



BERGISCHE  
UNIVERSITÄT  
WUPPERTAL

Fakultät für Maschinenbau und Sicherheitstechnik

# Masterthesis

im Studiengang **Sicherheitstechnik**

beim Fachgebiet für **Verkehrssicherheit und Zuverlässigkeit**

zur Erreichung des akademischen Grades

**Master of Science**

---

Thema:	<b>Vergleich modell- und datenbasierter Methoden zur Fehlerdetektion -und klassifikation im Kontext von Industrie 4.0</b>
Autor*in:	<b>Isabelle Müller</b>
MatNr.:	2013563
Bearbeitungszeitraum:	<b>14.03.2024 bis 14.08.2024</b>
Betreuer*in:	M. Sc. Tim Julitz
Erstprüfer*in:	Jun.-Prof. Dr. Antoine Tordeux

---



---

# Eidesstattliche Erklärung

*Hiermit erkläre ich, dass ich die von mir eingereichte Abschlussarbeit (Masterthesis) selbstständig verfasst und keine andere als die angegebene Quellen und Hilfsmittel benutzt sowie Stellen der Abschlussarbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen wurden, in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.*

*Ich bin damit einverstanden, dass die Arbeit durch Dritte eingesehen und unter Wahrung urheberrechtlicher Grundsätze zitiert werden darf.*

Ort und Datum: \_\_\_\_\_

Unterschrift: \_\_\_\_\_



---

# Abkürzungsverzeichnis



---

# Abbildungsverzeichnis

1.1	Heuristischer Bezugsrahmen zur vorliegenden Arbeit. In Anlehnung an [4] . . . . .	2
2.1	Industrie 1.0 bis 4.0 – zeitliche Entwicklung der Industrialisierung . . . . .	3
2.2	Struktureller Aufbau eines cyberphysischen Systems mit den zugehörigen Schnittstellen zwischen Mensch und Maschine über das Human Machine Interface (HMI) sowie der Verbindung zu anderen Systemen . . . . .	5
2.3	Darstellung der Lösungskomponenten von Industrie 4.0. Nach [3] . . . . .	5
2.4	Entwicklung von Ausfall- und Fehlfunktionsereignissen initiiert durch ein schritt- oder driftweise Veränderung des Merkmals. Nach [3] . . . . .	8
2.5	Allgemeiner Rahmen einer systemischen Darstellung der Fehlerdiagnose einschließlich der Prozessdarstellung eines dynamischen Systems. Nach [3], [15] . . . . .	9
2.6	Fehlermanagementstrategien (FM) zur Fehlerprävention und Fehlertoleranz. Nach [19] . . . . .	12
3.1	Transformationen in einem Diagnosticsystem und Einordnung von Methoden sowie Kombinationsstrategien. Nach [21], [3], [15] . . . . .	19
3.2	Cross Industry Standard Process für Data Mining . . . . .	20
4.1	Prozess- und Vorgehensmodell zur Durchführung der Fehlerdetektion und -diagnose . . . . .	24
4.2	Aufbau des Prüfstandes für die Wälzlager. Nach [34] . . . . .	25
4.3	Versuchsspezifische Aufnahmephasen zur Datengenerierung mit einer Ausnahme Frequenz von 27,6 kHz. Nach [34] . . . . .	25
4.4	Exemplarische Darstellung des Degradationsverhaltens drei getesteter Wälzlager: (a) Lager 1-1, (b) Lager 2-1, (c) Lager 3-1 . . . . .	26
4.5	Darstellung des vibrationsbasierten Degradationsverhalten von Lagern. Grundsätzlich besteht das Degradationssignal aus zwei Phase: Phase I entspricht der normalen (nicht defekten) Betriebsbedingung, Phase II entspricht der fehlerhaften (degradierten oder defekten) Betriebsbedingung. Nach [35] . . . . .	29
4.6	Struktureller Aufbau des Hidden Markov Modells . . . . .	31
4.7	Struktureller Aufbau eines CNN . . . . .	34
4.8	Darstellung des abgeleiteten Merkmals über die Betriebszustände I-III und jeweils der horizontalen und vertikalen Vibrationssignale . . . . .	38
4.9	LOWESS Glättung des abgeleiteten Merkmals über die Betriebszustände I-III und jeweils der horizontalen und vertikalen Vibrationssignale exemplarisch für (a) Lager 1-1, (b) Lager 2-1, (c) Lager 3-1 . . . . .	39
4.10	Ergebnis der Fehlerdetektion: Lager 1-1 Betriebsbedingung I . . . . .	40

4.11 Ergebnis der Fehlerdetektion: Lager 2-1 Betriebsbedingung II . . . . .	41
4.12 Ergebnis der Fehlerdetektion: Lager 3-1 Betriebsbedingung III . . . . .	42
4.13 Posterioren Übergangswahrscheinlichkeiten für Lager 3 Betriebsbedingung I . . . . .	42
4.14 Ergebnis der Fehlerdetektion: Lager 2-1 Betriebsbedingung II . . . . .	43
4.15 Ergebnis der Fehlerdetektion: Lager 3-1 Betriebsbedingung III . . . . .	44
4.16 Vergleich der Fehlerdetektionsergebnisse: Lager 1-1 Betriebsbedingung I .	44
4.17 Vergleich der Fehlerdetektionsergebnisse: Lager 2-1 Betriebsbedingung II .	45
4.18 Vergleich der Fehlerdetektionsergebnisse: Lager 3-5 Betriebsbedingung III	45
4.19 Klassenverteilung vor der Anwendung der Resampling-Techniken . . . . .	49
4.20 Klassenverteilung nach der Anwendung der Resampling-Techniken . . . . .	49
4.21 Aufteilung des Trainingsdatensatzes für die Betriebsbedingung I . . . . .	50
4.22 Aufbau der CNN Struktur mit integrierter Hyperparameter Optimierung . . .	51
4.23 Darstellung des Klassifikationsergebnis für Betriebsbedingung I: Klasse 0-Käfig; Klasse 1-Innerer Ring, Äußerer Ring; Klasse 2-Normal, Klasse 3-Äußerer Ring . . . . .	54
4.24 Darstellung Modellgenauigkeit und Modellverlust für Betriebsbedingung I .	54
4.25 Darstellung des Klassifikationsergebnis für Betriebsbedingung II: Klasse 0-Innerer Ring; Klasse 1-Käfig; Klasse 2-Normal, Klasse 3-Äußerer Ring .	55
4.26 Darstellung Modellgenauigkeit und Modellverlust für Betriebsbedingung II .	55
4.27 Darstellung des Klassifikationsergebnis für Betriebsbedingung III: Klasse 0-Innerer Ring; Klasse 1-Innerer Ring, Äußerer Ring, Käfig; Klasse 2-Normal, Klasse 3-Äußerer Ring . . . . .	56
4.28 Darstellung Modellgenauigkeit und Modellverlust für Betriebsbedingung III .	56

---

# Tabellenverzeichnis

4.1	Operating Conditions and Bearing Data . . . . .	26
4.2	Glättungsparameter für LOWESS . . . . .	38
4.3	Evaluation der RMS Werte für alle Lager . . . . .	39
4.4	Parameter der Fehlerdetektion des modellbasierten Ansatzes . . . . .	46
4.5	Parameter der Fehlerdetektion des datenbasierten Ansatzes . . . . .	47
4.6	Vergleich der Ergebnisse zwischen der statistischen Methode und dem HMM zur Fehlerdetektion . . . . .	48
4.7	Horizontale k-Fold Kreuzvalidierung mit 5 folds . . . . .	50
4.8	Best Hyperparameters for Different Operating Conditions . . . . .	52
4.9	Vergleich der Klassifikationsergebnisse anhand der Modellmetriken . . . . .	53



---

# Inhaltsverzeichnis

<b>Eidesstattliche Erklärung</b>	<b>I</b>
<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Zusammenfassung</b>	<b>XI</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Problemstellung . . . . .	1
1.2 Zielsetzung der Arbeit und Forschungsfrage . . . . .	1
1.3 Methodischer Forschungsansatz und Aufbau der Arbeit . . . . .	2
<b>2 Stand der Forschung und Terminologie</b>	<b>3</b>
2.1 Defintion und Einordnung von Industrie 4.0 . . . . .	3
2.2 Fehlerdetektion und -diagnose . . . . .	7
2.2.1 Definitionen . . . . .	7
2.2.2 Zusammenhang zwischen Prozess- und Überwachungsebene . . . . .	8
2.2.3 Charakteristiken eines Diagnosesystems . . . . .	9
2.2.4 Transformationen eines Diagnosesystems . . . . .	12
2.2.5 Maßnahmen zur Vermeidung von Ausfällen . . . . .	12
<b>3 Methoden der modell- und datenbasierten Fehlerdetektion und Diagnose</b>	<b>15</b>
3.1 Modellbasierte Ansätze . . . . .	15
3.2 Datenbasierte Ansätze . . . . .	16
3.3 Hybridansätze . . . . .	18
3.4 Ableitung eines Prozess- und Vorgehensmodells nach CRISP-DM . . . . .	20
<b>4 Durchführung ausgewählter Methoden zur Fehlerdetektion und -klassifikation</b>	<b>23</b>
4.1 Datenakquise . . . . .	23
4.1.1 Systemdarstellung und Betriebsbedingungen . . . . .	23
4.1.2 Datenstruktur und Visualisierung . . . . .	25
4.2 Auswahl geeigneter Methoden zur Fehlerdetektion und Multi-Klassen-Klassifikation	27
4.2.1 Datenvorverarbeitung . . . . .	27
4.2.2 Fehlerdetektion - modellbasierter Ansatz . . . . .	28
4.2.3 Datenbasierter Ansatz . . . . .	30
4.2.4 Diagnose . . . . .	34
4.3 Datenvorverarbeitung . . . . .	37

4.4	Implementierung der Modelle zur Fehlerdetektion . . . . .	40
4.4.1	Modellbasierte Fehlerdetektion . . . . .	40
4.4.2	Datenbasierte Fehlerdetektion . . . . .	41
4.4.3	Vergleich zwischen Fehlerdetektionsmethoden . . . . .	43
4.5	Implementierung des Modells zur Fehlerklassifikation . . . . .	49
4.5.1	Datenvorbereitung . . . . .	49
4.5.2	Datensatzaufteilung und Kreuzvalidierung . . . . .	50
4.5.3	Netzarchitektur . . . . .	51
4.5.4	Ergebnisse der Hyperparameter Optimierung . . . . .	51
4.5.5	Ergebnisse der Klassifikation . . . . .	52
<b>5</b>	<b>Limitation und Diskussion</b>	<b>57</b>
<b>6</b>	<b>Fazit</b>	<b>59</b>
	<b>Anhang</b>	<b>65</b>

---

# Zusammenfassung

**Zusammenfassung** Fehlerdetektion und Diagnose spielt eine wichtige Rolle im Kontext von modernen Industrieanwendungen. Bei der Umsetzung von FDD Methoden kommen modell- und datenbasierte Methoden zum Einsatz. Obgleich die Digitalisierung in der Industrie zu einer signifikanten Zunahme an erfassten und gespeicherten Prozessdaten führt, ist die Qualität dieser meist unstrukturiert vorliegenden Daten sowie die unmittelbare Verwendbarkeit nicht zwangsläufig gewährleistet. Im Theorieteil dieser Arbeit werden die relevanten Verknüpfungen zwischen Industrie 4.0 und der Fehlerdetektion und Diagnose beschrieben. Eine Gegenüberstellung von modell- und datenbasierten Ansätzen sowie die Fusionsstrategien werden kurz beschrieben. Eine Prozess- und Vorgehensmodell wird abgeleitet. Im Methodenteil werden geeignete Ansätze zur Umsetzung eines Diagnosesystems hergeleitet und angewendet. Die Ergebnisse werden verglichen. Darauffolgend werden Limitationen dargelegt und mögliche Lösungsansätze entwickelt. Im Fazit folgt eine Einordnung der Ergebnisse.

**Summary** Fault detection and diagnostics play an important role in the context of modern industrial applications. Model-based and data-based methods are used in the implementation of FDD methods. Although digitization in industry is leading to a significant increase in recorded and stored process data, the quality of this mostly unstructured data and its immediate usability is not necessarily guaranteed. The theory section of this work describes the relevant links between Industry 4.0 and fault detection and diagnostics. A comparison of model-based and data-based approaches as well as the fusion strategies are briefly described. A process and procedure model is derived. In the methods section, suitable approaches for implementing a diagnostic system are derived and applied. The results are compared. Subsequently, limitations are presented and possible solutions are developed. The conclusion categorises the results.

---

# 1 Einführung

Die vierte industrielle Revolution bedingt eine fortlaufende Digitalisierung, wodurch sich erhebliche Potentiale für die Überwachung industrieller Prozesse eröffnen [1]. Die mit der Digitalisierung einhergehenden exponentiell wachsenden Datenmengen führen zu einer gesteigerten Datenverfügbarkeit, welche eine umfassendere Nutzung von Systeminformationen ermöglicht [2]. Um ein hohes Maß an Verfügbarkeit produktionstechnischer Einrichtungen zu erzielen, werden Methoden der Fehlerdetektion und Diagnose herangezogen. Ziel der Fehlerdetektion ist fehlerhafte Prozessbedingungen zu identifizieren, die auf einen potentiell eingetretenen Systemfehler hinweisen können. Die Fehlerdiagnose identifiziert den konkreten Fehler und seine Ursache(n). Die exakte Lokalisierung, Quantifizierung und Bestimmung des Fehlereintrittszeitpunktes stellen dabei zentrale Elemente dar[3].

## 1.1 Problemstellung

Modellbasierte Methoden der Fehlerdetektion und Diagnose basieren auf physikalischen Modellen die auf analytischer Redundanz in Form von mathematischen Gleichungen des realen physischen Prozesses basieren. Der Erfolg im Hinblick auf die Umsetzung in Diagnosesystemen hängt maßgeblich von der Tiefe und dem Umfang des physikalischen Modells ab. In der Praxis sind genaue und vollständige Modelle nicht für alle Arten von Anwendungsfällen verfügbar. Der Hauptgrund dafür ist, dass die Modellierung großer komplexer Systeme spezifisches Fachwissen und viel menschliche Arbeit erfordert. Datenbasierte Methoden bedienen sich der Informationsextraktion von historischen Prozessdaten. Wenn die Datenverfügbarkeit, die Datenqualität sowie -quantität der historischen Prozessdaten nicht hinreichend sind, ist eine eingeschränkte Nutzung möglich oder das Ausweichen auf modellbasierte Ansätze.

## 1.2 Zielsetzung der Arbeit und Forschungsfrage

Wie verhalten sich modell- und datenbasierte Fehlerdetektions- und Klassifikationsmethoden im Vergleich zueinander in Bezug auf Genauigkeit des Modelloutputs, Interpretierbarkeit, sowie Anwendbarkeit im Rahmen industrieller Anwendungen?

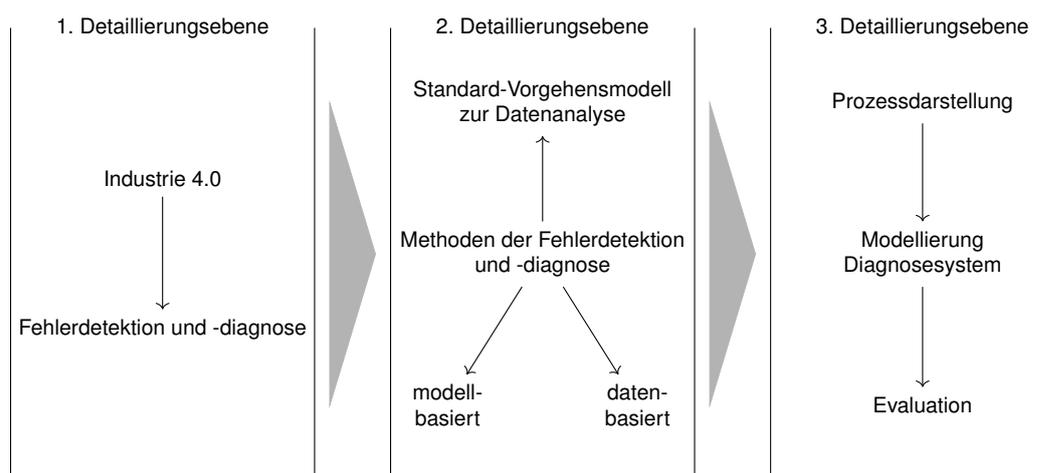
Daraus ergeben sich weitere Teilforschungsfragen, die zur Beantwortung der handlungsleitenden Forschungsfrage beitragen:

Welchen Einfluss hat die vierte industrielle Revolution im Hinblick auf die Fehlerdetektion und Diagnose? Welche daten- und modellbasierten Modelle ergeben sich für den

Einsatz von FDD und welches Vorgehensmodell lässt sich ableiten? Welche Modelle eignen sich für einen Hybridansatz zur Fehlerdetektion im anwendungsspezifischen Kontext und wie ist die Genauigkeit der Klassifikationsaufgabe?

### 1.3 Methodischer Forschungsansatz und Aufbau der Arbeit

Zur thematischen Eingrenzung und der Herausstellung relevanter Forschungsinhalte orientiert sich der Aufbau dieser Arbeit an der Methodik von Kubicek. Der heuristische Bezugsrahmen dient der Entwicklung eines konzeptuellen Entwurfs zu einem Forschungsvorhaben. Die erste Detaillierungsebene legt den Untersuchungsrahmen fest, dabei werden relevante Fachbereiche untersucht und die thematische Verknüpfung herausgestellt. Die zweite Detaillierungsebene gibt konkrete Themen an, die durch die Einarbeitung und Untersuchung relevanter Forschungsarbeiten eruiert werden sollen. Sie stellt die anzuwendende Methodik heraus. Die dritte Detaillierungsebene beschreibt das konkrete Vorgehen, die Entwicklung, sowie die Validierung des Modells.



**Abbildung 1.1** – Heuristischer Bezugsrahmen zur vorliegenden Arbeit. In Anlehnung an [4]

Auf Basis des heuristischen Bezugsrahmens kann der Aufbau der Arbeit abgeleitet werden. Im ersten Kapitel wird die thematische Einführung, die abgeleitete Problemstellung, sowie der forschungsmethodische Ansatz beschrieben. Terminologische Aspekte und die Verknüpfung relevanter Forschungsbereiche werden im zweiten Kapitel herausgestellt und beschrieben. Dabei wird der Einfluss von Industrie 4.0 auf die Fehlerdetektion und -diagnose beschrieben. Im dritten Kapitel werden konkrete Methoden der Fehlerdetektion und -diagnose dargestellt und mit einem abgeleiteten Vorgehensmodell zur Umsetzung von Datenanalyseprojekten untermauert. Im nachfolgenden Kapitel folgt die konkrete Anwendung eines modell- und datenbasierten Ansatzes zur Fehlerdetektion, die Fehlerklassifikation, sowie Evaluation. Eine kritische Bewertung der Ergebnisse erfolgt in Kapitel 5, sowie der Ausblick im letzten Kapitel.

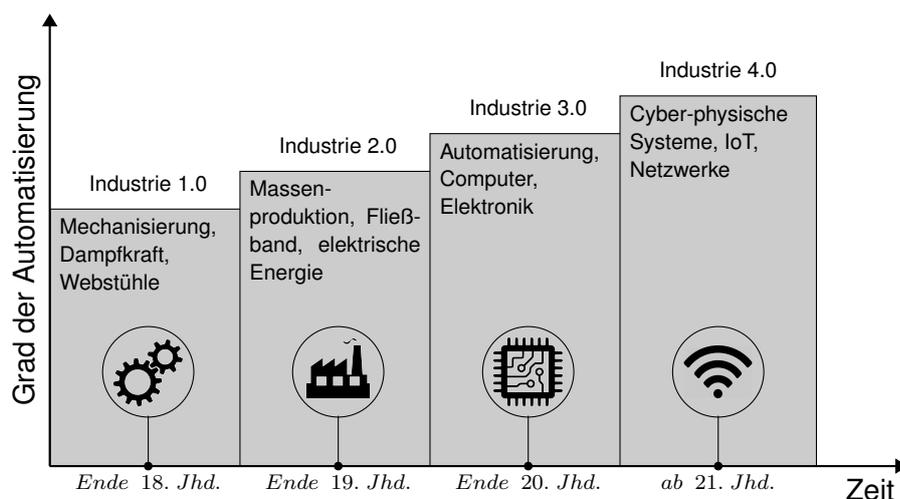
---

## 2 Stand der Forschung und Terminologie

Voranschreitende Digitalisierungsprozesse und die Vernetzung von physischen und digitalen Systemen haben einen erheblichen Einfluss auf Unternehmensstrukturen. Diese Veränderungen bezüglich der Organisations- und Prozessgestaltung werden durch die vierte industrielle Revolution eingeleitet.

### 2.1 Definition und Einordnung von Industrie 4.0

Der Begriff Industrie 4.0 wurde erstmals auf der Hannover Messe 2011 verwendet und beschreibt den anstehenden Paradigmenwechsel in der Industrie [1]. Im Gegensatz zu den vorherigen industriellen Revolutionen werden mit Industrie 4.0 die bevorstehenden Veränderungen, Herausforderungen und resultierende Chancen hervorgehoben. Die nachfolgende Abbildung 2.1 beschreibt die Entwicklung industrieller Technologien und des Organisationswandels im zeitlichen Kontext. Die erste industrielle Revolution erfolgte nach der Einführung mechanischer Produktionsanlagen und die Nutzung von Wasser- und Dampfkraft Ende des 18. Jahrhunderts. Ende des 19. Jahrhunderts führte der Einsatz von elektrischer Energie zur Einführung der Massenproduktion und Fließbandarbeit und beschreibt somit die zweite industrielle Revolution. Der Einsatz von Elektrik und IT zur Automatisierung der Produktion Ende des 20. Jahrhunderts prägen die dritte industrielle Revolution. Die vierte industrielle Revolution zeichnet sich durch die Vernetzung und Kommunikationsfähigkeit der Systeme mittels Internettechnologien aus und kann somit als Weiterentwicklung von Automatisierungs- und Produktionstechnik bezeichnet werden [5].



**Abbildung 2.1** – Industrie 1.0 bis 4.0 – zeitliche Entwicklung der Industrialisierung. Nach [5]

Die Umsetzung von Industrie 4.0 im produktionstechnischen Kontext bezieht sich im

wesentlichen mit der Ausschöpfung von Optimierungspotenzialen und der allgegenwärtigen Informationsverarbeitung. Die Innovation ergibt sich aus der Verknüpfung von bereits bestehenden Informationsquellen und die resultierende Prozess- und Optimierung [6].

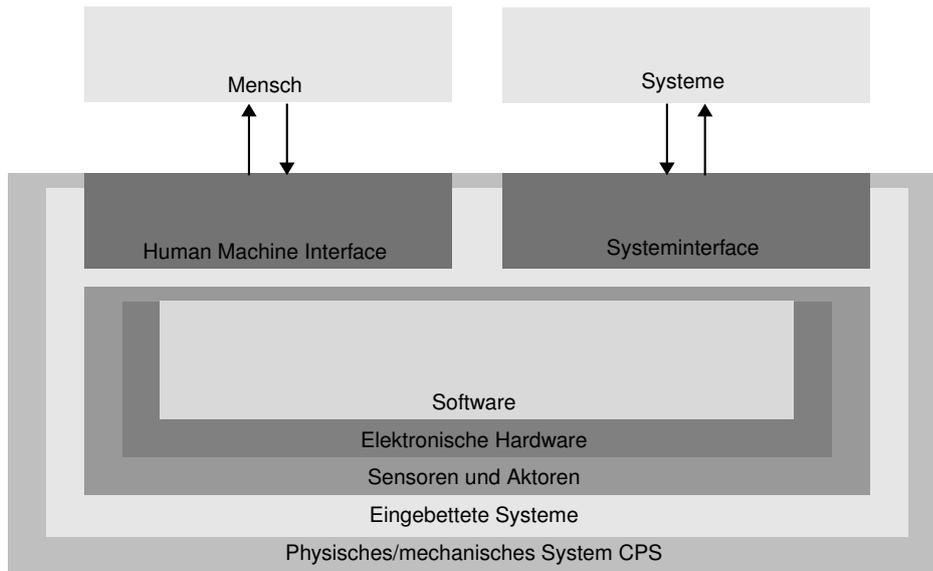
Im Hinblick auf die Umsetzung von Industrie 4.0 Anwendungen werden eine Reihe von Lösungskonzepten betrachtet. Diese inkludieren cyperphysische Systeme als intelligente Einheit in der Produktion, das Internet der Dinge und Dienste als Kommunikationsplattform für CPS, Cloud-Lösungen zur dezentralisierten Datenverwaltung sowie Big Data Lösungen für die Hochleistungsverarbeitung von großen Datenmengen [7].

Cyber-Physical Systems (CPS) sind hochvernetzte eingebettete Systeme, die eine Weiterentwicklung mechatronischer Systeme darstellen vgl. [8], [9]. Der Hauptunterschied zu eingebetteten Systemen (i.e. spezialisierte, echtzeitfähige Kombination von Hard- und Software) liegt in der Vernetzung mit anderen eingebetteten Systemen und zentralen Rechnern. CPS verfügen über Sensoren (z.B. Temperatur-, Druck- und Dehnungssensoren) und Aktoren (z.B. Stellmotoren, Radar- und Lasersysteme) und sind in der Lage, Daten in Echtzeit zu verarbeiten und zu senden. Sie bilden eine Schnittstelle zwischen der physischen und der digitalen Welt und verfügen über autonome Rechenkapazität zur Analyse und Interpretation der Daten (domänenspezifische Intelligenz) [8]. Die Datenerfassung erfolgt lokal, wobei die Daten verarbeitet und an andere Systeme sowie zentrale Rechner weitergeleitet werden. Dies ermöglicht eine Gegenüberstellung und Weiterverarbeitung der Daten sowie die Generierung neuer Einblicke in die physische Welt.

Neben der Vernetzung zwischen CPS und eingebetteten Systemen gibt es auch Sensor- und Aktornetzwerke, die physikalische Größen erfassen oder die Umgebung beeinflussen. Diese Netzwerke bestehen aus autonomen, räumlich verteilten Geräten, die drahtlos miteinander kommunizieren und Daten zu einem zentralen Datensammelpunkt senden vgl. [10].

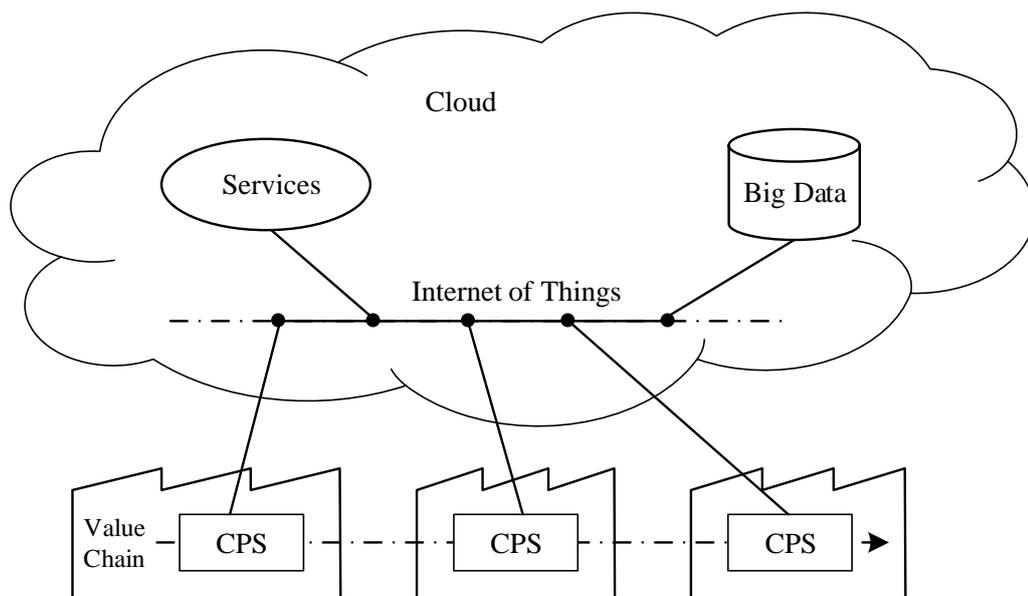
Die vertikale Integration bezeichnet allgemein die Integration der verschiedenen Systeme auf den unterschiedlichen Hierarchieebenen eines Produktionssystems. Hierbei handelt es sich beispielsweise um die Akteur- und Sensorebene, die Steuerungsebene, die Produktionsleitebene, die Manufacturing- und Execution-Ebene sowie die Unternehmensplanungsebene. Unter horizontaler Integration wird hingegen die Zusammenführung von Wertschöpfungsprozessen, d.h. abteilungsübergreifend, innerhalb eines Betriebes sowie außerhalb der Unternehmensgrenzen verstanden, um ganzheitliche Lösungen zu entwickeln vgl. [11].

Das Paradigma des Internets der Dinge (IoT) repräsentiert die Kommunikationsplattform für cyber-physische Systeme (CPS). Damit physische Objekte, die mit Kommunikationsmodulen ausgestattet sind, miteinander kommunizieren können, ist ein Verbindungs-



**Abbildung 2.2** – Struktureller Aufbau eines cyberphysischen Systems mit den zugehörigen Schnittstellen zwischen Mensch und Maschine (Human Machine Interface, kurz HMI) sowie der Verbindung zu anderen Systemen. Nach [8]

glied zwischen diesen Objekten und dem Internet erforderlich. Dieser Kommunikationsaspekt wird durch den Begriff „Internet der Dinge und Dienste“ (IoTS) beschrieben. IoTS schafft eine Verbindung zwischen physischen Objekten, dem Internet und den darauf zugreifenden Diensten, indem jedem Objekt eine eindeutige IP-Adresse zugewiesen wird (siehe Abbildung 2.3).



**Abbildung 2.3** – Darstellung der Lösungskomponenten von Industrie 4.0. Nach [3]

Durch die Einführung des Standard Internet Protocol Version 6 (IPv6) steht ausreichender Adressierungsraum für die Systemvernetzung zur Verfügung, die zur Identifizierung und Kommunikation von Geräten im Netzwerk verwendet werden. Der Hauptgrund für das langfristige Ablösen vom aktuellen Standard IPv4 (vgl. [12]) liegt am limitierten Adressierungsraum, also einer begrenzten Anzahl von verfügbaren IP-Adressen. IPv4 bietet durch das 32-Bit Format rein rechnerisch  $2^{32} (\approx 4,3 \text{ Milliarden})$  mögliche Adressen, die zufolge [13] bereits 2012 vergeben worden sind. Die nachfolgende Version IPv6 bietet hingegen eine 128-Bit Adressierung mit  $2^{128} (\approx 340 \text{ Sextillionen})$  möglichen Adressen. Das Internet der Dinge und Dienste (IoTS) schafft damit eine übergeordnete Struktur durch die Vernetzung von CPS zu cyberphysischen Produktionssystemen (entlang der Wertschöpfungskette).

Cloud-Dienste (auch Cloud Computing) ermöglichen eine zentrale Datenverwaltung und bieten neue Anwendungen, die Kapazitäten verlagern und hochverfügbare Dienste wie Cloud-Backups realisieren. Sie fördern Hochleistungsrechnen und multimediale Datenübertragungen aus verborgenen Clustern und Speichermedien. Durch die Nutzung virtualisierter Rechen- und Speicherressourcen sowie moderner Web-Technologien stellt CloudComputing skalierbare, netzwerk-zentrierte, abstrahierte IT-Infrastrukturen, Plattformen und Anwendungen als „on-demand“-Dienste zur Verfügung [14].

Unter dem Begriff Big Data wird die wirtschaftlich sinnvolle Gewinnung und Nutzung entscheidungsrelevanter Erkenntnisse aus einer Vielzahl an Informationen zusammengefasst, welche in unterschiedlicher Form vorliegen und sich durch eine hohe qualitative Vielfalt sowie eine dynamische Struktur auszeichnen. Charakteristisch für diese Daten ist zudem, dass sie in einem bislang ungekannten Umfang anfallen und einem rapiden Wandel unterliegen [2]. Daraus ergibt sich die Charakterisierung der Parameter von Big Data [14]:

- Volumen (Datenumfang)
- Velocity (Geschwindigkeit der Datenerzeugung, -verarbeitung und -übertragung)
- Variety (Vielfalt der Datentypen und -quellen)
- Value (Wert aus Nutzersicht)
- Validity (Datenqualität)

Fortgeschrittenes Datenmanagement bildet die Voraussetzung für die Prozessoptimierung. Die Bereitstellung von Informationen in Echtzeit ist die Grundlage für völlig neue Geschäftsprozesse.

---

## 2.2 Fehlerdetektion und -diagnose

Die Überwachung (engl. supervision) technischer Prozesse zielt darauf ab Ist-Zustände zu erfassen, auf unzulässige und unerwünschte Zustände hinzuweisen und geeignete Maßnahmen zur Vermeidung von Schäden und Unfällen durchzuführen. Abweichungen vom normalen Prozessverhalten resultieren aus Fehlern oder Störungen, die auf verschiedene Ursachen zurück geführt werden können. Diese führen zwangweise zu (Prozess-)Ausfällen wenn keine geeigneten Maßnahmen zur Beseitigung initiiert werden [3].

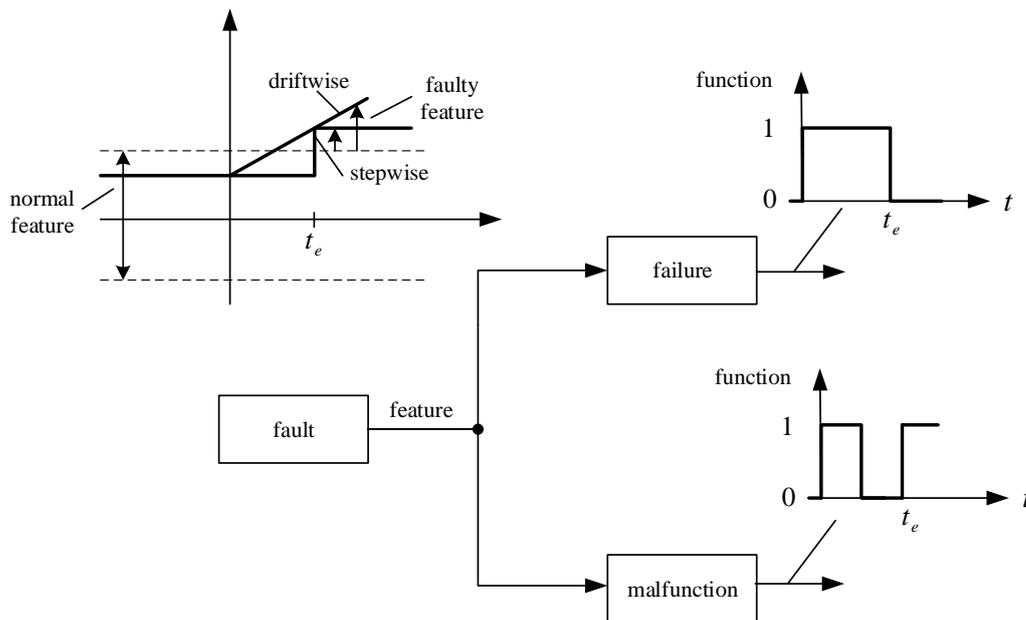
### 2.2.1 Definitionen

Die Betrachtung des Fachbereichs der Fehlerdetektion und Diagnose (kurz FDD) umfasst eine Vielzahl technologischer Disziplinen. Fehlerdetektion ist der Prozess der Identifizierung und Erkennung von Fehlern in einem System, die von den normalen Betriebsbedingungen abweichen. Ziel ist es, frühzeitig unzulässige und unerwünschte Zustände zu erkennen, um rechtzeitig Gegenmaßnahmen ergreifen zu können und größere Störungen zu vermeiden. Fehlerdiagnose ist der nachfolgende Prozess zur Fehlerdetektion, der sich mit der Bestimmung der Art, des Ortes und der Ursache des erkannten Fehlers beschäftigt. Dies beinhaltet die Analyse und Identifikation der spezifischen Bedingungen, die den Fehler verursacht haben, um gezielte Korrekturmaßnahmen zu ermöglichen. Da die Aufgabe der Diagnose als Klassifikationsproblem gesehen werden kann, wird das Diagnosesystem auch als diagnostischer Klassifikator bezeichnet (vgl. [15]).

Nach Isermann [3] wird ein Fehler (engl. fault) folgendermaßen definiert:

„Ein Fehler ist eine unzulässige Abweichung mindestens einer charakteristischen Eigenschaft (Merkmal) des Systems vom akzeptablen, üblichen Standardzustand.“ Ein Fehler kann also als Zustand in einem System definiert werden. Dabei liegt eine unzulässige Abweichung eines Toleranzfeldes vor, die durch die Differenz zwischen dem Fehlerwert (engl. fault value) und einem Schwellenwert (engl. threshold) beschrieben wird. Dabei können Fehler abrupt sowie graduell, d. h. driftweise auftreten und Ausfälle oder Fehlfunktionen zur Folge haben [3].

Gemäß der Definition in [3] stellt „ein Ausfall [...] eine dauerhafte Unterbrechung der Fähigkeit eines Systems, eine erforderliche Funktion unter bestimmten Betriebsbedingungen zu erfüllen“ dar. Ein Ausfall (engl. failure) kann also als Konsequenz von einfach oder mehrfach auftretenden Fehlern im System beschrieben werden. Unter einer Fehlfunktion (engl. malfunction) wird eine zeitlich begrenzte Unterbrechnungen verstanden. In Anlehnung an [3] lässt sich eine Fehlfunktion als eine zeitweilige Unregelmäßigkeit bei der Erfüllung der gewünschten Funktion eines Systems definieren. Die vorliegende Abbildung 2.4 veranschaulicht die Entwicklung von Ausfall- und Fehlfunktionsereignissen (engl. events) aufgrund merkmalspezifischer Abweichungen vom Normalverhalten.



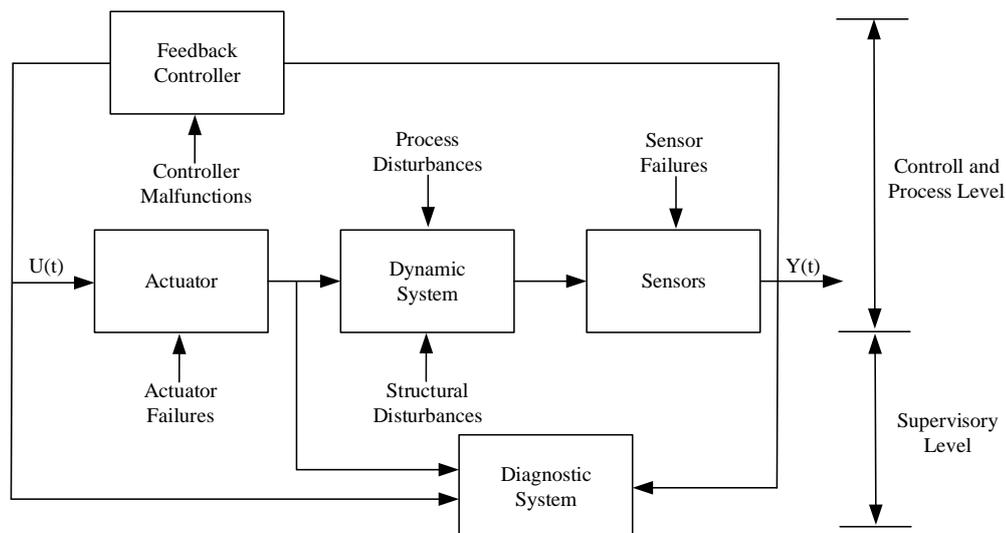
**Abbildung 2.4** – Entwicklung von Ausfall- und Fehlfunktionsereignissen initiiert durch ein schritt- oder driftweise Veränderung des Merkmals. Nach [3]

### 2.2.2 Zusammenhang zwischen Prozess- und Überwachungsebene

Die nachfolgend dargestellte Abbildung 2.5 zeigt einen überwachten Prozess und weist auf verschiedene Fehlerquellen in diesem System hin. Betrachtet wird dabei die Prozess- und Kontrollebene, sowie die Überwachungsebene i.e. das Diagnosesystem. Das Diagnosesystem wird als übergeordneter Begriff dargestellt, der die Aufgaben und Ziele der Fehlerdetektion und Diagnose (FDD) anwendungsspezifisch umfasst.

Sensoren geben relevante physikalische Informationen des Systems wie beispielsweise Temperaturen, Drücke, Kräfte als Ausgangssignal  $Y(t)$  wieder. Die Verarbeitung der Größe erfolgt durch den Feedback-Controller, woraufhin der Aktuator mit dem Stellsignal, d. h. dem Eingangssignal  $U(t)$ , beaufschlagt wird. Der Aktuator wird folglich für die Erzeugung von Bewegungen oder das Aufbringen von Kräften und Momenten benötigt. Die Stellenergie des Aktuators wirkt anschließend auf das dynamische System, beispielsweise eine Anlage [16].

Fehlerhafte Zustände können nun einzelne oder mehrere Komponenten des Prozesses betreffen, sowie externe und interne Ursachen haben. Prozessstörungen beeinflussen das dynamische System beispielweise durch äußere Umwelteinflüsse wie Staub oder Feuchtigkeit. Strukturelle Störungen können interne Ursachen haben, wie zum Beispiel fehlende Schmierung und daraus resultierende höhere Reibung oder Abnutzung, Überhitzung oder Leckagen. Fehlerzustände in den dargestellten Instrumenten (i.e. Aktuator, Sensor und Feedback Controller) können dazu führen, dass die Zustandsvariablen



**Abbildung 2.5** – Allgemeiner Rahmen einer systemischen Darstellung der Fehlerdiagnose einschließlich der Prozessdarstellung eines dynamischen Systems. Nach [3], [15]

des Systems über die zulässigen Grenzen hinaus abweichen. Diese Fehler beeinflussen zuerst die internen Prozessparameter wie Änderungen von Widerständen oder Kapazitäten und/oder interne Zustandsgrößen wie Änderungen von Massenströmen oder Temperaturen. Aufgrund des dynamischen Prozessstransferverhaltens verändern diese Fehler die messbare Ausgangsgröße  $Y(t)$ . Zusätzlich sei angemerkt, dass natürliche Prozessstörungen und Rauschen, sowie Änderungen der Stellgröße  $U(t)$  die Ausgangsgröße  $Y(t)$  beeinflussen [3], [15]. Das Ziel ist also durch die Verwendung aller verfügbarer Messungen mehr Informationen über den Prozess zu gewinnen und diese in Form von mathematischen Prozessmodellen miteinander in Beziehung zu setzen. Werden nicht nur Ausgangssignale  $Y(t)$  gemessen, sondern auch die entsprechenden Eingangssignale  $U(t)$ , zusätzliche interne Zustandsgrößen und gegebenenfalls Störsignale, dann können Änderungen des statischen und dynamischen Verhaltens der Prozesse durch die Störungen als wichtige Informationsquelle für das Diagnosesystem genutzt werden.

### 2.2.3 Charakteristiken eines Diagnosesystems

Um verschiedene Diagnoseansätze zu vergleichen, ist es sinnvoll, Charakteristiken für FDD zu identifizieren. Diese Merkmale ermöglichen die Bewertung der Ansätze anhand gemeinsamer Anforderungen oder Standards. Obwohl eine einzelne Methode selten alle Merkmale erfüllt, helfen sie, Methoden hinsichtlich der benötigten A-priori-Informationen, der Zuverlässigkeit, der Allgemeinheit und der Berechnungseffizienz, etc. zu vergleichen. Wichtige Begriffe in diesem Zusammenhang sind Vollständigkeit und Auflösung. Vollständigkeit bedeutet, dass der tatsächliche Fehler in der vorgeschlagenen Fehlermen-

ge enthalten ist. Auflösung bedeutet, dass diese Fehlermenge so klein wie möglich ist. Es besteht ein Kompromiss zwischen diesen beiden Konzepten, da die Genauigkeit der Vorhersagen von beiden beeinflusst wird. Diese Konzepte sind entscheidend bei der Bewertung von Diagnoseklassifikatoren. Im Folgenden werden wünschenswerte Merkmale eines Diagnosesystems vorgestellt:

### **Schnelle Detektion und Diagnose**

Eine Diagnosesystem sollte in der Lage sein schnelle Veränderung zu detektieren und Prozessfehlfunktionen zu diagnostizieren. Diese schnelle Reaktion steht mit der tolerierbaren Performance in Konflikt. Wenn ein System auf schnelle Änderungen reagiert, reagiert es ebenfalls auf hochfrequente Einflüsse. Dies macht das System empfindlich gegen Rauschen und kann im Normalbetrieb zu häufigen, störenden Fehlalarmen führen [15]. Analog dazu kann der Kompromiss zwischen Robustheit und Genauigkeit genannt werden. Robustheit bezieht sich auf die Fähigkeit eines Systems, zuverlässig zu funktionieren, auch wenn es äußeren Störungen oder Unsicherheiten ausgesetzt ist. Genauigkeit hingegen misst, wie präzise das System spezifische Fehler oder Anomalien identifiziert. In vielen Fällen verbessert eine erhöhte Robustheit die Widerstandsfähigkeit gegen Störungen, kann jedoch die Genauigkeit verringern, da das System konservativer agiert, um Fehlalarme zu minimieren. Dieses Phänomen wird als Robustness-Accuracy Trade-off beschrieben vgl. [17].

### **Isolierbarkeit**

Isolierbarkeit ist die Fähigkeit eines Diagnosesystems, zwischen verschiedenen Fehlern zu unterscheiden. Unter idealen Bedingungen, frei von Rauschen und Modellierungsunsicherheiten, bedeutet dies, dass der Diagnoseklassifikator in der Lage sein sollte, eine Ausgabe zu erzeugen, die eindeutig von nicht aufgetretenen Fehlern unterscheidbar ist. Die Entwicklung isolierbarer Klassifikatoren hängt stark von den Eigenschaften des Prozesses ab [15]. Ein entscheidender Aspekt ist der Kompromiss zwischen Isolierbarkeit und der Robustheit gegenüber Modellierungsunsicherheiten [18].

### **Erkennbarkeit der Neuartigkeit**

Eine der grundlegenden Anforderungen an ein Diagnosesystem ist die Fähigkeit, den aktuellen Zustand des Prozesses zu bewerten und zu bestimmen, ob er sich normal oder abnormal verhält. Bei einer Abweichung vom Normalverhalten muss das System ferner in der Lage sein zu unterscheiden, ob die Ursache auf eine bekannte Störung oder auf eine neuartige, bislang unbekannte Anomalie zurückzuführen ist. Dieses Kriterium wird als Erkennbarkeit der Neuartigkeit bezeichnet. In der Regel stehen ausreichend Daten zur Verfügung, um das normale Verhalten des Prozesses zu modellieren. Allerdings sind historische Daten über anomale Zustände oft unzureichend, um ein vollständiges Modell der anormalen Bereiche zu erstellen. Verfügt man über ein gutes dynamisches Modell des Prozesses, lassen sich solche Daten leichter generieren. Häufig sind jedoch nur wenige Datensätze vorhanden, die lediglich Teile des abnormalen Bereichs abdecken, was dazu

---

führt, dass ein großer Teil der möglichen anomalen Zustände nicht adäquat modelliert wird. Dies stellt eine erhebliche Herausforderung für die Erkennbarkeit von Neuartigkeit dar. Auch unter diesen schwierigen Bedingungen soll das Diagnosesystem in der Lage sein, das Auftreten neuartiger Fehler zu erkennen und sie nicht fälschlicherweise als eine der bekannten Fehlfunktionen oder als normalen Betrieb zu klassifizieren. Die Fähigkeit, neuartige Fehler zu erkennen, ist essenziell, um zuverlässige und robuste Diagnosesysteme zu entwickeln, die in dynamischen und sich verändernden Umgebungen effektiv arbeiten können [3], [15].

### **Adaptivität**

Prozesse unterliegen im Allgemeinen Veränderungen und Entwicklungen aufgrund externer Eingaben oder struktureller Anpassungen wie Nachrüstungen. Die Betriebsbedingungen eines Prozesses können sich nicht nur aufgrund von Störungen, sondern auch durch veränderte Umgebungsbedingungen ändern, wie beispielsweise Änderungen in den Produktionsmengen bei wechselnden Anforderungen oder Schwankungen in der Qualität des Rohmaterials. Ein Diagnosesystem sollte daher an solche Veränderungen anpassbar sein. Es sollte möglich sein, den Umfang des Systems schrittweise zu erweitern, wenn neue Fälle und Probleme auftreten und mehr Informationen verfügbar werden. Dies ermöglicht eine kontinuierliche Verbesserung und Anpassung des Systems an die sich verändernden Prozessbedingungen [3], [15].

### **Modellierungsaufwand und Modellgröße**

Der Umfang des Modellierungsaufwands für die Entwicklung eines diagnostischen Klassifikators ist ein entscheidendes Thema. Um diagnostische Klassifikatoren in Echtzeit schnell und effizient einsetzen zu können, sollte der Modellierungsaufwand so gering wie möglich gehalten werden. Echtzeitlösungen erfordern typischerweise Algorithmen und Implementierungen, die weniger rechnerisch komplex sind, jedoch einen hohen Speicherbedarf haben. Daher wird ein Diagnosesystem bevorzugt, das ein angemessenes Gleichgewicht zwischen diesen beiden konkurrierenden Anforderungen findet. Dies bedeutet, dass das System sowohl die Rechenzeit als auch den Speicherbedarf optimieren muss, um eine effiziente und schnelle Fehlererkennung und -diagnose zu gewährleisten [15].

**Erkennung von Mehrfachfehlern** Die Fähigkeit zur Erkennung von mehreren Fehlern ist eine wichtige, aber schwierige Anforderung. Dieses Problem ist besonders herausfordernd, weil die meisten Fehler miteinander interagieren. In einem allgemeinen nichtlinearen System sind die Wechselwirkungen normalerweise synergistisch, sodass ein Diagnosesystem die einzelnen Fehlermuster nicht zur Modellierung der kombinierten Wirkung der Fehler verwenden kann. Andererseits ist die Aufzählung und der Entwurf für verschiedene Fehlerkombinationen bei großen Prozessen aufgrund der kombinatorischen Explosion unzumutbar [15].

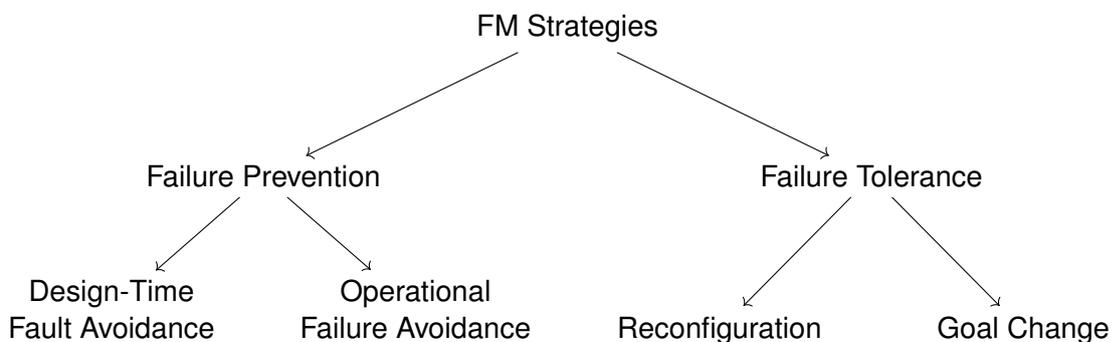
### 2.2.4 Transformationen eines Diagnosesystems

Im Allgemeinen kann der diagnostische Entscheidungsprozess als eine Reihe von Transformationen oder Abbildungen betrachtet werden. Zunächst existiert der Messraum, in dem die Messungen  $x_1, x_2, \dots, x_N$  als Eingaben des Diagnosesystems betrachtet werden. Diese Messungen enthalten zunächst kein a priori Wissen über den Prozess. Der Merkmalsraum hingegen stellt eine Transformation der Messungen in Merkmale  $y_i$  dar, wobei a priori Wissen verwendet wird, um relevante Informationen zur Diagnose zu extrahieren. Die Abbildung vom Merkmalsraum in den Entscheidungsraum erfolgt typischerweise durch Zielfunktionen wie die Minimierung der Fehlklassifikation, unter Einsatz von Diskriminanzfunktionen oder Schwellenwertfunktionen. Schließlich wird im Klassenraum das Diagnoseresultat interpretiert, indem die Entscheidungsklassen  $c_1, c_2, \dots, c_M$  bestimmt werden, welche die Fehlerklassen und den Normalzustand repräsentieren [15].

Zur Entwicklung des Merkmalsraums aus dem Messraum gibt es zwei Ansätze: Merkmalsauswahl und Merkmalsextraktion. Bei der Merkmalsauswahl werden wichtige Messungen direkt aus dem Messraum ausgewählt. Die Merkmalsextraktion hingegen reduziert den Messraum durch die Verwendung von Vorwissen in einen Raum geringerer Dimension. Beispielsweise kann eine Beziehung zwischen Messungen genutzt werden, um die Dimensionen zu reduzieren und somit die Merkmale effizienter darzustellen [15].

### 2.2.5 Maßnahmen zur Vermeidung von Ausfällen

Um Maßnahmen zur Vermeidung von Systemausfällen durchzuführen, werden bestimmte Anforderungen an die zu betrachteten Systeme gestellt. Da bei jeder Systemfunktion die Möglichkeit eines Ausfalls besteht, und das Ziel, die Funktionalität angesichts eines Ausfalls zu erhalten, besteht die Notwendigkeit der Überlegung, wie diese erhalten bleiben oder geschützt werden können. Betrachtet wird eine übergeordnete Ebene, die zwei Möglichkeiten zum Funktionserhalt bietet: die Verhinderung eines Ausfalls oder die Tolerierung eines Ausfalls wie in Abbildung 2.6 nach [19] dargestellt.



**Abbildung 2.6** – Fehlermanagementstrategien (FM) zur Fehlerprävention und Fehlertoleranz. Nach [19]

---

Die Ausfallprävention kann durch Maßnahmen in der Entwicklung oder im Betrieb erreicht werden. In der Entwicklung können bereits mögliche Ausfallursachen durch z. B. Änderungen an Bauteilen oder durch Qualitätssicherungsmaßnahmen erreicht werden. Zu den betrieblichen Ausfallvermeidung gehören Fehlerzustände zu detektieren, Fehlerklassen zu bestimmen, Prognosen über Ausfallzeitpunkte zu tätigen und die Ergreifung betrieblicher Maßnahmen, um sein Auftreten zu verhindern. Diese betrieblichen Maßnahmen können die Stilllegung des Systems vor dem Ausfall umfassen, wenn eine Reparatur nicht möglich ist, eine Änderung des Systembetriebs, um den unvermeidlichen Ausfall zu verzögern (Verringerung der Belastung der beanspruchten Komponenten) oder Reparatur oder Ersatz der beanspruchten Komponente vor dem Ausfall.

Zu den Strategien der Ausfalltoleranz gehören die Rekonfiguration und die Zieländerung. Die Strategie zur Funktionswiederherstellung oder Rekonfiguration basiert auf der Möglichkeit Funktionen über Redundanzen zu erfüllen. Die Systemziele werden bei der Strategie nicht verändert. Die Änderungsstrategie wird angewendet, wenn die Auswirkungen des Ausfalls so groß sind, dass die aktuellen Systemfunktionen nicht aufrechterhalten werden können. Bei dieser Strategie wechselt das System zu weniger anspruchsvollen Zielen z. B. auf einen SSicherungsModus zur Erhaltung der Energieversorgung.

Die Ausfallprävention kann durch Maßnahmen in der Entwicklung oder im Betrieb erreicht werden. In der Entwicklung können bereits mögliche Ausfallursachen durch Änderungen an Bauteilen oder durch Qualitätssicherungsmaßnahmen minimiert werden. Dazu gehören das Design robuster Komponenten, die Auswahl hochwertiger Materialien und die Implementierung strenger Testverfahren während der Entwicklungsphase.

Zu den betrieblichen Ausfallvermeidungsmaßnahmen gehört es, Fehlerzustände zu detektieren, Fehlerklassen zu bestimmen, Prognosen über Ausfallzeitpunkte zu tätigen und geeignete betriebliche Maßnahmen zu ergreifen, um das Auftreten von Ausfällen zu verhindern. Diese Maßnahmen umfassen die Stilllegung des Systems vor dem Ausfall, wenn eine Reparatur nicht möglich ist, die Änderung des Systembetriebs zur Verzögerung eines unvermeidlichen Ausfalls (z. B. durch Verringerung der Belastung der beanspruchten Komponenten), oder die Reparatur bzw. der Ersatz der betroffenen Komponente vor dem Ausfall z. B. über eine prädiktive Instandhaltungsstrategie. Prädiktive Instandhaltung basiert auf der kontinuierlichen Überwachung von Anlagenzuständen und der Analyse von Daten, um den optimalen Zeitpunkt für Wartungsmaßnahmen vorherzusagen. Dies erfolgt durch die Integration von Fehlerdetektion und Diagnose (FDD) mit fortschrittlichen Prognosemodellen [20].



---

# 3 Methoden der modell- und datenbasierten Fehlerdetektion und Diagnose

Übersicht über modell- und datenbasierte Methoden der Fehlerdetektion und Diagnose. Methoden zur Fehlerdetektion und -diagnose unterscheiden sich im Wesentlichen durch folgende Ansätze [20]:

- wissensbasiert
- datenbasiert
- modellbasiert (basierend auf physikalischen Modellen)
- hybride Ansätze (Kombinationen aus verschiedenen Modellen)

Nachfolgend werden modell- und datenbasierte Methoden mit einander verglichen.

## 3.1 Modellbasierte Ansätze

FDD-Methoden, die auf physikalischen Modellen beruhen, bauen auf analytischer Redundanz in Form von mathematischen Gleichungen des realen physikalischen Prozesses [21]. Die mathematischen Gleichungen beruhen auf dem Wissen über die Input-Output-Beziehungen des Prozesses. Aufgrund dieser bekannten Input-Output-Beziehungen benötigt das Modell im Vergleich zu datengesteuerten Ansätzen weniger Prozessdaten. Der Betriebszustand des Prozesses wird durch den Vergleich von Messungen des realen Prozesszustands mit dem durch das physikalische Modell vorhergesagten Zustand. Ein inkonsistenter Vergleich weist auf fehlerhafte Bedingungen hin. Konkrete Inkonsistenzen werden als Residuen bezeichnet. Die Residuen entsprechen den im Arbeitsablauf gezeigten Merkmalen. Sie zeigen in der Regel zuverlässig fehlerhaftes Prozessverhalten an, da sie so konzipiert sind, dass sie fehlerempfindlich sind und Rauschen und Unsicherheiten tolerieren [21].

Diese Residuen werden als Eingabe für Entscheidungsregeln verwendet. Diese Regeln prüfen in der Regel Schwellenwerte für bestimmte Residuen, um Fehlersymptome zu erzeugen. Die anschließende Fehlerdiagnose kann z. B. mit Schwellenwert-Logikfunktionen oder Fehlerkarten durchgeführt werden. Typische auf physikalischen Modellen basierende Methoden für FDD sind Bondgraphen, Paritätsgleichungen, Parameterschätzung, Zustandsschätzung (Kalman-Filter (KF)) und Zustandsbeobachter [3], [15]. Der Erfolg solcher modellbasierter Ansätze hängt von der Tiefe und dem Umfang des physikalischen Modells ab. In der Praxis sind genaue und vollständige Modelle nicht

für alle Arten von Anwendungsfällen verfügbar. Der Hauptgrund ist, dass die Modellierung großer komplexer Systeme Expertenwissen und viel menschliche Arbeit erfordert. Die Zustandsschätzung mittels Kalman-Filter stellt eine weit verbreitete für die Fehlererkennung in Systemen dar. Bei Zustandsschätzungsverfahren wird ein mathematisches Modell des Systems erstellt und zur Schätzung des aktuellen Zustands des Systems verwendet. Unterschiede zwischen dem geschätzten und dem tatsächlichen Zustand werden zur Erkennung von Fehlern verwendet. Es kann vergangene, aktuelle und zukünftige Systemzustände vorhersagen, ohne dass das modellierte System im Detail bekannt sein muss [22]. Bei der Verwendung von beobachtergestützten Methoden wird ein dynamisches Modell des Systems verwendet, um die Ausgaben auf der Grundlage der Eingaben zu schätzen und sie mit den tatsächlichen Ausgaben zu vergleichen, um Abweichungen zu erkennen, die auf Fehler hindeuten könnten. Eine weitere Methode ist die Fehlererkennung mittels Grenzwert- und/oder Trendüberprüfung [3]. Im Allgemeinen werden zwei Grenzwerte, auch Schwellenwerte genannt, vorgegeben, ein maximaler Wert und ein Minimalwert. Ein normaler Zustand ist dann gegeben, wenn die überwachte Variable innerhalb einer bestimmten Toleranzzone liegt. Das Überschreiten einer der Schwellenwerte deutet dann auf einen Fehler hin. Durch die erste Ableitung der überwachten Variable lässt sich der Trend prüfen. Die Paritätsraum-Methoden

Beschreibung: Bei diesen Methoden werden auf der Grundlage von Systemmodellen Residuen (d. h. die Differenz zwischen beobachteten und erwarteten Ausgaben) erzeugt. Die Residuen werden dann analysiert, um Fehler zu erkennen und zu isolieren. Beispiele: Paritätsgleichungen: Zur Überwachung der Konsistenz in Sensornetzen und Kontrollsystemen durch Vergleich von Messungen mit modellbasierten Vorhersagen.

#### 5. Methoden der Parameter-Schätzung:

Beschreibung: Hierbei werden die Systemparameter (z. B. Widerstand, Kapazität) in Echtzeit geschätzt und mit den erwarteten Werten verglichen. Signifikante Abweichungen können auf einen Fehler hinweisen. Beispiele: Schätzung der kleinsten Quadrate: Wird häufig zur Schätzung von Modellparametern verwendet, wobei Abweichungen von den erwarteten Parametern auf mögliche Fehler hinweisen. Rekursive kleinste Quadrate: Eine Erweiterung der kleinsten Quadrate, die für Echtzeit-Parameterschätzungen und Fehlererkennung verwendet wird.

## 3.2 Datenbasierte Ansätze

Datenbasierte Methoden extrahieren Informationen aus historischen Prozessdaten [21].

Grundsätzlich werden datenbasierte Methoden in überwachte, unüberwachte und semi-überwachte Ansätze unterteilt. Zu den überwachten Ansätzen zählen Neuronale Netze, Stützvektor Maschinen und auch Regressionstechniken. Clustertechniken sowie

---

statistische Modell zählen zu den unüberwachten Methoden.

Überwachte Methoden wenden Algorithmen auf historische Daten an, um Muster innerhalb des hochdimensionalen Raum von Prozessmessungen oder Merkmalen zu lernen und zu unterscheiden [21]. Diese Erkenntnisse dienen dann der Fehlererkennung und der Diagnose, d.h. zur Vorhersage von Fehlersymptomen (i.e. Klassen). Voraussetzung zur Anwendung von überwachten Methoden ist, dass gelabelte Daten vorliegen. Also für jeden Eingabe-Merkmalvektor muss die zugehörige Klassen bekannt sein [21]. Neuronale Netze (z. B. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs)) gehören zu den überwachten Klassifikatoren, die zur Fehlerdetektion und Diagnose beispielsweise für die Multi-Klassen-Klassifikation eingesetzt werden kann. Weitere Methoden wie SVMs, k-Nearest Neighbors oder Decision Trees können ebenfalls für diesen Anwendungsfall genutzt werden [23]. Die Klassen dieses Problems entsprechen der Fehlerart entsprechend der Ausgabe der Fehlerdiagnose. Dabei kann der Mess- oder Merkmalsraum direkt in das NN eingespeist werden. Die Ausgangsneuronen des NN repräsentieren den Entscheidungsraum [15]. Der Aufbau von NN ist ein zeitaufwendiger und iterativer Prozess. NN (und auch andere machine learning Algorithmen) werden durch eine Reihe von Hyperparametern konfiguriert, deren Werte sorgfältig ausgewählt werden müssen und die oft erhebliche Auswirkungen auf die Leistung haben. Um ein zeitaufwändiges und nicht reproduzierbares manuelles zu vermeiden, können verschiedene automatische Hyperparameter-Optimierungsverfahren (HPO) eingesetzt werden, um gut funktionierende Hyperparameter-Konfigurationen zu finden [24]. Zu den gängigsten Optimierungsstrategien zählen Gridsearch, Randomsearch, Bayesian Optimization oder Hyperband [24].

Unüberwachte Methoden benötigen keine Klassenbezeichnungen und sind somit für eine breiteren Anwedungsraum geeignet, in denen diese nicht verfügbar sind. Die meisten unüberwachten Methoden konzentrieren sich nur auf den ersten oder zweiten Transformationsschritt, zur Extraktion von Merkmalen oder Fehlersymptomen. Die Hauptkomponentenanalyse oder kurz *PCA Principle Component Analyse* bezeichnet die gängigste Vorgehensweise zur Dimensionsreduktion [25]. Mit zunehmender Dimensionalität steigt die erforderliche Datenmenge, um das gleiche Maß an Informationsdichte zu erhalten, exponentiell an (vgl. [26]). Die Komplexität steigt mit der Dimensionalität und verursacht deutlich erhöhte Rechenzeiten, sowie tendieren hochdimensionale Daten zu Überanpassung. Die Anzahl redundanter Merkmale steigt, wodurch die Extraktion relevanter Merkmale erschwert wird. Die Methode der Dimensionsreduktion (engl.: *dimension reduction*) bietet sich zur Lösung dieser Probleme an, da sie die Datentransformation von einem hochdimensionalen in einen niedrigdimensionalen Raum ermöglicht. Die Darstellung erhält dabei möglichst die intrinsische Dimension (engl. *intrinsic dimension*), also die aussagekräftigen Eigenschaften der ursprünglichen Daten [25].

Diese Hauptkomponenten werden verwendet, um Residuen, die als Merkmale  $f_i$  für

die anschließende Generierung von Fehlersymptomen  $s_i$  dienen [21].

Hidden Markov Modelle sind probabilistische Modelle: -> kann modell- und datenbasiert verwendet werden

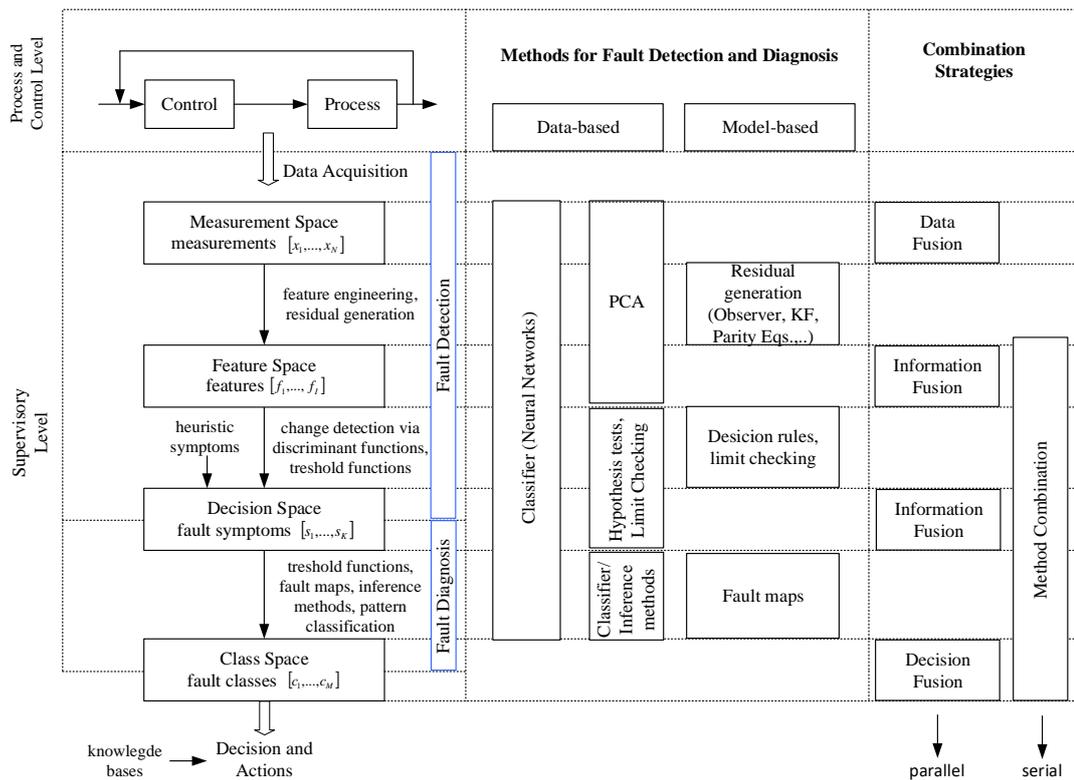
Datenbasierte Ansätze erfordern kein spezifisches Dömanenwissen für die Entwicklung eines physikalischen Modells. Ein weiterer Vorteil ist, dass sie mit dochdimensionale Daten umgehen kann. Die Leistung hängt jedoch stark von der Datenqualität und -quantität zusammen. Die Datenqualität ist jedoch im produktionstechnischen Kontext häufig unzureichend vgl. [27].

Die datenbasierte Herausforderungen für FDD in der Produktion sind häufig wechselnde Prozessbedingungen, eine unzureichende Datenerfassung mit zu geringem Informationsgehalt, Messrauschen, unausgewogene Datenverteilungen und die Tatsache, dass historische Daten oft nur eine Teilmenge aller möglichen Fehlerklassen enthalten [21]. Das Problem einer Unausgewogenheit ist bei Klassifikationsaufgaben, dass die überrepräsentierte Klasse für das Modell überwältigend ist und die unterrepräsentierte Klasse ignoriert wird. Dies hat Einfluss auf die Modellgenauigkeit. Abhilfemaßnahmen, die auf dem Ausgleich der Klassenverteilung Verteilung beruhen, umfassen verschiedene Techniken der Neuauswahl von Daten, wie zufälliges Oversampling (mit Ersetzung) der seltenen Klasse, zufälliges Downsampling der vorherrschenden Klasse (ohne Ersetzung) [28], oder die Erstellung von synthetischen Daten der unterrepräsentierten Klasse ohne Ersetzen auf Basis der k-nächsten Nachbarn namens SMOTE (Synthetic Minority Over-sampling technique) [29]. Eine weitere Resampling-Strategie ist die Kreuzvalidierung. Sie dient der Vermeidung von Überanpassung, wobei das Modell die Generalisierungsfähigkeit gegenüber ungesehen Daten verliert. Bei der Kreuzvalidierung wird der Datensatz zufällig in  $k$  Teilmengen (engl.: *folds*) aufgeteilt [30]. Spezifische Methoden der Kreuzvalidierung werden in [31] dargestellt.

Datengetriebene Methoden nicht für alle Arten von FDD-Anwendungsfällen geeignet. Immer dann, wenn die Verfügbarkeit und Qualität historischer Daten nicht ausreicht, kommen Ansätze ins Spiel, die z. B. auf physikalischen Modellen basieren.

### 3.3 Hybridansätze

Wie in den vorherigen Abschnitten erläuteter, haben alle FDD-Methoden Vor- und Nachteile. Da es keine universelle Methode, besteht die Motivation in der Entwicklung von sogenannten Hybridansätzen, die sich der Kombination mehrerer Modelle bedient. Daraus folgt die Erhöhung der Modellgenauigkeit und Robustheit und der Effektivität des Diagnosesystems. Abbildung 3.1 zeigt die Gegenüberstellung der Transformationen eines Diagnosesystems, die Einordnung gängiger Methoden und die zugehörige Kombinationsstrategie. Bei seriellen Kombinationsstrategien werden Fehlererkennungs- und/oder



**Abbildung 3.1** – Transformationen in einem Diagnosticsystem und Einordnung von Methoden sowie Kombinationsstrategien. Nach [21], [3], [15]

Fehlerdiagnoseverfahren hintereinander geschaltet, wobei der Output des vorherigen Verfahrens als Input für das nächste Verfahren dient. Dies ermöglicht die Kombination verschiedener Ansätze entlang der Transformationsschritte von der Erfassung der Prozessmessungen bis zur Entscheidungsfindung. Durch die serielle Methodenfusion wird die Gesamteffizienz gesteigert, da die Ergebnisse jeder vorherigen Ausgabe sequentiell neu bewertet werden. [21].

Parallele Kombinationsstrategien fusionieren die Ergebnisse mehrerer FDD-Methoden, die parallel und unabhängig voneinander ausgeführt wurden. Jede einzelne FDD-Methode geht das FDD-Problem mit ihrer eigenen Strategie an und leitet ihre Entscheidungsfindung unabhängig von anderen Methoden. Solche Ansätze sind im Allgemeinen stabiler und zuverlässiger, da sie eine Art Methodenredundanz schaffen. Diese Redundanz kompensiert Fehler oder Ungenauigkeiten einzelner Methoden und verbessert die Entscheidungsfindung unter Unsicherheit [21]. Eine parallele Ausgangsfusion kann für jeden der Transformationsräume die in Abbildung 3.1 dargestellt werden, durchgeführt werden.

### 3.4 Ableitung eines Prozess- und Vorgehensmodells nach CRISP-DM

Im Hinblick auf die in Kapitel 2.1 dargestellte Notwendigkeit der Datenanalyse zur Nutzbarkeit relevanter Informationen unstrukturierter Daten ist ein standardisiertes Vorgehensmodell erforderlich. Der CRISP-DM Prozess (*cross industry standard process for data mining*) [32] bezeichnet ein anwendungsneutrales *best practise* Modell zur Strukturierung von Data-Mining Anwendungen. CRISP-DM organisiert den Prozess in sechs Phasen wie in Abbildung 3.2 dargestellt.

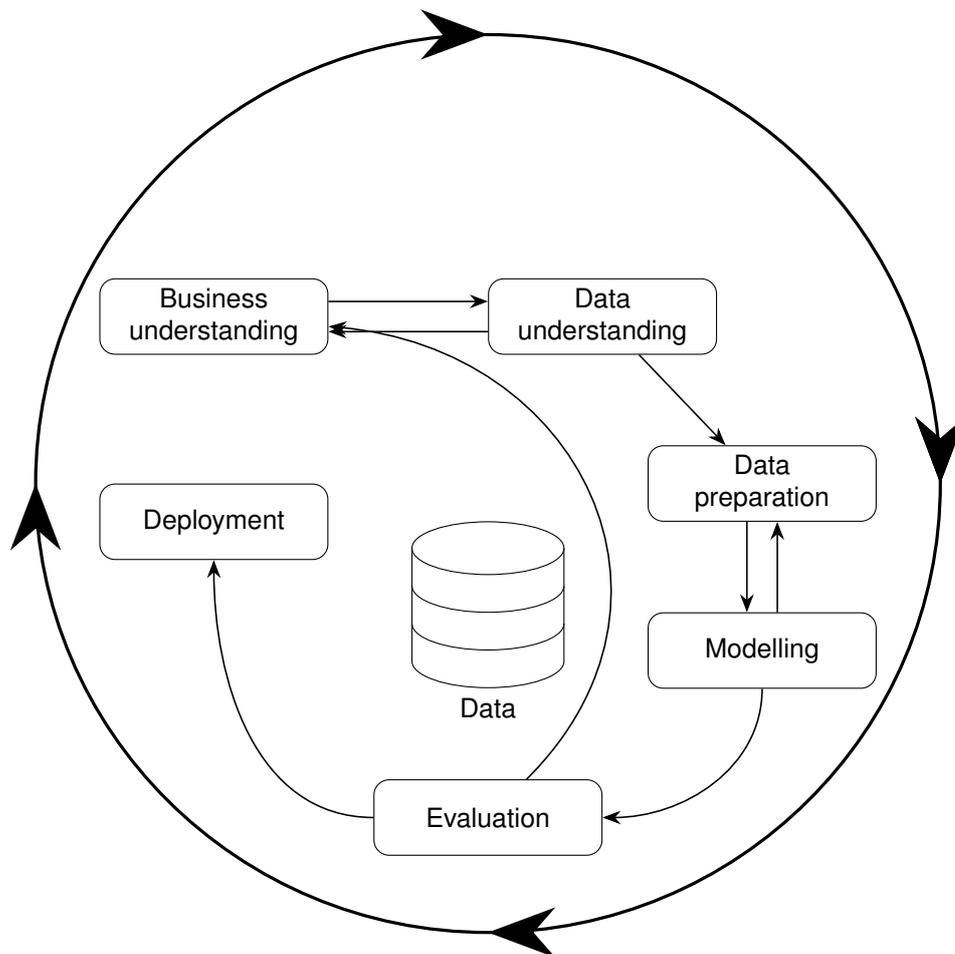


Abbildung 3.2 – Cross Industry Standard Process für Data Mining nach [32]

Die erste Phase des Prozesses beschreibt das *Business Understanding* und dient allgemein der Aufgabendefinition. Dabei wird die Relevanz der zu bearbeitenden Thematik herausgestellt und Ziele festgelegt. Notwendige Ressourcen werden dargelegt, um das übergeordnete Ziel zu erreichen. Dabei werden mögliche Risiken erörtert und die Wirtschaftlichkeit z.B. durch eine Kosten-Nutzen-Analyse überprüft. Zusätzlich werden Ziele des Data Mining festgesetzt, die quantifizierbar sind, wie die Genauigkeit (engl. accuracy).

---

Im zweiten Schritt, dem *Data Understanding*, werden initiale Datenbestände gesammelt und beschrieben. Dabei wird herausgestellt in welchem Format diese Daten vorliegen, die Datenquantität, um die Voraussetzungen für die Zielsetzung erfüllt sind. Neben der Datenvisualisierung ist ebenfalls die Sicherstellung der Datenqualität sowie -vollständigkeit relevant. Dabei wird überprüft ob die Daten repräsentativ für die Zielsetzung sind.

Phase drei beschreibt die Datenaufbereitung (engl.: *Data Preparation*). In diesem Schritt wird der Datenbestand für die Modellierungsphase vorbereitet:

- Auswahl der Daten
- Datensatzbereinigung, -ergänzungen, -qualität
- Datensatzkonstruktion, abgeleitete Merkmale, Datentransformation
- Datensatzintegration (Kombination geeigneter Informationen)

In Phase vier, *Modeling*, der Modellierungsphase werden mögliche Modellierungstechniken entsprechend der Data-Mining Zielsetzung. Da jede Modellierungstechnik bestimmte Anforderungen stellt, besteht nach Bedarf die Möglichkeit Änderungen am Datensatz vorzunehmen (Phase drei). Die Generierung des Testdesigns erfordert beispielsweise die Aufteilung der Daten in Trainings-, Test- und Validierungsdatensätze. Danach erfolgt der Aufbau des Modells sowie die Modellbewertung.

Phase fünf dient der Evaluation der Ergebnisse. Dabei werden relevante Evaluationsmetriken herangezogen. Zusätzlich wird überprüft, ob das Modell richtig aufgebaut wurde und z. B. die abgeleiteten Merkmale auch für nachfolgende Projekte verfügbar sind. Wenn keine weiteren Iterationen notwendig sind, kann in die die letzte Phase übergegangen werden. Phase sechs, die Integration (engl. Deployment), werden Integrationschritte geplant, Ergebnisse organisiert und Erfolge sowie Fehler beschrieben, die für die Verbesserung nachfolgender Projekte relevant sind.



---

## 4 Durchführung ausgewählter Methoden zur Fehlerdetektion und -klassifikation

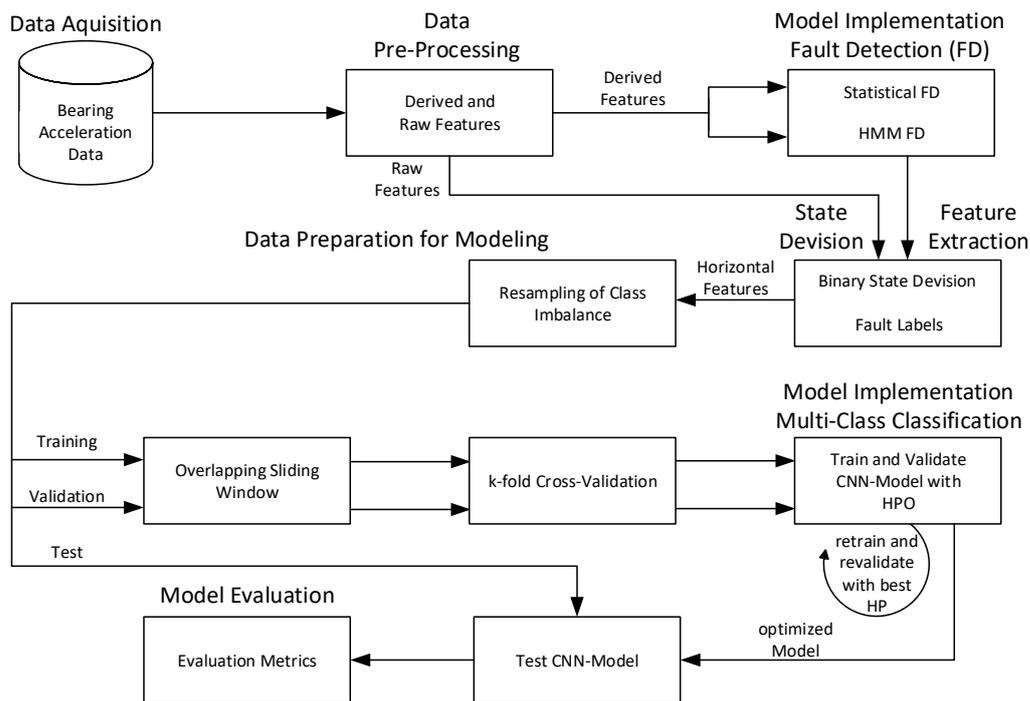
Für die Umsetzung der ausgewählten Methoden zur Fehlerdetektion und -diagnose orientiert sich dieses Implementierungskapitel am zuvor abgeleiteten Prozess- und Vorgehensmodells (vgl. Kapitel 3.4), welches im Nachfolgenden in Abbildung 4.1 dargestellt ist. Die Einordnung der Daten, die Zielsetzung sowie die Beschaffung der Daten und Informationen (i.e. Datenakquise) beschreiben den ersten Schritt des Vorgehensmodells. Dies ist für die nachfolgende Begründung der Methodenauswahl und die entsprechende Herleitung relevanter mathematischer Zusammenhänge relevant. Mit der Datenvorverarbeitung erfolgt die erste Sichtung und Analyse der Datenstrukturen und die Berechnung der abgeleiteten Merkmale. Zur Bestimmung des beginnenden Fehlerauftritts werden zwei Modelle implementiert und die Ergebnisse miteinander verglichen. Die binäre Zustandsdivision erfolgt mit den Rohdaten auf Basis des Fehlerauftrittszeitpunkts. Darauffolgend werden auf Basis der zuvor analysierten Ergebnisse der Fehlerdetektion geeignete Merkmale zur weiteren Verarbeitung extrahiert. Die Datenaufbereitung beinhaltet Resampling-Techniken (Resampling, Overlapping Sliding Window und Kreuzvalidierung), bevor die Einteilung in Trainings-, Validierungs- und Testdatensätze erfolgt. Zur Multi-Klassen Klassifikation wird ein CNN-Modell mit Hyperparameter Optimierungsstrategie trainiert sowie validiert und mit noch ungesehenen Testdaten im letzten Schritt evaluiert.

### 4.1 Datenakquise

Lager werden in einer Vielzahl von industriellen Anwendungen eingesetzt. Die Früherkennung von Fehlern spielt eine wichtige Rolle, da der Ausfall von rotierenden Teilen mit diesen Maschinenelementen zusammenhängt und somit die Verfügbarkeit und die Sicherheit beeinflusst. Lager haben die Aufgabe relativ zueinander bewegliche, insbesondere rotierende Teile in Maschinen und Geräten abzustützen und zu führen, die wirkenden äußeren Kräfte aufzunehmen und auf Gehäuse oder ähnliche Bauteile zu übertragen. Die zugehörige Baugruppe wird Lagerung genannt [33]. Wälzlager bestehen grundsätzlich aus vier Hauptkomponenten: dem Innenring, den Wälzkörpern, dem Käfig zur Fixierung der Wälzkörper und Außenring. Die Form und die Anzahl der Wälzkörper variieren nach Anwendungs- bzw. Belastungsfall [33].

#### 4.1.1 Systemdarstellung und Betriebsbedingungen

Der nachfolgend dargestellte Prüfstand (vgl. Abbildung 4.2) ist für beschleunigte Degradationstests an Wälzlagern unter verschiedenen Betriebsbedingungen (d.h. unterschied-



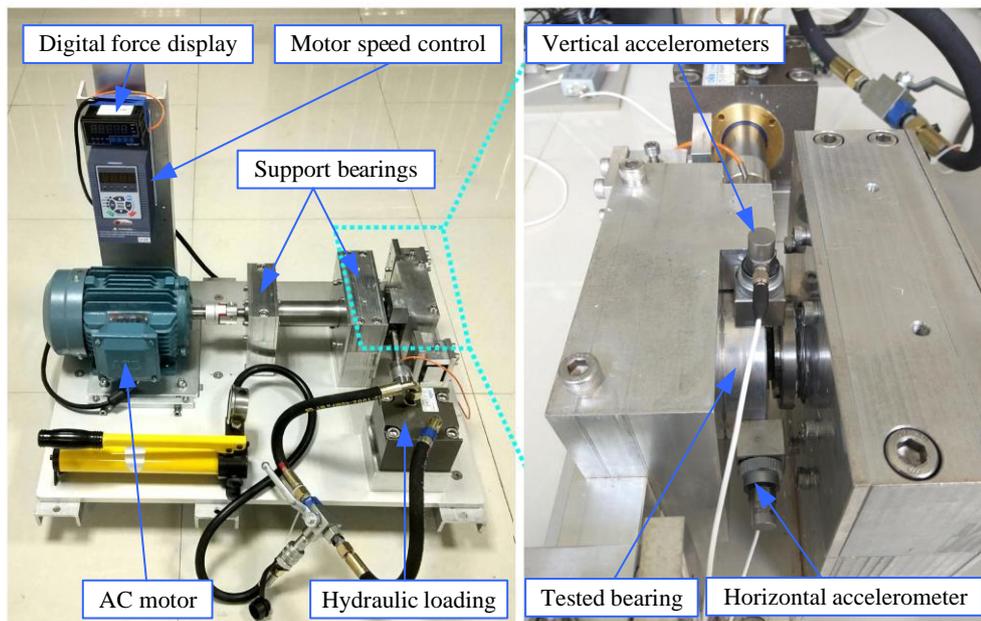
**Abbildung 4.1** – Prozess- und Vorgehensmodell zur Durchführung der Fehlerdetektion und -diagnose

liche Radialkräfte und Drehzahlen) ausgelegt und besteht aus einem Wechselstrom-Induktionsmotor, einem Motor-Drehzahlregler, einer Tragwelle, zwei Lager (Schwerlastrollenlager), einem hydraulischen Belastungssystem etc. Die Radialkraft wird durch das hydraulische Belastungssystem erzeugt und auf das Gehäuse der geprüften Lager ausgeübt, die Drehgeschwindigkeit wird durch den Drehzahlregler des AC-Induktionsmotors eingestellt und gehalten. Zwei Beschleunigungssensoren messen die vertikalen und horizontalen Vibrationssignale [34].

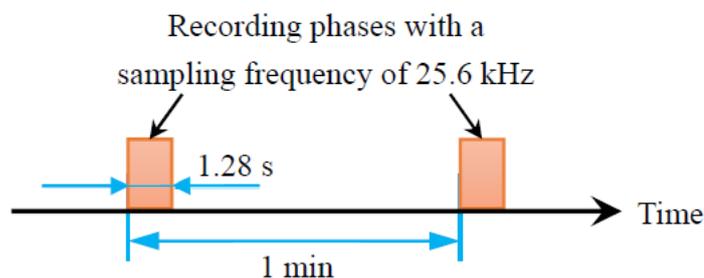
Die Experimente zur beschleunigten Degradation wurden insgesamt unter drei verschiedenen Betriebsbedingungen durchgeführt, unter welchen jeweils fünf Lager getestet wurden. Die Betriebsbedingungen sind:

1. 2100 U/min (35 Hz) und 12 kN;
2. 2250 U/min (37,5 Hz) und 11 kN;
3. 2400 U/min (40 Hz) und 10 kN.

Die Abtastrate bezeichnet die Aufnahmedauer und die gesamte Aufnahmephase des Versuchs. Zwischen dem Startzeitpunkt der jeweiligen Aufnahmephase liegt eine Minute, die Aufnahmedauer bei 1,28 Sekunden. Daraus ergibt sich eine Aufnahmefrequenz von 25,6 kHz und 32.768 Datenpunkte (vgl. Abbildung 4.3). Um den vollständigen Degradationsprozess der Lager zu beobachten, d.h. die geprüften Lager degradieren von



**Abbildung 4.2** – Aufbau des Prüfstandes für die Wälzlager. Nach [34]



**Abbildung 4.3** – Versuchsspezifische Aufnahmephasen zur Datengenerierung mit einer Aufnahme­frequenz von 27,6 kHz. Nach [34]

einem normalen Zustand bis zum Ausfall, wurde jeder der beschleunigte Degradations­tests durchgeführt, bis die maximale Amplitude der horizontalen und vertikalen Vibrationssignale  $10 \times A_h$  überstieg, wobei  $A_h$  die maximale Amplitude der horizontalen und vertikalen Vibrationssignale in der normalen Betriebsphase ist [34].

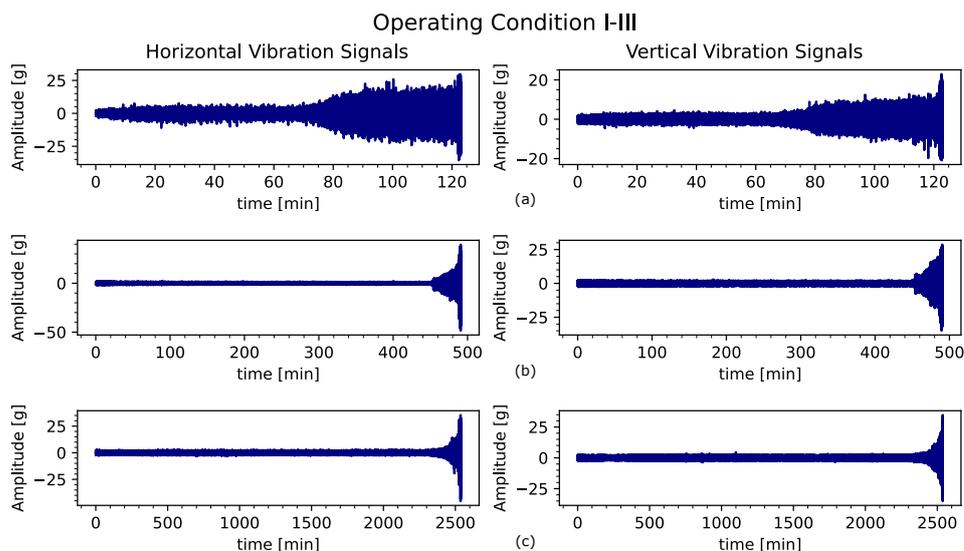
#### 4.1.2 Datenstruktur und Visualisierung

Die experimentell erhobenen Daten liegen im csv-Format vor, pro Aufnahmeperiode wurde eine csv-Datei mit jeweils 32.768 Datenpunkten exportiert. Für die weitere Daten­verarbeitung ist die Verknüpfung (engl. concatenation) der Datenpunkte je Lager erforderlich. Dabei wird ebenfalls die Vollständigkeit der Daten pro Lagerdatensatz überprüft,

**Tabelle 4.1** – Operating Conditions and Bearing Data

Operating Condition	Bearing Dataset	Number of Files	Bearing Lifetime	Fault Element
Condition 1 (35 Hz/12 kN)	Bearing 1_1	123	2 h 3 min	Outer race
	Bearing 1_2	161	2 h 41 min	Outer race
	Bearing 1_3	158	2 h 38 min	Outer race
	Bearing 1_4	122	2 h 32 min	Outer race
	Bearing 1_5	52	52 min	Inner race, outer race
Condition 2 (37.5 Hz/11 kN)	Bearing 2_1	491	8 h 11 min	Inner race
	Bearing 2_2	161	2 h 41 min	Outer race
	Bearing 2_3	533	8 h 53 min	Cage
	Bearing 2_4	42	42 min	Outer race
	Bearing 2_5	339	5 h 39 min	Outer race
Condition 3 (40 Hz/10 kN)	Bearing 3_1	2538	42 h 18 min	Outer race
	Bearing 3_2	2496	41 h 36 min	Inner race, ball, cage, outer race
	Bearing 3_3	371	6 h 11 min	Outer race
	Bearing 3_4	1515	25 h 15 min	Inner race
	Bearing 3_5	114	1 h 54 min	Outer race

indem die Ausfallzeitpunkte mit der Anzahl an insgesamt vorliegenden Datenpunkten übereinstimmt. Verdeutlicht wird dies bei der nachfolgenden Visualisierung des Degradationsverhaltens. Abbildung 4.4 stellt das Degradationsverhalten von drei exemplarisch ausgewählten Wälzlagern aus den Betriebsbedingungen I-III dar.



**Abbildung 4.4** – Exemplarische Darstellung des Degradationsverhaltens drei getesteter Wälzlager: (a) Lager 1-1, (b) Lager 2-1, (c) Lager 3-1

---

## 4.2 Auswahl geeigneter Methoden zur Fehlerdetektion und Multi-Klassen-Klassifikation

Im Hinblick auf die ausgewählten Degradationsdaten ergeben sich nachfolgend dargestellte Methoden zur Bestimmung des Fehlerauftrittszeitpunkts, sowie die Multi-Klassen-Klassifikation basierend auf den ausgefallenen Komponenten beschrieben in Tabelle 4.1.

### 4.2.1 Datenvorverarbeitung

Im Hinblick auf das vibrationsbasierte Degradationsverhalten von Lagern (vgl. Abbildung 4.4) wird der RMS-Wert als beschreibender Parameter herangezogen. Der RMS-Wert wird auch als effektive Größe eines Signals beschrieben. Aufgrund der hohen Anzahl an Datenpunkten, werden die Datensätze fragmentiert, d.h. jedes Fragment entspricht dabei 3.75 Sekunden i.e. 2048 Datenpunkte pro Fragment:

$$f = \frac{30768}{60} * 3.75 = 2048 \quad (1)$$

Der Wert wurde so gewählt, dass die Anschaulichkeit gewährleistet bleibt und die verfügbaren rechner-spezifischen Ressourcen berücksichtigt werden. Zur Berechnung des RMS-Werts für Fragmente der Zeitreihe wird das folgende Verfahren angewendet:

1. **Fragmentierung der Datenreihe:** Die Zeitreihe  $T = \{x_1, x_2, \dots, x_n\}$  wird in Fragmente der Länge  $f$  unterteilt.
2. **Quadratbildung:** Für jedes Fragment  $F_j = \{x_{j,1}, x_{j,2}, \dots, x_{j,f}\}$  werden die Werte quadriert.
3. **Mittelwertbildung:** Der Mittelwert der quadrierten Werte für jedes Fragment  $F_j$  wird berechnet:

$$\mu_j = \frac{1}{f} \sum_{i=1}^f x_{j,i}^2 \quad (2)$$

4. **Quadratwurzel:** Schließlich wird die Quadratwurzel dieses Mittelwerts gezogen, um den RMS-Wert für das Fragment zu erhalten:

$$\text{RMS}_j = \sqrt{\mu_j} = \sqrt{\frac{1}{f} \sum_{i=1}^f x_{j,i}^2} \quad (3)$$

Dieses Verfahren wird für alle Fragmente  $j$  der Zeitreihe angewendet, um die RMS-Werte  $\text{RMS}_j$  für jedes Fragment zu berechnen. Da die RMS Werte mit einer kleinen Fragmentgröße zu stark verrauscht sind und der Trend nicht analysiert werden kann, wird das Glättungsverfahren LOWESS angewendet. LOWESS (Locally Weighted Scatterplot Smoothing) ist eine nichtparametrische Methode zur Schätzung von Regressionsfunktio-

nen in einem gleitenden Fenster. Diese Methode ist besonders nützlich für das Glätten von Daten, ohne dass eine spezielle Form der Regressionsfunktion (z. B. linear oder quadratisch) vorgegeben werden muss. Die Grundidee von LOWESS besteht darin, die Beziehung zwischen einer abhängigen Variable  $y$  und einer unabhängigen Variable  $x$  durch eine Reihe lokaler Regressionsschätzungen zu modellieren. Für jeden Punkt  $x_i$  wird ein lokales Regressionsmodell basierend auf den nahe gelegenen Punkten geschätzt. Die nahe gelegenen Punkte werden gewichtet, wobei Punkte, die näher an  $x_i$  liegen, ein höheres Gewicht erhalten. Anwendungsspezifisch hängt die Stärke der Glättung mit dem Glättungsparameter zusammen.

Zur Evaluation des RMS-Parameters werden folgende Größen bestimmt: Trendfähigkeit Monotonie Robustheit

#### 4.2.2 Fehlerdetektion - modellbasierter Ansatz

Aus der Literatur (wie in [35]) geht hervor, dass das vibrationsbasierte Degradationssignal von Lagern mit der Betriebszeit ansteigt. Wenn die Amplitude einen vordefinierten Schwellenwert erreicht, gilt es nach Industriestandard als ausgefallen vgl. auch 4.1.1. Nachfolgende Abbildung 4.5 zeigt das vibrationsbasierte Degradationsverhalten, wobei das Signal dem RMS entspricht.

Phase I beschreibt das Normalverhalten. Dabei ist die Signalfunktion relativ konstant. Phase II beschreibt das degradierte Betriebsverhalten, wobei die Funktion exponentiell steigt (incipient fault behavior). Um den Übergang von Phase I zu Phase II festzustellen, wird im Hinblick auf das exponentiell wachsende Degradationssignal der Trend  $\dot{Y}(t)$  analysiert, wobei  $Y(t)$  die RMS Funktion beschreibt. Allgemein lässt sich festhalten, dass sobald ein definierter Schwellenwert überschritten wird, Degradationsverhalten festgestellt werden kann vgl. [3].

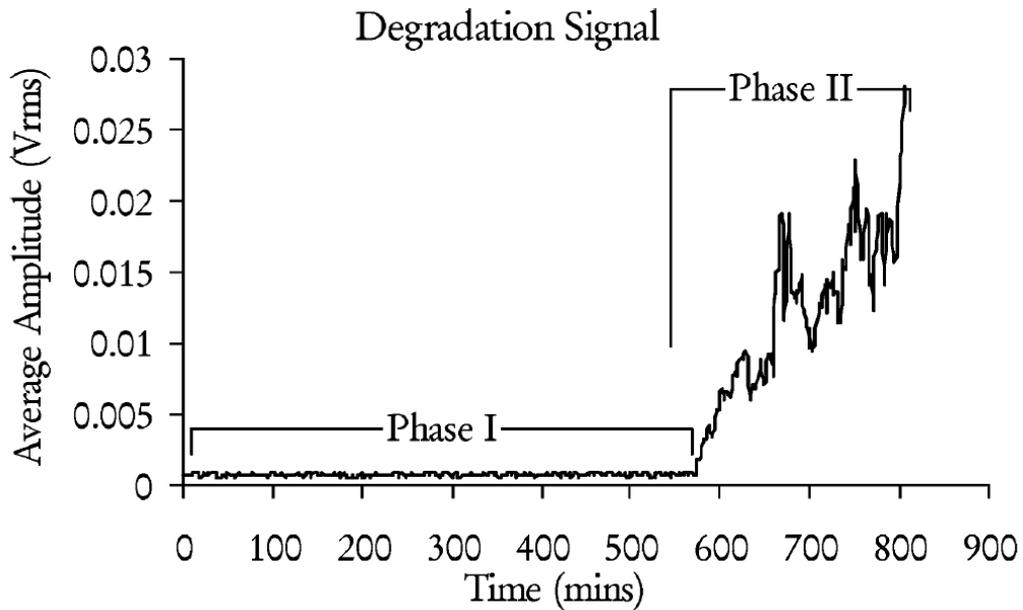
$$\dot{Y}(t) \leq \text{threshold} \rightarrow \text{normaler Betriebszustand} \quad (4)$$

$$\dot{Y}(t) > \text{threshold} \rightarrow \text{degradierender Betriebszustand} \quad (5)$$

Angenommen,  $T_i$  ist eine rationale Zahl, die in einem Intervall der Zeitreihe  $T$  liegt:

$$T_i \in \mathbb{Q} \cap (0, t_b]$$

wobei  $t_b$  eine Zeitgrenze darstellt. Die Zeitgrenze  $t_b$  wird definiert im Zusammenhang mit dem Degradationsverhalten in Phase II, d. h. die Grenze  $t_b$  muss im Bereich von Phase II liegen und entsprechend gewählt werden.



**Abbildung 4.5** – Darstellung des vibrationsbasierten Degradationsverhalten von Lagern. Grundlegend besteht das Degradationssignal aus zwei Phase: Phase I entspricht der normalen (nicht defekten) Betriebsbedingung, Phase II entspricht der fehlerhaften (degradierten oder defekten) Betriebsbedingung. Nach [35]

Definiert wird  $S$  als eine Funktion, die feststellt, ob ein bestimmter Wert  $x_i$  der Zeitreihe  $T_n$  größer ist als seine Nachbarn in Abhängigkeit des Sensitivitätsparameters  $k$ . Sei  $T = (T_1, T_2, \dots, T_n)$  die beobachtete Zeitreihe, die den Trend über die Zeit (i.e. die erste Ableitung der RMS-Funktion) darstellt. Wir definieren eine Funktion  $S(k, i, T_i, T)$ , wobei  $k \in \mathbb{N}$  der Sensitivitätsparameter ist.

1. Berechnung der Funktion  $S(k, i, T_i, T)$ :

$$S(k, i, T_i, T) = \begin{cases} T_i & \text{wenn } T_i > T_j \\ 0 & \text{sonst} \end{cases}$$

wobei  $T_i$  der Wert der Zeitreihe an der Position  $i$  ist.

Die Nachbarn von  $T_i$  sind definiert als:

$$\{T_j \mid j \in [\max(0, i - k), i - 1] \cup [i + 1, \min(i + k, n - 1)]\}$$

Dies bedeutet, dass  $T_i$  nur dann zurückgegeben wird, wenn es größer ist als alle seine Nachbarn im definierten Bereich (i.e.  $k$ ), andernfalls wird 0 zurückgegeben. Lokale poten-

tielle Spitzenwerte werden somit für das weitere Vorgehen berücksichtigt. Zur Detektion der Spitzenwerte wird folgendermaßen vorgegangen:

2. Berechnung des arithmetischen Mittelwerts und der Standardabweichung der Zeitreihe im definierten Intervall:

$$m = \frac{1}{t_b} \sum_{i=1}^{t_b} T_i \quad (6)$$

$$s = \sqrt{\frac{1}{t_b} \sum_{i=1}^{t_b} (T_i - m)^2} \quad (7)$$

Der Mittelwert und die Standardabweichung dienen als Basis zur Festlegung dynamischer Schwellenwerte (engl. dynamic thresholds). Diese Schwellenwerte helfen dabei, signifikante Spitzen von zufälligem Rauschen zu unterscheiden. Der Mittelwert bietet eine Referenzlinie, während die Standardabweichung die Variabilität der Daten um den Mittelwert misst. Dies ermöglicht es, Abweichungen zu identifizieren, die weit genug vom Mittelwert entfernt sind, um als signifikante Spitzen zu gelten.

3. Berechnung der Hilfsfunktion  $a(i)$  basierend auf  $S(k, i, T_i, T)$ :

$$a(i) = S(k, i, T_i, T)$$

4. Detektion der Spitzenwerte:

$$O = \{(i, T_i) \mid a(i) > 0 \text{ und } (a(i) - m) > h \cdot s\} \quad (8)$$

Die lokalen Spitzenwerte  $a(i)$  müssen also größer als der arithmetische Mittelwert und die  $h$ -fache Streuung sein. Ein Spitzenwert  $a(i)$  wird als signifikant betrachtet, wenn  $h \in \{2, 3\}$  vgl. [36].

5. Filtern der Spitzenwerte, sodass nur ein Spitzenwert pro  $k$ -Nachbarschaft beibehalten wird. Nachfolgende Gleichung stammt aus [37]:

$$O_{\text{filtered}} = \{O_i \mid \forall O_i \in O, |O_i[0] - O_{i-1}[0]| > k\} \quad (9)$$

### 4.2.3 Datenbasierter Ansatz

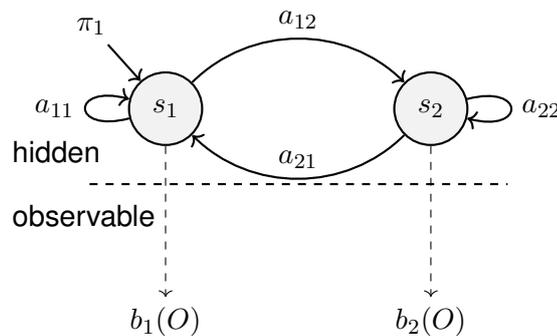
Ein HMM besteht aus zwei Hauptkomponenten: den versteckten Zuständen und den beobachtbaren Ereignissen. Die Übergänge zwischen den versteckten Zuständen folgen einer Markov-Kette, bei der die Wahrscheinlichkeit eines Übergangs nur vom aktuellen Zustand abhängt und nicht von vorherigen Zuständen. Die beobachtbaren Ereignisse hängen von den versteckten Zuständen ab und werden durch Emissionswahrscheinlichkeiten beschrieben. Wie eingehend beschrieben, kann das betrachtete System zwei Zustände einnehmen, entweder es liegt ein normaler oder ein fehlerhafter Betriebszustand

---

vor siehe Abbildung 4.6.

Ein HMM besteht aus folgenden Komponenten:

- **Zustände:** Eine endliche Menge von Zuständen, wobei jeder Zustand eine bestimmte Bedeutung im Kontext des Modells hat.
- **Übergangswahrscheinlichkeiten:** Die Wahrscheinlichkeiten, von einem Zustand in einen anderen zu wechseln. Diese sind in einer Übergangsmatrix  $\mathbf{A} = \{a_{ij}\}$  organisiert, wobei  $a_{ij}$  die Wahrscheinlichkeit darstellt, vom Zustand  $i$  in den Zustand  $j$  zu wechseln.
- **Emissionswahrscheinlichkeiten:** Die Wahrscheinlichkeiten, dass ein bestimmter beobachtbarer Wert (Emission) von einem bestimmten Zustand aus generiert wird. Diese werden durch eine Emissionsmatrix  $\mathbf{B}$  beschrieben.
- **Anfangswahrscheinlichkeiten:** Die Wahrscheinlichkeiten, dass das System in einem bestimmten Anfangszustand beginnt, dargestellt durch einen Vektor  $\pi$ .



**Abbildung 4.6** – Struktureller Aufbau des Hidden Markov Modells

Für den Anwendungsfall ergeben sich zwei Zustände mit  $N = 2$ , daraus resultiert die Zustandsübergangsmatrix:

$$\mathbf{A} = [a_{ij}] = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{mit } a_{ij} = P(S_{t+1} = j \mid S_t = i) \quad (10)$$

Dabei bezeichnet  $S_t$  den Zustand zum Zeitpunkt  $t$ . Die Elemente von  $A$  müssen den Bedingungen der Wahrscheinlichkeit genügen.

Die Anfangswahrscheinlichkeiten werden durch folgenden Vektor repräsentiert:

$$\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} \quad \text{mit } \pi_i = P(S_1 = i) \quad (11)$$

### Emissionswahrscheinlichkeiten

Die Matrix der Beobachtungswahrscheinlichkeiten  $B$  gibt an, wie wahrscheinlich es ist,

eine bestimmte Beobachtung in einem gegebenen Zustand zu erhalten:

$$\mathbf{B} = [b_{ij}] \text{ mit } b_{ij} = P(O_t = j \mid S_t = i) \quad (12)$$

Für kontinuierliche Beobachtungen werden die Emissionswahrscheinlichkeiten durch eine multivariate Normalverteilung modelliert:

$$B_i(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right) \quad (13)$$

wobei  $\mu_i$  der Mittelwert und  $\Sigma_i$  die Kovarianzmatrix des Zustands  $i$  sind.

### Berechnung der posterioren Wahrscheinlichkeiten

Die posterioren Wahrscheinlichkeiten  $\gamma_t(i)$  geben die Wahrscheinlichkeit an, dass sich das System zum Zeitpunkt  $t$  im Zustand  $i$  befindet, gegeben die gesamte Beobachtungssequenz  $X$  und das Modell  $\lambda$ :

$$\gamma_t(i) = P(q_t = S_i \mid X, \lambda) \quad (14)$$

Diese Wahrscheinlichkeiten können mithilfe des Vorwärts-Rückwärts-Algorithmus (Forward-Backward Algorithm) [38] berechnet werden:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(X \mid \lambda)} \quad (15)$$

wobei  $\alpha_t(i)$  die Vorwärts-Wahrscheinlichkeit (forward probability) und  $\beta_t(i)$  die Rückwärts-Wahrscheinlichkeit (backward probability) ist.

### Übergangswahrscheinlichkeiten aus der Zustandssequenz

Sei  $\{s_i\}$  die Zustandssequenz der Länge  $n$ . Die Differenzen der Zustandssequenz sind definiert als:

$$\Delta s_i = s_{i+1} - s_i \quad \text{für } i = 1, \dots, n-1 \quad (16)$$

Die Anzahl der Zustandswechsel  $N_t$  ist gegeben durch:

$$N_t = \sum_{i=1}^{n-1} \mathbf{1}(\Delta s_i \neq 0) \quad (17)$$

wobei  $\mathbf{1}(\cdot)$  die Indikatorfunktion ist, die 1 ist, wenn die Bedingung erfüllt ist, und 0 ansonsten.

---

Die Übergangswahrscheinlichkeit  $P_t$  wird dann berechnet als:

$$P_t = \frac{N_t}{n} \quad (18)$$

### Posterior Probabilities im Hidden Markov Model

Die posterioren Wahrscheinlichkeiten  $\gamma_t(i)$  geben die Wahrscheinlichkeit an, dass sich das System zum Zeitpunkt  $t$  im Zustand  $i$  befindet, gegeben die gesamte Beobachtungssequenz  $X$  und das Modell  $\lambda$ :

$$\gamma_t(i) = P(q_t = S_i \mid X, \lambda) \quad (19)$$

Diese Wahrscheinlichkeiten können mithilfe des Vorwärts-Rückwärts-Algorithmus (Forward-Backward Algorithm) auch Baum-Welch-Algorithmus berechnet werden. Die Berechnung erfolgt nach [38] wie folgt:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(X \mid \lambda)} \quad (20)$$

wobei  $\alpha_t(i)$  die Vorwärts-Wahrscheinlichkeit (forward probability) ist und  $\beta_t(i)$  die Rückwärts-Wahrscheinlichkeit (backward probability).

- $\alpha_t(i)$  ist die Wahrscheinlichkeit, die Beobachtungssequenz von Zeit 1 bis  $t$  zu sehen und sich im Zustand  $i$  zum Zeitpunkt  $t$  zu befinden.
- $\beta_t(i)$  ist die Wahrscheinlichkeit, die Beobachtungssequenz von Zeit  $t + 1$  bis zum Ende zu sehen, gegeben dass sich das System im Zustand  $i$  zum Zeitpunkt  $t$  befindet.

Die Vorwärts-Wahrscheinlichkeit  $\alpha_t(i)$  wird rekursiv berechnet:

$$\alpha_{t+1}(j) = \left( \sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(\mathbf{x}_{t+1}) \quad (21)$$

wobei  $a_{ij}$  die Übergangswahrscheinlichkeit vom Zustand  $i$  zu Zustand  $j$  und  $b_j(\mathbf{x}_{t+1})$  die Emissionswahrscheinlichkeit im Zustand  $j$  für die Beobachtung  $\mathbf{x}_{t+1}$  ist.

Die Rückwärts-Wahrscheinlichkeit  $\beta_t(i)$  wird rekursiv berechnet:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j) \quad (22)$$

#### 4.2.4 Diagnose

Wie aus der eingehenden Datenanalyse hervorgeht, ist eine Multi-Klassen-Klassifikation zur Bestimmung der ausgefallenen Bauteilen erforderlich. Jede Betriebsbedingung wird separat betrachtet. Aus der Dokumentation des Degradationsexperiments geht hervor, dass je Betriebsbedingung vier Fehlerklassen (bzw. Fault Location, Lokalisation des Ausfalls) auftreten können. Im Hinblick auf die Performance, die Modellgenauigkeit und die Optimierungsmöglichkeiten wird für die Klassifizierungsaufgabe ein neuronales Netz, i.e. ein CNN, verwendet. Faltungsnetze, auch bekannt als neuronale Faltungsnetze oder CNNs, sind eine spezielle Art von neuronalen Netzen für die Verarbeitung von Daten, die eine bekannte gitterartige Topologie aufweisen (beispielsweise eindimensionale Faltungsschichten für Zeitreihendaten, oder zweidimensionale Schichten für Bilder). Nachfolgende Abbildung 4.22 stellt den strukturellen Aufbau eines Faltungsnetzes.

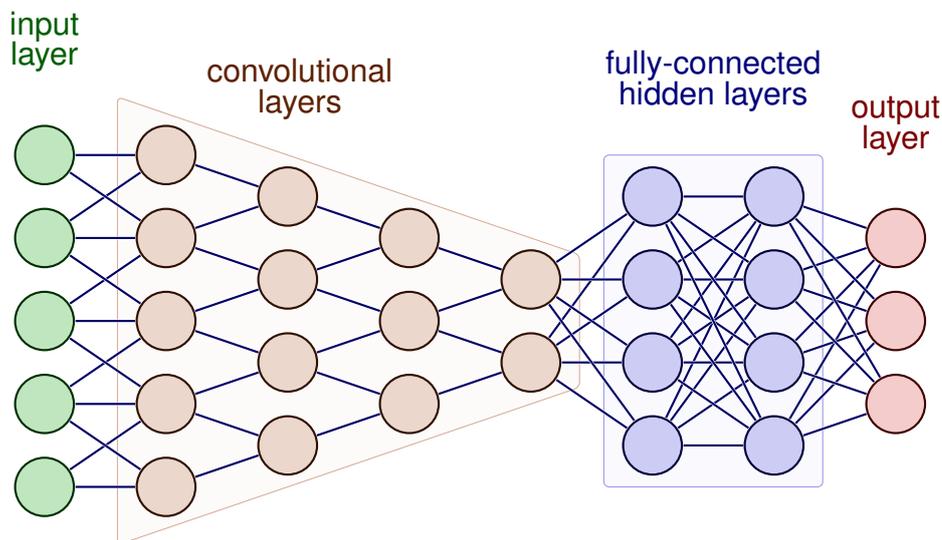


Abbildung 4.7 – Struktureller Aufbau eines CNN

##### Netzwerkarchitektur

Das implementierte neuronale Netz besteht aus mehreren Schichten, von denen jede zur Gesamtfunktionalität des Modells beiträgt. Das Modell wurde mithilfe der Keras Sequential API erstellt, die eine lineare Stapelung von Schichten ermöglicht.

##### Input Layer

Die Eingabeschicht nimmt Daten mit einer bestimmten Form an:

$$\text{Input} \in \mathbb{R}^{(T,d)} \quad (23)$$

wobei  $T$  die Fenstergröße und  $d = 1$  die Merkmalsdimension ist.

##### Faltungsschichten

---

Das Modell besteht aus drei 1D Faltungsschichten, die wie folgt definiert sind:

$$\text{Conv1D}(f, k) \quad \text{mit Aktivierungsfunktion ReLU} \quad (24)$$

wobei  $f$  die Anzahl der Filter und  $k$  die Kernelgröße ist. Üblicherweise wird die ReLU (rectified linear unit) Aktivierungsfunktion angewendet, da sie im Gegensatz zu anderen Aktivierungsfunktionen in der Regel bessere Genauigkeiten und Trainingszeit vorweist siehe [39].

### **Batch Normalization und Max Pooling**

Auf jede Faltungsschicht folgt eine Batch-Normalisierung und ein Max-Pooling Layer. Batch Normalisierungsschichten nach jeder Faltungsschicht ist für ein stabiles Training und eine schnelle Konvergenz hilfreich. Pooling Layer haben das Ziel, die Dimensionalität der Anzahl der Darstellungen zu und die Anzahl der Parameter um die rechner-spezifische Komplexität zu reduzieren [40]. Üblicherweise wird die Größe des Max Pooling Layers auf zwei gesetzt vgl. [40].

### **Vollständig verbundene Schichten**

Nach der Abflachung der Ausgabe der Faltungsschichten enthält das Modell eine vollständig verbundene (dichte) Schicht:

$$\text{Dense}(h) \quad \text{mit Aktivierungsfunktion ReLU}$$

wobei  $h$  die Anzahl der Einheiten in der dichten Schicht ist.

### **Output Layer**

Die Ausgabeschicht ist eine dichte Schicht mit einer Softmax-Aktivierungsfunktion, die die Wahrscheinlichkeitsverteilung über die Ausgabeklassen darstellt:

$$\text{Output}(x) = \text{Softmax}(Wx + b) \quad (25)$$

wobei  $W$  und  $b$  die Gewichte bzw. Verzerrungen sind. Die Softmax-Aktivierungsfunktion wird üblicherweise bei Multi-Klassen-Klassifikationsaufgaben verwendet [39].

### **Regularisierung**

Das Modell wendet  $L_2$  Regularisierung auf die Gewichte an, um eine Überanpassung zu verhindern:

$$\mathcal{L}_2(w) = \lambda \sum_i w_i^2 \quad (26)$$

wobei  $\lambda$  der Regularisierungsparameter ist.

### **Modellkompilierung**

Das Modell wird mit Hilfe des Adam-Optimierers mit einer während der Hyperparameter-suche abgestimmten Lernrate erstellt:

$$\text{Adam}(lr) \quad \text{mit Verlustfunktion Categorical Crossentropy}$$

Der Adam Optimizer reduziert die Lernrate automatisch und konvergiert relativ schnell. Experimentell wurde auch der SGD Optimizer vergleichsweise angewendet, dieser erzielte allerdings während der HPO deutlich schlechtere Genauigkeiten. Der kategoriale Kreuzentropieverlust ist definiert als:

$$\mathcal{L} = - \sum_i y_i \log(\hat{y}_i) \quad (27)$$

wobei  $y_i$  das wahre Label und  $\hat{y}_i$  die vorhergesagte Wahrscheinlichkeit ist. Der Kreuzentropieverlust gehört zu der gängigsten Auswahl in Kombination mit dem Softmax Layer am Ende des CNN [39].

#### **Prozess der Kreuzvalidierung**

Das Modell wird mithilfe der k-fachen Kreuzvalidierung trainiert und validiert, einer robusten Methode zur Bewertung der Generalisierungsfähigkeit des Modells.

#### **Sliding Window**

Zunächst werden aus den Zeitreihendaten gleitende Fenster erzeugt:

$$\text{sliding\_window}(X, \text{window\_size}, \text{overlap})$$

wobei  $X$  die Eingabedaten sind,  $\text{window\_size}$  die Größe des Fensters ist und  $\text{overlap}$  bestimmt, wie viel Überlappung zwischen aufeinanderfolgenden Fenstern besteht. Die Überlappung wird mit  $0.5 * \text{window\_size}$  festgelegt.

#### **K-Fold Kreuzvalidierung**

Die Daten werden in  $K$  Falten aufgeteilt, wobei jede Falte als Validierungssatz dient, während die verbleibenden  $K - 1$  Falten zum Training verwendet werden:

$$\text{K-Fold}(X) \rightarrow \{(X_{\text{train}}, X_{\text{val}})_i\}_{i=1}^K \quad (28)$$

wobei  $K=5$ .

#### **Hyperparameter Tuning**

Die optimalen Hyperparameter werden mit der Hyperband-Methode gesucht. Dabei wird die Anzahl der Filter, die Kernelgröße und die L2-Regularisierungsgröße bestimmt:

$$\text{Hyperband}(\text{objective} = \text{val\_accuracy}, \text{max\_epochs} = 30)$$

Diese Methode weist den vielversprechenden Hyperparameterkonfigurationen adaptiv Ressourcen zu und erforscht so effizient den Hyperparameterraum [24]. Der Hyperparameter  $\text{Epoch}$  gibt an, wie oft die gesamte Trainingsmenge in das neuronale Netz zum Training eingegeben wird. Die Größe sollte dabei so gewählt werden, dass die Lücke zwischen Trainings- und Validierungsgenauigkeit möglichst klein ist. Zu kleine Werte für zur Unteranpassung, zu große Werte resultieren in Überanpassung. Wenn es die rechner-spezifischen Ressourcen zu lassen, sollte der Wert möglichst groß gewählt werden.

---

Die Batch Size repräsentiert die Anzahl an Samples die in jedem Training zum Modell gegeben werden. Aus der Literatur geht hervor, dass die sich die Generalisierungsfähigkeit verbessert [39]. Ausgewählt wurde ein Größe die  $\frac{1}{4}$  oder  $\frac{1}{8}$  der Fenstergröße (i.e. `window_size`) entspricht.

### Callbacks

Während des Trainings und des Tests wird die Methode Early Stopping (frühzeiter Stopp) und eine Reduzierung der Lernrate eingesetzt, um eine Überanpassung zu verhindern und die Konvergenz sicherzustellen:

```
EarlyStopping(monitor = val_loss, patience = 1)
```

```
ReduceLROnPlateau(monitor = val_loss, factor = 0.5, patience = 3, min_lr = 10-6)
```

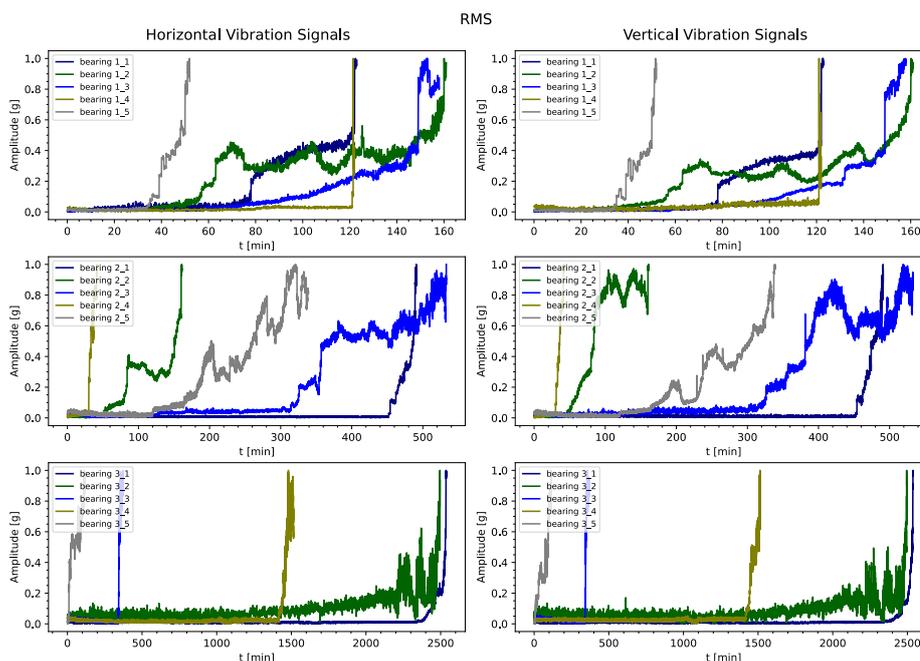
Sobald die Validierungsverlustfunktion nicht gegen Null konvergiert im Rahmen von einem Iterationsschritt (*Epoch*) wird der Berechnungsprozess gestoppt. Die Lernrate wird nach drei *Epochs* entsprechend reduziert.

## 4.3 Datenvorverarbeitung

Die Datenvorverarbeitung behinhalten die Ableitung eines geeigneten Merkmals zur darauffolgenden Fehlerdetektion. Um den Fehlerauftrittszeitpunkt zu bestimmen erfolgt die Umrechnung der Datenpunkte. Dafür wird der quadratische Mittelwert (engl. RMS: Root Mean Square) auch RMS Wert herangezogen. Um die Vergleichbarkeit zu gewährleisten werden die Daten normalisiert. Nachfolgende Abbildung 4.8 zeigt die Darstellung des abgeleiteten Merkmals (i.e. die RMS Werte) und Abbildung 4.9 das Ergebnis des Glättungsverfahrens, darauf folgt die Evaluation in Tabelle 4.3.

Die Auswahl der Glättungsparameter hängen maßgeblich von der Stärke des Rauschens ab. Kleine Werte glätten weniger stark, wobei höhere Werte im Umkehrschluss stärker glätten. Tabelle 4.2 stellt die Werte da. Für Lager1-1 wurde beispielweise ein Wert von 0.3 gewählt, da wie in den ersten zwei Plots von Abbildung 4.9 deutlich wird, eine stärkere Glättung erforderlich war.

Zur quantitativen Evaluation des gewählten Parameters (i.e. RMS Werte) werden die Berechnung der Trendfähigkeit (trendability), Monotonie (Monotonicity) und die Robustheit (robustness) herangezogen und nachfolgend in Tabelle 4.3 dargestellt.

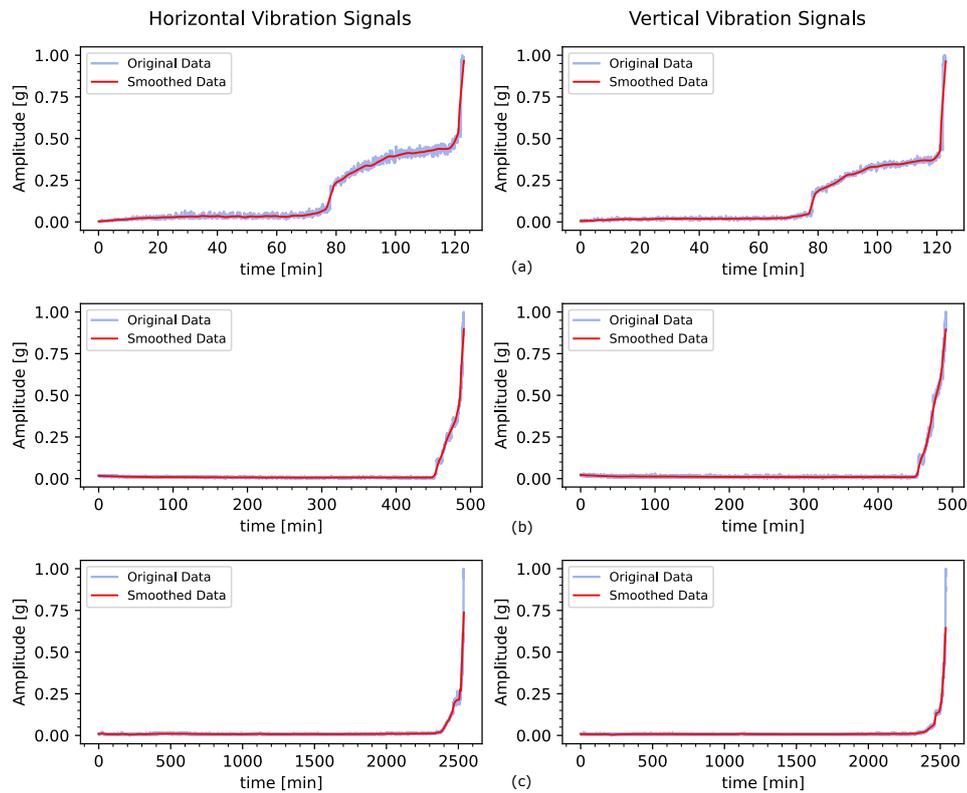


**Abbildung 4.8** – Darstellung des abgeleiteten Merkmals über die Betriebszustände I-III und jeweils der horizontalen und vertikalen Vibrationssignale

**Tabelle 4.2** – Glättungsparameter für LOWESS

Dataset	Parameter
bearing11	0.03
bearing12	0.02
bearing13	0.02
bearing14	0.01
bearing15	0.03
bearing21	0.02
bearing22	0.03
bearing23	0.01
bearing24	0.03
bearing25	0.02
bearing31	0.01
bearing32	0.04
bearing33	0.02
bearing34	0.03
bearing35	0.01

### LOWESS Smoothing of RMS Curves



**Abbildung 4.9** – LOWESS Glättung des abgeleiteten Merkmals über die Betriebszustände I-III und jeweils der horizontalen und vertikalen Vibrationssignale exemplarisch für (a) Lager 1-1, (b) Lager 2-1, (c) Lager 3-1

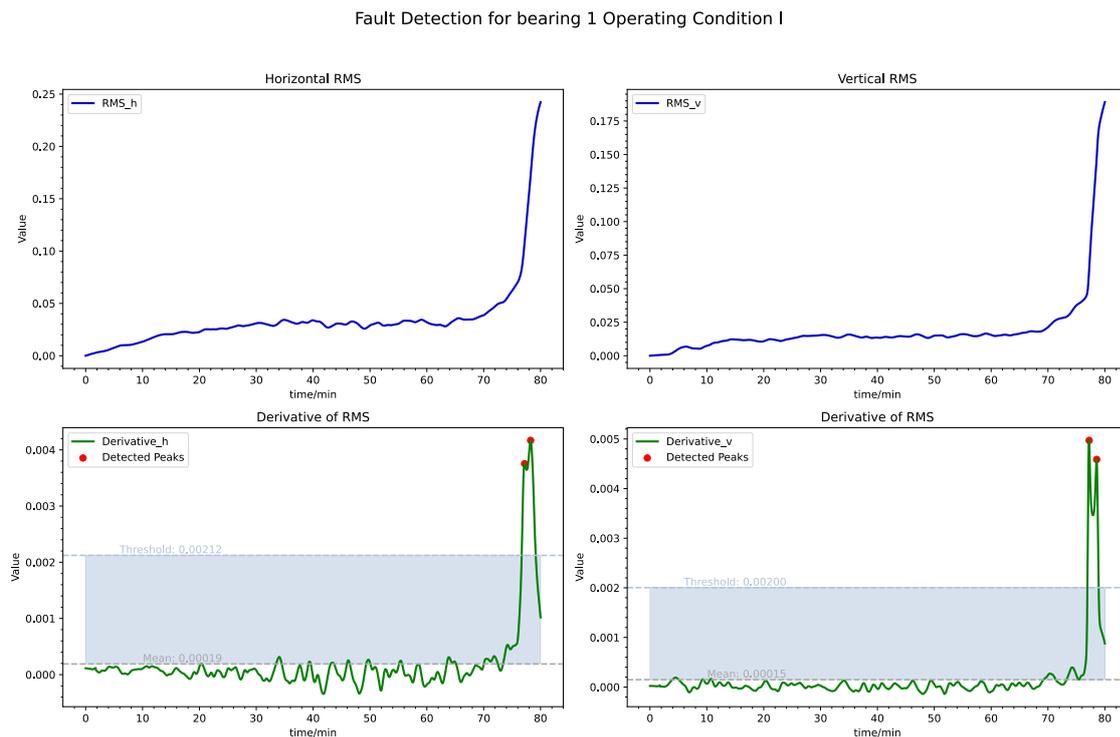
**Tabelle 4.3** – Evaluation der RMS Werte für alle Lager

Lager	Trendfähigkeit	Monotonie	Robustheit
bearing11	0.868245	0.260295	0.991057
bearing12	0.902934	0.127379	0.991819
bearing13	0.759385	0.235853	0.992490
bearing14	0.791347	0.008201	0.988549
bearing15	0.764347	0.120337	0.975583
bearing21	0.376874	0.059325	0.997064
bearing22	0.899909	0.104078	0.992950
bearing23	0.886319	0.163012	0.997998
bearing24	0.793467	0.149031	0.977010
bearing25	0.908397	0.153421	0.996569
bearing31	0.329889	0.029552	0.999109
bearing32	0.776422	0.139977	0.999624
bearing33	0.418404	0.029992	0.995853
bearing34	0.369955	0.039853	0.999209
bearing35	0.873770	0.115195	0.990390

## 4.4 Implementierung der Modelle zur Fehlerdetektion

### 4.4.1 Modellbasierte Fehlerdetektion

Nachfolgend werden exemplarisch die Ergebnisse der modellbasierten Fehlerdetektion von drei Lagern graphisch abgebildet und alle Ergebnisse in Tabelle 4.4 zusammengefasst. Im Hinblick auf die Feststellung des spezifischen Fehlerauftrittszeitpunktes, also der Zeitpunkt bei dem die Funktion von einem normalen in einen abnormalen oder fehlerhaften Zustand über geht, kann dieser explizit über eine Trendanalyse erfolgen. Aus Abbildung 4.10 geht das Ergebniss für das erste Lager der Betriebsbedingung I hervor. Aufgrund des gewählten Sensivitätsparameters  $k=2$  gehen mehrere Spitzenwerte hervor aus der Berechnung hervor. Das Lager1-1 zeigt das typische Degradationsverhalten und einen eindeutigen Übergang zwischen den zwei Phasen. Der Trend zeigt nach einem relativ konstanten Verlauf einen (oder mehrere) Maximalwerte auf und flacht danach wieder ab. Die detektierten Spitzenwerte liegen deutlich über dem Schwellenwert, was auf die Eindeutigkeit des Ergebnis hindeutet.

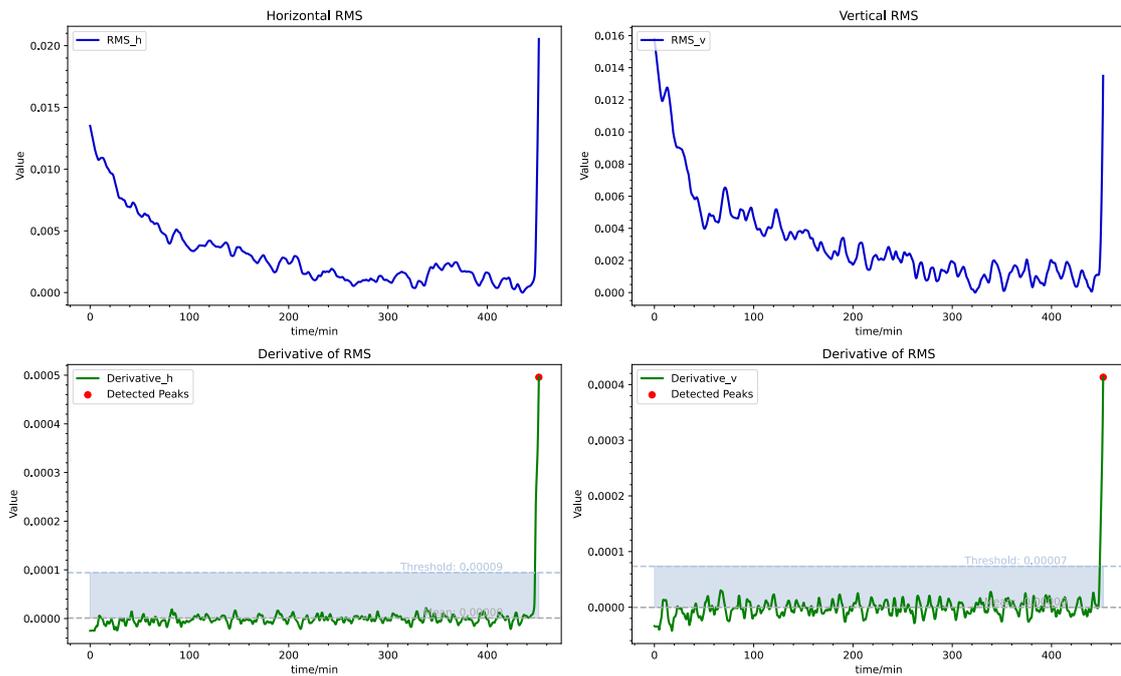


**Abbildung 4.10** – Ergebnis der Fehlerdetektion: Lager 1-1 Betriebsbedingung I

Lager2-1 zeigt ebenfalls ein sehr eindeutiges Detektionsergebnis wie Lager1-1. Obwohl das Lager zu Beginn des Experiments relativ hohe Amplitudenwerte aufweist, scheint der stetige Verlauf kaum einen Einfluss auf den Trend zu haben. Der Spitzenwert wird eindeutig detektiert.

---

### Fault Detection for bearing 1 Operating Condition II



**Abbildung 4.11** – Ergebnis der Fehlerdetektion: Lager 2-1 Betriebsbedingung II

Die Ergebnisse der Detektion von Lager3-1 sind weniger eindeutig wie bei den anderen Lagern zuvor beschrieben. Die Fluktuationen bzw. das Rauschen führt dazu, dass die meisten Werte des Trends relativ nah am Schwellenwert liegen. Die Spitzenwerte können trotzdem als eindeutig bezeichnet werden. Ein Grund für diese Beobachtung könnte der kleine Glättungsparameter sein.

#### 4.4.2 Datenbasierte Fehlerdetektion

Nachfolgend werden die Ergebnisse der Fehlerdetektion mittels HMM dargestellt, dabei werden die errechneten Parameter miteinander verglichen. Die nachfolgenden Abbildungen zeigen den jeweiligen Übergangspunkt vom ersten Zustand  $s_1$  zu  $s_2$ . In Abbildung 4.13 ist für das vertikale Signal zusehen, dass der Übergang zwischen den Zuständen nicht sofort, sondern allmählich erfolgt. Die Wahrscheinlichkeit, im vorherigen Zustand (blau) zu sein, nimmt allmählich ab, während die Wahrscheinlichkeit, im neuen Zustand (grün) zu sein, allmählich zunimmt. Dies könnte darauf hindeuten, dass der Übergang nicht abrupt, sondern über eine gewisse Zeitspanne hinweg stattfindet. Das kann beispielsweise durch Rauschen oder Unsicherheiten in den Beobachtungen verursacht werden.

Für das horizontale Signal erfolgt der Übergang zwischen den Zuständen sehr abrupt. Die Wahrscheinlichkeit, im vorherigen Zustand zu sein, fällt nahezu sofort auf Null,

## 4 Durchführung ausgewählter Methoden zur Fehlerdetektion und -klassifikation

Fault Detection for bearing 1 Operating Condition III

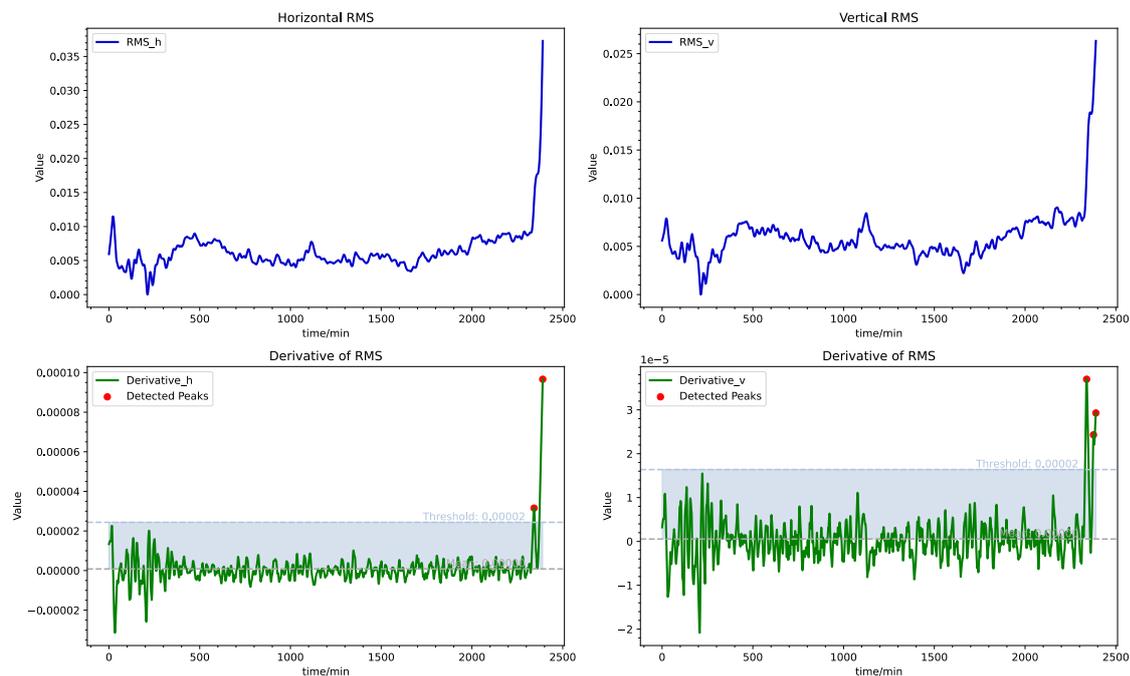


Abbildung 4.12 – Ergebnis der Fehlerdetektion: Lager 3-1 Betriebsbedingung III

Posterior Probabilities for bearing 3 Operating Condition I

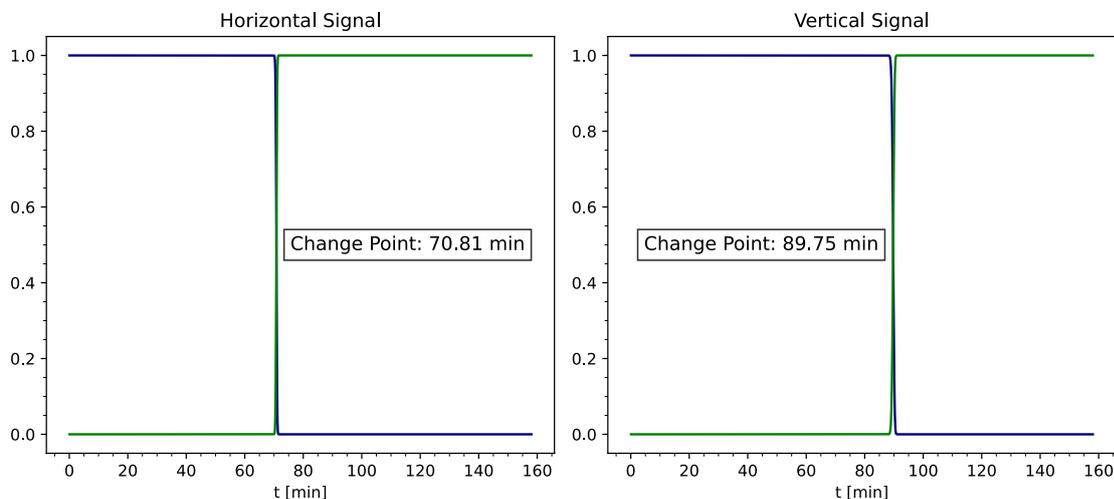
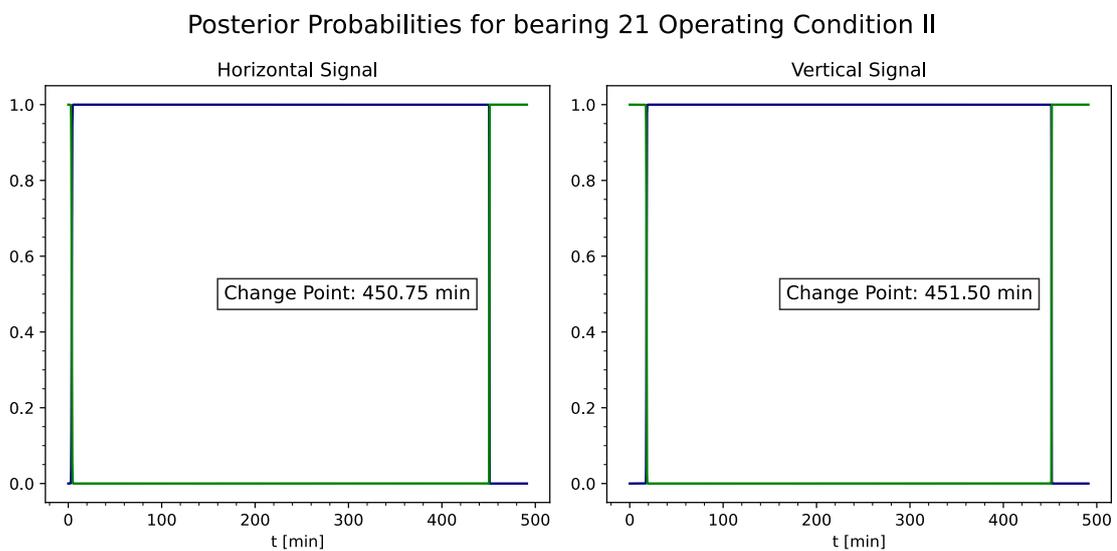


Abbildung 4.13 – Posterioren Übergangswahrscheinlichkeiten für Lager 3 Betriebsbedingung I

während die Wahrscheinlichkeit, im neuen Zustand zu sein, sehr schnell auf Eins steigt. Dies deutet darauf hin, dass der Übergang zwischen den Zuständen in diesem Fall sehr klar und ohne viel Unsicherheit erfolgt.

Der Übergang für Lager 2-1 ist nicht eindeutig bestimmbar vgl. 4.14. Bei der Bestimmung des Übergangs wird davon ausgegangen, dass das Lager nicht bereits in den

ersten Minuten ausgefallen ist. Ein Vergleich mit den Vibrationssignalen schließt dies ebenfalls aus. Tabelle 4.5 zeigt die berechneten Parameter. Dabei lässt sich feststellen, dass bei den Lagern 14, 21, 22, 32 und 34 mehr als zwei Übergangspunkte festgestellt worden sind. Ein Vergleich der Kovarianzen zeigt keine große Variabilität im Vergleich zu eindeutigen Übergangspunkten anderer Lager. Es ist davon auszugehen, dass das Modell die Übergänge erzwingt, weshalb es zu mehreren Übergängen kommen kann. Für die genannten Lager scheint es, dass die Daten nicht eindeutig genug sind, um eine konkrete Aussage zu treffen. Die geschätzten Übergangswahrscheinlichkeiten geben keinen direkten Hinweis auf die Übergangspunkte. Im Hinblick auf diese Ergebnisse erfolgt ein Vergleich der Übergangspunkte mit denen des modellbasierten Ansatzes.

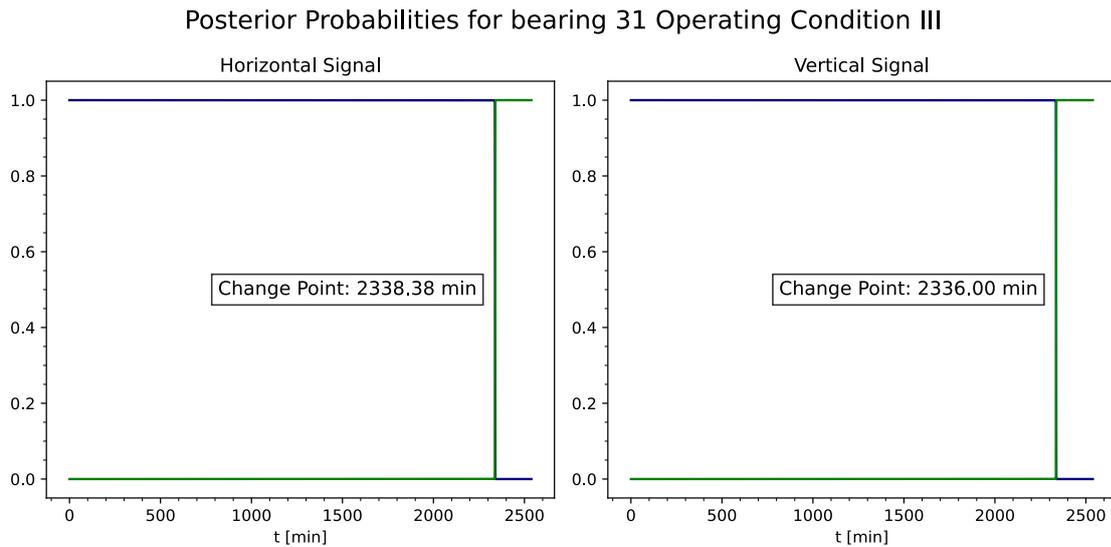


**Abbildung 4.14** – Ergebnis der Fehlerdetektion: Lager 2-1 Betriebsbedingung II

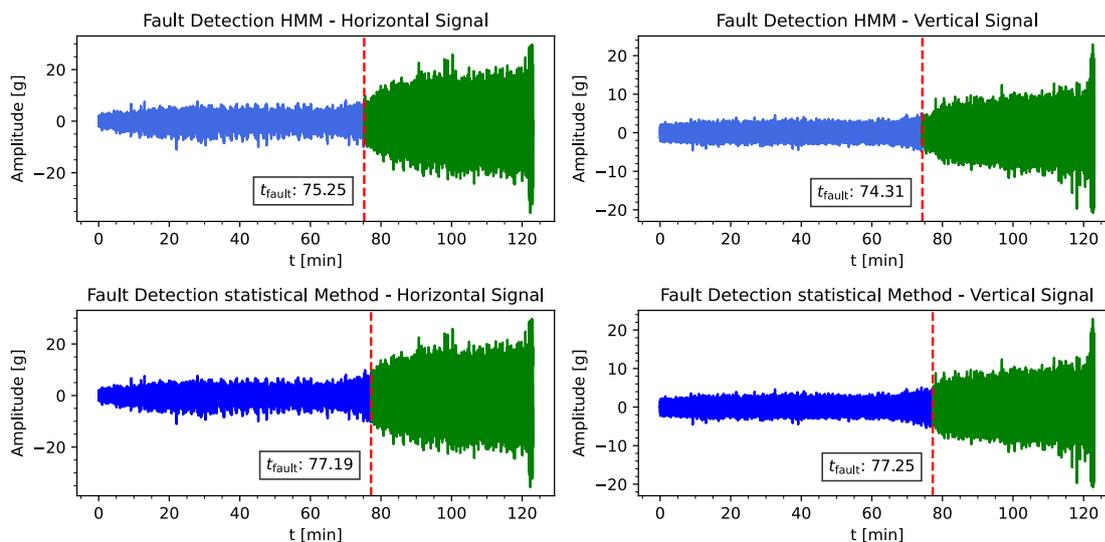
Lager 3-1 zeigt ein sehr eindeutiges Ergebnisse in der Betrachtung der posterioren Wahrscheinlichkeiten. Der Übergang findet abrupt statt vgl. Abbildung 4.15.

#### 4.4.3 Vergleich zwischen Fehlerdetektionsmethoden

Die Ergebnisse der Fehlerdetektion zwischen der modellbasierten und datenbasierten Methode werden nachfolgend dargestellt und miteinander verglichen. Der Vergleich erfolgt zum einen über Differenz der Ergebnisse des Fehlerauftrittszeitpunktes  $\Delta t_{fault}$  sowie in diesem Zusammenhang die grafische Einordnung aufgrund des zuvor dargestellten exponentiellen Degradationsverhaltens von Vibrationssignalen. Tabelle 4.6 stellt die alle Ergebnisse des Fehlerauftrittszeitpunktes gegenüber. Die Gegenüberstellung der Ergebnisse von Lager1-1 zeigt erwartungsgemäß ähnliche Fehlerauftrittszeitpunkte. Die Differenz zwischen den Methoden und den Merkmalen liegt zwischen einer und zwei Minuten.

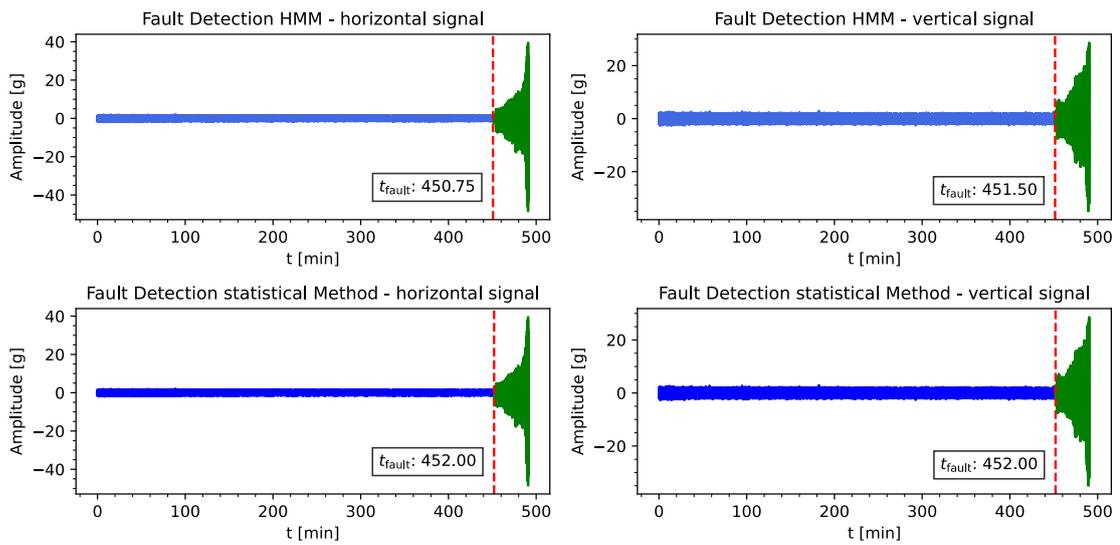


**Abbildung 4.15** – Ergebnis der Fehlerdetektion: Lager 3-1 Betriebsbedingung III



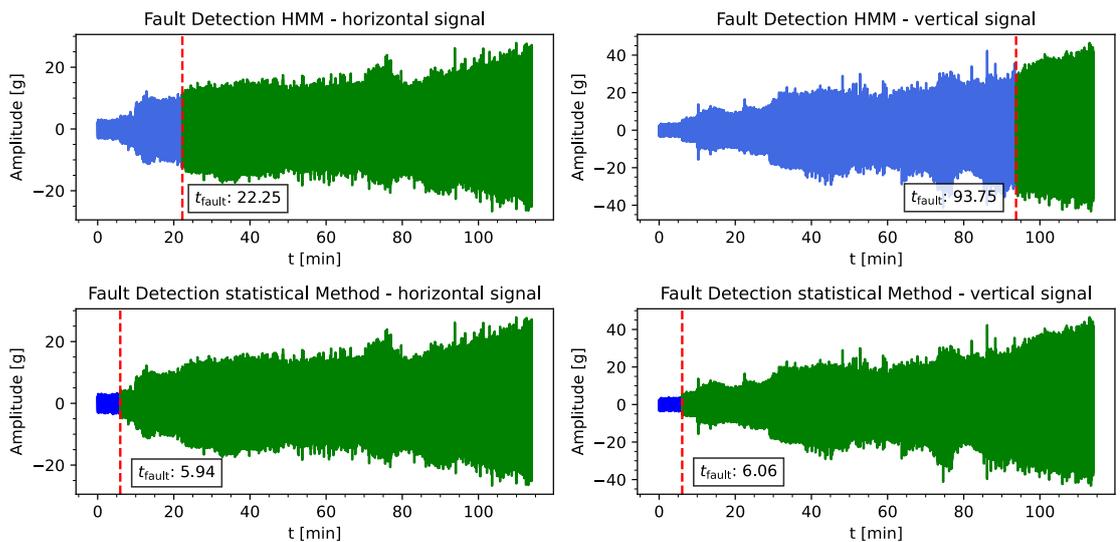
**Abbildung 4.16** – Vergleich der Fehlerdetektionsergebnisse: Lager 1-1 Betriebsbedingung I

Der Vergleich zwischen den Ergebnissen von Lager2-1 zeigt ebenfalls eine große Ähnlichkeit in der Detektion des Fehlerauftritts. Jedoch ist hier anzumerken, dass das HMM keine eindeutige Zuordnung eines Übergangs feststellen konnte. Lediglich durch die Annahmen, dass der Fehlerauftritt höchswahrscheinlich nicht zu Beginn auftritt, sowie die Hinzunahme der Ergebnisse des modellbasierten Ansatzes führt zu den ähnlichen Fehlerauftrittszeitpunkten. Auffällig bei der Betrachtung der unterschiedlichen Detektionsergebnisse für Lager 3-5 ist, dass die Differenzen der Fehlerauftrittszeitpunkte zwischen den vertikalen Vibrationssignalen relativ groß sind. Da beträgt die Differenz zwischen den Fehlerauftrittszeitpunkten bei  $\Delta t_{\text{fault}} = 87,69 \text{ min}$ , d. h. das HMM detektiert 87,69 min später als die statistische Methode. Der grafische Vergleich zeigt, dass



**Abbildung 4.17** – Vergleich der Fehlerdetektionsergebnisse: Lager 2-1 Betriebsbedingung II

die Detektion der statistischen Methode hier genauer ist vgl. Abbildung 4.18.



**Abbildung 4.18** – Vergleich der Fehlerdetektionsergebnisse: Lager 3-5 Betriebsbedingung III

Aufgrund der großen Differenzen zwischen horizontalen und vertikalen Vibrationsignalen bei der Detektion mittels HMM werden die vertikalen Signale für die Klassifikation nicht weiter berücksichtigt. Die Ergebnisse der modell- und datenbasierten Methoden liegen für die horizontalen Signale mit kleinen Differenzen nah beieinander.

Tabelle 4.4 – Parameter der Fehlerdetektion des modellbasierten Ansatzes

Lager	Mittelwert	Std	Faktor h	Schwellenwert (h * std)	Sensitivitäts- parameter k	Fehlerauftritts- zeitpunkt [min]	Fehlerwert
bearing11h	0.000190	0.000644	3	0.001933	2	77.1875	0.003754
bearing11v	0.000148	0.000618	3	0.001854	2	77.2500	0.004971
bearing12h	0.000049	0.000159	3	0.000477	2	45.3125	0.000687
bearing12v	0.000060	0.000139	3	0.000417	2	30.5625	0.000710
bearing13h	0.000025	0.000089	3	0.000267	2	58.3125	0.000500
bearing13v	0.000019	0.000071	3	0.000213	2	57.6250	0.000414
bearing14h	0.000182	0.004165	3	0.012496	2	82.6875	0.014233
bearing14v	0.000020	0.000384	3	0.001153	2	76.0000	0.001237
bearing15h	0.000092	0.000386	3	0.001157	2	34.1250	0.001502
bearing15v	0.000125	0.000610	3	0.001830	2	33.7500	0.001980
bearing21h	0.000001	0.000031	3	0.000093	2	452.0000	0.000496
bearing21v	0.000000	0.000025	3	0.000075	2	452.0000	0.000413
bearing22h	0.000027	0.000123	3	0.001296	2	45.25	0.000567
bearing22v	0.000050	0.000161	3	0.156667	2	46.75	0.000670
bearing23h	0.000037	0.000130	3	0.000391	2	324.1250	0.000889
bearing23v	0.000035	0.000117	3	0.000352	2	324.2500	0.000732
bearing24h	0.000826	0.003792	3	0.011376	2	29.75	0.033596
bearing24v	0.000421	0.002019	3	0.006057	2	29.75	0.008974
bearing25h	0.000009	0.000082	3	0.000244	2	120.375	0.000403
bearing25v	0.000001	0.000063	3	0.000187	2	119.9375	0.000295
bearing31h	0.000001	0.000008	3	0.000024	2	2342.3125	0.000032
bearing31v	0.000001	0.000005	3	0.000016	2	2339.1250	0.000037
bearing32h	0.000004	0.000016	2.5	0.000041	2	1224.625	0.000050
bearing32v	0.000000	0.000015	2.5	0.000037	2	1225.375	0.000053
bearing33h	0.000008	0.000670	3	0.002011	2	341.625	0.007638
bearing33v	0.000079	0.000806	3	0.002418	2	340.9400	0.021319
bearing34h	0.000002	0.000024	3	0.000071	2	1405.5600	0.000198
bearing34v	0.000003	0.000024	3	0.000073	2	1405.8100	0.000195
bearing35h	0.000394	0.000765	3	0.002295	2	5.9375	0.003077
bearing35v	0.000605	0.001337	3	0.004011	2	6.0625	0.005114

Tabelle 4.5 – Parameter der Fehlerdetektion des datenbasierten Ansatzes

Lager	Mittelwerte $\mu_i$	Kovarianzen $\Sigma_i$	Übergangswahrs. $a_{ij}$	Anfangswahrs. $\pi_i$	Übergangspunkte $t_{fault} [min]$
bearing11h	0.3718, 0.0293	0.0163, 0.0001	0.9992, 0.0008; 0.0000, 1.0000	1, 0	75.25
bearing11v	0.3054, 0.0183	0.0165, 0.0000	1.0000, 0.0000; 0.0008, 0.9992	0, 1	74.31
bearing12h	0.3601, 0.0226	0.0124, 0.0003	1.0000, 0.0000; 0.0011, 0.9999	0, 1	52.06
bearing12v	0.3073, 0.0257	0.0166, 0.0004	1.0000, 0.0000; 0.0011, 0.9999	0, 1	54.94
bearing13h	0.0140, 0.2392	0.0000, 0.0532	0.9992, 0.0008; 0.0000, 1.0000	1, 0	70.81
bearing13v	0.0166, 0.2703	0.0002, 0.0609	0.9993, 0.0007; 0.0000, 1.0000	1, 0	89.75
bearing14h	0.0124, 0.0336	0.0000, 0.0010	0.9992, 0.0008; 0.0015, 0.9985	0, 1	1.25, 80.19
bearing14v	0.0333, 0.3978	0.0002, 0.0241	0.9995, 0.0005; 0.0000, 1.0000	1, 0	120.635
bearing15h	0.0138, 0.3565	0.0000, 0.0525	0.9982, 0.0018; 0.0000, 1.0000	1, 0	33.94
bearing15v	0.0098, 0.3136	0.0000, 0.0412	0.9982, 0.0018; 0.0000, 1.0000	1, 0	33.81
bearing21h	0.0081, 0.2678	0.0000, 0.0435	0.9999, 0.0001; 0.0014, 0.9986	0, 1	3.69, 450.75
bearing21v	0.0111, 0.2600	0.0000, 0.0653	0.9999, 0.0001; 0.0011, 0.9989	0, 1	17.94, 451.5
bearing22h	0.0111, 0.2626	0.0000, 0.0483	0.9999, 0.0001; 0.0011, 0.9989	0, 1	25.06, 452.56
bearing22v	0.0624, 0.0127	0.0904, 0.0000	1.0000, 0.0000; 0.0013, 0.9987	0, 1	46.69
bearing23h	0.5178, 0.0293	0.0346, 0.0002	1.0000, 0.0000; 0.0002, 0.9998	0, 1	315
bearing23v	0.0357, 0.5524	0.0002, 0.0530	0.9998, 0.0002; 0.0000, 1.0000	1, 0	316.81
bearing24h	0.0221, 0.6191	0.0001, 0.0454	0.9980, 0.0021; 0.0000, 1.0000	1, 0	29.75
bearing24v	0.0153, 0.5068	0.0000, 0.0688	0.9980, 0.0021; 0.0000, 1.0000	1, 0	29.81
bearing25h	0.4853, 0.0361	0.0573, 0.0004	1.0000, 0.0000; 0.0004, 0.9996	0, 1	163.06
bearing25v	0.3384, 0.0257	0.0432, 0.0001	1.0000, 0.0000; 0.0004, 0.9996	0, 1	163.19
bearing31h	0.0087, 0.1472	0.0000, 0.0225	0.9999, 0.0000; 0.0000, 1.0000	1, 0	2338.38
bearing31v	0.0079, 0.1121	0.0000, 0.0192	0.9999, 0.0000; 0.0000, 1.0000	1, 0	2336
bearing32h	0.1712, 0.0507	0.0086, 0.0002	1.0000, 0.0000; 0.0000, 1.0000	0, 1	2336
bearing32v	0.0448, 0.1303	0.0000, 0.0051	0.9999, 0.0001; 0.0001, 0.9999	1, 0	1,44, 93,31, 1558.06
bearing33h	0.0208, 0.6896	0.0001, 0.0450	0.9998, 0.0002; 0.0000, 1.0000	1, 0	340.94
bearing33v	0.0225, 0.7084	0.0001, 0.0466	0.9998, 0.0002; 0.0000, 1.0000	1, 0	340.94
bearing34h	0.0175, 0.2821	0.0000, 0.0782	0.9999, 0.0000; 0.0004, 0.9996	0, 1	52.06, 103.13
bearing34v	0.0250, 0.3438	0.0000, 0.0467	0.9999, 0.0000; 0.0000, 1.0000	1, 0	1402.31
bearing35h	0.6067, 0.1922	0.0124, 0.0235	1.0000, 0.0000; 0.0028, 0.9972	0, 1	22,25
bearing35v	0.2821, 0.7736	0.0180, 0.0055	0.9993, 0.0007; 0.0000, 1.0000	1, 0	93,75

**Tabelle 4.6** – Vergleich der Ergebnisse zwischen der statistischen Methode und dem HMM zur Fehlerdetektion

<b>Lager</b>	<b>statistische Methode - horizontal</b>	<b>HMM - horizontal</b>	<b>statistische Methode - vertikal</b>	<b>HMM - vertikal</b>
	$t_{fault}[min]$	$t_{fault}[min]$	$t_{fault}[min]$	$t_{fault}[min]$
bearing11	77.19	75.25	77.25	74.31
bearing12	45.31	55.19	30.56	54.94
bearing13	58.31	77.12	57.62	89.75
bearing14	82.69	79.94	76.00	120.38
bearing15	34.12	33.94	33.75	33.81
bearing21	452.00	450.75	452.00	451.50
bearing22	45.25	79.44	46.75	84.69
bearing23	324.12	318..12	324.25	318.12
bearing24	29.75	33.44	29.75	29.88
bearing25	120.38	162.38	119.94	162.75
bearing31	2342.31	2338.38	2339.12	2336.00
bearing32	1224.62	1560.56	1225.38	1558.06
bearing33	341.62	340.94	340.94	340.94
bearing34	1405.56	1403.12	1405.81	1402.31
bearing35	5.94	22.25	6.06	93.75

## 4.5 Implementierung des Modells zur Fehlerklassifikation

### 4.5.1 Datenvorbereitung

Die Datenvorbereitung erfolgt mittels Resampling Strategien. Im ersten Schritt werden die Daten aufgrund der Klassenimbalance neu abgetastet (Resampling). Abbildung 4.19 zeigt die Verteilung vor und Abbildung 4.21 nach der Anwendung der Strategien. Zusätzlich wurden die Daten für Betriebsbedingung III (um 40% gekürzt), da aufgrund der deutlich größeren Datenmenge die Rechenoperationen zur Hyperparameter Optimierung zu lange gedauert haben.

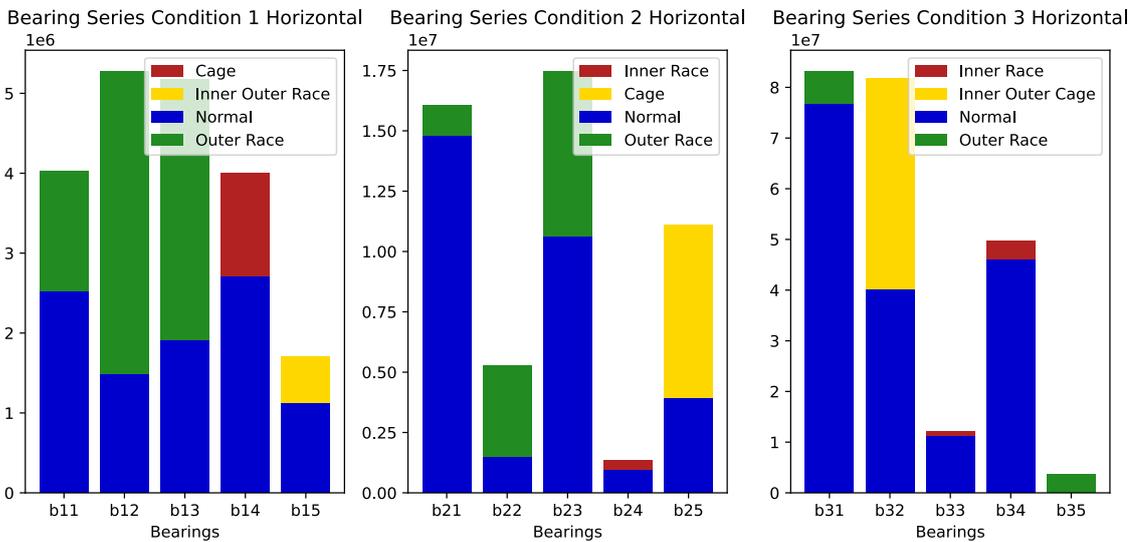


Abbildung 4.19 – Klassenverteilung vor der Anwendung der Resampling-Techniken

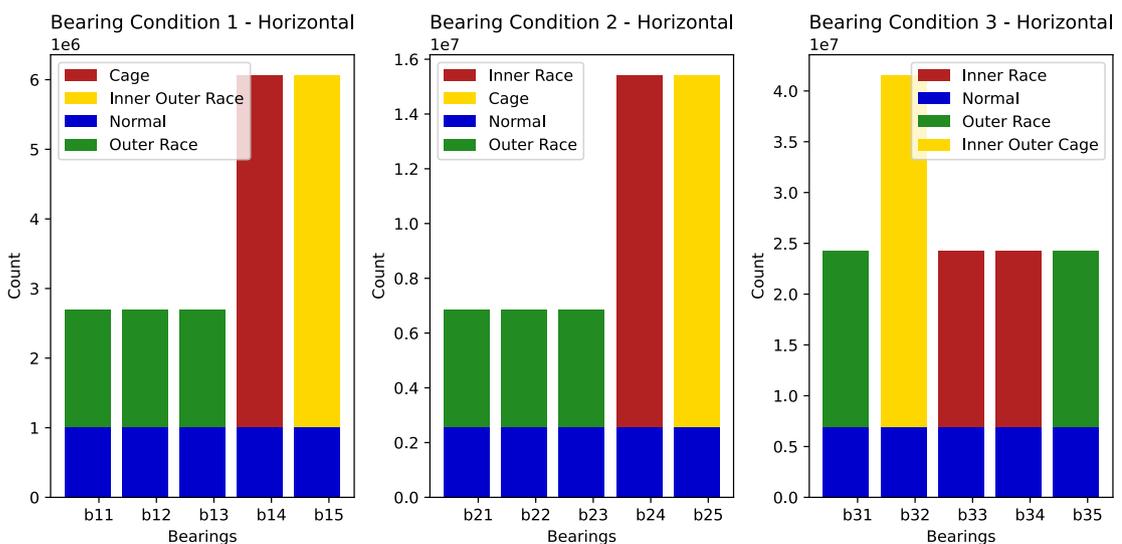


Abbildung 4.20 – Klassenverteilung nach der Anwendung der Resampling-Techniken

## 4.5.2 Datensatzaufteilung und Kreuzvalidierung

Nachdem die Klassenimbalance durch geeignete Resampling-Strategien angepasst wurden, erfolgt die Einteilung in Trainings-, Validierungs- und Testdatensätze. Die Aufteilung erfolgt nach dem 70/10/20 Verhältnis. Um die Klassenbalance in den Datensätzen zu gewährleisten, wird jeweils pro Klasse 70% zum Trainingsdatensatz, 10% zum Validierungsdatensatz und 20% zum Testdatensatz hinzugefügt. Nachfolgende Abbildung veranschaulicht diese Vorgehensweise.

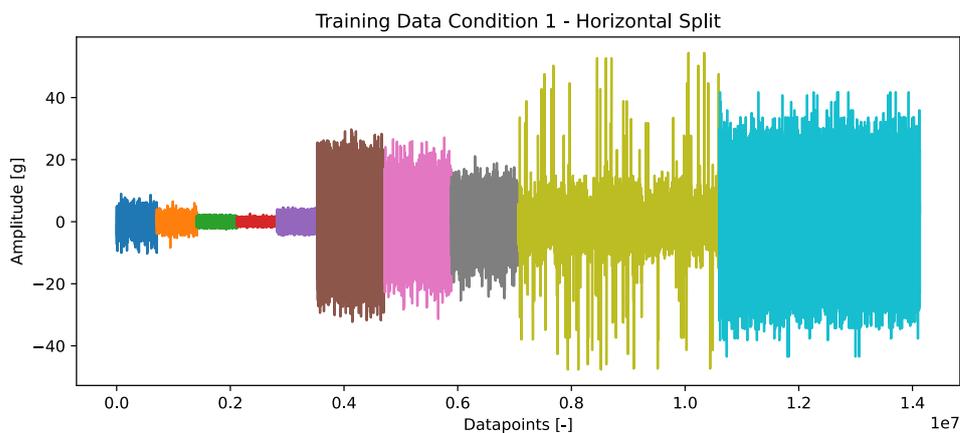


Abbildung 4.21 – Aufteilung des Trainingsdatensatzes für die Betriebsbedingung I

Für die Vorbereitung der Hyperparameter Strategie und zur Verhinderung von Überanpassung werden die Daten mit der k-fold-Kreuzvalidierung in fünf folds eingeteilt. Die Eigenschaften der Zeitreihe werden ab diesem Bearbeitungsschritt nicht weiter berücksichtigt. Tabelle 4.7 zeigt exemplarisch für Betriebsbedingung I die Aufteilung der folds.

Tabelle 4.7 – Horizontale k-Fold Kreuzvalidierung mit 5 folds

	Trainingsdimension	Validierungsdimensionen
<b>Fold 1</b>	X_train_fold: (23456, 1024, 1) y_train_fold: (23456, 1024, 4)	X_val_fold: (5864, 1024, 1) y_val_fold: (5864, 1024, 4)
<b>Fold 2</b>	X_train_fold: (23456, 1024, 1) y_train_fold: (23456, 1024, 4)	X_val_fold: (5864, 1024, 1) y_val_fold: (5864, 1024, 4)
<b>Fold 3</b>	X_train_fold: (23456, 1024, 1) y_train_fold: (23456, 1024, 4)	X_val_fold: (5864, 1024, 1) y_val_fold: (5864, 1024, 4)
<b>Fold 4</b>	X_train_fold: (23456, 1024, 1) y_train_fold: (23456, 1024, 4)	X_val_fold: (5864, 1024, 1) y_val_fold: (5864, 1024, 4)
<b>Fold 5</b>	X_train_fold: (23456, 1024, 1) y_train_fold: (23456, 1024, 4)	X_val_fold: (5864, 1024, 1) y_val_fold: (5864, 1024, 4)

### 4.5.3 Netzarchitektur

Abbildung 4.22 zeigt die Struktur des aufgebauten CNN-Modells. Berücksichtigt werden dabei die Ergebnisse der Hyperparameter Optimierungsstrategie mittels Hyperband. Die Ergebnisse werden in Tabelle 4.8 dargestellt.

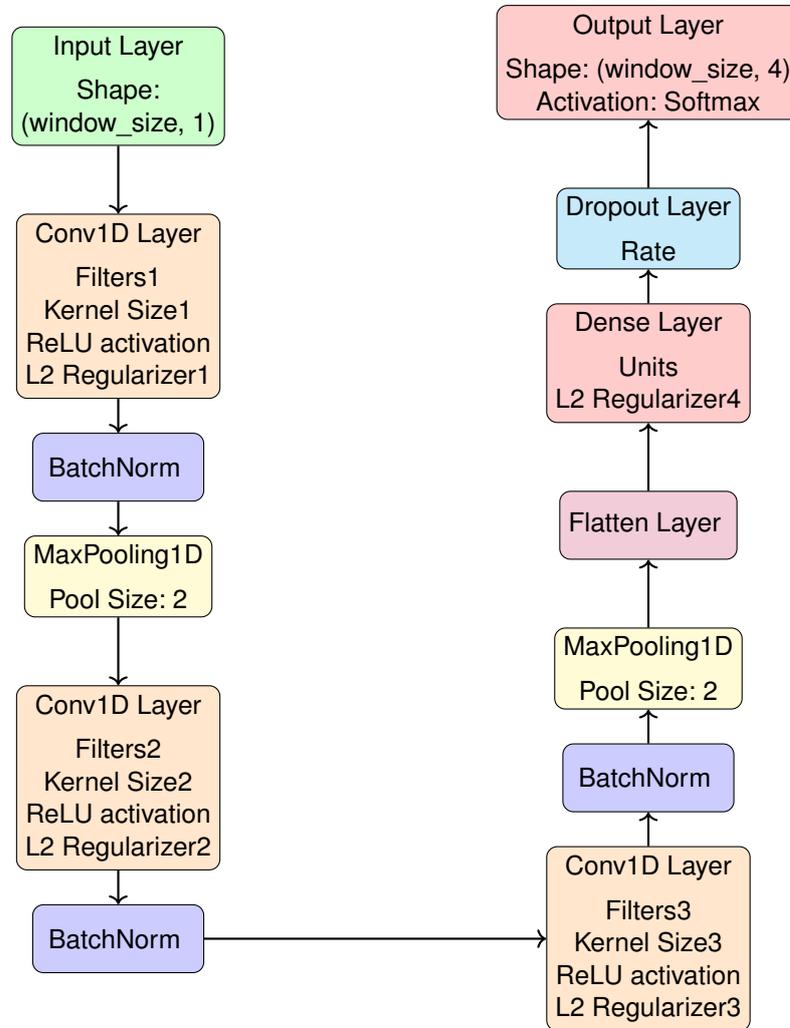


Abbildung 4.22 – Aufbau der CNN Struktur mit integrierter Hyperparameter Optimierung

### 4.5.4 Ergebnisse der Hyperparameter Optimierung

Nachfolgend werden in Tabelle 4.8 die Ergebnisse der Hyperparameter Optimierung dargestellt. Durchgeführt wurde diese separat für jede Betriebsbedingung und die entsprechend vorbereiteten Datensätze. Anzumerken ist, dass die *Epochs* je Optimierungsdurchlauf eine Empfehlung darstellen, beziehungsweise bei der Optimierung eine bestimmte Anzahl an *Epochs* für die Parameter verwendet wurde. Das Training und die Validierung mit den besten Hyperparameter erfolgte dann mit *Epochs=30*, wobei das Training und die Validierung maximal *Epochs=30* also Iterationsschritte benötigt, bis es

konvergiert. Im Hinblick auf die limitierten rechner-spezifischen Ressourcen wurde dieser Wert gewählt, da er in der Regel groß genug ist, damit das Modell konvergiert.

**Tabelle 4.8** – Best Hyperparameters for Different Operating Conditions

Hyperparameter	Betriebs- bedingung 1	Betriebs- bedingung 2	Betriebs- bedingung 3
filters_1	32	32	64
kernel_size_1	7	3	5
l2_regularizer_1	0.009338	5.5586e-05	0.001323
filters_2	192	64	64
kernel_size_2	3	3	7
l2_regularizer_2	1.6201e-05	2.9128e-05	4.1775e-05
filters_3	128	64	256
kernel_size_3	3	5	3
l2_regularizer_3	2.2644e-05	0.000218	0.000257
dense_units	150	50	50
l2_regularizer_4	2.2725e-05	0.000239	0.000310
dropout	0.1	0.4	0.4
learning_rate	0.000698	0.000217	0.000213
tuner/epochs	10	2	10

#### 4.5.5 Ergebnisse der Klassifikation

Die Darstellung der Klassifikationsergebnisse wird nachfolgend abgebildet. Dafür werden die Modellmetriken zur Evaluation des Modelloutputs herangezogen und die zugehörigen Konfusionsmatrizen dargestellt.

Die Accuracy (Genauigkeit) bezeichnet den Anteil der korrekt klassifizierten Instanzen an allen Instanzen. Werte von 0.991143, 0.999893 und 0.999831 zeigen, dass das Modell sehr zuverlässig arbeitet, da ein Großteil der Vorhersagen korrekt ist. Ein Wert nahe 1 weist auf eine hohe Genauigkeit hin, was bedeutet, dass das Modell die meisten Instanzen richtig klassifiziert hat.

Die Precision (Präzision) misst den Anteil der tatsächlich positiven Instanzen unter allen positiv klassifizierten Instanzen. Werte von 0.991165, 0.999893 und 0.999831 deuten darauf hin, dass das Modell sehr gut darin ist, relevante Instanzen zu identifizieren und wenige falsche Positive (fälschlicherweise als positiv klassifizierte Instanzen) erzeugt.

Der Recall (Sensitivität) gibt den Anteil der tatsächlich positiven Instanzen an, die korrekt als positiv klassifiziert wurden. Die Werte (0.991143, 0.999893, 0.999831) zeigen, dass das Modell auch sehr gut darin ist, tatsächliche positive Instanzen zu erkennen, und nur wenige relevante Instanzen übersieht. Der F1-Score ist das harmonische Mittel von Precision und Recall und bietet eine ausgewogene Metrik, die beide Aspekte berücksichtigt. Hohe Werte (0.991142, 0.999893, 0.999831) bedeuten, dass das Modell sowohl eine hohe Präzision als auch einen hohen Recall aufweist und somit insgesamt sehr gut performt.

Mean Squared Error, kurz MSE (Mittlerer quadratischer Fehler), misst den Durchschnitt der quadratischen Differenzen zwischen den tatsächlichen und vorhergesagten Werten. Kleinere Werte (0.035404, 0.000306, 0.001027) zeigen, dass das Modell sehr genau vorhersagt und die Abweichungen zwischen den vorhergesagten und den tatsächlichen Werten gering sind. Insbesondere der Vergleich der MSE-Werte zeigt, dass das Modell bei der zweiten und dritten Gruppe besonders genau arbeitet.

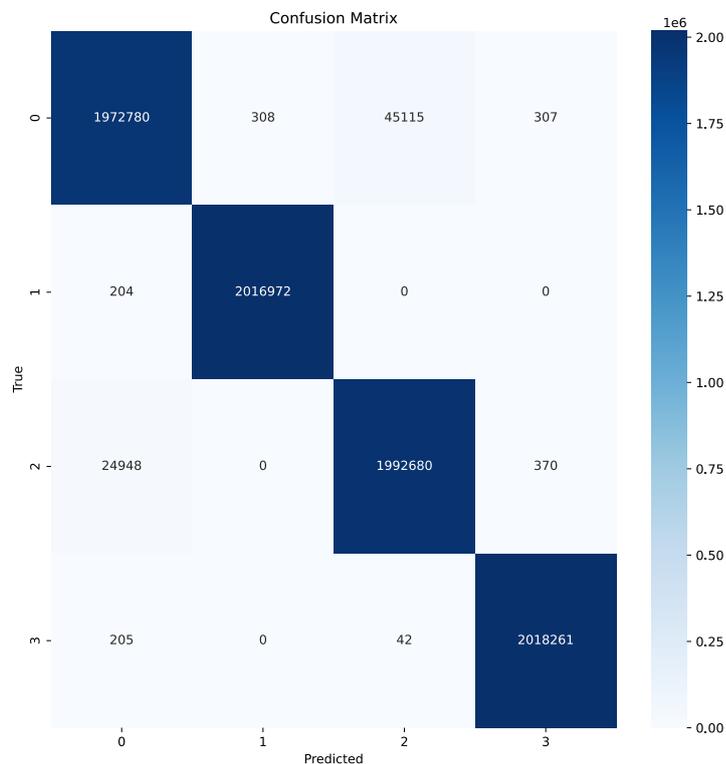
**Tabelle 4.9** – Vergleich der Klassifikationsergebnisse anhand der Modellmetriken

<b>Metriken</b>	<b>Werte Betriebs- bedingung I</b>	<b>Werte Betriebs- bedingung II</b>	<b>Werte Betriebs- bedingung III</b>
Accuracy	0.991143	0.999893	0.999831
Precision	0.991165	0.999893	0.999831
Recall	0.991143	0.999893	0.999831
F1 Score	0.991142	0.999893	0.999831
MSE	0.035404	0.000306	0.001027

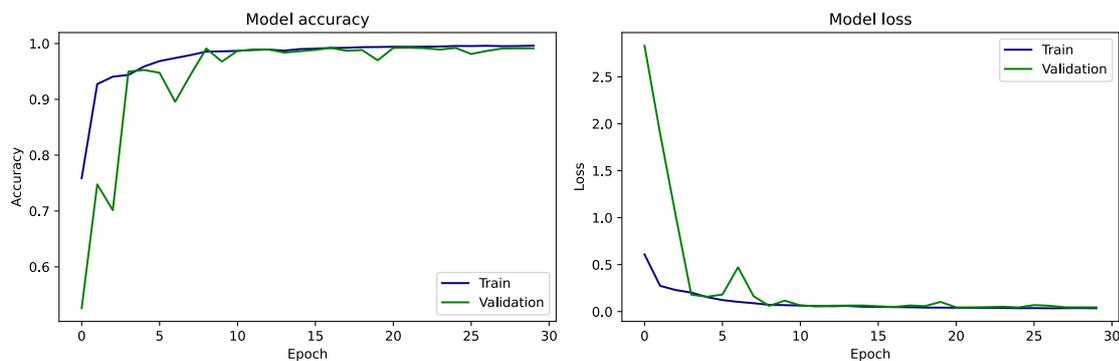
Die Konfusionsmatrizen stellen wahren und vorhergesagten Klassen dar. In Abbildung 4.23 wird deutlich, warum das Klassifikationsergebnis für Betriebsbedingung I schlechter ausgefallen ist, als für die anderen Bedingunge. Die Zelle in der ersten Zeile, dritte Spalte, zeigt, dass 45.115 Instanzen der Klasse 0 fälschlicherweise als Klasse 2 klassifiziert wurden. In der dritten Reihe, erste Spalte wurden 24.948 Instanzen der Klasse 2 als Klasse 0 klassifiziert. Es besteht eine gewisse Verwirrung zwischen den Klassen 0 und 2. Dies könnte darauf hinweise, dass diese beiden Betriebsbedingungen ähnliche Merkmale aufweisen. In Abbildung 4.24 wird die Modellgenauigkeit und der Modellverlust für Betriebsbedingung I dargestellt. Die Lücke zwischen dem Trainings- und dem Validierungsdatensatz schließt sich nach ca. zehn *Epochs*. Das Modell zeigt eine sehr gute Konvergenz.

Die Genauigkeit des Modells für Betriebsbedingung II ist sehr hoch, wie die Mehrheit der richtigen Klassifizierungen im Vergleich zu den falschen Klassifizierungen zeigt. Das Modell scheint besonders gut in der Unterscheidung zwischen den Klassen 0 und 1 zu sein, da es nur sehr wenige falsch positive und falsch negative Ergebnisse gibt vgl. Abbildung 4.25. In Abbildung 4.26 wird die Modellgenauigkeit und der Modellverlust für Betriebsbedingung II dargestellt. Die Trainingsgenauigkeit beginnt bei ca. 0.86 und reguliert sich nach wenigen *Epochs* auf über 0.99. Sowohl die Funktion der Modellgenauigkeit, als auch die Verlustfunktion zeigen kleinere Fluktuationen. Das Modell zeigt trotzdem eine sehr gute Konvergenz.

Auch die nachfolgende Abbildung 4.27 zeigt eine hohe Genauigkeit, die Mehrheit der Instanzen wird richtig klassifiziert. Das Modell zeigt eine sehr hohe Genauigkeit mit 13.826.113 richtigen Vorhersagen und minimalen Fehlklassifikationen for Klasse 0. Die Klasse 3 wird ebenfalls gut klassifiziert, weist aber etwas mehr Fehlklassifikationen auf, insbesondere durch Verwechslungen mit den Klassen 1 und 2. In Abbildung 4.28 wird die

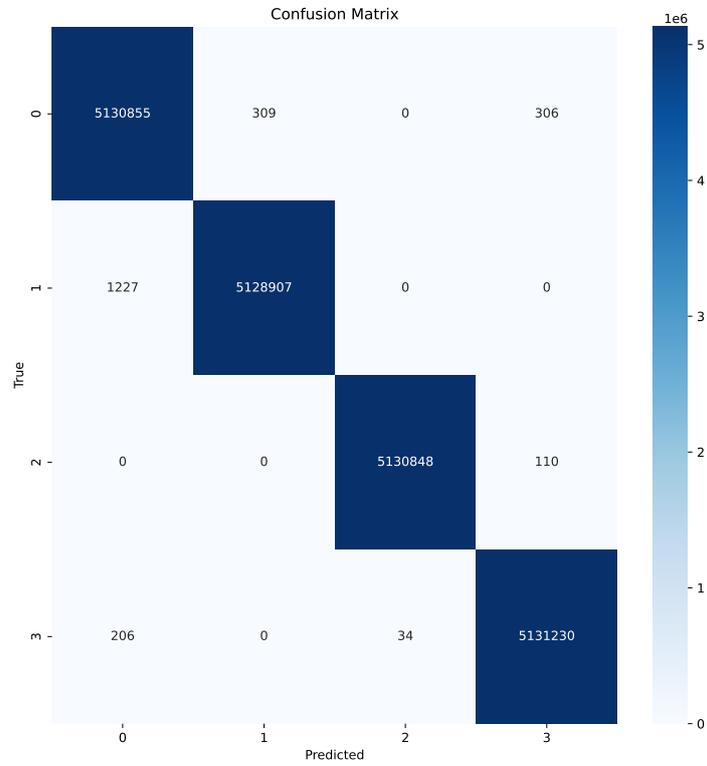


**Abbildung 4.23** – Darstellung des Klassifikationsergebnis für Betriebsbedingung I: Klasse 0-Käfig; Klasse 1-Innerer Ring, Äußerer Ring; Klasse 2-Normal, Klasse 3-Äußerer Ring

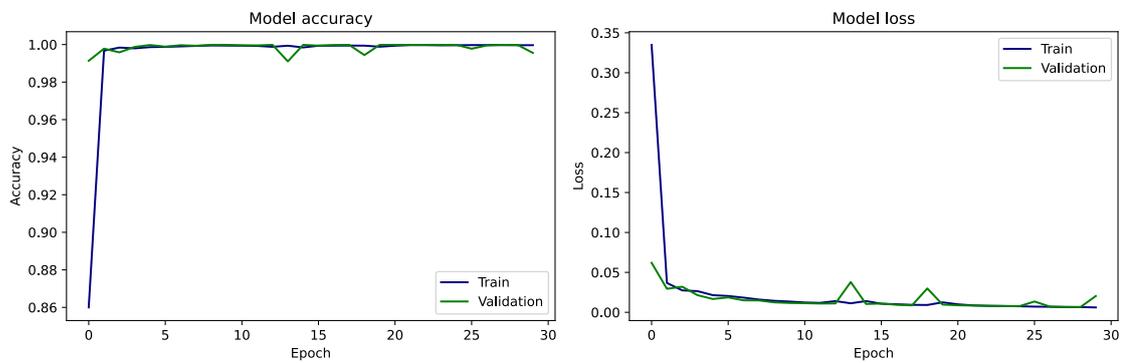


**Abbildung 4.24** – Darstellung Modellgenauigkeit und Modellverlust für Betriebsbedingung I

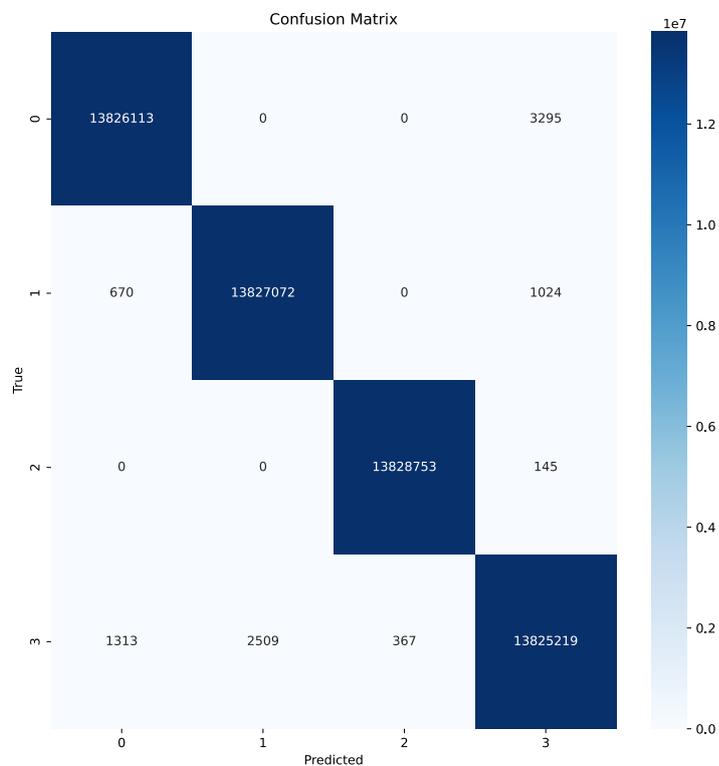
Modellgenauigkeit und der Modellverlust für Betriebsbedingung III dargestellt. Die Trainingsgenauigkeit pendelt sich bei ca. 0.97 ein und konvergiert innerhalb der 30 *Epochs* nicht. Die Validierungsgenauigkeit liegt deutlich über 0.97 und konvergiert gegen 1. Das stellt einen untypischen Verlauf dar. Das Modell kann mit den Validierungsdaten zu einfach auf die entsprechende Klasse schließen. Auch wenn der Verlauf untypisch ist, kann das Modell die Testdaten sehr gut voraussagen vgl. 4.27.



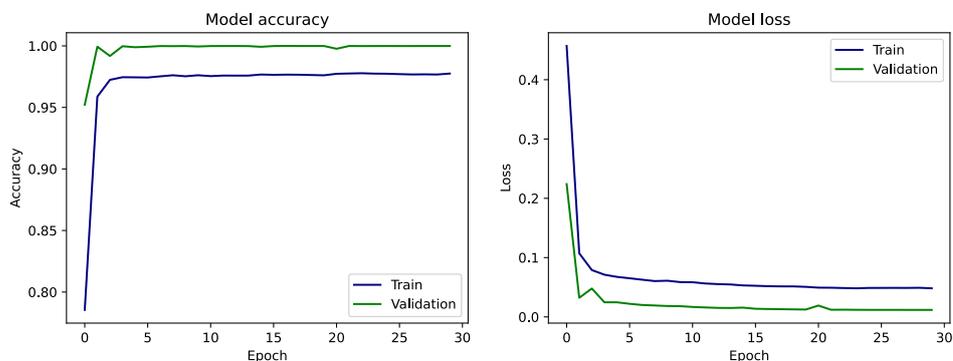
**Abbildung 4.25** – Darstellung des Klassifikationsergebnis für Betriebsbedingung II: Klasse 0-Innerer Ring; Klasse 1-Käfig; Klasse 2-Normal, Klasse 3-Äußerer Ring



**Abbildung 4.26** – Darstellung Modellgenauigkeit und Modellverlust für Betriebsbedingung II



**Abbildung 4.27** – Darstellung des Klassifikationsergebnis für Betriebsbedingung III: Klasse 0-Innerer Ring; Klasse 1-Innerer Ring, Äußerer Ring, Käfig; Klasse 2-Normal, Klasse 3-Äußerer Ring



**Abbildung 4.28** – Darstellung Modellgenauigkeit und Modellverlust für Betriebsbedingung III

---

## 5 Limitation und Diskussion

Im Hinblick auf die Auswahl eines zeitbasierten Parameters zur Darstellung des Degradationsverhaltens, können für die Vergleichbarkeit und die Erhöhung der Robustheit weitere Parameter herangezogen werden vgl. [41]. Spezifische Limitationen bietet der modellbasierte Ansatz durch die Vorauswahl eines geeigneten Intervalls, die Nutzung des Sensitivitätsparameters  $k$  und die resultierende Berechnung des arithmetischen Mittelwerts aufgrund des Intervalls. Zur Bewältigung dieser Limitationen kann eine regelbasierte Implementation zur Bestimmung des Mittelwerts sowie die zugehörige Standardabweichung zur Online-Applikation durchgeführt werden. Die Bestimmung der Parameter erfolgt nur für den Normalbereich, die ersten potentiellen Spitzenwerte werden inkludiert, die nachfolgenden exkludiert. Der Sensitivitätsparameter sollte bezüglich der zweifachen Nutzung getrennt werden. Ein Parameter regelt den Vergleich der potentiellen Spitzenwerte, der andere Parameter filtert die Spitzenwerte, um die tatsächlich höchsten Werte zu erhalten. Eine Ergänzung der Filterlogik kann im Hinblick auf die Früherkennung von Degradationsverhalten so gestaltet werden, dass der erste Wert nach dem Filterprozess ausgewählt.

Auswahl weiterer Parameter für die Klassifikationsaufgabe, die nicht zeitbasiert, sondern frequenzbasiert und/oder zeit-frequenzbasiert sind, siehe Signalprozessmodelle in [3]. Dies erhöht die Robustheit des Modelloutputs und der Vergleich des Klassifikationsergebnisses kann erfolgen. Das CNN hat sehr gute Ergebnisse erzielt, bei den Betriebsbedingungen II und III lagen die Klassifikationen nahezu bei 100%. Da eine sehr hohe Genauigkeit eine hohe Robustheit ausschließt, sollten geeignete Maßnahmen wie beispielsweise Datenaugmentierung durchgeführt werden. Dabei wird der Originaldatensatz verändert oder ergänzt, beispielsweise kann künstliches Rauschen hinzugefügt werden. Im Hinblick auf die erzielten Genauigkeiten bieten sich keine alternativen Klassifikatoren an. Es könnten lediglich Änderungen oder Ergänzungen der Struktur des neuronalen Netzes durchgeführt werden, z.B. CNN-GRU vgl. [40]. Der Verlauf der Modellgenauigkeit und des Modellverlusts zeigt innerhalb der 30 *Epochs* keine Konvergenz, auch wenn die Ergebnisse gegenüber der Testdaten sehr gute Voraussagen erzielen. Ein Grund dafür könnte die Datensatzkürzung sein. Aufgrund limitierter rechenspezifischer Ressourcen war die Hyperparameter Optimierung und somit das Training und die Validierung mit der Originalgröße nicht möglich. Die Rechenzeit belief sich pro Optimierungsdurchlauf (mit den genannten Rahmenbedingungen waren es insgesamt 90 pro Betriebsbedingung) zwischen 5-8 Stunden. Durch die Reduktion der Datenmenge hat sich die Berechnungszeit auf insgesamt 5-6 Stunden reduziert.



---

## 6 Fazit

Der Vergleich von datenbasierten und modellbasierten Ansätzen sowie die Anwendung eines Hybridmodells erhöht die Genauigkeit und die Robustheit der Detektionsaufgabe. Diese Kombination von Modellen führt zur Erhöhung der Gesamtperformance des Diagnosesystems. Besonders relevant ist die Performance im Zusammenhang mit der möglichen Prozessoptimierung, die einhergehende Erhöhung der Zuverlässigkeit und Anlagenverfügbarkeit aufgrund der frühzeitigen Fehlerdetektion und die genaue Diagnose über die Klassifikationsaufgabe. Mit den Ergebnissen lassen sich Entscheidungen und Maßnahmen ableiten (Fault Management, Maintenance, etc.). Weiter ist zu überprüfen inwiefern die Modellierung des neuronalen Netzes für andere Klassifikationsaufgaben verwendet werden kann (i.e. Transfer Learning), da durch eine mögliche Weiterverwendung auf ein anderes Problem in diesem Zusammenhang eine Kosten- und Zeitersparnis erzielt werden kann.

Im Hinblick auf die Umsetzbarkeit in einer produktionstechnischen Umgebung ist im betriebswirtschaftlichen Kontext zu prüfen, da besonders kleine und mittelständische Unternehmen weniger finanzielle Ressourcen zur Verfügung haben.



---

# Literaturverzeichnis

- [1] H. Kagermann, W.-D. Lukas und W. Wahlster, "Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution", *VDI nachrichten*, Jg. 13, Nr. 1, S. 2–3, 2011.
- [2] V. Gremium, B.-A. B. Data, J. Urbanski und M. Weber, "Herausgeber: BITKOM Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. Albrechtstraße 10 A 10117 Berlin-Mitte", 2012.
- [3] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2005.
- [4] R. Köhler, *Empirische und handlungstheoretische Forschungskonzeptionen in der Betriebswirtschaftslehre: Bericht über d. Tagung in Aachen, März 1976*. Poeschel, 1977.
- [5] A. Roth, *Einführung und Umsetzung von Industrie 4.0: Grundlagen, Vorgehensmodell und Use Cases aus der Praxis*. Springer-Verlag, 2016.
- [6] J. Schlick, P. Stephan, M. Loskyll und D. Lappe, "Industrie 4.0 in der praktischen Anwendung", *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendungstechnologien· Migration*, S. 57–84, 2014.
- [7] P. Gölzer, P. Cato und M. Amberg, "Data processing requirements of industry 4.0-use cases for big data applications", 2015.
- [8] M. Broy, *Cyber-physical systems: Innovation durch softwareintensive eingebettete Systeme*. Springer-Verlag, 2011.
- [9] B. Vogel-Heuser, T. Bauernhansl und M. Ten Hompel, "Handbuch Industrie 4.0 Bd. 4", *Allgemeine Grundlagen*, Jg. 2, 2017.
- [10] M. Krauß, R. Konrad, M. Krauß und R. Konrad, "Einführung in drahtlose Sensor-/Aktor-Netzwerke", *Drahtlose ZigBee-Netzwerke: Ein Kompendium*, S. 5–6, 2014.
- [11] R. Anderl, K. Bauer, T. Bauernhansl u. a., "Forschungsagenda Industrie 4.0: Aktualisierung des Forschungsbedarfs", *Ergebnispapier, Plattform Industrie*, Jg. 4, 2016.
- [12] C. Meinel und H. Sack, *Internetworking: Technological foundations and applications*. Springer Science & Business Media, 2013.
- [13] K. Perset, "Internet Addressing: Measuring Deployment of IPV6: Measuring Deployment of IPV6", 2010.
- [14] A. Luntovskyy und D. Gütter, "Moderne rechnernetze", *Protokolle, Standards und Apps in kombinierten drahtgebundenen, mobilen und drahtlosen Netzwerken*, 2020.

- [15] V. Venkatasubramanian, R. Rengaswamy, K. Yin und S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods", *Computers & chemical engineering*, Jg. 27, Nr. 3, S. 293–311, 2003.
- [16] A. Böge, *Handbuch Maschinenbau: Grundlagen und Anwendungen der Maschinenbau-Technik*. Springer, 2021, Bd. 24, S. 641, 657–658.
- [17] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui und M. Jordan, "Theoretically principled trade-off between robustness and accuracy", in *International conference on machine learning*, PMLR, 2019, S. 7472–7482.
- [18] A. Emami-Naeini, M. M. Akhter und S. M. Rock, "Effect of model uncertainty on failure detection: the threshold selector", *IEEE Transactions on Automatic Control*, Jg. 33, Nr. 12, S. 1106–1115, 1988.
- [19] S. B. Johnson, T. Gormley, S. Kessler u. a., *System health management: with aerospace applications*. John Wiley & Sons, 2011.
- [20] J. J. M. Jimenez, S. Schwartz, R. Vingerhoeds, B. Grabot und M. Salaün, "Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics", *Journal of Manufacturing Systems*, Jg. 56, S. 539–557, 2020.
- [21] Y. Wilhelm, P. Reimann, W. Gauchel und B. Mitschang, "Overview on hybrid approaches to fault detection and diagnosis: Combining data-driven, physics-based and knowledge-based models", *Procedia Cirp*, Jg. 99, S. 278–283, 2021.
- [22] G. Welch, G. Bishop u. a., "An introduction to the Kalman filter", 1995.
- [23] M. Aly, "Survey on multiclass classification methods", *Neural Netw*, Jg. 19, Nr. 1-9, S. 2, 2005.
- [24] T. Yu und H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications", *arXiv preprint arXiv:2003.05689*, 2020.
- [25] K. P. Murphy, *Probabilistic machine learning: an introduction*. MIT press, 2022.
- [26] J. Schneider, F. Igelbrink und W. Brockmann, "Inkrementelle nichtlineare Dimensionsreduktion", in *PROCEEDINGS 25. WORKSHOP COMPUTATIONAL INTELLIGENCE*, S. 183.
- [27] L. L. Pipino, Y. W. Lee und R. Y. Wang, "Data quality assessment", *Communications of the ACM*, Jg. 45, Nr. 4, S. 211–218, 2002.
- [28] N. Lunardon, G. Menardi und N. Torelli, "ROSE: a package for binary imbalanced learning.", *R journal*, Jg. 6, Nr. 1, S. 79–84, 2014.
- [29] N. V. Chawla, K. W. Bowyer, L. O. Hall und W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique", *Journal of artificial intelligence research*, Jg. 16, S. 321–357, 2002.

- [30] R. Kohavi u. a., “A study of cross-validation and bootstrap for accuracy estimation and model selection”, Jg. 14, Nr. 2, S. 1137–1145, 1995.
- [31] D. Berrar u. a., *Cross-validation*. 2019.
- [32] C. Shearer, “The CRISP-DM model: the new blueprint for data mining”, *Journal of data warehousing*, Jg. 5, Nr. 4, S. 13–22, 2000.
- [33] H. Wittel, D. Jannasch, J. Voßiek und C. Spura, *Roloff/Matek Maschinenelemente*. Springer, 2017, Bd. 23, S. 506, 510.
- [34] B. Wang, Y. Lei, N. Li und N. Li, “A hybrid prognostics approach for estimating remaining useful life of rolling element bearings”, *IEEE Transactions on Reliability*, Jg. 69, Nr. 1, S. 401–412, 2018.
- [35] N. Gebraeel, “Sensory-updated residual life distributions for components with exponential degradation patterns”, *IEEE Transactions on Automation Science and Engineering*, Jg. 3, Nr. 4, S. 382–393, 2006.
- [36] L. Karlöf, T. Ølgård, F. Godtliebsen, M. Kaczmarska und H. Fischer, “Statistical techniques to select detection thresholds for peak signals in ice-core data”, *Journal of Glaciology*, Jg. 51, Nr. 175, S. 655–662, 2005.
- [37] G. Palshikar u. a., “Simple algorithms for peak detection in time-series”, in *Proc. 1st Int. Conf. advanced data analysis, business analytics and intelligence*, Bd. 122, 2009.
- [38] C. M. Bishop und N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, Bd. 4.
- [39] Z. Li, F. Liu, W. Yang, S. Peng und J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects”, *IEEE transactions on neural networks and learning systems*, Jg. 33, Nr. 12, S. 6999–7019, 2021.
- [40] X. Li, “CNN-GRU model based on attention mechanism for large-scale energy storage optimization in smart grid”, *Frontiers in Energy Research*, Jg. 11, S. 1 228 256, 2023.
- [41] H. Zhou, X. Huang, G. Wen u. a., “Construction of health indicators for condition monitoring of rotating machinery: A review of the research”, *Expert Systems with Applications*, Jg. 203, S. 117 297, 2022.



---

# Anhang