

Bachelorarbeit

KI-Integration in Cloud Native Platform Engineering: Eine systematische Analyse aktueller Lösungsansätze und deren praktische Anwendung

von

Nils Arnold

zur Erlangung des akademischen Grades

Bachelor of Science

im Studiengang Wirtschaftsinformatik

an der Hochschule Konstanz Technik, Wirtschaft und Gestaltung und der Robert Bosch GmbH

Matrikelnummer: 307179

Abgabedatum: 31.01.2026

Erstbetreuer: Prof. Dr. Johannes Schneider

Zweitbetreuer: Lukas Grodmeier.



Abstract

Abstract...

Ehrenwörtliche Erklärung

Hiermit erkläre ich, Nils Arnold, geboren am 18. Januar 2003 in Balingen,

- (1) dass ich meine Bachelorarbeit mit dem Titel:

„KI-Integration in Cloud Native Platform Engineering: Eine systematische Analyse aktueller Lösungsansätze und deren praktische Anwendung“

bei der Robert Bosch GmbH unter Anleitung von Prof. Dr. Johannes Schneider und Lukas Grodmeier selbstständig und ohne fremde Hilfe angefertigt habe und keine anderen als die angeführten Hilfen benutzt habe;

- (2) dass ich die Übernahme wörtlicher Zitate, von Tabellen, Zeichnungen, Bildern und Programmen aus der Literatur oder anderen Quellen (Internet) sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Abstract	i
Ehrenwörtliche Erklärung	ii
Inhaltsverzeichnis	iii
1 Einleitung	1
1.1 Problemstellung	1
1.2 Zielsetzung der Arbeit	2
1.3 Aufbau der Arbeit	2
2 Grundlagen und verwandte Arbeiten	3
2.1 Grundlagen	3
2.1.1 Cloud Native Technologien und Platform Engineering	3
2.1.2 Zentrale Anwendungsbereiche und Herausforderungen	3
2.1.3 Lernparadigmen des maschinellen Lernens	3
2.1.4 AIOps und verwandte Konzepte	4
2.2 Verwandte Arbeiten	4
3 Methodisches Vorgehen	5
3.1 Vorgehen der Mapping Study	5
3.2 Forschungsfragen	6
3.3 Literatur Analyse Prozess	7
3.3.1 Suchstrategie	7
3.3.2 Auswahlkriterien	7
3.3.3 Schneeballmethode	8
3.3.4 Datenerhebung aus den Studien	8
3.3.5 Datenanalyse und Kategorisierung	9
4 Literatur Ergebnisse	10
4.1 Quantitative Analyse	10
4.1.1 Anwendungsbereiche der KI im Platform Engineering	11
4.1.2 Herausforderungen der KI-Integration im Platform Engineering	12
4.1.3 Formen des maschinellen Lernens	13
4.1.4 Verwendete Algorithmen	14

4.2	Ergebnisse der Mapping Study	15
4.2.1	Zusammenspiel der Anwendungsfelder und Herausforderungen	15
4.2.2	Zusammenspiel der Lernparadigmen und Algorithmen	17
4.2.3	Mapping von Anwendungsfelder und Lernparadigmen	18
4.2.4	Mapping von Herausforderungen und Lernparadigmen	19
4.3	Matching/ Framework	19
4.3.1	Überblick und Kategorisierung der KI-Anwendungen	19
5	Anwendung der Literaturrecherche	20
5.1	Bewertungskonzept/ Framework	20
5.2	Analyse der Bosch Digital Manufacturing Plattform	20
6	Diskussion	21
6.1	Beantwortung der Forschungsfragen	21
6.2	Praxisrückschluss	21
6.3	Limitationen	21
6.4	Handlungsempfehlungen	22
7	Zusammenfassung und Ausblick	23
7.1	Zusammenfassung	23
7.2	Ausblick	23
	Tabellenverzeichnis	24
	Abbildungsverzeichnis	25
	Literaturverzeichnis	26

1

Einleitung

Mit der zunehmenden Verbreitung von Cloud-Native-Technologien und Plattform-Engineering Ansätzen, stehen Unternehmen vor der Herausforderung, ihre Software-Entwicklung und den Plattformbetrieb effizient, skalierbar und resilient zu gestalten. Laut der CNF-Studie geben 89 Prozent der befragten Organisationen an, Cloud Native-Technologien zu nutzen [1]. Die Komplexität von Infrastrukturen, Plattformen und Entwicklungsumgebung wächst in diesem Umfeld stark. Diese Entwicklung eröffnet zwar enorme Potentiale für Agilität und Innovation, stellt Plattform-Teams aber gleichermaßen vor neue Herausforderungen. Der Wunsch nach Automatisierung durch den Einsatz von Künstlicher Intelligenz wird immer größer.

1.1. Problemstellung

Trotz der flächendeckenden Verbreitung von Cloud-Native Architekturen und dem klaren Fokus vieler Unternehmen auf Automatisierung und Effizienzsteigerung, bleibt die Frage offen, wie Plattform-Teams konkret von fortgeschrittenen Automatisierungs- und KI-gestützten Ansätzen profitieren. Während zahlreiche Unternehmen bereits erste KI-gestützte Tools in ihren Cloud-Native Umgebungen einsetzen, existiert bislang keine systematische und evidenzbasierte Analyse, welche Lösungen tatsächlich Mehrwert für Plattform-Teams schaffen. Die Forschung zu AIOps und KI im Software-Engineering ist zwar umfangreich, doch deren Bezug zu spezifischen Domänen des Plattform Engineering wie CI/CD-Automatisierung, Infrastruktur-Management und Monitoring bleibt häufig unscharf. Zudem fehlen praxisnahe Untersuchungen, die auf realen Plattform-

Stacks aufbauen und konkrete Pain Points der beteiligten Teams berücksichtigen. Daraus ergibt sich die Notwendigkeit, den aktuellen Stand der Forschung systematisch zu erfassen, mit industriellen Anforderungen abzugleichen und daraus Handlungsempfehlungen für die Integration von KI in bestehenden Plattformlandschaften abzuleiten.

1.2. Zielsetzung der Arbeit

Ziel dieser Arbeit ist es, eine wissenschaftlich fundierte Grundlage für die Integration von Künstlicher Intelligenz in Cloud-Native-Plattform-Engineering-Umgebung zu schaffen. Hierzu wird zunächst im Rahmen einer systematischen Mapping Study untersucht, welche aktuellen KI-Ansätze im Plattform Engineering existieren und wo genau sie Anwendung finden. Darauf aufbauend erfolgt eine Analyse der Bosch Digital Manufacturing Plattform, um bestehende Herausforderungen zu identifizieren und potenzielle Einsatzfelder von KI-Technologien zu evaluieren. Das Ende der Arbeit setzt sich aus der Entwicklung eines praxistauglichen Frameworks zur Bewertung von KI-Potenzialen in Plattformumgebung sowie mit einer prototypischen Implementierung zusammen. Damit leistet die Arbeit sowohl einen wissenschaftlichen als auch einen praktischen Beitrag zur Weiterentwicklung moderner Plattform-Engineering Praktiken.

1.3. Aufbau der Arbeit

Dieser Abschnitt gibt einen Überblick über den strukturellen Aufbau der Arbeit.

2

Grundlagen und verwandte Arbeiten

In diesem Kapitel werden die theoretischen Grundlagen dargestellt, die für das Verständnis der Arbeit notwendig sind, sowie verwandte Arbeiten eingeordnet.

2.1. Grundlagen

2.1.1. Cloud Native Technologien und Platform Engineering

Ein Überblick über Cloud-Native-Prinzipien (Containment, Orchestrierung, Microservices) und die Rolle von Platform Engineering.

2.1.2. Zentrale Anwendungsbereiche und Herausforderungen

DevOps-Praktiken, CI/CD-Pipelines und Betriebsaspekte für zuverlässige Softwarebereitstellung.

2.1.3. Lernparadigmen des maschinellen Lernens

Die drei Formen des maschinellen Lernens setzen sich aus Supervised Learning, Unsupervised Learning und Reinforcement Learning zusammen. Diese drei Lernparadigmen werden nachfolgend genauer beschrieben auf Basis der Definitionen von Enemosah [2].

Supervised Learning beschreibt Verfahren, bei denen Modelle anhand markierter Daten (labeled data) trainiert werden. Ziel ist es, auf Basis bekannter Eingabe-Ausgabe-Paare präzise Vorhersagen zu treffen, etwa für Klassifikations- oder Priorisierungsaufgaben.

Unsupervised Learning arbeitet mit unmarkierten Daten und dient der Erkennung verborgener Muster oder Strukturen. Typische Verfahren wie Clustering gruppieren ähnliche Datenpunkte und unterstützen so beispielsweise die Analyse von Systemlogs oder die Identifikation von Anomalien.

Reinforcement Learning nutzt ein belohnungsbasiertes Lernprinzip, bei dem ein Agent durch Interaktion mit seiner Umgebung optimale Strategien erlernt. Im Kontext der betrachteten Literatur wird RL u. a. zur dynamischen Optimierung von Ressourcenallokationen eingesetzt, beispielsweise für adaptive Skalierungs- oder Rollback-Entscheidungen [2].

Tabelle 2.1: Beschreibung Algorithmen und Methoden

Algorithmen und Methoden	Beschreibung
Deep Learning / Neuronale Netze	Erfassen komplexe Muster in Daten und eignen sich besonders für unstrukturierte Eingaben wie Log- oder Monitoring-Daten.
Ensemble und Baum-basiert	Kombinieren mehrere Modelle zur Steigerung der Vorhersagegenaugigkeit und verarbeiten große sowie semi-strukturirte Datensätze effizient.
Klassische Klassifikation/ Regression	Traditionelle ML-Modelle zur Vorhersage von Mustern oder Ereignissen auf Basis strukturierter Daten.
Clustering	Gruppieren Datenpunkte ohne Labels in inhaltlich ähnliche Cluster und unterstützen dadurch Musterkennung und Anomaliedetektion.

2.1.4. AIOps und verwandte Konzepte

Definition und Abgrenzung von AIOps, Monitoring, Observability und autonomen Betriebsansätzen.

2.2. Verwandte Arbeiten

Relevante wissenschaftliche Arbeiten, Industriereports und State-of-the-Art-Übersichten werden beschrieben und kritisch eingeordnet.

3

Methodisches Vorgehen

Diese Kapitel beschreibt das methodische Vorgehen dieser Arbeit. Es erläutert die zu-grunde liegende Forschungslogik, die Auswahl des methodischen Ansatzes sowie die Verfahren zur Datenerhebung und -auswertung. Ziel ist es, ein wissenschaftlich fundiertes und zugleich praxisorientiertes Vorgehen aufzuzeigen, das eine systematische Untersuchung der Forschungsfragen ermöglicht. In Abschnitt 3.1 wird eine systematische Mapping Study (SMS) nach Petersen et al [3] angewendet. Abschnitt 3.2 definiert die spezifischen Ziele und konkreten Forschungsfragen dieser Arbeit, welche als Basis für die anschließende Analyse dienen. Darauf aufbauend beschreibt Abschnitt 3.3 den Prozess der Literaturanalyse. Dieser umfasst die Entwicklung einer Suchstrategie, die Definition von Ein- und Ausschlusskriterien, die Schneeballmethode sowie die Schritte der Datenextraktion und -synthese.

3.1. Vorgehen der Mapping Study

Für diese Arbeit wird eine Systematic Mapping Study (SMS) nach den Richtlinien von Petersen, Vakkalanka und Kuzniarz [3] durchgeführt. Diese Methodik dient dazu, den aktuellen Forschungsstand zu einem Themengebiet systematisch zu erfassen, zu kategorisieren und bestehende Forschungslücken zu identifizieren.

Das Vorgehen umfasst die Phasen Planung, Durchführung und Auswertung. In der Planungsphase werden die Forschungsfragen definiert und die Suchstrategie entwickelt, einschließlich der Auswahl relevanter wissenschaftlicher Datenbanken. Dabei wird gezielt nach bestimmten Keywords gesucht, um Publikationen zu finden, die KI-Anwendungen

im Kontext von Platform Engineering adressieren.

Um eine fundierte Datenerhebung und Auswertung sicherzustellen, werden einzelne Prinzipien einer Systematic Literature Review (SLR) nach Kitchenham und Charters [4] berücksichtigt. In der Durchführungsphase werden identifizierte Studien anhand festgelegter Ein- und Ausschlusskriterien geprüft. Zusätzlich wird das Schneeballverfahren nach Wohlin [5] eingesetzt, um die Literatursammlung zu erweitern. Alle Schritte werden dokumentiert, um die Nachvollziehbarkeit und Reproduzierbarkeit sicherzustellen. Die Auswertung erfolgt durch eine systematische Kategorisierung der Studien entlang zentraler Themenfelder des Platform Engineerings. Darauf aufbauend werden Muster, Trends und Forschungslücken identifiziert.

Die Ergebnisse der Mapping Study bilden die Grundlage für die anschließende Analyse der Bosch Digital Manufacturing Platform sowie für die Entwicklung eines Frameworks zur Bewertung von KI-Potenzialen in Cloud-Native Plattformumgebungen.

3.2. Forschungsfragen

Die Forschungsfragen werden durch ein strukturiertes methodisches Vorgehen beantwortet. Jede Forschungsphase ist darauf ausgelegt, das Verständnis über den Einsatz von Künstlicher Intelligenz im Platform Engineering schrittweise zu vertiefen. Die Mapping Study dient der Beantwortung der ersten beiden Forschungsfragen, indem sie eine systematische Übersicht über bestehende KI-Ansätze und deren Anwendungsfelder liefert. Auf Grundlage der Ergebnisse der Mapping Study zielt die dritte Forschungsfrage darauf ab, ein praxisorientiertes Framework zur Beantwortung und Übertragbarkeit von KI-Lösungen zu entwickeln. Die Arbeit ist entlang der folgenden Forschungsfragen strukturiert:

- RQ1:** Welche typische Anwendungsfelder und Herausforderungen bestehen im Platform Engineering, in denen KI-Technologien potenziell Mehrwert bieten können?
Ziel: Systematische Erfassung und Kategorisierung relevanter Use Cases.
- RQ2:** Welche KI-Technologien (einschließlich Frameworks und Tools) finden derzeit im Platform Engineering Anwendung, und welche Formen des maschinellen Lernens und Algorithmen kommen dabei zum Einsatz?
Ziel: Erstellung einer Übersicht über vorhandene KI-Ansätze und deren typische Einsatzkontakte.
- RQ3:** Wie lassen sich die identifizierten KI-Lösungen auf typische Anwendungsfälle

in Cloud-Native-Platform-Umgebungen übertragen und hinsichtlich ihres Mehrwertes und ihrer Umsetzbarkeit bewerten?

Ziel: Entwicklung eines Bewertungsschemas, das die Passung zwischen KI-Lösungen und spezifischen Platform-Use-Cases beschreibt und die praktische Umsetzbarkeit aufzeigt.

3.3. Literatur Analyse Prozess

Der folgende Abschnitt beschreibt den Ablauf der Literaturanalyse im Rahmen der durchgeführten Mapping Study. Ziel ist es, den methodischen Prozess transparent darzustellen. Dazu werden zunächst die Suchstrategie und die Auswahlkriterien erläutert, gefolgt von der Anwendung der Schneeballmethode. Danach wird die Datenextraktion sowie die Datensynthese beschrieben.

3.3.1. Suchstrategie

Zur Durchführung der Mapping Study wurde eine systematische Suchstrategie angewendet, um relevante wissenschaftliche Publikationen zu identifizieren. Die Literaturrecherche erfolgte in den Datenbanken Google Scholar, SpringerLink, ScienceDirect und IEEE Xplorer, da diese eine breite Abdeckung im Bereich Software Engineering, Cloud Native Technologien und KI bieten. Ziel der Suche war, eine möglichst vollständige Übersicht aktueller Forschungsarbeiten zu KI-Anwendungen im Platform Engineering zu erhalten. Dafür wurden gezielt Suchbegriffe und Kombinationen von Suchstrings verwendet, die zentrale Themen der Arbeit abbilden. Die Suchbegriffe waren hierbei: „Platform Engineering“, „Cloud-Native“, AIOps“, „Artifical Intelligence“, „Machine Learning“, „MLOps“, „DevOps“ und „Kubernetes Cluster“.

Zur Transparenz und Vollständigkeit sind die exakten Suchstrings sowie ihre logischen Verknüpfungen im Anhang dokumentiert.

3.3.2. Auswahlkriterien

Um relevante Studien und Publikationen zu identifizieren, wurde ein systematischer Auswahlprozess durchgeführt, der auf klar definierten Ein- und Ausschlusskriterien basiert. Eine Studie wurde in die Analyse aufgenommen, wenn sie alle Einschlusskriterien erfüllt und zugleich keinem der Ausschlusskriterien unterlag. Die vollständige Übersicht

der Kriterien ist in Tabelle 3.1 dargestellt.

Tabelle 3.1: Auswahlkriterien

Kriterium	Beschreibung
EK1	Die Publikation befasst sich mit dem Einsatz von KI oder Machine Learning im Kontext von Platform Engineering, Cloud-Native-Technologien, DevOps oder AIOps.
EK2	Die Studie beschreibt konkrete KI-Methoden, Anwendungen, Architekturen oder Use Cases, die sich auf Plattformumgebungen beziehen.
EK3	Die Arbeit ist wissenschaftlich fundiert, z.B. als Konferenz-, Journal- oder White Paper, auch wenn kein Peer-Review-Verfahren vorliegt.
EK4	Veröffentlichungen sind in englischer oder deutscher Sprache verfasst und nach 2020 erschienen.
AK1	Arbeiten, die keinen direkten Bezug zu KI im Platform Engineering oder verwandten Domänen aufweisen.
AK2	Review-Paper oder systematische Übersichtsarbeiten, die keine eigenen empirischen oder technischen Beiträge enthalten.
AK3	Studien ohne nachvollziehbare methodische Grundlage oder ohne Beschreibung der verwendeten KI-Techniken.

3.3.3. Schneeballmethode

Zur Ergänzung der systematischen Suche wurde eine Schneeballmethode nach den Leitlinien von Wohlin [5] angewendet. Dabei erfolgte sowohl eine Rückwärtssuche als auch eine Vorwärtssuche. Als Ausgangspunkt dienten vier relevante Paper, auf deren Basis zwei Iterationen der Vorwärts- und Rückwärtssuche durchgeführt wurden. Die neu gefundenen Publikationen wurden nach denselben Ein- und Ausschlusskriterien geprüft. Der Prozess wurde beendet, sobald keine weiteren relevanten Studien identifiziert werden konnten.

3.3.4. Datenerhebung aus den Studien

Zur Sicherstellung von Konsistenzen und Nachvollziehbarkeit wurde ein strukturierter Prozess zur Datenerhebung aus den Studien umgesetzt. Hierzu wurde eine eigene Extraktionsvorlage entwickelt, die die wesentlichen Merkmale der identifizierten Studien erfasst. Diese Merkmale wurden anschließend entlang von fünf zentralen Dimensionen kategorisiert, wie in Tabelle 3.2 dargestellt.

Tabelle 3.2: Datenextraktion

Dimension	Beschreibung
Forschungskontext	Beschreibt Ziel, Umfang und Art der Studie.
KI-Ansatz und Methode	Erfasst die verwendeten KI- oder ML-Verfahren.
Platform-Domänen	Ordnet den Beitrag einem Bereich des Platform Engineerings zu.
Ergebnisse und Use-Cases	Fasst die zentralen Erkenntnisse, Anwendungsfälle oder Evaluationsergebnisse zusammen.
Forschungslücke	Dokumentiert identifizierte Limitationen und Ansätze für zukünftige Arbeiten.

Die Extraktion erfolgte qualitativ, wobei relevante Textpassagen und zentrale Aussagen aus jeder Publikation manuell erfasst und den entsprechenden Dimensionen zugeordnet wurden

3.3.5. Datenanalyse und Kategorisierung

Nach der Datenerhebung wurden die Ergebnisse zusammengeführt und ausgewertet, um zentrale Themen, Muster und Forschungslücken zu erkennen. Die ausgewählten Studien wurden nach ihren Inhalten und Schwerpunkten strukturiert und den Forschungsfragen zugeordnet. Zur besseren Übersicht erfolgte die Kategorisierung der Arbeit entlang wichtiger Bereiche des Platform Engineerings. Innerhalb dieser Kategorien wurden die identifizierten KI-Ansätze, Anwendungsfälle und Herausforderungen miteinander verglichen, um wiederkehrende Trends sichtbar zu machen. Die Ergebnisse der Datenanalyse und Kategorisierung werden anschließend in Form einer Mapping Study dargestellt. Diese Übersicht zeigt, in welchen Themenfelder bereits Forschungsschwerpunkte existieren und wo noch Forschungslücken bestehen.

4

Literatur Ergebnisse

Dieses Kapitel präsentiert die Ergebnisse der im methodischen Vorgehen beschriebenen Literaturuntersuchung. Zunächst werden in Abschnitt 4.1 die quantitativ erhobenen Merkmale der ausgewählten Publikationen ausgewertet. Abschnitt 4.2 stellt anschließend die Ergebnisse der eigentlichen Mapping Study vor, indem die Arbeiten systematisch klassifiziert und Muster sichtbar gemacht werden. Darauf aufbauend fasst Abschnitt 4.3 die identifizierten KI-Anwendungen im Platform-Engineering zusammen und leitet ein erstes strukturiertes Matching im Form eines konzeptionellen Frameworks ab. Die dargestellten Ergebnisse bilden die Grundlage für die Beantwortung der Forschungsfragen in Kapitel 6.

4.1. Quantitative Analyse

Zur Einordnung des untersuchten Forschungsfeldes wurden zunächst grundlegende Merkmale der insgesamt 21 Studien analysiert. Abbildung 4.1 a) zeigt die jährliche Verteilung der Publikationen sowie die Zuordnung zu verschiedenen Publikationstypen. Zwischen 2021 und 2023 erscheinen nur wenige Arbeiten (insgesamt fünf), während ab 2024 ein deutlicher Anstieg sichtbar wird. In den Jahren 2024 und 2025 wurden jeweils acht Publikationen identifiziert, überwiegend Journalartikel, ergänzt durch einzelne Konferenzbeiträge, ArXiv-Papers und ein Whitepaper. Dies weist auf ein zunehmendes wissenschaftliches Interesse am Einsatz von KI in Cloud-Native- und Platform-Engineering-Kontexten hin.

Ergänzend dazu zeigt eine Word Cloud (Abbildung 4.1 b) die am häufigsten vorkom-

menden Keywords aus allen Publikationen. Die Visualisierung bietet einen schnellen Überblick über zentrale thematische Schwerpunkte der Literatur und unterstützt die anschließende inhaltliche Analyse.

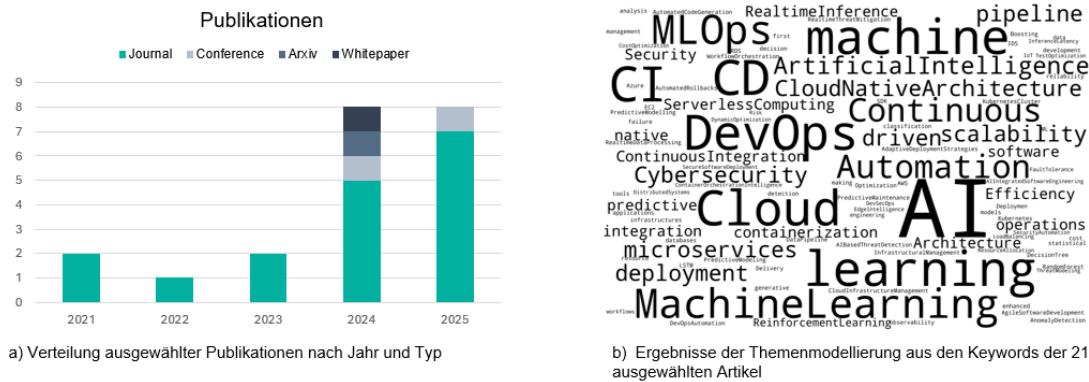


Abbildung 4.1: Jährliche und thematische Verteilung der DevOps AI-Forschung

Diese Betrachtung bildet die Grundlage für die folgenden Unterkapitel, in denen die Studien hinsichtlich ihrer inhaltlichen Merkmale detaillierter ausgewertet werden.

4.1.1. Anwendungsbereiche der KI im Platform Engineering

Im ersten Schritt der quantitativen Analyse wurden die insgesamt 21 identifizierten Studien hinsichtlich ihrer Hauptanwendungsbereiche untersucht. Dazu wurden die ausgewählten Bereiche in fünf Kategorien zusammenfasst: Optimierung von CI/CD Pipelines, Ressourcen- und Workload Optimierung, Sicherheits- und Bedrohungserkennung, Resilienz, Self-Healing und Predictive Maintenance sowie Intelligente Bereitstellung (Edge-AI und Serverless Deployment).

Die Auswertung zeigt, dass CI/CD-Automatisierung mit acht Publikationen am häufigsten als primärer Anwendungsbereich beschrieben wird. Darauf folgt die Ressourcen- und Workloadoptimierung, was in vier Paper als primärer Anwendun identifiziert wurde. Ebenfalls wird die intelligente Bereitstellung mittels Edge-AI oder Serverless Deploy- ment in vier Publikationen als zentraler Anwendungsbereich beschrieben. Der vierte Bereich, Resilienz, Self-Healing und Predictive Maintenance wird insgesamt drei mal als Hauptanwendungsbereich definiert. Sicherheits- und Bedrohungserkennung steht in zwei Studien im Mittelpunkt. Abbildung 4.2 zeigt diese Verteilung. Zu beachten ist, dass viele Studien mehr als einen Bereich ansprechen. Für die Vergleichbarkeit wurde jedoch jeweils der dominante Use Case pro Publikation ausgewählt. In der Praxis überschneiden sich die Anwendungsbereiche häufig, da KI-Lösungen oft mehrere Aufgaben gleichzeitig unterstützen.

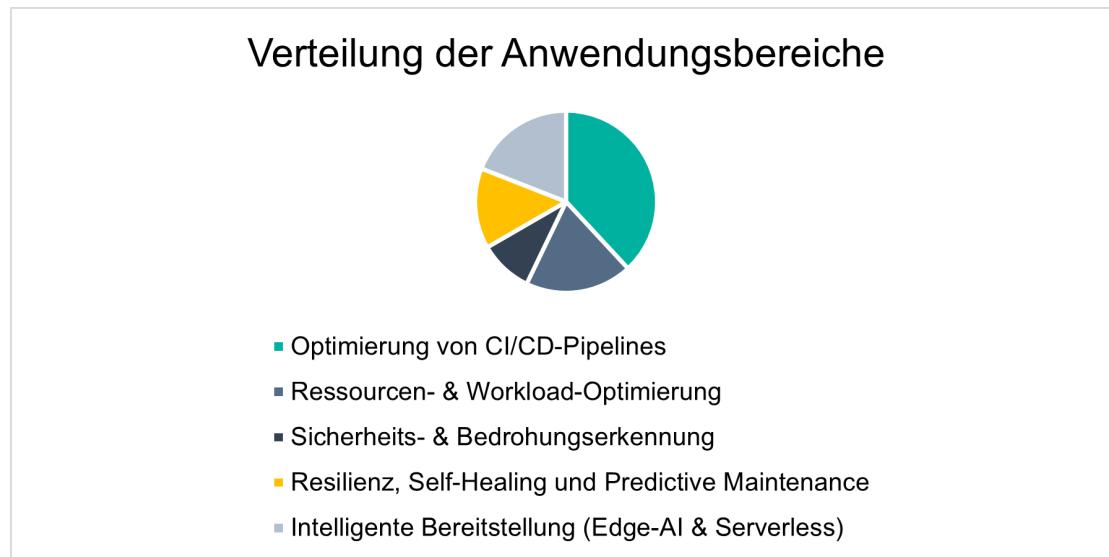


Abbildung 4.2: Verteilung der Anwendungsbereiche

4.1.2. Herausforderungen der KI-Integration im Platform Engineering

Im nächsten Schritt wurden die in den Publikationen beschriebenen Herausforderungen analysiert, die beim Einsatz von KI im Platform Engineering auftreten. Die analysierten Paper wurden fünf Kategorien zugeordnet. Die Auswertung zeigt, dass 95 % der Studien Herausforderungen im Bereich Ressourcenverbrauch und Kosten nennen. Ebenfalls häufig angesprochen werden Skalierbarkeit, Latenz und Monitoring mit 90 %. AI Governance, Datenschutz und Compliance findet sich in 81 % der Arbeiten wieder, ebenso wie die Kategorie Integrationskomplexität und Abhängigkeiten. Die fünfte Kategorie, Datenqualität, Datenverfügbarkeit und Heterogenität, wird in insgesamt 71 % der Arbeiten als Herausforderung thematisiert.

Diese Ergebnisse verdeutlichen, dass KI-Anwendungen im Platform Engineering häufig mit mehreren technischen, operativen und organisatorischen Hürden verbunden sind. Viele der Herausforderungen treten zudem gleichzeitig auf, da beispielsweise Skalierungsprobleme oft in engem Zusammenhang mit Ressourcenkosten oder Datenqualität stehen. Die prozentuale Verteilung ist in Abbildung 4.3 dargestellt.

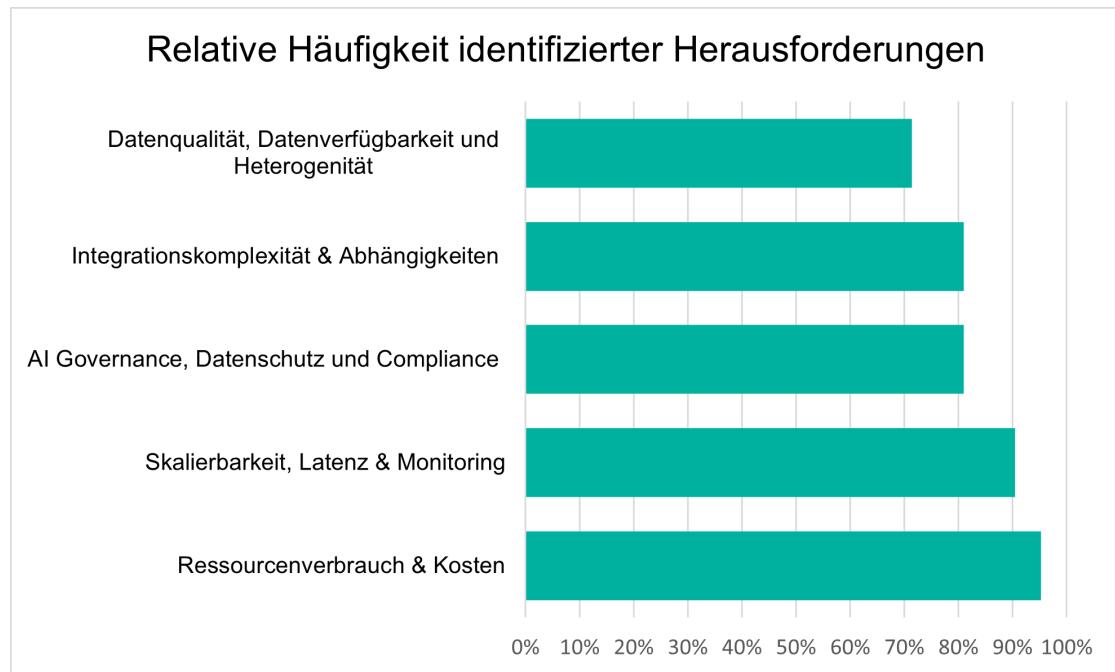


Abbildung 4.3: Relative Häufigkeit der Herausforderungen

4.1.3. Formen des maschinellen Lernens

Ein weiterer Bestandteil der quantitativen Analyse umfasst die in den Publikationen verwendeten Formen bzw. Lernparadigmen des maschinellen Lernens. Dabei wurde unterschieden, ob die jeweilige Form explizit, implizit genannt oder explizit erwähnt wird ohne tiefer einzugehen. Insgesamt wurden die drei grundlegenden Paradigmen identifiziert: Supervised Learning, Unsupervised Learning und Reinforcement Learning. In Kapitel 2.1.3 werden diese Paradigmen detaillierter erläutert und durch typische Einsatzformen und Methoden veranschaulicht.

Die Auswertung zeigt, dass Supervised Learning in den meisten Arbeiten eine zentrale Rolle spielt. Es wird es in sechs Publikationen ausdrücklich beschrieben, in elf Arbeiten erkennbar und in zwei kurz erwähnt. Reinforcement Learning wird insgesamt sieben mal explizit genannt und in vier weiteren Arbeiten erwähnt. Unsupervised Learning tritt mit drei expliziten und acht impliziten Nennungen seltener auf, ist jedoch ebenfalls präsent.

Die Verteilung ist in der folgenden Abbildung 2.1.3 dargestellt.

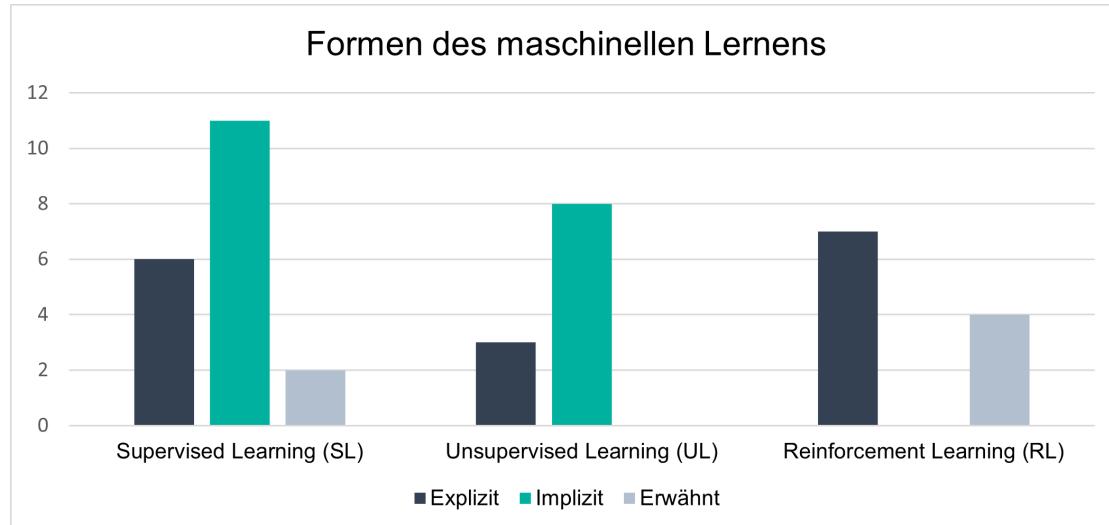


Abbildung 4.4: Formen des maschinellen Lernens

Auffällig ist die deutliche Differenz zwischen explizit und implizit erkennbaren Anwendungen, insbesondere bei Supervised und Unsupervised Learning. Dies zeigt, dass viele Publikationen die entsprechenden Paradigmen beschreiben, ohne die zugrunde liegende Lernform ausdrücklich zu benennen.

4.1.4. Verwendete Algorithmen

Die in den Publikationen verwendeten Algorithmen lassen sich den jeweiligen Lernparadigmen zuordnen. Die Analyse zeigt, dass eine Vielfalt an KI- und ML-Verfahren im Platform-Engineering-Kontext eingesetzt wird. Für eine systematische Bewertung wurde eine eigene Kategorisierung entwickelt. Den Ausgangspunkt bildet die Tabelle aus Enemosah [2], in der verschiedene AI-Techniken für Testfallpriorisierung beschrieben werden. Die Einteilung wurde für den breiteren Kontext dieser Arbeit angepasst.

Auf dieser Basis wurden vier Kategorien definiert, die in Kapitel 2.1 kurz beschrieben sind. Anschließend folgt die quantitative Auswertung der identifizierten Methoden und Algorithmen. Dies zeigt, wie häufig die jeweiligen Verfahren in den Publikationen genannt werden.

Die Ergebnisse zeigen ein deutliches Übergewicht von Deep-Learning-Verfahren, die in insgesamt 86 % der Publikationen eingesetzt oder thematisiert werden. Klassische Klassifikations- und Regressionsmethoden treten mit 52 % ebenfalls häufig auf. Demgegenüber werden Clustering-Verfahren und Ensemble- bzw. baumbasierte Modelle in 33 % der Arbeiten angesprochen. Insgesamt zeigt sich, dass insbesondere neuronale Netze und Deep-Learning-Verfahren die dominierenden Methoden sind.

Diese Verteilung ist in der Abbildung 4.5 dargestellt.

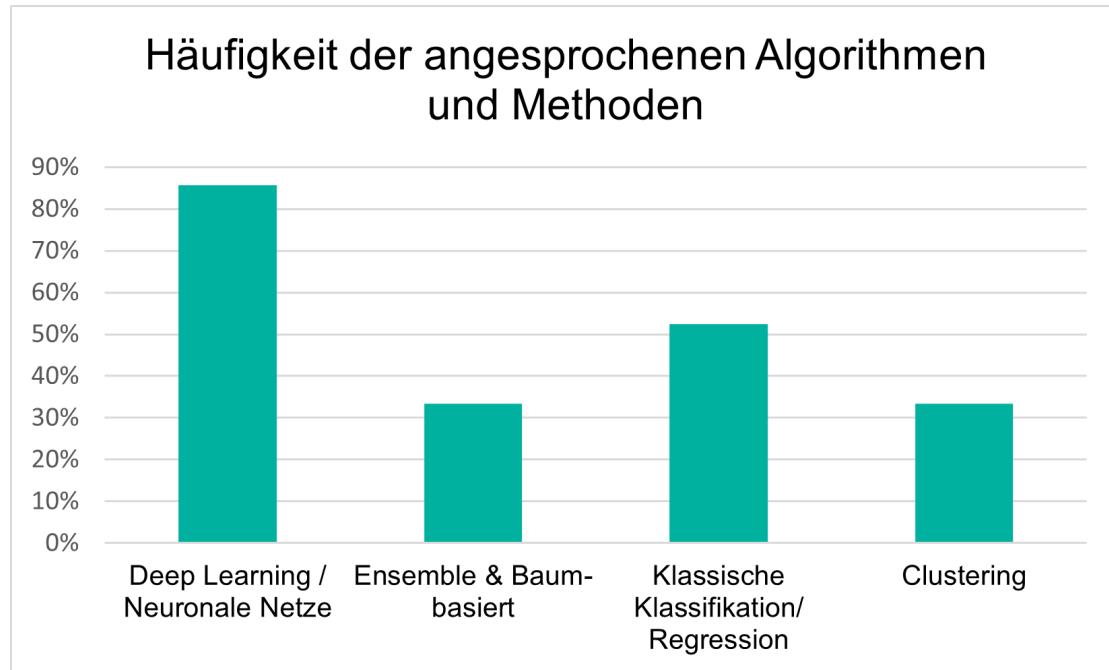


Abbildung 4.5: Verteilung der Algorithmen und angesprochenen Methoden

4.2. Ergebnisse der Mapping Study

Die Ergebnisse der Mapping Studie wurden in Hierzu wurden die Ergebnisse in Form von Bubble Chart Diagrammen visualisiert, in denen Blasengröße die Häufigkeit der entsprechenden Zuordnung widerspiegelt.

4.2.1. Zusammenspiel der Anwendungsfelder und Herausforderungen

Die Abbildung 4.6 zeigt das Zusammenspiel zwischen den ausgewählten Use cases und zentralen Herausforderungen im Platform Engineering. Im Gegensatz zur Abbildung 4.2, wurden hier alle Anwendungsbereiche pro Paper einbezogen und den angesprochenen Herausforderungen zugeordnet. Die in der Mapping Study extrahierten Häufigkeiten verdeutlichen, dass bestimmte Use cases besonders stark mit einer Vielzahl technischer und organisatorischer Herausforderungen zusammenhängen.

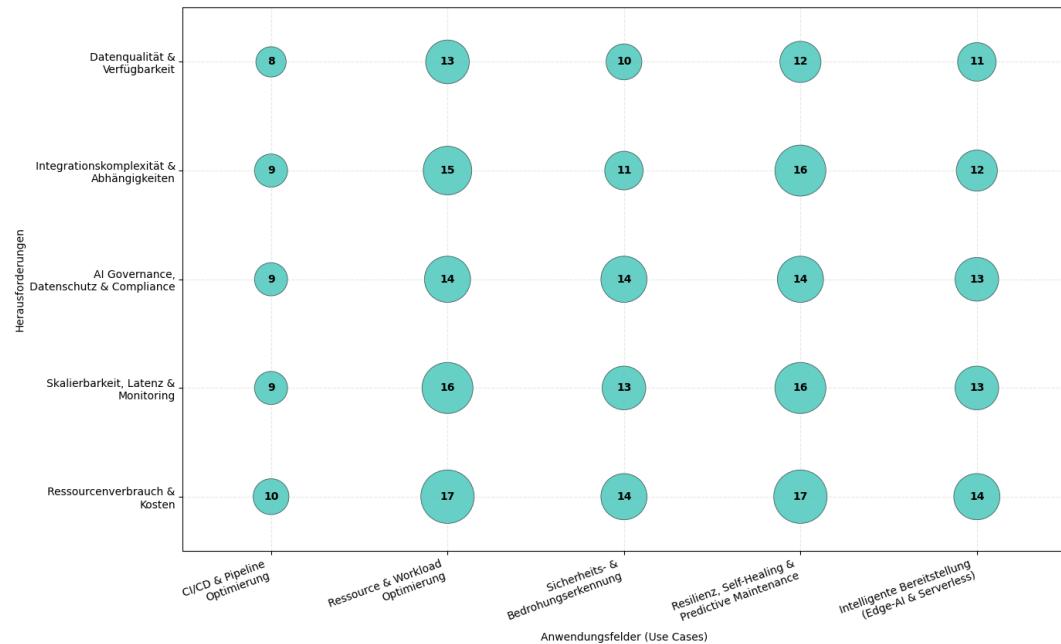


Abbildung 4.6: Korrelation zwischen Anwendungsbereichen und Herausforderungen

Am stärksten ausgeprägt sind die Zusammenhänge in den Use Cases Ressource und Workload Optimierung sowie Self-Healing und Predictive Maintenance, die in allen fünf Herausforderungskategorien hohe Werte erreichen. Diese Cluster zeigen, dass beide Anwendungsbereiche eng mit betrieblichen und technischen Anforderungen zusammenhängen und häufig als zentrale Ansatzpunkte für KI-basierte Optimierung diskutiert werden.

Der Anwendungsbereich Security und Threat Detection weist insbesondere in der Herausforderung Governance und Security eine hohe Ausprägung auf, während die übrigen Kategorien leicht geringere, aber dennoch hohe Werte zeigen. Dies deutet darauf hin, dass sicherheitsbezogene KI-Ansätze in der Literatur vor allem im Kontext von Compliance, Risikoerkennung und Schutzmechanismen betrachtet werden. Allerdings spielt diese Herausforderung auch im Monitoring sowie im Ressourcenmanagement eine Rolle.

Vergleichsweise etwas weniger thematisiert, aber dennoch deutlich erkennbar, sind die Werte im Bereich Intelligente Bereitstellung (Edge-AI und Serverless). Diese Verteilung spiegelt wider, dass diese Use Cases weniger stark im Mittelpunkt der untersuchten Literatur stehen als die zuvor genannten Anwendungsbereiche.

Insgesamt zeigt die Verteilung der Häufigkeiten, dass KI-Anwendungen im Platform Engineering vor allem dort adressiert werden, wo Effizienz, Stabilität und Sicherheitsaspekte zusammenspielen. Die identifizierten Zahlen verdeutlichen bevorzugte Kombinationen aus Anwendungsbereiche und Herausforderungen und ermöglichen damit

eine strukturierte Einordnung bestehender Forschungsschwerpunkte.

4.2.2. Zusammenspiel der Lernparadigmen und Algorithmen

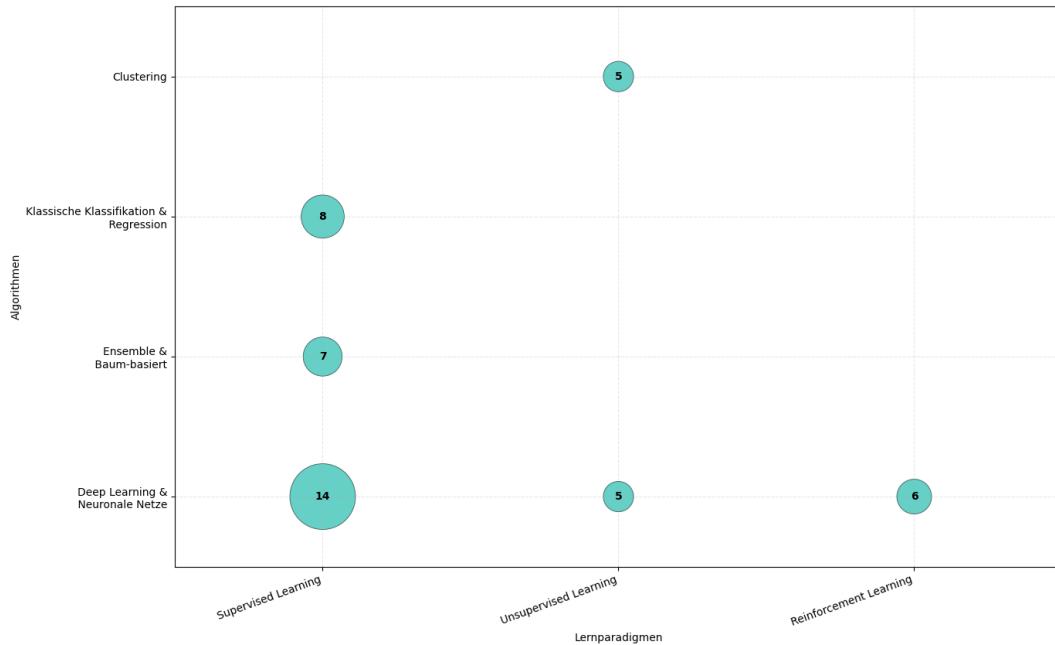


Abbildung 4.7: Korrelation zwischen Lernparadigmen und Algorithmen

4.2.3. Mapping von Anwendungsfeldern und Lernparadigmen

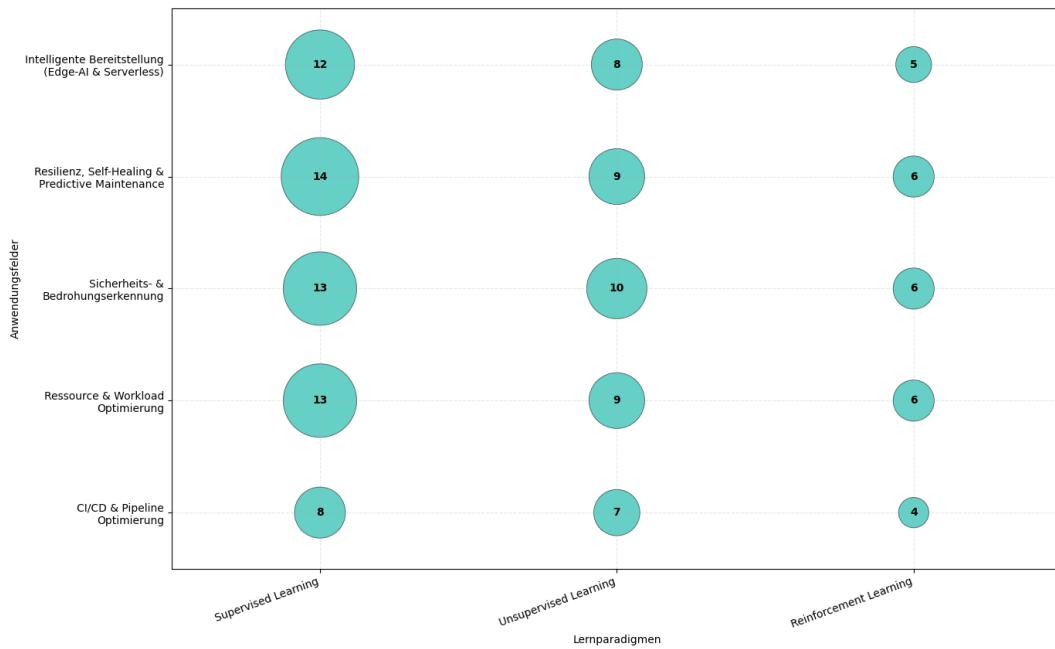


Abbildung 4.8: Mapping von Anwendungsfeldern und Lernparadigmen

4.2.4. Mapping von Herausforderungen und Lernparadigmen

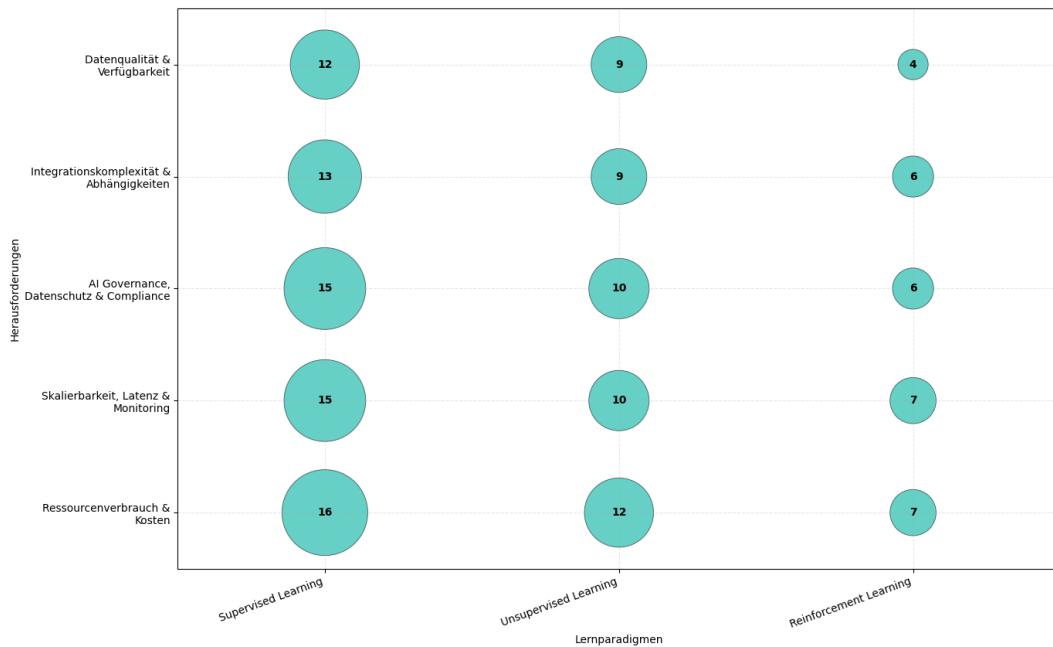


Abbildung 4.9: Mapping von Herausforderungen und Lernparadigmen

4.3. Matching/ Framework

Beschreibung des Matching-/Framework-Ansatzes und seiner Ergebnisse.

4.3.1. Überblick und Kategorisierung der KI-Anwendungen

5

Anwendung der Literaturrecherche

Darstellung des theoretischen Konzepts, seiner Bestandteile und Begründung.

5.1. Bewertungskonzept/ Framework

Hier kommt meine Antwort zu RQ3 hin.

5.2. Analyse der Bosch Digital Manufacturing Plattform

6

Diskussion

Die Ergebnisse werden eingeordnet, Limitationen diskutiert und Implikationen abgeleitet.

6.1. Beantwortung der Forschungsfragen

Zusammenföhrung der Ergebnisse zur direkten Beantwortung der Forschungsfragen.

6.2. Praxisrückschluss

Übertragbarkeit und Nutzen der Ergebnisse für die Praxis.

6.3. Limitationen

Grenzen der Studie und Ansatzpunkte für Verbesserungen.

6.4. Handlungsempfehlungen

An Bosch gerichtet

7

Zusammenfassung und Ausblick

Abschließende Zusammenfassung der Arbeit sowie ein Ausblick auf zukünftige Forschung.

7.1. Zusammenfassung

Die wichtigsten Erkenntnisse und Ergebnisse der Arbeit werden hier zusammengefasst.

7.2. Ausblick

Mögliche zukünftige Forschungsrichtungen und offene Fragen werden hier diskutiert.

Tabellenverzeichnis

2.1 Beschreibung Algorithmen und Methoden	4
3.1 Auswahlkriterien	8
3.2 Datenextraktion	9

Abbildungsverzeichnis

4.1	Jährliche und thematische Verteilung der DevOps AI-Forschung	11
4.2	Verteilung der Anwendungsbereiche	12
4.3	Relative Häufigkeit der Herausforderungen	13
4.4	Formen des maschinellen Lernens	14
4.5	Verteilung der Algorithmen und angesporchenen Methoden	15
4.6	Korrelation zwischen Anwendungsbereichen und Herausforderungen . .	16
4.7	Korrelation zwischen Lernparadigmen und Algorithmen	17
4.8	Mapping von Anwendungsfeldern und Lernparadigmen	18
4.9	Mapping von Herausforderungen und Lernparadigmen	19

Literaturverzeichnis

- [1] Rahul Amte. "Cloud-Native AI: Challenges and Innovations in Deploying Large-Scale Machine Learning Models". In: *ISCSITR - INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (ISCSITR-IJSRAIML) ISSN (Online)*: 3067-753X 6.2 (März 2025), S. 9–18. (Besucht am 05. 11. 2025).
- [2] Aliyu Enemosah. "Enhancing DevOps Efficiency through AI-Driven Predictive Models for Continuous Integration and Deployment Pipelines". In: *International Journal of Research Publication and Reviews* 6.1 (Jan. 2025), S. 871–887. ISSN: 25827421. DOI: [10.55248/gengpi.6.0125.0229](https://doi.org/10.55248/gengpi.6.0125.0229). (Besucht am 03. 11. 2025).
- [3] Kai Petersen, Sairam Vakkalanka und Ludwik Kuzniarz. "Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update". In: *Information and Software Technology* 64 (Aug. 2015), S. 1–18. ISSN: 0950-5849. DOI: [10.1016/j.infsof.2015.03.007](https://doi.org/10.1016/j.infsof.2015.03.007). (Besucht am 06. 11. 2025).
- [4] Barbara Kitchenham und Stuart M. Charters. ([PDF](#)) *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. <https://www.researchgate.net/publication/301230123> (Besucht am 06. 11. 2025).
- [5] Claes Wohlin. "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. EASE '14. New York, NY, USA: Association for Computing Machinery, Mai 2014, S. 1–10. ISBN: 978-1-4503-2476-2. DOI: [10.1145/2601248.2601268](https://doi.org/10.1145/2601248.2601268). (Besucht am 06. 11. 2025).
- [6] Muhammad Talha Tahir Bajwa u.a. "CLOUD-NATIVE ARCHITECTURES FOR LARGE-SCALE AI-BASED PREDICTIVE MODELING". In: *Journal of Emerging Technology and Digital Transformation* 4.2 (Aug. 2025), S. 207–221. ISSN: 3006-9726. (Besucht am 25. 10. 2025).
- [7] Majid Ghasemi u.a. *An Introduction to Reinforcement Learning: Fundamental Concepts and Practical Applications*. Aug. 2024. DOI: [10.48550/arXiv.2408.07712](https://arxiv.org/abs/2408.07712).

- [8] Vijay Govindarajan. *Machine Learning Based Approach for Handling Imbalanced Data for Intrusion Detection in the Cloud Environment*. März 2025, S. 815. DOI: [10.1109/ICDT63985.2025.10986614](https://doi.org/10.1109/ICDT63985.2025.10986614).
- [9] Abhishek Gupta und Yashovardhan Chaturvedi. "Cloud-Native ML: Architecting AI Solutions for Cloud-First Infrastructures". In: *Nanotechnology Perceptions* 20 (Dez. 2024), S. 930–939. DOI: [10.62441/nano-ntp.v20i7.4004](https://doi.org/10.62441/nano-ntp.v20i7.4004).
- [10] Yu Jeffrey Hu, Jeroen Rombouts und Ines Wilms. *MLOps Monitoring at Scale for Digital Platforms*. Apr. 2025. DOI: [10.48550/arXiv.2504.16789](https://doi.org/10.48550/arXiv.2504.16789). arXiv: [2504.16789 \[econ\]](https://arxiv.org/abs/2504.16789). (Besucht am 18.11.2025).
- [11] Giridhar Kankanala und Sudheer Amgothu. "AI/ML – DevOps Automation". In: 13 (Okt. 2024), S. 111–117.
- [12] Karthik Puthraya, Rachit Gupta und Beverly DSouza. "The Role of Cloud-Native Architectures in Accelerating Machine Learning Workflows through Data Engineering Innovations". In: (Feb. 2025). ISSN: 2945-3437. DOI: [10.5281/ZENODO.15106432](https://doi.org/10.5281/ZENODO.15106432). (Besucht am 05.11.2025).
- [13] Gopinath Kathiresan. "Cybersecurity Risk Modeling in CI/CD Pipelines Using Reinforcement Learning for Test Optimization". In: *International Journal of Innovative Science and Research Technology* (Mai 2025), S. 15–25. DOI: [10.38124/ijisrt/25may339](https://doi.org/10.38124/ijisrt/25may339). (Besucht am 03.11.2025).
- [14] Alexander Keusch u. a. "Platform-Agnostic MLOps on Edge, Fog and Cloud Platforms in Industrial IoT:" In: *Proceedings of the 20th International Conference on Web Information Systems and Technologies*. Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2024, S. 71–79. ISBN: 978-989-758-718-4. DOI: [10.5220/0012977500003825](https://doi.org/10.5220/0012977500003825). (Besucht am 18.11.2025).
- [15] Dominik Kreuzberger, Niklas Kühl und Sebastian Hirschl. "Machine Learning Operations (MLOps): Overview, Definition, and Architecture". In: *IEEE Access* 11 (2023), S. 31866–31879. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2023.3262138](https://doi.org/10.1109/ACCESS.2023.3262138). (Besucht am 03.11.2025).
- [16] Yao Lu u. a. *Computing in the Era of Large Generative Models: From Cloud-Native to AI-Native*. Jan. 2024. DOI: [10.48550/arXiv.2401.12230](https://doi.org/10.48550/arXiv.2401.12230). arXiv: [2401.12230 \[cs\]](https://arxiv.org/abs/2401.12230[cs]). (Besucht am 12.11.2025).
- [17] *MLOps Approach in the Cloud-Native Data Pipeline Design — Acta Technica Jaurinensis*. <https://acta.sze.hu/index.php/acta/article/view/581>. (Besucht am 04.11.2025).
- [18] Swarup Panda. *Scalable Artificial Intelligence Systems: Cloud-Native, Edge-AI, MLOps, and Governance for Real-World Deployment*. Deep Science Publishing, Juli 2025. ISBN: 978-93-7185-753-6.

- [19] Sudip Poudel u. a. "AI-Driven Intelligent Auto-Scaling for Cloud Resource Optimization 1". In: *Journal of Advanced College of Engineering and Management* Vol. 11 (Okt. 2025), S. 27–36. DOI: [10.3126/jacem.v11i1.84521](https://doi.org/10.3126/jacem.v11i1.84521).
- [20] Satheesh Reddy Gopireddy. "Integrating AI into DevOps: Leveraging Machine Learning for Intelligent Automation in Azure". In: *International Journal of Science and Research (IJSR)* 11.6 (Juni 2022), S. 2035–2039. ISSN: 23197064. DOI: [10.21275/SR22619111757](https://doi.org/10.21275/SR22619111757). (Besucht am 03.11.2025).
- [21] Vijay Kartik Sikha. "Cloud-Native Application Development for AI-Conducive Architectures." In: *International Journal on Recent and Innovation Trends in Computing and Communication* 11.11 (2023).
- [22] Suprit Pattanayak, Pranav Murthy und Aditya Mehra. "Integrating AI into DevOps Pipelines: Continuous Integration, Continuous Delivery, and Automation in Infrastructural Management: Projections for Future". In: *International Journal of Science and Research Archive* 13.1 (Okt. 2024), S. 2244–2256. ISSN: 25828185. DOI: [10.30574/ijjsra.2024.13.1.1838](https://doi.org/10.30574/ijjsra.2024.13.1.1838). (Besucht am 03.11.2025).
- [23] Venkata Mohit Tamanampudi. "AI-Enhanced Continuous Integration and Continuous Deployment Pipelines: Leveraging Machine Learning Models for Predictive Failure Detection, Automated Rollbacks, and Adaptive Deployment Strategies in Agile Software Development". In: 10 () .
- [24] Varun Tamminedi. "Automating Kubernetes Operations with AI and Machine Learning". In: *IJFMR - International Journal For Multidisciplinary Research* 6.6 (Dez. 2024). ISSN: 2582-2160. DOI: [10.36948/ijfmr.2024.v06i06.33430](https://doi.org/10.36948/ijfmr.2024.v06i06.33430). (Besucht am 03.11.2025).
- [25] Sumanth Tatineni. *Integrating Artificial Intelligence with DevOps: Advanced Techniques, Predictive Analytics, and Automation for Real-Time Optimization and Security in Modern Software Development*. Libertatem Media Private Limited, März 2024. ISBN: 978-81-971382-1-8.
- [26] Jeanette Uddoh u. a. "AI-Based Threat Detection Systems for Cloud Infrastructure: Architecture, Challenges, and Opportunities". In: *Journal of Frontiers in Multidisciplinary Research* 2.2 (2021), S. 61–67. ISSN: 30509718, 30509726. DOI: [10.54660/.IJFMR.2021.2.2.61-67](https://doi.org/10.54660/.IJFMR.2021.2.2.61-67). (Besucht am 04.11.2025).
- [27] Dirk Valkenborg u. a. "Supervised Learning". In: *American Journal of Orthodontics and Dentofacial Orthopedics* 164.1 (Juli 2023), S. 146–149. ISSN: 08895406. DOI: [10.1016/j.ajodo.2023.04.010](https://doi.org/10.1016/j.ajodo.2023.04.010). (Besucht am 27.11.2025).

- [28] Dirk Valkenborg u. a. "Unsupervised Learning". In: *American Journal of Orthodontics and Dentofacial Orthopedics* 163.6 (Juni 2023), S. 877–882. ISSN: 08895406. DOI: [10.1016/j.ajodo.2023.04.001](https://doi.org/10.1016/j.ajodo.2023.04.001). (Besucht am 27.11.2025).
- [29] Adel Zaalouk u. a. "CLOUD NATIVE ARTIFICIAL INTELLIGENCE". In: ().