



BERGISCHE
UNIVERSITÄT
WUPPERTAL

Fakultät für Maschinenbau und Sicherheitstechnik

Masterthesis

im Studiengang **Qualitätsingenieurwesen** beim Fachgebiet für
Verkehrssicherheit und Zuverlässigkeit

zur Erreichung des akademischen Grades

Master of Science

Thema:	Zuverlässigkeitstechnik und prädikative Instandhaltung in der Industrie 4.0 am Beispiel eines Druckluftkompressors
Autor:	Mustapha El Adnani
MatNr:	1942579
Bearbeitungszeitraum:	29.11.2023 bis 28.04.2024
Betreuer:	Jun.-Prof. Dr. Antoine Tordeux
Erstprüfer:	Jun.-Prof. Dr. Antoine Tordeux
Zweitprüfer:	Tim Maurice Julitz, M.Sc.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die von mir eingereichte Abschlussarbeit (Master-Thesis) selbstständig verfasst und keine andere als die angegebenen 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: Wuppertal, 15.04.2024

Unterschrift:

Abkürzungsverzeichnis

a_1	Eigenvektor
β_x	Parameter
$\hat{\beta}_x$	Parameter
E_i	Effekt
e_n^2	Differenz zwischen der tatsächlichen und der vorhergesagten Ausgabe
ε	Messfehler
F_k	Aktivierungsfunktion
HK_1	Hauptkomponente
N	Anzahl der Werte
N_F	Faktoren
N_S	Faktorstufen
p	Anzahl der Prädiktoren
\hat{p}_{mk}	Anteil der Trainingsdaten
R_x	Bereiche des Regressionsraums
r_{xy}	Bravais-Pearson-Koeffizient
s_k	Resultierender Wert
θ_k	Uni Bias
σ_x^2	Varianz
λ_1	Eigenwert
w_{jk}	Gewicht
\bar{x}	arithmetisches Mittel
x_i	Werte
X	Eingangsmatrix
X_p	Prädiktoren
Y	Labelvektor
γ	Momentum
y_j	Eingehende Werte
y_k	Ausgangswerte
F_{Ti}	Überlebenswahrscheinlichkeit
T_i	Dauer eines Instandhaltungszyklus

ANOVA	Analysis of Variance
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno Algorithmus
MAE	Mean Absolute Error
MS	Mean Square
MSE	Mean Square Error
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RSS	Residuenquadratsumme
SGD	Stochastic Gradient Descent
SSB	Sum of Square between Groups
SSW	Varianzanteil

TSS	Total Sum of Squares
CPS	Cyber-physische Systeme
IoT	Internet der Dinge
RUL	Remaining Useful Life
KI	Künstliche Intelligenz
ML	Maschinelles Lernen
IT	Informationstechnologie
ISO	International Standards Organization
DIN	Deutsches Institut für Normung
BetrSichV	Betriebssicherheitsverordnung
IEC	International Electrotechnical Commission
SIL	Safety Integrity Level
PFD	mittlere Ausfallwahrscheinlichkeit der Sicherheitsfunktion bei Anforderung
PFH	Wahrscheinlichkeit eines gefahrbringenden Ausfalls pro Stunde
E/E/PES	Elektronisch/Programmierbar Elektronisches System
EN	Europäische Norm

Abbildungsverzeichnis

Abbildung 1-1 – Industrieller Kompressor [1].....	1
Abbildung 2-1 Stufen der industriellen Revolution [15]	6
Abbildung 2-2: Die Schlüsseltechnologien der Industrie 4.0 [16].....	7
Abbildung 2-3: Zwiebschalenstruktur der CPSs [22].....	9
Abbildung 2-4: Wachstum der Datenmenge über die Zeit [26].....	10
Abbildung 2-5 Historische Entwicklung von „Sensor 1.0“ zu intelligent Sensoren, d.h. „Sensor 4.0“ in Anlehnung an Peter Krause, Vorsitzender von AMA e.V. [40].	11
Abbildung 2-6: Die Struktur der Lernverfahren (Maschinelles Lernen ist ein Teilbereich der Künstlichen Intelligenz) [44]	13
Abbildung 2-7: Badewannenkurve für Hardware-Verlässlichkeit [52].....	15
Abbildung 2-8: Überarbeitete Badewannenkurve für Software-Zuverlässigkeit [52].	15
Abbildung 2-9: Gesamter Sicherheitslebenszyklus [61].....	19
Abbildung 2-10: Strukturierung von Instandhaltungsstrategien in Anlehnung an DIN EN 13306 [63].....	20
Abbildung 2-11: Auswahl der Dauer des Instandhaltungszyklus [63].....	22
Abbildung 2-12 - Aufbau eines Kompressors [67].....	24
Abbildung 2-13: Datensatz aus Kaggle in CSV-Datei mit allen gemessenen Daten [68].....	26
Abbildung 3-1 - Pipeline maschinelles Lernen nach [80]	33
Abbildung 3-2 - Trainings und Test Splits zur 3-fach Cross Validation.....	34
Abbildung 3-3 - Unit eines neuronalen Netzes nach [79].....	36
Abbildung 3-4 - Schematischer Aufbau neuronaler Netze [85]	37
Abbildung 3-5 – Entscheidungsbaumschema nach [86].	39
Abbildung 3-6 - Vergleich Entscheidungsbaum und Random Forest nach [86].....	40
Abbildung 4-1 – Rohdaten RPM, MOTOR_POWER, TORQUE	43
Abbildung 4-2 – Rohdaten OUTLET_PRESSURE_BAR, AIR_FLOW, NOISE_DB	43
Abbildung 4-3 – Rohdaten OUTLET_TEMP, WPUMP_OUTLET_PRESSURE, WATER_INLET_TEMP	44
Abbildung 4-4 – Rohdaten WATER_OUTLET_TEMP, WPUMP_POWER, WATER_FLOW	45
Abbildung 4-5 – Rohdaten OILPUMP_POWER, OIL_TANK_TEMP	45
Abbildung 4-6 - Rohdaten GAACX, GAACY, GAACZ.....	46
Abbildung 4-7 – Rohdaten HAACX, HAACY, HAACZ.....	46
Abbildung 4-8 - Rohdaten des Zustands der Kompressorkomponenten	47
Abbildung 4-9 - Histogramm der Zustände der Kompressorkomponenten.....	47
Abbildung 4-10 - Korrelationsanalyse von allen Features und Zuständen	48
Abbildung 4-11 - Effekte der Messwerte auf den Zustand der Lager	49
Abbildung 4-12 - Analyse des F-Werts für den Lagerzustand.....	50
Abbildung 4-13 - Effekte der Messwerte auf den Zustand der Wasserpumpe	51
Abbildung 4-14 - Analyse des F-Werts für den Wasserpumpenzustand	51
Abbildung 4-15 - Effekte der Messwerte auf den Zustand des Lüfters	52
Abbildung 4-16 -Analyse des F-Werts für des Lüfters	52
Abbildung 4-17 - Effekte der Messwerte auf den Zustand des Ventils	53
Abbildung 4-18 - Analyse des F-Werts für das Ventil.....	54
Abbildung 4-19 - Hauptkomponentenanalyse des Datensatzes.....	55
Abbildung 4-20 - Vorgehen in der statistischen Analyse.....	56
Abbildung 4-21 - Ergebnis der polynomialen Regression Vorhersagen im Vergleich zu realen Daten	58

Abbildung 4-22 - Ergebnis der polynomialen Regression und tatsächliche Werte im zeitlichen Verlauf	59
Abbildung 4-23 - Ergebnis der polynomialen Regression mit angepassten Vorhersagen	60
Abbildung 4-24 - Ergebnisse der Random Forest Regression Vorhersagen über tatsächlichen Werten	61
Abbildung 4-25 - Angepasste vorhergesagte und tatsächliche Werte der Random Forest Regression.....	62
Abbildung 4-26 - Ergebnisse des neuronalen Netzes Vorhersagen über tatsächlichen Werten	63
Abbildung 4-27 - Angepasste vorhergesagte und tatsächliche Werte des neuronalen Netzes	64
Abbildung 4-28 - Vorgehen in der Datenanalyse	65
Abbildung 5-1 - Vorgehen bei der Zustandsbewertung.....	67

Tabellenverzeichnis

Tabelle 2-1: Die Unterschiede zwischen Software- und Hardwarezuverlässigkeit [54]	16
Tabelle 2-2: SIL niedrige und höhere Anforderungsrate in Anlehnung zu IEC/EN 61508 [61]	18
Tabelle 3-1 - Korrelationsstärke anhand des Rangkorrelationskoeffizienten nach [73]	29
Tabelle 4-1 - Nötige Messwerte zur Bestimmung des Zustands der Komponente	54
Tabelle 4-2 - Vergleich der statistischen Fehler vor Anpassung der Vorhersagen	66
Tabelle 4-3 - Vergleich der statistischen Fehler nach Anpassung der Vorhersage	66

Inhaltsverzeichnis

Eidesstattliche Erklärung	I
Abkürzungsverzeichnis	II
Abbildungsverzeichnis	IV
Tabellenverzeichnis	VI
Zusammenfassung	IX
1 EINLEITUNG	1
1.1 Problemstellung	1
1.2 Zielstellung	3
1.3 Aufbau	3
2 DIE VIERTE INDUSTRIELE REVOLUSION – STAND DER TECHNIK	5
2.1 Grundlagen der Industrie 4.0	5
2.1.1 Die Entwicklung der Industrie 4.0	5
2.1.2 Industrie 4.0	7
2.2 Technologische Grundlagen für das Internet der Dinge	8
2.2.1 Cyber-physische Systeme	8
2.2.2 Big Data und IoT	9
2.2.3 Sensorik 4.0	10
2.2.4 Künstliche Intelligenz und maschinelles Lernen	12
2.3 Sicherheits- und Zuverlässigkeitsaspekte in der Industrie 4.0	13
2.3.1 Zuverlässigkeit in der Industrie 4.0	13
2.3.2 Betriebssicherheit und funktionale Sicherheit	17
2.4 Methoden der Instandhaltung	19
2.4.1 Korrektive Instandhaltung	20
2.4.2 Periodische Instandhaltung	21
2.4.3 Präventive Instandhaltung	21
2.4.4 Prädiktive Instandhaltung	22
2.4.5 Vergleich der Instandhaltungsmethoden	23
2.5 Kompressoren und Datenanalyse-Software: eine integrierte Perspektive.	23
2.5.1 Kompressoren	23
2.5.2 Datensatz und angewendete Programmiersprache	25

3	DATENBASIERTE PRÄDIKTIVE INSTANDHALTUNGSSTRATEGIEN	28
3.1	Statistische Methoden zur Bewertung eines Datensatzes	28
3.1.1	Statistische Grundlagen	28
3.1.2	Korrelationsanalyse	29
3.1.3	Hauptkomponentenanalyse	30
3.1.4	Effektanalyse	30
3.1.5	Varianzanalyse	31
3.2	Methoden zur Datenanalyse	32
3.2.1	Regression	34
3.2.2	Neuronale Netze	36
3.2.3	Decision Tree	38
3.3	Methoden zur Bewertung der Datenanalyse	40
4	ANWENDUNGSBEISPIEL DER PRÄDIKTIVEN INSTANDHALTUNG	42
4.1	Physikalische Analyse des Datensatzes	42
4.2	Statistische Bewertung des Datensatzes	47
4.3	Datenanalyse	56
4.4	Bewertung	65
5	FAZIT UND AUSBLICK	67
5.1	Fazit	67
5.2	Ausblick	68
	Literaturverzeichnis	69
	Anhang	75

Zusammenfassung

Zusammenfassung

In dieser Arbeit werden datenbasierte prädiktive Instandhaltungsstrategien im Kontext der vierten industriellen Revolution erforscht und entwickelt, um auf die Herausforderungen der zunehmenden Komplexität und Vernetzung in industriellen Systemen zu reagieren. Die Optimierung von Instandhaltungsprozessen spielt eine zentrale Rolle, um Effizienz und Zuverlässigkeit in der Industrie 4.0 zu steigern. Nach einer detaillierten Einführung in die Problemstellung, die Zielsetzung und den strukturellen Aufbau der Arbeit, bietet das zweite Kapitel einen umfassenden Überblick über den aktuellen Stand der Technik. Hierzu zählen die Bedeutung von Kompressoren, die Einflüsse der verschiedenen Technologien in Industrie 4.0 und diverse Instandhaltungsmethoden: von traditionellen Ansätzen bis hin zu fortschrittlichen prädiktiven Techniken.

Der Schwerpunkt liegt auf der Entwicklung und Evaluierung datenbasierter prädiktiver Instandhaltungsstrategien. Im dritten Kapitel werden verschiedene statistische und analytische Methoden wie Korrelations-, Hauptkomponenten- und Varianzanalyse sowie fortschrittliche Techniken zur Datenanalyse, darunter Regression, neuronale Netze und Entscheidungsbäume, vorgestellt und diskutiert.

Ein besonderes Augenmerk liegt auf dem Anwendungsbeispiel im vierten Kapitel, in dem die Implementierung und der Vergleich von drei spezifischen Methoden – Random Forest, polynomiale Regression und ein neuronales Netz – zur Vorhersage von Instandhaltungsbedarfen detailliert beschrieben werden. Das neuronale Netz zeigt dabei die besten Ergebnisse, indem es präzisere Vorhersagen und eine höhere Zuverlässigkeit im Vergleich zu den anderen Methoden bietet. Dies unterstreicht die Effektivität und das Potenzial neuronaler Netze in der prädiktiven Instandhaltung.

Das abschließende Kapitel fasst die Erkenntnisse zusammen und bietet einen Ausblick auf zukünftige Entwicklungen und Forschungsfelder. Die Arbeit demonstriert eindrucksvoll, wie durch den Einsatz moderner datenbasierter Analysemethoden und insbesondere neuronaler Netze präzise Vorhersagen über den Instandhaltungsbedarf getroffen werden können, was zu einer signifikanten Reduzierung von Ausfallzeiten und einer Verlängerung der Lebensdauer industrieller Anlagen führt.

Summary

In this thesis, data-based predictive maintenance strategies in the context of the fourth industrial revolution are explored and developed to address the challenges of increasing complexity and connectivity in industrial systems. Optimizing maintenance processes is central to enhancing efficiency and reliability in Industry 4.0. After a detailed introduction to the problem statement, objectives, and structural layout of the thesis, the second chapter provides a comprehensive overview of the current state of technology. This includes the importance of compressors, the impacts of diverse Technologies of Industry 4.0, and various maintenance methods, from traditional approaches to advanced predictive techniques.

The focus is on the development and evaluation of data-based predictive maintenance strategies. The third chapter introduces and discusses various statistical and analytical methods such as correlation, principal component, and variance analysis, as well as advanced data analysis techniques, including regression, neural networks, and decision trees.

Particular attention is paid to the application example in the fourth chapter, where the implementation and comparison of three specific methods – Random Forest, polynomial regression, and a neural network – for predicting maintenance needs are detailed. The neural network demonstrates the best results by providing more accurate predictions and higher reliability compared to the other methods. This highlights the effectiveness and potential of neural networks in predictive maintenance.

The final chapter summarizes the findings and offers an outlook on future developments and research fields. The thesis impressively demonstrates how the use of modern data-based analysis methods, especially neural networks, can lead to precise predictions of maintenance needs, resulting in a significant reduction of downtime and an extension of the lifespan of industrial facilities.

1 EINLEITUNG

Kompressoren, siehe Abbildung 1-1, sind in der Industrie unverzichtbar und tragen wesentlich zu verschiedenen Kernaspekten bei. Sie dienen als zentrale Energiequelle in pneumatischen Systemen, die Druckluft für den Antrieb von Werkzeugen und Maschinen liefern. Diese Druckluft spielt eine entscheidende Rolle in Fertigungs- und Montageprozessen.



Abbildung 1-1 – Industrieller Kompressor [1]

1.1 Problemstellung

In der Prozessindustrie, einschließlich der chemischen und petrochemischen Industrie, sind Kompressoren für das Komprimieren von Gasen für Reaktionen, Trennverfahren sowie zur Lagerung und zum Transport von Gasen erforderlich.

Darüber hinaus sind sie in Kühlsystemen und Klimaanlageanlagen von großer Bedeutung, wo sie Kühlmittel komprimieren und so den Kältekreislauf ermöglichen, was für die Temperaturregelung in Industrieanlagen, Lagerräumen und Bürogebäuden unerlässlich ist. Moderne Kompressoren tragen zudem zur Energieeffizienz und zum Umweltschutz bei, indem sie den Gesamtenergieverbrauch reduzieren und teilweise Energie zurückgewinnen, die ansonsten als Wärme verloren gehen würde [2].

In der Qualitätskontrolle sorgen Kompressoren für eine konstante, kontaminationsfreie Luftversorgung, was für Präzisionsarbeiten wie das Lackieren oder Reinigen von Komponenten entscheidend ist. Sie spielen auch eine wichtige Rolle in der Verpackungsindustrie, indem sie Maschinen zur Verpackung und Etikettierung von

Produkten steuern und bei pneumatischen Fördersystemen zum Transport von Materialien eingesetzt werden.

Abschließend sind Kompressoren in einigen Industriezweigen Teil von Sicherheitssystemen, wie beispielsweise in Atemluftsystemen für Feuerwehrleute und in Notfallluftversorgungen in chemischen Anlagen. Ihre kontinuierliche Weiterentwicklung und Optimierung bleiben entscheidend, um den steigenden Anforderungen der Industrie gerecht zu werden und deren Effizienz, Produktivität, Sicherheit sowie Umweltschutz zu gewährleisten.

Die Implementierung von Condition Monitoring auf Kompressoren im Rahmen der Industrie 4.0 stellt eine wesentliche Innovation dar, die die Effizienz und Zuverlässigkeit industrieller Prozesse maßgeblich verbessert. In einer Ära, in der Digitalisierung und Vernetzung von Produktionsanlagen dominieren, ermöglicht die fortschrittliche Überwachung von Maschinen und Anlagen, einschließlich Kompressoren, zahlreiche Vorteile [3].

Durch die kontinuierliche Überwachung der Betriebszustände von Kompressoren können Anomalien und potenzielle Defekte frühzeitig erkannt werden. Dies führt zu einer präventiven Wartungsstrategie, die nicht nur Ausfallzeiten minimiert, sondern auch die damit verbundenen Kosten reduziert. Darüber hinaus trägt das Condition Monitoring zur optimalen Nutzung und Wartung der Kompressoren bei, was deren Lebensdauer verlängert und den Verschleiß effektiv kontrolliert [4].

Ein weiterer entscheidender Vorteil ist die Optimierung des Energieverbrauchs. Da Kompressoren oft große Energieverbraucher sind, ermöglicht eine effiziente Überwachung und Steuerung, Ineffizienzen und Leistungsverluste zu identifizieren und zu korrigieren. Dies senkt nicht nur die Energiekosten, sondern unterstützt auch nachhaltige Umweltpraktiken.

Die zustandsorientierte Wartung, ermöglicht durch Condition Monitoring, verbessert die Planbarkeit von Wartungsarbeiten. Wartungseinsätze können basierend auf den realen Betriebsbedingungen der Kompressoren geplant werden, was zu einer gesteigerten Wartungseffizienz führt. Zudem bieten die gesammelten Daten wertvolle Einblicke in die Betriebsbedingungen und Leistung der Kompressoren. Diese Informationen sind entscheidend für die Prozessoptimierung, die Steigerung der Produktivität und unterstützen langfristige strategische Entscheidungen.

Die Integration von Kompressoren mit Condition Monitoring in IoT-Systeme in der Industrie 4.0 fördert zudem eine umfassende Vernetzung und Kommunikation zwischen verschiedenen Anlagenkomponenten. Dies unterstützt die Entwicklung einer intelligenten und automatisierten Produktionsumgebung und trägt zur Steigerung der Gesamtanlageneffektivität bei [5].

Insgesamt ist die Einführung von Condition Monitoring auf Kompressoren in der Industrie 4.0 ein entscheidender Schritt zur Realisierung einer intelligenteren, vernetzten und effizienteren industriellen Produktion. Sie hilft nicht nur, Kosten zu senken und die Betriebszeit zu maximieren, sondern fördert auch eine nachhaltige und zukunftsorientierte Industriepraxis.

1.2 Zielstellung

Ziel der Arbeit ist es, die Techniken der Arbeitssicherheit in Verbindung mit Instandhaltungsstrategien innerhalb der Industrie 4.0 zu untersuchen, wobei ein besonderer Fokus auf der prädiktiven Instandhaltung und dem Condition-Monitoring liegt. Angesichts der zunehmenden Digitalisierung und Vernetzung in der modernen Industrie, besonders im Kontext der Kompressor Überwachung, wird die RUL-Prognose (Remaining Useful Life) als zentrales Element betrachtet. Diese ermöglicht es, die verbleibende Lebensdauer von Maschinen und Anlagen vorherzusagen und stellt einen innovativen Ansatz in der Instandhaltung dar.

Die Hauptziele dieser Arbeit umfassen eine tiefgreifende Untersuchung und Analyse der verschiedenen Instandhaltungsstrategien und Methoden in Smart Factories, insbesondere der prädiktiven Instandhaltung und des Condition-Monitorings. Dabei wird auch die Rolle der Arbeitssicherheit im Rahmen dieser Techniken betrachtet, um sicherzustellen, dass ihre Implementierung keine Risiken für Mitarbeiter und Umwelt birgt. Ferner soll eine Ausfallanalyse auf Basis von Maschinendaten durchgeführt werden, um die Praktikabilität und Effektivität der RUL-Prognose zu bewerten und deren Funktionsweise in realen Produktionsumgebungen zu verstehen.

Ein weiteres Ziel ist es, einen detaillierten Vergleich der verschiedenen Methoden zur RUL-Prognose anzustellen, um deren Vor- und Nachteile sowie die Genauigkeit und Effektivität zu ermitteln. Durch die Erkenntnisse dieser Arbeit sollen Unternehmen in der Industrie 4.0 bei der Auswahl und Implementierung effektiver und sicherer Instandhaltungsstrategien unterstützt werden. Dies dient der Gewährleistung von Arbeitssicherheit, der Minimierung von Produktionsausfällen und der Steigerung der Effizienz von Produktionsprozessen. Die RUL-Prognose bildet in diesem Kontext einen wesentlichen Leitfaden und ein exemplarisches Anwendungsgebiet für die prädiktive Instandhaltung in einer hochtechnologischen Umgebung.

1.3 Aufbau

Diese Arbeit gliedert sich in fünf Hauptabschnitte, die sich auf die Instandhaltung eines Kompressors unter Anwendung der Methoden der Industrie 4.0 konzentrieren. Die Einleitung bildet den Auftakt und setzt den Rahmen für die Untersuchung. Hier wird zunächst das grundlegende Problem der Instandhaltung eines Kompressors erörtert und wie Industrie 4.0-Ansätze dazu beitragen können, diese Herausforderungen zu bewältigen. Weiterhin wird die Zielsetzung der Arbeit klar definiert und analysiert, wobei besonderes Augenmerk auf die Relevanz und den potenziellen Nutzen für die Branche gelegt wird.

Der zweite Abschnitt widmet sich dem aktuellen Stand der Technik. In diesem Teil werden die grundlegenden Konzepte und Technologien eingehend beschrieben, die für das Verständnis der gesamten Arbeit wesentlich sind. Dazu gehören die technischen Aspekte von Kompressoren, die Kernprinzipien und Technologien der Industrie 4.0, die Grundlagen der Instandhaltung und die damit verbundenen Arbeitssicherheitsmaßnahmen. Dieser Abschnitt dient als Fundament für das weitere Verständnis der spezifischen Anwendungen und Strategien.

Im dritten Abschnitt rückt die Arbeit in den Fokus der spezifischen Instandhaltungsstrategien, die auf das ausgewählte Anwendungsbeispiel des Kompressors zugeschnitten sind. Es werden verschiedene Ansätze und Methoden vorgestellt, die im Kontext von Industrie 4.0 für die Wartung und Instandhaltung von Kompressoren besonders relevant sind. Diese Strategien werden detailliert erläutert, um ein tiefes Verständnis ihrer Funktionsweise und ihres Potenzials zu vermitteln.

Der vierte Abschnitt stellt die praktische Anwendung dieser Instandhaltungsstrategien dar. Hier wird das zuvor theoretisch erörterte Wissen auf ein konkretes Beispiel angewandt, wobei der Fokus auf die Realisierung der Strategien im Rahmen eines spezifischen Datensatzes liegt. Dieser Teil illustriert, wie die in den vorherigen Abschnitten beschriebenen Konzepte und Methoden in der Praxis umgesetzt werden können.

Der abschließende Abschnitt bietet eine Zusammenfassung und Reflexion der gesamten Arbeit. Die erzielten Ergebnisse werden hier zusammengefasst, wobei die wichtigsten Erkenntnisse hervorgehoben werden. Weiterhin wird ein Fazit gezogen, das die zentralen Beiträge der Arbeit bewertet und kritisch betrachtet. Abschließend wird ein Ausblick auf zukünftige Anwendungsmöglichkeiten und Forschungsbereiche gegeben, die auf den Erkenntnissen dieser Arbeit aufbauen könnten. Dieser Teil schließt die Arbeit ab und bietet gleichzeitig Anregungen für weiterführende Untersuchungen.

2 DIE VIERTE INDUSTRIELLE REVOLUTION – STAND DER TECHNIK

In der Einleitung zum Stand der Technik werden grundlegende Aspekte behandelt, die für das Verständnis der Instandhaltungsstrategien in der Industrie 4.0 essenziell sind. Es wird ein Überblick über Kompressoren gegeben, ihre zentrale Rolle in industriellen Prozessen und die technologischen Entwicklungen in diesem Bereich. Das Konzept der Industrie 4.0, geprägt durch Digitalisierung, Vernetzung und Automatisierung, wird ebenso beleuchtet, einschließlich der damit verbundenen technologischen Fortschritte wie IoT, KI und Big Data [6].

Des Weiteren wird auf die verschiedenen Methoden der Instandhaltung eingegangen, von traditionellen Ansätzen bis hin zu modernen Strategien wie prädiktive Instandhaltung und Condition-Monitoring.

Zusammengefasst bietet dieser Teil einen kompakten und informativen Überblick über die technologischen und methodischen Grundlagen, die für die Instandhaltung in der modernen industriellen Landschaft entscheidend sind.

2.1 Grundlagen der Industrie 4.0

2.1.1 Die Entwicklung der Industrie 4.0

Der Begriff „Industrie 4.0“ ist weit verbreitet und wird häufig verwendet, jedoch bleibt seine genaue Definition oft unklar. Zahlreiche Autoren greifen bei dem Versuch einer Definition auf verschiedene theoretische Modelle zurück. Laut Roth existieren trotz laufender Forschungsarbeiten noch keine einheitlichen Standards für den Begriff der Industrie 4.0 [7]. Die Bezeichnung „Industrie 4.0“ legt nahe, dass bereits drei vorherige industrielle Revolutionen stattgefunden haben. Eine industrielle Revolution ist dadurch gekennzeichnet, dass eine neue Technologie eine breite Anwendung in der Industrie findet und dadurch etablierte Prozesse grundlegend verändert werden [8]. Der Rückgriff auf numerische Zählungen verdeutlicht die kontinuierliche Weiterentwicklung und Transformation des industriellen Sektors im Verlauf der Geschichte.

Die Entwicklung der Industrie 4.0 in Deutschland lässt sich am besten verstehen, wenn man sie im Kontext der gesamten industriellen Revolutionen betrachtet, die die moderne Wirtschaft geprägt haben. Von der ersten industriellen Revolution bis zur gegenwärtigen Ära der Industrie 4.0 hat Deutschland eine herausragende Rolle in der Gestaltung und Umsetzung technologischer Innovationen gespielt.

Dieser Abschnitt bietet eine allgemeine historische Einordnung, um einen wissenschaftlichen Kontext zu schaffen. Dadurch wird der Weg für eine eingehende Analyse geebnet.

Die erste industrielle Revolution, die im späten 18. Jahrhundert begann, wurde durch die Einführung von Dampfmaschinen und mechanisierten Produktionsprozessen geprägt [9]. Deutschland war in dieser Ära zwar nicht führend, jedoch haben industrielle Entwicklungen in Großbritannien und anderen Ländern die Grundlage für zukünftige Innovationen gelegt [10]. Deutsche Ingenieure und Unternehmer begannen, diese Technologien aufzugreifen und weiterzuentwickeln [11].

Während der zweiten industriellen Revolution im späten 19. und frühen 20. Jahrhundert spielte Deutschland eine entscheidende Rolle. Die Einführung von Elektrizität und die Entwicklung von Massenproduktionstechniken führten zu einer dramatischen Veränderung der Industrielandschaft. Diese Periode war auch durch den Aufstieg von Forschungs- und Bildungseinrichtungen gekennzeichnet, die die technologische Innovation weiter vorantrieben.

In der dritten industriellen Revolution ab den späten 1950er Jahren spielte Deutschland eine entscheidende Rolle, insbesondere in der Automobilindustrie und im Maschinenbau [12]. Die Einführung von Computern und die Automatisierung von Produktionsprozessen prägten diese Ära maßgeblich.

Der Begriff „Industrie 4.0“ wurde erstmals auf der Hannover Messe 2011 geprägt und markiert den Beginn der vierten industriellen Revolution [13]. Dieser Abschnitt ist durch die umfassende Digitalisierung und Vernetzung von Produktionsprozessen gekennzeichnet. In Deutschland war die Initiative zur Entwicklung von Industrie 4.0 eng mit der Förderung von Forschung und Entwicklung verbunden [14].

Die deutsche Industrie hat sich aktiv an Pilotprojekten und Forschungsinitiativen beteiligt, um die Machbarkeit und den Nutzen von Industrie 4.0 zu demonstrieren. Diese Erfahrungen flossen in die Entwicklung von Leitlinien und Best Practices ein, die weltweit Beachtung fanden [13]. Insgesamt zeigt die Entwicklung der Industrie 4.0 in Deutschland einen fortschreitenden Trend der technologischen Führung und Innovationsbereitschaft. Von den Anfängen der industriellen Revolution bis heute hat Deutschland kontinuierlich dazu beigetragen, die industrielle Landschaft zu gestalten und die Möglichkeiten neuer Technologien auszuschöpfen. Industrie 4.0 stellt dabei einen Höhepunkt dieser Entwicklung dar, der auf den Erfahrungen und Innovationen vergangener industrieller Revolutionen aufbaut.

In der Abbildung 2.1 ist zu sehen, wie sich der Komplexitätsgrad und die Art der Aufgaben in den Unternehmen im Laufe der Zeit verändert haben.

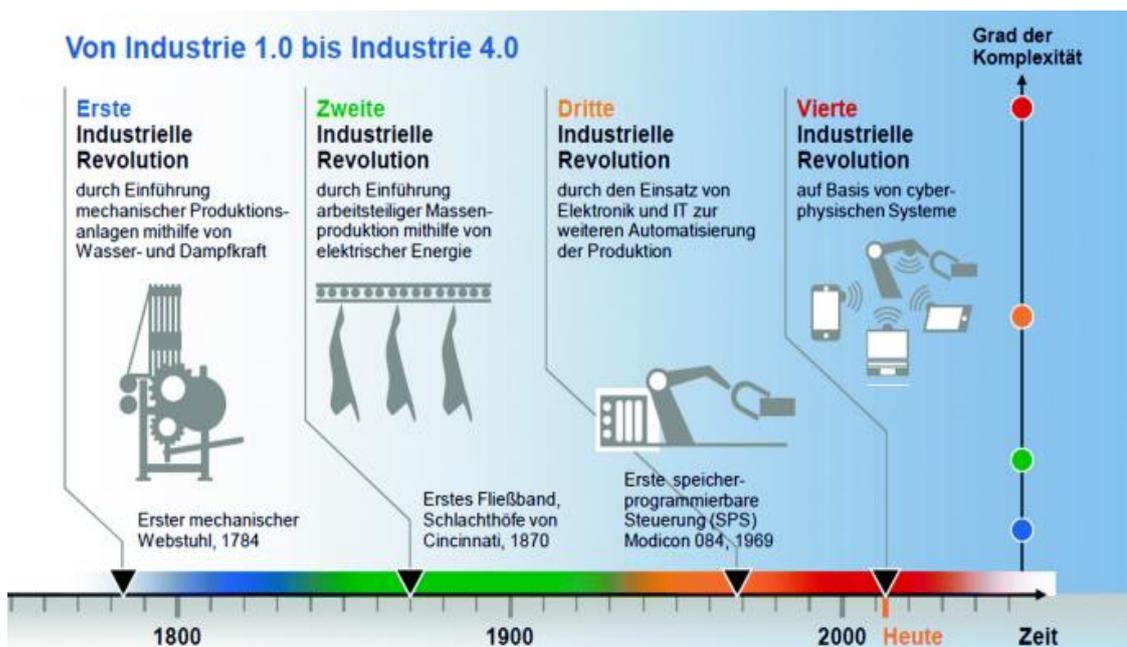


Abbildung 2-1 Stufen der industriellen Revolution [15]

Wissenschaftliche Arbeiten im Bereich Industrie 4.0 haben sich auf verschiedene Aspekte konzentriert. Die Integration von Cyber-Physischen Systemen (CPS), das Internet der Dinge (IoT) und künstliche Intelligenz standen im Mittelpunkt der technologischen Innovation [15]. Die grundlegenden Komponenten von Industrie 4.0 sind in Abbildung 2.2 dargestellt.

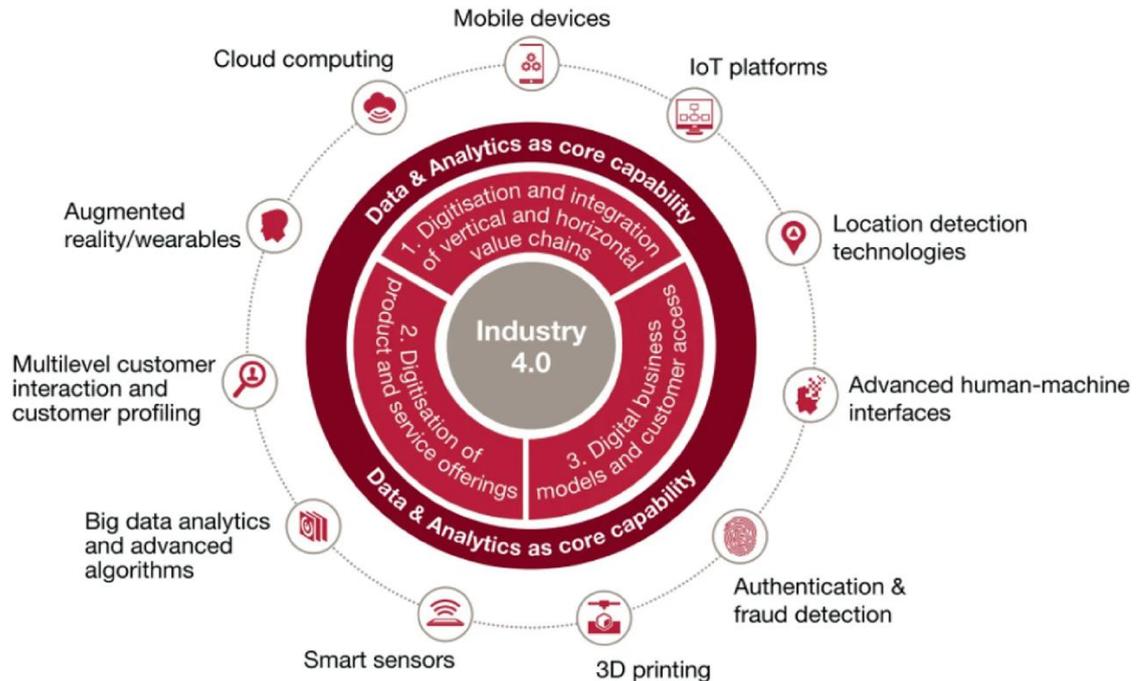


Abbildung 2-2: Die Schlüsseltechnologien der Industrie 4.0 [16]

Forscher analysierten die Auswirkungen auf die Arbeitswelt, die Qualifizierung von Mitarbeitern und organisatorische Veränderungen. Die Sicherheit von vernetzten Systemen wurde ebenfalls intensiv erforscht, um die Risiken im Zusammenhang mit Cyberangriffen zu minimieren [17].

2.1.2 Industrie 4.0

Industrie 4.0 markiert den Beginn der vierten industriellen Revolution, die sich durch eine umfassende Digitalisierung der Produktion auszeichnet. Dieses Konzept wurde, wie bereits erwähnt, erstmals 2011 auf der Hannover Messe öffentlich präsentiert und hat sich seitdem zu einem zentralen Leitbild für die moderne Fertigung entwickelt. Im Springer Gabler Wirtschaftslexikon wird Industrie 4.0 als ein Marketing- und Wissenschaftsbegriff beschrieben, der auch ein Zukunftsprojekt der deutschen Bundesregierung darstellt. [18] Die Besonderheit dieser industriellen Revolution liegt in der Individualisierung oder Hybridisierung der Produkte und in der engen Integration von Kunden und Geschäftspartnern in die Geschäftsprozesse.

Das Konzept von Industrie 4.0 hat international an Bedeutung gewonnen und wird weltweit unter verschiedenen Bezeichnungen diskutiert, wie zum Beispiel als „Internet of Things“ in den USA oder „Industrie du Futur“ in Frankreich. Die zugrundeliegenden Prinzipien sind jedoch universell: die Verknüpfung physischer und digitaler Systeme, die Implementierung von Cyber-Physischen Systemen und die Nutzung großer Datenmengen (Big Data) zur Optimierung der Produktionsprozesse.

Für die Zuverlässigkeitstechnik bringt Industrie 4.0 wichtige Herausforderungen mit sich. Das Erfassen und Überwachen von Maschinenzuständen in Echtzeit ermöglicht eine erhebliche Steigerung der Anlagenverfügbarkeit und -zuverlässigkeit, während gleichzeitig die Wartungskosten reduziert werden können. Techniken wie die vorausschauende Instandhaltung, auch prädiktive Instandhaltung genannt, und das Echtzeit-Condition-Monitoring spielen dabei eine wesentliche Rolle. Diese Methoden nutzen die Sammlung und Analyse von Daten, um Ausfälle vorherzusagen und den aktuellen Betriebszustand sowie die verbleibende Betriebsdauer (Remaining Useful Life, RUL) von Systemen zu prognostizieren.

Die erfolgreiche Umsetzung dieser Techniken erfordert nicht nur fundierte statistische Kenntnisse, sondern auch die Entwicklung und Anwendung von Algorithmen, die für die Verarbeitung multivariater Datensätze geeignet sind. Die Fähigkeit, aus diesen Datensätzen sinnvolle Muster und Erkenntnisse zu extrahieren, ist entscheidend, um vorbeugende Wartungsmaßnahmen effektiv planen und durchführen zu können. Insgesamt verspricht die Einführung von Industrie 4.0-Prinzipien eine Revolutionierung der industriellen Produktionslandschaft, indem sie eine höhere Flexibilität, Effizienz und eine stärkere Kundenorientierung ermöglicht.

2.2 Technologische Grundlagen für das Internet der Dinge

2.2.1 Cyber-physische Systeme

Die Entwicklung von Industrie 4.0 basiert auf etablierten wissenschaftlich-technischen Grundlagen, wobei Cyber-physische Systeme eine zentrale Rolle spielen. Die acatech Studie AgendaCPS von 2012 definiert Cyber-Physical Systems – die Basis für Industrie 4.0 – als eingebettete Systeme, die mittels Sensoren unmittelbar physikalische Daten erfassen und mittels Aktoren auf physikalische Vorgänge einwirken [19]. Wie in Studien betont, ermöglicht Interoperabilität die Echtzeitintegration von Menschen, Dingen und Systemen [13], Virtualisierung erlaubt eine digitale Repräsentation der realen Welt [20], während Dezentralisierung flexible Entscheidungen auf verschiedenen Ebenen ermöglicht [21].

Sie sind über digitale Netze untereinander verbunden, weltweit verfügbar und verfügen über eine Reihe multimodaler Mensch-Maschine-Schnittstellen. Diese Merkmale ermöglichen eine effektive Nutzung von Daten und Diensten sowie eine differenzierte Kommunikation und Steuerung. Die Struktur von Cyber-Physical Systems lässt sich am besten durch ein Zwiebelschalenmodell veranschaulichen, wie in Abbildung 2.3 dargestellt [22].

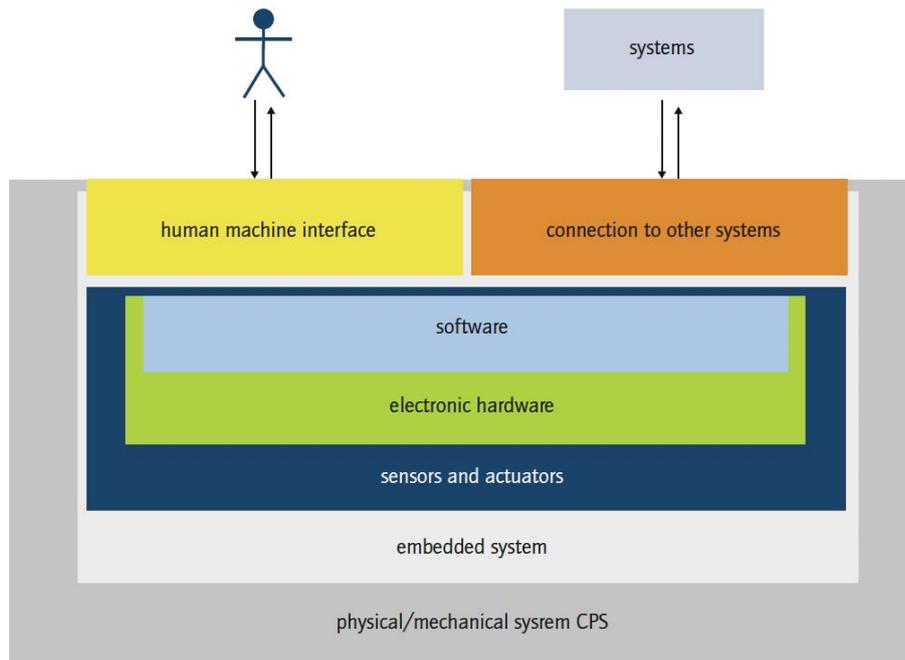


Abbildung 2-3: Zwiebelschalenstruktur der CPSs [22]

Die Integration dieser CPS in die so genannten Smart factories repräsentieren einen Übergang zur prädiktiven und präskriptiven Instandhaltung, was einen Höhepunkt in der wissensbasierten Instandhaltung darstellt [23]. Als eingebettete Systeme mit erweiterten Fähigkeiten können CPS eigenständig mit anderen CPS kommunizieren und Daten austauschen, was eine optimierte Vernetzung und Effizienz in Produktionsprozessen ermöglicht als auch die Interaktion zwischen Menschen und Maschine verbessert [24].

Die Integration von CPS revolutioniert die industrielle Landschaft, indem sie eine direkte Verbindung zwischen der physikalischen und der digitalen Welt schafft. CPS bieten neue Funktionalitäten durch Informations-, Daten- und Funktionsintegration.

Cyber-physische Systeme müssen hohen Sicherheits- und Zuverlässigkeitsstandards genügen, um unter anspruchsvollen physikalischen Bedingungen einsatzfähig zu sein [25]. Fragen der funktionalen Sicherheit, des Datenschutzes und der Datenintegrität sind daher von entscheidender Bedeutung. Insgesamt stellen CPS eine Schlüsseltechnologie dar, die die moderne Industrie und andere Sektoren maßgeblich beeinflusst.

2.2.2 Big Data und IoT

Die zunehmende Vernetzung von CPS führt zu einem rapiden Anstieg verfügbarer Daten, auch als Big Data bekannt. Gemäß dem Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. bezeichnet Big Data die Analyse großer Datenmengen aus vielfältigen Quellen mit dem Ziel, wirtschaftlichen Nutzen zu erzeugen [26]. Doch Big Data bleibt in Forschung und Praxis uneinheitlich definiert [27]. Diverse Perspektiven und Interessen führen zu unterschiedlichen Auffassungen über große Datenmengen, Technologie und Wirtschaftlichkeit [26]. Das Ziel eröffnet neue Möglichkeiten für Predictive Maintenance. Studien zeigen, dass durch Predictive

Maintenance eine Steigerung der Verfügbarkeit um 10 % bis 15 % sowie erhebliche Einsparungen bei den Instandhaltungskosten erzielt werden können [38].

Das rapide Wachstum der Datenmenge, insbesondere durch das Internet, treibt den Big-Data-Trend an [29]. Big Data verspricht wirtschaftlichen Nutzen in verschiedenen Unternehmensbereichen [26] (siehe Abbildung 2.4).

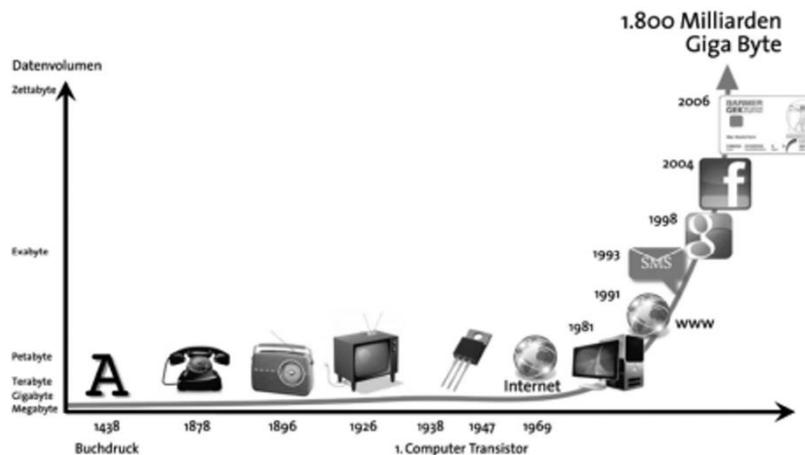


Abbildung 2-4: Wachstum der Datenmenge über die Zeit [26]

Die umfangreiche Datenerfassung und -analyse ermöglicht es, präzise Anforderungen und Anwendungsfälle in der Industrie 4.0 zu identifizieren und zu verstehen [30]. Die Verknüpfung von Big Data und Cyber-physischen Systemen bietet die Möglichkeit, Muster für die Instandhaltung zu erkennen und zu nutzen. Durch die strukturierte Aufarbeitung dieser Daten können Unternehmen potenzielle Schwachstellen und Engpässe in ihren Systemen frühzeitig erkennen und proaktiv handeln. Die Integration von Big Data-Analysen in CPS eröffnet somit neue Wege für eine effiziente und vorausschauende Instandhaltung, die dazu beiträgt, Ausfallzeiten zu minimieren und die Gesamtleistung der Anlagen zu optimieren.

Außerdem Algorithmen zur Verarbeitung großer Datenmengen ermöglichen die Mustererkennung und potenziell die Substitution oder Standardisierung von nicht-routinebasierten Tätigkeiten [31]. Dies betrifft Installation, Wartung und Reparatur, jedoch sind nicht alle Instandhaltungstätigkeiten davon betroffen [32]. Die Aufwertung von Instandhaltungsarbeit und -kompetenzen durch neue Technologien und Echtzeitdaten führt zur Flexibilisierung von Wartungszyklen und Intensivierung von Fernwartung [33]. Dynamisierung von Instandhaltungsstrategien und eine quantitative und qualitative Zunahme im Inspektionsbereich werden erwartet [34].

Um dieses Potenzial vollständig auszuschöpfen, ist die Implementierung eines effektiven Daten- und Informationsmanagements sowie die Qualifizierung von Fachexperten unerlässlich.

2.2.3 Sensorik 4.0

Wie Jörg Fischer betont, sind Sensoren entscheidend für die Umsetzung von Industrie 4.0, da sie die Verbindung zwischen der digitalen und realen Welt herstellen. Fehlen jedoch die passenden Sensoren, bleiben alle darüber liegenden Systeme zur Dateninterpretation im Dunkeln [35]. Sensoren finden sich Anwendungen in

Verkehrsmitteln wie Zügen, Flugzeugen und Autos, sowie in Maschinen wie Verkaufsautomaten und Geldautomaten, auch in Haushaltsgeräten wie Waschmaschinen und Kühlschränken sowie in Fabriken. Sie erfassen Veränderungen wie Temperatur, Luftfeuchtigkeit und Druck. Sensoren empfangen Signale in Form von elektrischen Impulsen, Wärme, Lichtenergie oder chemischen Reaktionen und wandeln diese in digitale Daten um. Diese Daten werden dann von Verarbeitungseinheiten interpretiert, um entsprechende Reaktionen auszulösen oder Informationen bereitzustellen.

Mit der Integration und Komplexität moderner industrieller Systeme werden Sicherheitsprobleme immer wichtiger. Die fehlererkennungsbasierte Methode auf analytischem Modell hat in den letzten Jahrzehnten herausragende Beiträge geleistet, um den normalen Betrieb des Systems und die Sicherheit von Leben und Eigentum zu gewährleisten. Moderne industrielle Systeme sind mit einer Vielzahl von Sensoren ausgestattet, was die Datenerfassung und Arbeitsprotokolle bequemer und effizienter macht. Die datengesteuerte Fehlererkennungstechnologie kann die Merkmale dieser Daten analysieren und den Systemzustand überwachen. Sie hat umfangreiche Aufmerksamkeit erhalten. Die Identifizierung, ob Prozessfehler mit der Produktqualität zusammenhängen, kann die Produktqualität verbessern und die Betriebsdauer der Ausrüstung verlängern.

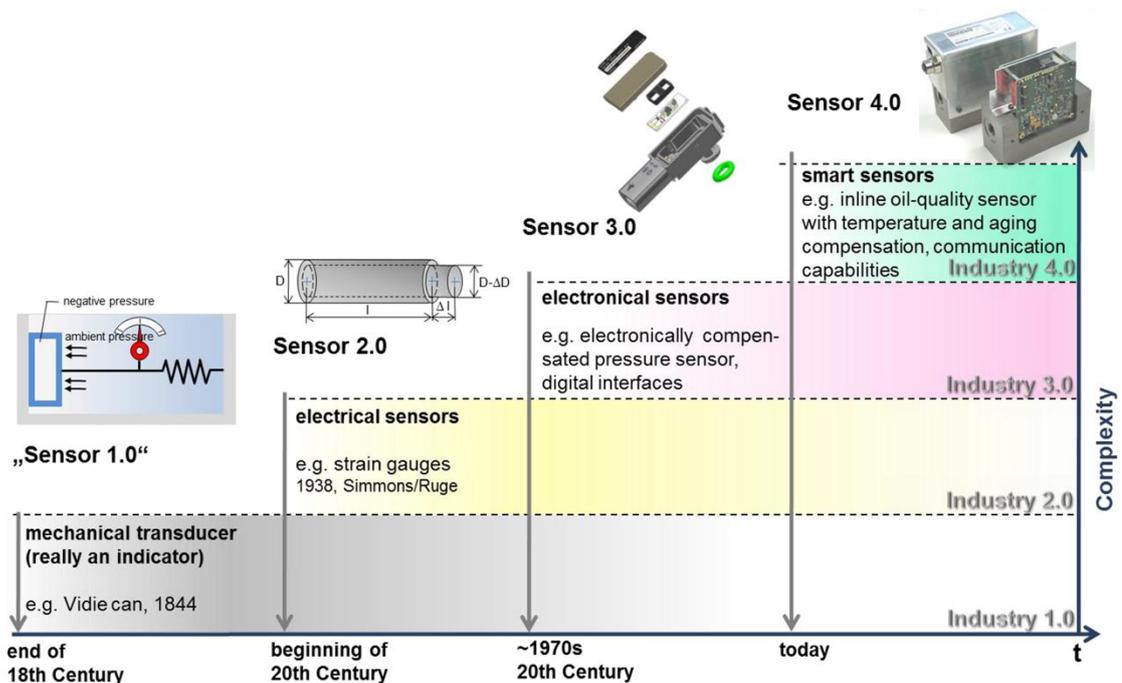


Abbildung 2-5 Historische Entwicklung von „Sensor 1.0“ zu intelligent Sensoren, d.h. „Sensor 4.0“ in Anlehnung an Peter Krause, Vorsitzender von AMA e.V. [40]

Die Sensorik hat eine lange historische Entwicklung durchlaufen, von einfachen mechanischen Sensoren bis hin zu hochentwickelten, vernetzten Systemen der Sensorik 4.0, Smart Sensors genannt (siehe Abbildung 2.5). Frühe Sensoren waren mechanisch und analog, während moderne Sensoren digitale Technologien und künstliche Intelligenz nutzen, um komplexe Daten zu sammeln, zu analysieren und zu interpretieren, was neue Anwendungen und Effizienz ermöglicht.

Die aktuelle Forschung befasst sich mit selbstlernenden Systemen und semantischen Technologien für die Analyse komplexer Systeme [37]. Die Bedeutung von Sensoren

und Messtechnik wird durch Trends wie die Rückverfolgbarkeit einzelner Komponenten und die Bereitstellung von Messungen als Dienstleistung weiter gestärkt.

Die Revolution der Sensoren 4.0 ist geprägt durch ihre integrierte dezentrale Rechenleistung und flexible Programmierung, was eine agilere und effizientere Produktion ermöglicht. Sensoren spielen eine Schlüsselrolle in der Automatisierung von Produktions-, Montage-, Logistikprozessen und Instandhaltung von Maschinen und schaffen transparentere Abläufe in der Industrie 4.0 [13].

2.2.4 Künstliche Intelligenz und maschinelles Lernen

Künstliche Intelligenz (KI) bezeichnet die Fähigkeit von technischen Systemen, Aufgaben zu lösen, die normalerweise menschliche Intelligenz erfordern. Dabei ist es wichtig zu betonen, dass KI nicht direkt mit menschlicher Intelligenz verglichen werden kann, sondern lediglich intelligentes Verhalten eines technischen Systems bedeutet [38]. Der Begriff der KI ist seit dem Dartmouth Summer Research Project on Künstliche Intelligenz im Jahr 1956 bekannt [39].

In den Anfängen der KI-Forschung lag der Fokus auf der Verarbeitung natürlicher Sprache und humanoiden Robotern. Der Turing-Test wurde als Maßstab für KI entwickelt, wobei ein Computer als intelligent gilt, wenn er Antworten liefert, die nicht von menschlichen Antworten zu unterscheiden sind [40]. Allerdings führte die Unmöglichkeit, menschliche Fähigkeiten auf technische Systeme zu übertragen, in den 1970er Jahren zu einem ersten „Winter“ der KI [41].

Maschinelles Lernen (ML) hingegen bietet eine Alternative zur manuellen Erstellung von Regeln und Wissensrepräsentationen für Computer. ML-Algorithmen ermöglichen es, Muster in vorhandenen Datensätzen zu erkennen und zu nutzen, um automatisch allgemeingültige Regeln abzuleiten [42]. Dabei sind maschinelle Lernverfahren unabhängig vom Anwendungsfall einsetzbar.

Ein Beispiel für ML ist die Gruppierung ähnlicher Werkstoffe basierend auf ihren Kenngrößen. ML-Algorithmen können Regeln ableiten, um weitere Werkstoffe der richtigen Gruppe zuzuordnen, basierend auf den vorhandenen formalen Klassifikationen [42]. ML ermöglicht die automatisierte Ableitung von Regeln aus Daten, ohne spezifisches Fachwissen über die Daten zu benötigen.

Maschinelles Lernen ist besonders leistungsfähig bei der Mustererkennung im hochdimensionalen Raum. Moderne Computersysteme können hunderte oder tausende Eigenschaften gleichzeitig berücksichtigen, was ihnen ermöglicht, komplexe Muster zu erkennen [43]. ML kann daher für Aufgaben eingesetzt werden, die für Menschen schwer zu lösen sind, und zielt darauf ab, menschliche Intelligenz zu ergänzen, nicht zu ersetzen [43].

Es ist wichtig zu unterscheiden, dass künstliche neuronale Netze und Deep Learning spezifische Unterkategorien von ML sind. Obwohl häufig synonym verwendet, sind sie nicht dasselbe wie KI [43]. Klarstellungen über die Nutzung von Begriffen wie künstliche neuronale Netze und Deep Learning sind in der Praxis entscheidend, um Missverständnisse zu vermeiden [43].

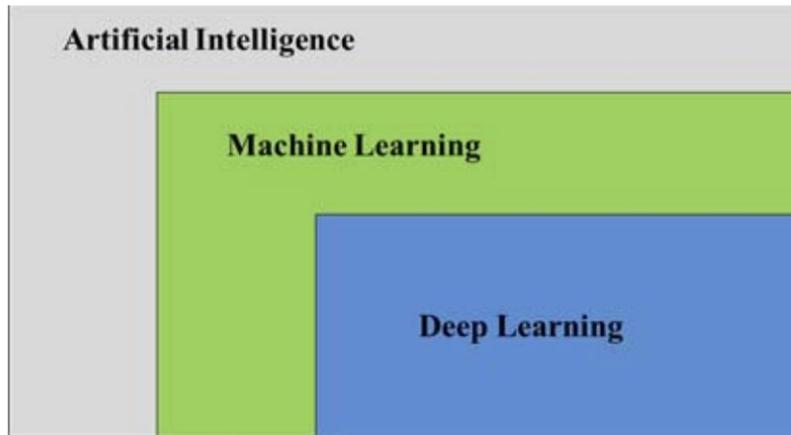


Abbildung 2-6: Die Struktur der Lernverfahren (Maschinelles Lernen ist ein Teilbereich der Künstlichen Intelligenz) [44]

Im Bereich des maschinellen Lernens werden drei Hauptarten unterschieden (siehe Abbildung 2.6). Beim überwachten Lernen werden sowohl die Prädiktoren als auch die Antwortvariablen bekannt, um das Modell zu erstellen. Im Gegensatz dazu sind beim unüberwachten Lernen nur die Antwortvariablen bekannt, während die Prädiktoren unbekannt sind. Beim bestärkenden Lernen lernt der Agent durch Interaktion mit der Umgebung, indem er Aktionen ausführt und die daraus resultierenden Konsequenzen beobachtet. Diese Klassifizierung ermöglicht es, verschiedene Lernparadigmen zu verstehen und anzuwenden [45].

In der Praxis erfordert ML oft große Datensätze, die als „Big Data“ bezeichnet werden. Diese werden verwendet, um Modelle zu trainieren und Muster zu erkennen. ML und KI haben das Potenzial, in verschiedenen Anwendungsbereichen eingesetzt zu werden, erfordern jedoch ein klares Verständnis der Begriffe und ihrer Anwendungen [43].

2.3 Sicherheits- und Zuverlässigkeitsaspekte in der Industrie 4.0

2.3.1 Zuverlässigkeit in der Industrie 4.0

Zuverlässigkeits- und Qualitätsmanagement sind zwei grundlegende Säulen, die die Leistungsfähigkeit und Sicherheit von Produkten und Dienstleistungen gewährleisten. Eine effektive Implementierung dieser Konzepte ist unabdingbar, um den Ansprüchen der Kunden und den Industriestandards gerecht zu werden.

Auf Basis der Definitionen nach Messerschmidt-Bölkow-Blohm GmbH (Bitter 1986) und DIN 40041 (1990), Zuverlässigkeit ist die Wahrscheinlichkeit dafür, dass eine Einheit während eines definierten Zeitintervalls oder eines Intervalls bezogen auf eine zeitkorrelierende Lebensdauervariable unter gegebenen Funktions- und Umgebungsbedingungen hinsichtlich ihrer funktions- und sicherheitskritischen Merkmale nicht ausfällt [46]. Und Qualität wird laut der Norm DIN EN ISO 9000:2015-11 als „Grad, in dem ein Satz inhärenter Merkmale eines Objekts Anforderungen erfüllt“ definiert [47].

Das Zuverlässigkeitsmanagement zielt darauf ab, die Fähigkeit eines Systems oder Produkts sicherzustellen, seine Funktionen unter definierten Bedingungen und für einen festgelegten Zeitraum ohne Ausfälle oder Störungen auszuführen [48]. Dies umfasst die Fehleranalyse, die Vorhersage von Ausfallwahrscheinlichkeiten und die Entwicklung von Strategien zur Fehlervermeidung und -behebung.

Im Gegensatz dazu befasst sich das Qualitätsmanagement mit der Sicherstellung, dass Produkte und Dienstleistungen den Anforderungen laut ISO 9001:2015, Spezifikationen und Erwartungen der Kunden entsprechen. Dies erfordert die Implementierung von Qualitätskontrollverfahren, die Überwachung von Qualitätskennzahlen und die kontinuierliche Verbesserung der Prozesse.

Die Integration von Zuverlässigkeits- und Qualitätsmanagement in Unternehmen trägt nicht nur zur Minimierung von Risiken und Kosten bei, sondern steigert auch die Kundenzufriedenheit und stärkt die Wettbewerbsfähigkeit [49].

Angesichts der zunehmenden Bedeutung eines sicheren, schnellen und nachhaltigen Informationsaustauschs in der Industrie 4.0 ist es von entscheidender Bedeutung, dass Zuverlässigkeitskonzepte für IT-Systeme, Cloud-Services und industrielle IT miteinander kompatibel und integrierbar sind. In vielen Fällen sind heutige Produktionssysteme unzureichend oder gar nicht auf Zuverlässigkeitsanalyse ausgelegt. Daher ist es erforderlich, dass Zuverlässigkeitsmethoden und -systeme neu entwickelt oder überarbeitet werden, um den spezifischen Anforderungen der industriellen Umgebung gerecht zu werden. Diese Anforderungen umfassen das Prozessumfeld, das Echtzeitverhalten, die Zuverlässigkeit und die Langlebigkeit [13]. Es ist von entscheidender Bedeutung, dass Zuverlässigkeitsaspekte in den Fokus gerückt werden, um sicherzustellen, dass Produktionsprozesse effizient, stabil und zuverlässig ablaufen.

Im Jahr 2016 hat Columbus in einer Untersuchung festgestellt, dass die prädiktive und präskriptive Instandhaltung von Maschinen als die wichtigste Anwendung der industriellen Datenanalyse identifiziert wurde. Aufgrund der Bedeutung der Zuverlässigkeit als zentrale Messgröße für die Beurteilung von Instandhaltungsplänen, -strategien und -verfahren gewinnt eine präzise Zuverlässigkeitsanalyse und -bewertung an Bedeutung [50]. Es ist jedoch wichtig anzumerken, dass Entscheidungen, die auf ungenauen Modellen basieren, sich häufig negativ auf die Leistung eines Systems auswirken und sowohl seine Sicherheit als auch seine Zuverlässigkeit beeinträchtigen können. Deshalb spielt die Ausbildung von Personal eine wichtige Rolle bei den Entscheidungen und der Verwirklichung der Korrekten Interpretierung von Software gestützter Analyse.

Die Zuverlässigkeit von Produktionsanlagen, die als sogenannte cyber-physische Systeme fungieren, wird maßgeblich von drei Hauptfaktoren beeinflusst: der Software-Zuverlässigkeit, der Hardware-Zuverlässigkeit und der Zuverlässigkeit in Bezug auf menschliche Aktivitäten. Jeder dieser Faktoren trägt auf unterschiedliche Weise zur Gesamtzuverlässigkeit bei und erfordert eine separate Analyse, bevor sie zusammengefasst werden können, um ein einheitliches Maß für die Gesamtzuverlässigkeit zu ermitteln.

Die Hardware-Zuverlässigkeit betrifft die Fähigkeit der physischen Komponenten der Anlage, ihre Funktionen über einen bestimmten Zeitraum hinweg beizubehalten, ohne Ausfälle oder Störungen zu erleiden. Hier spielen Faktoren wie Materialqualität, Herstellungsprozesse und Umweltbedingungen eine Rolle [51]. Im Laufe der Zeit zeigt die Hardware-Ausfallcharakteristiken, die als Badewannenkurve bekannt sind. Die

Perioden A, B und C in Abbildung 2.7 stehen für die Einbrennphase, die Nutzungsdauerphase und die End-of-Life-Phase.

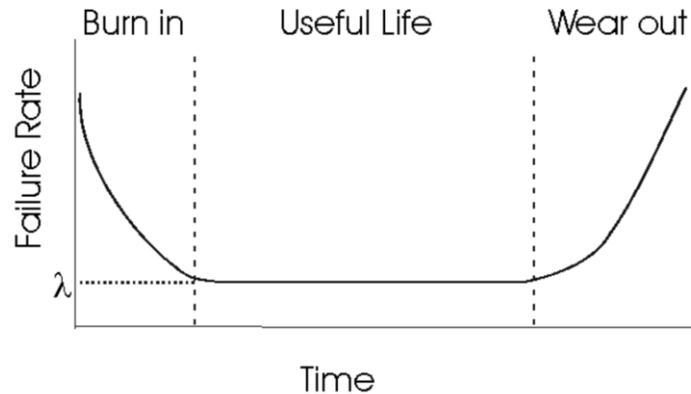


Abbildung 2-7: Badewannenkurve für Hardware-Verlässlichkeit [52]

Die Software-Zuverlässigkeit bezieht sich auf die Fähigkeit der Software, ihre Funktionen unter verschiedenen Bedingungen ohne Ausfälle oder Störungen auszuführen. Sie wird durch verschiedene Faktoren wie Programmierung, Design und Fehlerbehebung beeinflusst [53]. Die Softwarezuverlässigkeit zeigt jedoch nicht die gleichen Merkmale wie Hardware. Eine mögliche Kurve wird in Abbildung 2.8 gezeigt, wenn wir die Softwarezuverlässigkeit auf die gleichen Achsen projizieren. Es gibt zwei wesentliche Unterschiede zwischen Hardware- und Softwarekurven. Ein Unterschied besteht darin, dass die Software in der letzten Phase keine zunehmende Ausfallrate wie die Hardware aufweist. In dieser Phase nähert sich die Software der Veraltbarkeit; es gibt keine Motivation für Upgrades oder Änderungen an der Software. Daher wird sich die Ausfallrate nicht ändern. Der zweite Unterschied besteht darin, dass die Software während der Nutzungsdauerphase bei jedem Upgrade einen drastischen Anstieg der Ausfallrate erleben wird. Die Ausfallrate nivelliert sich allmählich ab, teilweise aufgrund der nach den Upgrades gefundenen und behobenen Mängel.

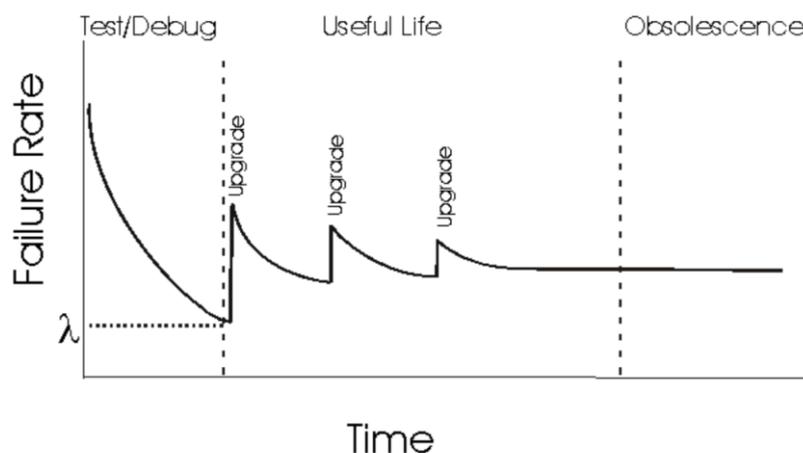


Abbildung 2-8: Überarbeitete Badewannenkurve für Software-Zuverlässigkeit [52]

Die Tabelle zeigt die unterschiedlichen Merkmale von Hardware- und Softwarezuverlässigkeit.

Tabelle 2-1: Die Unterschiede zwischen Software- und Hardwarezuverlässigkeit [54]

Software-Zuverlässigkeit	Hardware- Zuverlässigkeit
Ohne die Berücksichtigung der Programmentwicklung ist die Ausfallrate statistisch nicht abnehmend.	Die Ausfallrate weist eine Badewannenkurve auf. Der Burn-in-Zustand ähnelt dem Software-Debugging-Zustand.
Ein Ausfall tritt nie auf, wenn die Software nicht verwendet wird.	Materielle Verschlechterung kann Ausfälle verursachen, auch wenn das System nicht verwendet wird.
Die meisten Modelle werden analytisch aus Annahmen abgeleitet. Der Schwerpunkt liegt auf der Entwicklung des Modells, der Interpretation der Modellannahmen und der physischen Bedeutung der Parameter.	Einige Verteilungen werden an Ausfalldaten angepasst. Die Auswahl der zugrunde liegenden Verteilung basiert auf der Analyse von Ausfalldaten und Erfahrungen. Der Schwerpunkt liegt auf der Analyse von Ausfalldaten.
Falsche Logik, falsche Aussagen oder falsche Eingabedaten verursachen Ausfälle. Dies ähnelt den Designfehlern eines komplexen Hardware-Systems.	Materielle Verschlechterung, zufällige Ausfälle, Designfehler, Fehlgebrauch und Umgebung sind die Ursachen für Ausfälle.
Eine Erhöhung des Testaufwands und die Korrektur erkannter Fehler können die Softwarezuverlässigkeit verbessern. Die Zuverlässigkeit ändert sich während des Tests kontinuierlich aufgrund von Problemen im neuen Code oder der Behebung von Fehlern durch Fehlerbehebung.	Besseres Design, besseres Material, Anwendung von Redundanz und beschleunigte Lebensdauerprüfung können die Hardwarezuverlässigkeit verbessern.
Software-Reparaturen führen zur Entwicklung eines neuen Softwareteils.	Hardware-Reparaturen stellen den ursprünglichen Zustand wieder her.
Softwareausfälle werden selten von Warnungen begleitet.	Hardwareausfälle werden normalerweise von Warnungen begleitet.
Softwarekomponenten wurden selten standardisiert.	Hardwarekomponenten können standardisiert werden.
Software erfordert im Wesentlichen unendliche Tests.	Hardware kann in der Regel umfassend getestet werden.

Die Zuverlässigkeit in Bezug auf menschliche Aktivitäten bezieht sich auf die Konsistenz und Genauigkeit der menschlichen Interaktionen mit der Anlage, einschließlich Bedienungsfehler, Wartung und Fehlerbehebung [55].

Durch die Analyse und Berücksichtigung dieser drei Faktoren können Unternehmen ein umfassendes Verständnis der Gesamtzuverlässigkeit ihrer Produktionsanlagen entwickeln und geeignete Maßnahmen zur Verbesserung ergreifen.

2.3.2 Betriebssicherheit und funktionale Sicherheit

Die Betriebssicherheitsverordnung (BetrSichV) ist ein zentraler Bestandteil der betrieblichen Sicherheitsstandards und gilt für die Verwendung von Arbeitsmitteln sowie den Betrieb überwachungsbedürftiger Anlagen, da sie die Sicherheitsanforderungen und Prüfverfahren für Arbeitsmittel und Anlagen festlegt, um Unfälle und Gefährdungen zu minimieren. Gemäß § 15 der BetrSichV muss der Arbeitgeber vor der erstmaligen Inbetriebnahme und nach prüfpflichtigen Änderungen überwachungsbedürftige Anlagen auf ihre Einhaltung der Verordnung und ihren sicheren Zustand prüfen [56]. Zu den überwachungsbedürftigen Anlagen gehören Aufzugsanlagen, Anlagen in explosionsgefährdeten Bereichen, Druckanlagen, Krane, Flüssiggasanlagen und maschinentechnische Arbeitsmittel der Veranstaltungstechnik. Die BetrSichV zielt darauf ab, den Arbeitsschutz für Beschäftigte zu verbessern und Dritte beim Betrieb solcher Anlagen zu schützen.

Die funktionale Sicherheit eines Systems steht in direktem Zusammenhang mit seiner Betriebssicherheit, da sie entscheidend dazu beiträgt, potenzielle Risiken während des Betriebs zu minimieren. Gemäß der ISO 26262 definiert funktionale Sicherheit die Fähigkeit eines Systems, Risiken zu reduzieren und Sicherheit zu gewährleisten, während es bestimmungsgemäß arbeitet [60]. Die Norm legt Standards für die Entwicklung und den Betrieb sicherheitskritischer Systeme fest. Die IEC 61508 ist eine weitere maßgebliche Norm, die sich mit der funktionalen Sicherheit von elektrischen/elektronischen/programmierbaren elektronischen sicherheitsrelevanten Systemen befasst und Richtlinien für deren sicherheitsgerichtete Entwicklung bereitstellt. Diese Standards spielen eine entscheidende Rolle bei der Gestaltung von Systemen, die robust, zuverlässig und fehlersicher sind, was wiederum die Betriebssicherheit gewährleistet.

Die Systemanforderungen umfassen verschiedene funktionale, nichtfunktionale und sicherheitstechnische Aspekte, die zusammen mit den allgemeinen Systemcharakteristiken spezifische Eigenschaften definieren. Diese umfassen: Zuverlässigkeit, Verfügbarkeit, fehlersicheren Betrieb, Systemintegrität, Datenintegrität, Systemwiederherstellung, Wartbarkeit und Verlässlichkeit. [58]

Die IEC 61508 definiert auch die erforderlichen Tätigkeiten, identifiziert Verantwortlichkeiten in verschiedenen Phasen und legt das erforderliche Fachwissen und die Kompetenz für jede Phase des Lebenszyklus fest [59].

Der Standard bietet Unterstützung bei der Bestimmung von Sicherheitsintegritätsstufen (SIL), die in vier Stufen unterteilt sind: SIL1, SIL2, SIL3 und SIL4. Mit jedem höheren SIL steigt das Ausfallrisiko. Eine Risikobewertung wird durchgeführt, um den erforderlichen SIL zu bestimmen. Die Festlegung eines SIL erfolgt durch die Untersuchung der systematischen Fähigkeit, der Architektur-Einschränkungen und der Wahrscheinlichkeit eines gefährlichen Ausfalls. Die untenstehende Tabelle zeigt die SIL-Klassifizierung entweder nach Ausfallwahrscheinlichkeit auf Anforderung oder nach Ausfallwahrscheinlichkeit pro Stunde [60].

Bei SILs in Bezug auf IEC 61508 unterscheiden sie sich von den Sicherheitsintegritätsstufen in anderen Standards, und es sollte darauf geachtet werden, sie nicht zu verwechseln.

Tabelle 2-2: SIL niedrige und höhere Anforderungsrate in Anlehnung zu IEC/EN 61508 [61]

SIL	Betriebsart mit niedriger Anforderungsrate (Low Demand Mode)	Betriebsart mit hoher Anforderungsrate (High Demand Mode)
	PFD (mittlere Ausfallwahrscheinlichkeit der Sicherheitsfunktion bei Anforderung)	PFH (Wahrscheinlichkeit eines gefahrbringenden Ausfalls pro Stunde)
1	$\geq 10^{-2}$ bis $< 10^{-1}$	$\geq 10^{-6}$ bis $< 10^{-5}$
2	$\geq 10^{-3}$ bis $< 10^{-2}$	$\geq 10^{-7}$ bis $< 10^{-6}$
3	$\geq 10^{-4}$ bis $< 10^{-3}$	$\geq 10^{-8}$ bis $< 10^{-7}$
4	$\geq 10^{-5}$ bis $< 10^{-4}$	$\geq 10^{-9}$ bis $< 10^{-8}$

Sicherheitslebenszyklus

Zur systematischen Bewältigung funktionaler Sicherheitsprobleme wird gemäß IEC 61508 ein Sicherheitslebenszyklus definiert (Siehe Abbildung 2.9). Die Anwendung dieses Lebenszyklus führt zur Erreichung des Sicherheitsintegritätslevels (SIL). Innerhalb des Sicherheitslebenszyklus sind drei Maßnahmen zur Risikominimierung vorgesehen, nämlich die Phasen 9, 10 und 11 [62].

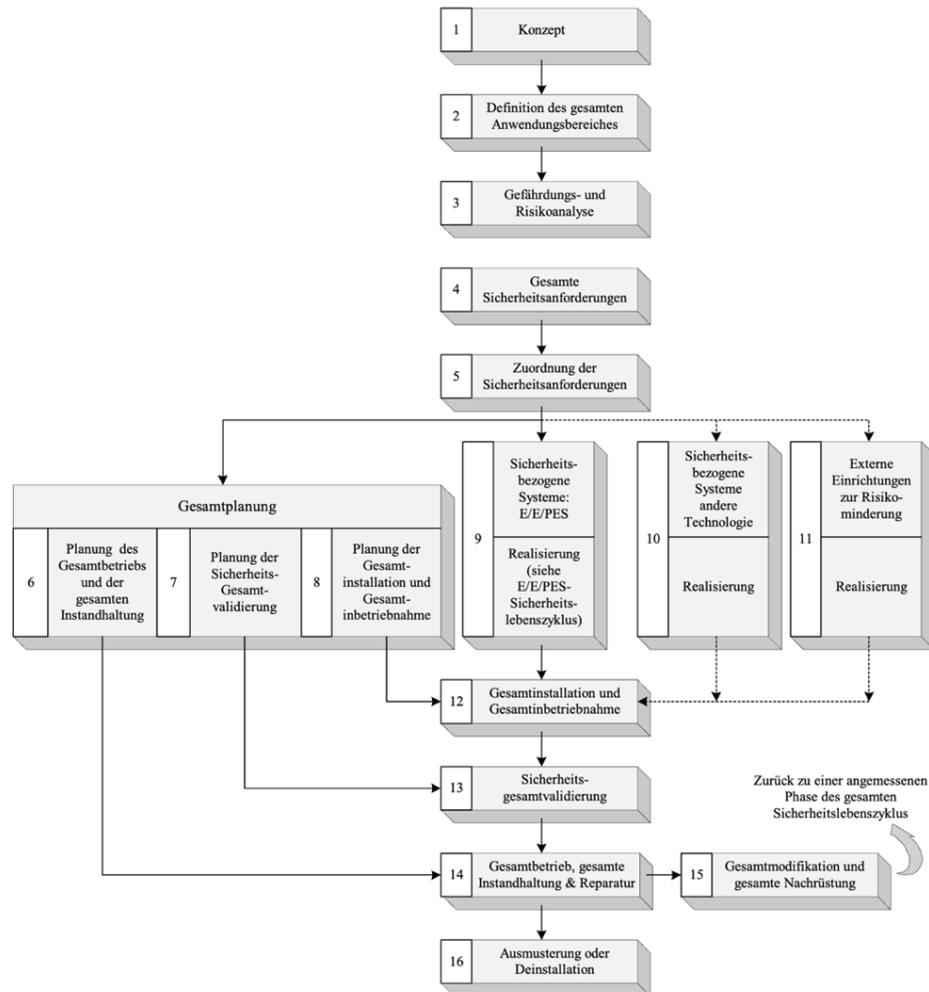


Abbildung 2-9: Gesamter Sicherheitslebenszyklus [61]

Verifikation

Zur Durchführung der Verifikation ist es notwendig, dass ein Verifikationsplan erstellt wird, der klare Kriterien und Methoden zur Bewertung enthält. Dieser Plan muss für jede Phase des Sicherheitslebenszyklus nominale Kriterien und spezifische Verfahren vorsehen, um eine gründliche Überprüfung zu ermöglichen [61].

Beurteilung der funktionalen Sicherheit

Zur Bewertung der funktionalen Sicherheit des elektrischen/elektronischen/programmierbaren elektronischen Systems (E/E/PES) gemäß den Vorgaben der IEC 61508 ist es erforderlich, Aspekte wie die Unabhängigkeit und Kompetenz der beteiligten Personen zu berücksichtigen [61].

2.4 Methoden der Instandhaltung

Die Instandhaltung spielt eine entscheidende Rolle für die Leistungsfähigkeit und Langlebigkeit von industriellen Anlagen. In der modernen Industrie wird sie durch eine Vielzahl an Methoden realisiert, die von grundlegenden geplanten Korrektiven Wartungen bis hin zu präventiven und prädiktiven Wartungstechniken reichen. Jede dieser Methoden zielt darauf ab, Ausfallzeiten zu minimieren und die Anlageneffizienz zu maximieren. Während Schalt- und redundante Konfigurationen auf

Systemsicherheit durch Duplizierung kritischer Komponenten fokussieren, konzentriert sich die präventive Instandhaltung auf vorbeugende Maßnahmen basierend auf festgelegten Intervallen. Die prädiktive Instandhaltung hingegen nutzt fortschrittliche Datenanalyse und Echtzeitüberwachung, um den optimalen Wartungszeitpunkt zu bestimmen und somit die Betriebsbereitschaft der Anlagen proaktiv zu gewährleisten. Insgesamt bilden diese Methoden das Fundament für eine effektive Instandhaltungsstrategie in der Industrie 4.0.

Instandhaltung gemäß EN 13306, Abschn. 2.1 und DIN 31051, Abschn. 4.1.1 ist die Kombination aller technischen und administrativen Maßnahmen sowie Maßnahmen des Managements während des Lebenszyklus einer Betrachtungseinheit zur Erhaltung des funktionsfähigen Zustandes oder der Rückführung in diesen, so dass sie die geforderte Funktion erfüllen kann [62].

Instandhaltungsstrategien (siehe Abbildung) sind von großer Bedeutung in der Umsetzung dieser Definition, indem sie den Rahmen für die Planung und Durchführung von Instandhaltungsmaßnahmen festlegen.



Abbildung 2-10: Strukturierung von Instandhaltungsstrategien in Anlehnung an DIN EN 13306 [63]

2.4.1 Korrektive Instandhaltung

Die korrektive Instandhaltung, auch als Instandhaltung nach Ausfall bekannt, erfolgt grundsätzlich reaktiv und setzt auf die Behandlung von Ausfallsituationen. Dabei ist stets mit potenziellen Ausfällen zu rechnen, wobei die Anwendung der Methode von der Häufigkeit und Schwere der Ausfälle abhängt. Während sie in einigen Branchen akzeptabel ist, wird sie in anderen, wie beispielsweise in Kraftwerken, Anlagen der Energieerzeugung wie in der Luft- und Raumfahrt oder in kritischen medizinischen Systemen, möglichst vermieden. Diese Strategie sollte nur dann verfolgt werden, wenn redundante Systeme vorhanden sind. Besonders bei kritischen Anlagen wie Kompressoren ist die Implementierung von Schalt- und redundanten Konfigurationen von entscheidender Bedeutung. Bei denen Ausfallzeiten extrem kostspielig oder gefährlich sind, Die Kosten für zusätzliche Ausrüstung können gerechtfertigt sein, wenn die Folgen eines Ausfalls gravierend sind.

Ein wesentlicher Ansatz hierbei ist die Nutzung von Parallelsystemen, wie beispielsweise die Implementierung eines Notkompressors, der automatisch einspringt, sollte der Hauptkompressor ausfallen. Dieses Prinzip erhöht die Zuverlässigkeit des Gesamtsystems erheblich, da bei einem Versagen einer Komponente sofort ein Ersatz zur Verfügung steht und die Produktion ohne nennenswerte Unterbrechungen weiterlaufen kann.

Es wird auch empfohlen, diese Strategie auf Anlagen oder Komponenten zu beschränken, die im Unternehmen von untergeordneter Bedeutung sind und deren Ausfall keine sicherheitstechnischen Anforderungen beeinträchtigt oder zu Produktionsunterbrechungen führt.

2.4.2 Periodische Instandhaltung

Die periodische präventive Instandhaltungsstrategie beinhaltet den Austausch von Systemkomponenten gemäß definierten Nutzungsintervallen, unabhängig vom aktuellen Zustand der Bauteile. Diese Intervalle können zeit- oder ereignisbasiert sein, basierend auf Kriterien wie Kalenderzeit, Betriebsstunden oder gefahrenen Kilometern. Diese Strategie empfiehlt sich besonders in Branchen, in denen Sicherheits- und Umweltauflagen eine regelmäßige Wartung erfordern oder wenn die Lebensdauer von Komponenten gut bekannt ist [64].

Die periodische präventive Wartung ermöglicht eine bessere Planbarkeit der Instandhaltungsmaßnahmen, da sie in Zeiten ohne Produktion durchgeführt werden kann. Durch wiederholte und planbare Abläufe wird die Ausfallzeit der Anlagen minimiert. Allerdings besteht das Risiko, dass Komponenten zu früh ausgetauscht werden, was zu einer Verschwendung von Verschleißreserven und einem erhöhten Verbrauch von Ersatzteilen führt.

Die Bestimmung des idealen Austauschzeitpunkts für Komponenten ist eine Herausforderung und erfordert die Berücksichtigung des unterschiedlichen Ausfallverhaltens der Komponenten eines Systems. Die vorbeugende Instandhaltung zielt darauf ab, verschleißanfällige Teile rechtzeitig auszutauschen, um die Leistungsfähigkeit und Zuverlässigkeit der Anlagen zu erhöhen und ungeplante Wartungskosten zu reduzieren [65].

2.4.3 Präventive Instandhaltung

Präventive Instandhaltung ist eine vorbeugende Strategie, die darauf abzielt, Ausfälle und die damit verbundenen Stillstandszeiten sowie Kosten zu reduzieren, indem Anlagen und Maschinen proaktiv gewartet oder ersetzt werden, bevor ein Defekt auftritt. Bei der präventiven Instandhaltung werden Kompressorkomponenten nach einer festgelegten Betriebsdauer ersetzt, unabhängig davon, ob ein tatsächlicher Ausfall vorliegt oder nicht.

Diese präventive Ersatzzeit wird so gewählt, dass sie in der Regel vor der statistischen Ausfallzeit, also der durchschnittlichen Lebensdauer der Kompressorkomponente, liegt. Das Problem der optimalen Auswahl des Instandhaltungszyklus lässt sich mit dem kausalen Zusammenhang von diesem mit der Überlebenswahrscheinlichkeit darstellen (Abbildung 2.11). Die beiden Fälle A und B repräsentieren unterschiedliche Szenarien. Es ist ratsam, Maßnahmen rechtzeitig zu ergreifen (Fall A), um ein weiteres Absinken der Überlebenswahrscheinlichkeit wie im Fall B zu verhindern [63].

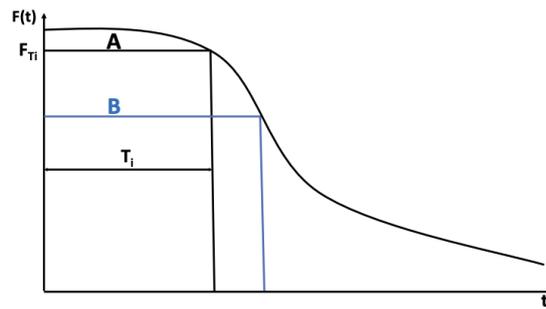


Abbildung 2-11: Auswahl der Dauer des Instandhaltungszyklus [63]

T_i : Dauer eines Instandhaltungszyklus unter Beachtung einer ausreichenden Überlebenswahrscheinlichkeit

F_{T_i} : Überlebenswahrscheinlichkeit zum Zeitpunkt des Einsetzens der planmäßigen Instandhaltung

Die Idee dabei ist, potenzielle Ausfälle und die höheren Kosten, die durch korrektive Maßnahmen nach einem Ausfall entstehen würden, zu vermeiden. Diese korrektiven Maßnahmen sind oftmals teurer als präventive Maßnahmen, da sie ungeplante Stillstände, Notfallreparaturen und möglicherweise Schäden an weiteren Systemkomponenten umfassen können.

Die präventive Instandhaltungspolitik führt zu einer neuen Betriebsdauer und Kostenstruktur für den Kompressor. Die neue Betriebsdauer entspricht der Zeitspanne, in der der Kompressor störungsfrei läuft, bis zum Zeitpunkt, an dem er präventiv ersetzt wird. Die damit verbundenen Kosten setzen sich aus den präventiven Wartungs- oder Ersatzkosten und etwaigen Betriebskosten zusammen, die während der Betriebsdauer anfallen.

Präventive Instandhaltung eignet sich für Ausrüstungen, bei denen Verschleiß vorhersehbar ist und regelmäßige Wartung die Lebensdauer verlängert oder kostspielige Ausfälle verhindert. Präventive Instandhaltung ist sinnvoll, wenn die Kosten für regelmäßige Wartung niedriger sind als die potenziellen Kosten eines unerwarteten Ausfalls.

Zusammengefasst versucht die präventive Instandhaltung, die Gesamtbetriebskosten über die Lebensdauer einer Anlage zu optimieren, indem sie geplante, vorbeugende Maßnahmen anstatt reaktiver Reparaturen nach einem Ausfall einsetzt. Durch diese Strategie sollen Zuverlässigkeit und Verfügbarkeit der Anlage erhöht und gleichzeitig die Instandhaltungskosten gesenkt werden [2].

2.4.4 Prädiktive Instandhaltung

Die prädiktive Instandhaltung eines Kompressors beruht auf der frühzeitigen Erkennung von Veränderungen im Betriebszustand des Geräts, bevor es zu einem vollständigen Ausfall kommt. Diese Art der Instandhaltung nutzt Condition-Monitoring-Techniken, bei denen physikalische Größen wie Schwingungen, Temperaturen und Geräusche regelmäßig gemessen und analysiert werden. So können zum Beispiel ansteigende Vibrationen auf ein Lagerproblem hinweisen, während ungewöhnliche Geräusche auf Probleme mit den Rotoren oder anderen beweglichen Teilen hindeuten könnten.

In den letzten Jahren hat die Integration von Technologien wie Big Data, Industrie 4.0, Sensorik 4.0 und Digitalisierung eine entscheidende Rolle in der prädiktiven Instandhaltung gespielt. Durch die Sammlung und Analyse großer Datenmengen aus verschiedenen Sensoren und anderen Informationsquellen ist es möglich, Muster zu erkennen und vorherzusagen, die auf zukünftige Probleme des Kompressors hinweisen. Diese umfassende und genaue Überwachung und Analyse des Zustands des Kompressors ist ein wesentlicher Bestandteil der prädiktiven Wartungsstrategie.

Zur Zustandsklassifizierung werden fortschrittliche KI-Methoden und maschinelles Lernen eingesetzt, darunter Clustering-Techniken, Hauptkomponentenanalyse, Hidden-Markov-Modelle und neuronale Netze. Diese Methoden sind in der Lage, komplexe Muster in den Daten zu erkennen und Vorhersagen zu treffen, die weit über das hinausgehen, was mit traditionellen analytischen Methoden möglich ist. Dadurch kann die prädiktive Instandhaltung nicht nur potenzielle Probleme frühzeitig erkennen, sondern auch Ausfallzeiten reduzieren, die Betriebseffizienz steigern und letztendlich zu Kosteneinsparungen führen [66].

Dieser Ansatz ist ideal für komplexe Systeme, bei denen Ausfallmuster schwer vorherzusagen sind und wo Echtzeit-Daten zur Vorhersage von Problemen vor ihrem Auftreten genutzt werden können. Sie ist besonders nützlich, wenn Ausrüstungen teuer sind und Ausfallzeiten erhebliche Auswirkungen haben. Prädiktive Instandhaltung ist vorteilhaft, wenn ausreichende Daten und analytische Ressourcen verfügbar sind, um genaue Vorhersagen zu treffen [2].

2.4.5 Vergleich der Instandhaltungsmethoden

Zusammenfassend ist die Wahl der Instandhaltungsstrategie stark von den spezifischen Umständen und Anforderungen des jeweiligen Betriebs abhängig. Während redundante Instandhaltung für kritische Systeme mit hohen Ausfallkosten ideal ist, eignet sich präventive Instandhaltung für weniger komplexe Systeme mit vorhersehbarem Verschleiß. Prädiktive Instandhaltung wiederum bietet die fortschrittlichste Lösung für komplexe und teure Ausrüstungen, bei denen die Verfügbarkeit von Daten und Analysetechnologien genutzt werden kann, um Ausfallzeiten zu minimieren und die Effizienz zu maximieren [2].

2.5 Kompressoren und Datenanalyse-Software: eine integrierte Perspektive.

2.5.1 Kompressoren

Der in Abbildung 2-12 dargestellte Kompressor ist ein typisches Gerät zur Erzeugung von Druckluft und besteht aus mehreren Hauptkomponenten, die harmonisch zusammenarbeiten. An der Basis befindet sich ein großer Druckluftbehälter, dessen Aufgabe es ist, die komprimierte Luft zu speichern. Oberhalb des Behälters ist der Antrieb oder Motor positioniert, der als Herzstück des Geräts dient, indem er die nötige Energie für den Kompressionsvorgang liefert.

Auf der einen Seite des Motors befindet sich der Luftfilter und der Einlass, der für die Reinigung der angesaugten Luft sorgt, bevor sie komprimiert wird. Diese gereinigte Luft wird dann in den Zylinder mit Kolben geleitet, der für die eigentliche Kompression der

Luft zuständig ist. Auf der gegenüberliegenden Seite des Zylinders ist die Ventilplatte angebracht, die den Luftfluss innerhalb des Kompressors regelt.

Ein Kurbelgehäuse verbindet den Motor mit dem Zylinder und überträgt die rotierende Bewegung des Motors in eine hin- und hergehende Bewegung des Kolbens. Am Kurbelgehäuse befindet sich auch das Manometer mit Druckregler, das es dem Benutzer ermöglicht, den Druck zu überwachen und einzustellen, um eine sichere und effiziente Funktion des Kompressors zu gewährleisten.

Für den Transport ist der Kompressor mit Transporträdern ausgestattet, die es ermöglichen, das Gerät leicht von einem Ort zum anderen zu bewegen. Ein Griff, der sich an der Oberseite oder Seite des Kompressors befinden kann, unterstützt diesen Vorgang. Für die Entfernung von Kondenswasser, das sich im Laufe der Zeit im Behälter ansammeln kann, gibt es einen Kondenswasser-Ablauf am unteren Teil des Behälters. Dieser ist wichtig, um Korrosion und Schäden am Behälter zu verhindern.

Der Anschluss für den Schlauch, über den die komprimierte Luft zu verschiedenen Werkzeugen und Anwendungen geleitet wird, ist üblicherweise an einer gut zugänglichen Stelle positioniert. Nicht zuletzt sorgt ein stabiler Standfuß dafür, dass der Kompressor auch bei Betrieb sicher steht.

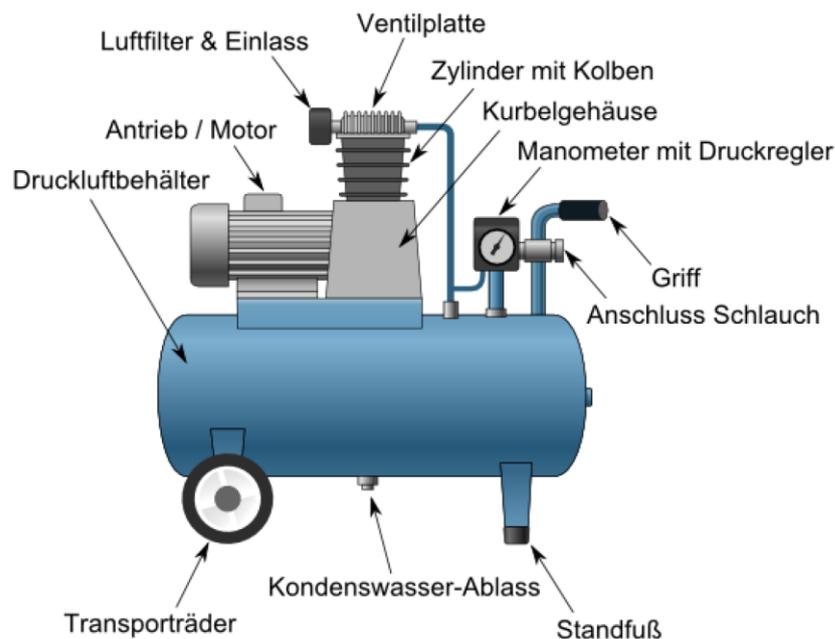


Abbildung 2-12 - Aufbau eines Kompressors [67]

In Kompressorsystemen sind bestimmte Komponenten aufgrund ihrer Funktionen und der damit verbundenen Belastungen anfälliger für Ausfälle. Wälzlager, Wasserpumpen, Lüfter, Auslassventile und Elektromotoren zählen zu den Elementen, die meist Defekte aufweisen.

Wälzlager sind entscheidend für die reibungsarme Drehbewegung von Wellen und rotierenden Teilen. Sie können durch übermäßige Belastung, unzureichende Schmierung oder das Eindringen von Schmutzpartikeln ausfallen. Verschleißerscheinungen oder Korrosion führen schließlich dazu, dass die Laufbahnoberflächen oder Wälzkörper der Lager eine geometrische Abweichung

aufweisen, was zu einem erhöhten Energiebedarf führt oder im schlimmsten Fall zu einem Totalausfall des Lagers.

Wasserpumpen sind für die Kühlung des Systems zuständig. Ihre Ausfälle sind oft auf Verschleiß der Dichtungen oder des Pumpenrades zurückzuführen. Leckagen oder eine verminderte Pumpenleistung können zur Überhitzung führen und somit den Kompressor gefährden.

Lüfter dienen zur Aufrechterhaltung einer angemessenen Betriebstemperatur. Sie können durch mechanische Schäden, Verschleiß oder Blockaden durch Fremdkörper beeinträchtigt werden. Ein defekter Lüfter führt häufig zu einer unzureichenden Kühlung und damit zu einer Überhitzung des Systems.

Auslassventile regulieren den Luftstrom aus dem Kompressor. Sie sind hohen Druckunterschieden ausgesetzt und können durch Materialermüdung oder Schmutzansammlungen beschädigt werden. Ein Versagen dieser Ventile kann zu einem unkontrollierten Druckanstieg oder zu Leistungsverlust führen.

Elektromotoren sind das Herzstück des Kompressors und anfällig für Überlastung, elektrische Störungen oder Wärmeentwicklung. Probleme mit der Isolierung, den Wicklungen oder den Anschlüssen können zu einem Motordefekt führen, was häufig eine umfangreiche Reparatur oder einen Austausch erforderlich macht.

Um Ausfälle zu vermeiden, ist ein regelmäßiges Wartungsprogramm, das Inspektionen, Reinigung und den Austausch von Verschleißteilen umfasst, unerlässlich. Ein effektives Condition-Monitoring-System kann dabei helfen, die Frühzeichen eines drohenden Ausfalls zu erkennen und präventive Maßnahmen einzuleiten.

2.5.2 Datensatz und angewendete Programmiersprache Datensatz

Das vorliegende Dataset (siehe Abbildung) umfasst eine breite Palette von gemessenen Parametern, die für die prädiktive Instandhaltung von entscheidender Bedeutung sind. Dazu gehören Motor-RPM, Motorleistung, Motordrehmoment, Auslassdruck des Kompressors, Luftstrom in der Druckluftleitung, Kompressorgeräusche, Temperatur am Auslass der Druckluftleitung, Temperatur am Auslass der Wasserpumpe, Eintrittstemperatur des Wassers, Ausgangstemperatur des Wassers, Leistung der Wasserpumpe, Wasserfluss, Leistung der Ölpumpe und Temperatur des Öltanks. Zusätzlich werden Bodenbeschleunigung und Kopfbeschleunigung in den Achsen X, Y und Z gemessen. Die Resultat- und Beobachtungsparameter umfassen den Zustand der Wasserpumpe, den Zustand des Radiators und den Zustand des Auslassventils sowie den Zustand des Motors.

what we have?	
Measured Parameters	Result & Observation Parameters
<ul style="list-style-type: none"> • MOTOR RPM • MOTOR POWER • MOTOR TORQUE • COMPRESSOR OUTLET PRESSURE • PRESSURED LINE AIR FLOW • COMPRESSOR NOISE • PRESSURED AIR LINE OUTLET TEMPERATURE • WATER PUMP OUTLET TEMPERATURE • WATER INLET TEMPERATURE • WATER OUTLET TEMPERATURE • WATER PUMP POWER • WATER FLOW • OIL PUMP POWER • OIL TANK TEMPERATURE • GROUND ACCELERATION X,Y,Z • HEAD CAP ACCELERATION X,Y,Z 	<ul style="list-style-type: none"> • BEARING CONDITION • WATER PUMP CONDITION • RADIATOR CONDITION • OUTLET VALVE CONDITION • MOTOR CONDITION

Abbildung 2-13: Datensatz aus Kaggle in CSV-Datei mit allen gemessenen Daten [68]

Dieser Datensatz in Abbildung 2.13 wurde auf der Plattform Kaggle als CSV-Datei zu Verfügung gestellt. Kaggle bietet eine dynamische Umgebung, in der Fachleute aus verschiedenen Bereichen zusammenkommen, um an herausfordernden Projekten zu arbeiten. Die Plattform fördert den Wissensaustausch, die Zusammenarbeit und den Wettbewerb durch eine Vielzahl von Projekten, Wettbewerben und Diskussionen. Mit einer umfangreichen Sammlung von Datensätzen und einer lebendigen Community bietet Kaggle eine einzigartige Möglichkeit, praktische Erfahrungen zu sammeln, innovative Lösungen zu entwickeln und sich beruflich weiterzuentwickeln. Die breite Palette von Daten und die Vielfalt der Projekte machen Kaggle zu einem unverzichtbaren Instrument für die Entwicklung und Umsetzung von datengetriebenen Anwendungen in verschiedenen Branchen.

Programmiersprache und Software

Python ist eine Programmiersprache, die dank ihrer klaren Syntax und einfachen Lesbarkeit leicht zu erlernen ist und sich sehr vielseitig einsetzen lässt. Für die gängigen Betriebssysteme ist Python frei verfügbar. Die üblichen Programmierparadigmen wie die objektorientierte oder funktionale Programmierung werden unterstützt.

Während Colaboratory, oft als „Colab“ abgekürzt, ist ein Produkt von Google Research. Colab ermöglicht es jedem, Python-Code seiner Wahl über den Browser zu schreiben und auszuführen. Es ist eine Umgebung, die besonders gut für maschinelles Lernen, Datenanalyse und Bildung geeignet ist. Technisch gesehen ist Colab ein gehosteter Dienst für Jupyter-Notebooks, der keine Konfiguration erfordert und kostenlosen Zugriff auf Rechenressourcen bietet, einschließlich GPUs.

Das Jupyter-Notebook ist eine Open-Source-Webanwendung, mit der Dokumente erstellt und geteilt werden können, die Live-Text, Code, Gleichungen und Visualisierungen enthalten. Der Name Jupyter leitet sich von den unterstützten Programmiersprachen ab, die Jupyter unterstützt: Julia, Python und R.

Es ermöglicht den Benutzern, Code in „Zellen“ zu schreiben, d. h. in Codeblöcken, die einzeln oder gemeinsam ausgeführt werden können. Dies erleichtert das Testen und

Debuggen von Code sowie die Organisation und Präsentation komplexer Analysen. Darüber hinaus ermöglicht das Notizbuch das Einfügen von Markdown-Text und LaTeX-Gleichungen, so dass es einfach ist, die Arbeit zu dokumentieren und die Ergebnisse mit anderen zu teilen.

Es kann mehrere Programmiersprachen unterstützen. Dazu gehören beliebte Sprachen wie Python, R und Julia sowie viele andere. Das macht es zu einem unglaublich vielseitigen Tool für Datenwissenschaftler, die damit in der Sprache arbeiten können, die ihnen am besten liegt.

Ein weiterer Vorteil von Jupyter-Notebooks ist seine Fähigkeit, interaktive Visualisierungen zu erstellen. Das Notebook unterstützt eine breite Palette von Visualisierungsbibliotheken, darunter Matplotlib, Seaborn und Plotly, mit denen Benutzer interaktive Diagramme, Karten und Diagramme erstellen können. Dies erleichtert die Erforschung und Visualisierung von Daten auf eine Weise, die sowohl leistungsstark als auch intuitiv ist.

Jupyter-Notebook ist auch ein großartiges Werkzeug für die Zusammenarbeit. Benutzer können ihre Notizbücher mit anderen teilen, entweder durch die Freigabe einer Datei oder durch die Verwendung eines webbasierten Dienstes wie GitHub oder Binder. Auf diese Weise können andere den Code anzeigen und ausführen, was die Zusammenarbeit an Projekten und den Austausch von Ergebnissen erleichtert.

Jupyter-Notebook ist ein großartiges Werkzeug für das Lehren und Lernen. Seine Benutzerfreundlichkeit und Vielseitigkeit machen es zu einem wertvollen Werkzeug für Lehrkräfte, die Programmieren oder Data Science unterrichten möchten. Darüber hinaus gibt es viele Online-Ressourcen und Tutorials, die den Benutzern helfen können, die Verwendung von Jupyter-Notebooks zu erlernen und das Beste aus seinen Funktionen herauszuholen.

3 DATENBASIERTE PRÄDIKTIVE INSTANDHALTUNGSSTRATEGIEN

Das Kapitel "Datenbasierte prädiktive Instandhaltungsstrategien" stellt einen modernen Ansatz zur Instandhaltung vor, der auf der Analyse von Daten basiert. Ziel ist es, durch die frühzeitige Erkennung potenzieller Probleme Ausfallzeiten zu minimieren und die Effizienz zu steigern. Der Schwerpunkt liegt auf der Nutzung statistischer Methoden zur Bewertung von Datensätzen, der Anwendung verschiedener Datenanalyse-Techniken und der anschließenden Bewertung dieser Analysen, um präzise und vorausschauende Wartungsmaßnahmen zu ermöglichen. Dieser Ansatz markiert einen signifikanten Fortschritt gegenüber traditionellen, reaktiven Wartungsmethoden.

3.1 Statistische Methoden zur Bewertung eines Datensatzes

Das erste Unterkapitel "Statistische Methoden zur Bewertung eines Datensatzes" im Rahmen des Kapitels über datenbasierte prädiktive Instandhaltungsstrategien bietet eine umfassende Einführung in die statistischen Techniken und Methoden, die für die Analyse und Bewertung von Datensätzen unerlässlich sind. Es beginnt mit einer Erläuterung der statistischen Grundlagen, die das Fundament für das Verständnis der Datenanalyse bilden. Anschließend werden spezifische Analysemethoden behandelt, darunter die Korrelationsanalyse, die wichtige Einblicke in die Beziehungen zwischen verschiedenen Datenpunkten gibt. Die Hauptkomponentenanalyse wird als ein Instrument zur Reduzierung der Datendimension und zur Hervorhebung der wichtigsten Datenmerkmale vorgestellt. Im weiteren Verlauf wird die Effektanalyse erörtert, die darauf abzielt, den Einfluss einzelner Variablen zu identifizieren und zu bewerten. Abschließend wird die Varianzanalyse behandelt, ein entscheidendes Werkzeug zur Untersuchung von Unterschieden zwischen Gruppen und Variablen. Insgesamt bietet dieses Unterkapitel einen tiefen Einblick in die statistischen Verfahren, die für eine effektive Datenbewertung in der prädiktiven Instandhaltung notwendig sind [69].

3.1.1 Statistische Grundlagen

Für die Bewertung des Datensatzes werden einige statistische Grundlagen benötigt. Das arithmetische Mittel \bar{x} berechnet sich aus allen Werten x_i aus der Anzahl der Werte N [70]

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (1)$$

Die Varianz σ_x^2 einer Verteilung ist das arithmetische Mittel der quadrierten Abweichung eines Datenpunktes x_i vom arithmetischen Mittel der Verteilung \bar{x} und wird berechnet als [71]

$$\sigma_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2. \quad (2)$$

3.1.2 Korrelationsanalyse

Korrelationsanalysen werden eingesetzt, um die statistische Wechselbeziehung zwischen zwei Variablen x und y in einer Datenreihe zu erforschen. Eine gängige Methode zur Messung dieser Beziehung ist der Bravais-Pearson-Korrelationskoeffizient [72].

$$r_{xy} = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}) * (y_i - \bar{y})}{\sigma_x * \sigma_y} \quad (3)$$

Er hilft zu bestimmen, wie stark x und y zusammenhängen, wobei seine Werte zwischen -1 und 1 liegen. Positive Werte deuten auf eine direkte Beziehung hin (wenn x steigt, steigt auch y), während negative Werte auf eine inverse Beziehung hinweisen (wenn x steigt, fällt y). Je näher der Wert an 1 oder -1 liegt, desto stärker ist der lineare Zusammenhang [73].

Eine weitere Analysemethode ist der Spearman-Rangkorrelationskoeffizient, der besonders nützlich ist, wenn die Daten nicht linear sind oder auf einer ordinalen Skala basieren. Dieser Ansatz beurteilt die Korrelation anhand der Rangordnung der Datenpunkte. Jeder Datenpunkt x_i erhält einen Rang basierend auf seiner Position in der aufsteigend sortierten Datenreihe.

$$rg(x_i) = rg(x_j) = j \quad (4)$$

Auch der Spearman-Koeffizient variiert zwischen -1 und 1, wobei die Interpretation der Korrelationsstärke ähnlich wie beim Bravais-Pearson-Koeffizienten erfolgt. Der Hauptunterschied liegt darin, dass der Spearman-Koeffizient die Monotonie der Beziehung zwischen den Variablen bewertet, unabhängig davon, ob die Beziehung linear ist oder nicht [73].

$$r_{xy} = \frac{\frac{1}{N} \sum_{i=1}^N (rg(x_i) - \bar{rg}(x)) * (rg(y_i) - \bar{rg}(y))}{\sigma_{rg(x)} * \sigma_{rg(y)}} \quad (5)$$

In dieser Arbeit wird, wenn nicht anders angegeben, der Spearman-Rangkorrelationskoeffizient bevorzugt verwendet. Dies ermöglicht die Analyse von Daten auf einer ordinalen Skala und konzentriert sich auf die Bewertung der Monotonie in der Beziehung zwischen zwei Variablen. Die Aussage der Korrelationen ist in Tabelle 3-1 dargestellt [73].

Tabelle 3-1 - Korrelationsstärke anhand des Rangkorrelationskoeffizienten nach [73]

Aussage	Korrelation
Kein Zusammenhang	$ r \leq 0,1$
Schwacher Zusammenhang	$0,1 < r \leq 0,3$
Mittlerer Zusammenhang	$0,3 < r \leq 0,8$
Starker Zusammenhang	$0,8 < r \leq 1$

3.1.3 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse zielt darauf ab, den Informationsgehalt in einem Datensatz verschiedener Messgrößen zu analysieren, indem sie sich auf die Gesamtvarianz konzentriert. Im Rahmen dieser Methode werden durch eine Koordinatentransformation sogenannte Hauptkomponenten gebildet. Diese neuen Größen repräsentieren in abnehmender Reihenfolge den größtmöglichen Teil der Gesamtvarianz, der noch nicht durch frühere Hauptkomponenten erfasst wurde. [74]

Der Prozess zur Ermittlung der Hauptkomponenten erfolgt in mehreren Schritten. Zuerst werden die ursprünglichen Messgrößen standardisiert, sodass jede Größe $x_{S,j}$ einen Mittelwert von 0 und eine Standardabweichung von 1 aufweist. [74]

$$x_{S,j} = \frac{x_i - \bar{x}_j}{\sigma_{x,j}} \quad (6)$$

Aus diesen standardisierten Größen wird eine Matrix x erstellt, deren Varianz durch die Matrix $S = \text{Var}(x)$ beschrieben wird. Das Ziel ist es, die Linearkombination $a'_1 x$ zu finden, die die maximale Varianz unter der Bedingung $a'_1 a_1 = 1$ aufweist.

$$\max_a (\text{Var}(a'_1 x)) = \max_a (a'_1 a_1) \quad (7)$$

Dies führt zu einem Eigenwertproblem

$$S a_1 = \lambda a_1 \quad (8)$$

dessen Lösung den größten Eigenwert λ_1 und den zugehörigen Eigenvektor a_1 ergibt, welcher die erste Hauptkomponente HK_1 bildet.

Die zweite Hauptkomponente wird ähnlich berechnet, mit der zusätzlichen Bedingung der Orthogonalität zu der ersten Hauptkomponente ($a'_2 a_1 = 0$). Weitere Hauptkomponenten werden nach demselben Prinzip bestimmt.

Da jede nachfolgende Hauptkomponente eine geringere Varianz aufweist, enthalten die ersten Hauptkomponenten einen größeren Teil der Gesamtvarianz, während die späteren einen kleineren Anteil haben. Daher kann es sinnvoll sein, Hauptkomponenten mit geringerem Informationsgehalt zu vernachlässigen und sich auf jene zu konzentrieren, die einen signifikanten Varianzanteil und somit Informationsgehalt aufweisen. Ein Maß für die Bewertung ist die Varianz jeder Hauptkomponente. Nach Joliffe [75] kann ein kritischer Grenzwert für die Varianz der Hauptkomponenten bei $\lambda_{krit} = 0,7$ angesetzt werden. Hauptkomponenten, die unter diesem Wert liegen, können ausgelassen werden, wodurch die Anzahl der betrachteten Größen verringert wird, ohne den wesentlichen Informationsgehalt des Datensatzes zu beeinträchtigen [74].

3.1.4 Effektanalyse

In der Statistik wird der Zusammenhang zwischen verschiedenen Eingangsgrößen, auch als Faktoren F_i bezeichnet, und einer oder mehreren Ausgangsgrößen x analysiert. Verschiedene Methoden der statistischen Versuchsplanung werden von

Siebertz et al. beschrieben. Hier sollen die grundlegenden Methoden, die in dieser Arbeit verwendet werden, erläutert werden. In einem Datensatz ist der Zusammenhang aller N_F Faktoren auf allen N_S Faktorstufen dargestellt. Hierbei lassen sich die Beziehungen zwischen der betrachteten Ausgangsgröße und den variierten Faktoren berechnen. [76,77]

$$x = \bar{x} + \sum_{i=1}^{N_F} fn(F_i) + \sum_{i=1}^{N_F-1} \sum_{j=i+1}^{N_F} fn(F_i, F_j) + \sum_{i=1}^{N_F-2} \sum_{j=i+1}^{N_F-1} \sum_{k=i+2}^{N_F} fn(F_i, F_j, F_k) \quad (9)$$

$$\dots + fn(F_1, \dots, F_{N_F})$$

Die verwendeten Funktionen sind Polynome des Grades $N_S - 1$. Der Mittelwert aller Messungen wird mit \bar{x} angegeben, die Haupteffekte der untersuchten Faktoren mit (F_i) , und die restlichen Terme repräsentieren die Wechselwirkungen der Faktoren. Die Modellkonstanten der Polynome werden im Rahmen der Auswertung bestimmt, wobei die Anzahl der Datenkombinationen der Anzahl der Parameter entsprechen muss [76].

$$N_M = N_S^{N_F} \quad (10)$$

Es wird deutlich, dass die Auswertungskomplexität exponentiell mit der Anzahl der untersuchten Faktoren N_F und der Anzahl der Faktorstufen N_S ansteigt. Für eine erste Untersuchung, ob ein Faktor oder eine Wechselwirkung zwischen Faktoren einen signifikanten Effekt auf die untersuchte Größe hat, reicht es aus, diese in zwei Faktorstufen zu betrachten. Es wird angenommen, dass vor allem Haupteffekte und Wechselwirkungen zwischen zwei Faktoren den größten Einfluss auf die Ausgangsgröße haben, während Wechselwirkungen höherer Ordnung zunächst vernachlässigt werden können [76].

$$x = \bar{x} + \sum_{i=1}^{N_F} \frac{E_i}{2} F_i + \sum_{i=1}^{N_F-1} \sum_{j=i+1}^{N_F} \frac{E_i}{2} F_i F_j \quad (11)$$

Der Effekt E_i eines einzelnen Faktors F_i wird durch einen einzelnen Parameter dargestellt, der sich als die Differenz des Mittelwerts von x aller Versuche auf der höheren Faktorstufe $\bar{x}(F_i, +)$ und aller Versuche auf der niedrigeren Faktorstufe $\bar{x}(F_i, -)$ berechnet [76].

$$fn(F_i) = E_i = \bar{x}(F_i, +) - \bar{x}(F_i, -) \quad (12)$$

3.1.5 Varianzanalyse

Im vorangegangenen Abschnitt wurde dargelegt, wie die Effekte einzelner Faktoren aus einem Datensatz abgeleitet werden. Dieser Abschnitt konzentriert sich auf die Anwendung der Varianzanalyse zur Überprüfung der Echtheit dieser Effekte. Nicht signifikante Effekte können aus der Modellgleichung ausgeschlossen werden, was das Modell vereinfacht. Die Varianzanalyse ist ein etabliertes Verfahren in der statistischen Datenanalyse und wird in vielen Datenanalyseprogrammen genutzt.

Das Ziel der Varianzanalyse besteht darin, festzustellen, welcher Teil der Gesamtvarianz einer untersuchten Ausgangsgröße durch den Effekt eines

Versuchsfaktors erklärt werden kann und wie wahrscheinlich es ist, dass dieser Effekt zufällig ist. Der hierfür verwendete Wert ist die "Sum of Squares between Groups" SSB_i , die sich aus der Anzahl der Versuche auf jeder Stufe berechnet. Diese Formel gilt für einen zweistufigen Versuchsplan; bei mehreren Stufen muss sie angepasst werden [76].

$$SSB_i = N_+(\bar{x}(F_{i,+}) - \bar{x})^2 + N_-(\bar{x}(F_{i,-}) - \bar{x})^2 \quad (13)$$

Die "Total Sum of Squares" TSS repräsentiert die Gesamtvarianz. Der nicht durch die N betrachteten Effekte erklärte Varianzanteil wird durch die Differenz SSW ausgedrückt, was als Fehler (Error) bezeichnet wird. Die Größen SSB und SSW werden durch ihre jeweiligen Freiheitsgrade geteilt.

$$TSS = \sigma_x^2 \quad (14)$$

$$SSW = TSS - \sum_{i=1}^N SSB_i \quad (15)$$

Ein Faktor oder eine Faktorwechselwirkung mit zwei Stufen hat einen Freiheitsgrad, ein Versuchsplan mit 16 Versuchen hat 15 Freiheitsgrade. Die Differenz zwischen den Freiheitsgraden des Versuchsplans und der Modellparameter wird dem Fehler zugeordnet. Die Quotienten aus SSB und SSW und den zugehörigen Freiheitsgraden werden als "Mean Square" MS bezeichnet. Um die Wirkung eines Faktors im Verhältnis zum Fehler zu bewerten, werden die Mean Squares verglichen, um den f -Quotienten des Faktors F_i zu berechnen [72].

$$f = \frac{MS(SSW_i)}{MS(SSB)} \quad (16)$$

Dann wird getestet, ob der untersuchte Effekt mit einem bestimmten f -Quotienten zufällig ist. Diese Wahrscheinlichkeit wird mit der Fisher-Verteilung ermittelt und mit einem festgelegten Signifikanzniveau verglichen. Liegt die berechnete Wahrscheinlichkeit unter diesem Niveau, wird die Hypothese eines zufälligen Effekts verworfen. Effekte, bei denen diese Hypothese nicht verworfen werden kann, gelten als zufällig und werden aus weiteren Analysen ausgeschlossen. Dabei kann es passieren, dass Effekte, die zunächst als nicht zufällig galten, nach dem Entfernen eines anderen Effekts als zufällig eingestuft werden, da sich das $MS(SSB)$, der F -Quotient und somit auch das Ergebnis des Hypothesentests für die verbleibenden Effekte ändert. Das Ergebnis der Varianzanalyse wird typischerweise in einer ANOVA-Tabelle (Analysis of Variance) dargestellt.

3.2 Methoden zur Datenanalyse

Das zweite Unterkapitel "Methoden zur Datenanalyse" konzentriert sich auf die zentralen Analysetechniken in der prädiktiven Instandhaltung. Es umfasst die Regression zur Erkennung von Beziehungen und Trends, die Neuronalen Netze zur Erkennung von Mustern und die Verwendung von Entscheidungsbäumen (Decision Trees) für die Klassifizierung und Vorhersage. Jede Methode bietet einzigartige Einsichten und ist entscheidend für die effektive Analyse und Interpretation von Daten in der prädiktiven Instandhaltung [78].

Beim überwachten Lernen werden Probleme behandelt, bei denen Daten einschließlich ihrer Lösungen, bekannt als Labels, zur Verfügung stehen. In diesem Bereich sind Klassifikation und Regression typische Aufgaben. Klassifikationsalgorithmen entwickeln Regeln zur Analyse von Bildern, um das Motiv zuverlässig zu identifizieren. Ein klassisches Beispiel im maschinellen Lernen ist die Unterscheidung zwischen Hunden und Katzen. Regression hingegen befasst sich mit kontinuierlichen Werten, wobei aus kontinuierlichen Eingangsdaten ein numerischer Wert abgeleitet wird. Ein Beispiel hierfür ist die Analyse linearer Trends in den Stückkosten der Produktion. Diese Prozesse werden in einer maschinellen Lernpipeline dargestellt und umfassend betrachtet, einschließlich gängiger Hilfsmethoden, die später in der Arbeit angewendet werden, siehe Abbildung 3-1 [79].

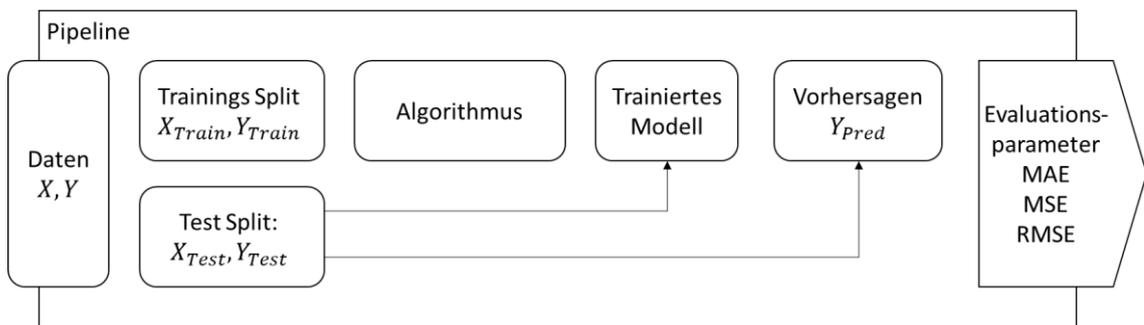


Abbildung 3-1 - Pipeline maschinelles Lernen nach [80]

In Anwendungen des überwachten maschinellen Lernens liegen Daten in Form eines Datensatzes vor, der in eine Eingangsmatrix X und einen Labelvektor Y unterteilt ist. Die Eingangsmatrix besteht aus p verschiedenen Prädiktoren X_p , die den Ausgangswert Y beeinflussen. Der Zusammenhang zwischen ihnen ist oft unbekannt und wird als unbekannte Funktion f dargestellt. Da Daten in der Realität immer einen zufälligen Messfehler ε aufweisen, wird die Messung allgemein so dargestellt. Ziel des überwachten Lernens ist es, diese Funktion zuverlässig zu ermitteln, um Vorhersagen und Schlussfolgerungen zu treffen [81].

$$Y = f(X) + \varepsilon \quad (17)$$

Vorhersagen sind wichtig, da oft die Eingangsmatrix X bekannt, das Ergebnis Y jedoch nicht einfach zu ermitteln ist. Die approximierte Funktion \hat{f} kann je nach Algorithmus bekannt oder unbekannt sein. Beispielsweise ist die Funktion der linearen Regression bekannt, während die eines Black-Box-Algorithmus unbekannt ist. Ein neuronales Netz ist ein Beispiel für eine solche Black-Box-Methode [81].

$$\hat{Y} = \hat{f}(X) \quad (18)$$

Inferenzen aus datengetriebenen Fragestellungen zu ziehen, ist ein weiteres Anwendungsfeld des maschinellen Lernens. Maschinelles Lernen kann Zusammenhänge zwischen Eingangs- und Ausgangsparametern darstellen und quantifizieren. Der Trade-off zwischen der Interpretierbarkeit eines Modells und seiner Approximationsgüte ist ein wichtiger Aspekt. Funktionen werden je nach Methode mit einfachen oder komplexen, nichtlinearen Abschnitten beschrieben [82].

Zu Beginn der maschinellen Lernpipeline wird der Datensatz in Trainings- und Test-Sets aufgeteilt. Typische Aufteilungen sind 70% für das Training und 30% für das Testen, wobei die Daten zufällig verteilt werden. Cross-Validation ist eine Methode, die die Datenausnutzung für Lern- und Testzwecke erhöht, indem der Datensatz in n zufällige Trainings- und Test-Splits aufgeteilt wird, siehe Abbildung 3-2. Dadurch können trotz Nutzung aller Daten für das Training Leistungsangaben gemacht und das Risiko vermieden werden, eine besonders gute oder schlechte zufällige Datenauswahl zu treffen [81].

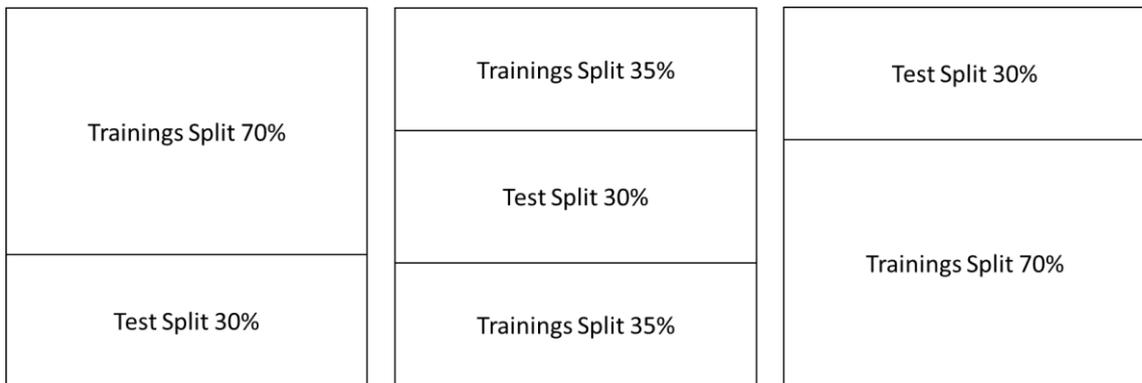


Abbildung 3-2 - Trainings und Test Splits zur 3-fach Cross Validation

3.2.1 Regression

In diesem Teil wird zuerst die einfache lineare Regression erläutert. Sie ist ein Modell, das die Beziehung zwischen zwei Variablen linear beschreibt. Bei der einfachen linearen Regression hängt die abhängige Variable Y von nur einer unabhängigen Variable ab. Die mathematische Formel dafür ist: [81]

$$Y = \beta_0 + \beta_1 X \quad (19)$$

Hierbei ist X die unabhängige Variable und β_0 sowie β_1 sind Parameter, die anfangs noch unbekannt sind. Diese Parameter werden durch Analyse eines Test-Datensatzes ermittelt. Mit der folgenden Formel werden dann Vorhersagen gemacht:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \quad (20)$$

Die Bestimmung der Koeffizienten β_0 und β_1 erfolgt während des Trainingsprozesses, indem die Summe der quadrierten Abweichungen (least squares) minimiert wird. Die Güte der Vorhersage wird durch die Residuenquadratsumme (RSS) beurteilt:

$$RSS = e_1^2 + e_1^2 + \dots + e_n^2 \quad (21)$$

Hier steht e_n^2 für das Quadrat der Differenz zwischen der tatsächlichen und der vorhergesagten Ausgabe:

$$e_n = y_i - \hat{y}_i \quad (22)$$

Die Suche nach der optimalen Annäherung an die gemessenen Werte zielt darauf ab, den Fehler (RSS) zu minimieren. Dies geschieht durch die gleichzeitige Betrachtung

aller Trainingsdaten. Nach Einsetzen der Residuen und Umformung der Gleichungen ergeben sich die folgenden Ausdrücke für die Koeffizienten:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (23)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (24)$$

Dabei sind $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ und $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ die Durchschnittswerte der Beobachtungen. Die Minimierung dieser Koeffizienten erfolgt meistens numerisch, häufig mittels der Gradient Descent Methode. Diese repräsentiert das globale Optimum der Minimierungsfunktion.

Mathematisch lässt sich der Gradient Descent Algorithmus wie folgt beschreiben:

$$\hat{\beta}(k+1) = \hat{\beta}(k) - a \frac{\delta RSS}{\delta \beta} \quad (25)$$

Hierbei ist a die Lernrate und ein wichtiger Hyperparameter im Algorithmus. Hyperparameter sind zusätzliche Einstellungen in Algorithmen, die deren Leistungsfähigkeit beeinflussen. Die Lernrate bestimmt sowohl die Konvergenzfähigkeit des Algorithmus als auch die Geschwindigkeit, mit der das Minimum erreicht wird [81].

In der realen Welt sind physikalische Phänomene häufig von mehreren Zustandsgrößen abhängig, was die Anwendung der einfachen linearen Regression für eine präzise Schätzung einschränkt. In diesem Abschnitt wird daher ein lineares Regressionsmodell entwickelt, das mehrere Eingangsvariablen berücksichtigt, bekannt als multidimensionale lineare Regression.

Die Grundfunktion dieses Verfahrens basiert auf einer Erweiterung der Gleichung 20. Sie bildet p Parameter unterschiedlich gewichtet auf die abhängige Variable Y ab, wodurch eine p -dimensionale Hyperebene entsteht [81]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p \quad (26)$$

Wie bei der einfachen linearen Regression sind auch hier die Koeffizienten zunächst unbekannt und müssen mathematisch geschätzt werden. Vorhersagen können dann mit folgender Formel gemacht werden:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots + \hat{\beta}_p x_p \quad (27)$$

Ein verbreiteter Ansatz zur Erweiterung einer linearen Regression ist die polynomiale Regression. Bei dieser Methode wird die Beziehung zwischen Eingangs- und Ausgangsparametern durch nichtlineare Funktionen dargestellt, die als Polynome d -ter Ordnung ausgedrückt werden. Für den eindimensionalen Fall haben diese Polynome die allgemeine Form:

$$\hat{y}_1 = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_1^2 + \dots + \hat{\beta}_d x_1^d \quad (28)$$

In der Praxis werden Polynome meist bis zur 3. oder 4. Ordnung verwendet, da Polynome höherer Ordnung zu flexibel werden und eine steigende Varianz mit sich bringen.

Das Training solcher Modelle erfolgt mittels Gradient Descent (Gleichung 24), das bei linearen Problemen ein globales Optimum garantieren kann. Obwohl die polynomiale Regression eine nichtlineare Funktion ist, kann das Problem linear gelöst werden, da die Eingangsparameter potenziert werden und nicht die Parameter $\hat{\beta}_d$. Diese Eigenschaft macht die polynomiale Regression effizient.

3.2.2 Neuronale Netze

Das neuronale Netz, vorgestellt als ein zu lösendes, nichtlineares Optimierungsproblem, wird zunehmend für komplexe Aufgaben eingesetzt, da es komplexe Zusammenhänge lernen kann. Der Basisalgorithmus wurde vor Jahrzehnten publiziert. Dieses Kapitel behandelt die Grundlagen neuronaler Netze und Techniken, die sich für präzisere Vorhersagen der Lagerimpedanz als nützlich erwiesen haben. [83]

Der Trend in der Forschung hat sich von biologischen Analogien entfernt, und aktuelle Publikationen bevorzugen neutrale Bezeichnungen für die Bestandteile eines neuronalen Netzes. Die sogenannte Unit, siehe Abbildung 3-3, folgt der Rechenvorschrift:

$$s_k = \sum_j (w_{jk} y_j) + \theta_k. \quad (29)$$

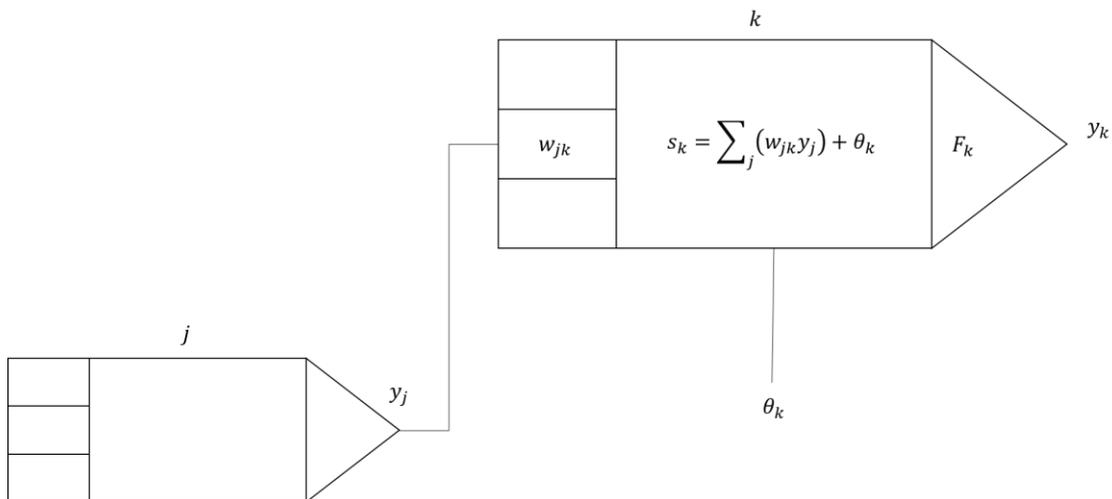


Abbildung 3-3 - Unit eines neuronalen Netzes nach [79]

Diese Vorschrift, bekannt als Forward Propagation, beschreibt, wie eingehende Werte y_j in der Unit zu einem Ausgangswert y_k verarbeitet werden. Jeder Eingangswert wird mit einem Gewicht w_{jk} multipliziert und summiert. Diese Gewichte sind entscheidende Optimierungsparameter des Netzes.

Zu jeder Unit gehört ein Bias θ_k , ein zusätzlicher Optimierungsparameter (Gleichung 29). Der daraus resultierende Wert s_k ist der effektive Input der Unit. Dieser wird durch eine Aktivierungsfunktion F_k modifiziert, was als Veränderung der Signalstärke der Unit interpretiert werden kann. Verschiedene Funktionen sind für unterschiedliche Aufgaben wie Regression und Klassifikation verfügbar. [84]

Für Regressionsprobleme ist die nichtlineare Rectified Linear Unit (ReLU) Funktion Standard, definiert als

$$f_x = \max(0, x) = \begin{cases} x_i, & \text{wenn } x_i \geq 0 \\ 0, & \text{wenn } x_i < 0 \end{cases} \quad (30)$$

Ist der Unit-Input 0, bleibt das Signal aus. Ist er größer gleich 0, erfolgt eine lineare Verstärkung. Diese Funktion beschleunigt den Lernprozess, da sie komplexe analytische Differenzierungsrechnungen überflüssig macht.

Die beschriebenen Units werden in sogenannten Layern organisiert. Das neuronale Netz besteht aus einem Input-Layer, Hidden Layern und einem Output Layer. Die Anzahl der Layer und Units variiert je nach Problemstellung. Wichtig ist eine ausreichende Anzahl von Units zur Darstellung der Komplexität des Problems, jedoch ohne Overfitting. Im Output Layer entspricht die Anzahl der Units den Ausgangsparametern. Für Regressionen ist ebenfalls die ReLU Funktion Standard, während Klassifikationen mittels Sigmoid- oder Tangens-hyperbolicus-Funktionen umgesetzt werden, passend für binäre Ausgaben. Der Aufbau ist in Abbildung 3-4 dargestellt. (84)

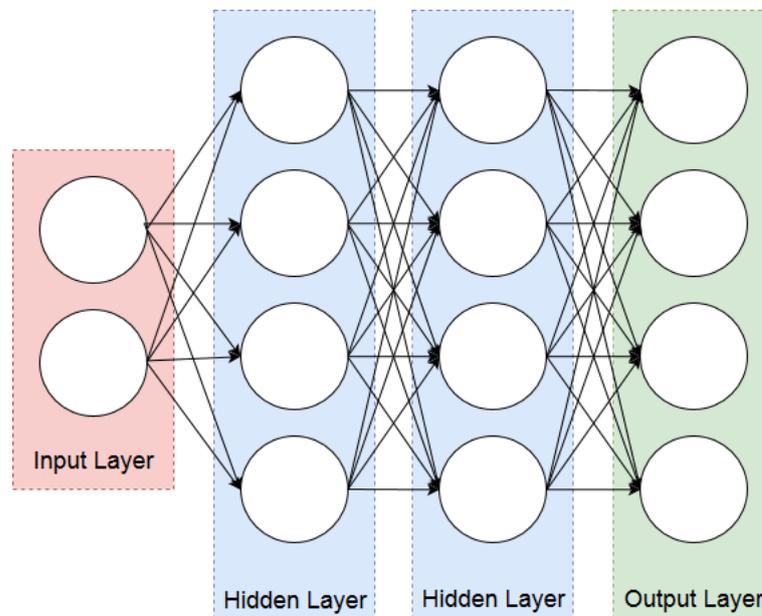


Abbildung 3-4 - Schematischer Aufbau neuronaler Netze [85]

Der Output Layer umfasst eine Unit für eine Ausgangsvariable. Eine Gittersuche zeigt, dass zwei versteckte Layer mit jeweils zehn Units eine optimale Balance zwischen Bias und Varianz bieten.

Die Balance zwischen Varianz und Bias wird am Root Mean Square Error (RSME) des neuronalen Netzes gemessen, wie bei anderen Algorithmen auch. Das Training des Netzwerks basiert auf der Optimierung der Gewichtsparameter w und der Bias-Parameter θ .

Die komplexe Struktur neuronaler Netze ermöglicht es ihnen, beliebig komplexe Funktionen zu approximieren, was allerdings das Optimierungsproblem nichtlinear macht. Daher liegt die Lösung nicht in einem konvexen Raum, und klassische iterative Methoden wie der Gradientenabstieg können in lokalen Minima stecken bleiben, anstatt eine nahezu ideale Lösung zu finden.

Unter den Methoden zur Lösung in nichtlinearen Räumen ist der Stochastic Gradient Descent (SGD) eine Erweiterung des Gradientenabstiegs. Der Unterschied liegt in der Anzahl der in einer Iteration betrachteten Trainingsdaten. Während der Gradientenabstieg alle Trainingsdaten gleichzeitig nutzt, verwendet SGD ein Subset der Daten pro Iteration. Dies führt zu schnelleren Updates der Optimierungsparameter, kann aber zu hohen Varianzen führen, was es schwierig macht, ein Abbruchkriterium festzulegen [84].

Performance-Verbesserungen können durch modifizierte SGD-Methoden erzielt werden, wie die Erweiterung um einen Momentum-Term, der die Varianz des Verfahrens reduziert und schneller zu Optima führt. Diese Methode verwendet das Momentum γ , üblicherweise auf 0,9 gesetzt, und aktualisiert die Optimierungsparameter wie folgt:

$$v_t = \gamma v_{t-1} + a \nabla_{\theta} J(\theta) \quad (31)$$

$$\theta = \theta - v_t. \quad (32)$$

Ein weiterer gängiger Algorithmus ist das Limited-memory BFGS (L-BFGS), ein Newton-Verfahren, das den Broyden-Fletcher-Goldfarb-Shanno Algorithmus approximiert. Es ist effizient, da die Hesse-Matrix durch eine positiv definite Matrix H_0 approximiert wird, und benötigt weniger Speicher:

$$s_k = x_{k+1} - x_k \quad (33)$$

$$y_k = g_{k+1} - g_k \quad (34)$$

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T \quad (35)$$

Wobei

$$\rho_k = 1/y_k^T s_k \text{ und } V_k = 1 - \rho_k y_k s_k^T \quad (36)$$

3.2.3 Decision Tree

Ein Entscheidungsbaum kann eingesetzt werden, um Regressionen in einem Datensatz durchzuführen. Als anschauliches Beispiel, das unabhängig von der Impedanz-Regression ist, wird der Einsatz von Eingangsparametern wie Gewicht und Blutdruck zur Vorhersage von Herzkrankheiten betrachtet. Der hypothetische Datensatz besteht aus dem Gewicht der Probanden und einer Kennzeichnung, ob eine Herzkrankheit vorliegt oder nicht. [81]

Zunächst wird mit dem Gini-Koeffizienten ermittelt, welche Gewichtsgrenze den Datensatz am effektivsten in zwei Gruppen teilt. Der Gini-Koeffizient misst die Unreinheit (Impurity) jeder potenziellen Gewichtseinteilung. Er berechnet sich durch:

$$G = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk}) \quad (37)$$

Hierbei steht \hat{p}_{mk} für den Anteil der Trainingsdaten in der m-ten Region der k-ten Klasse. Angenommen, bei einer Gewichtsgrenze von 75 kg ergibt sich eine Unreinheit von 0,3, was bedeutet, dass höhere Unreinheiten auf weniger geeignete Unterteilungen hinweisen. In Abbildung 3-5 ist das Schema dargestellt [86].

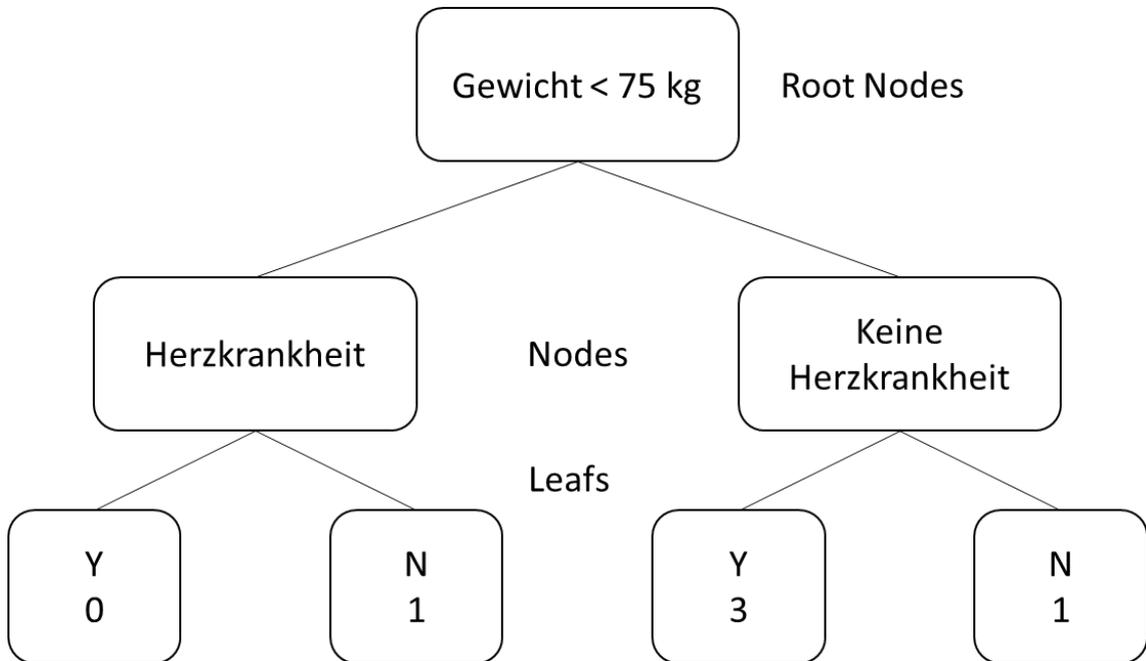


Abbildung 3-5 – Entscheidungsbaumschema nach [86].

Der Startpunkt des Entscheidungsbaums ist die Wurzel (Root Node). Weitere Entscheidungen tiefer im Baum, wie das Vorhandensein einer Herzkrankheit, sind als Knoten (Nodes) bezeichnet. Die tiefsten Entscheidungspunkte, die Blätter (Leafs), liefern das Endergebnis des Baums für gegebene Parameter.

Der optimale Entscheidungsbaum wird durch Überprüfung aller möglichen Entscheidungskombinationen gefunden. Neben dem Gini-Koeffizienten kann auch die Entropie zur Bewertung eines Baumes herangezogen werden. Bei rein numerischen Datensätzen wird oft das Recursive Binary Splitting verwendet, wie in der vorliegenden Abschlussarbeit für reine Regressionsprobleme. Dieser Ansatz wird durch folgendes Optimierungsproblem dargestellt:

$$\min \sum_{i: x_i \in R_1(j,s)} (y_i - \hat{y}_{R1})^2 + \sum_{i: x_i \in R_2(j,s)} (y_i - \hat{y}_{R2})^2 \quad (38)$$

Hierbei wird der Regressionsraum iterativ rekursiv binär in die Bereiche R_1 und R_2 aufgeteilt, um den Fehler im Trainingsset zu minimieren.

Obwohl Entscheidungsbäume gute Trainingswerte liefern können, sind sie bei der Vorhersage neuer Datenpunkte im Vergleich zu anderen Algorithmen oft unterlegen. Sie reagieren auch empfindlich auf Änderungen in den Daten, was zu weniger robusten Ergebnissen führt. Eine Möglichkeit, die Generalisierbarkeit zu verbessern, ist der Einsatz mehrerer Entscheidungsbäume, was im folgenden Abschnitt näher betrachtet wird.

Der Random Forest Algorithmus erstellt Vorhersagen, indem er die Entscheidungen mehrerer einzelner Entscheidungsbäume zusammenfasst. Diese Methode zielt darauf ab, die Vorhersagegenauigkeit zu erhöhen, indem sie eine Vielzahl von unabhängigen

Prognosen kombiniert. Die am häufigsten auftretende Vorhersage aus den verschiedenen Entscheidungsbäumen wird als die wahrscheinlichste angesehen.

Im Gegensatz zu einzelnen Entscheidungsbäumen, die bei identischem Input zum gleichen Ergebnis neigen, da sie ein globales Fehlerminimum erreichen, basiert der Random Forest auf dem Prinzip, dass jeder Baum auf einer zufälligen Teilmenge des ursprünglichen Datensatzes trainiert wird. Dies führt zu dekorrelierten und somit verbesserten Vorhersagen und ist in Abbildung 3-6 dargestellt.

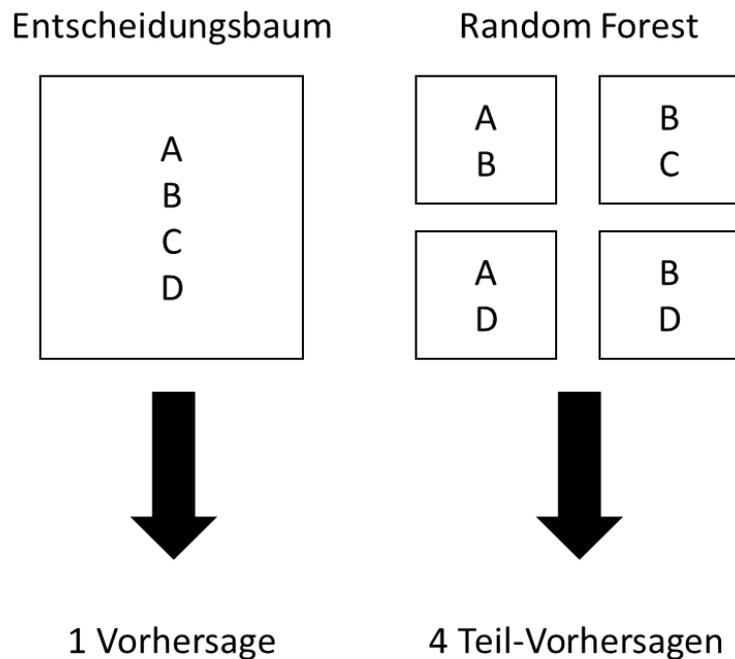


Abbildung 3-6 - Vergleich Entscheidungsbaum und Random Forest nach [86]

3.3 Methoden zur Bewertung der Datenanalyse

In diesem Teil wird die Einführung von Evaluationsparametern behandelt, die dazu dienen, die Leistungsfähigkeit einer Approximationsfunktion quantitativ zu bewerten. Ein zentraler Aspekt dabei ist der Vergleich zwischen prognostizierten Werten und tatsächlich gemessenen Daten als Maßstab für die Qualität [87].

Eines der ersten Bewertungskriterien für einen Algorithmus ist der durchschnittliche absolute Fehler, auch bekannt als Mean Absolute Error (MAE). Dieser gibt den mittleren Fehlerbetrag an:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (39)$$

Hierbei steht \hat{y}_i für die n vorhergesagten Werte und y_i für die n tatsächlich gemessenen Werte [81].

Ein weiteres Kriterium ist der durchschnittliche quadratische Fehler (Mean Squared Error, MSE). Er misst die mittlere quadratische Differenz zwischen Beobachtung und Vorhersage:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (40)$$

Der Root Mean Squared Error (RMSE) ist der Quadratwurzelwert des MSE und wird wie folgt definiert:

$$RMSE = \sqrt{MSE}$$

MSE und RMSE werden häufig im Bereich des maschinellen Lernens verwendet, da sie größere Abweichungen zwischen Vorhersage und Beobachtung stärker gewichten. Der RMSE hat den Vorteil, dass er in der gleichen Größenordnung wie das zugrundeliegende Problem liegt und somit einfacher zu interpretieren ist. Dennoch ist es im maschinellen Lernen notwendig, denn zu optimierenden Fehlerbegriff je nach Fall und den statistischen Bedingungen der Messungen individuell zu wählen [77].

4 ANWENDUNGSBEISPIEL DER PRÄDIKTIVEN INSTANDHALTUNG

Im Kapitel "Anwendungsbeispiel der prädiktiven Instandhaltung" wird ein praktisches Beispiel zur Anwendung prädiktiver Instandhaltungstechniken vorgestellt. Es beginnt mit der physikalischen Analyse des Datensatzes, gefolgt von einer statistischen Bewertung der Daten. Anschließend erfolgt die Datenanalyse, um relevante Muster zu identifizieren. Das Kapitel schließt mit der Bewertung der Ergebnisse und deren Implikationen für die Instandhaltungsstrategien ab. Diese Struktur ermöglicht ein umfassendes Verständnis der Schritte und Methoden in der prädiktiven Instandhaltung.

4.1 Physikalische Analyse des Datensatzes

Der Datensatz enthält verschiedene Messgrößen, die von verschiedenen Sensoren am Kompressor aufgezeichnet werden. Im Folgenden werden die verschiedenen Messgrößen vorgestellt.

Motor Drehzahl (RPM):

Die Drehzahl des Motors, der die Luftkompressoren antreibt, ist eine entscheidende Messgröße in industriellen Anwendungen. Diese Kompressoren versorgen üblicherweise die Produktionslinie in einer Fabrik und werden von einem Elektromotor angetrieben. Die Verbindung zwischen Motor und Kompressor kann über einen Riemenantrieb oder eine direkte Kupplung erfolgen. Die Drehzahl, ausgedrückt in RPM, gibt an, wie oft sich der Motor pro Minute dreht. Diese Information ist wichtig, um die Effizienz und die Leistungsfähigkeit des Kompressors sicherzustellen.

Motorleistung (MOTOR_POWER):

Die Motorleistung beschreibt, wie viel elektrische Energie ein Motor in mechanische Arbeit umwandeln kann. Diese Leistung ist essenziell für den Antrieb von Maschinen oder Ausrüstungen. Typischerweise wird die Motorleistung in Watt (W) oder Kilowatt (kW) angegeben und zeigt die Arbeitsrate des Motors an. Die Leistung eines Elektromotors wird durch verschiedene Faktoren bestimmt, einschließlich des Designs des Motors, der Menge des zugeführten elektrischen Stroms und der Effizienz des Motors bei der Umwandlung von elektrischer in mechanische Energie.

Motor Drehmoment (TORQUE):

Das Drehmoment von Elektromotoren unterscheidet sich von dem der Verbrennungsmotoren. Verbrennungsmotoren müssen eine bestimmte Drehzahl erreichen, um Drehmoment zu erzeugen. Gleichzeitig sind bei Verbrennungsmotoren Leistung und Drehmoment proportional zueinander. Elektromotoren hingegen können bereits bei 0 U/min (Umdrehungen pro Minute) Drehmoment produzieren.

Die Rohdaten der drei genannten Messwerte sind in Abbildung 4-1 dargestellt.

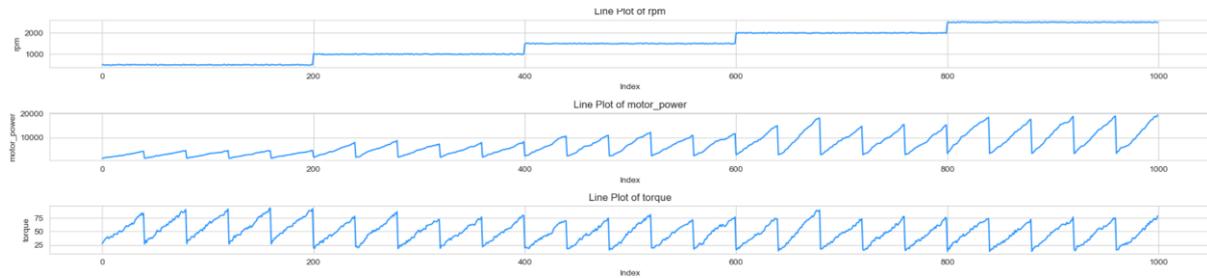


Abbildung 4-1 – Rohdaten RPM, MOTOR_POWER, TORQUE

Kompressor Auslassdruck (OUTLET_PRESSURE_BAR):

Der atmosphärische Druck beträgt rund 1 Bar. Wenn Luft in einem geschlossenen Volumen komprimiert wird, steigt ihr Druck an und gleichzeitig erwärmt sich die Luft. Der Luftkompressor saugt Luft mit einem Kolben an und versucht, sie in einen Tank zu drücken. Mit fortschreitendem Zyklus erhöht sich die Luftmasse im konstanten Volumen des Tanks, und dadurch steigt der Luftdruck. Der Auslassdruck wird direkt neben dem Kolbenventil gemessen und in Bar angegeben, wobei 1 Bar 100 Kilopascal (kPa) entspricht.

Luftstrom (AIR_FLOW):

Der Luftstrom eines Luftkompressors bezeichnet die Menge an Luft, die ein Kompressor liefern kann, und wird typischerweise in Kubikfuß pro Minute (CFM) oder in Litern pro Minute (L/min) gemessen. Er repräsentiert das Volumen der Luft, das der Kompressor bei einem bestimmten Druck liefern kann, und ist ein wichtiger Faktor für die Eignung eines Kompressors für eine spezifische Anwendung. Der Luftstrom eines Luftkompressors kann sich je nach Faktoren wie Größe und Design des Kompressors, der Energiequelle und den Betriebsbedingungen, wie z.B. Temperatur und Feuchtigkeit der komprimierten Luft, verändern. Der Luftstrom wird in der Regel vom Hersteller angegeben und kann zur Auswahl eines geeigneten Kompressors für eine bestimmte Aufgabe verwendet werden.

Geräuschpegel (NOISE_DB):

Der Geräuschpegel eines Luftkompressors bezieht sich auf die Lautstärke, die während des Betriebs des Kompressors entsteht. Diese wird in Dezibel (dB) gemessen und hängt von verschiedenen Faktoren ab, wie dem Design und der Größe des Kompressors, der Geschwindigkeit des Motors, der Art der verwendeten Kompressortechnologie und den akustischen Bedingungen, unter denen der Kompressor arbeitet. Übermäßiger Lärm von einem Luftkompressor kann lästig für nahegelegene Arbeiter sein und sogar ein Risiko für ihr Gehör darstellen, falls der Geräuschpegel hoch genug ist. Normalerweise, wenn der Klang des Luftkompressors nach einer gewissen Betriebszeit zunimmt, liegt das daran, dass die Lager ausgefallen sind.

Die Rohdaten der drei vorher genannten Messwerte sind in Abbildung 4-2 dargestellt.

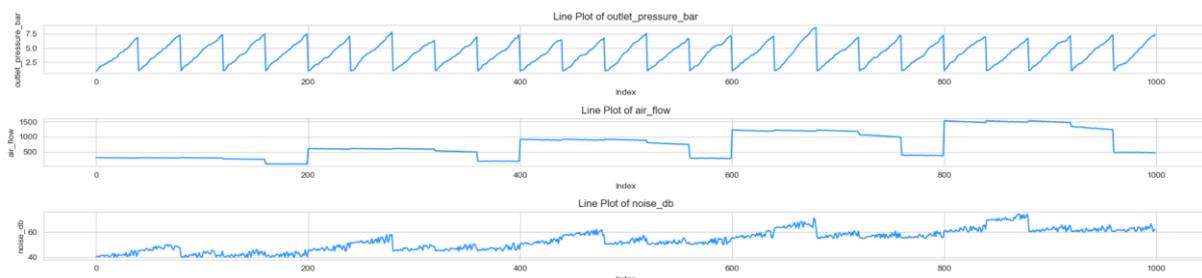


Abbildung 4-2 – Rohdaten OUTLET_PRESSURE_BAR, AIR_FLOW, NOISE_DB

Kompressor Auslasstemperatur (OUTLET_TEMP):

Die Auslasstemperatur eines Luftkompressors bezieht sich auf die Temperatur der komprimierten Luft beim Verlassen des Kompressors. Diese Temperatur wird durch verschiedene Faktoren beeinflusst, wie die Temperatur der zu komprimierenden Luft, das Kompressionsverhältnis, die Effizienz des Kompressors und die Art der verwendeten Kompressortechnologie. Generell wird die Auslasstemperatur eines Kompressors aufgrund des Wärmegenerierungsprozesses während der Kompression höher sein als die Umgebungstemperatur. Übermäßig hohe Auslasstemperaturen können jedoch auf Probleme wie Überhitzung, unzureichende Kühlung oder mangelhafte Schmierung hinweisen, was den Kompressor beschädigen oder seine Effizienz verringern kann. Die Überwachung der Auslasstemperatur ist daher wichtig, um potenzielle Probleme zu erkennen und sicherzustellen, dass der Kompressor innerhalb sicherer und effizienter Parameter arbeitet.

Wasserpumpen Auslassdruck (WPUMP_OUTLET_PRESSURE):

Kompressoren verdichten Luft, und während dieses Vorgangs erwärmt sich die komprimierte Luft. Die Metallteile des Kompressors, die gute Wärmeleiter sind, müssen gekühlt werden, um die Dichtungen nicht zu beschädigen und die Luftzirkulationseffizienz zu erhöhen. Lamellen- oder Liner-Kühlung wird bevorzugt. Lamellenkompressoren verfügen nicht über eine Flüssigkeitskühlung, sondern werden durch einen Ventilator gekühlt, der in der Nähe des Kompressors angebracht ist. Bei Modellen mit Ummantelungskühlung wird die Verbrennungskammer durch Wasser gekühlt. Die Energie des erwärmten Wassers wird durch einen Kühler abgeführt. Zur Zirkulation des Wassers werden Getriebe- und Zentrifugalpumpen eingesetzt. Der Druck am Auslass der Wasserpumpe ist ein wichtiger Indikator für die Leistung des Kühlsystems.

Wasser Einlasstemperatur (WATER_INLET_TEMP):

In Luftkompressoren, die durch die Zirkulation von Wasser aus dem Mantel gekühlt werden, erfolgt die Einlasstemperatur des Wassers entsprechend der Größe des Kühlers und der Kapazität des Ventilators. Die Wassertemperatur am Einlass ist ein Indikator dafür, wie effektiv der Kühler und der Ventilator die Wärme aufnehmen können.

Die Rohdaten der drei vorher genannten Messwerte sind in Abbildung 4-3 dargestellt.

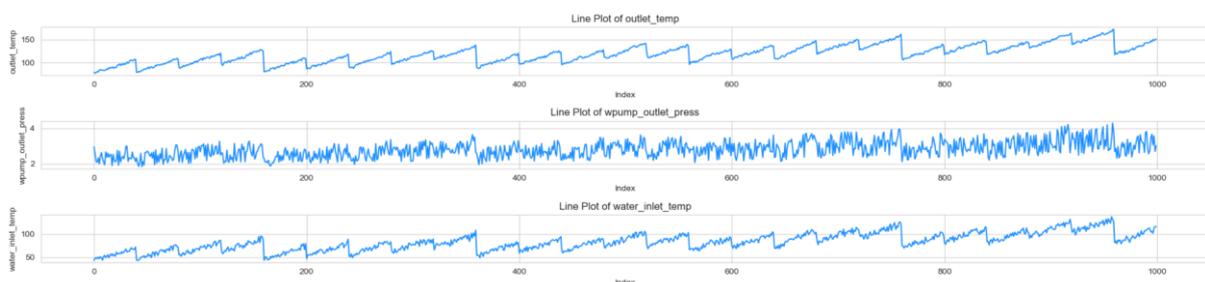


Abbildung 4-3 – Rohdaten OUTLET_TEMP, WPUMP_OUTLET_PRESSURE, WATER_INLET_TEMP

Wasser Auslasstemperatur (WATER_OUTLET_TEMP):

Die Wassertemperatur am Auslass steht im Zusammenhang damit, wie viel Wärme, die durch das Erhitzen der komprimierten Luft entsteht, an das Wasser übertragen wird. Die Tatsache, dass die Wassertemperatur am Einlass nahe der Wassertemperatur am Auslass liegt, könnte darauf hinweisen, dass der Kühler verschmutzt ist. Die Überwachung dieser

Temperaturen ist entscheidend, um die Effizienz des Kühlsystems zu gewährleisten und Überhitzungsprobleme zu vermeiden.

Wasserpumpenleistung (WPUMP_POWER):

Die Wasserpumpenleistung beschreibt die Energie, die die Pumpe in Volumenstrom umwandeln kann.

Wasservolumenstrom (WATER_FLOW):

Wenn der Kühler verschmutzt oder verstopft ist, verbraucht die Wasserpumpe mehr Energie, da der Wasserdruck steigt. Ebenso führt ein verschmutzter Kühler zu einer Verringerung des Wasservolumenstroms. Im Gegensatz zu Verbrennungsmotoren ist es wünschenswert, dass das Kühlmittel, das in den Kompressor eintritt, eine minimale Temperatur hat. Daher sind die Sauberkeit und Effizienz der Wasserpumpe und des Kühlers entscheidende Parameter für die Effizienz des Systems.

Die Rohdaten der drei vorher genannten Messwerte sind in Abbildung 4-4 dargestellt.

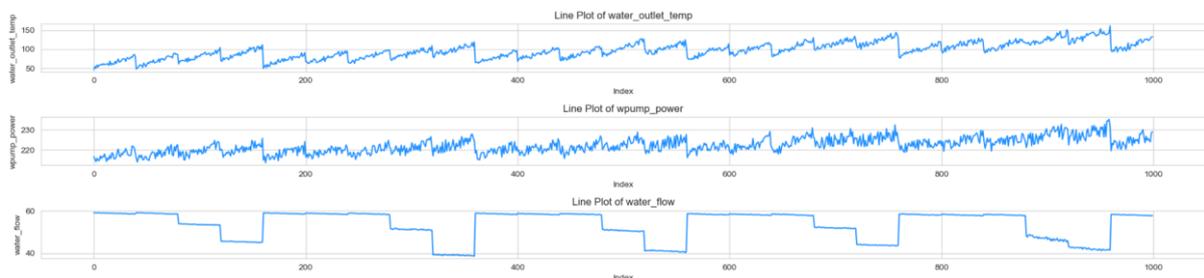


Abbildung 4-4 – Rohdaten WATER_OUTLET_TEMP, WPUMP_POWER, WATER_FLOW

Öldruck (OILPUMP_POWER):

Der Öldruck in einem Luftkompressor ist ein wichtiger Parameter, der die Schmierung und den reibungslosen Betrieb des Kompressors gewährleistet. Das Schmieröl wird in den Kompressor eingespritzt, um die beweglichen Teile zu schmieren und die Reibung zu reduzieren. Ein ausreichender Öldruck ist notwendig, um sicherzustellen, dass das Schmieröl alle Teile des Kompressors erreicht und eine effiziente Schmierung gewährleistet wird. Ein zu niedriger Öldruck kann zu erhöhtem Verschleiß der Komponenten und letztendlich zu einer Beschädigung des Kompressors führen. Daher wird der Öldruck in der Regel eng überwacht, um Probleme frühzeitig zu erkennen und die Lebensdauer des Kompressors zu verlängern.

Öltemperatur (OIL_TANK_TEMP):

Die Öltemperatur hat einen großen Einfluss auf die Schmierung. Einerseits führt eine zu hohe Temperatur zu geringerer Tragfähigkeit des Öls, sodass Schäden eher auftreten. Andererseits führt viel Reibung zu einer Erwärmung des Öls, sodass die Temperatur auf Schäden hindeutet.

Die Rohdaten der zwei vorher genannten Messwerte sind in Abbildung 4-5 dargestellt.

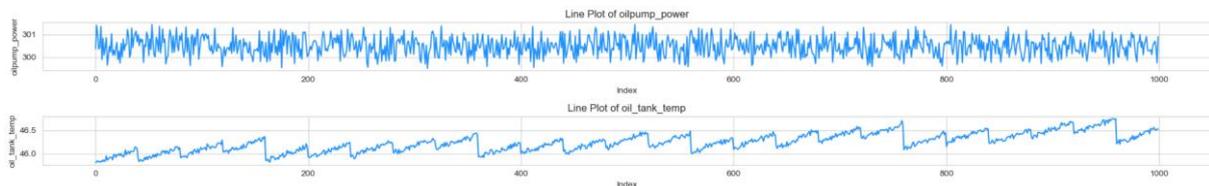


Abbildung 4-5 – Rohdaten OILPUMP_POWER, OIL_TANK_TEMP

Bodenbeschleunigung (GAACX, GAACY, GAACZ):

Die Bodenbeschleunigung ist ein Maß für die Änderung der Geschwindigkeit eines Objekts über die Zeit. Sie wird mathematisch als Geschwindigkeitsänderung geteilt durch die Zeitänderung ausgedrückt und in Metern pro Quadratsekunde (m/s^2) gemessen. Diese Beschleunigungswerte werden an einem starren Teil des Kompressors gemessen und geben Auskunft über eventuelle Abnutzung oder Spielraum in Teilen wie Kurbel, Lager oder Kappen.

Die Rohdaten der Bodenbeschleunigung sind in Abbildung 4-6 dargestellt.

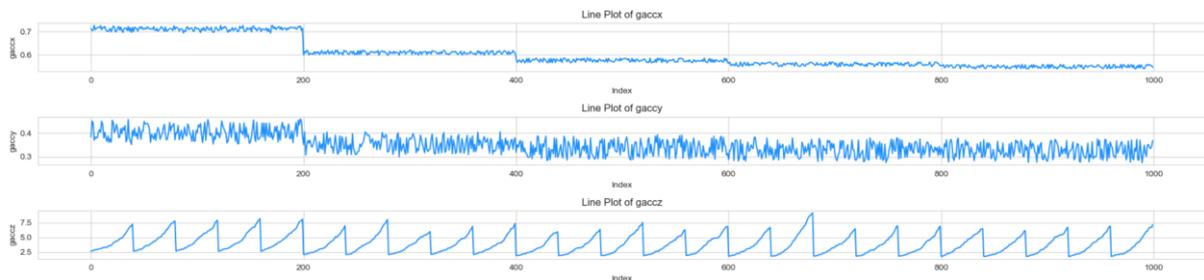


Abbildung 4-6 - Rohdaten GAACX, GAACY, GAACZ

Kopfbeschleunigung (HAACX, HAACY, HAACZ):

Die Kopfbeschleunigung wird an der Kühlung gemessen. Schlacke sammelt sich in den Auslassventilen der Kompressoren, deren Luftfilter sich verschlechtern und Staub aufnehmen, und die Steifigkeit der Struktur verändert sich. Mit der Änderung der Steifigkeit ändern sich das Kompressorgeräusch und die Kopfbeschleunigung. Die Zunahme oder Abnahme dieser Werte hängt vom Verdichtertyp ab, aber die Beobachtung dieser Werte ist ein guter Indikator für die Stabilität des Auslassventils.

Die Rohdaten der Kopfbeschleunigung sind in dargestellt.

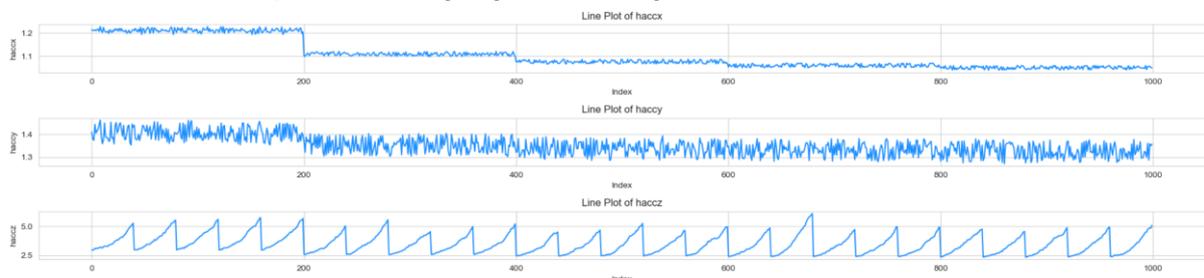


Abbildung 4-7 – Rohdaten HAACX, HAACY, HAACZ

Die relevanten Teile des Kompressors, die überwacht werden, sind die Lager (Bearings), die Kühlwasserpumpe (WPump), der Lüfter (Radiator), das Ventil (Exvalve) und der elektrische Antrieb (ACMotor), die Rohdaten sind in Abbildung 4-8 dargestellt. Während den oben beschriebenen Messdaten jeweils ein Messwert im Datensatz zugeordnet ist, ist den Komponenten des Kompressors ein Zustand zugeordnet. Der Gut-Zustand der Lager und der Kühlwasserpumpe ist „OK“ und der Schlecht-Zustand ist „Noisy“. Der Gutzustand des Lüfters und des Ventils ist „OK“ und der Schlecht-Zustand ist „Dirty“. Beim Antrieb gibt es nur den Gut-Zustand „Stable“.

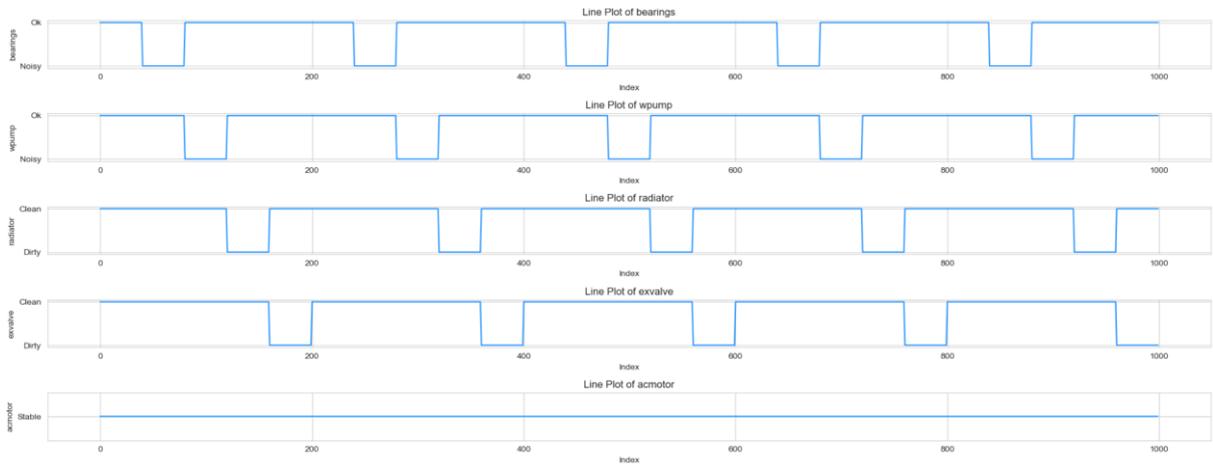


Abbildung 4-8 - Rohdaten des Zustands der Kompressorkomponenten

4.2 Statistische Bewertung des Datensatzes

Zur Erstellung von Modellen in der Datenanalyse müssen die Daten vorverarbeitet werden. Im ersten Schritt der Auswertung wird betrachtet, welche Zustände die relevanten Komponenten haben, die sind in Abbildung 4-9 dargestellt. Beim Zustand der Lager, der Wasserpumpe, des Lüfters und des Ventils sind jeweils 20% der Zeitpunkte in einem Schlecht-Zustand und 80% in einem Gut-Zustand. Beim Antrieb gibt es jedoch nur den Gut-Zustand. Deshalb kann zur Bestimmung des Antriebszustands dieser Datensatz nicht genutzt werden. Das Modell hat keine Punkte, um einen Schlecht-Zustand zu trainieren.

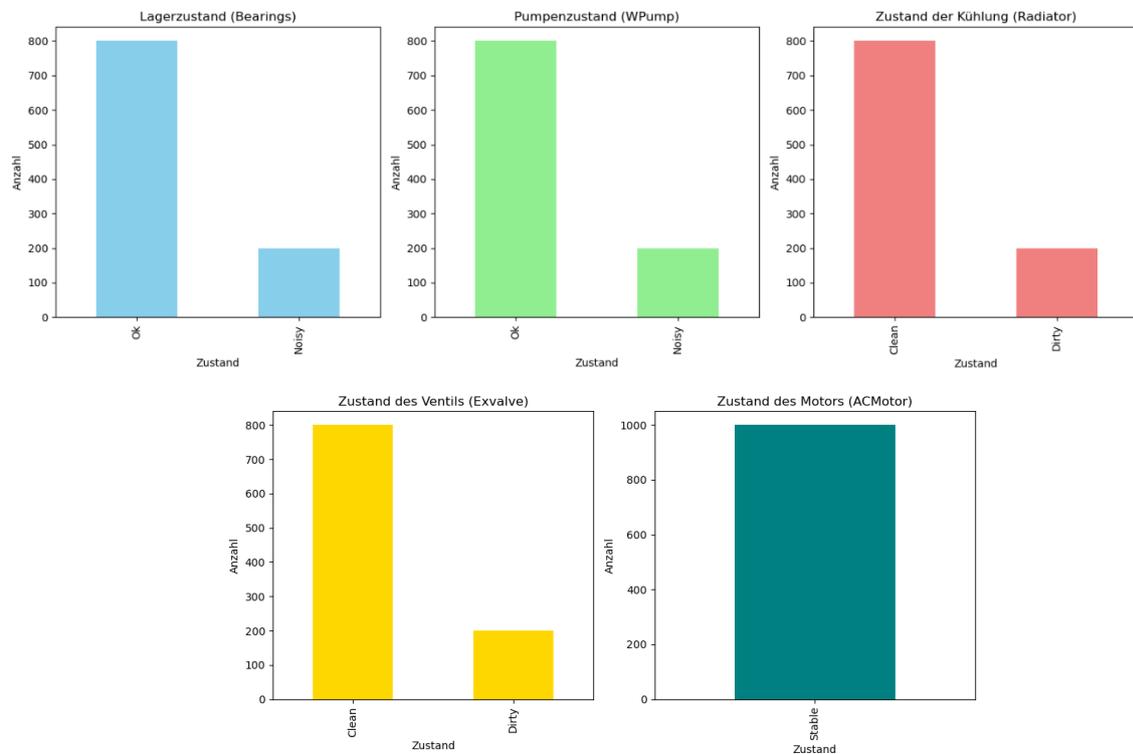


Abbildung 4-9 - Histogramm der Zustände der Kompressorkomponenten

Im nächsten Schritt wird eine Korrelationsanalyse dargestellt, um Zusammenhänge zu entdecken und zu quantifizieren, was besonders nützlich ist, um Muster und Trends in Datensätzen zu identifizieren. Ziel ist es, Vorhersagen zu treffen, indem auf Basis einer bekannten Variable eine andere geschätzt wird. Wichtig ist dabei zu beachten, dass eine Korrelation nicht automatisch eine Kausalbeziehung bedeutet, also nicht direkt darauf schließen lässt, dass eine Variable die andere verursacht. Für die Analyse werden ab jetzt die Zustände der Komponenten mit „1“ als Gut-Zustand und „0“ als Schlecht-Zustand umgewertet, um statistische Aussagen einfacher zu ermöglichen. Auf den ersten Blick wird deutlich, dass die Messwerte untereinander teilweise stark korrelieren. Dies deutet darauf, dass diese untereinander abhängig sind und für eine Analyse nicht alle Messwerte benötigt werden, um eine Aussage treffen zu können. Beispielsweise korrelieren die Eingangs- und Ausgangstemperatur vom Wasser mit einem Wert von 0,98. Physikalisch deutet dies darauf hin, dass es in diesem Datensatz einen festen Zusammenhang zwischen den beiden Temperaturen gibt und für eine statistische Analyse einer der beiden Werte ausreicht. Bei der Betrachtung der Zustände wird deutlich, dass der Zustand des Motors mit keinem Wert korreliert, da es nur einen Zustand gibt. Dieser wird in der weiteren Arbeit nicht mehr betrachtet.

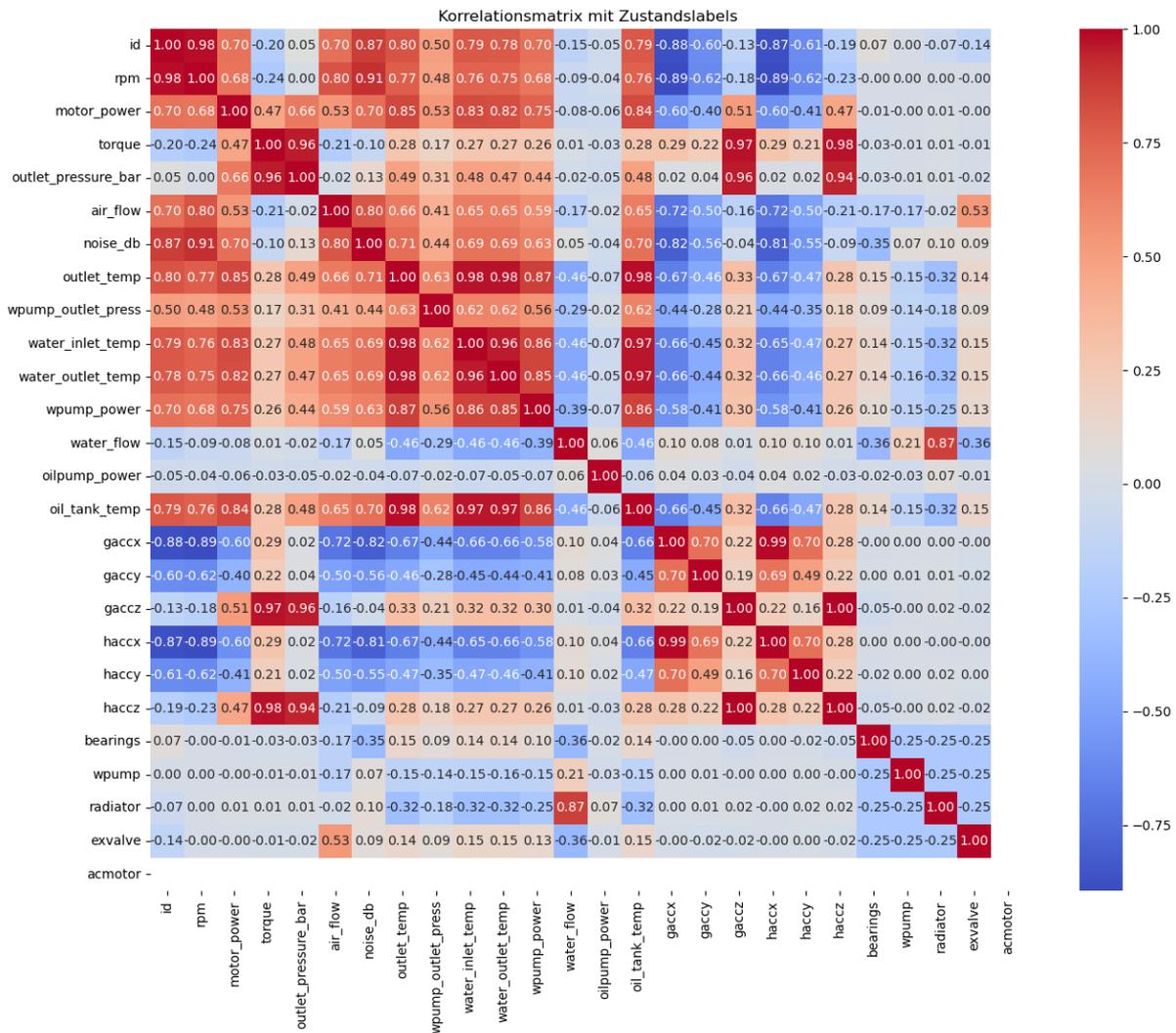


Abbildung 4-10 - Korrelationsanalyse von allen Features und Zuständen

Nach Tabelle 3-1 gilt für Korrelationen unter 0,1, dass es keinen Zusammenhang gibt. Somit wird deutlich, dass die vier Zustände nur mit den neun Features „air-flow“, „noise_db“, „outlet_temp“, „wpump_outlet_press“, „water_inlet_temp“, „water_outlet_temp“, „wpump_power“, „water_flow“ und „oil_tank_temp“ zusammenhängen. Die weiteren statistischen Untersuchungen sollen den ersten Hinweis bestätigen, um die folgende Modellbildung zu erleichtern. Die Vorhersagemodelle werden qualitativ besser, wenn diese nur mit den wirklich relevanten Daten trainiert werden.

Lagerzustand

Aus diesem Grund betrachten wir die verschiedenen Zustände genauer. Die Effekte der verschiedenen Messwerte auf den Zustand der Lager sind in Abbildung 4-11 dargestellt. Die Messwerte hängen nur mit dem Zustand zusammen, wenn der Betrag des Effekts größer ist als 0,1. Somit besteht ein mittlerer Zusammenhang mit noise_db und und water-flow. Ein schwacher Zusammenhang besteht mit air_flow, outlet_temp, water_outlet_temp, water_inlet_temp, oil_tank_temp und wpump_power. Mit den restlichen Messwerten besteht kein Zusammenhang.

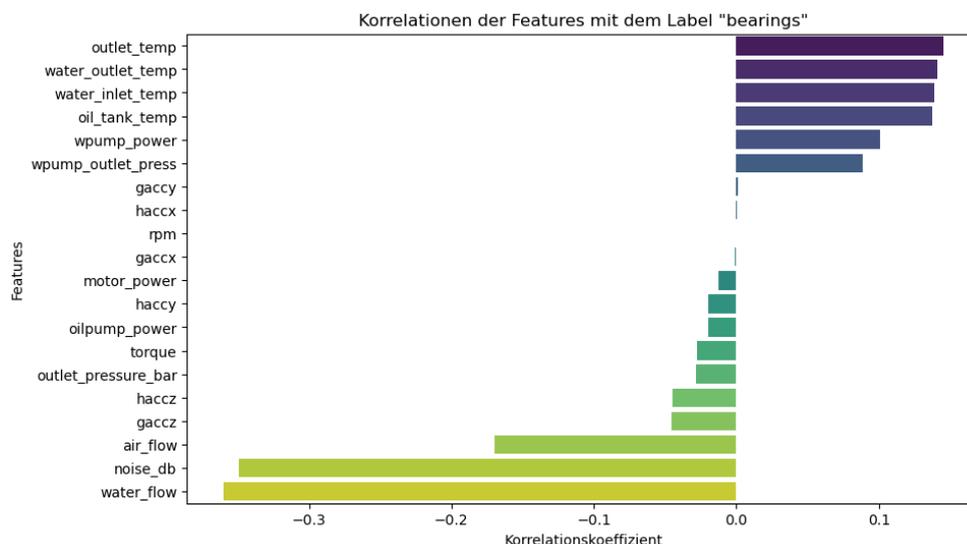


Abbildung 4-11 - Effekte der Messwerte auf den Zustand der Lager

Die reine Korrelation sagt jedoch nicht aus, ob ein Wert tatsächlich kausal zusammenhängt oder zufällig. Aus diesem Grund wird eine F-Wert Analyse der ANOVA Methode durchgeführt. Die ANOVA Methode sagt aus, wie der Beitrag eines bestimmten Faktors zur Gesamtvariabilität der Daten bestimmt wird. Dies geschieht durch den Vergleich der F-Werte von Faktoren und Interaktionen, um ihren Einfluss auf das Gesamtmodell in einer übersichtlichen Form darzustellen. Zusätzlich zu den F-Werten der Faktoren wird auch deren kumulative Summe aufgeführt, um zu bewerten, inwiefern die einflussreichsten Bestandteile des Modells zur Erklärung beitragen.

In Abbildung 4-12 ist der F-Wert für den Lagerzustand dargestellt. Es zeigt sich, dass die Variabilität zu ca. 70% auf die beiden Messwerte water_flow und noise_db.

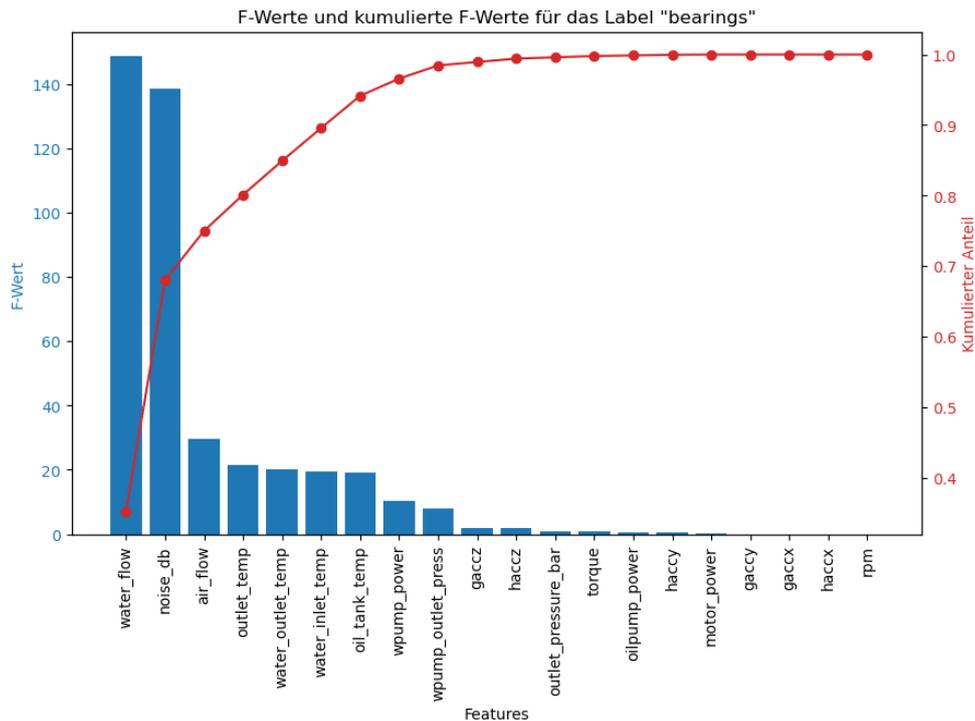


Abbildung 4-12 - Analyse des F-Werts für den Lagerzustand

Um 95% der Variabilität in den Modellen zu berücksichtigen, müssen die Messwerte air_flow, outlet_temp, water_outlet_temp, water_inlet_temp und oil_tank_temp berücksichtigt werden. Im Vergleich zu den Ergebnissen der Effekte, kann also auf den Messwert wpump_power verzichtet werden.

Zustand der Wasserpumpe

Beim Zustand der Wasserpumpe werden ebenfalls die Effekte untersucht. In Abbildung 4-13 sind die Effekte der Messwerte auf den Zustand der Wasserpumpe dargestellt. Hier liegen nur schwache und keine Zusammenhänge vor. Die schwachen Zusammenhänge sind water_flow, noise_db, wpump_outlet_press, water_inlet_temp, oil_tank_temp, wpump_power, outlet_temp, water_outlet_temp und air_flow.

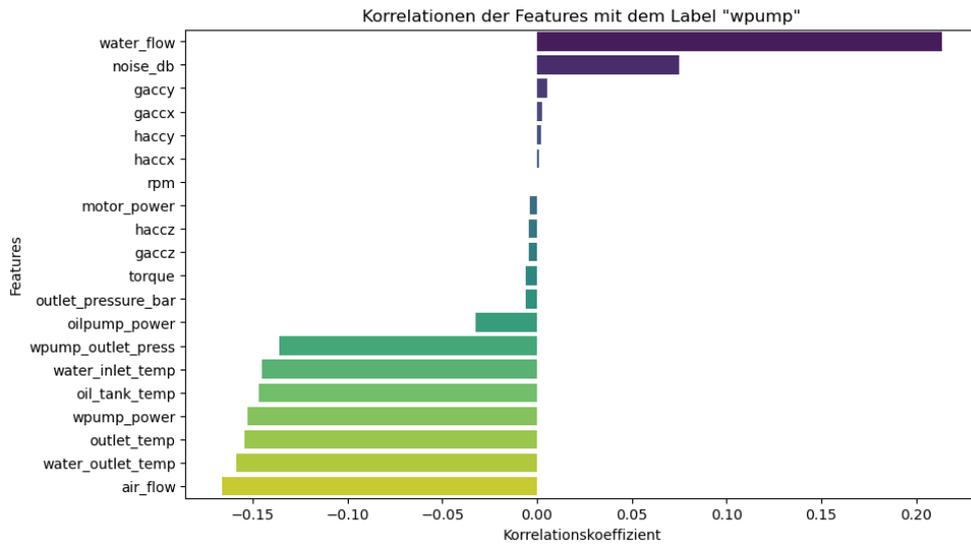


Abbildung 4-13 - Effekte der Messwerte auf den Zustand der Wasserpumpe

Die weitere Analyse des F-Wertes zeigt hier jedoch, dass alle relevanten Daten aus der Analyse der Effekte wichtig sind, um 95 % der Varianz des Zustands der Wasserpumpe zu erklären. In Abbildung 4-14 ist zu erkennen, dass alle oben genannten Werte nötig sind, um die kumulierte Varianz von mindestens 95% zu erreichen.

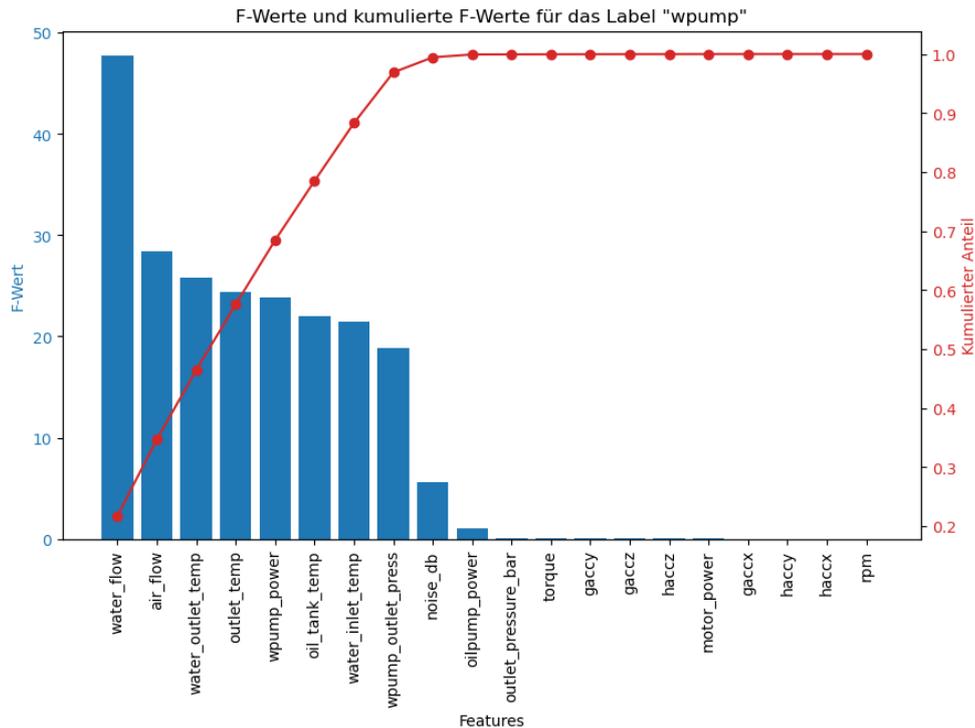


Abbildung 4-14 - Analyse des F-Wertes für den Wasserpumpenzustand

Zustand des Lüfters

Die Effekte der einzelnen Messwerte auf den Zustand des Lüfters sind in Abbildung 4-15 dargestellt. Der water_flow sticht mit einem Betrag von mehr als 0,8 heraus und weist somit einen starken Zusammenhang auf. Keine Daten weisen einen mittleren Zusammenhang auf.

Die Daten aus noise_db, wpump_outlet_press, wpump_power, outlet_temp, water_inlet_temp, water_outlet_temp und oil_tank_temp weisen einen schwachen Zusammenhang auf. Alle weiteren Daten weisen keinen Zusammenhang zum Zustand des Lüfters auf.

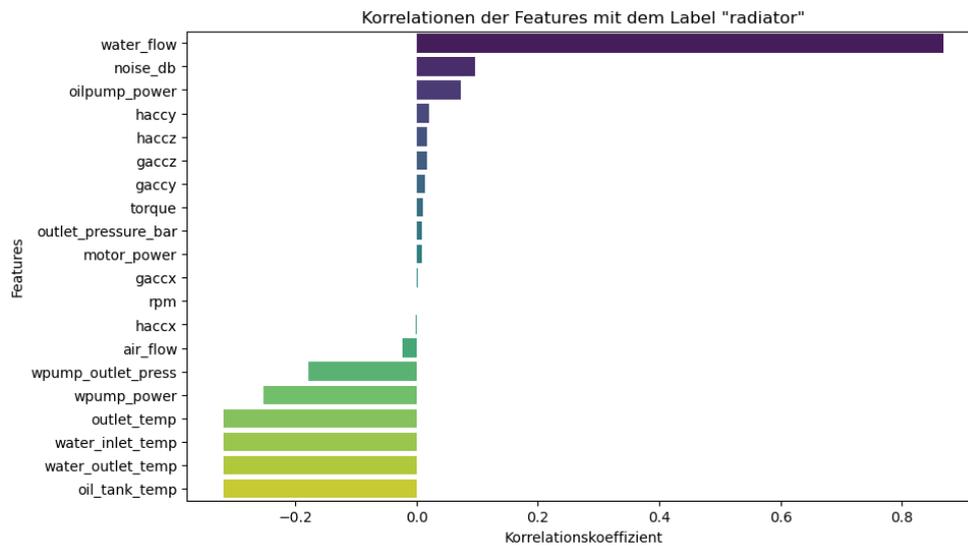


Abbildung 4-15 - Effekte der Messwerte auf den Zustand des Lüfters

Um den Informationsinhalt der Daten zu ermitteln, wird auch hier eine Analyse des F-Wertes durchgeführt. Die Ergebnisse sind in Abbildung 4-16 dargestellt. Hier zeigt sich, dass nur 5 der 7 Messdatenreihen nötig sind, um 95 % der Variabilität darzustellen. Diese sind water_flow, oil_tank_temp, water_inlet_temp, water_outlet_temp und outlet_temp.

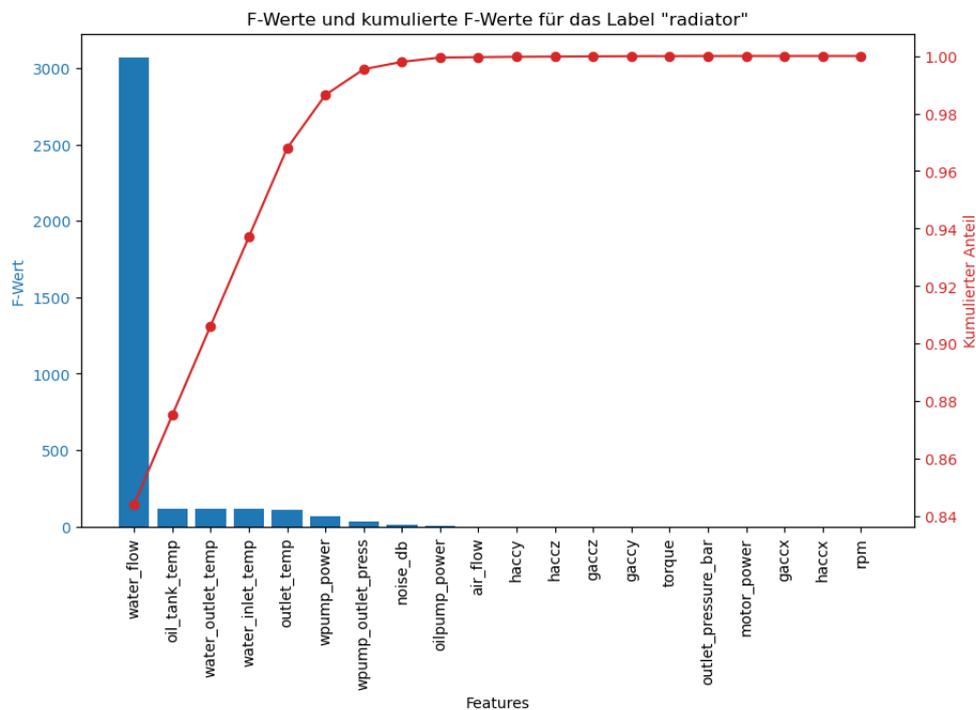


Abbildung 4-16 -Analyse des F-Werts für des Lüfters

Zustand des Ventils

Für den Zustand des Ventils wird ebenfalls eine Effektanalyse durchgeführt, die Ergebnisse sind in Abbildung 4-17 dargestellt. Mit `air_flow` und `water_flow` weisen zwei Messdatenreihen einen mittleren Zusammenhang zum Zustand des Ventils auf. Die Messdatenreihen `water_inlet_temp`, `water_outlet_temp`, `oil_tank_temp`, `outlet_temp` und `wpump_power` weisen einen schwachen Zusammenhang auf. Alle weiteren Messdaten weisen statistisch keinen Zusammenhang zum Zustand des Ventils auf.

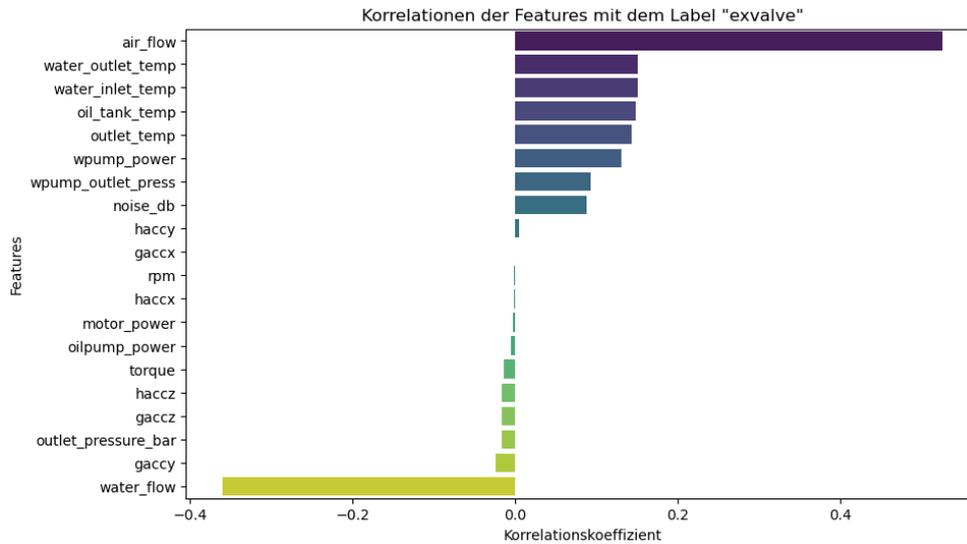


Abbildung 4-17 - Effekte der Messwerte auf den Zustand des Ventils

Um tatsächlich nachzuweisen, welche der Daten, die einen Zusammenhang zum Zustand des Ventils aufweisen, wird eine Analyse des F-Wertes vorgestellt. Die Ergebnisse sind in Abbildung 4-18 dargestellt. Es zeigt sich, dass nur sechs der sieben Messdatenreihen benötigt werden, um den Wert von mehr als 95 % der Variabilität zu erreichen. Diese sind `air_flow`, `water_flow`, `water_inlet_temp`, `water_outlet_temp`, `oil_tank_temp` und `outlet_temp`.

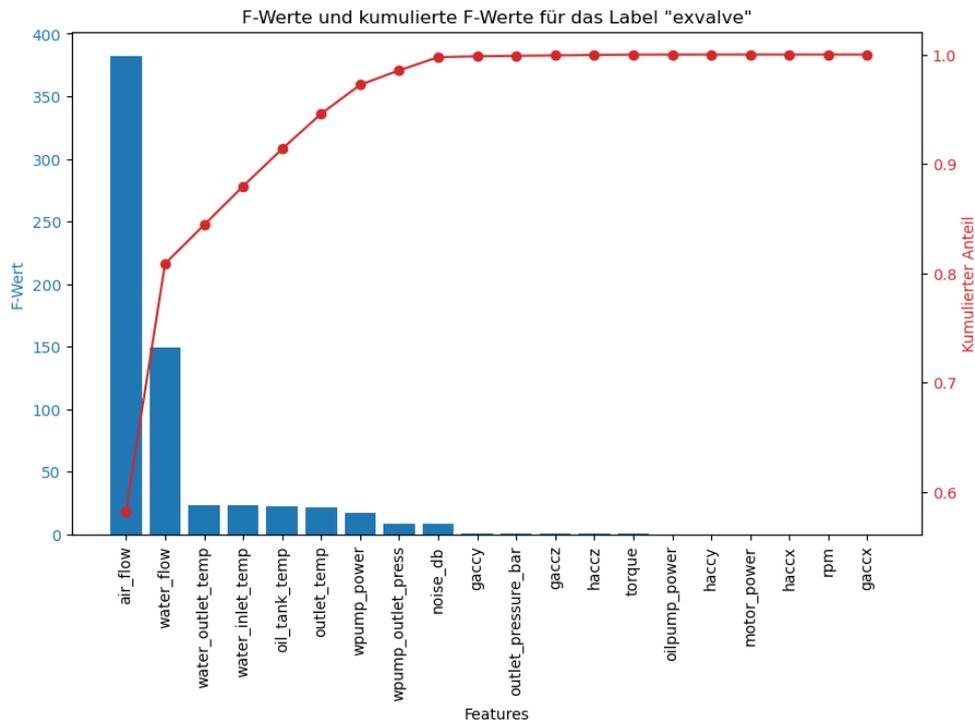


Abbildung 4-18 - Analyse des F-Werts für das Ventil

In Tabelle 4-1 sind die relevanten Komponenten des Kompressors und die neun Messdatenreihen, mit denen es einen Zusammenhang gibt, aufgelistet. Farblich markiert sind die Messdatenreihen, die zur Bewertung des Zustands der jeweiligen Komponente nötig sind. Wie statistisch gezeigt, sind für jede Komponente unterschiedliche Messdaten nötig, um den Zustand zu bestimmen. Die Tabelle lässt es jedoch nicht zu, den Datensatz um weitere Messreihen zu reduzieren, da in jeder Zeile mindestens eine grüne Markierung vorhanden ist und die Messdaten nötig sind, um ein zuverlässiges Modell zu erstellen.

Tabelle 4-1 - Nötige Messwerte zur Bestimmung des Zustands der Komponente

	Lager	Wasserpumpe	Lüfter	Ventil
air-flow				
noise_db				
outlet_temp				
wpump_outlet_press				
water_inlet_temp				
water_outlet_temp				
wpump_power				
water_flow				
oil_tank_temp				

Eine Methode zur Reduzierung des Datensatzes ist die Hauptkomponentenanalyse (englisch: Principal Component Analysis, kurz PCA). Datensätze enthalten eine große Anzahl von Variablen, von denen einige möglicherweise stark korreliert sind. PCA ermöglicht

es, diese hochdimensionalen Daten in eine kleinere Anzahl von Dimensionen (Hauptkomponenten) zu transformieren, während ein Großteil der in den Daten enthaltenen Varianz erhalten bleibt. Dies kann besonders nützlich sein, um die Komplexität von Modellen zu reduzieren. Die Ergebnisse der PCA des Datensatzes ist in Abbildung 4-19 dargestellt. Um wie bisher eine Variabilität von 95 % zu erreichen, werden nicht wie bisher neun sondern lediglich acht verschiedene Datenreihen benötigt, um Modelle mit der nötigen Genauigkeit zu trainieren. Das Diagramm zeigt, dass die Schwelle von 95 % der Variabilität ab der achten Datenreihe erreicht wird.

Um die Komplexität in den folgenden Abschnitten zu reduzieren werden für die weiteren Analysen und das Trainieren der Modelle die acht Hauptkomponentenvektoren verwendet.

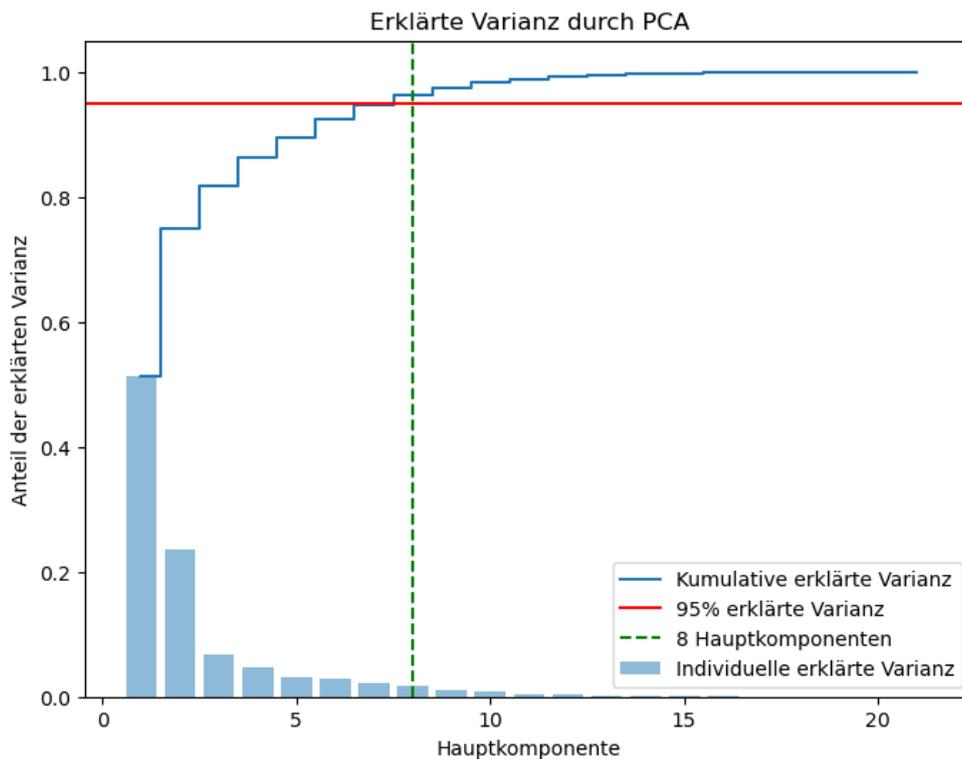


Abbildung 4-19 - Hauptkomponentenanalyse des Datensatzes

Abbildung 4-20 zeigt das Vorgehen von Verarbeitungsschritten in der Datenvorbereitung in diesem Abschnitt, beginnend mit dem Laden der Daten. Anschließend erfolgt eine Bearbeitung der Labels, welche für das Training des Modells verwendet werden. Hier wird im Wesentlichen der Zustand des Labels aus Textform in 1 für Gut-Zustand und 0 für Schlecht-Zustand geändert. Um die Komplexität der Daten zu reduzieren, werden mehrere Analysen durchgeführt. Zuerst wird eine Korrelationsanalyse vorgenommen, um die Beziehungen zwischen den verschiedenen Merkmalen zu verstehen. Ergebnis, waren der Ausschluss des Label AC Motors, da hier keine Korrelationen vorlagen, sowie die ersten Hinweise, dass einige Features für die Analyse nicht relevant sind. Darauf folgt eine Effektanalyse, die die Auswirkungen der Merkmale auf die Zielvariable untersucht, sowie eine Varianzanalyse, die die Variabilität der Daten betrachtet. Schließlich wird eine Hauptkomponentenanalyse durchgeführt, um die Dimensionalität der Daten zu reduzieren und die wichtigsten Merkmale zu extrahieren. Am Ende dieses Prozesses stehen acht Hauptmerkmale und vier Labels, die den Zustand eines Kompressors beschreiben, wobei

für jedes dieser Hauptmerkmale und Labels jeweils 1000 Datenpunkte zur Verfügung stehen. Diese methodische Herangehensweise zielt darauf ab, die Daten so aufzubereiten, dass sie für effektive maschinelle Lernverfahren geeignet sind.

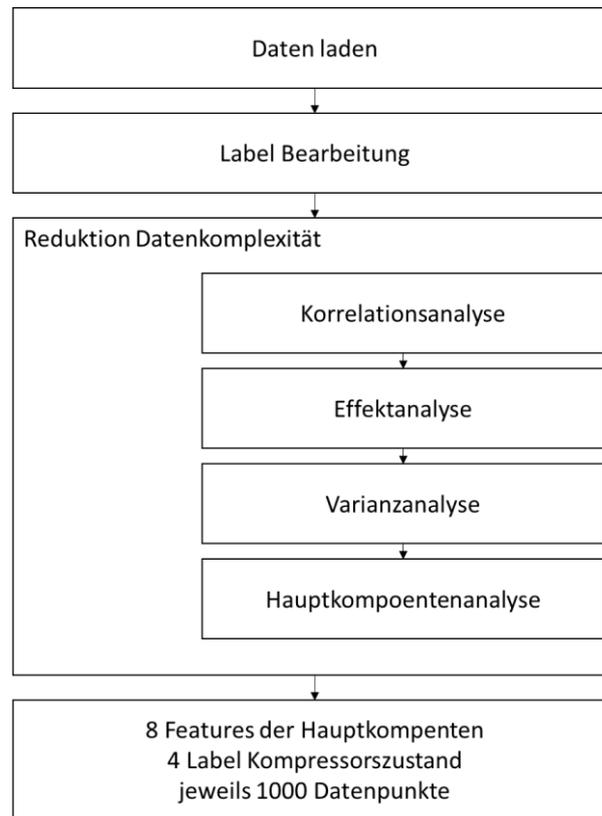


Abbildung 4-20 - Vorgehen in der statistischen Analyse

4.3 Datenanalyse

Für die Datenanalyse durch maschinelles Lernen wird eine umfangreiche Datengrundlage verwendet, die sich aus 8 Hauptkomponenten (Features) der Hauptkomponentenanalyse (PCA) zusammensetzt. Zusätzlich wurden 4 Labels des Kompressorzustands berücksichtigt, mit jeweils 1000 Datenpunkten für jede Gruppe. Diese Konstellation bildet die Basis für die Untersuchung der Maschinenzustände und ermöglicht eine detaillierte Analyse der Performance und möglicher Ausfallrisiken.

Der erste Schritt der Analyse bestand in der Aufteilung der Daten in Trainings- und Testsets, um sowohl eine effektive Anlernphase des Modells zu ermöglichen als auch eine valide Bewertung der Modellperformance auf unbekanntem Daten zu gewährleisten. Hierbei wurde ein Verhältnis von 70% der Daten für das Training und 30% für das Testen gewählt. Diese Aufteilung folgt den gängigen Praktiken im maschinellen Lernen, welche betonen, dass eine ausgewogene Verteilung zwischen Trainings- und Testdaten entscheidend ist, um Overfitting zu vermeiden und die Generalisierbarkeit des Modells zu erhöhen.

Die 8 Features der Hauptkomponentenanalyse wurden ausgewählt, um die Dimensionalität der Daten zu reduzieren und dabei die wichtigsten Informationen zu bewahren, die für die Vorhersage des Kompressorzustands relevant sind. Durch diese Reduktion wird die Effizienz des Lernprozesses gesteigert und die Rechenlast verringert, ohne signifikante Einbußen bei der Vorhersagegenauigkeit hinnehmen zu müssen. Die 4 Zustandslabels des

Kompressors bieten dabei eine klare Zielvariable für die Klassifizierungsaufgabe und ermöglichen eine präzise Bewertung der Modellleistung in Bezug auf die Zustandserkennung und -vorhersage.

Insgesamt bildet die sorgfältige Vorbereitung der Daten, einschließlich der Feature-Auswahl durch PCA und der sinnvollen Aufteilung in Trainings- und Testsets, eine solide Grundlage für die nachfolgenden Schritte des maschinellen Lernens. Diese methodische Herangehensweise gewährleistet, dass das entwickelte Modell nicht nur auf den Trainingsdaten gut performt, sondern auch eine hohe Vorhersagekraft auf neuen, unbekanntem Daten zeigt.

Polynomiale Regression

Im nächsten Schritt der Analyse wurden die Daten mit Hilfe der polynomialen Regression analysiert, wobei ein Polynom zweiten Grades zur Modellierung verwendet wurde. Dieser Ansatz wurde gewählt, da er es ermöglicht, nicht nur lineare Beziehungen zwischen den Variablen zu erfassen, sondern auch quadratische Beziehungen, die eine Krümmung in den Daten anzeigen können. Dies ist besonders nützlich, um komplexere Zusammenhänge zwischen den Features und den Labels zu identifizieren, die in realen Anwendungen häufig auftreten. Für jedes der vier Labels – 'bearings', 'pump', 'radiator' und 'valve' – wurde eine separate polynomiale Regression durchgeführt. Das Ziel war es, individuelle Modelle zu erstellen, die die spezifischen Muster und Trends jedes Labels genau abbilden und dabei die besonderen Eigenschaften jedes Kompressorzustands berücksichtigen.

In Abbildung 4-21 ist das Ergebnis der polynomialen Regression dargestellt. Hier sind die Vorhersagen über den tatsächlichen Werten dargestellt. Es ist zu erkennen, dass die Vorhersagen um den tatsächlichen Wert streuen, aber auf den ersten Blick auch eine relevante Vorhersagegüte zu erkennen. Besonders ist, dass die Labels nur den Wert 0 und 1 aufweisen, die Vorhersagen aber auch andere Werte.

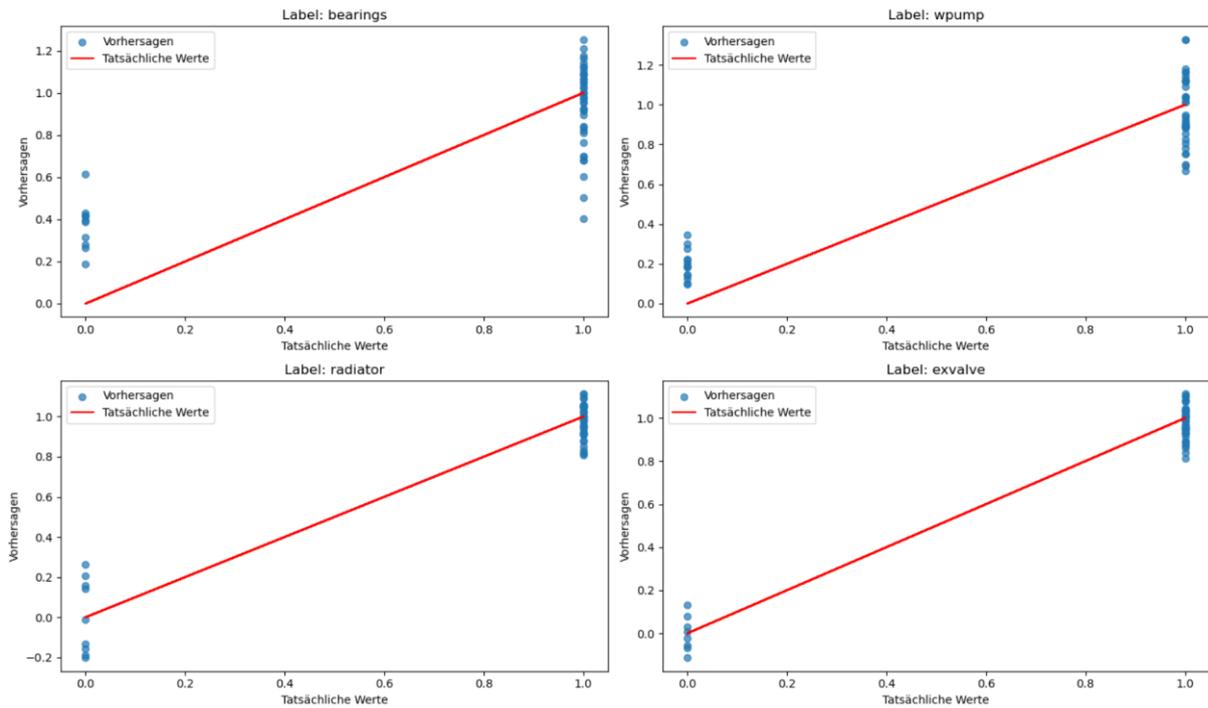


Abbildung 4-21 - Ergebnis der polynomialen Regression Vorhersagen im Vergleich zu realen Daten

Um die genauen Werte der Vorhersage genauer zu betrachten, sind in Abbildung 4-22 die vorhergesagten Werte und die tatsächlichen Werte über die Zeit geplottet. Es wird noch deutlichen, wie sich die Vorhersagen zwischen -0,75 und 1,75 je nach vorhergesagtem Label bewegen. Für die Anwendung im Kompressor sind jedoch die lediglich die vorhergesagten Zustände der Kompressorkomponenten relevant. Der naheliegende Schritt ist es, die Werte kleiner 0,5 als Schlecht-Zustand also 0 und die Werte größer oder gleich 0,5 als Gutzustand, also 1 zu verändern. Auf den ersten Blick scheinen in den Diagrammen diese Aufteilungen eine gute Vorhersage darzustellen.

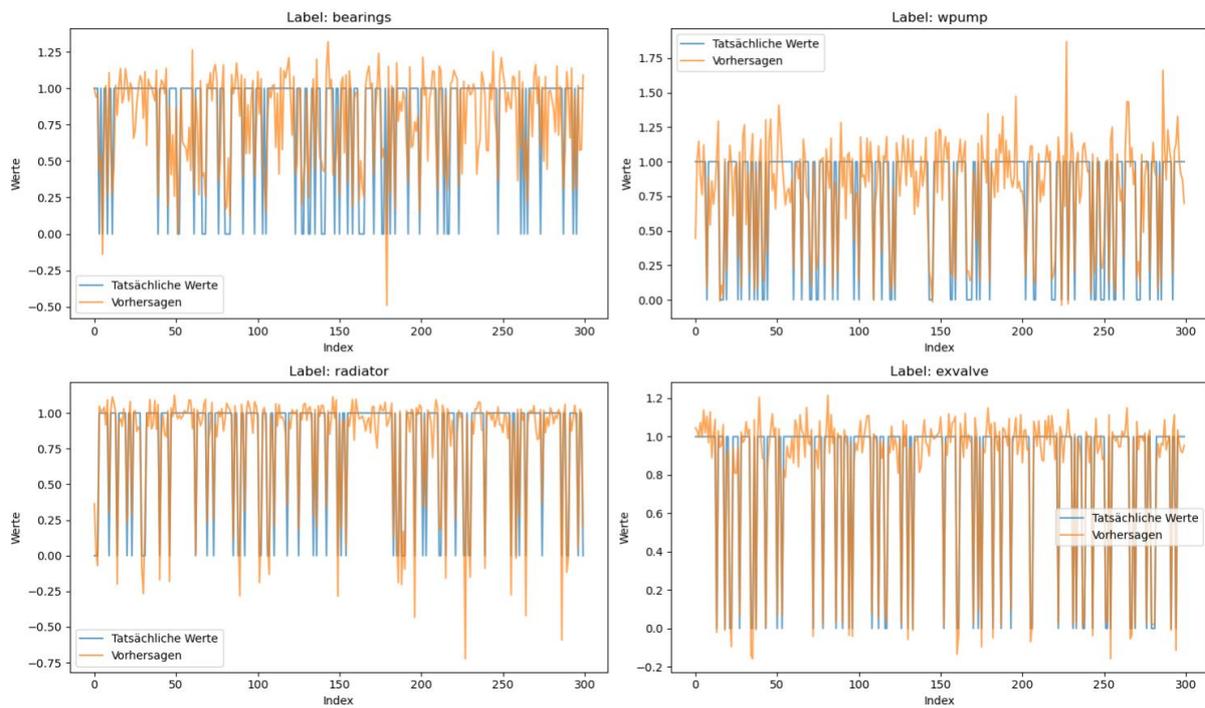


Abbildung 4-22 - Ergebnis der polynomialen Regression und tatsächliche Werte im zeitlichen Verlauf
Nach der Vorschrift

$$y_{pred} = \begin{cases} y_{pred} < 0,5 & y_{pred} = 0 \\ y_{pred} \geq 0,5 & y_{pred} = 1 \end{cases} \quad (41)$$

werden die Vorhersagen y_{pred} angepasst und die Ergebnisse sind in Abbildung 4-23 dargestellt. Auf den ersten Blick wird deutlich dass es keine Streuung mehr gibt, die Vorhersagen liegen auf 0 oder 1. Für die Vorhersagen der Label Lüfter und Ventil sind alle Vorhersagen korrekt. Bei den Lagern gibt es falsch positive sowie falsch negative und bei der Wasserpumpe falsch negative Vorhersagen. Die genaue Bewertung des Algorithmus im Vergleich zu den folgenden erfolgt jedoch erst im nächsten Abschnitt. Hier werden die Statistischen Fehler der verschiedenen Vorhersagen miteinander verglichen und anschließend eine Auswahl vorgenommen.

Polynomiale Regressionsergebnisse (Grad 2)

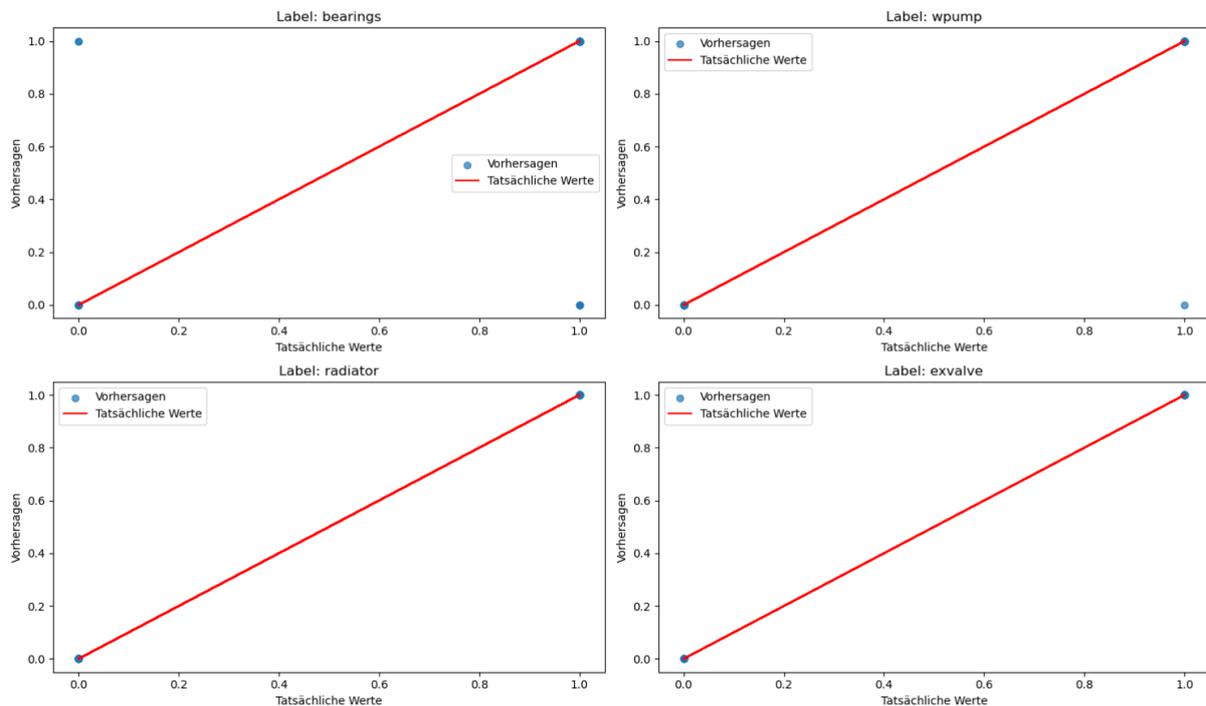


Abbildung 4-23 - Ergebnis der polynomialen Regression mit angepassten Vorhersagen

Random Forest Regression

Nun wird der Random Forest-Algorithmus als regressionsanalytisches Verfahren herangezogen, um dessen Eignung und Vorhersagequalität im Vergleich zur polynomialen Regression zu bewerten. Random Forest ist ein Ensemble-Lernverfahren, das durch die Kombination mehrerer Entscheidungsbäume die Variabilität einzelner Modelle reduziert und somit robustere Vorhersagen ermöglicht. Für jedes der vier Labels – 'bearings', 'pump', 'radiator', und 'valve' – wurde ein eigener Random Forest-Regressor trainiert. Die Parameter des Algorithmus wurden sorgfältig gewählt, um eine Überanpassung zu vermeiden und eine hohe Generalisierungsfähigkeit zu gewährleisten.

Bei der Auswertung der durch Random Forest erzielten Ergebnisse, die in Abbildung 4-24 dargestellt sind, wird jedoch deutlich, dass die Vorhersagegenauigkeit leicht hinter der der polynomialen Regression zurückbleibt. Die Vorhersagen des Random Forest-Modells für die Labels 'bearings' und 'pump' zeigen eine größere Streuung um die tatsächlichen Werte, was auf eine geringere Modellgenauigkeit hinweist.

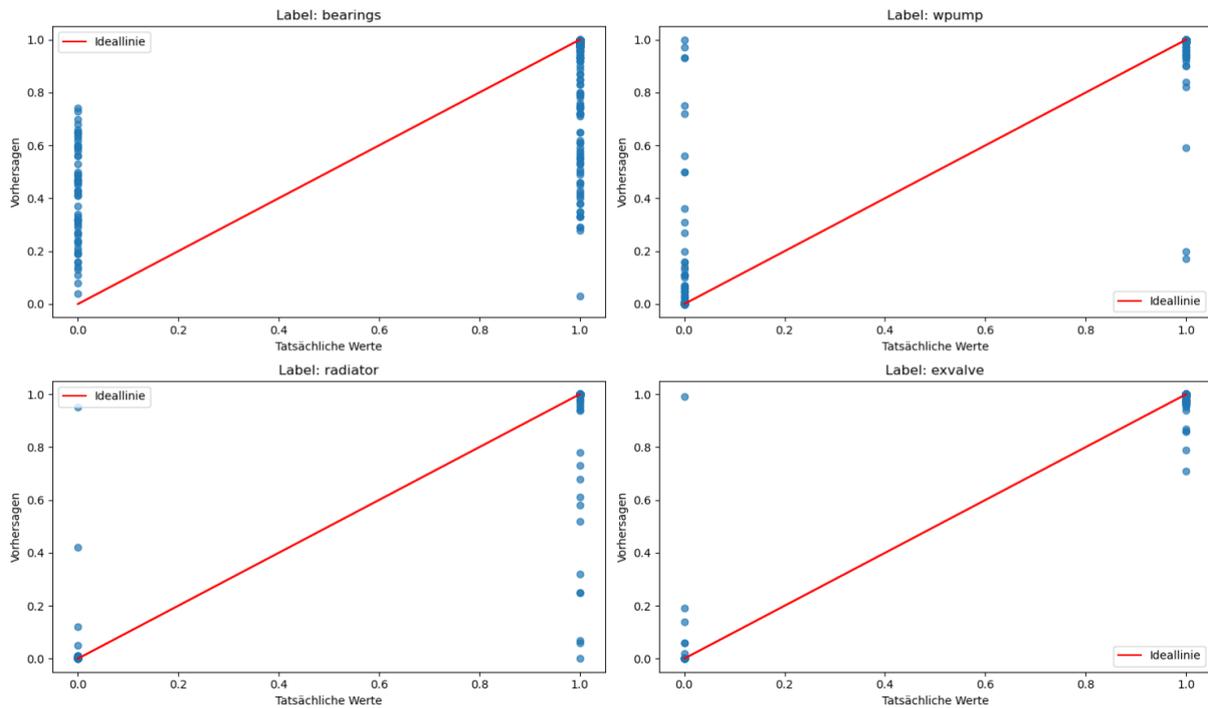


Abbildung 4-24 - Ergebnisse der Random Forest Regression Vorhersagen über tatsächlichen Werten

Auch hier wird das Verfahren aus Gleichung 41 angewendet, um die Vorhersagen auf die Zustände der Kompressorkomponenten zu reduzieren. Bei den vorhergesagten Werten des Random Forest-Modells zeigt sich eine höhere Fehlerrate im Vergleich zur polynomialen Regression, was in den Diagrammen in Abbildung 4-25 durch eine größere Anzahl inkorrekt zugeordneter Kategorien 0 oder 1 erkennbar ist.

Es wird sofort deutlich, dass bei den Zuständen der Lager, der Wasserpumpe und des Lüfters falsch positive und falsch negative Vorhersagen vorliegen. Lediglich bei der Vorhersage des Ventils gibt's es keine falsch negativen Vorhersagen. Auch hier wird die genaue Bewertung des Algorithmus im Vergleich zu den anderen Modellen im nächsten Abschnitt vorgenommen.

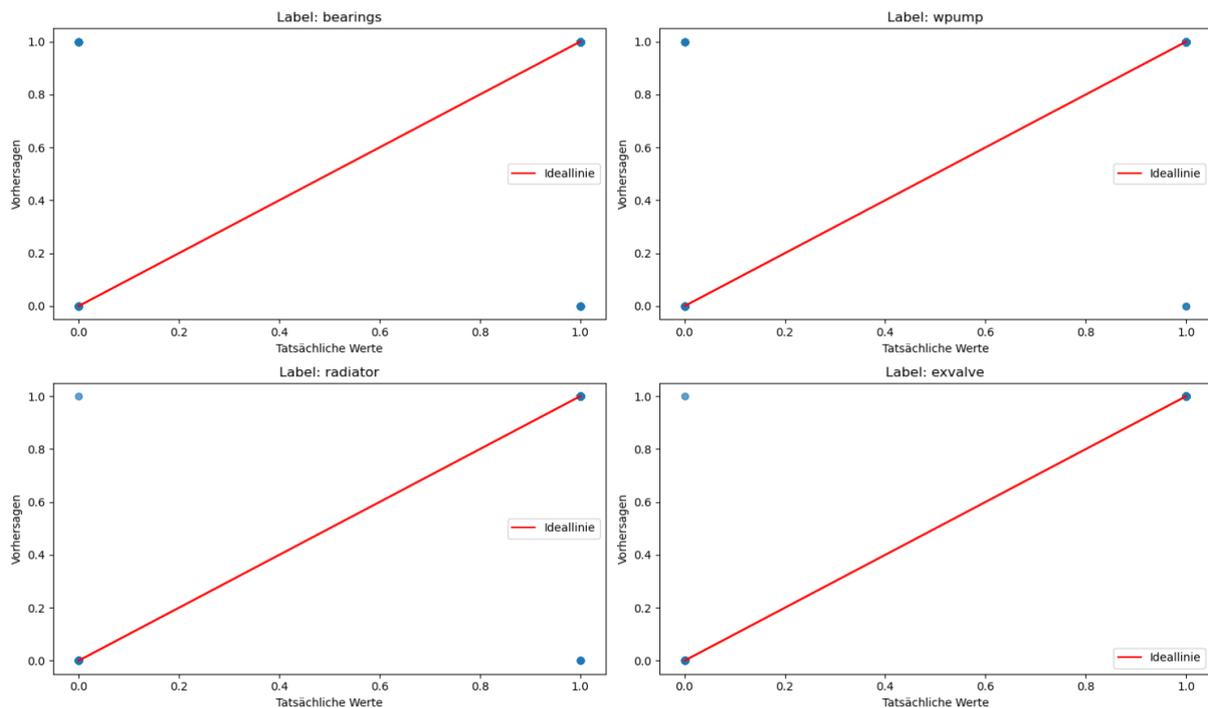


Abbildung 4-25 Angepasste vorhergesagte und tatsächliche Werte der Random Forest Regression

Neuronales Netz

Die finale Vorhersagemethode für die Zustände des Kompressors wurde mithilfe eines neuronalen Netzes realisiert. Neuronale Netze sind für ihre Fähigkeit bekannt, komplexe Muster und nichtlineare Beziehungen in Daten zu erkennen, was sie zu einem idealen Werkzeug für präzise Vorhersagen in einer Vielzahl von Anwendungen macht, insbesondere in Situationen, in denen traditionelle statistische Methoden an ihre Grenzen stoßen. Die Diagramme in Abbildung 4-26, die die Vorhersagen des neuronalen Netzes vor der Anpassung zeigen, ist zu erkennen, dass die Vorhersagewerte bereits eng an den tatsächlichen Werten liegen, mit einer geringen Streuung und ohne extreme Ausreißer, die teilweise in den Ergebnissen der anderen Modelle beobachtet wurden.

Nach der Anwendung der Regel, Vorhersagewerte unter 0,5 als 0 und solche von 0,5 oder höher als 1 zu klassifizieren, zeigen die Diagramme, dass die Vorhersagen nahezu perfekt mit den tatsächlichen Zuständen der Kompressorlabels 'bearings', 'pump', 'radiator', und 'valve' übereinstimmen. Die vorherige Streuung wurde eliminiert, und die Vorhersagen liegen präzise bei den Werten 0 oder 1, was darauf hindeutet, dass das neuronale Netzwerk effektiv zwischen den Zuständen differenzieren kann.

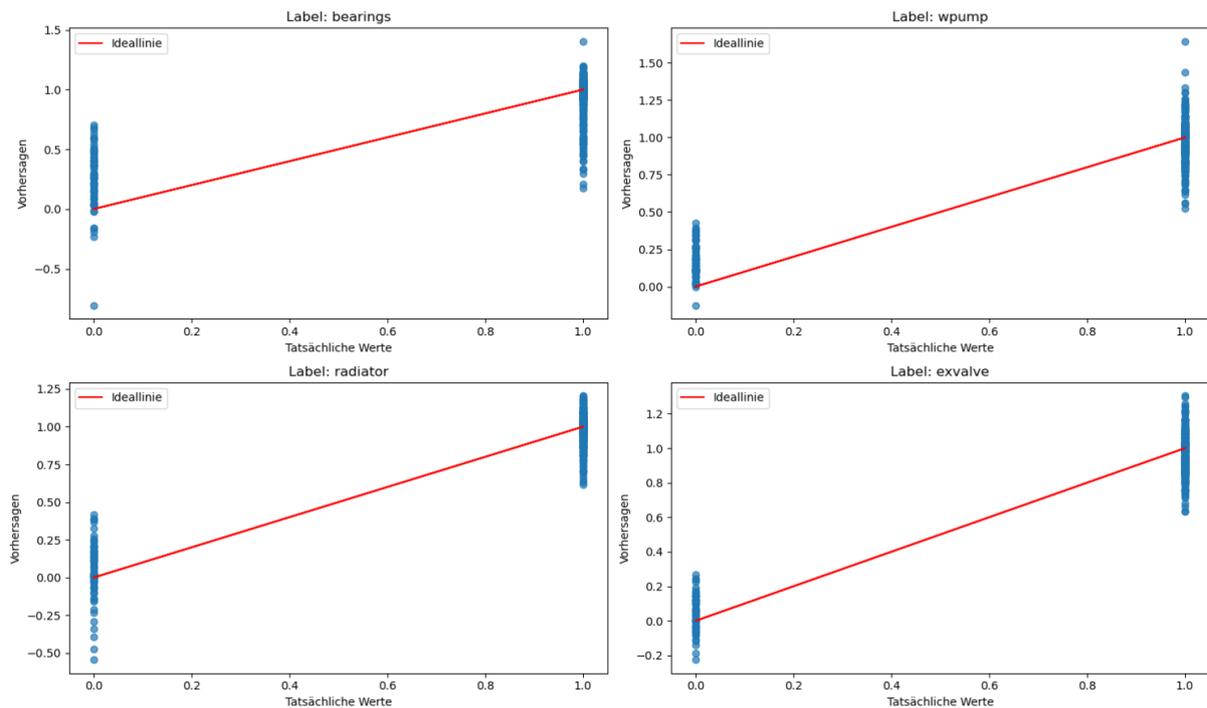


Abbildung 4-26 Ergebnisse des neuronalen Netzes Vorhersagen über tatsächlichen Werten

Diese herausragende Leistung des neuronalen Netzes wird in Abbildung 4-27, dass die Vorhersagen nach der Anpassung zeigen, weiter bestätigt, wo die klare Trennung der Zustände ohne jegliche Streuung ersichtlich ist. Die Genauigkeit der Vorhersagen ist somit signifikant höher als die der anderen Modelle, und es gibt kaum Fehlzuweisungen. Insbesondere bei den kritischen Komponenten Wasserpumpe, Lüfter und Ventil sind die Vorhersagen des neuronalen Netzes fehlerfrei, was auf eine hohe Zuverlässigkeit in der praktischen Anwendung schließen lässt. Lediglich bei den Lagern gibt es falsch positive und falsch negative Vorhersagen

Die Effektivität des neuronalen Netzes wird im nachfolgenden Abschnitt noch genauer untersucht, indem die statistischen Fehler der Vorhersagen detailliert analysiert und mit den Ergebnissen der vorherigen Modelle verglichen werden.

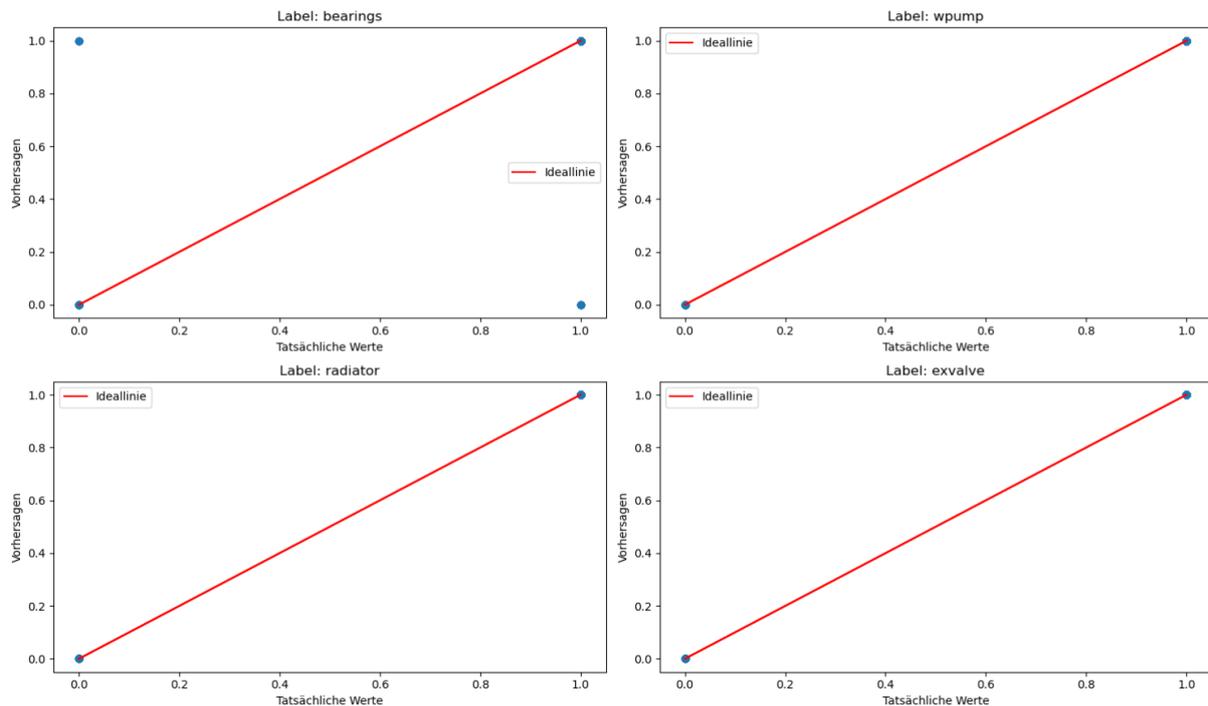


Abbildung 4-27 - Angepasste vorhergesagte und tatsächliche Werte des neuronalen Netzes

Abbildung 4-28 skizziert das methodische Vorgehen bei der Analyse und Vorhersage des Zustandes von Kompressorkomponenten mithilfe maschinellen Lernens in diesem Abschnitt. Die grundlegende Datengrundlage besteht aus 8 Features, die aus einer Hauptkomponentenanalyse (PCA) gewonnen wurden, und 4 Labels, die den Zustand des Kompressors beschreiben. Für jedes Label liegen 1000 Datenpunkte vor.

Im ersten Schritt werden die Daten aufgeteilt. Die gesammelten Daten werden zunächst in zwei Gruppen aufgeteilt: 70 % der Daten werden für das Training des Modells verwendet, während die restlichen 30 % als Testdaten dienen. Diese Aufteilung ist ein Standardverfahren im maschinellen Lernen und dient dazu, das Modell mit einem Teil der Daten zu trainieren und seine Leistung anschließend anhand der restlichen Daten zu bewerten. Anschließend werden die Modelle trainiert. Mit dem Trainingsdatensatz werden drei verschiedene Arten Polynomiale Regression, Random Forest Regression und das neuronale Netz trainiert. Die polynomiale Regression ist ein Modell, das versucht, die Beziehung zwischen den Datenpunkten mit einem Polynom zweiten Grades zu modellieren. Random Forest ist ein Ensemble-Lernverfahren, das viele Entscheidungsbäume nutzt, um die Vorhersagegenauigkeit zu verbessern. Das neuronale Netz ist ein komplexes Modell, das in der Lage ist, nichtlineare Muster in den Daten durch eine Reihe von Neuronen und Schichten zu erfassen und zu lernen. Nach dem Training werden die Modelle mit den Testdaten geprüft. Hierbei wird analysiert, wie gut jedes Modell in der Lage ist, die tatsächlichen Datenpunkte vorherzusagen. Dies ist ein entscheidender Schritt, um die Leistungsfähigkeit und Generalisierbarkeit der Modelle zu beurteilen. Da die Labels des Kompressors binär sind (also die Werte 0 für einen schlechten Zustand und 1 für einen guten Zustand haben), müssen die kontinuierlichen Vorhersagewerte der Modelle in binäre Werte umgewandelt werden. Dafür wird eine Schwelle definiert – üblicherweise bei 0,5 –, um die Vorhersagen entsprechend zu klassifizieren. Schließlich werden die Ergebnisse der Vorhersagen in einer geeigneten Form visualisiert und dargestellt. Dies ermöglicht es, die

Leistung der verschiedenen Modelle direkt miteinander zu vergleichen und zu bewerten, welche Methode am besten für die Vorhersage der Zustände der Kompressorkomponenten geeignet ist.

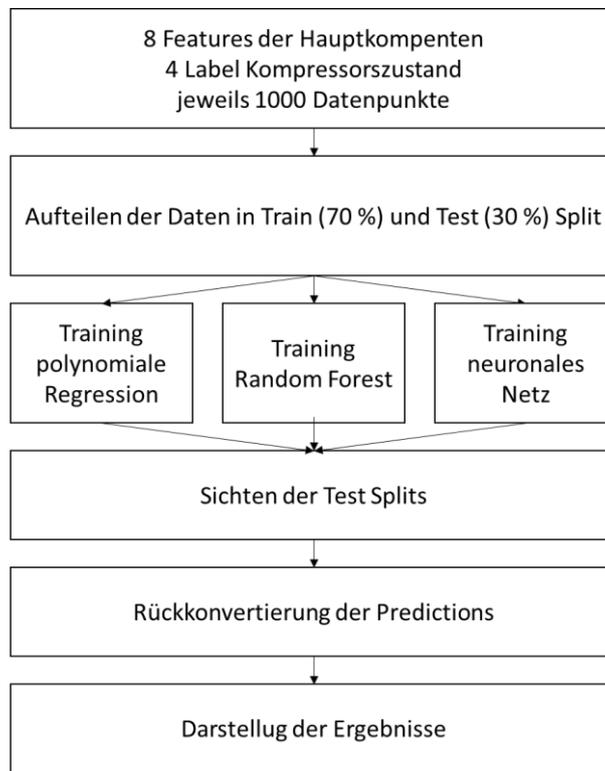


Abbildung 4-28 - Vorgehen in der Datenanalyse

4.4 Bewertung

In diesem Abschnitt werden die Ergebnisse der Datenanalyse miteinander verglichen. Hierzu werden die vorgestellten Bewertungstools genutzt. In Tabelle 4-2 In Tabelle 4-3 sind Fehlerkennwerte der Modelle mit den angepassten Daten dargestellt. Hier ist, wie sich in den Diagrammen bereits angedeutet hat, das neuronale Netz im Vergleich zu Random Forest und polynomiale Regression der beste Algorithmus. Bei der Wasserpumpe, dem Lüfter und dem Ventil sind die Vorhersagen Perfekt. Es gibt keine Fehler und auch bei der Vorhersage des Lagerzustands weist das neuronale Netz den geringsten Fehler auf.

Tabelle 4-3 sind die MAE und MSE aus Kapitel 3.3 vorgestellt. Aufgrund der geringen Fehler und eindeutigen Ergebnisse wird auf die Darstellung des RMSE verzichtet. Diese Fehlerkennwerte sind noch für die Ergebnisse vor der Anpassung der Vorhersagen dargestellt. Hier zeigt sich, dass die einfache polynomiale Regression im Vergleich die geringsten Fehler aufweist.

Tabelle 4-2 - Vergleich der statistischen Fehler vor Anpassung der Vorhersagen

	Polynomiale Regression MAE/MSE		Random Forest MAE/MSE		Neuronales Netz MAE/MSE	
Lager	0,21	0,08	0,41	0,08	0,42	0,06
Wasserpumpe	0,16	0,04	0,44	0,03	0,44	0,03
Lüfter	0,09	0,02	0,42	0,02	0,42	0,02
Ventil	0,06	0,01	0,06	0,01	0,06	0,01

In Tabelle 4-3 sind Fehlerkennwerte der Modelle mit den angepassten Daten dargestellt. Hier ist, wie sich in den Diagrammen bereits angedeutet hat, das neuronale Netz im Vergleich zu Random Forest und polynomiale Regression der beste Algorithmus. Bei der Wasserpumpe, dem Lüfter und dem Ventil sind die Vorhersagen Perfekt. Es gibt keine Fehler und auch bei der Vorhersage des Lagerzustands weist das neuronale Netz den geringsten Fehler auf.

Tabelle 4-3 - Vergleich der statistischen Fehler nach Anpassung der Vorhersage

	Polynomiale Regression MAE/MSE		Random Forest MAE/MSE		Neuronales Netz MAE/MSE	
Lager	0,08	0,08	0,13	0,13	0,07	0,07
Wasserpumpe	0,01	0,01	0,04	0,04	0	0
Lüfter	0	0	0,02	0,02	0	0
Ventil	0	0	0,01	0,01	0	0

5 FAZIT UND AUSBLICK

Abschließend wird in diesem Kapitel ein Fazit zur Arbeit gezogen und ein Ausblick für weites Vorgehen gegeben.

5.1 Fazit

Die Analyse der vorhandenen Daten der Sensoren des Kompressors hat ein Modell ergeben, welches es ermöglicht mit hoher Genauigkeit den Zustand der Lager, des Lüfters, der Wasserpumpe und des Ventils vorherzusagen. Eine Vorhersage des Zustands des Motors ist nicht möglich. In Abbildung 5-1 ist das Vorgehen gezeigt, das mit neuen Daten aus dem Betrieb des Kompressors durchgeführt wird, um auf den Zustand der einzelnen Komponenten zu schließen. Nach der Erfassung der Rohdaten werden mit der durchgeführten Hauptkomponentenanalyse die Rohdaten wieder in die Hauptkomponenten umgewandelt. Mit den 8 ersten Hauptkomponenten wird das trainierte neuronale Netz genutzt. Das Ergebnis des Netzes wird noch mit der Methode größer kleiner 0,5 in den Gut- bzw. Schlecht-Zustand der einzelnen Komponenten umgewandelt. Mit diesem Ergebnis kann der Anlagenbetreiber nun seine Wartung und Instandhaltung planen.

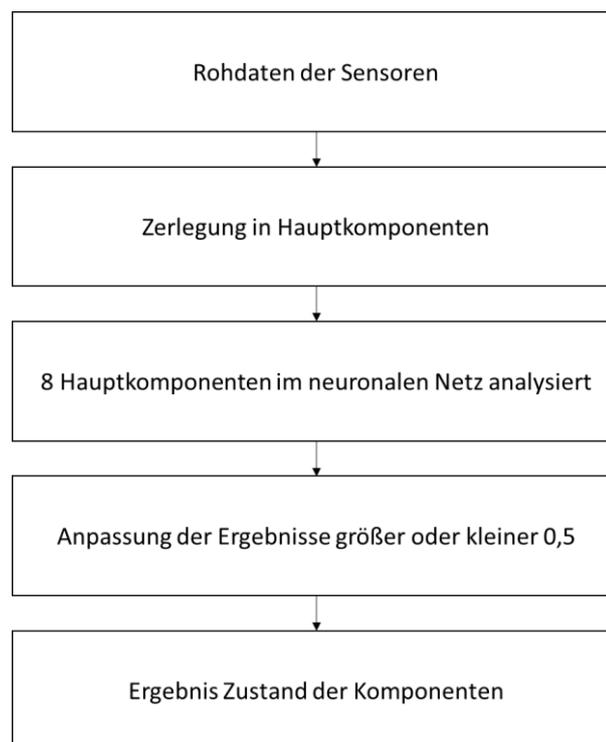


Abbildung 5-1 - Vorgehen bei der Zustandsbewertung

Die Vorhersagegenauigkeit des Modells ist überdurchschnittlich gut. Aus diesem Grund ist die größte Hürde bei der Einführung von KI-basiertem Condition Monitoring vergleichsweise leicht, nämlich den Anwender zu überzeugen den Analysen zu vertrauen und einer tatsächlichen Umsetzung zuzustimmen. Der einzig negative Punkt

ist es, dass keine Aussage über den Zustand aufgrund fehlender Schlecht-Daten möglich ist.

5.2 Ausblick

Nach der erfolgreichen Umsetzung der Datenanalyse steht nun die physische Umsetzung an. Das heißt, dass der Kompressor des Kunden nun mit einer Liveanalyse der Daten ausgestattet wird und auf Basis der Vorhersageergebnisse die Wartung und Instandhaltung geplant wird. In der ersten Betriebszeit muss jedoch eine parallele Überwachung und Datenerhebung stattfinden, um einerseits die Zuverlässigkeit der Vorhersage zu prüfen und andererseits mehr Daten zu haben. Mit Daten zum Schlecht-Zustand des Motors kann dieser möglicherweise auch in das Vorhersagemodell integriert werden.

Literaturverzeichnis

- [1] Zugriff am: 11.02.2024. Verfügbar unter: <https://mowotas.com/industrieprodukte/kompressortechnik/renner-kolbenkompressoren/riko/riko-700/500-ecn-industrie-kompressor-4-0-5-5-kw-500-l-ecn-10-bar>.
- [2] R. Obermaier, Handbuch Industrie 4.0 und Digitale Transformation. Wiesbaden: Springer Fachmedien Wiesbaden (2019).
- [3] T. Haase, "Industrie 4.0: Technologiebasierte Lern- und Assistenzsysteme für die Instandhaltung," Dissertation, Fakultät für Maschinenbau, Otto-von-Guericke-Universität Magdeburg und W. Bertelsmann Verlag, Otto-von-Guericke-Universität Magdeburg (2017).
- [4] C. Ferreira und G. Gonçalves, Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods, Journal of Manufacturing Systems, Jg. 63: S. 550–562 (2022), Verfügbar unter: doi: 10.1016/j.jmsy.2022.05.010.
- [5] W. Kersten, H. Koller und H. Lödding, Hg. Industrie 4.0: Wie intelligente Vernetzung und kognitive Systeme unsere Arbeit verändern (Schriftenreihe der Hochschulgruppe für Arbeits- und Betriebsorganisation e.V. (HAB)). Berlin: Gito mbH Verlag (2014). Verfügbar unter: http://www.industrie-management.de/sites/industrie-management.de/files/sonderdrucke/loedding_Industrie-4-0.pdf.
- [6] W. Babel, Internet of Things und Industrie 4.0. Wiesbaden: Springer Fachmedien Wiesbaden (2023).
- [7] A. Roth, Einführung und Umsetzung von Industrie 4.0: Grundlagen, Vorgehensmodell und Use Cases aus der Praxis. Heidelberg: Springer-Verlag Berlin, S.5 (2016).
- [8] K. Zhou, T. Liu & L. Zhou, Industry 4.0: Towards future industrial opportunities and challenges. 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, S. 2147–2152 (2015).
- [9] G. Clark, The conditions for industrial revolution in Britain. The Journal of Economic History 75, no. 3: S. 901-940 (2015).
- [10] J. U. Nef, The rise of the British coal industry. Economic History Review 3, no. 2: S. 241-267 (1950).
- [11] J. Mokyr, The Enlightened Economy: An Economic History of Britain 1700-1850, Yale University Press, USA, S. 112-115 (2015).
- [12] M. Castells, The Rise of the Network Society. John Wiley & Sons., New Jersey, S. XX (2000).
- [13] H. Kagermann, W. Wahlster & J. Helbig, Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Abschlussbericht des Arbeitskreises Industrie 4.0., acatech – Deutsche Akademie der Technikwissenschaften e.V., Frankfurt/Main (2013).

-
- [14] Acatech - Deutsche Akademie der Technikwissenschaften. Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution. München: Herbert Utz Verlag (2013).
- [15] R. Obermaier, Handbuch Industrie 4.0 und Digitale Transformation: Industrie 4.0 und Digitale Transformation als unternehmerische Gestaltungsaufgabe, Wiesbaden: Springer Gabler, S. 3–46 (2019).
- [16] E. Oztemel, S. Gursev, Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing* 31, S. 127–182 (2020). Verfügbar unter: <https://doi.org/10.1007/s10845-018-1433-8>.
- [17] D. Zühlke, & M. Schmitt. Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien, Migration. Wiesbaden, Springer Vieweg (2014). Verfügbar unter: <https://doi.org/10.1007/978-3-658-04682-8>.
- [18] Springer Gabler Wirtschaftslexikon: Industrie 4.0. Zugriff am: 20.02.2024 verfügbar unter: <https://wirtschaftslexikon.gabler.de/definition/industrie-40-54032>.
- [19] B. Vogel-Heuser, C. Diedrich, A. Hein, S. Kowalewski, C. Legat, F. Meyer auf der Heide, ... & O. Schütze. Informatik für Ingenieure, München: Carl Hanser Verlag, S.17-22 (2012).
- [20] Y. Lu, J. You, Y. Lu, & B. P. Rimal. Virtualization and Simulation of Cyber-Physical Systems: A Survey. In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC): 268-275. IEEE (2017).
- [21] M. Hermann, T. Pentek, & B. Otto. Design principles for Industrie 4.0 scenarios: A literature review. In 2016 Hawaii International Conference on System Sciences (HICSS) 49 : 3928–3937. IEEE (2016).
- [22] M. Broy. Cyber-Physical Systems — Wissenschaftliche Herausforderungen Bei Der Entwicklung. *Cyber-Physical Systems. acatech DISKUTIERT*, vol 0. Berlin: Springer Heidelberg, S.17-31 (2010): Verfügbar unter: https://doi.org/10.1007/978-3-642-14901-6_2.
- [23] S. S. Ansari, M. S. Farooq, M. A. Khan, & M. T. Khadem. Cyber-physical systems: Architecture, applications, and future directions. *Computing and Communication Workshop and Conference (CCWC)* 8:21–26. IEEE (2018).
- [24] D. Siepman, N. Graef. Industrie 4.0 – Grundlagen und Gesamtzusammenhang. In: Roth, A. (eds) Einführung und Umsetzung von Industrie 4.0. Berlin: Springer Gabler (2016). Verfügbar unter: https://doi.org/10.1007/978-3-662-48505-7_2.
- [25] Broy, M. (Hrsg.). *Cyber-Physical Systems: Innovation durch softwareintensive eingebettete Systeme*. Berlin: Springer Verlag. S. 17-31 (2010).
- [26] Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. (BITKOM). Big Data im Praxiseinsatz – Szenarien, Beispiele, Effekte. Berlin, BITKOM S.7-18 (2012): Zugriff am: 15.02.2024. Verfügbar: <https://www.bitkom.org/sites/default/files/file/import/BITKOM-LF-big-data-2012-online1.pdf>.
-

-
- [27] M. et al. Hartmann. Defining Smart Service Systems: A Viable Systems Approach. 47th Hawaii International Conference on System Sciences (HICSS). Hawaii, USA (2014).
- [28] McKinsey Digital. The age of analytics: Competing in a data-driven world. McKinsey Global Institute, S.15 (2016): Zugriff am: 28.01.2024. Verfügbar unter:
<https://www.mckinsey.com/~media/mckinsey/industries/public%20and%20social%20sector/our%20insights/the%20age%20of%20analytics%20competing%20in%20a%20data%20driven%20world/mgi-the-age-of-analytics-full-report.pdf>
- [29] Statista - Das Statistik-Portal, (2024): Zugriff am: 11.02.2024. Verfügbar unter: <https://de.statista.com/statistik/daten/studie/3564/umfrage/durchschnittliches-datenvolumen-pro-anschluss-seit-2001/>.
- [30] P. Gölzer, Big data in industrie 4.0: Eine strukturierte Aufarbeitung von Anforderungen, Nürnberg: Friedrich-Alexander-Universität Erlangen-Nürnberg. (2016). Zugriff am: 06.02.2024. Verfügbar unter: <https://d-nb.info/1123806233/34>.
- [31] Acatech - Deutsche Akademie der Technikwissenschaften, Kompetenzen für die Industrie 4.0: Qualifizierungsbedarfe und Lösungsansätze, Frankfurt/Main. S.16 (2016).
- [32] J. Gutsche, & K.-I. Voigt. Industrie 4.0: Potenziale erkennen und umsetzen. Wiesbaden: Springer Gabler. S.163 (2018).
- [33] T. Lager. Digitalisierung in der Instandhaltung: Strategien und Anwendungsbeispiele für die Praxis. Wiesbaden: Springer Vieweg. (2020).
- [34] K. Biedermann. Arbeit 4.0 - Herausforderungen und Chancen für Wirtschaft und Gesellschaft. Frankfurt am Main: Frankfurt University Press. S.26 (2016).
- [35] J. Fleischer, B. Klee, A. Spohrer, S. Merz. Leitfaden Sensorik für Industrie 4.0 Wege zu kostengünstigen Sensorsystemen, VDMA Forum Industrie: Frankfurt am Main. S.2 (2018). Verfügbar unter: https://www.vdma.org/documents/34570/0/Leitfaden_Sensorik_1529498469726.pdf/7e5717f1-9ae2-fd31-5d00-ea72ea7c7e96.
- [36] A. Schütze, N. Helwig, & T. Schneider. Sensors 4.0 – smart sensors and measurement technology enable Industry 4.0, J. Sens. Syst., 7, 359–371 (2018). Verfügbar unter: <https://doi.org/10.5194/jsss-7-359-2018>.
- [37] J. Cachay, & E. Abele. Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution. VDI nachrichten, 66:16-23 (2012).
- [38] D. Battini, M. Calzavara, A. Persona, & F. Sgarbossa. Sustainable Packaging Development for Fresh Food Supply Chains. Package. Technol. Sci., 29: 25–43. (2015) Verfügbar unter: doi: 10.1002/pts.2185.
- [39] M. Lebold, M. Thurston. Open standards for condition-based maintenance and Prognostic systems. In: Proceedings of MARCON 2001—fifth annual maintenance and reliability conference, Gatlinburg, USA (2001). Verfügbar unter: https://www.researchgate.net/publication/238290356_Open_Standards_for_Condition-Based_Maintenance_and_Prognostic_Systems.
-

-
- [40] A. Ethem, Introduction to Machine Learning, 3rd ed. Cambridge Massachusetts London (2014).
- [41] M. Ahmed, M. Baqqar, f. Gu, A. D. Ball. Fault detection and diagnosis using principal component analysis of vibration data from a reciprocating compressor, in: Proceedings of the UKACC International Conference on Control, 3-5 (2012), IEEE Press.
- [42] D. Djurdjanovic, J. Lee. J. Ni. Watchdog agent, an infotronics-based prognostics approach for product performance degradation assessment and prediction. Advance Engineering Informatics. 109–25 (2003).
- [43] H. Hotelling. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24: 417–441 (1933).
- [44] M. Nagdev & G. Tarun. A Research Study on Unsupervised Machine Learning Algorithms for Fault Detection in Predictive Maintenance. 5th International Conference on Industrial Engineering and Applications, Michigan, USA (2018). Verfügbar unter: doi: 10.13140/RG.2.2.28822.24648.
- [45] H. Trevor, R. Tibshirani, & J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer New York, NY (2009).
- [46] S. BRACKE. Technische Zuverlässigkeit: Datenanalytik, Modellierung, Risikoprognose. Berlin: Springer Vieweg. (2022).
- [47] ISO 9001:2015 Quality management systems – Requirements. International Organization for Standardization. (2015).
- [48] M. Modarres et al.. Reliability Engineering and Risk Analysis: A Practical Guide, CRC Press, Rochester Institute of Technology Rochester, New York (2016).
- [49] J. M. Juran, & J. A. De Feo. Juran's Quality Handbook: The Complete Guide to Performance Excellence, McGraw Hill Professional. S.920 (2010).
- [50] M. Catelani, L. Ciani, M.J. Venzi. RBD Model-Based Approach for Reliability Assessment in Complex Systems, IEEE Systems Journal. S. 1-9 (2018).
- [51] D.P. Siewiorek, & R.S Swarz. Reliable computer systems: design and evaluation. CRC Press. Pittsburgh, Pennsylvania (1992).
- [52] S. Lazarova-Molnar, N. Mohamed, International Conference on Emerging Data and Industry 4.0 (EDI40), Reliability Assessment in the Context of Industry 4.0: Data as a Game Changer, 2: (2019).
- [53] C. Jones, Software reliability assessment with ORACLE. McGraw-Hill, Inc, Ort (1996).
- [54] P. Hoang, Handbook of Engineering Statistics, London: Springer-Verlag London, Zweite Edition, London (2023).
- [55] N. Leveson. Safeware: System safety and computers. Addison-Wesley, (1995).
- [56] Bundesministerium für Arbeit und Soziales. Betriebssicherheitsverordnung (BetrSichV), Bundesgesetzblatt Teil I, Nr. 34, (2002).
- [57] International Organization for Standardization (ISO). ISO 26262: Road vehicles - Functional safety. Genf, Schweiz (2018).

-
- [58] J. Börcsök. Funktionale Sicherheit, VDE Verlag, Berlin. S.26 (2021). Zugriff am: 15.01.2024 Verfügbar unter: <https://content-select.com/goto/9783800753581/27> und <https://www.tuvsud.com/en-us/services/functional-safety/iec-61508>.
- [59] International Electrotechnical Commission (IEC). IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems. Genf, Schweiz: IEC, (2010).
- [60] Zugriff am: 04.02.2024. Verfügbar unter: <https://www.tuvsud.com/en-us/services/functional-safety/iec-61508>.
- [61] Josef Börcsök, Funktionale Sicherheit (2021), Berlin: VDE Verlag, 5. überarbeitete Auflage, S. 408 (2021).
- [62] DIN EN 13306:2018-02 Instandhaltung - Begriffe der Instandhaltung Verfügbar unter doi: <https://dx.doi.org/10.31030/2641990> & DIN 31051:2019-06 Grundlagen der Instandhaltung. Verfügbar unter: doi: <https://dx.doi.org/10.31030/3048531>.
- [63] H. Apel. Instandhaltungs- und Servicemanagement Intelligente Systeme München: Carl Hanser Verlag, 2. aktualisierte und erweiterte Auflage, (2023).
- [64] K. Matyas. Ganzheitliche Optimierung durch individuelle Instandhaltungsstrategien. Industrie Management, Verlag Nr. 2, S. 13–16 (2002).
- [65] W. Schröder. Ganzheitliches Instandhaltungsmanagement: Aufbau, Ausgestaltung und Bewertung, Wiesbaden: Gabler Verlag, (2010).
- [66] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael & B. Safaei, Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0, Sustainability, Jg. 12, Nr. 19, S. 8211: 2020, doi: 10.3390/su12198211.
- [67] Zugriff am: 29.01.2024. Verfügbar unter: <https://www.manomano.de/beratung/ihr-einkaufsguide-fuer-einen-kompressor-3385>.
- [68] Zugriff am: 15.12.2023. Verfügbar unter: <https://www.kaggle.com/datasets/afumetto/predictive-maintenance-dataset-air-compressor?select=6.png>.
- [69] S. Bracke, Technische Zuverlässigkeit. Berlin: Springer Berlin Heidelberg, (2022).
- [70] E. Cramer und U. Kamps, Grundlagen der Wahrscheinlichkeitsrechnung und Statistik. Berlin: Springer Berlin Heidelberg, (2020).
- [71] Y. Lei, Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery. Oxford, United Kingdom: Elsevier, (2017). Verfügbar unter: <http://www.sciencedirect.com/science/book/9780128115343>.
- [72] L. Fahrmeir, C. Heumann, R. Künstler, I. Pigeot und G. Tutz, Statistik. Berlin: Springer Berlin Heidelberg, (2016).
- [73] R. Kosfeld, H. F. Eckey und M. Türck, Deskriptive Statistik. Wiesbaden: Springer Fachmedien Wiesbaden, (2016).
-

-
- [74] A. Handl und T. Kuhlenkasper, *Multivariate Analysemethoden*. Berlin: Springer Berlin Heidelberg, (2017).
- [75] I. T. Jolliffe, Discarding Variables in a Principal Component Analysis. I: Artificial Data, *Applied Statistics*, Jg. 21, Nr. 2, S. 160 (1972), doi: 10.2307/2346488.
- [76] K. Siebertz, D. van Bebber und T. Hochkirchen, *Statistische Versuchsplanung: Design of Experiments (DoE)*, 2. Aufl. (VDI-Buch). Berlin: Springer Vieweg, (2017).
- [77] D. C. Montgomery, *Design and Analysis of Experiments*. Hoboken, NJ: Wiley, (2020).
- [78] S. Namuduri, B. N. Narayanan, V. S. P. Davuluru, L. Burton und S. Bhansali, Review—Deep Learning Methods for Sensor Based Predictive Maintenance and Future Perspectives for Electrochemical Sensors, *J. Electrochem. Soc.*, Jg. 167, Nr. 3, S. 37552 (2020), doi: 10.1149/1945-7111/ab67a8.
- [79] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (Covid-19 collection)*. Beijing, Boston, Farnham, Sebastopol, Tokyo: O'Reilly, (2019).
- [80] N. Panten, N. Strobel, J. Sossenheimer und E. Abele, Framework for an Energy Efficient and Flexible Automation Strategy and Control Optimization Approach of Supply Systems within a Thermally-Linked Factory, *Procedia CIRP*, Jg. 72, S. 526–532 (2018), doi: 10.1016/j.procir.2018.03.238
- [81] G. James, D. Witten, T. Hastie und R. Tibshirani, *An introduction to statistical learning: With applications in R*, 8. Aufl. (Springer texts in statistics). New York, Heidelberg, Dordrecht, London: Springer, (2017).
- [82] J. J. Montero 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), Verfügbar unter: doi: 10.1016/j.jmsy.2020.07.008.
- [83] S. L. Brunton, B. R. Noack und P. Koumoutsakos, "Machine Learning for Fluid Mechanics," *Annu. Rev. Fluid Mech.*, Jg. 52, Nr. 1, S. 477–508 (2020). Verfügbar unter: doi: 10.1146/annurev-fluid-010719-060214.
- [84] K. Gurney, *An Introduction to Neural Networks*. Hoboken: Taylor and Francis, (2003). Verfügbar unter: <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=182103>
- [85] Zugriff am: 28.01.2024. Verfügbar unter: <https://novustat.com/statistik-blog/kuenstliches-neuronales-netz-einfach-erklaert.html>
- [86] J. R. Quinlan, decision trees as probabilistic classifiers, in *Proceedings of the Fourth International Workshop on MACHINE LEARNING*, Elsevier, (1987), S. 31–37.
- [87] S. Matzka, *Künstliche Intelligenz in den Ingenieurwissenschaften*. Wiesbaden: Springer Fachmedien Wiesbaden, (2021).

Anhang

Anhang A: Physikalische Analyse des Datensatzes

Anhang B: Statistische Bewertung des Datensatzes

Anhang C: Datenanalyse

Anhang A: Physikalische Analyse des Datensatzes

Abbildung 4.1 bis Abbildung 4.8

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Input data files are available in the read-only "../input/"
# directory
# For example, running this (by clicking run or pressing
# Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('data'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
data = pd.read_csv('data.csv')
```

```
# Visualization

plt.figure(figsize=(20, 40))

# Plotting line plots for each numerical column
for i, col in enumerate(data.columns, 1):
    plt.subplot(len(data.columns), 1, i)
    sns.lineplot(data=data, x=data.index, y=col,
color='dodgerblue')
    plt.title(f'Line Plot of {col}')
    plt.xlabel('Index')
    plt.ylabel(col)

plt.tight_layout()
plt.show()
```

Anhang B: Statistische Bewertung des Datensatzes

Abbildung 4.9: Histogramm der Zustände der Kompressorkomponenten

```
import pandas as pd
import matplotlib.pyplot as plt
# Laden der Daten
file_path = 'data.csv'
data = pd.read_csv(file_path)

# Erstellen der Histogramme für alle Zustandslabels in einem
Diagramm
fig, axes = plt.subplots(2, 3, figsize=(15, 10))

# Lagerzustand (bearings)
data['bearings'].value_counts().plot(kind='bar', ax=axes[0, 0],
color='skyblue')
axes[0, 0].set_title('Lagerzustand (Bearings)')
axes[0, 0].set_xlabel('Zustand')
axes[0, 0].set_ylabel('Anzahl')

# Pumpenzustand (wpump)
data['wpump'].value_counts().plot(kind='bar', ax=axes[0, 1],
color='lightgreen')
axes[0, 1].set_title('Pumpenzustand (WPump)')
axes[0, 1].set_xlabel('Zustand')
axes[0, 1].set_ylabel('Anzahl')

# Zustand der Kühlung (radiator)
data['radiator'].value_counts().plot(kind='bar', ax=axes[0, 2],
color='lightcoral')
axes[0, 2].set_title('Zustand der Kühlung (Radiator)')
axes[0, 2].set_xlabel('Zustand')
axes[0, 2].set_ylabel('Anzahl')

# Zustand des Ventils (exvalve)
data['exvalve'].value_counts().plot(kind='bar', ax=axes[1, 0],
color='gold')
axes[1, 0].set_title('Zustand des Ventils (Exvalve)')
axes[1, 0].set_xlabel('Zustand')
axes[1, 0].set_ylabel('Anzahl')
```

```

# Zustand des Motors (acmotor)
data['acmotor'].value_counts().plot(kind='bar', ax=axes[1, 1],
color='teal')
axes[1, 1].set_title('Zustand des Motors (ACMotor)')
axes[1, 1].set_xlabel('Zustand')
axes[1, 1].set_ylabel('Anzahl')

# Entfernen des leeren Subplots
fig.delaxes(axes[1][2])

plt.tight_layout()
plt.show()

```

Abbildung 4.10: Korrelationsanalyse von allen Features und Zuständen

```

import seaborn as sns
# Umwandlung der Zustandslabels in numerische Werte
label_mapping = {
    'Ok': 1, 'Noisy': 0, # Für bearings und wpump
    'Clean': 1, 'Dirty': 0, # Für radiator und exvalve
    'Stable': 1 # Für acmotor (nur guter Zustand vorhanden)
}

# Anwenden der Umwandlung
for column in ['bearings', 'wpump', 'radiator', 'exvalve',
'acmotor']:
    data[column] = data[column].map(label_mapping)

# Berechnung der neuen Korrelationsmatrix
correlation_matrix_with_labels = data.corr()

# Visualisierung der neuen Korrelationsmatrix
plt.figure(figsize=(15, 12))
sns.heatmap(correlation_matrix_with_labels, annot=True, fmt=".2f",
cmap='coolwarm')
plt.title('Korrelationsmatrix mit Zustandslabels')
plt.show()

```

Abbildung 4.19: Hauptkomponentenanalyse des Datensatzes

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
# Auswahl der Variablen für PCA
label_columns = ['bearings', 'wpump', 'radiator', 'exvalve',
                 'acmotor'] # Liste der Labelspalten
pca_data = data.drop(label_columns, axis=1)

# Standardisierung der Daten
scaler = StandardScaler()
pca_data_scaled = scaler.fit_transform(pca_data)

# Anwendung der PCA
pca = PCA()
pca.fit(pca_data_scaled)

# Anteil der erklärten Varianz durch jede Komponente
explained_variance_ratio = pca.explained_variance_ratio_

# Kumulative erklärte Varianz
cumulative_explained_variance = np.cumsum(explained_variance_ratio)

# Anzahl der Komponenten für signifikante Varianzerklärung
significant_components = np.argmax(cumulative_explained_variance >=
0.95) + 1

# Visualisierung der erklärten Varianz mit hervorgehobener
signifikanter Komponente
plt.figure(figsize=(8, 6))
plt.bar(range(1, len(explained_variance_ratio) + 1),
        explained_variance_ratio, alpha=0.5, align='center',
        label='Individuelle erklärte Varianz')
plt.step(range(1, len(cumulative_explained_variance) + 1),
        cumulative_explained_variance, where='mid', label='Kumulative
        erklärte Varianz')
```

```

plt.axhline(y=0.95, color='r', linestyle='-', label='95% erklärte
Varianz')
plt.axvline(x=significant_components, color='g', linestyle='--',
label=f'{significant_components} Hauptkomponenten')
plt.ylabel('Anteil der erklärten Varianz')
plt.xlabel('Hauptkomponente')
plt.legend(loc='best')
plt.title('Erklärte Varianz durch PCA')
plt.show()

significant_components

```

Abbildung 4 11 : Effekte der Messwerte auf den Zustand der Lager

```

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# Angenommen, `data` ist Ihr DataFrame
# data = pd.read_csv('your_data.csv') # Beispiel, wie Sie Ihren
Datensatz laden könnten

# Auswahl der Variablen für PCA
label_columns = ['bearings', 'wpump', 'radiator', 'exvalve',
'acmotor']
pca_data = data.drop(label_columns, axis=1)

# Standardisierung der Daten
scaler = StandardScaler()
pca_data_scaled = scaler.fit_transform(pca_data)

# Anwendung der PCA mit 8 Komponenten
pca = PCA(n_components=8)
pca.fit(pca_data_scaled)

# Transformation der Daten in die ersten 8 Hauptkomponenten
pca_data_transformed = pca.transform(pca_data_scaled)

# Erstellung eines DataFrame mit den ersten 8 Hauptkomponenten
df_data = pd.DataFrame(data=pca_data_transformed,
columns=[f'PC{i+1}' for i in range(8)])

df_data
df_data.to_csv('data_H.csv', index=False)
# Standardisierung der Daten
scaler = StandardScaler()

```

```

pca_data_scaled = scaler.fit_transform(pca_data)

# Anwendung der PCA
pca = PCA()
pca.fit(pca_data_scaled)

# Ausgabe der Hauptkomponenten (die ersten paar)
pca_components = pca.components_
pca_components_df = pd.DataFrame(pca_components,
                                columns=pca_data.columns)

pca_components_df.head() # Anzeige der ersten paar
Hauptkomponenten

# Anzeige der Gewichte (Ladungen) für jede der ersten paar
Hauptkomponenten
pca_components_df_transposed = pca_components_df.transpose()
pca_components_df_transposed.columns = [f'PC{i+1}' for i in
                                        range(pca_components_df_transposed.shape[1])]

pca_components_df_transposed.head() # Anzeigen der Gewichte für
die ersten paar Hauptkomponenten

```

```

import matplotlib.pyplot as plt
import seaborn as sns

# Auswahl der relevanten Spalten für die Korrelationsanalyse
features = data.columns.drop(['id', 'acmotor', 'bearings', 'wpump',
                              'radiator', 'exvalve']) # Ignoriere ID und Labels
label = 'bearings'
# Berechnung der Korrelationen
correlations = data[features].corrwith(data[label]).sort_values(ascending=False)
# Erstellen eines Balkendiagramms zur Darstellung der Korrelationen
plt.figure(figsize=(10, 6))
sns.barplot(x=correlations.values, y=correlations.index,
            palette="viridis")
plt.title(f'Korrelationen der Features mit dem Label "{label}"')
plt.xlabel('Korrelationskoeffizient')
plt.ylabel('Features')
plt.show()

```

Abbildung 4.12: Analyse des F-Werts für den Lagerzustand

```
import numpy as np
from sklearn.feature_selection import f_classif
import pandas as pd

def calculate_anova_f_values(data, features, label):
    # ANOVA F-Wert für die gegebenen Features und Label berechnen
    F_values, p_values = f_classif(data[features], data[label])

    # F-Werte in einem Dictionary speichern, wobei der Schlüssel
    # der Feature-Name ist
    f_values_dict = dict(zip(features, F_values))
    return f_values_dict

# Funktion zur Erstellung des Diagramms mit F-Werten und kumulierten
# F-Werten
def plot_f_values_and_cumulative(data, label):
    # Berechnung der F-Werte
    f_values = calculate_anova_f_values(data, features, label)

    # Sortieren der F-Werte in absteigender Reihenfolge
    sorted_f_values = dict(sorted(f_values.items(), key=lambda
item: item[1], reverse=True))

    # Berechnung der kumulierten F-Werte
    total_f = sum(sorted_f_values.values())
    cumulative_f = np.cumsum(list(sorted_f_values.values())) /
total_f

    # Erstellen des Diagramms
    fig, ax1 = plt.subplots(figsize=(10, 6))

    color = 'tab:blue'
    ax1.set_xlabel('Features')
    ax1.set_ylabel('F-Wert', color=color)
    ax1.bar(range(len(sorted_f_values)),
list(sorted_f_values.values()), color=color)
    ax1.tick_params(axis='y', labelcolor=color)
```

```

ax2 = ax1.twinx() # Zweite Achse für die kumulierten Werte
color = 'tab:red'
ax2.set_ylabel('Kumulierter Anteil', color=color)
ax2.plot(range(len(cumulative_f)), cumulative_f, color=color,
marker='o')
ax2.tick_params(axis='y', labelcolor=color)

plt.title(f'F-Werte und kumulierte F-Werte für das Label
"{label}"')
ax1.set_xticks(range(len(sorted_f_values)))
ax1.set_xticklabels(list(sorted_f_values.keys()), rotation=90)
plt.show()

# Diagramm für das Label 'bearings'
plot_f_values_and_cumulative(data, 'bearings')

```

Abbildung 4.13: Effekte der Messwerte auf den Zustand der Wasserpumpe

```

# Analyse für das Label 'wpump'
label = 'wpump'

# Berechnung der Korrelationen
correlations_wpump =
data[features].corrwith(data[label]).sort_values(ascending=False)
# Erstellen eines Balkendiagramms zur Darstellung der Korrelationen
plt.figure(figsize=(10, 6))
sns.barplot(x=correlations_wpump.values,
y=correlations_wpump.index, palette="viridis")
plt.title(f'Korrelationen der Features mit dem Label "{label}"')
plt.xlabel('Korrelationskoeffizient')
plt.ylabel('Features')
plt.show()

```

Abbildung 4.14: Analyse des F-Werts für den Wasserpumpenzustand

```

# Diagramm für das Label 'wpump'
plot_f_values_and_cumulative(data, 'wpump')

```

Abbildung 4.15: Effekte der Messwerte auf den Zustand des Lüfters

```

# Analyse für das Label 'radiator'
label = 'radiator'

# Berechnung der Korrelationen
correlations_radiator =
data[features].corrwith(data[label]).sort_values(ascending=False)

# Erstellen eines Balkendiagramms zur Darstellung der Korrelationen
plt.figure(figsize=(10, 6))
sns.barplot(x=correlations_radiator.values,
y=correlations_radiator.index, palette="viridis")
plt.title(f'Korrelationen der Features mit dem Label "{label}"')
plt.xlabel('Korrelationskoeffizient')
plt.ylabel('Features')
plt.show()

```

Abbildung 4.16: Analyse des F-Werts für des Lüfters

```

# Diagramm für das Label 'radiator'
plot_f_values_and_cumulative(data, 'radiator')

```

Abbildung 4.17 Effekte der Messwerte auf den Zustand des Ventils

```

# Analyse für das Label 'exvalve'
label = 'exvalve'

# Berechnung der Korrelationen
correlations_exvalve =
data[features].corrwith(data[label]).sort_values(ascending=False)

# Erstellen eines Balkendiagramms zur Darstellung der Korrelationen
plt.figure(figsize=(10, 6))
sns.barplot(x=correlations_exvalve.values,
y=correlations_exvalve.index, palette="viridis")
plt.title(f'Korrelationen der Features mit dem Label "{label}"')
plt.xlabel('Korrelationskoeffizient')
plt.ylabel('Features')
plt.show()

```

Abbildung 4.18: Analyse des F-Werts für des Ventils

```

# Diagramm für das Label 'exvalve'
plot_f_values_and_cumulative(data, 'exvalve')

```

Anhang C: Datenanalyse

```
import pandas as pd

# Laden der Daten
data_path = 'data_H.csv'
label_path = 'Mappe3.xlsx'

data = pd.read_csv(data_path)
labels = pd.read_excel(label_path)

# Anzeigen der ersten paar Zeilen der Daten und Labels
data_head = data.head()
labels_head = labels.head()

data_head, labels_head
(  PC1  PC2  PC3  PC4  PC5  PC6  PC7 \
0 -6.429320 -1.261981 0.174264 0.272840 0.275921 -1.456887 -1.138535
1 -6.686883 -1.126503 -0.025443 -2.182906 0.714585 -1.264271 0.838674
2 -6.827636 -1.067591 -0.234252 -1.068145 0.665516 -0.522560 1.192931
3 -6.697989 -0.696508 -0.068765 0.474633 0.897109 0.537122 -1.025476
4 -6.441886 -0.842070 -0.222573 0.230296 0.348656 -0.248000 -0.457604

      PC8
0 -0.346113
1 -0.126885
2 -0.208057
3 -0.182553
4 -0.317094 ,
  bearings wpump radiator exvalve
0   1   1   1   1
1   1   1   1   1
2   1   1   1   1
3   1   1   1   1
4   1   1   1   1)
```

```
from sklearn.model_selection import train_test_split

# Standardaufteilung in 70% Training und 30% Test
train_data, test_data, train_labels, test_labels =
train_test_split(data, labels, test_size=0.3, random_state=42)

# Größe der aufgeteilten Daten überprüfen
train_data_shape = train_data.shape
test_data_shape = test_data.shape
train_labels_shape = train_labels.shape
test_labels_shape = test_labels.shape
```

```
train_data_shape, test_data_shape, train_labels_shape,  
test_labels_shape
```

```
((700, 8), (300, 8), (700, 4), (300, 4))
```

Regression

```
from sklearn.preprocessing import PolynomialFeatures  
from sklearn.linear_model import LinearRegression  
from sklearn.metrics import mean_squared_error, r2_score,  
mean_absolute_error  
import matplotlib.pyplot as plt  
import numpy as np  
  
# Polynomgrad  
poly_degree = 2  
  
# Polynomiale Features vorbereiten  
poly = PolynomialFeatures(degree=poly_degree)  
  
# Ergebnisse speichern  
models = {}  
predictions = {}  
predictions_a = {}  
scores = {}  
scores_a = {}  
  
for column in train_labels.columns:  
    # Polynomiale Features transformieren  
    X_train_poly = poly.fit_transform(train_data)  
    X_test_poly = poly.transform(test_data)  
  
    # Modell für jedes Label trainieren  
    model = LinearRegression()  
    model.fit(X_train_poly, train_labels[column])  
    models[column] = model  
  
    # Vorhersagen und Bewertung  
    pred = model.predict(X_test_poly)  
    predictions[column] = pred  
    scores[column] = {  
        "MSE": mean_squared_error(test_labels[column], pred),  
        "R2": r2_score(test_labels[column], pred),  
        "MAE": mean_absolute_error(test_labels[column], pred)  
    }  
  
    # Anpassen der Vorhersagen  
    adjusted_pred = np.where(pred < 0.5, 0, 1)
```

```

predictions_a[column] = adjusted_pred

scores_a[column] = {
    "MSE": mean_squared_error(test_labels[column],
adjusted_pred),
    "R2": r2_score(test_labels[column], adjusted_pred),
    "MAE": mean_absolute_error(test_labels[column],
adjusted_pred)
}

# Ergebnisse anzeigen
scores_a

```

```

{'bearings': {'MSE': 0.07666666666666666,
'R2': 0.5147338068781209,
'MAE': 0.07666666666666666},
'wpump': {'MSE': 0.006666666666666667,
'R2': 0.9619675456389453,
'MAE': 0.006666666666666667},
'radiator': {'MSE': 0.0, 'R2': 1.0, 'MAE': 0.0},
'exvalve': {'MSE': 0.0, 'R2': 1.0, 'MAE': 0.0}}

```

```

scores

{'bearings': {'MSE': 0.07491540468223121,
'R2': 0.5258185230043739,
'MAE': 0.20812423443328198},
'wpump': {'MSE': 0.04199438420914126,
'R2': 0.7604275748717854,
'MAE': 0.1644099440779817},
'radiator': {'MSE': 0.017732251056912555,
'R2': 0.8891734308942966,
'MAE': 0.09118858312292794},
'exvalve': {'MSE': 0.006064845924656164,
'R2': 0.958304473820254,
'MAE': 0.06085396863472217}}

```

Abbildung 4. 21: Ergebnis der polynomialen Regression Vorhersagen im Vergleich zu realen Daten

```

# Korrektur der Indizes für die Visualisierung
num_samples = 50
indices = np.random.choice(test_data.shape[0], size=num_samples,
replace=False)

# Visualisierung der Vorhersagen gegenüber den tatsächlichen Werten
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Polynomiale Regressionsergebnisse (Grad 2)')

```

```

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.scatter(test_labels.iloc[indices][column],
predictions[column][indices], alpha=0.7, label='Vorhersagen')
    ax.plot(test_labels.iloc[indices][column],
test_labels.iloc[indices][column], color='red', label='Tatsächliche
Werte')
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Tatsächliche Werte')
    ax.set_ylabel('Vorhersagen')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

Abbildung 4. 22: Ergebnis der polynomialen Regression und tatsächliche Werte im zeitlichen Verlauf

```

# Visualisierung der tatsächlichen Labels und der Vorhersagen über
die Indizes
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Tatsächliche Labels vs. Vorhersagen über Indizes')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.plot(test_labels[column].reset_index(drop=True),
label='Tatsächliche Werte', alpha=0.7)
    ax.plot(predictions[column], label='Vorhersagen', alpha=0.7)
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Index')
    ax.set_ylabel('Werte')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

Abbildung 4. 23: Ergebnis der polynomialen Regression mit angepassten Vorhersagen

```

num_samples = 50
indices = np.random.choice(test_data.shape[0], size=num_samples,
replace=False)

# Visualisierung der Vorhersagen gegenüber den tatsächlichen Werten
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Polynomiale Regressionsergebnisse (Grad 2)')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):

```

```

    ax.scatter(test_labels.iloc[indices][column],
predictions_a[column][indices], alpha=0.7, label='Vorhersagen')
    ax.plot(test_labels.iloc[indices][column],
test_labels.iloc[indices][column], color='red', label='Tatsächliche
Werte')
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Tatsächliche Werte')
    ax.set_ylabel('Vorhersagen')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

Random Forest

```

from sklearn.ensemble import RandomForestRegressor

# Random Forest Modelle für jedes Label trainieren
rf_models = {}
rf_predictions = {}
rf_scores = {}
rf_predictions_a = {}
rf_scores_a = {}

for column in train_labels.columns:
    # Modell für jedes Label trainieren
    rf_model = RandomForestRegressor(random_state=42)
    rf_model.fit(train_data, train_labels[column])
    rf_models[column] = rf_model

    # Vorhersagen und Bewertung
    rf_pred = rf_model.predict(test_data)
    rf_predictions[column] = rf_pred
    rf_scores[column] = {
        "MSE": mean_squared_error(test_labels[column], rf_pred),
        "R2": r2_score(test_labels[column], rf_pred),
        "MAE": mean_absolute_error(test_labels[column], pred)
    }

    # Anpassen der Vorhersagen
    adjusted_pred = np.where(rf_pred < 0.5, 0, 1)
    rf_predictions_a[column] = adjusted_pred

    rf_scores_a[column] = {
        "MSE": mean_squared_error(test_labels[column],
adjusted_pred),
        "R2": r2_score(test_labels[column], adjusted_pred),

```

```

        "MAE": mean_absolute_error(test_labels[column],
adjusted_pred)
    }

# Ergebnisse anzeigen
rf_scores_a
{'bearings': {'MSE': 0.12666666666666668,
'R2': 0.19825585484211283,
'MAE': 0.12666666666666668},
'wpump': {'MSE': 0.03666666666666667,
'R2': 0.7908215010141988,
'MAE': 0.03666666666666667},
'radiator': {'MSE': 0.023333333333333334,
'R2': 0.8541666666666667,
'MAE': 0.023333333333333334},
'exvalve': {'MSE': 0.0033333333333333335,
'R2': 0.9770834924757467,
'MAE': 0.0033333333333333335}}

rf_scores
{'bearings': {'MSE': 0.08008966666666667,
'R2': 0.4930677262817358,
'MAE': 0.4173242182318769},
'wpump': {'MSE': 0.025624333333333336,
'R2': 0.8538165567951318,
'MAE': 0.43897756817034206},
'radiator': {'MSE': 0.020760333333333332,
'R2': 0.8702479166666667,
'MAE': 0.42012966561346005},
'exvalve': {'MSE': 0.0041813333333333333,
'R2': 0.9712535329615767,
'MAE': 0.06085396863472217}}

```

Abbildung 4. 24: Ergebnisse der Random Forest Regression Vorhersagen über tatsächlichen Werten

```

# Scatterplots der tatsächlichen Labels vs. Random Forest-
Vorhersagen
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Random Forest: Tatsächliche Werte vs. Vorhersagen')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.scatter(test_labels[column], rf_predictions[column],
alpha=0.7)
    ax.plot(test_labels[column], test_labels[column], color='red',
label='Ideallinie')
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Tatsächliche Werte')
    ax.set_ylabel('Vorhersagen')
    ax.legend()

```

```
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

```
# Visualisierung der tatsächlichen Labels und der Random Forest-
Vorhersagen über die Indizes
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Random Forest: Tatsächliche Labels vs. Vorhersagen
über Indizes')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.plot(test_labels[column].reset_index(drop=True),
label='Tatsächliche Werte', alpha=0.7)
    ax.plot(rf_predictions[column], label='Vorhersagen', alpha=0.7)
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Index')
    ax.set_ylabel('Werte')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

Abbildung 4. 25: Angepasste vorhergesagte und tatsächliche Werte der Random Forest Regression

```
# Scatterplots der tatsächlichen Labels vs. Random Forest-
Vorhersagen
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Random Forest: Tatsächliche Werte vs. Vorhersagen')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.scatter(test_labels[column], rf_predictions_a[column],
alpha=0.7)
    ax.plot(test_labels[column], test_labels[column], color='red',
label='Ideallinie')
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Tatsächliche Werte')
    ax.set_ylabel('Vorhersagen')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

Neuronales Netz

Abbildung: 4. 26: Ergebnisse des neuronalen Netzes Vorhersagen über tatsächlichen Werten

```
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPRegressor

# Daten normalisieren
scaler = StandardScaler()
train_data_scaled = scaler.fit_transform(train_data)
test_data_scaled = scaler.transform(test_data)

# Neuronales Netz konfigurieren und trainieren
nn_models = {}
nn_predictions = {}
nn_scores = {}
nn_predictions_a = {}
nn_scores_a = {}

for column in train_labels.columns:
    # Neuronales Netz konfigurieren
    nn_model = MLPRegressor(hidden_layer_sizes=(100,),
                            activation='relu',
                            solver='adam',
                            random_state=42,
                            max_iter=500)

    # Training
    nn_model.fit(train_data_scaled, train_labels[column])
    nn_models[column] = nn_model

    # Vorhersagen und Bewertung
    nn_pred = nn_model.predict(test_data_scaled)
    nn_predictions[column] = nn_pred
    nn_scores[column] = {
        "MSE": mean_squared_error(test_labels[column], nn_pred),
        "R2": r2_score(test_labels[column], nn_pred),
        "MAE": mean_absolute_error(test_labels[column], nn_pred)
    }

    # Anpassen der Vorhersagen
    adjusted_pred = np.where(nn_pred < 0.5, 0, 1)
    nn_predictions_a[column] = adjusted_pred

    nn_scores_a[column] = {
        "MSE": mean_squared_error(test_labels[column],
adjusted_pred),
        "R2": r2_score(test_labels[column], adjusted_pred),
```

```

        "MAE": mean_absolute_error(test_labels[column],
adjusted_pred)
    }

# Ergebnisse anzeigen
nn_scores_a

for column in train_labels.columns:
    # Neuronales Netz konfigurieren
    nn_model = MLPRegressor(hidden_layer_sizes=(100,),
        activation='relu',
        solver='adam',
        random_state=42,
        max_iter=500)

    # Training
    nn_model.fit(train_data_scaled, train_labels[column])

    # Vorhersagen
    nn_pred = nn_model.predict(test_data_scaled)
    nn_predictions[column] = nn_pred

# Scatterplots der tatsächlichen Labels vs. neuronalen Netz-
# Vorhersagen
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Neuronales Netz: Tatsächliche Werte vs. Vorhersagen')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.scatter(test_labels[column], nn_predictions[column],
alpha=0.7)
    ax.plot(test_labels[column], test_labels[column], color='red',
label='Ideallinie')
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Tatsächliche Werte')
    ax.set_ylabel('Vorhersagen')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

Abbildung: 4. 27: Angepasste vorhergesagte und tatsächliche Werte des neuronalen Netzes

```

fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Neuronales Netz: Tatsächliche Werte vs. Vorhersagen')
for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):

```

```

    ax.scatter(test_labels[column], nn_predictions_a[column],
alpha=0.7)
    ax.plot(test_labels[column], test_labels[column], color='red',
label='Ideallinie')
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Tatsächliche Werte')
    ax.set_ylabel('Vorhersagen')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

```

# Da der Code-Execution-State zurückgesetzt wurde, müssen wir die
Daten und die erforderlichen Schritte erneut laden und ausführen

```

```

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split

# Daten erneut laden
data_path = 'data_H.csv'
label_path = 'Mappe3.xlsx'
data = pd.read_csv(data_path)
labels = pd.read_excel(label_path)

# Aufteilen in Trainings- und Testdaten
train_data, test_data, train_labels, test_labels =
train_test_split(data, labels, test_size=0.3, random_state=42)

# Daten normalisieren
scaler = StandardScaler()
train_data_scaled = scaler.fit_transform(train_data)
test_data_scaled = scaler.transform(test_data)

# Neuronales Netz konfigurieren und trainieren
nn_predictions = {}

for column in train_labels.columns:
    # Neuronales Netz konfigurieren
    nn_model = MLPRegressor(hidden_layer_sizes=(100,),
activation='relu',
solver='adam',
random_state=42,

```

```

max_iter=500)

# Training
nn_model.fit(train_data_scaled, train_labels[column])

# Vorhersagen
nn_pred = nn_model.predict(test_data_scaled)
nn_predictions[column] = nn_pred

# Scatterplots der tatsächlichen Labels vs. neuronalen Netz-
Vorhersagen
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Neuronales Netz: Tatsächliche Werte vs. Vorhersagen')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.scatter(test_labels[column], nn_predictions[column],
alpha=0.7)
    ax.plot(test_labels[column], test_labels[column], color='red',
label='Ideallinie')
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Tatsächliche Werte')
    ax.set_ylabel('Vorhersagen')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

```

# Visualisierung der tatsächlichen Labels und der Random Forest-
Vorhersagen über die Indizes
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Neuronales Netz: Tatsächliche Labels vs. Vorhersagen
über Indizes')

for i, (column, ax) in enumerate(zip(train_labels.columns,
axes.flatten())):
    ax.plot(test_labels[column].reset_index(drop=True),
label='Tatsächliche Werte', alpha=0.7)
    ax.plot(predictions[column], label='Vorhersagen', alpha=0.7)
    ax.set_title(f'Label: {column}')
    ax.set_xlabel('Index')
    ax.set_ylabel('Werte')
    ax.legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

nn_scores

```
{'bearings': {'MSE': 0.06303836884035083,  
  'R2': 0.6009949225943052,  
  'MAE': 0.4173242182318769},  
'wpump': {'MSE': 0.0281808405560261,  
  'R2': 0.8392320201545164,  
  'MAE': 0.43897756817034206},  
'radiator': {'MSE': 0.01897669847793405,  
  'R2': 0.8813956345129123,  
  'MAE': 0.42012966561346005},  
'exvalve': {'MSE': 0.01604162415499123,  
  'R2': 0.88971459980527,  
  'MAE': 0.06085396863472217}}
```

nn_scores_a

```
{'bearings': {'MSE': 0.07, 'R2': 0.5569308671495887, 'MAE': 0.07},  
'wpump': {'MSE': 0.0, 'R2': 1.0, 'MAE': 0.0},  
'radiator': {'MSE': 0.0, 'R2': 1.0, 'MAE': 0.0},  
'exvalve': {'MSE': 0.0, 'R2': 1.0, 'MAE': 0.0}}
```