferenz an das bereits trainierte Modell aus der FIT-Funktionalität übergeben. Der maßgebliche Unterschied zur FIT-Funktionalität besteht darin, dass nicht ein Modell auf einer Gruppe von Zeitreihen trainiert wird, sondern das bereits trainierte Modell zur Vorhersage einer einzelnen Schweißung dient.

6.1.3.2 Agent implementieren

In Anlehnung an Abschnitt 5.3.2.2 wird im Folgenden die praktische Umsetzung der Agenten und der zugehörigen Gegenmaßnahmen dargestellt. In der vorliegenden

Fallstudie werden zwei spezifische Agenten betrachtet, die unterschiedliche Formen der Interaktion mit dem System und dem Anwender bereitstellen.

Logging: Der erste Agent ist für die Protokollierung der Modellergebnisse in einem lokalen Skript verantwortlich, das auf dem Einplatinencomputer ausgeführt wird. Diese Form der Interaktion mit dem System ermöglicht eine direkte Überprüfung der Ergebnisse der energetischen Anomalieerkennung und kann helfen, schnell auf die Modellergebnisse zu reagieren. Die Implementierung dieses Agenten erfordert eine enge Integration mit der PREDICT-Funktionalität. Nach jeder Schweißung wird Letztere aufgerufen, um eine Vorhersage für die aktuelle Schweißung zu erhalten. Das Ergebnis des Modells wird mit Hilfe des Logging-Agenten direkt in der Kommandozeile dargestellt. Um dies zu ermöglichen, muss eine geeignete Schnittstelle für die Kommunikation zwischen dem Agenten und dem Modell implementiert werden – bspw. durch eine Funktion innerhalb des Python-Skripts, die das Modell aufruft und die Vorhersage zurückgibt. Darüber hinaus sollte der Agent in der Lage sein, die Ausgabe der Vorhersage auf der Konsole zu formatieren und zu protokollieren. Aus diesem Grund werden Tagesprotokolle aus dem Logging-Agenten erstellt, der die energetischen Anomalien innerhalb eines Tages zusammenfasst. Mit dem Tagesprotokoll können die energetischen Anomalien innerhalb des Tages im nächsten Schritt analysiert werden, um etwaige grundlegende Ursachen zu identifizieren. Dennoch bietet die Bereitstellung von Echtzeit-Feedback auch die Möglichkeit, die energetischen Anomalien direkt nach der Schweißung zu erkennen, um daraufhin die Ursache zu finden und diese zu beseitigen.

Dashboard: Der Dashboard-Agent arbeitet ähnlich wie der Logging-Agent. Der Hauptunterschied besteht in der Art der ausgegebenen Alarme. Statt die Ergebnisse einfach in der Konsole des Einplatinencomputers zu protokollieren, überträgt der Dashboard-Agent bei der Erkennung einer energetischen Anomalie die Informationen an das visuelle Dashboard des Anwenders. Der Dashboard-Agent dient zur visuellen Darstellung der Modellergebnisse in einem Dashboard, das dem Anwender zur Verfügung gestellt wird. Dieses Dashboard bietet bereits in der aktuellen Version eine intuitiv verständliche Möglichkeit zur Visualisierung der Sensordaten aus der Schweißstromquelle und der Betriebsdaten aus der mobilen Betriebsdatenerfassung. Durch den modularen

Aufbau des Dashboards kann ein weiterer Bereich zur Erkennung energetischer Anomalien implementiert werden. Dies erfordert eine geeignete Schnittstelle für die Kommunikation zwischen dem Agenten und dem Dashboard. In der Regel wird dies über die gleiche Schnittstelle realisiert, die für die Übertragung der bereits definierten Sensordaten zum Einsatz kommt. Somit ist es nicht notwendig, zusätzliche Schnittstellen zu entwickeln. Hierbei kann das Ergebnis der Inferenz durch das Modell als zusätzlicher Datenpunkt in die Übertragung der Sensordaten vom Einplatinencomputer integriert werden. Dazu muss das JSON-Schema zur Übertragung der Sensordaten in die lokale Datenbank des Anwenders um das Datenfeld „Anomalie“ erweitert werden.

6.1.3.3 Anomaliedetektor bereitstellen

Für die Bereitstellung des Anomaliedetektors wurden zwei verschiedene Optionen eingehend untersucht und diskutiert. In einem Workshop mit Fachexperten des Schweißanlagenherstellers wurde zunächst die Möglichkeit erwogen, einen Minicomputer direkt in die Schweißstromquelle zu integrieren. Dieser Minicomputer hätte das Training und die Inferenz (FIT und PREDICT) direkt in der Schweißstromquelle durchführen können. Bei diesem Ansatz wären jedoch verschiedene Herausforderungen aufgetreten. Eine davon wäre der erhöhte Aufwand für die Beschaffung zusätzlicher Hardware gewesen. Durch die Wärmeentwicklung in der Schweißstromquelle und die geringe Verfügbarkeit von Bauraum wäre der Einbau größerer Rechenkapazitäten nur deutlich eingeschränkt möglich gewesen. Darüber hinaus hätte die Implementierung dieser Lösung aufgrund von Inkompatibilitäten zwischen den Minicomputern und bestimmten Python-Bibliotheken einen enormen Arbeitsaufwand erfordert. Außerdem hätten die technischen Spezifikationen der Minicomputer für ein schnelles Training nicht ausgereicht.

Aufgrund dieser Faktoren wurde schließlich entschieden, den Anomaliedetektor auf einem Einplatinencomputer als Python-Skript zur Verfügung zu stellen. Bei dieser Entscheidung spielten mehrere Überlegungen eine Rolle. Erstens bot die Verwendung

eines Einplatinencomputers größere Flexibilität in Bezug auf die Hardwareanforderungen: Die technischen Spezifikationen eines solchen Computers können in der Regel an die Trainings- und Inferenzanforderungen angepasst werden. Zweitens sind Einplatinencomputer in der Regel mit zahlreichen Python-Bibliotheken kompatibel, was die Implementierung erheblich erleichtert. Darüber hinaus reduziert die hardwarenahe Bereitstellung des Anomaliedetektors als Python-Skript den Aufwand für die Integration in die bestehende Infrastruktur.

6.1.4 Parameteroptimierung des föderierten Lernens

Dieser Abschnitt beschreibt die dritte Phase zur Auslegung des FL-basierten Systems zur energetischen Anomalieerkennung für die vorliegende Fallstudie auf Basis der in Abschnitt 5.3.3 entwickelten Vorgehensweise.

6.1.4.1 Konfigurationsparameter definieren

Zunächst sollen in diesem Abschnitt die Konfigurationsparameter für die Durchführung der Simulation definiert werden. Diese Parameter werden auf Grundlage von Abschnitt 5.3.3.1 gewählt. Dabei fließen zusätzlich die Ergebnisse aus Phase I mit ein. In Tabelle 4 sind die gewählten Konfigurationsparameter aufgelistet.

Tabelle 4 Definition der Konfigurationsparameter für die Simulation von FL

Parameter	Beschreibung	Wert
Aggregations-mechanismus	Bestimmt, wie die Modelle der Clients aggregiert werden	FedAvg FedOpt FedAdam FaultTolerantFedAvg QFedAvg
Anzahl der Trainingsrunden	Anzahl der Durchführungen des föderierten Lernprozesses	100
Client-Anzahl	Anzahl der am Lernprozess beteiligten Clients	[2, 8]
Lokale Epochen	Anzahl der Trainingsschleifen auf jedem Client	[1, 20]
Batch-Größe	Anzahl der Datenpunkte, die in jedem Schritt verwendet werden	[16, 128]

6.1.4.2 Simulationsbasierte Experimente durchführen

Für die Simulations- und Optimierungsberechnungen wird eine *Dell-Precision-7920*-Workstation verwendet. Diese verfügt über zwei Prozessoren vom Typ Intel(R) Xeon(R) Platinum 8268 mit einer Taktfrequenz von 2,90 GHz. Für die Simulation wurden 32 Cores gewählt, damit die volle Kapazität ausgenutzt werden kann und jedem Segment ausreichend Kerne zur Verfügung stehen. Zusätzlich können 512 GB Arbeitsspeicher verwendet werden. Der Simulations- und Optimierungscode wird mit Python 3.9 interpretiert. Für die Simulation von FL wird das Framework *Flower* verwendet, das auf Basis von Ray die Segmente auf die verschiedenen Kerne des Prozessors verteilt, um das lokale Training zu simulieren. Es wurden einige Anpassungen am Code vorgenommen, um eine einfache, benutzerfreundliche Simulation zu ermöglichen. So ist die Simulation

über einen Docker-Container und eine Konfigurationsdatei und unabhängig vom Rechensystem ausführbar.

Für die Simulation werden im ersten Schritt die Trainings- und Testdaten eingesetzt, um eine konsistente und gleichbleibende Trennung der Daten zu gewährleisten. Dabei werden diese Testdaten als der globale Testdatensatz definiert und die Trainingsdaten zufällig in zehn gleiche Segmente aufgeteilt. Somit werden für die Trainingsdaten 1980 Schweißungen und für den Testdatensatz 220 Schweißungen verwendet. Innerhalb der Segmente wird explizit nicht auf eine Gleichverteilung zwischen guten und anomalen Schweißungen geachtet, um eine gewisse Heterogenität in den einzelnen Segmenten zu ermöglichen. Innerhalb der einzelnen Segmente werden für den lokalen Testdatensatz 10 % der Segmente zufällig ausgewählt, sodass jedes einzelne Segment nach dem lokalen Training mit den rein lokalen Daten getestet werden kann.

Für die Optimierung der Hyperparameter wurden fünfzig Runden festgelegt (siehe Abschnitt 5.3.3.2). Es werden ebenso viele unterschiedliche Konfigurationen ausgeführt, wobei die in Abschnitt 5.3.3.2 definierte kombinierte Evaluationsmetrik als Grundlage für die Anpassung der Konfigurationsparameter nach jeder Runde fungiert. Insgesamt wurden somit 25 000 lokale Modelle trainiert und 5650 Validierungsschritte auf dem globalen Testdatensatz ausgeführt. Die Ergebnisse wurden für die spätere Analyse und Bewertung der Konfigurationsparameter (siehe Abschnitt 5.3.3.3) in eine Datenbank geschrieben.

Nach der Simulation und Optimierung werden in einer separaten CSV-Datei das Ergebnis der Optimierung und die optimalen Konfigurationsparameter gespeichert. Für diese Fallstudie sind die optimalen Konfigurationsparameter Tabelle 5 zu entnehmen.

Tabelle 5 Ergebnis der bayesschen Optimierung mit dem Tree-Parzen Estimator

Parameter	Parameterraum	Optimaler Wert
Aggregations-mechanismus	FedAvg FedOpt FedAdam FaultTolerantFedAvg QFedAvg	FaultTolerantFedAvg
Client-Anzahl	[2, 8]	4
Lokale Epochen	[1, 20]	8
Batch-Größe	[16, 128]	32

Mit den in Tabelle 5 ermittelten Konfigurationsparametern wird ein F1-Gesamtscore von 0,81 erreicht – eine sehr gute Annäherung an den F1-Score von 0,86 in der initialen Entwicklung des Modells in Abschnitt 6.1.2. Für den Vergleich des zentralen Trainings mit dem dezentralen FL-Training wurde die in Abschnitt 5.3.3.2 beschriebene Methode angewandt, um die Verbesserung des F1-Scores auf Grundlage der lokalen F1-Scores zu ermitteln. Abbildung 6-8 zeigt dazu ein Balkendiagramm, das die lokalen F1-Scores der einzelnen Clients im dezentralen Training, den globalen F1-Score auf Basis der ermittelten optimalen Konfigurationsparameter von FL sowie das zentrale Training einander gegenüberstellt. Es ist deutlich zu erkennen, dass sich der Ansatz mit FL (hellblau) dem zentralen Training stark annähert. Für die Datenpartitionen der Clients liegt der durchschnittliche F1-Score bei ca. 0,45 und damit deutlich niedriger als bei FL oder dem zentralen Training des Modells. Für die Erkennung von Bedienfehlern beim Lichtbogenschweißen ist somit FL dem rein dezentralen Training vorzuziehen. Dies ermöglicht eine im Mittel bessere energetische Anomalieerkennung.

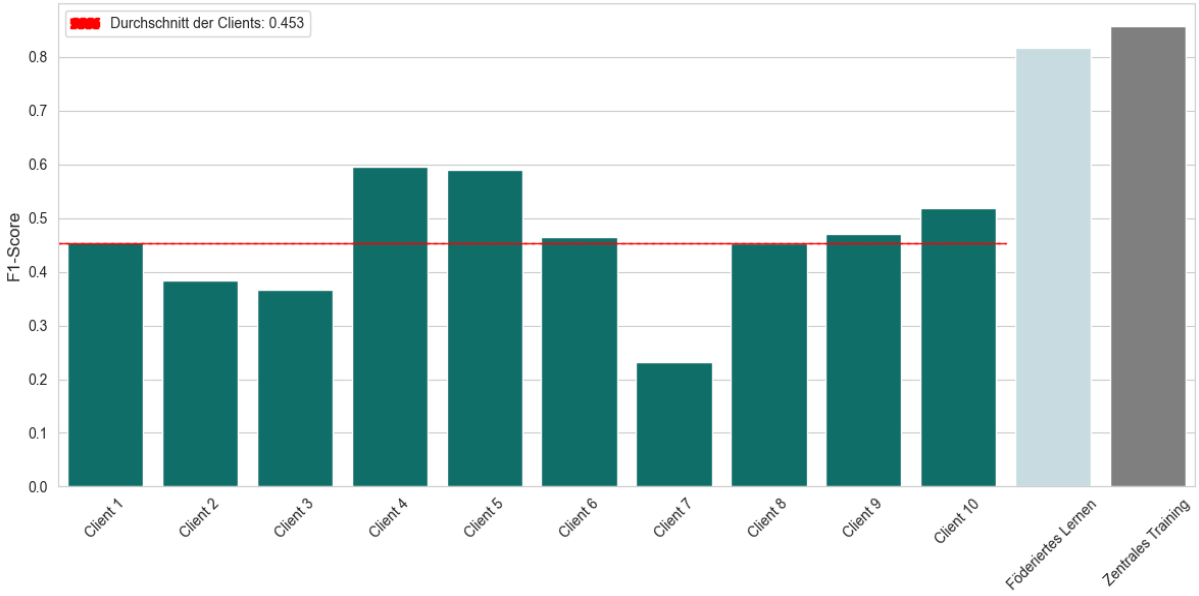


Abbildung 6-8 Darstellung des Vergleichs der Evaluationsmetrik zwischen dezentralem, zentralem Training und FL

Aus Abbildung 6-9 kann die durchschnittliche Entwicklung des F1-Scores und der Fairness entnommen werden. Auf der linken Y-Achse ist der F1-Score (grün) und auf der rechten Y-Achse die Fairness als Standardabweichung der F1-Scores aller lokalen Clients aufgetragen. Hierbei ist zu beachten, dass eine niedrige Standardabweichung einer besseren Fairness entspricht. Der F1-Score steigt stetig von einem Anfangswert von ca. 0,29 in der ersten Runde nach einem kurzen Absenken. In den ersten Runden zeigt der F1-Score deutliche Schwankungen. Diese können mit den typischen Verlustschwankungen in den frühen Stadien des Trainings von KNN verglichen werden. In der letzten Runde erreicht der F1-Score einen Wert von ca. 0,70, was eine Verbesserung gegenüber dem Ausgangswert darstellt.

Abbildung 6-9 zeigt ferner die Fairness von FL über die Runden hinweg. Der Trend geht dahin, dass die Fairness im Allgemeinen zunimmt, dargestellt durch eine Verringerung der Standardabweichung zwischen den Clients. Dies indiziert, dass FL über die Runden hinweg fairer wird und die Modelle der verschiedenen Clients einen relativ ähnlichen F1-Score aufweisen. Die Fairness erreicht in der 100. Runde einen Wert von ca. 0,14, was darauf hindeutet, dass die Unterschiede zwischen den Clients im Laufe der Zeit geringer werden.

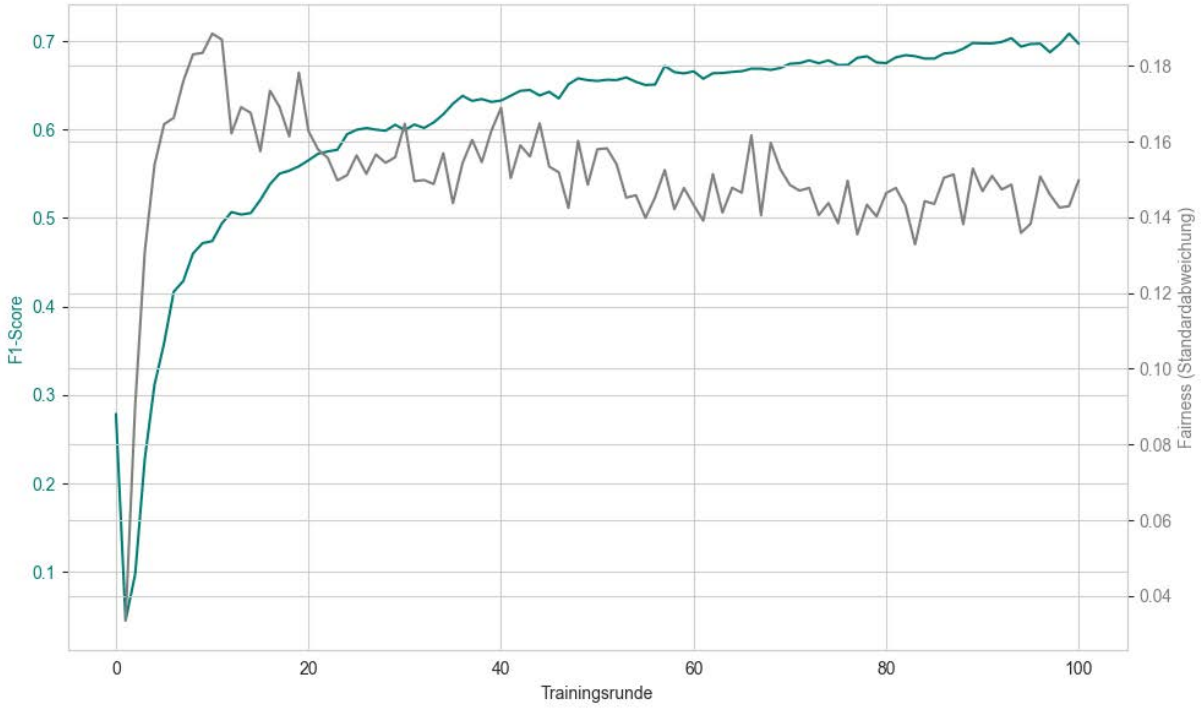


Abbildung 6-9 Durchschnittliche Entwicklung des F1-Scores und Fairness pro Runde

Abbildung 6-10 zeigt die Verteilung der F1-Scores, wobei die Werte auf den letzten zehn Werten (10 %) jeder Konfiguration basieren und der gleitende Mittelwert berechnet wurde (siehe Abschnitt 5.3.3.2). Für eine bessere Veranschaulichung wird die Kernel-Dichte-Schätzung genutzt, die die beobachteten Werte in einer kontinuierlichen Wahrscheinlichkeitsdichtekurve darstellt. Die X-Achse stellt die verschiedenen F1-Scores dar, während die Y-Achse die Anzahl der Konfigurationen zeigt, die jeweils einen bestimmten F1-Score erreicht haben.

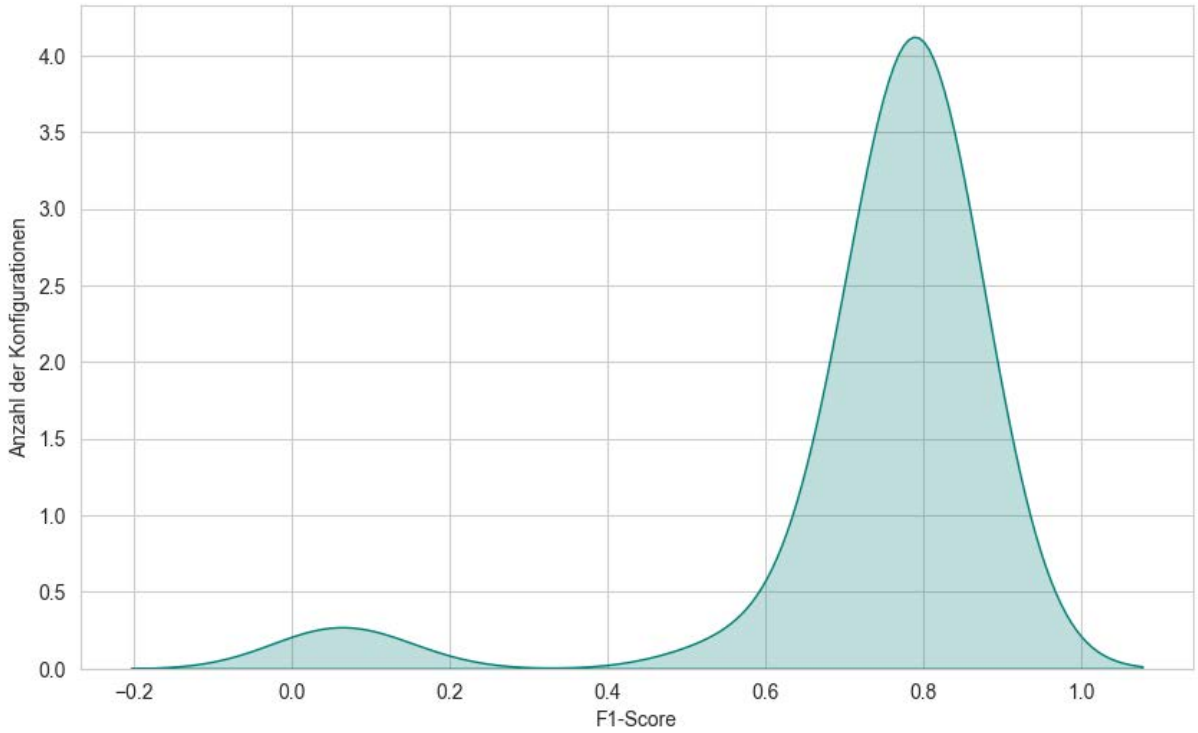


Abbildung 6-10 Verteilung des F1-Scores

Die visuelle Analyse zeigt, dass die meisten Konfigurationen einen hohen F1-Score zwischen 0,7 und 0,85 erreichen. Dies legt nahe, dass die Mehrheit der Konfigurationen einen ähnlichen F1-Score aufweist.

Es sind jedoch auch mehrere Konfigurationen mit einem deutlich niedrigeren F1-Score zu beobachten. Dies könnte auf suboptimale Konfigurationsparameterkombinationen hinweisen, die mit FL oder den verwendeten Daten nicht produktiv zusammenpassen. Besonders hervorzuheben sind die Ausnahmen mit niedrigen F1-Scores, die nahe 0,01 liegen. Sie könnten auf Konfigurationen hinweisen, die während des Trainingsprozesses nicht konvergieren und zu niedrigen F1-Scores führten. Auffällig ist, dass Konfigurationen mit niedrigem F1-Score meist den gleichen Aggregationsmechanismus – FedAdam – aufweisen.

Die meisten Konfigurationen führen zu F1-Scores, die in einem definierten Wertebereich mit nicht mehr als einem Hochpunkt liegen. Dennoch konnten schon in einer ersten Beobachtung einige Auffälligkeiten im F1-Score identifiziert werden. Die Einflüsse und Zusammenhänge der einzelnen Konfigurationsparameter auf den F1-Score und die Fairness sollen im nächsten Abschnitt weiter analysiert werden.

6.1.4.3 Analyse und Bewertung der Konfigurationsparameter

Abbildung 6-11 zeigt den Verlauf des mittleren F1-Scores in Abhängigkeit vom Aggregationsmechanismus. Die verschiedenen Aggregationsmechanismen sind farblich unterschiedlich gekennzeichnet.

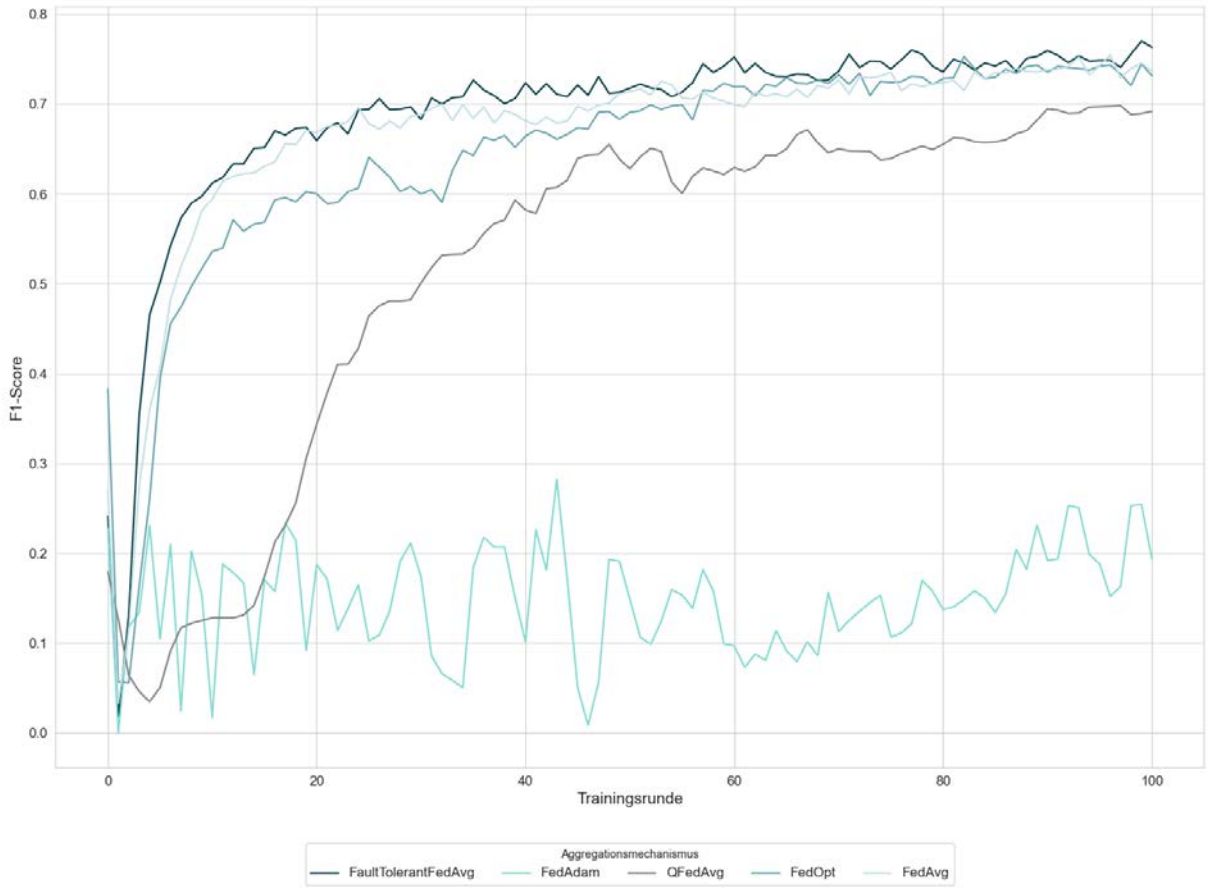


Abbildung 6-11 Mittelwert der F1-Scores für jeden Aggregationsmechanismus

Der Aggregationsmechanismus FedAdam weist im Vergleich zu den anderen Aggregationsmechanismen durchgehend einen niedrigen F1-Score auf. Außerdem zeigt FedAdam eine deutlich höhere Volatilität, insbesondere in den ersten Trainingsrunden. Es gibt mehrere Einbrüche auf signifikant niedrige Werte. Es gibt jedoch auch einige Hochpunkte im Verlauf, an denen sich der F1-Score vorübergehend verbessert. Bei allen anderen Aggregationsmechanismen zeigt sich ein stetiger Anstieg des F1-Scores ohne signifikante Einbrüche in der Mitte oder am Ende des Trainings. FaultTolerantFedAvg, FedAvg und FedOpt demonstrieren einen stetigen Anstieg des F1-Scores. Bei QFedAvg ist

zu Beginn ein klarer Abwärtstrend erkennbar. Im Vergleich zu den anderen Aggregationsmechanismen jedoch konvergiert QFedAvg ohne signifikante Schwankungen und Volatilität.

Abbildung 6-12 zeigt, dass die meisten Strategien zwar in Richtung eines Leistungsoptimums konvergieren, die Geschwindigkeit und Stabilität dieser Konvergenz aber je nach Ansatz variiert. Daher ist eine Bewertung und Optimierung immer in Verbindung mit den anderen Konfigurationsparametern durchzuführen.

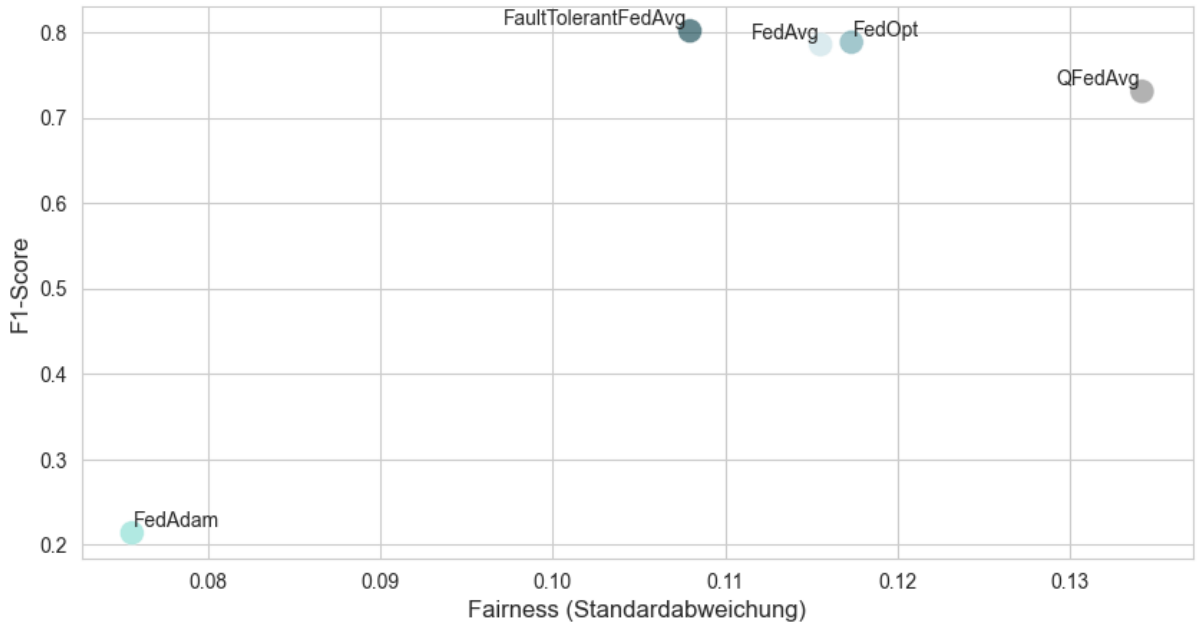


Abbildung 6-12 Darstellung der einzelnen Strategien in Relation zu F1-Score und Fairness

Werden $\overline{F_R}$ und $\overline{Fair_R}$ mit $R = 10$ zusammenhängend betrachtet (siehe Abbildung XY2), so fällt auf, dass sich alle vier genannten Aggregationsmechanismen, die stetig in Richtung eines Leistungsoptimums hinsichtlich des F1-Scores ansteigen, im oberen rechten Quadranten der Abbildung befinden. Allerdings geht dieser hohe F1-Score zu Lasten der Fairness, wie die hohe Standardabweichung zeigt. Eine größere Standardabweichung deutet in diesem Zusammenhang auf größere Unterschiede im F1-Score zwischen den verschiedenen Clients hin.

Im Gegensatz dazu befindet sich FedAdam im unteren linken Quadranten der Grafik. Während es deutlich niedrigere Werte für den F1-Score aufweist, was auf eine geringere Vorhersagekraft hindeutet, zeigt es auch eine geringere Standardabweichung. Dies

bedeutet, dass FedAdam zwar möglicherweise nicht die höchsten F1-Scores erzielt, aber über die verschiedenen Clients hinweg konsistenter ist und keine großen Unterschiede in den F1-Scores der verschiedenen Clients aufweist. Im vorliegenden Anwendungsfall ist dies besonders kritisch, da fehlerhafte Schweißungen nicht nur zu höheren Kosten, sondern auch zu Sicherheitsrisiken durch fehlerhafte Schweißverbindungen führen können. Wenn ein höherer F1-Score erreicht werden soll, kann dies daher bedeuten, dass das Modell insgesamt effektiver bei der Erkennung dieser kritischen Fehler ist, wenngleich zu Lasten der Fairness.

Abbildung 6-13 visualisiert den F1-Score und die Fairness in Form der Standardabweichung der F1-Scores der Clients in Abhängigkeit von der Anzahl der lokalen Epochen. Der F1-Score sowie die Fairness zeigen bei der lokalen Epochenanzahl ein volatiles Verhalten. Das Minimum des F1-Scores liegt bei ca. 0,55 (für vier Epochen), das Maximum bei ca. 0,82 (für 19 Epochen). Der F1-Score indiziert einen deutlichen Anstieg in der fünften Epoche auf 0,77 und bleibt in den folgenden Epochen häufig über 0,7. Die Fairness schwankt zwischen 0,08 (bei 17 Epochen) und 0,17 (bei 5 Epochen). Es gibt keine klare Beziehung zwischen der Zunahme des F1-Scores und der Fairness. In einigen Fällen nehmen beide zu; in anderen Fällen steigt einer und der andere nimmt ab.

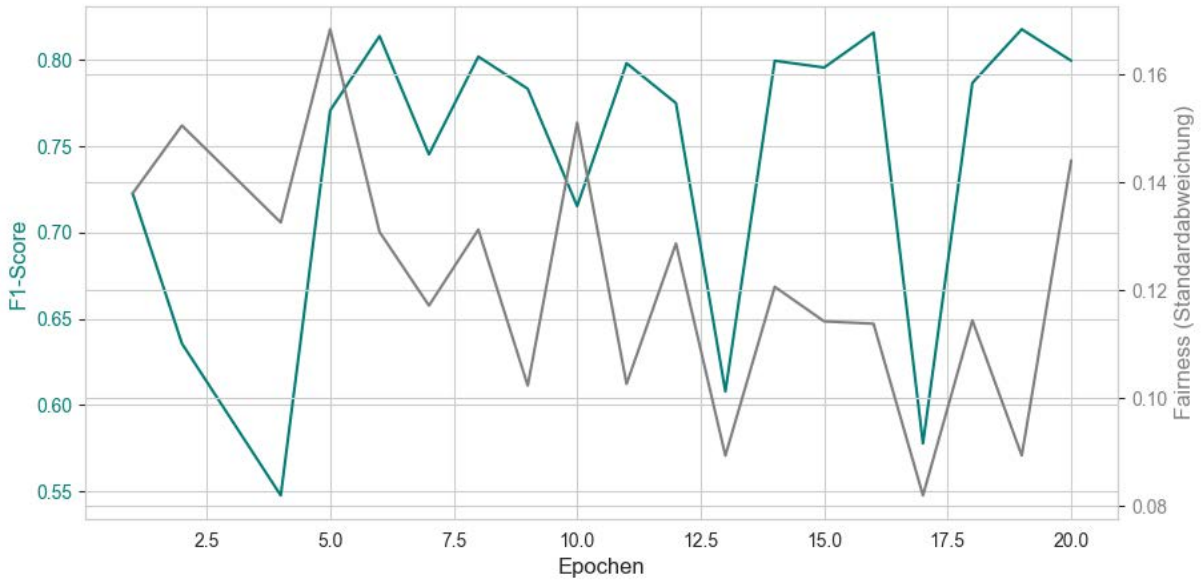


Abbildung 6-13 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit der Epochen

Die Anzahl der lokalen Epochen hat offenbar keinen Einfluss auf den F1-Score. Das Training konvergiert nach einer bestimmten Anzahl lokaler Epochen und zusätzliche Epochen bringen somit nur marginale Verbesserungen oder gar Verschlechterungen. Die Daten zeigen, dass es mit einer Vielfalt von Werten für die lokalen Epochen möglich ist, einen hohen F1-Score zu erreichen, was die Wahl des optimalen Wertes flexibler macht. Die Variation des F1-Scores und der Fairness deutet auf einen Einfluss von möglicherweise anderen Faktoren als die Anzahl der Epochen hin. In diesem speziellen Fall sind die lokalen Epochen weniger bedeutsam als erwartet.

Abbildung 6-14 zeigt den Verlauf des durchschnittlichen F1-Scores und der Fairness in Abhängigkeit von der Batch-Größe. Die Daten umfassen die Batch-Größen, beginnend mit 16 und endend mit 128, mit inkrementellen Erhöhungen. Der F1-Score variiert über die verschiedenen Batch-Größen, mit dem höchsten Wert bei einer Batch-Größe von 32 und dem niedrigsten bei 16. Die Standardabweichung, die als Maß für die Fairness zwischen den Clients verwendet wird, variiert ebenfalls, bleibt aber im Allgemeinen in einem engen Bereich.

Es gibt keinen linearen Trend zwischen der Batch-Größe und dem F1-Score. Während der F1-Score bei einer Batch-Größe von 32 am höchsten ist, gerät er bei den benachbarten Werten 16 und 48 niedriger. Es gibt also keinen stetigen Anstieg oder Abfall des F1-Scores mit zunehmender Batch-Größe.

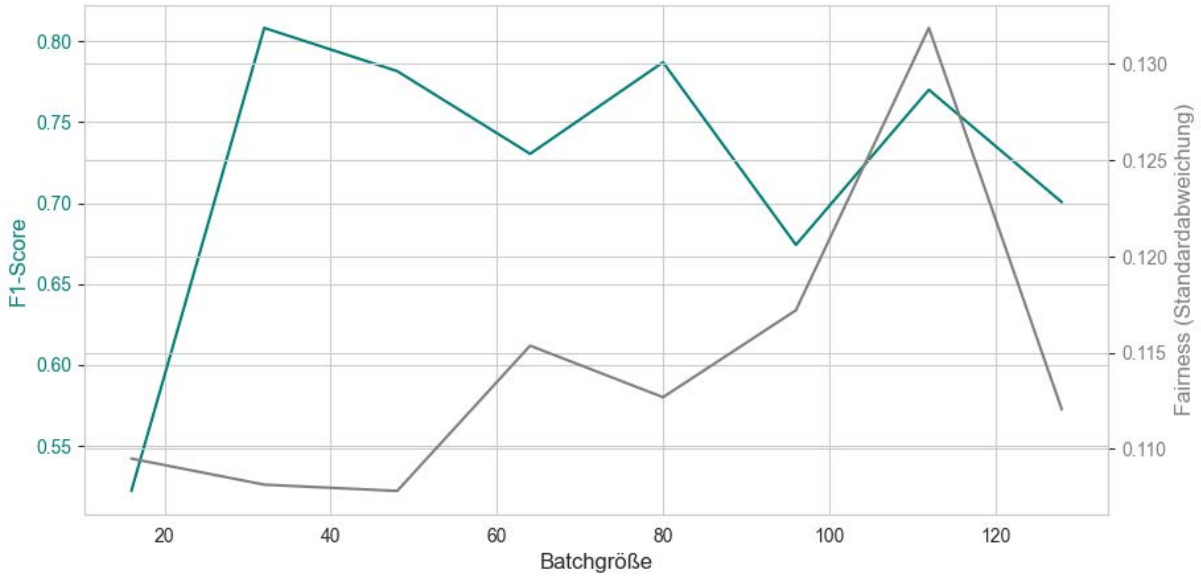


Abbildung 6-14 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit von der Batchgröße

Die Fairness (gemessen als Standardabweichung) zeigt einen leichten Anstieg der Standardabweichung bis zu einer Batch-Größe von 112. Dieser Trend setzt sich nicht fort, wenn die Batch-Größe auf 128 erhöht wird.

Bei FL scheint die Wahl der Batch-Größe Einfluss auf den F1-Score zu haben, wobei größere Batches nicht unbedingt zu besseren Ergebnissen führen. Dies könnte darauf hindeuten, dass die optimale Batch-Größe je nach Daten, Netzwerkarchitektur oder anderen Faktoren variieren kann. Die Fairness zwischen den Clients, gemessen an der Standardabweichung des F1-Scores, zeigt insgesamt nur geringe Schwankungen über die untersuchten Batch-Größen. Die Batch-Größe ist in dieser Fallstudie nicht der beeinflussende Faktor für die Fairness zwischen den Clients. Das unerwartet niedrige Ergebnis bei einer Batch-Größe von 128 weist möglicherweise auf Probleme beim Training oder der Konvergenz hin. Auch könnte eine Überanpassung der lokalen Modellparameter durch eine höhere Batch-Größe verursacht werden.

Die Entwicklung des durchschnittlichen F1-Scores in Abhängigkeit vom Trainingsanteil der beteiligten Clients ist in Abbildung 6-15 dargestellt. Der F1-Score beträgt bei einem Anteil von 20 % – was zwei Clients entspricht – 0,56, also deutlich niedriger als bei den anderen Anteilen. Bei einer Erhöhung des Anteils auf 40 % (vier Clients) steigt der F1-Score

merklich an: Er erreicht einen Wert von 0,80. Danach gibt es Schwankungen, aber die Werte bleiben vergleichsweise hoch. Die Fairness, gemessen als Standardabweichung, steigt zunächst von 20 % auf 30 % Beteiligung an, bleibt dann bis 40 % relativ konstant und zeigt danach Schwankungen ohne klaren Trend.

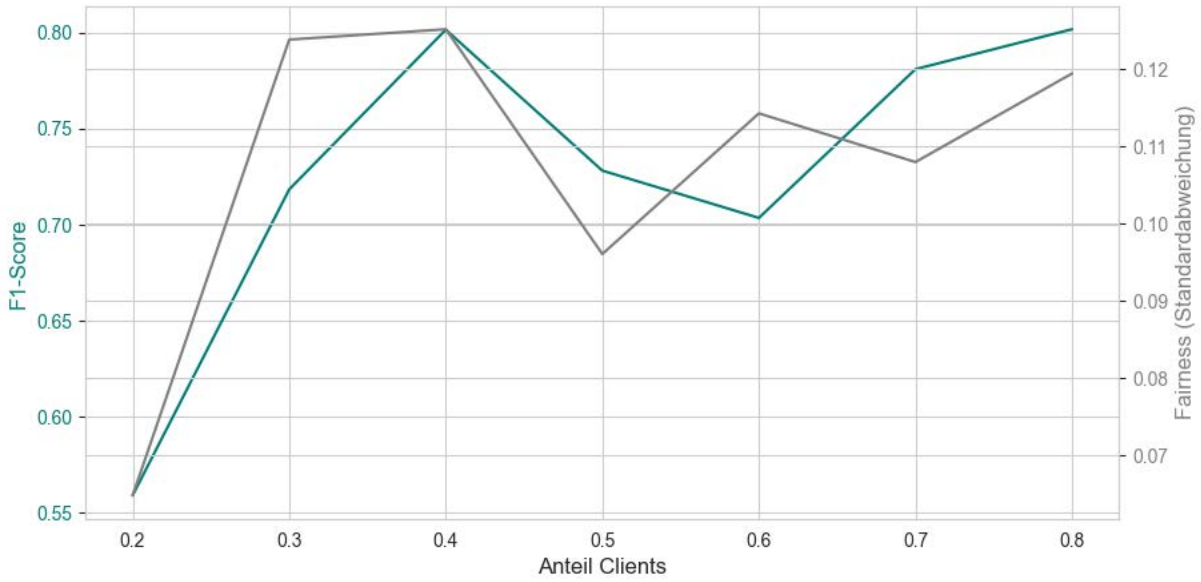


Abbildung 6-15 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit des Anteils der Clients

Die Beteiligung von mehr Clients in den frühen Phasen (bis zu 40 %) führt zu einem deutlichen Anstieg des F1-Scores, was auf eine Verbesserung des Modells hindeutet.

Der Anstieg der Standardabweichung zwischen 20 und 30 % Beteiligung legt eine Abnahme der Fairness nahe, da mehr Clients größere Unterschiede in ihren individuellen Beiträgen zum Gesamtmodell haben könnten. Ab einer Beteiligung von 40 % bleibt die Fairness jedoch stabil. Es scheint, dass die Erhöhung der Beteiligung der Clients bis zu einem bestimmten Punkt (hier: 40 %) zu einem besseren F1-Score des Modells führt. Dies könnte darauf zurückzuführen sein, dass das Modell mit mehr oder unterschiedlicheren Daten verschiedener Clients generischer und leistungsfähiger wird.

Die Fairness zwischen den Clients wird jedoch beeinträchtigt, wenn der Anteil der Beteiligung steigt, insbesondere zwischen 20 und 30 %. Es ist denkbar, dass einige Kunden über Daten verfügen, die sich stark von denen der anderen unterscheiden. Wenn diese einen signifikanten Anteil beitragen, verschlechtert sich die Fairness. Oberhalb von

40 % Beteiligung bleiben sowohl der F1-Score als auch die Fairness stabil, was darauf hindeuten könnte, dass das Hinzufügen weiterer Clients über diesen Punkt hinaus nicht mehr zu signifikanten Veränderungen im Modell führt.

Ein weiterer wichtiger Gesichtspunkt ist der gesamte Rechenaufwand aller lokalen Schweißstromquellen. Ein höherer Anteil an Clients, die am Training beteiligt sind, bedeutet, dass mehr Schweißstromquellen gleichzeitig trainieren müssen. Der Koordinationsaufwand des Modelkoordinators kann dabei jedoch vernachlässigt werden, da die Aggregation von mehr Werten die Rechenzeit nicht signifikant steigert. Dies könnte zu höheren Kosten, vermehrtem Energieverbrauch und anderen Herausforderungen führen, etwa mit Blick auf die Konnektivität durch den erhöhten Kommunikationsaufwand. Daher könnte es sinnvoll sein, den Anteil der Clients zu reduzieren, bspw. auf 40 % (Ergebnis der Optimierung), um einen Kompromiss zwischen Evaluationsmetrik, Fairness und Energieverbrauch zu finden. Ein solcher Ansatz würde den Gesamtaufwand und die Komplexität des Trainingsprozesses reduzieren, während durchgehend ein hoher F1-Score beibehalten wird.

Im nächsten Schritt wurde mithilfe des Pearson-Korrelationskoeffizienten untersucht, inwiefern ein linearer Zusammenhang zwischen den verschiedenen Konfigurationsparametern und dem F1-Score besteht. Die Koeffizienten werden in einer Matrix dargestellt, um anschaulich die Zusammenhänge zwischen den Parametern und Metriken darzustellen (siehe Abbildung 6-16).

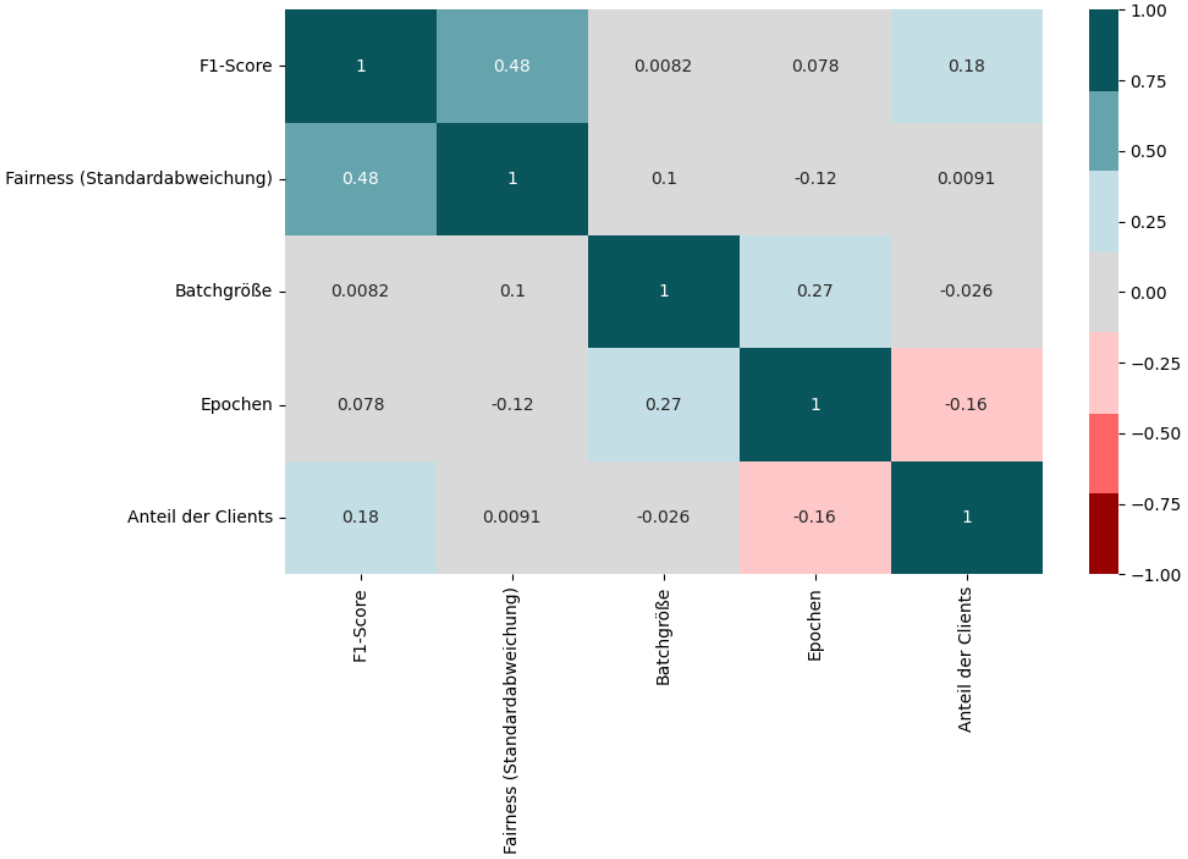


Abbildung 6-16 Korrelationsmatrix dargestellt als Heatmap der Konfigurationsparameter und der definierten Leistungsmetriken

Eine zentrale Beobachtung ist die positive Korrelation von 0,48 zwischen dem F1-Score und der Fairness. Bei genauerer Betrachtung bedeutet dies, dass eine Erhöhung des F1-Scores mit einer Erhöhung der Standardabweichung und insofern mit einer Verringerung der Fairness korreliert. Das Modell wird insgesamt genauer, die Standardabweichung des F1-Scores zwischen den verschiedenen Clients nimmt jedoch ab. Dies ist ein kritischer Punkt – insbesondere für diesen Anwendungsfall, in dem sowohl der allgemeine F1-Score als auch die Standardabweichung der lokalen F1-Scores wichtig sind – und zeigt erneut den Konflikt zwischen Fairness und F1-Score. Für den Anlagenbauer der Schweißstromquelle ist es daher von Interesse, einen guten Kompromiss zwischen Fairness und F1-Score zu finden, der durch die kombinierte Metrik bei der Auswahl der Konfigurationsparameter (siehe Abschnitt 6.1.4.2) genutzt wurde.

In Bezug auf die Konfigurationsparameter zeigt der F1-Score eine positive Korrelation von 0,18 mit dem Anteil der am Training beteiligten Clients. Möglicherweise deutet dies

darauf hin, dass ein größerer Anteil an Clients tendenziell zu einer besseren Konvergenz des globalen Modells führt. Dennoch erkennt man anhand von der Korrelationsmatrix, dass keine klaren linearen Zusammenhänge zwischen den Konfigurationsparametern und dem F1-Score oder der Fairness existieren.

6.2 Erkennung von Druckluftleckagen

Für die zweite Fallstudie wurde die energetische Anomalieerkennung bei pneumatischen Aktorbewegungen betrachtet, da aufgrund von Leckagen im Druckluftsystem viel Energie und Kosten eingespart werden können. Insbesondere liegt die Herausforderung bei der hinreichenden Erkennung der Druckluftleckagen nur unter Einbezug eines Merkmals: des Massenstroms. Mithilfe eines Labordemonstrators konnten hierfür reale Daten erfasst und aufbereitet werden, die für die Entwicklung des Modells, die Implementierung des Anomaliedetektors und die Parameteroptimierung von FL genutzt werden konnten.

6.2.1 Druckluftleckagen in Produktionsprozessen

Druckluft ist eine der wichtigsten Formen der Nutzenergie in Produktionsanlagen. Sie ist allerdings auch kostspielig (Bierbaum 2011). Druckluft wird in zahlreichen Prozessen eingesetzt, angefangen vom Antreiben von Druckluftwerkzeugen und -motoren über das Transportieren in pneumatischen Förderanlagen bis hin zum Spannen, Klemmen und Pressen in pneumatischen Haltevorrichtungen oder Pressen. Obwohl Druckluft in Produktionsprozessen unverzichtbar ist, wird oft vernachlässigt, wie teuer ihre Erzeugung sein kann resp. welche wirtschaftlichen und energetischen Einsparpotenziale in ineffizienten Druckluftsystemen möglich sind.

In Deutschland werden 7 % des industriellen Stromverbrauchs auf Drucklufterzeugung verwendet (IHK Nürnberg 2012). Dieser Anteil ist beträchtlich, da es sich dabei um Milliarden von Kilowattstunden handelt. Trotz ihrer Bedeutung wird die

Druckluftversorgung oft als zweitrangig betrachtet und es wird zu wenig in ihre effiziente Nutzung investiert.

Die Ineffizienz von Druckluftsystemen stellt in vielen Unternehmen ein großes Problem dar. Die Studie des Umweltbundesamtes verdeutlicht, dass das Aufspüren und Beseitigen von Druckluftleckagen das größte Einsparpotenzial bietet (Radermacher et al. 2021, S. 68). Obwohl Leckagen in Druckluftsystemen als unvermeidbar gelten, weist ein gut gewartetes System Leckage-Anteile von unter 10 % auf. Die Autoren schätzen jedoch, dass die durchschnittliche Leckage-Rate bei 20 bis 30 % liegt, was die Effizienz erheblich reduziert. Das Effizienzpotenzial wird mit 10 bis 20 % bei einer hohen Anwendbarkeit von 60 bis 80 %, das Einsparpotenzial indes mit 9 bis 16 % angegeben.

Leckagen sind nicht nur Energieverschwender, sondern auch kostspielig. Schätzungen zufolge liegen die Leckage-Raten bezogen auf die jährlichen Druckluftverbräuche in Unternehmen bei 30 bis 50 %. Um die wirtschaftlichen Auswirkungen in eine Perspektive zu rücken: Bei einem Relativdruck von 8 bar kann eine Leckage von 1 mm Durchmesser etwa 4,5 m³/h Druckluft entweichen lassen. Dies entspricht einem Energieaufwand von 0,6 kW bzw. jährlichen Mehrkosten von etwa 480 EUR (Bierbaum 2011).

In Anwendungen, die als Energieträger Druckluft nutzen, ist fast die Hälfte der Verluste durch gezielte Leckage-Beseitigung vermeidbar, was die Notwendigkeit für die Erkennung und Beseitigung der Leckagen unterstreicht. In Produktionsprozessen treten diverse Fehlzustände an pneumatischen Aktoren auf, darunter undichte Verbindungen, Löcher oder Knicke in Schläuchen (Dierolf 2022). Obwohl zahlreiche Interventionsmaßnahmen zur Behebung solcher Verluste existieren, stellt das automatisierte Erkennen von Leckagen eine technologische Herausforderung mit beträchtlichem wirtschaftlichen Mehrwert dar (Dierolf & Sauer 2021). Das automatisierte und messtechnisch minimierte Erkennen von Leckagen, aus denen Druckluft im Betrieb austritt, bleibt somit eine herausfordernde Aufgabe.

Der Einsatz von ML-Methoden zur Erkennung von Leckagen in Druckluftanwendungen gewinnt zunehmend an Bedeutung. Das ML bietet die Möglichkeit, Druckluftsysteme nicht nur zu überwachen, sondern auch selbstständig Anomalien und Leckagen zu identifizieren. Auf ML basierte Lösungen bieten das Potenzial, diese Systeme sowohl

automatisiert als auch selbstlernend und anwendungsspezifisch zu optimieren (Dierolf & Sauer 2021; Dierolf 2022; Mallia et al. 2023).

Ein zentrales Problem in diesem Zusammenhang ist jedoch die Sensibilität der Daten. Nutzer von Sensoren in Produktionsprozessen haben oft Bedenken in Bezug auf den Datenaustausch, da diese Daten Rückschlüsse auf interne Produktionsprozesse und -geheimnisse zulassen können. Angesichts der vielfältigen und datensensiblen Natur von Maschinen- und Betriebsdaten in Produktionsprozessen ist FL eine mögliche Methode, um die Erkennung von Druckluftleckagen zu optimieren und gleichzeitig die gebotene Datensicherheit zu gewährleisten. Aus diesem Grund wird die Erkennung von Druckluftleckagen als zweiter Anwendungsfall in dieser Arbeit betrachtet.

6.2.2 Entwicklung des Modells zur energetischen Anomalieerkennung

In diesem Abschnitt wird die erste Phase zur Auslegung des FL-basierten Systems zur energetischen Anomalieerkennung für die vorliegende Fallstudie auf Basis der in Abschnitt 5.3.1 entwickelten Vorgehensweise durchgeführt.

6.2.2.1 Problem definieren

In der vorliegenden Problemstellung wird das PEMD-Framework zur Strukturierung und Analyse der relevanten Dimensionen herangezogen. Die vier Dimensionen sind in Tabelle 6 dargestellt und im Folgenden detailliert erläutert.

Tabelle 6 Darstellung des PEMD-Frameworks für die Fallstudie

	Dimension	Ausprägung
PEMD-Framework	***Leistung***	Erkennung von Druckluftleckagen in Produktionsprozessen zur Erhöhung der Energieeffizienz; Minimierung der Kosten zur Erzeugung der Druckluft pro Prozessschritt; F1-Score
	Umgebung	Druckluftsystem, Kompressor, Aktoren, speicherprogrammierbare Steuerung, Durchflusssensor
	Maßnahmen	Logging in der speicherprogrammierbaren Steuerung, Visualisierung in der Betriebsleitebene, Kontaktieren des Wartungsteams zur Beseitigung der Leckage
	Daten	Massenstrom, Steuerungsdaten der speicherprogrammierbaren Steuerung

Das primäre Ziel, das der Dimension **Leistung** entspricht, besteht in der Erkennung von Druckluftleckagen in Produktionsprozessen. Durch eine zeitnahe Leckagen-Identifizierung anhand von ML-Modellen und historischen Daten können signifikante Energieverluste und damit Kosten zur Erzeugung von Druckluft vermieden werden. Um die Leistungsfähigkeit des Systems zur Erkennung von Leckagen in Produktionsprozessen zu bewerten, eignet sich, wie in der ersten Fallstudie, aufgrund des harmonischen Mittelwerts aus Genauigkeit und Erkennungsrate der F1-Score.

Die **Umgebung** der betrachteten Fallstudie besteht aus dem Druckluftsystem, dem Kompressor, Aktoren, einer SPS und verschiedenen Sensoren wie etwa Beschleunigungssensor, Positionssensor und dem Durchflusssensor der Aktoren. Diese Komponenten sind miteinander vernetzt, um die Energiedaten des Systems zu überwachen. Die Sensoren generieren die für die energetische Anomalieerkennung notwendigen historischen Energiedaten.

Für diese Fallstudie wurden drei zentrale **Maßnahmen** identifiziert: die Protokollierung der Anomalien auf einer Recheninstanz, die Visualisierung auf der Betriebsleitebene und

eine Alarmfunktion für die Instandhaltung. Im Falle einer erkannten Leckage werden die relevanten Informationen in der SPS protokolliert, um eine historische Erfassung der erkannten Anomalien zu gewährleisten, die zu einem späteren Zeitpunkt ausgelesen und analysiert werden können. Zusätzlich werden die Leckagen auf der Betriebsleitebene visualisiert und in Form eines Alarms weitergeleitet, sodass z. B. ein Instandhaltungsteam umgehend informiert wird, um entsprechende Maßnahmen zur Behebung der Leckage einleiten zu können.

Die **Daten**, die für die Modellierung von Druckluftleckagen verwendet werden können, umfassen den Massenstrom sowie die Steuerungsdaten der SPS. Der Massenstrom stellt einen kritischen Parameter dar, der auf potenzielle Leckagen im System hinweisen kann. Die Steuerungsdaten der SPS hingegen liefern Informationen über den Betriebszustand der verschiedenen pneumatischen Aktoren und können folglich als kontextbasierte Betriebsdaten genutzt werden. Durch die Kombination dieser Daten kann ein ML-Modell zur Erkennung energetischer Anomalien anhand der pneumatischen Aktorbewegungen entwickelt werden, das sowohl auf Sensordaten als auch auf den kontextbasierten Informationen über den Betriebszustand des Systems basiert. Als Ergebnis können mögliche Leckagen im Druckluftsystem erkannt werden.

Für die Erkennung energetischer Anomalien anhand pneumatischer Aktorbewegungen auf Basis von Massenstromdaten der Aktorbewegungen wurde ein experimenteller Ansatz unter Laborbedingungen gewählt, in dem Leckagen im Druckluftsystem real simuliert werden können.

Zur Datenerfassung wurde der Labordemonstrator *Wandlungsfähige Druckluftanlage* (WanDa) eingesetzt (siehe Abbildung 6-17). Dieser Demonstrator ist in der Lage, verschiedene Aktorbewegungen zu simulieren. Für diese Fallstudie wurden Zylinderköpfe mit variablen Gewichten eingesetzt, die pneumatisch bewegt werden konnten. Die Programmierung dieser Bewegungsabläufe erfolgte mit Node-RED und einer eigens entwickelten Bedienoberfläche, die eine individuelle Anpassung der Bewegungsabläufe durch den Bediener ermöglicht.

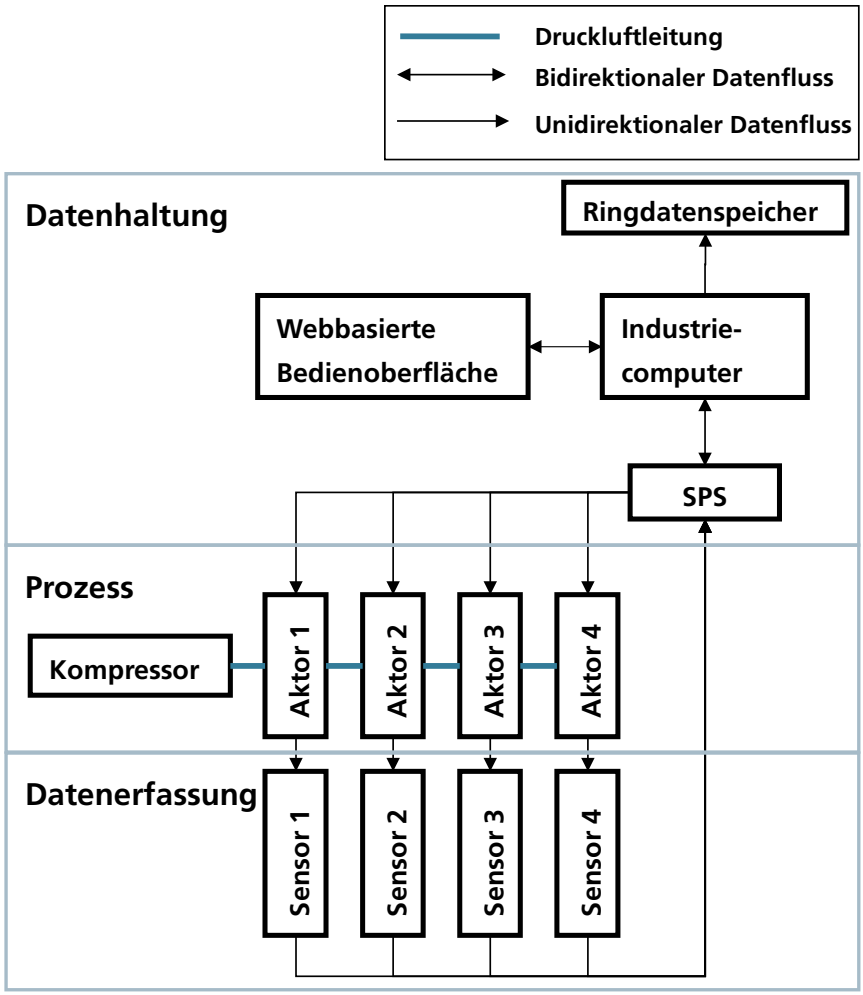


Abbildung 6-17 Darstellung des Versuchsaufbaus des Laborexperiments

Das gesamte System wurde mit einem Druck von 4 bis 6 bar betrieben. Für jeden Aktor im System sind Sensoren installiert, die den Massenstrom sowie weitere Parameter wie Temperatur, Volumenstrom, Druck und Fließgeschwindigkeit erfassen. Obwohl die Sensoren zahlreiche Datenpunkte liefern, wurde in diesem Anwendungsfall bewusst nur der Massenstrom zur Erkennung von Druckluftleckagen verwendet. Dies sollte zeigen, dass auch mit einem im Vergleich einfach berechenbaren Parameter und einer reduzierten Datenmenge zuverlässige Aussagen getroffen werden können. Um eine Variation in den Daten zu gewährleisten, wurden drei verschiedene Gewichtsszenarien betrachtet: kein Gewicht, halbes Gewicht und volles Gewicht. Zusätzlich wurden die SPS-Daten der einzelnen Aktoren erfasst, die insbesondere für die Kontexterstellung von Bedeutung waren.

Eine weitere Variation wurde durch das manuelle Bohren von Löchern in die Schläuche des Druckluftsystems eingeführt. Es wurden Löcher mit einem Durchmesser von 1 mm, 2 mm und 3 mm gebohrt, um verschiedene Leckage-Größen zu simulieren. Die Daten des Durchflusssensors und der SPS-Daten wurden lokal auf einer Recheninstanz gespeichert. Die Abtastrate für die Datenerfassung betrug 100 Hz. In Abbildung 6-18 ist der Massenstrom während den Aktorbewegungen dargestellt. Dabei wurden die Massenstromverläufe jeweils für die unterschiedlichen Lochgrößen (0–3 mm) visualisiert.

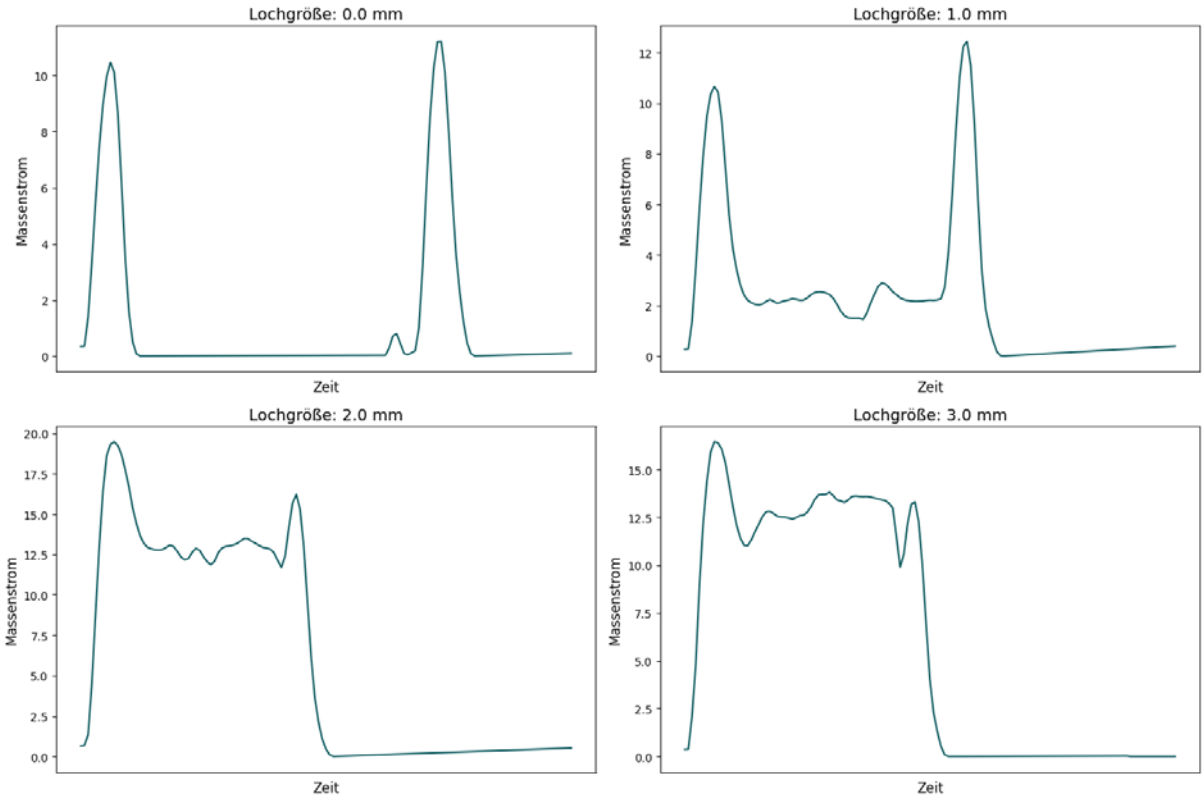


Abbildung 6-18 Visualisierung des Massenstroms bei unterschiedlichen Lochgrößen

Alle gesammelten Daten wurden entsprechend gekennzeichnet, um Informationen über das jeweilige Gewicht und die Größe des eingebauten Loches zu enthalten. Dies gewährleistet eine eindeutige Zuordnung und erleichtert die spätere Auswertung.

6.2.2.2 Kontextbasierte Energiezeitreihe aufbereiten

Im Folgenden werden die erfassten Rohdaten im ersten Schritt vorverarbeitet, um daraufhin eine aufbereitete, kontextbasierte Energiezeitreihe auf Grundlage des Massenstroms zu erstellen. In dieser Fallstudie wird aufgrund der Daten der SPS mithilfe der WAHR/FALSCH-Werte die rein kontextbasierte Sicht betrachtet. Daraufhin erfolgt die Transformation der kontextbasierten Energiezeitreihe hin zur kontextbasierten Energiezeitreihengruppe.

Im Zuge der Datenaufbereitung für die Erkennung von Druckluftleckagen werden die Daten aufgrund der nicht äquidistanten Messung interpoliert und auf die gewünschte maximale Abtastrate von 10 Hz gebracht. Besonderes Augenmerk wird auf die Anpassung der Daten im zeitlichen Kontext gelegt. Um einen äquidistanten Datensatz mit konsistenten Zeitabständen zwischen den Datenpunkten zu generieren, ist die Ermittlung der minimalen und maximalen Zeitstempel erforderlich. Basierend auf diesen Werten wird ein Zielzeitindex erstellt, der die gewünschte Abtastrate (in diesem Fall 10 Hz) widerspiegelt. Mit dem festgelegten Zielzeitindex werden die Daten reindiziert, um sicherzustellen, dass jeder Zeitpunkt des Zielzeitindexes in den Daten vorkommt. Anschließend werden die Daten interpoliert, wobei numerische Daten mit ihrem Index und nichtnumerische Daten durch Auffüllen interpoliert wurden.

Nach der Interpolation erfolgen die erneute Überprüfung der Daten und die Anpassung des Zielzeitindex, um sicherzustellen, dass sie der gewünschten Abtastrate von 10 Hz entsprechen. Schließlich werden die interpolierten Daten und die Daten mit 10 Hz in einer Parquet-Datei gespeichert. Nachdem die Rohdaten mit einer Abtastrate von 10 Hz in eine konsistente Form gebracht wurden, erfolgte im nächsten Schritt die weitere Aufbereitung der Daten in eindeutig identifizierbare Aktorbewegungen. Dies beinhaltet eine detaillierte Segmentierung, Identifikation und Kategorisierung der Datensätze, um sie für die weiteren Schritte in dieser Phase nutzbar zu machen. Zu Beginn wird ein Datensatz geladen, der Sensordaten, Aktorbezeichnungen und Steuersignale enthält. Um die Datenqualität zu gewährleisten, werden fehlende Werte in den Beschriftungsspalten fortlaufend ergänzt und Datenpunkte ohne Sensordaten entfernt. Anschließend werden

die Daten aufgeteilt, um sie für jeden einzelnen Sensor zu isolieren. Für jeden Sensor werden die zugehörigen Daten extrahiert, beschriftet, anhand ihrer Bezeichnungen gefiltert und entsprechenden numerischen Labels zugewiesen. Sobald sie für jeden Sensor einzeln aufbereitet worden sind, werden sie wieder zu einem Datensatz zusammengeführt. Dieser konsolidierte Datensatz wird weiterbearbeitet, um eindeutige ID für jede einzelne Aktorbewegung zu erstellen.

Ein weiterer notwendiger Vorverarbeitungsschritt für diesen Anwendungsfall ist die Eliminierung von Ausreißern, da die entnommene Zykluszeit aus der SPS einzelner Aktorbewegungen höher war als die der Mehrheit der Aktorbewegungen. Dazu wurde der Interquartilbereich aller Zykluszeiten verwendet, um Datenpunkte zu identifizieren, die fehlerhaft oder unerwünscht sein könnten. Datenpunkte, deren ID-Länge außerhalb eines bestimmten Bereichs liegen, wurden als Ausreißer betrachtet und entfernt. Schließlich wird das Zero-Padding-Verfahren angewandt, um sicherzustellen, dass alle Datensegmente die gleiche Länge haben. Dies ist besonders für bestimmte ML-Modelle relevant, da eine definierte Dimensionalität der Daten erforderlich ist (z. B. KNN).

Auf Grundlage der aufbereiteten, kontextbasierten Energiezeitreihengruppe erfolgt die Generierung der kontextbasierten Merkmalstabelle. Daraufhin erfolgt die Trennung des Datensatzes in einzelne Zeitreihen. Anhand dieser können im nächsten Schritt Merkmale extrahiert werden.

6.2.2.3 Merkmale extrahieren

Wie in Abschnitt 5.3.1.3 beschrieben, werden in diesem Schritt die relevanten Merkmale der Aktorbewegungen extrahiert. Für Letztere wurde lediglich der Massenstrom als zusätzliche Zeitreihe integriert. Der Massenstrom gibt an, wie viel Material pro Zeiteinheit durch einen Querschnitt fließt, und ist somit ein aussagekräftiges Merkmal zur Charakterisierung der Aktorbewegung. Da keine weiteren prozessabhängigen Merkmale gebildet wurden, beschränkt sich die Erweiterung der ursprünglichen Datensätze auf diesen einen Wert. Nach der Integration des Massenstroms in den Datensatz erfolgte die

Erstellung der kontextbasierten Merkmalstabelle. Dazu wurden Merkmale aus der vorhandenen multivariaten Zeitreihe der Aktorbewegungen und pro Zyklus anhand der kontextbasierten Betriebsdaten berechnet. Da die Extraktion der Merkmale pro Zeitreihe erfolgt, kann bei der Merkmalsextraktion eine große Menge an Merkmalen entstehen, die in die Modellierung einfließen können.

Die kontextbasierte Merkmalstabelle mit einer Abtastrate von 10 Hz beschreibt die jeweiligen Aktorbewegungsprozesse mit maximal 758 Merkmalen, die aus der Zeitreihe des Massenstroms extrahiert wurden.

6.2.2.4 Modelle trainieren

Für das Training werden die in Abschnitt 5.3.1.4 definierten Modelle initialisiert und parametrisiert, um sicherzustellen, dass die Modellkonfiguration dem jeweiligen Datensatz entspricht. Dabei wird, wie in den Abschnitten 5.3.1.4 und 5.3.1.5 beschrieben, nach Trainings- und Testdaten getrennt. Das Training erfolgt mithilfe einer zehnfachen Kreuzvalidierung, um die statistische Signifikanz der Ergebnisse zu erhöhen.

Für die Detektion von Druckluftleckagen bei Aktorbewegungen werden verschiedene Experimentkonfigurationen erstellt. Die verwendeten Abtastraten liegen bei 1, 2, 5, 8 und 10 Hz. Anschließend wird für jede Experimentkonfiguration ein Experiment durchgeführt, bei dem die Abtastrate des Datensatzes entsprechend angepasst wird. Ein spezielles Datenansicht-Objekt wird erstellt, um den Datensatz für die anschließende Verarbeitung und Analyse vorzubereiten. Für jedes Experiment werden zwei verschiedene Datensichten verwendet – eine auf Grundlage der kontextbasierten Merkmalstabelle sowie eine, die auf der kontextbasierten Energiezeitreihengruppe basiert. Die Kreuzvalidierung wird unter Berücksichtigung der jeweiligen Experimentkonfiguration durchgeführt, wobei die ausgewählten Augmentationsmechanismen und die Abtastrate zu berücksichtigen sind. Schließlich werden die Ergebnisse der verschiedenen Experimente in einer Ergebnistabelle (siehe Anhang: Tabelle 12) zusammengefasst und für die weitere Analyse wie auch Interpretation gespeichert.

In Phase I wurden 550 Modelle mit insgesamt 4582 Aktorbewegungen trainiert und evaluiert.

6.2.2.5 Modell auswählen

In diesem Abschnitt sollen die Ergebnisse vorgestellt werden, um das Modell mit dem höchsten LEI auszuwählen, das für die Implementierung des Anomaliedetektors und für die Simulation von FL genutzt werden kann. Da die Modelle mit den bereits vorhanden realen Daten im Durchschnitt einen F1-Score über 0,9 erreichen und eine ausreichende Menge an Beispielen für Aktorbewegungen vorhanden sind, wurde auf eine Augmentation der Daten verzichtet.

Für die Analyse der Abtastraten zeigt Abbildung 6-19 auf der X-Achse die verschiedenen Abtastraten und auf der Y-Achse den durchschnittlichen F1-Score.

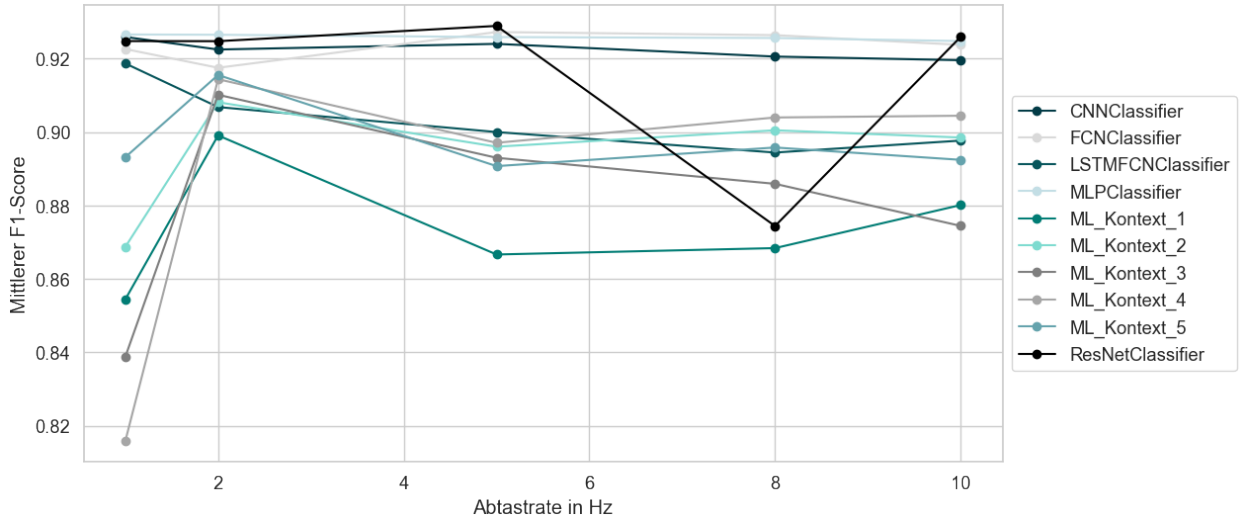


Abbildung 6-19 Visualisierung der F1-Scores in Abhängigkeit von der Abtastrate und der Modelle

Nach dem Modellauswahlverfahren auf Grundlage des $sLEI$ ist die beste Konfiguration, gemessen am höchsten $sLEI$, das Modell CNN mit einer Abtastrate von 1 Hz. Der $sLEI$ beträgt 0,950 und hat eine Trainingszeit von 83 s sowie einen F1-Score von 0,926.

Auch in dieser Fallstudie hat das Modell ResNet mit einer Abtastrate von 5 Hz den höchsten F1-Score (0,929) aller betrachteten Modelle. Allerdings erfordert es eine lange Trainingszeit von ca. 2000 s, sodass der $sLEI$ (0,873) weitaus niedriger ist als der des CNN-Modells mit einer Abtastrate von 2 Hz.

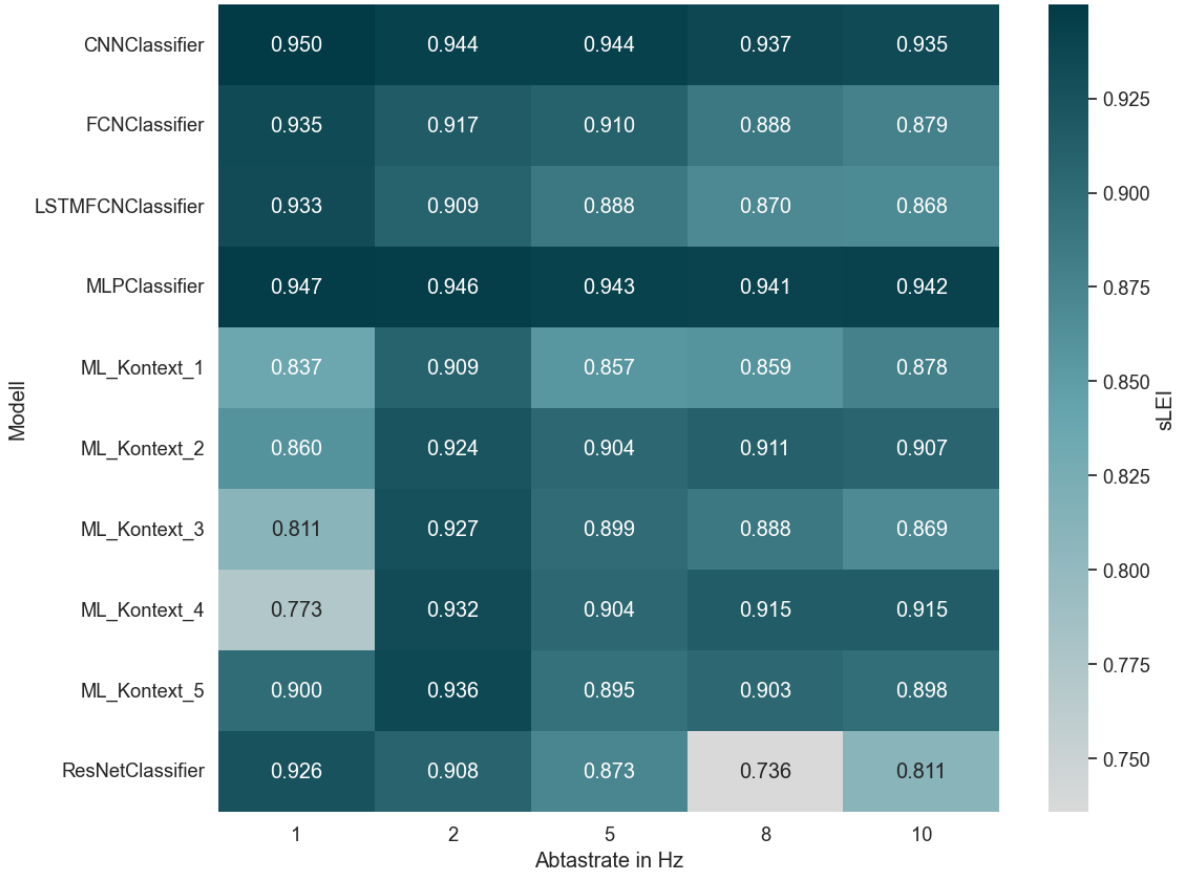


Abbildung 6-20 Heatmap des LEI der unterschiedlichen Modell-Abtastrate-Kombinationen

Abbildung 6-20 zeigt u. a., dass die Modelle, basierend auf den kontextbasierten Energiezeitreihengruppen bei niedrigeren Abtastraten höhere $sLEI$ als die Modelle basierend auf den kontextbasierten Merkmalstabellen aufweisen. Insbesondere die CNN- und MLP-Modelle mit niedrigeren Abtastraten (1 und 2 Hz) weisen ein höheren $sLEI$ auf als ihre Pendants mit höheren Abtastraten.

Wie erwartet, steigt die Trainingszeit der meisten Modelle mit zunehmender Abtastrate. Dies ist insbesondere bei Modellen wie FCN und ResNet zu beobachten, die bei höheren Abtastraten einen deutlichen Anstieg aufweisen. Die Zunahme der Trainingszeit mit steigender Abtastrate kann auf die größere Datenmenge zurückgeführt werden, die

verarbeitet werden muss. Je höher die Abtastrate, desto mehr Datenpunkte liegen in einem festen Zeitintervall vor, was zu längeren Trainingszeiten führt. Einige Modelle – z. B. MLP – zeigen über alle Abtastraten hinweg einen stabilen F1-Score und einen stabilen $sLEI$. Aus Sicht der Dateneffizienz und somit auch der Kosten der Rechenkapazität sowie der Datenspeicherung ist es vorteilhaft, eine niedrigere Abtastrate zu wählen, da dies zu kürzeren Trainingszeiten und geringeren Verarbeitungskosten führt.

Insgesamt scheint somit das CNN-Modell mit einer Abtastrate von 1 Hz die effizienteste Konfiguration zu sein, wenn der $sLEI$ als Evaluationsmetrik verwendet wird.

6.2.3 Implementierung des Anomaliedetektors

Im Folgenden wird die konzeptionelle Architektur des Anomaliedetektors für die vorliegende Fallstudie auf Basis der in Abschnitt 5.3 entwickelten Vorgehensweise abgeleitet. Dabei werden sowohl die Anforderungen als auch die Restriktionen des Anwendungsfalls berücksichtigt.

6.2.3.1 Datenpipeline aufbauen

In diesem Abschnitt wird die grundlegende Datenverarbeitungspipeline mit dem Ziel aufgebaut, den durchgängigen Datenfluss von Datenquelle bis hin zum Agenten innerhalb des Anomaliedetektors darzustellen (siehe Abbildung 6-21). Dabei wird auf die einzelnen Komponenten des Anomaliedetektors eingegangen. Im ersten Schritt sollen die für diese Fallstudie identifizierten Datenquellen vorgestellt werden.

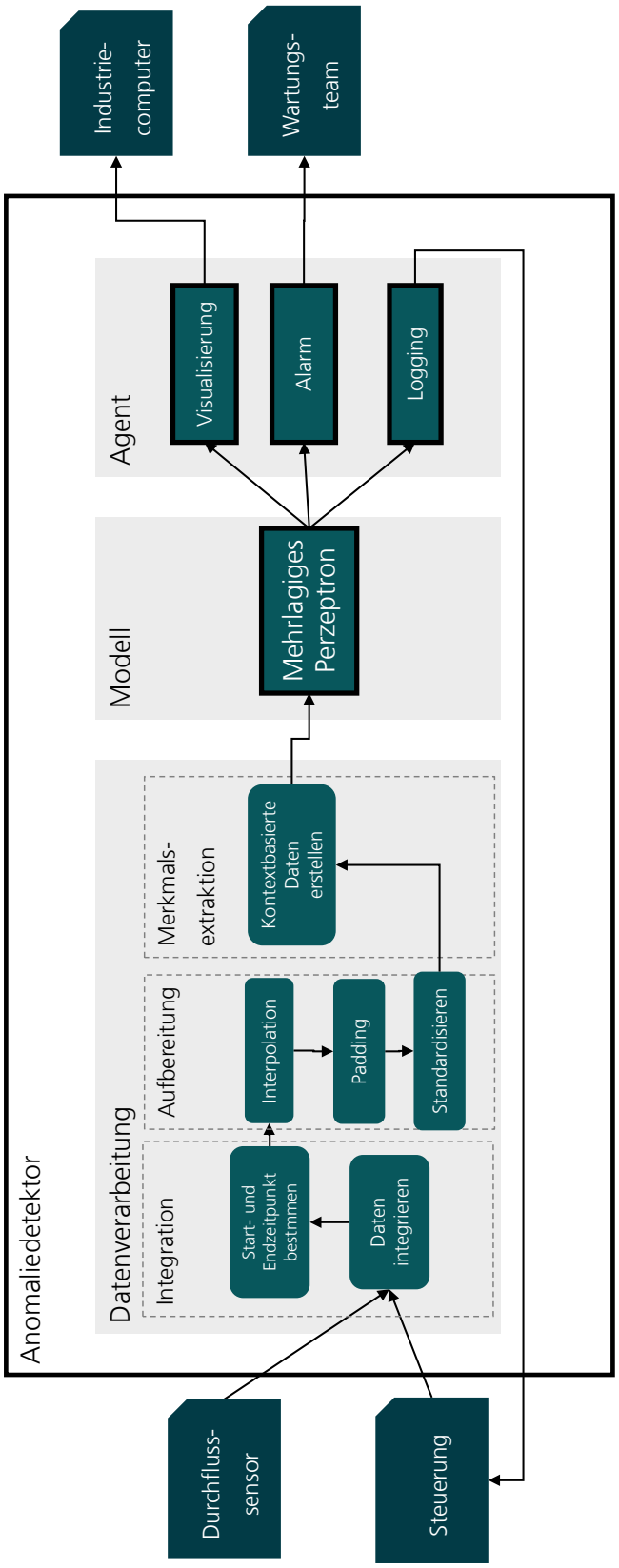


Abbildung 6-21 Konzept des Anomaliedetektors für die Erkennung von Druckluftleckagen bei pneumatischen Aktorbewegungen

Wie beim PEMD-Framework in Abschnitt 6.2.2.1 beschrieben, werden zwei Datenquellen betrachtet: der Durchflusssensor als multivariate Energiezeitreihe und die kontextbasierten Betriebsdaten der SPS, die zur Steuerung des Aktors genutzt werden. Für beide Datenquellen wird die GET-Funktionalität beschrieben, was eine weitere Verarbeitung der Daten innerhalb des Anomaliedetektors ermöglicht.

Der Durchflusssensor ermöglicht die Erfassung von Daten des Druckluftverbrauchs. Hierzu gehört der Massenstrom, der Volumenstrom und die verbrauchte Energie, aber auch die Temperatur. Die mit einer maximalen Abtastrate von 100 Hz erfassten Daten werden über IO-Link oder Ethernet an einen MQTT-Broker übertragen. Aus der SPS werden die zusätzlichen Informationen erfasst, die einer Aktorbewegung direkt zugeordnet werden können. Diese WAHR/FALSCH-Werte je Zylinder werden auch auf den OPC-UA-Server geladen. Die Übertragung der Daten aus der SPS erfolgt durch den MQTT-Broker und Telegraf. Die Steuersignale aus der SPS werden über den internen OPC-UA-Server an den gleichen MQTT-Broker und an Telegraf, ein agentenbasiertes Tool für die Orchestrierung von Datenquellen, übertragen, also gleichsam bereits integriert als ein Datensatz vereinheitlicht. Die Daten werden in einem vordefinierten JSON-Schema übertragen und können so mithilfe von Bibliotheken der Programmiersprache Python weiterverarbeitet werden, da das Datenformat – anders als bei Pickle oder Parquet – auch von anderen Programmiersprachen und Bibliotheken gelesen werden kann. Mithilfe von Telegraf werden die Daten im Rohdatenformat auf der eigenen Festplatte des Rechners gespeichert. Die Daten aus beiden Datenquellen werden bereits integriert in Datenblöcken mit einer Größe von 100 MB im JSON-Format gespeichert und können weiterverarbeitet werden.

Nach der Definition der Datenquellen sowie der Übertragung der beiden Datensätze aus dem Durchflusssensor und der SPS erfolgt, wie bereits in 6.3.2.2 beschrieben, die Aufbereitung der Daten. Durch die Verwendung einer ID ist keine aufwendige Zeitsynchronisation wie bei der klassischen Signalverarbeitung notwendig. Die Betriebsdaten aus der SPS und die Sensordaten des Durchflusssensors können somit integriert werden.

Um einen äquidistanten Datensatz mit konsistenten Zeitabständen zwischen den Datenpunkten zu gewährleisten, kann die gleiche Vorgehensweise wie aus Abschnitt 6.2.2.2 angewandt werden. Da in 6.2.2 schon eine Abtastrate identifiziert wurde, erfolgt die Interpolation direkt auf die Abtastrate von 1 Hz. Schließlich werden die interpolierten Daten und die Daten mit 1 Hz in einer Parquet-Datei gespeichert. Nachdem die Daten mit einer Abtastrate von 1 Hz in eine konsistente Form gebracht wurden, erfolgt im nächsten Schritt die weitere Aufbereitung der Daten in eindeutig identifizierbare Aktorbewegungen. Dies beinhaltet eine detaillierte Segmentierung, Identifikation und Kategorisierung der Datensätze, um Letztere für die weiteren Schritte in dieser Phase nutzbar zu machen. Dabei wird eine Aktorbewegung als ein Wechsel von WAHR zu FALSCH zu WAHR definiert. Dieser Vorgang liegt für die meisten Bewegungen im Bereich zwischen 5 und 20 s. Damit jedoch alle Aktorbewegungen für das Training und für die Inferenz des Anomaliedetektors die gleiche Länge aufweisen, wird das Zero-Padding-Verfahren zur Angleichung der einzelnen Längen genutzt (siehe Abschnitt 6.2.2.2).

Da in Abschnitt 6.2.2.5 das Modell CNN ausgewählt wurde, ist keine Extraktion von statistischen Merkmalen erforderlich. Dennoch können anhand der bereitgestellten Skripte die prozessabhängigen Merkmale anhand der Daten aus der Schweißstromquelle erstellt werden. Mithilfe der eindeutigen ID lassen sich die einzelnen Aktorbewegungen in kontextbasierte Energiezeitreihengruppen einteilen. Diese Zeitreihengruppen können für das Training und somit für die FIT-Funktionalität genutzt werden.

Im Kontext der PREDICT-Funktionalität wird eine einzelne Aktorbewegung und nicht die gesamte kontextbasierte Energiezeitreihengruppe betrachtet. Diese Aktorbewegung durchläuft die gleiche Datenpipeline mit den gleichen Datenverarbeitungseinheiten wie bei der FIT-Funktionalität. Entsprechend werden die Daten aus dem Durchflusssensor integriert und bereinigt. Die bereinigte und vorverarbeitete Aktorbewegung wird dann zur Inferenz an das bereits trainierte Modell aus der FIT-Funktionalität übergeben. Wichtig ist, dass das Modell von einer grundlegenden Aktorbewegung von unter 20 s ausgeht, da das Zero-Padding-Verfahren und auch das trainierte Modell von einer Datenstruktur mit zwanzig Werten bei 1 Hz ausgehen, weshalb die Datenstruktur auf 20 s festgelegt ist. Anstatt ein Modell auf der kontextbasierten Energiezeitreihengruppe zu trainieren, wird

das bereits trainierte Modell verwendet, um Vorhersagen für eine einzelne Aktorbewegung zu treffen. Dabei kann die PREDICT-Funktionalität immer den letzten generierten Datenblock aufbereiten und stapelweise verarbeiten. Auch die Ergebnisse der PREDICT-Funktionalität können als gestapelte Datenblöcke für die jeweiligen Aktorbewegungen auf der Recheninstanz von WanDa gespeichert werden. Dies ist vor allem für darauffolgende Gegenmaßnahmen von Vorteil, die diese Daten direkt nutzen können, ohne die PREDICT-Funktionalität durchgehend zu nutzen. Eine Rekonfiguration des Telegraf-Agenten ermöglicht auch die Vorhersage einer Aktorbewegung. Es kann eine regelbasierte Abfrage nach jedem WAHR-FALSCH-WAHR Vorgang der SPS implementiert werden. So können alle Datenblöcke als einzelne Aktorbewegung gespeichert werden.

6.2.3.2 Agent implementieren

In Anlehnung an Abschnitt 5.3.2.2 wird im Folgenden die praktische Umsetzung der Agenten und der zugehörigen Gegenmaßnahmen dargestellt. In der vorliegenden Fallstudie werden drei Agenten konzeptioniert, die unterschiedliche Formen der Interaktion mit dem System und dem Anwender bereitstellen.

Visualisierung: Der erste Agent stellt einen integralen Bestandteil eines übergeordneten Leitsystems dar, um die Integration der Ergebnisse des in Abschnitt 6.2.2.5 trainierten Modells in ein *Manufacturing Execution System* (MES) zu ermöglichen. Dies bedeutet, dass die Erkennung einer Leckage anhand einer Aktorbewegung und des gemessenen Massenstroms ermöglicht wird. Die Hauptaufgabe besteht darin, die vom Modell generierten Daten in eine für das MES verständliche Form zu transformieren und sicherzustellen, dass diese Daten innerhalb des MES visualisiert werden. Die Datenübertragung erfolgt in zwei Schritten:

Nach jeder Aktorbewegung wird die PREDICT-Funktionalität aufgerufen, um eine Vorhersage für die aktuelle Aktorbewegung zu erhalten. Diese Funktionalität ermöglicht in dieser Anwendung eine Vorhersage dazu, ob eine Leckage vorliegt oder nicht. Dies

erfordert eine geeignete Schnittstelle für die Kommunikation zwischen dem Aktor und der Visualisierung. Dabei kann das Programm zur Visualisierung als ein Service auf der Recheninstanz von WanDa bereitgestellt werden. Durch Telegraf ist es möglich, die gespeicherten Ergebnisse direkt abzurufen. Es müssen also keine zusätzlichen Schnittstellen entwickelt werden. So kann z. B. ein zusätzliches Merkmal, das für jeden Zeitstempel innerhalb des Datensatzes vorhanden ist, über die genannte Schnittstelle übertragen werden. Die Visualisierung erfolgt dann durch die Markierung der Zeitstempel, in denen Leckagen identifiziert wurden. Zusätzlich können bei mehreren Aktoren auch jene markiert werden, bei denen eine Leckage festgestellt wurde.

Logging: Dieser Agent ist für die Protokollierung der Modellergebnisse in einem lokalen Skript verantwortlich, das auf dem Einplatinencomputer ausgeführt wird. Diese Form der Interaktion mit dem System ermöglicht eine direkte Überprüfung der Ergebnisse der energetischen Anomalieerkennung und kann helfen, schnell auf die Modellergebnisse zu reagieren. Die Implementierung dieses Agenten erfordert eine enge Integration der PREDICT-Funktionalität. Nach jeder Aktorbewegung wird die PREDICT-Funktionalität aufgerufen, um eine Vorhersage für die aktuelle Aktorbewegung zu erhalten. Das Ergebnis des Modells wird mit Hilfe des Logging-Agenten direkt in die Konsole geschrieben. Um dies zu ermöglichen, muss eine geeignete Schnittstelle für die Kommunikation zwischen dem Agenten und dem Modell implementiert werden – zu realisieren bspw. durch eine Funktion innerhalb des Python-Skripts, die das Modell aufruft und die Vorhersage zurückgibt. Darüber hinaus sollte der Agent in der Lage sein, die Ausgabe der Vorhersage auf der Konsole zu formatieren und zu protokollieren. Aus diesem Grund werden Tagesprotokolle aus dem Logging-Agenten erstellt, der die energetischen Anomalien innerhalb eines Tages zusammenfasst. Mit dem Tagesprotokoll können im nächsten Schritt die energetischen Anomalien innerhalb des Tages analysiert werden, um etwaige grundlegende Ursachen zu identifizieren. Dennoch bietet die Bereitstellung von Echtzeit-Feedback auch die Möglichkeit, die energetischen Anomalien direkt nach der Aktorbewegung zu erkennen, um daraufhin die Ursache zu finden und diese zu beseitigen.

Alarm: Der Alarm-Agent arbeitet ähnlich wie die Visualisierung oder das Logging. Der Hauptunterschied besteht in der Art der ausgegebenen Alarme: Statt die Ergebnisse einfach in der Konsole des intelligenten Sensors zu protokollieren oder die zusätzlichen Daten an das MES zu versenden, überträgt der Alarm-Agent bei der Erkennung einer energetischen Anomalie die Informationen an ein registriertes mobiles Endgerät des Anwenders. Hierbei soll eine Kommunikationsschnittstelle genutzt werden, die entweder eine E-Mail oder eine SMS versendet. Für den E-Mail-Versand ist eine SMTP-Server-Integration erforderlich. Der Alarm Agent könnte eine vordefinierte E-Mail-Vorlage verwenden, die die notwendigen Informationen über die erkannte Leckage enthält. Es kann auch nützlich sein, zusätzliche Anhänge wie Logs oder Visualisierungen der Anomalie hinzuzufügen, um dem Empfänger ein klareres Profil des Massenstroms zu geben. Eine weitere Funktionalität des Alarm-Agenten ist die Möglichkeit, Alarme nach Dringlichkeit oder Art zu priorisieren – zu erreichen etwa durch eine Kombination aus benutzerdefinierten Regelalgorithmen oder durch ML. Dadurch wird sichergestellt, dass kritische Alarme die notwendige Aufmerksamkeit erhalten und weniger kritische Alarme entsprechend kategorisiert werden.

6.2.3.3 Anomaliedetektor bereitstellen

Für die Bereitstellung des Anomaliedetektors wird die Recheninstanz von WanDa genutzt. Da die unterschiedlichen Dienste, etwa der bereits vorgestellte MQTT-Broker oder der Telegraf-Agent, über containerbasierte Software bereitgestellt werden, wird im Zuge einer einfachen Bereitstellung auch für den vorgestellten Anomaliedetektor ein Softwarecontainer genutzt. Grundsätzlich sind auf diesem Python 3.9 und alle notwendigen Python-Bibliotheken mitinstalliert. Mithilfe von Telegraf und der vom Softwarecontainer vorgehaltenen IP-Adresse können die vom MQTT-Broker bereitgestellten Datenblöcke direkt über Telegraf vom Softwarecontainer des Anomaliedetektors empfangen werden. Mithilfe eines Python-Skripts auf Basis der zuvor vorgestellten Funktionalitäten (GET, FIT und PREDICT) können die Datenblöcke direkt

verarbeitet und das Modell für die Erkennung der Anomalien genutzt werden. Die Ergebnisse wiederum lassen sich über Telegraf auf der Recheninstanz von WanDa stapelweise speichern. Zusätzlich ermöglicht die PREDICT-Funktionalität den bereits vorgestellten Agenten eine direkte Kommunikation mit dem Softwarecontainer des Anomaliedetektors.

6.2.4 Parameteroptimierung des föderierten Lernens

Dieser Abschnitt beschreibt die dritte Phase zur Auslegung des FL-basierten Systems zur energetischen Anomalieerkennung für die vorliegende Fallstudie auf Basis der in Abschnitt 5.3.3 entwickelten Vorgehensweise durchgeführt.

6.2.4.1 Konfigurationsparameter definieren

Zunächst sollen in diesem Abschnitt die Konfigurationsparameter für die Durchführung der Simulation definiert werden. Diese Konfigurationsparameter werden auf Grundlage von Abschnitt 5.3.3.1 gewählt. Dabei fließen zusätzlich die Ergebnisse aus Phase I mit ein. In Tabelle 7 sind die gewählten Konfigurationsparameter aufgelistet.

Tabelle 7 Definition der Konfigurationsparameter für die Simulation von FL

Parameter	Beschreibung	Wert
Aggregations-mechanismus	Bestimmt, wie die Modelle der Clients aggregiert werden	FedAvg FedOpt FedAdam FaultTolerantFedAvg QFedAvg
Anzahl der Trainingsrunden	Anzahl der Durchführungen des föderierten Lernprozesses	100
Client-Anzahl	Anzahl der am Lernprozess beteiligten Clients	[2, 8]
Lokale Epochen	Anzahl der Trainingsschleifen auf jedem Client	[1, 20]
Batch-Größe	Anzahl der Datenpunkte, die in jedem Schritt verwendet werden	[16, 128]

6.2.4.2 Simulationsbasierte Experimente durchführen

Für die Simulations- und Optimierungsberechnungen wurden die gleichen Hardware- und Softwarespezifikationen wie im ersten Anwendungsfall in Abschnitt 6.1 genutzt. Für die Simulation werden im ersten Schritt die Trainings- und Testdaten aus Abschnitt 6.2.2 eingesetzt, um eine konsistente und gleichbleibende Trennung der Daten zu gewährleisten. Dabei werden die Testdaten aus 6.2.2 hier als der globale Testdatensatz definiert und die Trainingsdaten auf Zufallsbasis in zehn gleiche Segmente aufgeteilt. Somit werden für die Trainingsdaten 4156 Aktorbewegungen und für den Testdatensatz 426 Aktorbewegungen verwendet. Innerhalb der Segmente wird explizit nicht auf eine Gleichverteilung zwischen guten und anomalen Schweißungen geachtet, um eine gewisse Heterogenität in den einzelnen Segmenten zu ermöglichen. Innerhalb der einzelnen Segmente werden für den lokalen Testdatensatz 10 % der Segmente zufällig

ausgewählt, sodass jedes einzelne Segment nach dem lokalen Training mit den rein lokalen Daten getestet werden kann.

Für die Optimierung wurden fünfzig Runden festgelegt. Daher werden ebenso viele unterschiedliche Konfigurationen ausgeführt, wobei die in Abschnitt 5.3.3.2 definierte kombinierte Metrik als Grundlage für die Anpassung der Konfigurationsparameter nach jeder Runde fungiert. Insgesamt wurden somit 25 000 lokale Modelle trainiert und 7950 Validierungsschritte auf dem globalen Testdatensatz ausgeführt. Die Ergebnisse wurden für die spätere Analyse und Bewertung der Konfigurationsparameter (siehe Abschnitt 5.3.3.3) in eine Datenbank geschrieben. Nach der Simulation und Optimierung werden in einer separaten CSV-Datei das Ergebnis der Optimierung und die optimalen Konfigurationsparameter gespeichert. Für diese Fallstudie zeigt Tabelle 8 die optimalen Konfigurationsparameter.

Tabelle 8 Ergebnis der bayesschen Optimierung mit dem Tree-Parzen Schätzer

Parameter	**Parameterraum**	**Optimaler Wert**
Aggregations-mechanismus	FedAvg FedOpt FedAdam FaultTolerantFedAvg QFedAvg	QFedAvg
Client-Anzahl	[2, 8]	4
Lokale Epochen	[1, 20]	12
Batch-Größe	[16, 128]	32

Mit den in Tabelle 8 ermittelten Konfigurationsparametern wird ein gesamter F1-Score von 0,93 erreicht – eine Annäherung an den F1-Score von 0,95. Für den Vergleich des zentralen und dezentralen Trainings mit FL wurde die in Abschnitt 5.3.3.2 beschriebene Methode angewandt, um die Verbesserung des F1-Scores auf Grundlage der lokalen F1-

Scores zu ermitteln. In Abbildung 6-22 ist ein Balkendiagramm dargestellt, das die F1-Scores der einzelnen Clients im dezentralen Training, den F1-Score auf Basis der ermittelten optimalen Konfigurationsparameter von FL sowie das zentrale Training einander gegenüberstellt. Deutlich wird, dass sich der Ansatz mit FL (hellblau) dem zentralen Training erkennbar stark annähert. Für die Datenpartitionen der Clients liegt der durchschnittliche F1-Score bei ca. 0,91 und damit geringfügig niedriger als für den Ansatz mit FL oder das zentrale Training des Modells. Für die Erkennung von Leckagen in Druckluftsystemen bei pneumatischen Aktorbewegungen ist der Ansatz mit FL auf Basis des F1-Scores besser als das rein dezentrale Training. Allerdings ist aus Sicht des Anwenders ein F1-Score von 0,91 bereits ein ausreichend gutes Ergebnis. Dennoch konnte durch FL das Ergebnis nochmals gesteigert werden und ermöglicht somit eine im Mittel bessere energetische Anomalieerkennung.

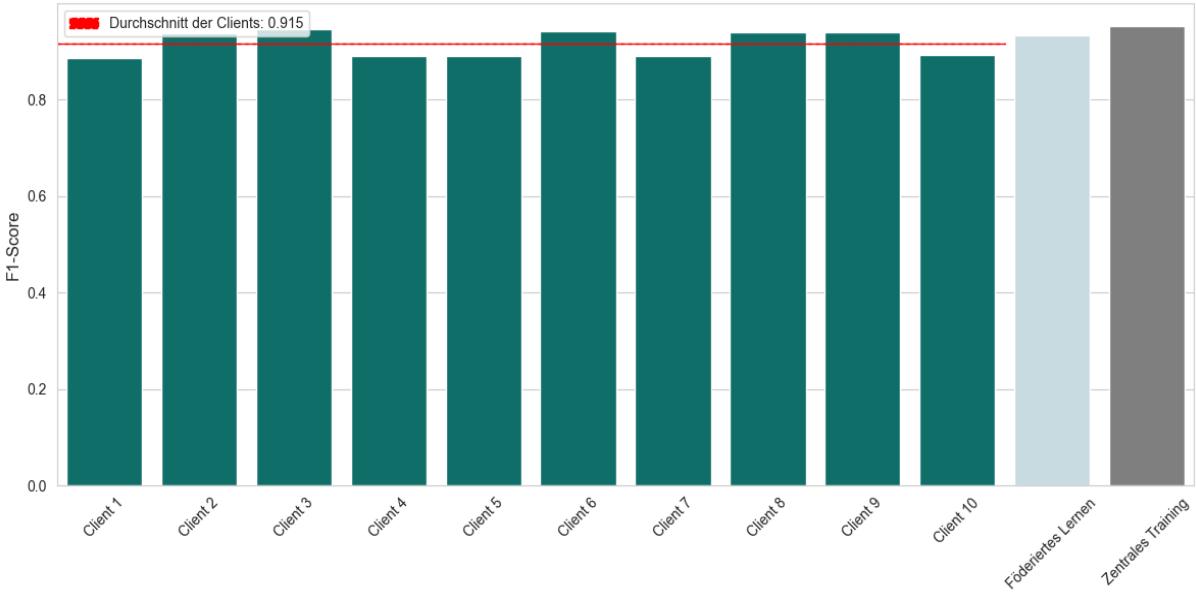


Abbildung 6-22 Darstellung des Vergleichs der Evaluationsmetrik zwischen dezentralem Training, zentralem Training und FL

Abbildung 6-23 kann die durchschnittliche Entwicklung des F1-Scores und der Fairness entnommen werden. Auf der linken Y-Achse ist der F1-Score (grün), auf der rechten Y-Achse hingegen die Fairness als Standardabweichung der F1-Scores aller lokalen Clients aufgetragen. Hierbei ist zu beachten, dass eine niedrige Standardabweichung einer besseren Fairness entspricht. Der F1-Score steigt von 0,45 in der ersten Runde stetig an.

Insbesondere zeigt die Abbildung einen starken Anstieg des F1-Scores bis zur sechsten Runde, in der der F1-Score bereits 0,84 erreicht. Danach verlangsamt sich das Wachstum, bleibt jedoch weiterhin positiv. In der letzten Runde erreicht er 0,94 – eine deutliche Verbesserung gegenüber dem Ausgangswert.

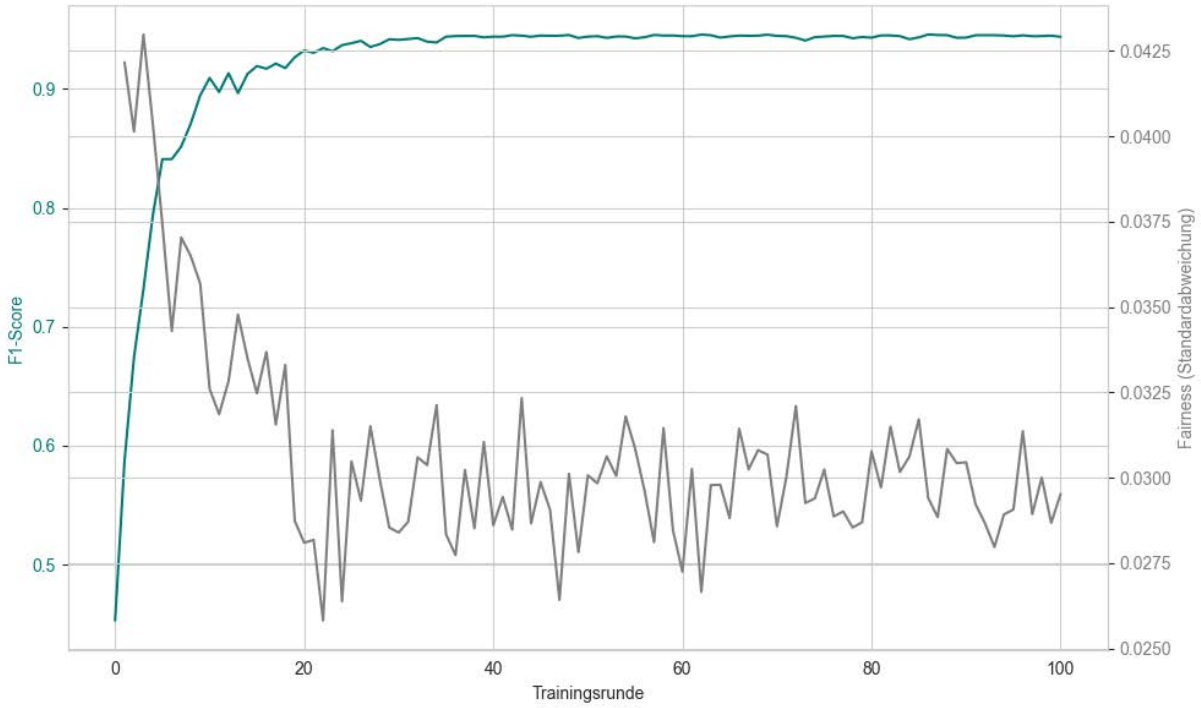


Abbildung 6-23 Durchschnittliche Entwicklung von F1-Scores und Fairness pro Runde

Abbildung 6-23 zeigt die Fairness über die Trainingsrunden von FL. Der Trend belegt, dass die Fairness im Allgemeinen zunimmt, dargestellt durch die Verringerung der Standardabweichung zwischen den Clients. Dies zeigt, dass der Ansatz mit FL über die Runden hinweg fairer wird und die Modelle der verschiedenen Clients konvergieren. Anfangs liegt die Standardabweichung bei etwa 0,042. Über die Runden hinweg zeigt sich ein allgemeiner Abwärtstrend, was darauf hindeutet, dass die Leistung zwischen den Clients konsistenter wird. In den letzten Runden stabilisiert sich die Standardabweichung bei etwa 0,03.

Abbildung 6-24 zeigt die Verteilung der F1-Scores, wobei diese Werte auf den letzten zehn Werten jeder Konfiguration basieren und der gleitende Mittelwert berechnet wurde (siehe Abschnitt 5.3.3.2). Das Histogramm stellt die Häufigkeitsverteilung der F1-Scores dar und verdeutlicht, welche F1-Scores am häufigsten auftreten. Für eine bessere

Veranschaulichung wird die Kernel-Dichte-Schätzung genutzt, die die beobachteten Werte in einer kontinuierlichen Wahrscheinlichkeitsdichtekurve visualisiert. Die X-Achse stellt die verschiedenen F1-Scores dar, während die Y-Achse die Anzahl der Konfigurationen zeigt, die jeweils einen bestimmten F1-Score erreicht haben.

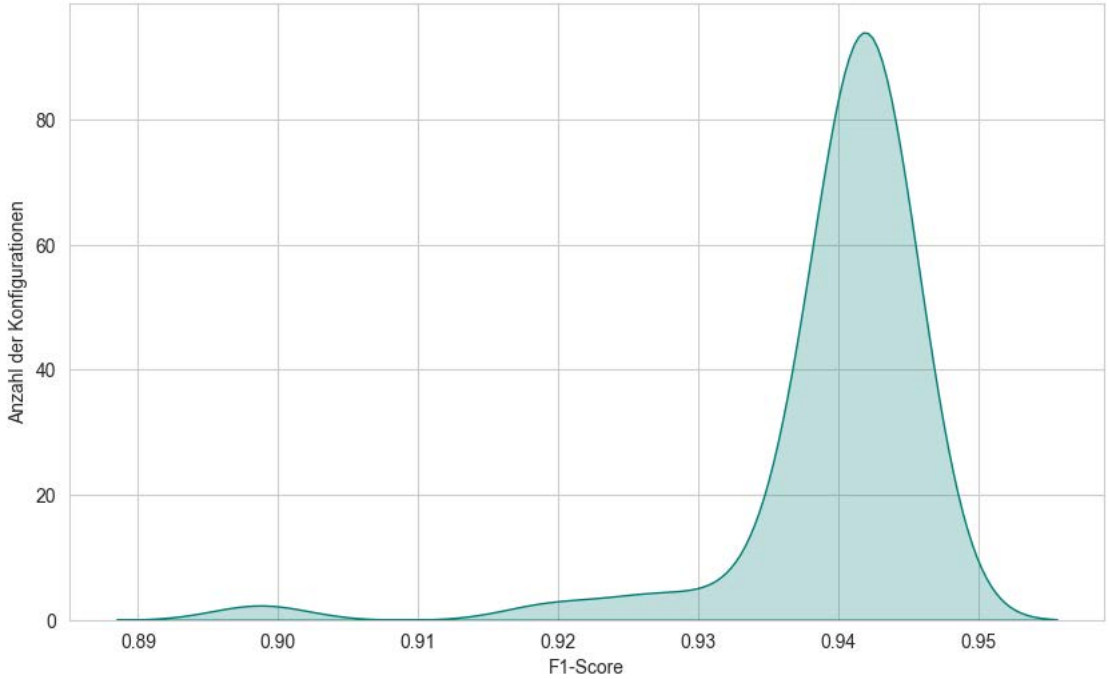


Abbildung 6-24 Verteilung des F1-Scores

Das Histogramm offenbart eine deutliche, fast glockenförmige Verteilung der F1-Scores. Die meisten Konfigurationen fallen in den Bereich von etwa 0,92 bis 0,95, nur ein Wert – in Versuch 17 – liegt leicht darunter, mit einem F1-Score von 0,89.

Die dichte Verteilung im Bereich von 0,92 bis 0,95 zeigt an, dass die meisten Konfigurationen ähnliche und hohe F1-Scores erreichen. Die meisten Konfigurationen führen zu ähnlichen und hohen F1-Scores. Die Einflüsse und Zusammenhänge der einzelnen Konfigurationsparameter auf den F1-Score und die Fairness sollen im nächsten Abschnitt weiter analysiert werden.

6.2.4.3 Analyse und Bewertung der Konfigurationsparameter

Abbildung 6-25 zeigt den Verlauf des mittleren F1-Scores in Abhängigkeit vom Aggregationsmechanismus. Die verschiedenen Aggregationsmechanismen sind farblich unterschiedlich gekennzeichnet.

QFedAvg beginnt bei 0,43 und zeigt einen stetigen Anstieg über die Runden, mit einem allmählichen Anstieg bis zu einem Wert von 0,95 am Ende. FedAvg zeigt eine robuste Leistung, beginnt mit einem F1-Score von 0,48, die schnell zu einem stabilen Wert von 0,95 konvergiert. Der Aggregationsmechanismus FaultTolerantFedAvg startet mit einem F1-Score von 0,4, steigt schnell an und stabilisiert sich in den späteren Runden bei Werten um 0,94. Der F1-Score von FedAdam variiert von einem Tiefstwert (0,38) in der ersten Runde bis zu Werten um 0,91, zeigt aber in den späteren Runden wieder Schwankungen, wobei der Score in Runde 73 auf 0,87 fällt. FedOpt hat einen Startwert von 0,55 und zeigt einen anfänglichen Anstieg, gefolgt von einigen Schwankungen, erreicht aber in späteren Runden Werte um 0,94.

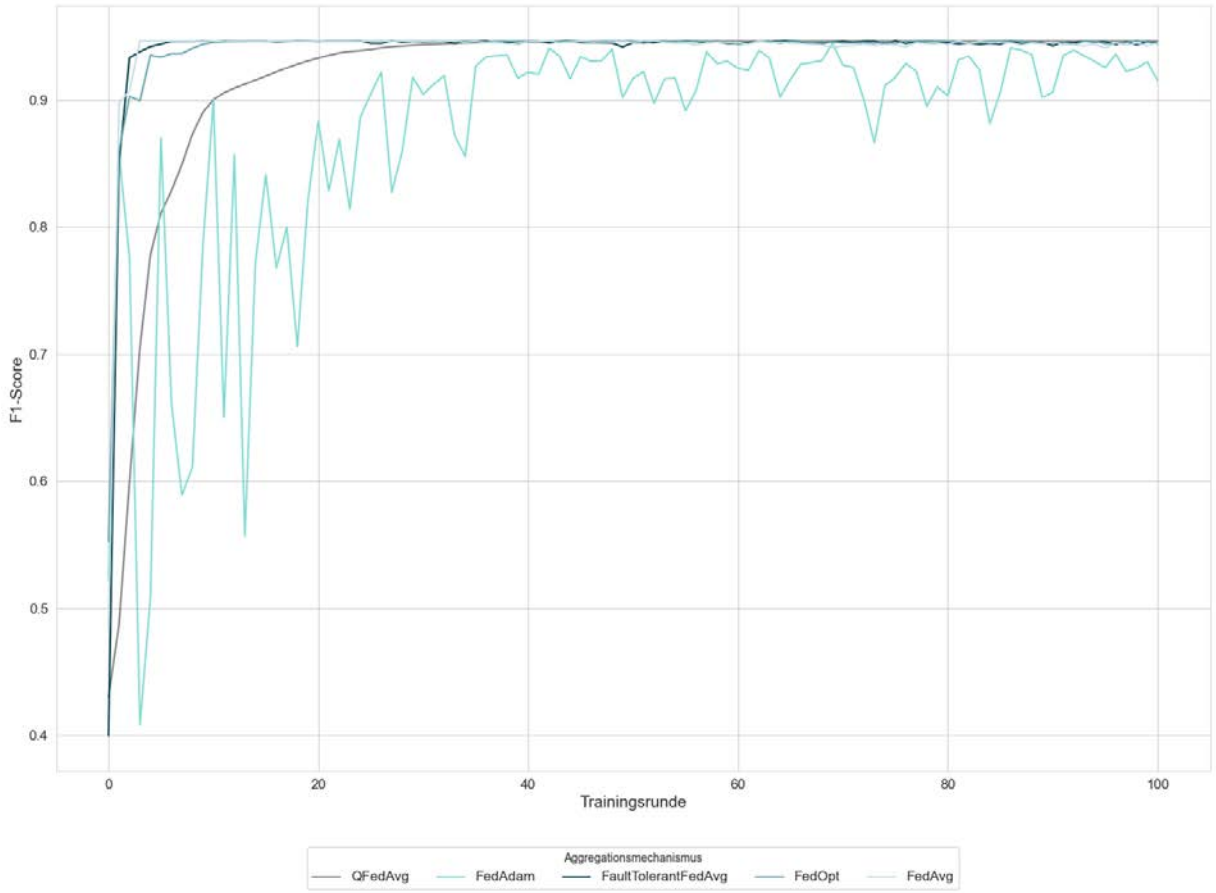


Abbildung 6-25 Durchschnittliche Entwicklung des F1-Scores für jeden Aggregationsmechanismus

Die meisten Aggregationsmechanismen zeigen nach anfänglichen Schwankungen eine Stabilisierung des F1-Scores. FedAvg und QFedAvg zeigen die schnellste und vor allem stabilste Konvergenz aller Aggregationsmechanismen. Während FedAdam anfänglich nicht konvergiert, erreicht es schließlich ähnliche F1-Scores wie die anderen Aggregationsmechanismen. Diese Schwankungen, insbesondere bei FedAdam, könnten auf eine Reihe von Faktoren zurückzuführen sein – einschließlich des Lernverhaltens des Mechanismus, der Behandlung von Ausreißern in den Daten oder der Art und Weise, wie die Gewichte der verschiedenen lokalen Clients aggregiert werden.

Werden $\overline{F_R}$ und $\overline{Fair_R}$ mit $R = 10$ zusammenhängend betrachtet (siehe Abbildung 6-26), zeigt sich, dass die Gesamtverteilung der Aggregationsmechanismen sowohl beim F1-Score als auch bei der Fairness in einem engen Bereich liegt. Insbesondere bewegen sich

die durchschnittlichen F1-Scores aller Mechanismen in einem Intervall zwischen 0,90 und 0,94.

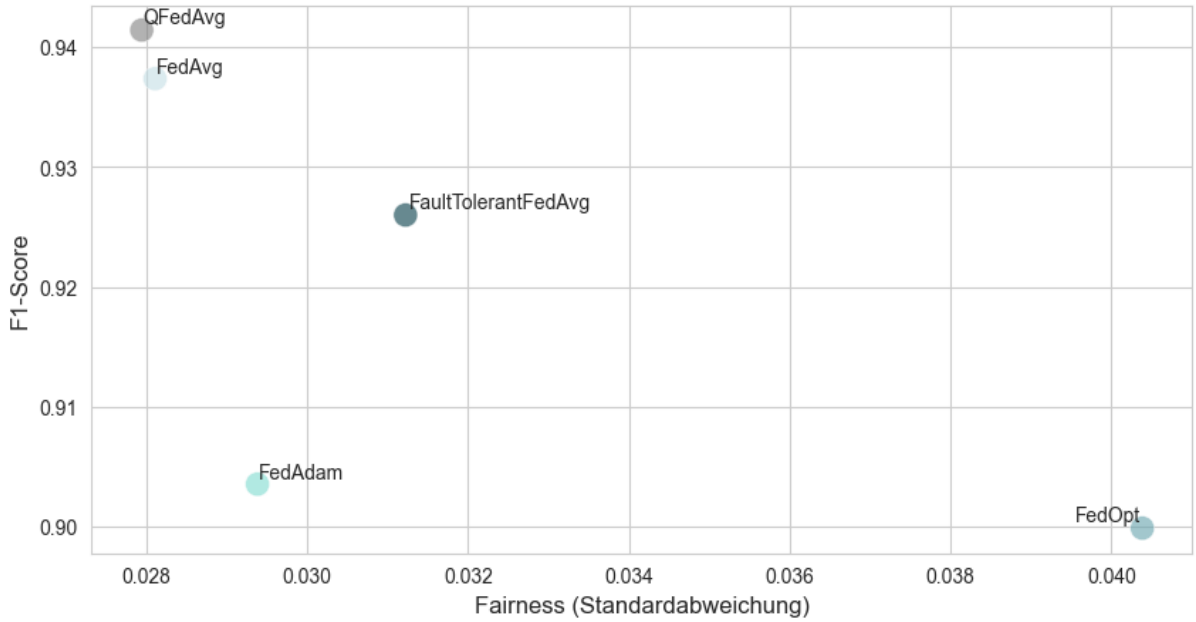


Abbildung 6-26 Darstellung der einzelnen Strategien in Relation zu F1-Score und Fairness

Eine nähere Betrachtung der Fairness-Werte, die durch die Standardabweichung repräsentiert werden, zeigt eine leichte, aber deutliche Differenzierung zwischen den Mechanismen. Die Spanne der Standardabweichungen liegt zwischen 0,028 und 0,040, was darauf hindeutet, dass es trotz der engen Spanne der F1-Scores Unterschiede in der Standardabweichung der Ergebnisse zwischen den verschiedenen Datenquellen gibt.

Besonders auffällig ist die Position von QFedAvg und FedAvg in der Abbildung. Diese Mechanismen liegen im linken oberen Quadranten – ein Zeichen dafür, dass sie nicht nur hohe F1-Scores, sondern auch eine höhere Fairness aufweisen. Demgegenüber hat FedOpt mit einer Standardabweichung von 0,040 die geringste Fairness unter den betrachteten Mechanismen. Dies könnte auf inkonsistentere Ergebnisse bei unterschiedlichen Datenquellen hindeuten.

Abbildung 6-27 zeigt den Zusammenhang zwischen der Anzahl lokaler Epochen auf der X-Achse und dem F1-Score sowie der Fairness auf der Y-Achse. Die Analyse der Daten liefert mehrere Beobachtungen. Der F1-Score zeigt für die meisten lokalen Epochen eine stabile Tendenz mit einigen Schwankungen.

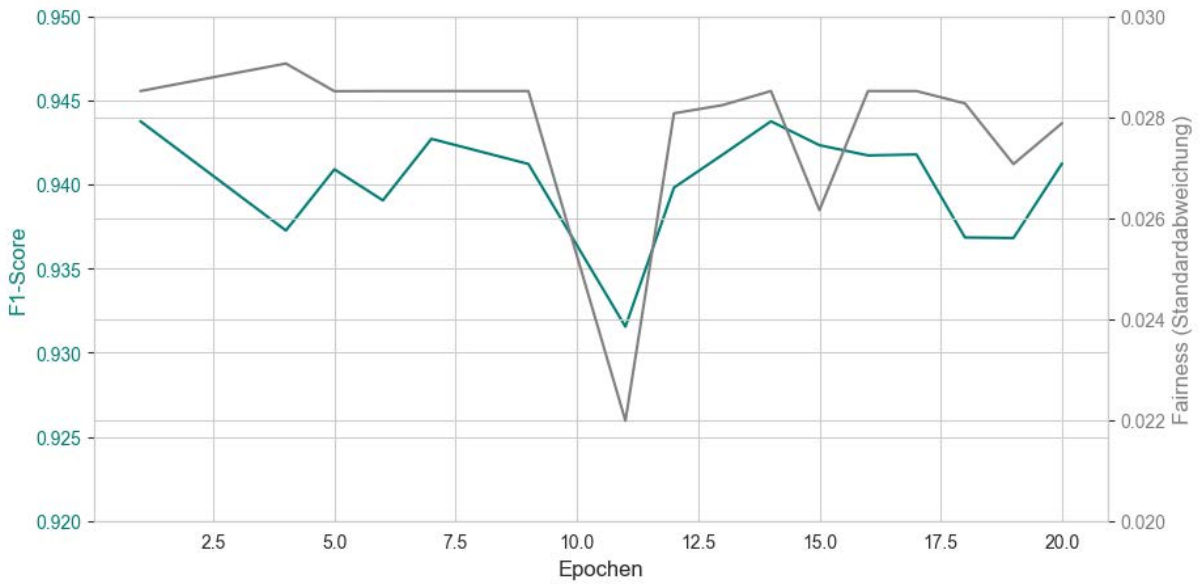


Abbildung 6-27 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit von den Epochen

Die Variation des F1-Scores in Abhängigkeit von den lokalen Epochen zeigt, dass die Anzahl Letzterer nicht immer direkt proportional zum F1-Score ist. Während in einigen Fällen, z. B. bei fünf der Epochen, ein relativ hoher F1-Score erreicht wird, nimmt der Wert bei elf Epochen deutlich ab. Der höchste F1-Score wird bei 13 Epochen erreicht; bei 19 Epochen kommt es zu einer deutlichen Abnahme des Scores. Insgesamt kann festgestellt werden, dass die Daten dazu tendieren, in einem bestimmten Bereich zu liegen, auch wenn es einige Ausreißer gibt.

Grundsätzlich ist davon auszugehen, dass das Training des Modells nach einer gewissen Anzahl lokaler Epochen konvergiert und zusätzliche Epochen nicht immer zu einer Verbesserung des F1-Scores führen. Dies wird durch den höheren F1-Score bei sieben der Epochen im Vergleich zu den Werten bei einer höheren Anzahl von Epochen unterstützt. Es ist möglich, dass die aus der Optimierung resultierenden zwölf Epochen im Zusammenhang mit anderen Konfigurationsparametern einen guten Kompromiss zwischen F1-Score und Fairness darstellen. Die Fairness und der F1-Score zeigen im Allgemeinen keine direkt proportionale Beziehung. Während der F1-Score in der Regel über die verschiedenen Epochen stabil bleibt, gibt es Epochen, in denen trotz hoher Fairness der F1-Score sinkt – und umgekehrt.

Abbildung 6-28 zeigt den Verlauf des durchschnittlichen F1-Scores und der Standardabweichung des F1-Scores der Clients in Abhängigkeit von der Batch-Größe. Bei einem Anstieg Letzterer von 16 bis 64 ist eine allgemeine Zunahme des F1-Scores zu beobachten.

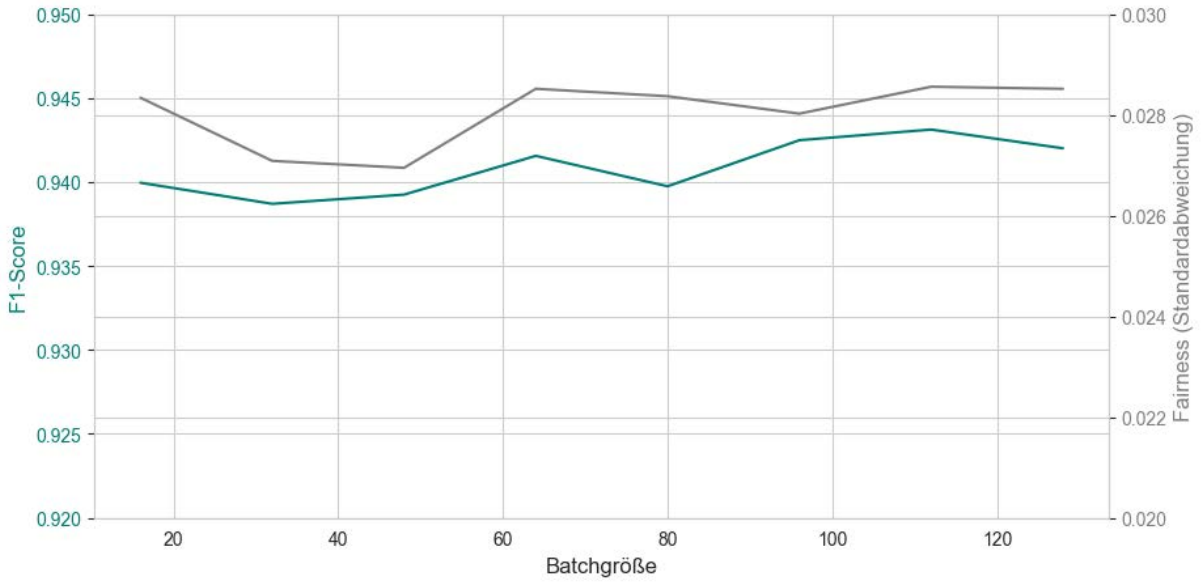


Abbildung 6-28 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit von der Batch-Größe

Nach dem Höhepunkt bei einer Batch-Größe von 64 nehmen der F1-Score und die Fairness bei den darauffolgenden Batch-Größen nur leicht ab. Die Batch-Größe 80 zeigt einen leichten Abfall im F1-Score, jedoch nicht den niedrigsten Punkt. Der F1-Score erholt sich bei den Batch-Größen 96 und 112 wieder, fällt aber bei 128 minimal. Die Fairness bleibt im Allgemeinen stabil, mit leichten Schwankungen über die gesamte Bandbreite der Batch-Größen.

Anhand der Visualisierung existiert kein starker linearer Zusammenhang zwischen der Batch-Größe und dem F1-Score. Während der F1-Score von 16 bis 64 ansteigt, zeigt er danach einen nichtmonotonen Trend. Der Fairnesswert bleibt über den gesamten Bereich der Batch-Größen relativ konstant, was darauf hindeutet, dass Letztere keinen Einfluss auf die Fairness haben.

Die Wahl der Batch-Größe scheint indes einen wichtigen nichtlinearen Einfluss auf den F1-Score auszuüben. Wie in Abbildung 6-28 dargestellt, steigt der F1-Score zunächst mit

zunehmender Batch-Größe an, zeigt aber nach Erreichen eines optimalen Wertes von 64 eine gewisse Volatilität. Diese Beobachtung könnte bedeuten, dass eine mittlere Batch-Größe, in diesem Fall 32–96, für den F1-Score optimal ist. Niedrigere oder höhere Batch-Größen führen nicht unbedingt zu besseren Ergebnissen. Dass die Fairness über die verschiedenen Batch-Größen konstant bleibt, legt nahe, dass die Batch-Größe keinen wesentlichen Einfluss auf die Fairness hat. Wie bereits in der ersten Fallstudie erwähnt, weisen sowohl kleine als auch große Batch-Größen nicht konsequent hohe F1-Scores auf.

Abbildung 6-29 zeigt den durchschnittlichen F1-Score und die Fairness in Abhängigkeit vom Anteil der im Training befindlichen Clients. Der F1-Score erhöht sich im Allgemeinen, wenn der Anteil von 0,2 auf 0,7 ansteigt, und erreicht bei 0,7 ein Maximum. Danach sinkt er bei einem Anteil von 0,8. Die Standardabweichung ist bei einem Anteil von 0,8 am geringsten, was auf eine höhere Fairness hindeutet. Die höchste Standardabweichung, d. h. die geringste Fairness, wird bei einem Bruchteil von 0,2 beobachtet.

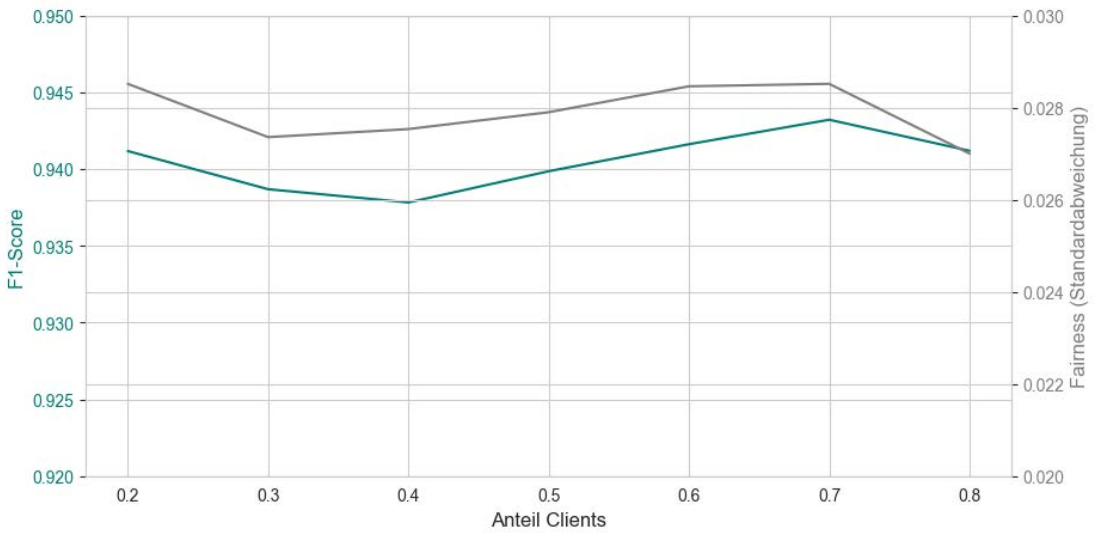


Abbildung 6-29 Visualisierung des F1-Scores und der Standardabweichung in Abhängigkeit des Anteils der Clients

Die Wahl des Anteils der trainierenden Clients wirkt sich sowohl auf den F1-Score als auch auf die Fairness des Modells aus. Ein Anteil von 0,4 bis 0,7 scheint ein Gleichgewicht zwischen einem guten F1-Score und akzeptabler Fairness zu bieten, während ein Anteil von 0,8 die höchste Fairness auf Kosten eines etwas niedrigeren F1-Scores realisiert. Dies

könnte darauf hindeuten, dass es – sofern gewünscht ist, die Fairness unter den Clients zu maximieren – sinnvoll sein könnte, einen größeren Anteil der Clients in das Training einzubeziehen, selbst wenn dies zu einem etwas niedrigeren F1-Score führt.

Wie in der ersten Fallstudie ist der gesamte Rechenaufwand, der mit der Teilnahme einer größeren Anzahl von Clients verbunden ist, höher. Mehr Clients bedeuten, dass mehr Sensoren gleichzeitig trainiert werden müssen, was zu höheren Kosten, höherem Energieverbrauch und anderen Herausforderungen wie z. B. Kommunikationsaufwand führen kann. Daher wäre es sinnvoll, die Anzahl der Clients so zu optimieren, dass ein guter Kompromiss zwischen F1-Score, Fairness und Energieverbrauch gefunden wird. Ein solcher Ansatz würde den Gesamtaufwand und die Komplexität des Trainingsprozesses reduzieren und gleichzeitig einen hohen F1-Score aufrechterhalten.

Im nächsten Schritt wurde mithilfe des Pearson-Korrelationskoeffzienten untersucht, inwiefern ein linearer Zusammenhang zwischen den verschiedenen Konfigurationsparametern und dem F1-Score besteht (siehe Abbildung 6-30). Der Wertebereich der Korrelationen liegt zwischen –1 und 1, wobei ein Wert nahe 1 eine starke positive Korrelation, ein Wert nahe –1 eine starke negative Korrelation und ein Wert nahe 0 keine oder eine sehr schwache Korrelation anzeigt.

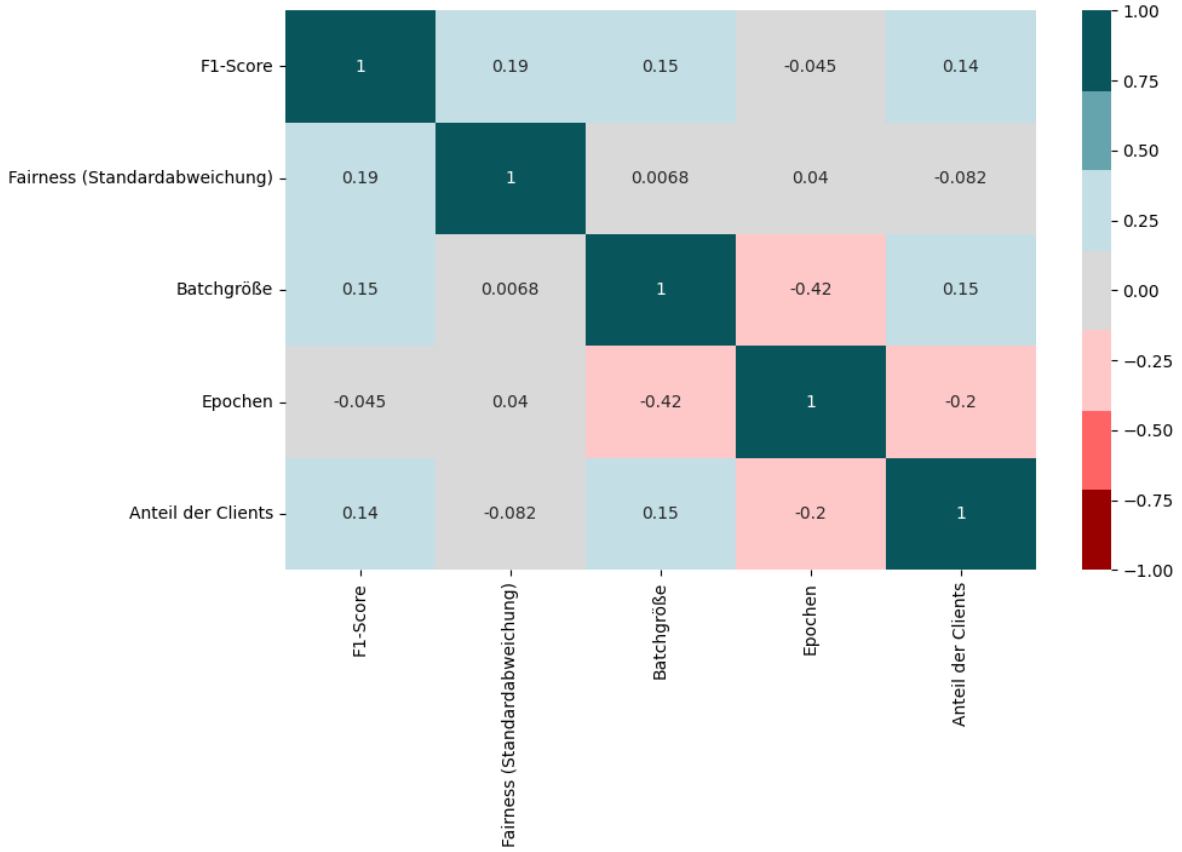


Abbildung 6-30 Korrelationsmatrix dargestellt als Heatmap der Konfigurationsparameter, F1-Score und Fairness

Es besteht eine positive Korrelation von 0,19 zwischen dem F1-Score und der Fairness (gemessen als Standardabweichung). Das bedeutet, dass mit steigendem F1-Score auch die Standardabweichung steigt, was auf eine Abnahme der Fairness schließen lässt. Der positive Zusammenhang zwischen F1-Score und Fairness (Standardabweichung) weist auf ein mögliches Problem hin: Ein höherer F1-Score ist mit einer geringeren Fairness (höhere Standardabweichung) verbunden. Das bedeutet, dass die Modelle zwar insgesamt gut abschneiden können, dies aber auf Kosten der Fairness zwischen den einzelnen Clients geht. Der F1-Score zeigt nur schwach positive Korrelationen mit den Konfigurationsparametern *Batch-Größe* (0,15) und *Anteil der am Training teilnehmenden Clients* (0,14).

Die Fairness (Standardabweichung) zeigt generell schwache Korrelationen mit den Konfigurationsparametern. Die geringen Korrelationen zwischen den Konfigurationsparametern und der Fairness legen nahe, dass allein die Wahl dieser

Parameter möglicherweise nicht ausreicht, um optimale Fairness zu gewährleisten. Möglicherweise spielen auch andere Faktoren oder eine Kombination von Parametern eine Rolle.

Interessanterweise besteht eine negative Korrelation von –0,42 zwischen der Anzahl der Epochen und der Batch-Größe, was nahelegt, dass mit zunehmender Batch-Größe die optimale Anzahl der Epochen tendenziell abnimmt. Da es jedoch um Konfigurationsparameter geht, die sich aus der bayesschen Optimierung ergeben, wurde implizit ein Zusammenhang zwischen Batch-Größe und Epochen identifiziert. Wird die Batch-Größe erhöht, können in jeder Iteration mehr Datenpunkte gleichzeitig verarbeitet werden. Das Modell erhält mit jedem Schritt eine umfangreichere Ansicht der Daten und kann schneller konvergieren. Folglich könnte es erforderlich sein, weniger Epochen zu nutzen, um vergleichbare (oder sogar bessere) F1-Scores zu erzielen, da das Modell in jeder Epoche mehr Daten betrachtet. Dies könnte die negative Korrelation erklären. Eine Erhöhung der Batch-Größe könnte dazu führen, dass weniger Epochen benötigt werden, um eine Konvergenz zu erreichen.

Die Korrelationen des Clientanteils mit anderen Parametern sind im Allgemeinen schwach, was darauf hindeutet, dass der Anteil der am Training beteiligten Clients keinen starken linearen Einfluss auf die anderen Parameter oder Metriken hat.

6.3 Zwischenfazit

In diesem Kapitel wurden zwei Anwendungsfälle vorgestellt, um die Anwendbarkeit der in Kapitel 5 entwickelten Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen zu validieren. Konkret wurden energetische Anomalien in zwei unterschiedlichen Kontexten untersucht: die Erkennung von Bedienfehlern beim Lichtbogenschweißen und die Identifikation von Leckagen im Druckluftsystem bei pneumatischen Aktorbewegungen.

Im Rahmen des Anwendungsfalls zur Erkennung von Bedienfehlern beim Lichtbogenschweißen wurde in der ersten Phase ein LSTMFCN als das Modell mit dem

höchsten LEI von 0,83 ausgewählt. Darüber hinaus konnte gezeigt werden, dass die Augmentation der 200 Schweißungen signifikanten Einfluss auf den LEI hat, sodass die augmentierten Daten (2200 Schweißungen) für die weiteren Phasen verwendet werden können. Interessanterweise zeigte sich, dass eine Abtastrate von 2 Hz für die energetische Anomalieerkennung den höchsten LEI aufweist. Die Implementierung des Anomaliedetektors auf einem Einplatinencomputer, der an der Schweißstromquelle angebracht war, ermöglichte eine sofortige Auswertung und Reaktion nach jeder Schweißung. Die Reaktion erfolgt durch zwei Agenten. Der erste Agent ist für die Protokollierung der Modellergebnisse in einem lokalen Skript verantwortlich, das auf dem Einplatinencomputer ausgeführt wird. Der zweite Agent visualisiert die Ergebnisse der energetischen Anomalieerkennung in einem Dashboard. Die Anwendung des FL in diesem Kontext zeigte deutliche Vorteile gegenüber einem rein dezentralen Trainingsansatz. Während der durchschnittliche F1-Score beim dezentralen Ansatz bei 0,46 lag, erreichte er mit dem FL-Ansatz 0,81 und näherte sich damit dem zentralen Ansatz mit einem F1-Score von 0,86 an.

Im zweiten Anwendungsfall sollten Druckluftleckagen durch die Analyse von Massenstromdaten aus pneumatischen Aktorbewegungen erkannt werden. Dazu wurden vier verschiedene Szenarien betrachtet: keine Leckage und Leckage mit einem Durchmesser von 1 bis 3 mm. Um ein geeignetes Modell zu finden, wurden insgesamt 550 Modelle trainiert. Dabei erzielte das CNN mit einer Abtastrate von 1 Hz den höchsten LEI (0,95). Bei der Parameteroptimierung für das FL zeigten die Simulationen, dass der durchschnittliche F1-Score bei einer rein dezentralen Datenpartitionierung der Clients bei ca. 0,91 lag. Obwohl dieser Wert bereits hoch ist, konnte durch den Einsatz von FL der F1-Score weiter auf 0,93 gesteigert werden und nähert sich damit dem rein zentralen Ansatz mit einem F1-Score von 0,95 an.

Hinsichtlich der Modellentwicklung wurde festgestellt, dass die Einbeziehung zusätzlicher prozessabhängiger Merkmale, die auf Stromstärke und Spannung basieren, insbesondere beim Lichtbogenschweißen effektiv ist. Beispielsweise erwiesen sich die mittlere Kurzschlussfrequenz und die quadratischen Mittelwerte der Stromstärke und Spannung

als relevante Merkmale. Bei Datensätzen mit relativ wenigen Schweißungen (200) wurde deutlich, dass eine Augmentation mit Augmentationsmechanismen für Zeitreihen die Leistungsmetrik signifikant und modellunabhängig erhöht. Außerdem ist hervorzuheben, dass der Verzicht auf die Transformation von Zeitreihen in kontextbasierte Merkmalstabellen einen deutlichen Vorteil bietet. Dies wird dadurch bestätigt, dass Modelle, die kontextbasierte Merkmale verwenden, überwiegend schlechtere F1-Scores zeigen als Modelle, die auf kontextbasierte Energiezeitreihengruppen zurückgreifen. Dies wurde auch durch einen gepaarten t-Test bestätigt und ist darauf zurückzuführen, dass die Merkmalstabellen Merkmale anhand der Zeitreihe vordefiniert extrahieren, jedoch die grundlegenden Strukturen und die Zeitabhängigkeit der Daten nicht mehr verfügbar ist. Modellarchitekturen mit Komponenten wie dem LSTM oder das CNN sind in der Lage, die Zeitdimension ohne Vorverarbeitung und Merkmalsextraktion mit in die Approximation der Modellparameter einzubeziehen.

Zusätzlich zeigt die Analyse der verschiedenen Modellarchitekturen, dass komplexe Modelle (z. B. ResNet) zwar tendenziell die beste Evaluationsmetrik in Bezug auf den F1-Score aufweisen, der damit verbundene Trainingsaufwand in Bezug auf Zeit, Rechenkapazität und Energieverbrauch jedoch stark variiert und oft keinen zusätzlichen Mehrwert in Relation zum Trainingsaufwand bietet. Es wird empfohlen, sich zunächst auf einfachere Modellarchitekturen zu konzentrieren und erst dann zu ressourcenintensiveren Modellen überzugehen, wenn die Anforderungen an die Evaluationsmetrik nicht erfüllt werden.

Darüber hinaus hat die Untersuchung der Abtastraten gezeigt, dass eine Erhöhung der Abtastraten nicht zwangsläufig zu einer Verbesserung der energetischen Anomalieerkennung führt. Es sollte daher sorgfältig abgewogen werden, ob der mit einer Abtastrateerhöhung verbundene zusätzliche Speicherbedarf, die Rechenleistung und damit die Kosten und der Energieverbrauch tatsächlich gerechtfertigt sind. Um den Aufwand in die Bewertung einzubeziehen, wurde der LEI eingeführt. Die Verwendung desselben ermöglicht die Berücksichtigung der Leistungsmetrik und der Trainingszeit. Da die Abtastrate mit der Trainingszeit des Modells zusammenhängt, fließt die Trainingszeit implizit in die Bewertung über den LEI ein.

In Bezug auf die Implementierung des Anomaliedetektors haben die Untersuchungen gezeigt, dass aus der ersten Phase der Methode Funktionen und Komponenten in die zweite Phase übernommen werden können. Dies betrifft insbesondere Datenverarbeitungseinheiten wie die Extraktion von Merkmalen oder die Integration von Daten aus verschiedenen -quellen. Während für die Implementierung einfacher Modellarchitekturen in der Regel ein Einplatinencomputer ausreicht, können Computer mit eingeschränkten Hardwarespezifikationen besondere Herausforderungen darstellen, da einige Softwarebibliotheken nicht unterstützt werden, was individuelle Lösungen erforderlich macht.

Bei der Optimierung der Parameter von FL kann die Anwendbarkeit von FL für die energetische Anomalieerkennung durch simulationsbasierte Experimente mit den verfügbaren Daten und dem vorausgewählten Modell aus Phase I demonstriert werden. Der simulationsbasierte Ansatz ermöglicht eine hardwareunabhängige Simulation und Optimierung. So ist es z. B. nicht notwendig, eine Vielzahl von Geräten – z. B. mehrere Einplatinencomputer – für die Evaluierung zu verwenden. In beiden Fallstudien konnte gezeigt werden, dass der Einsatz von FL zu einem höheren F1-Score führte als das Training der Clients auf dem lokalen Datensatz. Zudem nähert sich der F1-Score beim Einsatz von FL dem F1-Score im Szenario mit vollständiger Datenverfügbarkeit an. Dies bedeutet, dass ein adäquates Ergebnis für den Maschinen- und Anlagenbauer auch dann möglich ist, wenn die Anwender dem Maschinen- und Anlagenbauer keinen Zugriff auf die Daten gewähren.

Ein zentrales Ergebnis der Analyse und Bewertung ist die Rolle der Aggregationsmechanismen. Die Ergebnisse zeigen, dass diese Aggregationsmechanismen den größten Einfluss auf den F1-Score und die Fairness des Systems haben. Darüber hinaus zeigt sich, dass die Konfigurationsparameter keinen direkten linearen Zusammenhang zum F1-Score und zur Fairness aufweisen. Die Untersuchung von FL ergab jedoch eine mittlere Korrelation zwischen F1-Score und Fairness: Es wurde festgestellt, dass der F1-Score und die Standardabweichung eine positive Korrelation aufweisen, was eine negative Korrelation mit der Fairness impliziert. Dies bedeutet, dass

mit steigendem F1-Score auch die Standardabweichung der Ergebnisse zwischen den verschiedenen Clients zunimmt.

Dies unterstreicht einen wichtigen Aspekt bei der Anwendung von FL: Häufig erfolgt eine Abwägung zwischen der eigentlichen Evaluationsmetrik und der Fairness. Während ML-Modelle nicht selten ausschließlich auf Basis des F1-Scores, des Recalls, der Präzision oder der Genauigkeit optimiert werden, ist es unerlässlich, auch die Fairness zu berücksichtigen – insbesondere bei Anwendungen, in denen Mindestanforderungen an den F1-Score für alle Nutzer gelten (z. B. eine Schweißstromquelle). Die Herausforderung besteht darin, ein Gleichgewicht zu finden, das sowohl hohe Leistung als auch akzeptable Fairness gewährleistet.

Insgesamt ist festzuhalten, dass die beiden Anwendungsfälle aus der Praxis die intersubjektive Anwendbarkeit und den Nutzen der beiden entwickelten Artefakte gezeigt haben. Durch die beiden Fallstudien konnte zudem gezeigt werden, dass das entwickelte FL-basierte System zur Erkennung energetischer Anomalien in Produktionsprozessen sowohl anwendbar als auch reproduzierbar ist. Die umfassende Validierung und die systematische Analyse der Ergebnisse tragen zur Generalisierbarkeit bei. Die Verwendung standardisierter Evaluationsmetriken, die detaillierte Methodenentwicklung sowie die Berücksichtigung mehrerer unabhängiger Variablen und Konfigurationen gewährleisten die Nachvollziehbarkeit und Reproduzierbarkeit der Ergebnisse, was die wissenschaftliche Robustheit und Praxisrelevanz unterstreicht Die Anwendungsfälle zeigen das Potenzial von FL zur Erkennung energetischer Anomalien in Produktionsprozessen und bieten die Möglichkeit für weitere Untersuchungen.

7 Reflexion

Nach der Entwicklung der zwei Artefakte zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen in Kapitel 5 sowie der Implementierung und Validierung in Kapitel 6 können die Grenzen der Artefakte hinsichtlich der Anforderungen aufgezeigt sowie die Forschungsfragen beantwortet werden. In diesem Kapitel erfolgt die Reflexion hinsichtlich der Erfüllung der in Abschnitt 4.1 definierten Anforderungen sowie der in Abschnitt 1.3 formulierten grundlegende Forschungsfrage.

7.1 Bewertung der Anforderungserfüllung

Angelehnt an Abschnitt 4.1 sollen vorliegend die Bewertungen der Anforderungserfüllung auch in die jeweiligen Kategorien unterteilt werden. Die Anforderungen wurden in drei Gruppen strukturiert: *Anforderungen an die energetische Anomalieerkennung*, *Anforderungen an FL* und *Anforderungen an die verteilte Systemarchitektur*. Die Bewertung der Anforderungserfüllung zeigt, dass die Umsetzung der Anforderungen erfolgreich war und somit die Anforderungen erfüllt wurden.

7.1.1 Anforderungen an die energetische Anomalieerkennung

Für die Anomalieerkennung aus Energiedaten ist die Berücksichtigung von Zeitreihen notwendig, da Sensordaten von Maschinen und Anlagen eine zeitliche Abhängigkeit aufweisen. Bei der Auswahl von Modellen ist daher die *Eignung für Zeitreihen* zu prüfen. Durch den Fokus auf temporal abhängige Energiedaten von Produktionsprozessen und die Extraktion von Merkmalen aus multivariaten Energiezeitreihen wurde der in dieser Arbeit vorgestellte Ansatz spezifisch auf Energiezeitreihen angepasst. Zusätzlich konnten

kontextbasierte Betriebsdaten für die Segmentierung der multivariaten Energiezeitreihe genutzt werden. Daher ist die Anforderung vollständig erfüllt.

Für die praxisorientierte Umsetzung ist zusätzlich die Definition und Ausführung von Gegenmaßnahmen erforderlich. Die Anforderung der gezielten *Maßnahmenintegration* in die energetische Anomalieerkennung zeichnet erst den Mehrwert der Nutzung von Methoden aus dem Bereich der künstlichen Intelligenz aus und ist somit ein wichtiger Faktor für den Erfolg eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen. Eine Klärung dessen, welche Maßnahmen erforderlich sind, findet in der Problemdefinition statt. Zusätzlich werden bei der Implementierung des Anomaliedetektors die Maßnahme und damit auch die zugehörigen Schnittstellen für die Architektur definiert. Die Anforderung gilt somit als vollständig erfüllt.

Der Energieverbrauch von Systemen zur Erkennung energetischer Anomalien in Produktionsprozessen sollte minimiert werden, um den Effizienzgewinn nicht zu beeinträchtigen. Eine der wichtigen Faktoren ist die Datenmenge, die erfasst, gespeichert, verwaltet und analysiert werden muss. Die *Dateneffizienz* erfordert daher eine Minimierung der für die energetische Anomalieerkennung verwendeten Daten. Im in dieser Arbeit vorgestellten Ansatz werden zwei Wege für die Reduktion der Datenmenge vorgestellt:

Zum einen erfolgt die Reduktion durch die Gestaltung der Erkennung energetischer Anomalien nur mithilfe von Energiedaten. Somit werden keine Bild-, Video oder anderen Datenquellen genutzt. Zusätzlich wird mithilfe der Dateneffizienzanalyse der Einfluss der Abtastrate auf die zu erkennenden energetischen Anomalien untersucht. Falls eine hohe Abtastrate nicht notwendig ist, können Speicher- und Rechenkapazität eingespart werden. Für eine dennoch hinreichende Evaluationsmetrik können bei wenigen Beispielen eines Produktionsprozesses zusätzliche Augmentationsmechanismen genutzt werden, um für die Entwicklung des Modells und die Parameteroptimierung von FL genug Daten bereitzustellen. Die Anforderung gilt als vollständig erfüllt.

7.1.2 Anforderungen an das föderierte Lernen

Daten, die lokal von Anwendern von Maschinen und Anlagen erfasst werden, sollen von außen nicht zugänglich sein und nicht in der Cloud gespeichert werden. Dennoch sollen Verfahren zur Approximation von Vorhersagemodellen anhand der lokal gespeicherten Daten funktionieren. Durch die Sensibilität von Energiedaten auf Grundlage der Maschinen- und Betriebsdaten von Anwendern der Maschinen- und Anlagenbauer ist der *Schutz von sensiblen Maschinen- und Betriebsdaten* eine essenzielle Anforderung in der Industrie. Vermittels der dargelegten Methode und der Architektur kann ein FL-basiertes System eingeführt werden, um die Maschinen- und Betriebsdaten der Anwender zu schützen. Die Daten der Produktionsprozesse werden nicht an den Maschinen- und/oder Anlagenbauer weitergegeben, sondern bleiben innerhalb der Systemgrenzen des Anwenders. Somit ist die Anforderung vollständig erfüllt.

Das in dieser Arbeit entwickelte System soll die zuvor definierte Evaluationsmetrik im Vergleich zum rein lokalen Training der Daten verbessern. Diese Anforderung wurde mithilfe der in Abschnitt 5.3.1 vorgestellten Phase der Methode, sowie der Gestaltung der FLEA-Architektur dargestellt. Hierbei konnte für beide Anwendungsfälle gezeigt werden, dass die Verwendung von FL eine höhere Evaluationsmetrik in Relation zur durchschnittlichen lokalen Evaluationsmetrik aufweist. Die Anforderung der lokalen Verbesserung der Evaluationsmetrik ist somit vollständig erfüllt.

Angesichts der unterschiedlichen Prozess- und Betriebsparameter eines Produktionsprozesses – bspw. des Schweißprozesses – sind die erfassten Energiezeitreihen meist distinkt heterogen und stammen nicht aus einer einzigen Verteilung. Für das FL-basierte System zur energetischen Anomalieerkennung in Produktionsprozessen ist daher ein Umgang mit einer heterogenen Datenbasis unerlässlich, um die Datenanalyse und -nutzung in den Produktionsprozessen zu ermöglichen. Dies macht die *Berücksichtigung der individuellen Evaluationsmetriken* bei FL erforderlich. Mithilfe der Identifikation und Optimierung der Konfigurationsparameter von FL anhand einer kombinierten Metrik aus dem F1-Score und der Standardabweichung innerhalb der individuellen Evaluationsmetriken können Verfahren zur Aggregation der Modellparameter ausgewählt

werden, die die Heterogenität der Daten abbilden. Auch diese Anforderung ist damit vollständig erfüllt.

Für eine praxisnahe Anwendung der energetischen Anomalieerkennung in Produktionsprozessen ist es essenziell, zu bestimmen, welche *Konfigurationsparameter* von FL für den jeweiligen Anwendungsfall optimal sind. Daher wurde als Anforderung an die Methode eine Optimierung der Konfigurationsparameter festgelegt. Mithilfe der bayesschen Optimierung und einer umfassenden Analyse der Simulationsergebnisse konnten die Konfigurationsparameter identifiziert werden, die die kombinierte Metrik aus F1-Score und Fairness maximieren.

7.1.3 Anforderungen an die verteilte Systemarchitektur

Eine wichtige Anforderung für den entwickelten Ansatz ist die *Übertragbarkeit auf diverse Produktionsökosysteme*. Die Maschinen und Anlagen eines Herstellers können in verschiedenen Produktionsumgebungen eingesetzt werden. Die vorgestellte Architektur ist agnostisch vom Produktionsökosystem und benötigt mindestens eine Rechenkapazität zum Betreiben einer interpretierfähigen Programmiersprache wie Python. Durch die Erfassung von Daten aus unterschiedlichen Produktionsökosystemen für die Entwicklung des Modells sowie für die Parameteroptimierung von FL können die Komponenten der Architektur für unterschiedliche Produktionsökosysteme genutzt werden. Dennoch können Konnektivitätsprobleme in Umgebungen wie bspw. ländlichen Gebieten oder Baustellen auftreten, die geeignet sind, eine Übertragung der Modellparameter zu verhindern. Dieses Problem lässt sich durch Nachtrainieren und Zwischenspeichern der Daten lösen, was vorliegend jedoch nicht behandelt wurde. Daher ist diese Anforderung teilweise erfüllt.

Eine weitere Anforderung an die Architektur ist die Berücksichtigung einer *ganzheitlichen Datenverarbeitung*. Eine solche Architektur umfasst nicht nur die Entwicklung und Implementierung eines Modells, sondern auch die Entwicklung der Datenverarbeitungsschritte vom Sensor bis zur Gegenmaßnahme. Mithilfe des

Anomaliedetektors und der Erweiterung zur FLEA-Architektur erfolgt die Implementierung und Bereitstellung des Modells. In Phase I der Methode sollen dazu die ersten DVE definiert werden, die für die Implementierung des Anomaliedetektors in Phase II verwendet werden. Diese DVE sowie die Modelle und Gegenmaßnahmen können eingebettet in den Anomaliedetektor, die ganzheitliche Datenverarbeitung vom Sensor bis zur Gegenmaßnahme gewährleisten. Mithilfe der erarbeiteten Komponenten ist eine einfache Modellierung des Anomaliedetektors möglich. Die Anforderung ist vollständig erfüllt.

Die Herausforderung der Vielzahl an IT-Systemen in Produktionsumgebungen erfordert eine *Integration des Systems in bestehende Systeme*. Durch die standardisierte Kommunikation mithilfe der GET/FIT/PREDICT-Funktionalitäten wurde eine einfach nutzbare Schnittstelle entwickelt und somit die Interoperabilität mit anderen Systemen und Plattformen gewährleistet. Kommunikationsschnittstellen können unabhängig davon modular implementiert werden. Diese Anforderung ist vollständig erfüllt.

Die in Abschnitt 4.1 definierten Anforderungen wurden somit vollständig oder teilweise erfüllt.

7.2 Beantwortung der Forschungsfrage

Ziel dieser Arbeit ist die Entwicklung einer Methode zur Auslegung eines FL-basierten Systems zur Erkennung energetischer Anomalien in Produktionsprozessen. Das Ziel gilt als erreicht, wenn die zuvor definierten Forschungsfragen (siehe Abschnitt 1.3) umfassend beantwortet werden können. Die Beantwortung der zentralen Forschungsfrage schließt zudem den Forschungsprozess nach der Wissenschaftstheorie von Ulrich und Hill (1976) ab. Dabei wurde die zentrale Forschungsfrage (*Wie kann ein FL-basiertes System zur energetischen Anomalieerkennung in Produktionsprozessen ausgelegt werden?*) in Teilforschungsfragen untergliedert. Im Folgenden werden die Antworten auf diese Teilforschungsfragen gegeben und aggregiert.

1. *Was sind die relevanten Bestandteile eines auf dem föderierten Lernen basierten Systems zur energetischen Anomalieerkennung eines Produktionsprozesses?*

In Kapitel 2 wurden die relevanten Bestandteile eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen definiert und vorgestellt. Dabei wurden zunächst die Begriffe unter Zuhilfenahme der wissenschaftlichen Literatur definiert. Zusätzlich wurden die theoretischen Grundlagen von Produktion, Produktionssystemen und -prozessen, die energetische Anomalieerkennung sowie das FL vorgestellt und beschrieben.

2. *Welche Anforderungen für das föderierte Lernen zur energetischen Anomalieerkennung in Produktionsprozessen existieren?*

Anhand einer systematischen Literaturanalyse konnten in Kapitel 3 wissenschaftliche Arbeiten zum Stand der Technik identifiziert werden. Die Anforderungen wurden daraufhin in Kapitel 4 unterschiedliche Kategorien gruppiert – Energetische Anomalieerkennung, FL und verteilte Systemarchitektur. Die wissenschaftlichen Arbeiten aus Kapitel 3 wurden auf den Erfüllungsgrad der Anforderungen überprüft, um daraus den Handlungsbedarf abzuleiten.

3. *Wie kann das föderierte Lernen in die Auslegung eines Systems zur energetischen Anomalieerkennung in Produktionsprozessen integriert werden?*

In Kapitel 5 wurden zwei Artefakte zur Beantwortung der Forschungsfrage gestaltet: die Architektur des FL-basierten Systems zur energetischen Anomalieerkennung und die Methode zur Auslegung des Systems. Dabei erfolgt die Entwicklung der Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen von der Problemdefinition bis hin zur optimalen Konfiguration von FL. Dies ermöglicht eine ganzheitliche Auslegung des Systems.

4. *Wie kann die simulative Validierung des entwickelten Systems zur energetischen Anomalieerkennung in Produktionsprozessen erfolgen?*

Die simulative Validierung erfolgt durch Implementierung der Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung (siehe Kapitel 6).

Hierbei können die in den Anwendungsfällen erfassten Datensätze für die Simulation von FL genutzt werden, um somit die optimale Konfiguration der Parameter zu identifizieren. Erkenntnisse konnten basierend auf zwei Anwendungsfällen aggregiert und übergreifend formuliert werden.

Zusammenfassend konnten alle Teilforschungsfragen dieser Arbeit umfassend beantwortet werden. Der entwickelte Ansatz zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen adressiert somit auch die zentrale Forschungsfrage. Nach Ulrich und Hill (1976) ist der iterative Forschungsprozess damit abgeschlossen – die Artefakte sowie die Erkenntnisse können in die Wissensbasis und in die Praxis überführt werden.

8 Zusammenfassung und Ausblick

Angesichts der enormen Herausforderungen durch die Klimakrise und die Energiewende ist es von Bedeutung, dass produzierende Unternehmen ihr gesamtes Potenzial zur Steigerung der Energieeffizienz ausschöpfen. Durch den Einsatz von maschinellem Lernen und datengetriebenen Ansätzen wie der Erkennung energetischer Anomalien in Produktionsprozessen können sowohl der Energieverbrauch als auch die CO_2-Emissionen reduziert werden.

Die Reduktion des Energieverbrauchs durch energieeffizientere Produktionsprozesse, die den Ausschuss verringern und somit auch die Qualität des fertigen Produkts erhöhen, war die Motivation dieser Arbeit. Da die sensiblen Energiedaten der industriellen Maschinen und Anlagen in Produktionsprozessen, die für die Entwicklung eines Modells zur energetischen Anomalieerkennung genutzt werden, durch deren Anwender meist nicht zur Verfügung gestellt werden, ist hinsichtlich der energetischen Anomalieerkennung in Produktionsprozessen eine andere Herangehensweise erforderlich. Mit dem in dieser Arbeit vorgestellten Ansatz soll eine energetische Anomalieerkennung in Produktionsprozessen mithilfe von FL ermöglicht werden.

Daher war das Ziel dieser Arbeit, eine Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen zu gestalten. Hierfür wurde die FLEA-Architektur des Systems definiert, die mithilfe der Methode zur Auslegung eines FL-basierten Systems zur energetischen Anomalieerkennung in Produktionsprozessen implementiert werden kann.

Zunächst ist in einem ersten Schritt die Entwicklung eines Modells erforderlich. Die Analyse der Modellentwicklung zeigte, dass insbesondere bei kleinen Datensätzen die Anwendung von Augmentationsmechanismen für Zeitreihen die Evaluationsmetrik des Modells signifikant verbessert. Im zweiten Schritt erfolgt die Implementierung des in der FLEA-Architektur definierten Anomaliedetektors, der mithilfe der Bereitstellung des

Modells und der Implementierung von Agenten eine Reaktion auf eine erkannte energetische Anomalie ermöglicht. Als Drittes erfolgt die Parameteroptimierung von FL, indem Konfigurationsparameter definiert, simulationsbasierte Experimente durchgeführt und die optimierte Parameterkonfiguration ausgewählt wird. Es konnte gezeigt werden, dass der Einsatz von FL zu einer höheren Evaluationsmetrik führt als ein rein dezentraler Ansatz, bei dem die Daten des einzelnen Anwenders verwendet werden. Darüber hinaus nähern sich die Evaluationsmetriken beim Einsatz von FL dem klassischen Szenario an, in dem Maschinen- und Anlagenbauern alle Daten der Anwender zur Verfügung stehen. Mithilfe der entwickelten FLEA-Architektur können die Parameter aus der Methode in das FL-basierte System überführt werden.

Für künftige Forschungen ergeben sich zahlreiche Möglichkeiten. Eine Integration von unüberwachten und teilüberwachten Verfahren in die vorhandene Methode könnte die Abhängigkeit von annotierten Daten verringern und die Anwendbarkeit der Modelle in Situationen erweitern, in denen solche Daten schwer zugänglich sind. Die Erweiterung der Anwendungsbereiche, bspw. die Integration von FL in Lastprognosen, könnte wertvolle zusätzliche Erkenntnisse über die Energieverbrauchsmuster von Produktionsprozessen liefern.

Die Verwendung von Entscheidungsbäumen als Ergänzung zu neuronalen Netzen eröffnet eine vielversprechende Forschungsrichtung. Entscheidungsbäume können in bestimmten Szenarien effizienter sein und bieten häufig eine bessere Interpretierbarkeit, die für industrielle Anwendungen von großem Wert sein kann. Zudem gibt es im Bereich des föderierten Lernens neue Entwicklungen, z. B. das Clustern von Anwendungen mit ähnlichen Mustern. Diese haben das Potenzial, die Leistung der Modelle durch isoliertes Training von Modellen ähnlicher Anwender zu steigern.

Literaturverzeichnis

Aggarwal 2017
Aggarwal, Charu C., 2017. *Outlier Analysis.* 2nd ed. 2017. Cham, s.l.: Springer International Publishing
ISBN 9783319475783
DOI: 10.1007/978-3-319-47578-3

Ahn et al. 2023
Ahn, Jisu; Lee, Younjeong; Kim, Namji; Park, Chanho; Jeong, Jongpil, 2023. Federated Learning for Predictive Maintenance and Anomaly Detection Using Time Series Data Distribution Shifts in Manufacturing Processes
Sensors **23** (17), S. 7331
DOI: 10.3390/s23177331

Aledhari et al. 2020
Aledhari, Mohammed; Razzak, Rehma; Parizi, Reza M.; Saeed, Fahad, 2020. Federated learning: A survey on enabling technologies, protocols, and applications
IEEE Access **8**, S. 140699-140725

Bagheri et al. 2020
Bagheri, Behrad; Rezapoor, Maryam; Lee, Jay, 2020. A unified data security framework for federated prognostics and health management in smart manufacturing
Manufacturing Letters **24**, S. 136-139
DOI: 10.1016/j.mfglet.2020.04.011

Basora et al. 2019
Basora, Luis; Olive, Xavier; Dubot, Thomas, 2019. Recent Advances in Anomaly Detection Methods Applied to Aviation
Aerospace **6** (11), S. 117
DOI: 10.3390/aerospace6110117

Bauer et al. 2020
Bauer, Dennis; Hieronymus, Aljoscha; Kaymakci, Can; Köberlein, Jana; Schimmelpfennig, Jens; Wenninger, Simon; Zeiser, Reinhard, 2020. Wie IT die Energieflexibilitätsvermarktung von Industrieunternehmen ermöglicht und die Energiewende unterstützt
HMD Praxis der Wirtschaftsinformatik, S. 1-14
DOI: 10.1365/s40702-020-00679-8

Bauer et al. 2021
Bauer, Dennis; Kaymakci, Can; Bauernhansl, Thomas; Sauer, Alexander, 2021. Intelligent Energy Systems as Enabler for Increased Resilience of Manufacturing Systems
Procedia CIRP **104**, S. 217-222
DOI: 10.1016/j.procir.2021.11.037

BDEW 2023
BDEW, 2023. *Stromverbrauch in Deutschland: Letztverbrauch nach Verbrauchergruppen - Zehnjahresvergleich*
Verfügbar: https://www.bdew.de/service/daten-und-grafiken/stromverbrauch-deutschland/
Zugriff am: 29.08.2023

Bierbaum 2011 Bierbaum, Ulrich, 2011. *Druckluft-Kompendium.* 8. unveränd. Aufl. Darmstadt: Hoppenstedt Bonnier Zeitschriften
ISBN 3935772114

Bonawitz et al. 2019 Bonawitz, Keith; Eichner, Hubert; Grieskamp, Wolfgang; Huba, Dzmitry; Ingerman, Alex; Ivanov, Vladimir; Kiddon, Chloé; Konečný, Jakub; Mazzocchi, Stefano; McMahan, Brendan; van Overveldt, Timon; Petrou, David; Ramage, Daniel; Roselander, Jason, 2019. Towards Federated Learning at Scale: System Design
In: A. Talwalkar; V. Smith; M. Zaharia (Hrsg.)*: Proceedings of Machine Learning and Systems*, S. 374-388
Verfügbar: https://proceedings.mlsys.org/paper_files/paper/2019/file/7b770da633baf74895be22a8807f1a8f-Paper.pdf

Chandola et al. 2009 Chandola, Varun; Banerjee, Arindam; Kumar, Vipin, 2009. Anomaly detection
ACM Computing Surveys **41** (3), S. 1-58
DOI: 10.1145/1541880.1541882

Chen et al. 2014 Chen, Hui; Fei, Xiang; Wang, Sheng; Lu, Xin; Jin, Guoqin; Li, Weidong; Wu, Xuyang, 2014. Energy Consumption Data Based Machine Anomaly Detection
In: *2014 Second International Conference on Advanced Cloud and Big Data*, S. 136-142
DOI: 10.1109/CBD.2014.24

Chen et al. 2020 Chen, Yiqiang; Qin, Xin; Wang, Jindong; Yu, Chaohui; Gao, Wen, 2020. FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare
IEEE Intelligent Systems **35** (4), S. 83-93
DOI: 10.1109/MIS.2020.2988604

Cook et al. 2020 Cook, Andrew A.; Misirli, Goksel; Fan, Zhong, 2020. Anomaly Detection for IoT Time-Series Data: A Survey
IEEE Internet of Things Journal **7** (7), S. 6481-6494
DOI: 10.1109/JIOT.2019.2958185

da Silveira Dib et al. 2021 da Silveira Dib, Mario Alberto; Ribeiro, Bernardete; Prates, Pedro, 2021. Federated Learning as a Privacy-Providing Machine Learning for Defect Predictions in Smart Manufacturing
2520-6478 **5** (1), S. 20200029
DOI: 10.1520/SSMS20200029

DIN EN 62264-1 DIN EN 62264-1:2014-07. *Integration von Unternehmensführungs- und Leitsystemen - Teil 1: Modelle und Terminologie*

DIN EN ISO 50001 DIN EN ISO 50001:2018-12. *Energiemanagementsysteme: Anforderungen mit Anleitung zur Anwendung*
DOI: 10.31030/2852891

DIN-Norm 8580 DIN-Norm 8580:2022-12. *Fertigungsverfahren: Begriffe, Einteilung*
DOI: 10.31030/3217872

Dierolf 2022
Dierolf, Christian, 2022. *Identifikation und Klassifikation von Druckluft-Leckagen mit Methoden des maschinellen Lernens in Maschinen*, Stuttgart, Univ., Diss., 2022
DOI: 10.18419/opus-12124

Dierolf & Sauer 2021
Dierolf, Christian; Sauer, Alexander, 2021. Methoden zur Erkennung von Druckluftleckagen
wt Werkstattstechnik online **111** (01-02), S. 37-43
DOI: 10.37544/1436-4980-2021-01-02-41

Dyckhoff & Spengler 2010
Dyckhoff, Harald; Spengler, Thomas Stefan, 2010. *Produktionswirtschaft: Eine Einführung.* 3., überarb. und erw. Aufl. Berlin: Springer. Springer-Lehrbuch
ISBN 978-3-642-13684-9

Europäisches Parlament 2006
Europäisches Parlament (Hrsg.), 2006. *Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on machinery, and amending Directive 95/16/EC (recast) (Text with EEA relevance)* L 157
Verfügbar: http://data.europa.eu/eli/dir/2006/42/oj
Zugriff am: 29.10.2024

Faltinski et al. 2012
Faltinski, Sebastian; Flatt, Holger; Pethig, Florian; Kroll, Bjorn; Vodencarevic, Asmir; Maier, Alexander; Niggemann, Oliver, 2012. Detecting anomalous energy consumptions in distributed manufacturing systems
In: *IEEE 10th International Conference on Industrial Informatics*
DOI: 10.1109/indin.2012.6301142

Faubel et al. 2023
Faubel, Leonhard; Schmid, Klaus; Eichelberger, Holger, 2023. MLOps Challenges in Industry 4.0
SN Computer Science **4** (6)
DOI: 10.1007/s42979-023-02282-2

Flick et al. 2020
Flick, Dominik; Keck, Claudio; Herrmann, Christoph; Thiede, Sebastian, 2020. Machine learning based analysis of factory energy load curves with focus on transition times for anomaly detection
Procedia CIRP **93**, S. 461-466
DOI: 10.1016/j.procir.2020.04.073

Friedrich et al. 2021
Friedrich, Robert; Ploner, Florian; Schäfer, Christian Thomas; Disselhoff, Tim; Petkau, Andreas; Hennemann, Constantin; Moecke, Jakob; Wätzig, Tobias; Zimmert, Oliver; Waltersmann, Lara; Kiemel, Steffen; Miehe, Robert; Sauer, Alexander; Menn, Jan Philipp; Oberender, Christof; Vogt, Martin, 2021. *Potenziale der schwachen künstlichen Intelligenz für die betriebliche Ressourceneffizienz*. Berlin
Verfügbar: https://www.ressource-deutschland.de/fileadmin/user_upload/1_Themen/h_Publikationen/Studien/VDI-ZRE_Studie_KI-betriebliche-Ressourceneffizienz_Web_bf.pdf
Zugriff am: 29.10.2024

Ge et al. 2021
Ge, Ning; Li, Guanghao; Zhang, Li; Liu, Yi Liu Yi, 2021. *Failure Prediction in Production Line Based on Federated Learning: An Empirical Study*
DOI: 10.48550/arXiv.2101.11715

Geckler et al. 2020
Geckler, Dieter; Fliege, Holger; Nagel, Joachim; Bracht, Uwe; Seewaldt, Marco; Wolff, Daniel, 2020. Der Energiebedarf der Automobilproduktion im Fokus der Simulation
In: Mayer, Gottfried; Pöge, Carsten; Spieckermann, Sven; Wenzel, Sigrid (Hrsg.)*: Ablaufsimulation in der Automobilindustrie*. Berlin: Springer, S. 303-317
ISBN 978-3-662-59387-5
DOI: 10.1007/978-3-662-59388-2_21

Geilhausen et al. 2015
Geilhausen, Marko; Bränzel, Juliane; Engelmann, Dirk; Schulze, Olaf, 2015. *Energiemanagement*. Wiesbaden: Springer
ISBN 978-3-658-02833-6
DOI: 10.1007/978-3-658-02834-3

Geng et al. 2022
Geng, DaoQu; He, HanWen; Lan, XingChuan; Liu, Chang, 2022. Bearing fault diagnosis based on improved federated learning algorithm
Computing **104** (1), S. 1-19
DOI: 10.1007/s00607-021-01019-4

Grimme & Bossek 2018
Grimme, Christian; Bossek, Jakob, 2018. Nichtlineare Optimierung
In: Grimme, Christian; Bossek, Jakob (Hrsg.)*: Einführung in die Optimierung*. Wiesbaden: Springer, S. 149-189
ISBN 978-3-658-21150-9
DOI: 10.1007/978-3-658-21151-6_4

Groover 2020
Groover, Mikell P., 2020. *Fundamentals of modern manufacturing: Materials, processes, and systems.* Seventh edition. Hoboken, NJ: John Wiley & Sons, Inc
ISBN 978-1-119-72201-4

Gupta et al. 2014
Gupta, Manish; Gao, Jing; Aggarwal, Charu C.; Han, Jiawei, 2014. Outlier Detection for Temporal Data: A Survey
IEEE Transactions on Knowledge and Data Engineering **26** (9), S. 2250-2267
DOI: 10.1109/TKDE.2013.184

Hartigan & Wong 1979
Hartigan, J. A.; Wong, M. A., 1979. Algorithm AS 136: A K-Means Clustering Algorithm
Applied Statistics **28** (1), S. 100
DOI: 10.2307/2346830

Hevner et al. 2004
Hevner, Alan; March, Salvatore; Park, Jinsoo; Ram, Sudha, 2004. Design Science in Information Systems Research
MIS Quarterly **28** (1), S. 75
DOI: 10.2307/25148625

Himeur et al. 2021 Himeur, Yassine; Ghanem, Khalida; Alsalemi, Abdullah; Bensaali, Faycal; Amira, Abbes, 2021. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives
Applied Energy **287**, S. 116601
DOI: 10.1016/j.apenergy.2021.116601

Hofmann et al. 2020 Hofmann, Peter; Jöhnk, Jan; Protschky, Dominik; Urbach, Nils, 2020. Developing Purposeful AI Use Cases – A Structured Method and Its Application in Project Management
In: Gronau, Norbert; Heine, Moreen; Poustcchi, K.; Krasnova, H. (Hrsg.)*: 15. Internationalen Tagung Wirtschaftsinformatik*: GITO Verlag 1, S. 33-49
ISBN 9783955453350
DOI: 10.30844/wi_2020_a3-hofmann

Hutter et al. 2019 Hutter, Frank; Kotthoff, Lars; Vanschoren, Joaquin, 2019. *Automated Machine Learning*. Cham: Springer International Publishing
ISBN 978-3-030-05317-8
DOI: 10.1007/978-3-030-05318-5

IEEE 3652 IEEE 3652. *IEEE Guide for Architectural Framework and Application of Federated Machine Learning*
DOI: 10.1109/IEEESTD.2021.9382202

IHK Nürnberg 2012 IHK Nürnberg, 2012. *Druckluft effizient nutzen: Praxisleitfaden für Energieeffizienz und Kosteneinsparung*. Nürnberg

International Energy Agency 2023 International Energy Agency, 2023. *Energy Efficiency - The Decade for Action*, Ministerial Briefing. Versailles
Verfügbar: https://iea.blob.core.windows.net/assets/f6df3a56-2257-4f47-a130-bf0862c31065/EnergyEfficiency-TheDecadeforAction.pdf
Zugriff am: 11.12.2023

Ju et al. 2020 Ju, Ce; Gao, Dashan; Mane, Ravikiran; Tan, Ben; Liu, Yang; Guan, Cuntai, 2020. Federated Transfer Learning for EEG Signal Classification
In: *2020 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Montreal, QC, Canada, 20-24 July 2020* S. 3040-3045
DOI: 10.1109/EMBC44109.2020.9175344

Kanagavelu et al. 2021 Kanagavelu, Renuga; Li, Zengxiang; Samsudin, Juniarto; Hussain, Shaista; Yang, Feng; Yang, Yechao; Goh, Rick Siow Mong; Cheah, Mervyn, 2021. Federated Learning for Advanced Manufacturing Based on Industrial IoT Data Analytics
In: Toro, Carlos; Wang, Wei; Akhtar, Humza (Hrsg.)*: Implementing Industry 4.0*: Springer Nature. Intelligent Systems Reference Library, S. 143-176
ISBN 978-3-030-67269-0
DOI: 10.1007/978-3-030-67270-6_6

Karim et al. 2019
Karim, Fazle; Majumdar, Somshubra; Darabi, Houshang; Harford, Samuel, 2019. Multivariate LSTM-FCNs for Time Series Classification
Neural Networks **116** (2), S. 237-245
DOI: 10.1016/j.neunet.2019.04.014
ArXiv-ID: 1801.04503v2

Karnebogen et al. 2023
Karnebogen, Philip; Kaymakci, Can; Willburger, Lukas; Häckel, Björn; Sauer, Alexander, 2023. Structuring Federated Learning Applications : A Literature Analysis and Taxonomy
In: *Proceedings of the 31th European Conference on Information Systems (ECIS).* European Conference on Information Systems (ECIS). Kristiansand, 11.-16.06.2023
Verfügbar: https://eref.uni-bayreuth.de/id/eprint/76251/

Kaymakci et al. 2021a
Kaymakci, Can; Wenninger, Simon; Sauer, Alexander, 2021. A Holistic Framework for AI Systems in Industrial Applications
In: Ahlemann, Frederik; Schütte, Reinhard; Stieglitz, Stefan (Hrsg.)*: Innovation Through Information Systems*. Cham: Springer International Publishing. Lecture Notes in Information Systems and Organisation, S. 78-93
ISBN 978-3-030-86796-6
DOI: 10.1007/978-3-030-86797-3_6

Kaymakci et al. 2021b
Kaymakci, Can; Wenninger, Simon; Sauer, Alexander, 2021. Energy Anomaly Detection in Industrial Applications with Long Short-term Memory-based Autoencoders
Procedia CIRP **104**, S. 182-187
DOI: 10.1016/j.procir.2021.11.031

Kaymakci et al. 2021c
Kaymakci, Can; Baur, Lukas; Sauer, Alexander, 2021. Federated Machine Learning Architecture for Energy-Efficient Industrial Applications
In: Herberger, David; Hübner Marco (Hrsg.)*: Proceedings of the 2nd Conference on Production Systems and Logistics.* CPSL, 10.-11.08.2021
DOI: 10.15488/11237

Kingma & Ba 2014
Kingma, Diederik P.; Ba, Jimmy, 2014. *Adam: A Method for Stochastic Optimization*
DOI: 10.48550/arXiv.1412.6980

Kiran et al. 2018
Kiran, B.; Thomas, Dilip; Parakkal, Ranjith, 2018. An Overview of Deep Learning Based Methods for Unsupervised and Semi-Supervised Anomaly Detection in Videos
Journal of Imaging **4** (2), S. 36
DOI: 10.3390/jimaging4020036

Kletti 2015
Kletti, Jürgen, 2015. *MES - Manufacturing Execution System*. Berlin: Springer
ISBN 978-3-662-46901-9
DOI: 10.1007/978-3-662-46902-6

Kroll et al. 2014
Kroll, Bjorn; Schaffranek, David; Schriegel, Sebastian; Niggemann, Oliver, 2014. System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants
In: *2014 IEEE [International Conference on] Emerging Technologies and Factory Automation (ETFA 2014): Barcelona, Spain, 16 - 19 September 2014.* 2014 IEEE Emerging Technology and Factory Automation (ETFA). Barcelona, Spain, 9/16/2014 - 9/19/2014, S. 1-7
DOI: 10.1109/ETFA.2014.7005202

Li et al. 2019
Li, Tian; Sanjabi, Maziar; Beirami, Ahmad; Smith, Virginia, 2019. *Fair Resource Allocation in Federated Learning*
DOI: 10.48550/arXiv.1905.10497

Liang et al. 2018
Liang, Peng; Yang, Hai-Dong; Chen, Wen-Si; Xiao, Si-Yuan; Lan, Zhao-Ze, 2018. Transfer learning for aluminium extrusion electricity consumption anomaly detection via deep neural networks
International Journal of Computer Integrated Manufacturing **31** (4-5), S. 396-405
DOI: 10.1080/0951192X.2017.1363410

Lo et al. 2022
Lo, Sin Kit; Lu, Qinghua; Zhu, Liming; Paik, Hye-Young; Xu, Xiwei; Wang, Chen, 2022. Architectural patterns for the design of federated learning systems
Journal of Systems and Software **191**, S. 111357
DOI: 10.1016/j.jss.2022.111357

Lorenti et al. 2022
Lorenti, Luciano; Rossi, Gaia de; Annoni, Alberto; Rigutto, Silvano; Susto, Gian Antonio, 2022. CUAD-Mo: Continuos Unsupervised Anomaly Detection on Machining Operations
In: *2022 IEEE Conference on Control Technology and Applications (CCTA).* 2022 IEEE Conference on Control Technology and Applications (CCTA). Trieste, Italy, 8/23/2022 - 8/25/2022, S. 881-886
DOI: 10.1109/CCTA49430.2022.9966138

Maesschalck et al. 2000
Maesschalck, R. de; Jouan-Rimbaud, D.; Massart, D. L., 2000. The Mahalanobis distance
Chemometrics and Intelligent Laboratory Systems **50** (1), S. 1-18
DOI: 10.1016/S0169-7439(99)00047-7

Mallia et al. 2023
Mallia, Jasmine; Francalanza, Emmanuel; Xuereb, Peter; Refalo, Paul, 2023. Intelligent Approaches for Anomaly Detection in Compressed Air Systems: A Systematic Review
Machines **11** (7), S. 750
DOI: 10.3390/machines11070750

Maschler et al. 2019
Maschler, Benjamin; Jazdi, Nasser; Weyrich, Michael, 2019. Maschinelles Lernen für intelligente Automatisierungssysteme mit dezentraler Datenhaltung am Anwendungsfall Predictive Maintenance
VDI-Berichte (2351), S. 739-751
DOI: 10.18419/opus-10503

Matthes & Schneider 2016 Matthes, Klaus-Jürgen; Schneider, Werner (Hrsg.), 2016. *Schweißtechnik: Schweißen von metallischen Konstruktionswerkstoffen.* 6., aktualisierte Auflage. München: Fachbuchverlag Leipzig im Carl Hanser Verlag ISBN 978-3-446-42832-4

McMahan et al. 2016 McMahan, H. Brendan; Konečný, Jakub; Yu, Felix X.; Richtárik, Peter; Suresh, Ananda Theertha; Bacon, Dave, 2016. *Federated Learning: Strategies for Improving Communication Efficiency* DOI: 10.48550/arXiv.1610.05492

Mohammed & Mohamed 2019 Mohammed M. Alani; Mohamed Alloghani, 2019. Security Challenges in the Industry 4.0 Era In: *Industry 4.0 and Engineering for a Sustainable Future*. Cham: Springer, S. 117-136 ISBN 978-3-030-12955-2 DOI: 10.1007/978-3-030-12953-8_8

Nayak et al. 2022 Nayak, Sabuzima; Patgiri, Ripon; Waikhom, Lilapati; Ahmed, Arif, 2022. A review on edge analytics: Issues, challenges, opportunities, promises, future directions, and applications *Digital Communications and Networks* DOI: 10.1016/j.dcan.2022.10.016

Offermann et al. 2010 Offermann, Philipp; Blom, Sören; Schönherr, Marten; Bub, Udo, 2010. Artifact Types in Information Systems Design Science – A Literature Review In: Hutchison, David; Kanade, Takeo; Kittler, Josef; Kleinberg, Jon M.; Mattern, Friedemann; Mitchell, John C.; Naor, Moni; Nierstrasz, Oscar; Pandu Rangan, C.; Steffen, Bernhard; Sudan, Madhu; Terzopoulos, Demetri; Tygar, Doug; Vardi, Moshe Y.; Weikum, Gerhard; Winter, Robert; Zhao, J. L.; Aier, Stephan (Hrsg.)*: Global Perspectives on Design Science Research*. Berlin: Springer. Lecture Notes in Computer Science, S. 77-92 ISBN 978-3-642-13334-3 DOI: 10.1007/978-3-642-13335-0_6

Ozaki et al. 2022 Ozaki, Yoshihiko; Tanigaki, Yuki; Watanabe, Shuhei; Nomura, Masahiro; Onishi, Masaki, 2022. Multiobjective Tree-Structured Parzen Estimator *Journal of Artificial Intelligence Research* **73**, S. 1209-1250 DOI: 10.1613/jair.1.13188

Pang et al. 2021 Pang, Guansong; Shen, Chunhua; Cao, Longbing; van den Hengel, Anton, 2021. Deep Learning for Anomaly Detection: A Review *ACM Computing Surveys* **54** (2), S. 1-38 DOI: 10.1145/3439950

Parto et al. 2020
Parto, Mahmoud; Saldana, Christopher; Kurfess, Thomas, 2020. A Novel Three-Layer IoT Architecture for Shared, Private, Scalable, and Real-time Machine Learning from Ubiquitous Cyber-Physical Systems
Procedia Manufacturing **48**, S. 959-967
DOI: 10.1016/j.promfg.2020.05.135

Parvathy et al. 2020
Parvathy, S.; Safni Usman, T.; Harikrishnan, K., 2020. Energy Management via Anomaly Detection for Manufacturing Enterprises
In: *2020 International Conference for Emerging Technology (INCET).* Belgaum, India. 05.06.2020 - 07.06.2020
DOI: 10.1109/incet49848.2020.9153989

Peffers et al. 2007
Peffers, Ken; Tuunanen, Tuure; Rothenberger, Marcus A.; Chatterjee, Samir, 2007. A Design Science Research Methodology for Information Systems Research
Journal of Management Information Systems **24** (3), S. 45-77
DOI: 10.2753/MIS0742-1222240302

Petladwala et al. 2019
Petladwala, Murtuza; Ishii, Yuko; Sendoda, Mitsuru; Kondo, Reishi, 2019. Canonical Correlation Based Feature Extraction with Application to Anomaly Detection in Electric Appliances
In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*
DOI: 10.1109/icassp.2019.8683671

Radermacher et al. 2021
Radermacher, Tobias; Merx, Marcel; Sitte, Andre; Boyko, Vladimir; Unger, Manuel, 2021. *Potenzialstudie Energie-/Kosteneinsparung in der Fluidtechnik*. Dessau-Roßlau

Reddi et al. 2020
Reddi, Sashank; Charles, Zachary; Zaheer, Manzil; Garrett, Zachary; Rush, Keith; Konečný, Jakub; Kumar, Sanjiv; McMahan, H. Brendan, 2020. *Adaptive Federated Optimization*
DOI: 10.48550/arXiv.2003.00295

Ruff et al. 2021
Ruff, Lukas; Kauffmann, Jacob R.; Vandermeulen, Robert A.; Montavon, Grégoire; Samek, Wojciech; Kloft, Marius; Dietterich, Thomas G.; Müller, Klaus-Robert, 2021. A Unifying Review of Deep and Shallow Anomaly Detection
Proceedings of the IEEE **109** (5), S. 756-795
DOI: 10.1109/JPROC.2021.3052449

Runkler 2015
Runkler, Thomas A., 2015. *Data Mining*. Wiesbaden: Springer
ISBN 978-3-8348-1694-8
DOI: 10.1007/978-3-8348-2171-3

Russell & Norvig 2021
Russell, Stuart J.; Norvig, Peter, 2021. *Artificial intelligence: A modern approach.* Fourth Edition. Hoboken, NJ: Pearson. Pearson Series in Artificial Intelligence
ISBN 0134610997

Sauer et al. 2022 Sauer, Alexander; Buhl, Hans Ulrich; Mitsos, Alexander; Weigold, Matthias (Hrsg.), 2022. *Markt- und Stromsystem, Managementsysteme und Technologien energieflexibler Fabriken: Kopernikus Projekte SynErgie - Die Zukunft unserer Energie*. Stuttgart: Fraunhofer Verlag. Energieflexibilität in der deutschen Industrie / Alexander Sauer, Eberhard Abele, Hans Ulrich Buhl Band 2
ISBN 978-3-8396-1778-6

Siepmann & Graef 2016 Siepmann, David; Graef, Norbert, 2016. Industrie 4.0 – Grundlagen und Gesamtzusammenhang
In: Roth, Armin (Hrsg.)*: Einführung und Umsetzung von Industrie 4.0*. Berlin: Springer, S. 17-82
ISBN 978-3-662-48504-0
DOI: 10.1007/978-3-662-48505-7_2

Soni et al. 2016 Soni, Nimish; Gopalan, Venkoparao Vijendran; Varadharajan, R., 2016. Electrical and operational anomaly detection in energy intensive manufacturing industries
In: *2016 IEEE Annual India Conference (INDICON): 16-18 Dec. 2016.* 2016 IEEE Annual India Conference (INDICON). Bangalore, India, 12/16/2016 - 12/18/2016, S. 1-6
DOI: 10.1109/INDICON.2016.7839161

Tharwat 2021 Tharwat, Alaa, 2021. Classification assessment methods
Applied Computing and Informatics **17** (1), S. 168-192
DOI: 10.1016/j.aci.2018.08.003

Theumer et al. 2021 Theumer, Philipp; Zeiser, Reinhard; Trauner, Ludwig; Reinhart, Gunther, 2021. Anomaly detection on industrial time series for retaining energy efficiency
Procedia CIRP **99**, S. 33-38
DOI: 10.1016/j.procir.2021.03.006

Tomczak 1992 Tomczak, Torsten, 1992. Forschungsmethoden in der Marketingwissenschaft: Ein Plädoyer für den qualitativen Forschungsansatz
Marketing ZFP **14** (2)
DOI: 10.15358/0344-1369-1992-2

Tran et al. 2022 Tran, Duc Hoang; Nguyen, Van Linh; Utama, Ida Bagus Krishna Yoga; Jang, Yeong Min, 2022. An Improved Sensor Anomaly Detection Method in IoT System using Federated Learning
In: *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN).* 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN). Barcelona, Spain, 7/5/2022 - 7/8/2022, S. 466-469
DOI: 10.1109/ICUFN55119.2022.9829561

Ulrich & Hill 1976 Ulrich, Peter; Hill, Wilhelm, 1976. Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre
In: *Wirtschaftswissenschaftliches Studium: WiSt; Zeitschrift für Studium und Forschung*. München, S. 304-309

VDI-Richtlinie 5600 Blatt 1
VDI-Richtlinie 5600 Blatt 1:2016-10.
Fertigungsmanagementsysteme (Manufacturing Execution Systems - MES)

VDI-Richtlinie 5600 Blatt 6
VDI-Richtlinie 5600 Blatt 6:2017-10. *Energiemanagement mit MES (Manufacturing Execution Systems - MES)*

vom Brocke et al. 2009
vom Brocke, Jan; Simons Alexander; Niehaves Björn; Riemer, Kai; Plattfaut, Ralf; Cleven, Anne, 2009. Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process
Proceedings of the 17th European Conference on Information Systems (ECIS).
Verfügbar unter: https://aisel.aisnet.org/ecis2009/161
Zugriff am: 29.10.2024

Waltersmann et al. 2021
Waltersmann, Lara; Kiemel, Steffen; Stuhlsatz, Julian; Sauer, Alexander; Miehe, Robert, 2021. Artificial Intelligence Applications for Increasing Resource Efficiency in Manufacturing Companies—A Comprehensive Review
Sustainability **13** (12), S. 6689
DOI: 10.3390/su13126689

Wang et al. 2017
Wang, Zhiguang; Yan, Weizhong; Oates, Tim, 2017. Time series classification from scratch with deep neural networks: A strong baseline
In: *2017 International Joint Conference on Neural Networks (IJCNN).* 2017 International Joint Conference on Neural Networks (IJCNN). Anchorage, AK, USA, 14.05.2017 - 19.05.2017, S. 1578-1585
DOI: 10.1109/IJCNN.2017.7966039

Wang, Haoxiang **et al. 2021**
Wang, Haoxiang; Liu, Chao; Jiang, Dongxiang; Jiang, Zhanhong, 2021. Collaborative deep learning framework for fault diagnosis in distributed complex systems
Mechanical Systems and Signal Processing **156**, S. 107650
DOI: 10.1016/j.ymssp.2021.107650

Wang, Qizhao **et al. 2021b**
Wang, Qizhao; Li, Qing; Wang, Kai; Wang, Hong; Zeng, Peng, 2021. Efficient federated learning for fault diagnosis in industrial cloud-edge computing
Computing **103** (10), S. 2319-2337
DOI: 10.1007/s00607-021-00970-6

Wen & Chen 2020
Wen, Xianhe; Chen, Heping, 2020. Heterogeneous Connection and Process Anomaly Detection of Industrial Robot in Intelligent Factory
International Journal of Pattern Recognition and Artificial Intelligence **34** (12), S. 2059041
DOI: 10.1142/S0218001420590417

Wenninger et al. 2022
Wenninger, Simon; Kaymakci, Can; Wiethe, Christian; Römmelt, Jörg; Baur, Lukas; Häckel, Björn; Sauer, Alexander, 2022. How Sustainable is Machine Learning in Energy Applications? – The Sustainable Machine Learning Balance Sheet
In: *Wirtschaftsinformatik 2022 Proceedings*
Verfügbar: https://aisel.aisnet.org/wi2022/sustainable_it/sustainable_it/1
Zugriff am: 29.10.2024

Westkämper 2006
Westkämper, Engelbert, 2006. *Einführung in die Organisation der Produktion*. Berlin: Springer. Springer-Lehrbuch
ISBN 978-3-540-26039-4

Westkämper 2013
Westkämper, Engelbert (Hrsg.), 2013. *Digitale Produktion*. Berlin: Springer
ISBN 978-3-642-20258-2
DOI: 10.1007/978-3-642-20259-9

Westkämper & Hummel 2009
Westkämper, Engelbert; Hummel, Vera, 2009. Grundlagen des Stuttgarter Unternehmensmodells
In: Westkämper, Engelbert; Zahn, Erich (Hrsg.)*: Wandlungsfähige Produktionsunternehmen*. Berlin: Springer, S. 47-66
ISBN 978-3-540-21889-0
DOI: 10.1007/978-3-540-68890-7_4

Windmann et al. 2013
Windmann, Stefan; Jiao, Shuo; Niggemann, Oliver; Borcherding, Holger, 2013. A stochastic method for the detection of anomalous energy consumption in hybrid industrial systems
In: Institute of Electrical and Electronics Engineers (Hrsg.)*: 2013 IEEE 11th International Conference on Industrial Informatics (INDIN).* Bochum, Germany, S. 194-199
DOI: 10.1109/INDIN.2013.6622881

Wirth & Hipp 2000
Wirth, Rüdiger; Hipp, Joachim, 2000. CRISP-DM: Towards a Standard Process Model for Data Mining
In: Mackin, Neil (Hrsg.)*: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining.* 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining. Manchester, S. 29-39

Xu et al. 2021
Xu, Chuqiao; Wang, Junliang; Zhang, Jie; Li, Xiaoou, 2021. Anomaly detection of power consumption in yarn spinning using transfer learning
Computers & Industrial Engineering **152**, S. 107015
DOI: 10.1016/j.cie.2020.107015

Yang et al. 2012
Yang, Hai-Dong; Guo, Jian-hua; Liu, Guo-sheng, 2012. Energy anomaly detection in tire curing by using data integration and forecasting techniques
Journal of Scientific and Industrial Research (JSIR) (71), S. 385-391.
Verfügbar unter:
http://nopr.niscair.res.in/handle/123456789/14155

Yang et al. 2019
Yang, Qiang; Liu, Yang; Chen, Tianjian; Tong, Yongxin, 2019. Federated Machine Learning
ACM Transactions on Intelligent Systems and Technology **10** (2), S. 1-19
DOI: 10.1145/3298981

Yang & Shami 2020
Yang, Li; Shami, Abdallah, 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice
Neurocomputing **415**, S. 295-316
DOI: 10.1016/j.neucom.2020.07.061

Zellinger et al. 2021
Zellinger, Werner; Wieser, Volkmar; Kumar, Mohit; Brunner, David; Shepeleva, Natalia; Gálvez, Rafa; Langer, Josef; Fischer, Lukas; Moser, Bernhard, 2021. Beyond federated learning: On confidentiality-critical machine learning applications in industry
Procedia Computer Science **180**, S. 734-743
DOI: 10.1016/j.procs.2021.01.296

Zhang et al. 2020
Zhang, Chaoyang; Wang, Zhengxu; Ding, Kai; Chan, Felix T.S.; Ji, Weixi, 2020. An energy-aware cyber physical system for energy Big data analysis and recessive production anomalies detection in discrete manufacturing workshops
International Journal of Production Research **58** (23), S. 7059-7077
DOI: 10.1080/00207543.2020.1748904

Zhang & Ji 2020
Zhang, Chaoyang; Ji, Weixi, 2020. Edge Computing Enabled Production Anomalies Detection and Energy-Efficient Production Decision Approach for Discrete Manufacturing Workshops
IEEE Access **8**, S. 158197-158207
DOI: 10.1109/access.2020.3020136

Zhang, Kai **et al. 2021**
Zhang, Kai; Jiang, Yushan; Seversky, Lee; Xu, Chengtao; Liu, Dahai; Song, Houbing, 2021. Federated Variational Learning for Anomaly Detection in Multivariate Time Series
In: *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC).* 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC). Austin, TX, USA, 29.10.2021 - 31.10.2021, S. 1-9
DOI: 10.1109/IPCCC51483.2021.9679367

Zhang, Jinglin **et al. 2021**
Zhang, Jinglin; Wang, Yanbo; Zhu, Kai; Zhang, Yi; Li, Yuanjiang, 2021. Diagnosis of Interturn Short-Circuit Faults in Permanent Magnet Synchronous Motors Based on Few-Shot Learning Under a Federated Learning Framework
IEEE Transactions on Industrial Informatics **17** (12), S. 8495-8504
DOI: 10.1109/TII.2021.3067915

Zhang, Weishan **et al. 2021**
Zhang, Weishan; Lu, Qinghua; Yu, Qiuyu; Li, Zhaotong; Liu, Yue; Lo, Sin Kit; Chen, Shiping; Xu, Xiwei; Zhu, Liming, 2021. Blockchain-Based Federated Learning for Device Failure Detection in Industrial IoT
IEEE Internet of Things Journal **8** (7), S. 5926-5937
DOI: 10.1109/JIOT.2020.3032544

Zhang, Wei **et al. 2021** Zhang, Wei; Li, Xiang; Ma, Hui; Luo, Zhong; Li, Xu, 2021. Federated learning for machinery fault diagnosis with dynamic validation and self-supervision
Knowledge-Based Systems **213**, S. 106679
DOI: 10.1016/j.knosys.2020.106679

Zhang, Zehui **et al. 2021** Zhang, Zehui; Xu, Xiaobin; Gong, Wenfeng; Chen, Yuwang; Gao, Haibo, 2021. Efficient federated convolutional neural network with information fusion for rolling bearing fault diagnosis
Control Engineering Practice **116**, S. 104913
DOI: 10.1016/j.conengprac.2021.104913

Zhang & Li 2022 Zhang, Wei; Li, Xiang, 2022. Data privacy preserving federated transfer learning in machinery fault diagnostics using prior distributions
Structural Health Monitoring **21** (4), S. 1329-1344
DOI: 10.1177/14759217211029201

Zhao et al. 2017 Zhao, Bendong; Lu, Huanzhang; Chen, Shangfeng; Liu, Junliang; Wu, Dongya, 2017. Convolutional neural networks for time series classification
Journal of Systems Engineering and Electronics **28** (1), S. 162-169
DOI: 10.21629/JSEE.2017.01.18

Anhang

A.1 Detaillierte Beschreibung der Modelle für das Modelltraining

Tabelle 9 Ausführliche Darstellung der genutzten ML-Modelle für das Modelltraining

Modellbezeichnung	Architektur	Datensicht
ResNet	4×Conv(64), 4×Conv(128), AvgPool()	Kontextbasierte Energiezeitreihengruppen
CNN	Conv(6), AvgPool(3), Conv(12), AvgPool()	Kontextbasierte Energiezeitreihengruppen
FCN	Conv(128), Conv(256), Conv(128), AvgPool()	Kontextbasierte Energiezeitreihengruppen
LSTM-FCN	LSTM(8), Conv(128), Conv(256), Conv(128), AvgPool()	Kontextbasierte Energiezeitreihengruppen
MLP	Dense(500, 500, 500)	Kontextbasierte Energiezeitreihengruppen
$MLPKontext_1$	Dense(100)	Kontextbasierte Merkmale
$MLPKontext_2$	Dense(256, 128, 64, 32, 16)	Kontextbasierte Merkmale
$MLPKontext_3$	Dense(128, 64, 32, 16)	Kontextbasierte Merkmale
$MLPKontext_4$	Dense(512, 256, 128, 64, 32)	Kontextbasierte Merkmale
$MLPKontext_5$	Dense(128, 64, 32, 16, 16, 32, 64, 128)	Kontextbasierte Merkmale

A.2 Detaillierte Beschreibung der Augmentationsmechanismen

Tabelle 10 Auflistung der genutzten Augmentationsmechanismen zur Generierung von synthetischen Daten

Name der Methode	**Beschreibung**	**Label**
Jitter	Zufälliges Rauschen hinzufügen	Nicht notwendig
Scaling	Zufällige Skalierung	Nicht notwendig
Rotation	Zufällige Drehung und Spiegelung	Nicht notwendig
Permutation	Zufällige Segmentpermutation	Nicht notwendig
Magnitude Warp	Zufällige Verzerrung der Magnitude	Nicht notwendig
Time Warp	Zeitliche Verzerrung der Daten	Nicht notwendig
Window Slice	Zeitfensterkomprimierung	Nicht notwendig
Window Warp	Verzerrung eines Zeitfensters	Nicht notwendig
Spawner	Kombination ähnlicher Muster	Notwendig
Weighted Dynamic Time Warping (wDTW)	Kombiniert Klassenmuster mithilfe von DTW	Notwendig
Random Guided Warping	Verzerrt zeitliche Reihen durch Zufallsauswahl innerhalb der Klasse	Notwendig

Random Guided Warping Shape	Verzerrt zeitliche Reihen durch Zufallsauswahl und Form-DTW	Notwendig
Discriminative Guided Warping	Verzerrt zeitliche Reihen durch Auswahl von unterscheidenden Mustern	Notwendig
Discriminative Guided Warping Shape	Verzerrt zeitliche Reihen durch Auswahl von unterscheidenden Mustern und Form-DTW	Notwendig

A.3 Ergebnisse der Modellauswahl für ersten Anwendungsfall

Tabelle 11 Ergebnisse des Modelltrainings der jeweiligen Modelle (bestes Modell und Zeit) des ersten Anwendungsfalls zur Erkennung von Bedienfehlern beim Lichtbogenschweißen

Name	Abtastrate	Zeit	F1-Score	LEI
CNN	1	57,267	0,624	0,593
	2	74,400	0,651	0,620
	5	122,282	0,627	0,584
	8	194,039	0,610	0,550
	10	226,838	0,621	0,557
FCN	1	175,857	0,853	0,832
	2	298,310	0,875	0,833
	5	595,905	0,869	0,768
	8	942,298	0,858	0,688

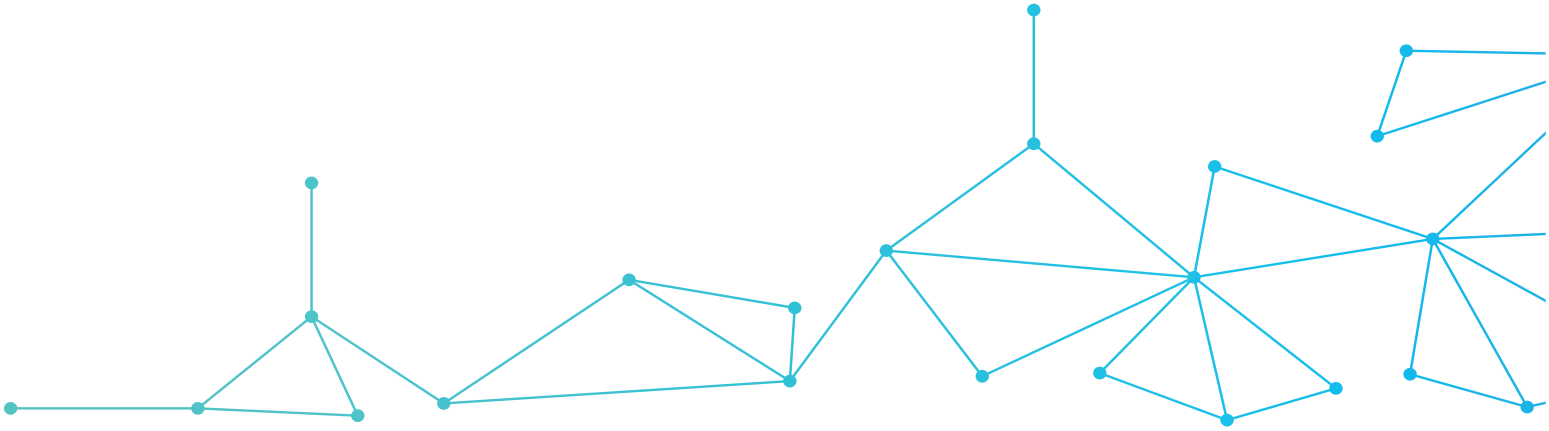
	10	1082,376	0,834	0,634
LSTMFCN	1	112,949	0,827	0,814
	2	174,900	0,855	**0,834**
	5	366,924	0,869	0,813
	8	498,690	0,868	0,786
	10	671,912	0,850	0,732
MLP	1	108,545	0,621	0,580
	2	229,745	0,619	0,554
	5	426,494	0,618	0,515
	8	539,907	0,618	0,492
	10	616,978	0,621	0,481
$MLKontext_1$	1	**16,081**	0,614	0,589
	2	17,730	0,668	0,652
	5	25,778	0,657	0,637
	8	44,209	0,712	0,696
	10	39,667	0,656	0,633
$MLKontext_2$	1	62,755	0,663	0,637
	2	67,093	0,683	0,658
	5	75,106	0,596	0,558
	8	83,384	0,680	0,652
	10	107,959	0,570	0,522
$MLKontext_3$	1	32,834	0,614	0,586
	2	37,612	0,671	0,650

	5	58,386	0,659	0,633
	8	52,353	0,674	0,651
	10	75,107	0,620	0,585
$MLKontext_4$	1	112,962	0,604	0,559
	2	94,294	0,644	0,608
	5	95,889	0,618	0,579
	8	130,965	0,693	0,658
	10	141,304	0,573	0,518
$MLKontext_5$	1	46,391	0,547	0,507
	2	36,270	0,553	0,516
	5	40,603	0,537	0,497
	8	53,315	0,505	0,458
	10	68,295	0,508	0,458
ResNet	1	307,482	0,830	0,780
	2	544,620	**0,891**	0,803
	5	1066,059	0,884	0,693
	8	1672,840	0,861	0,550
	10	1848,953	0,851	0,503

A.4 Ergebnisse der Modellauswahl für zweiten Anwendungsfall

Tabelle 12 Ergebnisse des Modelltrainings der jeweiligen Modelle (bestes Modell und Zeit) des zweiten Anwendungsfalls zur Erkennung von Druckluftleckagen bei pneumatischen Aktorbewegungen.

Name	Abtastrate	Zeit	F1-Score	LEI
CNN	1	83,860	0,926	**0,950**
	2	100,546	0,922	0,944
	5	155,652	0,924	0,944
	8	190,701	0,921	0,937
	10	197,334	0,920	0,935
FCN	1	316,980	0,923	0,935
	2	528,844	0,918	0,917
	5	1062,049	0,927	0,910
	8	1564,927	0,926	0,887
	10	1675,649	0,924	0,877
LSTMFCN	1	208,947	0,919	0,933
	2	304,670	0,907	0,909
	5	548,669	0,900	0,888
	8	755,588	0,894	0,870
	10	920,824	0,898	0,868
MLP	1	169,630	0,927	0,947
	2	193,029	0,927	0,946
	5	245,473	0,926	0,943

	8	277,519	0,926	0,941
	10	235,318	0,925	0,942
$MLKontext_1$	1	**1,635**	0,854	0,837
	2	2,966	0,899	0,910
	5	3,544	0,867	0,857
	8	5,392	0,868	0,860
	10	5,097	0,880	0,879
$MLKontext_2$	1	7,451	0,869	0,860
	2	11,711	0,908	0,924
	5	20,814	0,896	0,904
	8	31,389	0,900	0,911
	10	39,716	0,899	0,907
$MLKontext_3$	1	2,544	0,839	0,812
	2	6,430	0,910	0,928
	5	10,918	0,893	0,900
	8	10,612	0,886	0,888
	10	14,968	0,874	0,869
$MLKontext_4$	1	16,613	0,816	0,774
	2	53,185	0,914	0,933
	5	62,657	0,897	0,904
	8	54,706	0,904	0,916
	10	68,043	0,904	0,916
$MLKontext_5$	1	5,530	0,893	0,900

	2	12,128	0,916	0,936
	5	22,746	0,891	0,895
	8	19,617	0,896	0,904
	10	22,119	0,892	0,898
ResNet	1	592,609	0,925	0,926
	2	1017,517	0,925	0,908
	5	1999,471	**0,929**	0,872
	8	3139,311	0,874	0,734
	10	3356,963	0,926	0,808