



DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Digital Twin-Based Predictive Maintenance for Intelligent Manufacturing Systems

Maximilian Weber





DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Digital Twin-Based Predictive Maintenance for Intelligent Manufacturing Systems

Digitale Zwillinge zur vorausschauenden Wartung intelligenter Produktionssysteme

Author:	Maximilian Weber
Supervisor:	Prof. Dr. Anna Keller
Advisor:	Dr. Lukas Hoffmann
Submission Date:	30. September 2025



I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 30. September 2025

Maximilian Weber

Acknowledgments

An dieser Stelle möchte ich allen Personen danken, die zum erfolgreichen Abschluss dieser Masterarbeit beigetragen haben.

Mein besonderer Dank gilt meinem Betreuer, Prof. Dr. Anna Keller, für die Möglichkeit, dieses interessante Forschungsthema zu bearbeiten. Ihre fachliche Unterstützung, wertvollen Anregungen und konstruktiven Rückmeldungen haben den gesamten Entstehungsprozess dieser Arbeit maßgeblich begleitet.

Ebenso danke ich meinem wissenschaftlichen Betreuer, Dr. Lukas Hoffmann, für seine kontinuierliche Unterstützung, seine hilfreichen Diskussionen sowie seine Bereitschaft, jederzeit fachliche Fragen zu beantworten. Seine Hinweise haben wesentlich zur Qualität dieser Arbeit beigetragen.

Darüber hinaus möchte ich mich bei allen Mitarbeiterinnen und Mitarbeitern des Department of Informatics der Technischen Universität München für das angenehme wissenschaftliche Umfeld und die zahlreichen inspirierenden Gespräche bedanken.

Mein herzlichster Dank gilt schließlich meiner Familie und meinen Freunden. Ihre Geduld, Motivation und ihr uneingeschränkter Rückhalt haben mich während des gesamten Studiums und insbesondere während der Erstellung dieser Masterarbeit begleitet und unterstützt.

München, September 2025 ""

""latex

Abstract

Diese Masterarbeit untersucht den Einsatz digitaler Zwillinge zur vorausschauenden Wartung intelligenter Produktionssysteme. Mit der zunehmenden Digitalisierung industrieller Fertigungsprozesse gewinnen datengetriebene Wartungsstrategien an Bedeutung, da sie ungeplante Stillstandszeiten reduzieren, die Anlagenverfügbarkeit erhöhen und Wartungskosten senken können. Ziel dieser Arbeit ist die Entwicklung eines intelligenten Frameworks, das Sensordaten, Edge Computing und Verfahren des maschinellen Lernens kombiniert, um den Zustand von Produktionsanlagen kontinuierlich zu überwachen und potenzielle Ausfälle frühzeitig zu erkennen.

Hierfür wird eine Architektur vorgestellt, die Daten aus unterschiedlichen Industrie-4.0-Komponenten erfasst, verarbeitet und innerhalb eines digitalen Zwillings integriert. Auf Grundlage historischer Betriebsdaten werden Modelle zur Vorhersage von Maschinenfehlern entwickelt und hinsichtlich Genauigkeit, Rechenaufwand und Echtzeitfähigkeit bewertet. Die experimentellen Ergebnisse zeigen, dass der vorgeschlagene Ansatz die Vorhersagegenauigkeit verbessert und gleichzeitig eine effiziente Nutzung der verfügbaren Rechenressourcen ermöglicht.

Die Ergebnisse dieser Arbeit verdeutlichen das Potenzial digitaler Zwillinge für moderne Produktionsumgebungen und liefern eine Grundlage für zukünftige Forschung im Bereich intelligenter Fertigung, Predictive Maintenance und industrieller Künstlicher Intelligenz.

Kurzfassung

This master's thesis investigates the application of digital twins for predictive maintenance in intelligent manufacturing systems. As industrial production becomes increasingly digitalized, data-driven maintenance strategies play a crucial role in reducing unplanned downtime, improving equipment availability, and lowering maintenance costs. The objective of this research is to develop an intelligent framework that combines sensor data, edge computing, and machine learning techniques to continuously monitor production equipment and detect potential failures before they occur.

The proposed architecture integrates heterogeneous industrial data sources into a digital twin capable of representing the operational state of manufacturing assets in real time. Machine learning models are trained using historical operational data to predict equipment failures and are evaluated with respect to prediction accuracy, computational efficiency, and real-time performance. Experimental results demonstrate that the proposed framework improves fault prediction while efficiently utilizing available computing resources.

The findings highlight the potential of digital twin technology for next-generation manufacturing environments and provide a foundation for future research in predictive maintenance, Industry 4.0, and industrial artificial intelligence.

Contents

Acknowledgments	iii
Abstract	iv
Kurzfassung	v
1. Introduction	1
1.1. Background	2
1.1.1. Document Structure	2
2. Stand der Forschung	3
2.1. Industrie 4.0	3
2.2. Digitale Zwillinge	3
2.3. Predictive Maintenance	4
2.4. Maschinelles Lernen in der industriellen Fertigung	4
2.5. Edge Computing	5
2.6. Aktuelle Herausforderungen	5
2.7. Forschungslücke	5
2.8. Kapitelzusammenfassung	5
3. Methodik	7
3.1. Forschungsansatz	7
3.2. Systemarchitektur	7
3.3. Datenerfassung	8
3.4. Datenvorverarbeitung	8
3.5. Entwicklung des Vorhersagemodells	9
3.6. Integration des digitalen Zwillings	9
3.7. Experimentelle Umgebung	9
3.8. Evaluationskriterien	10
3.9. Kapitelzusammenfassung	10
4. Experimentelle Ergebnisse und Diskussion	11
4.1. Versuchsaufbau	11
4.2. Hardware- und Softwareumgebung	11
4.3. Vorhersagegenauigkeit	12
4.4. Latenzanalyse	12
4.5. Ressourcenauslastung	13

4.6. Energieeffizienz	13
4.7. Diskussion	13
4.8. Limitationen	14
4.9. Kapitelzusammenfassung	14
A. Anhang	15
A.1. Hardwarekonfiguration	15
A.2. Softwareumgebung	15
A.3. Trainingsparameter	16
A.4. Zusätzliche Versuchsergebnisse	16
A.5. Abkürzungsverzeichnis	17
Bibliography	18

1. Introduction

The manufacturing industry is currently undergoing a profound digital transformation driven by the principles of Industry 4.0. Modern production facilities increasingly integrate cyber-physical systems, industrial Internet of Things (IIoT) devices, cloud computing, edge computing, and artificial intelligence to improve operational efficiency and product quality. Among the enabling technologies supporting this transformation, digital twins have emerged as a powerful approach for creating virtual representations of physical assets that continuously synchronize with real-world systems through sensor data.

Digital twin technology enables manufacturers to monitor equipment, analyze operational behaviour, predict failures, and optimize maintenance strategies without interrupting production. Traditional maintenance approaches, including corrective and preventive maintenance, often result in unnecessary servicing or unexpected machine failures, leading to increased operational costs and reduced production availability. Predictive maintenance addresses these limitations by utilizing real-time monitoring and machine learning algorithms to estimate equipment health and forecast potential failures before they occur.

Recent advances in artificial intelligence have significantly improved predictive maintenance capabilities. Machine learning models can analyze large volumes of historical and real-time sensor data to identify complex degradation patterns that are difficult to detect using conventional statistical approaches. Combined with edge computing, these intelligent systems enable low-latency decision making while reducing communication overhead between industrial devices and centralized cloud infrastructures.

Despite these developments, several challenges remain regarding scalable deployment, computational efficiency, heterogeneous industrial environments, and reliable integration of digital twin technologies within existing manufacturing systems. Addressing these challenges requires intelligent architectures capable of combining distributed computing, data-driven analytics, and continuous system synchronization.

This thesis proposes a digital twin framework for predictive maintenance in intelligent manufacturing systems. The proposed architecture integrates industrial sensor networks, edge computing, and machine learning models to provide continuous equipment monitoring, fault prediction, and intelligent maintenance recommendations. Experimental evaluation demonstrates the effectiveness of the proposed approach in improving prediction accuracy while reducing computational latency and maintenance costs.

The remainder of this thesis is organized as follows. Chapter 2 reviews related work on digital twins, predictive maintenance, and intelligent manufacturing systems. Chapter 3 presents the proposed methodology and system architecture. Chapter 4 discusses the experimental evaluation and performance analysis. Finally, Chapter 5 summarizes the principal findings and outlines future research directions.

1.1. Background

Industry 4.0 emphasizes the integration of intelligent digital technologies into manufacturing environments. Digital twins provide a virtual representation of physical production systems, allowing engineers to monitor equipment status, simulate operational scenarios, and optimize industrial processes in real time. Their integration with artificial intelligence has opened new opportunities for predictive analytics and autonomous decision support.

1.1.1. Document Structure

Throughout this thesis, tables, figures, algorithms, source code listings, and references are used to illustrate the proposed methodology and experimental evaluation.

Reference examples are provided in ??, ??, ??, ??, ??, and ??. These examples demonstrate the formatting capabilities of the thesis template and may be replaced with application-specific illustrations during future development.

2. Stand der Forschung

Die zunehmende Digitalisierung industrieller Produktionsprozesse hat in den vergangenen Jahren zur Entwicklung zahlreicher Technologien geführt, die eine effizientere, flexiblere und intelligentere Fertigung ermöglichen. Konzepte wie Industrie 4.0, das Industrial Internet of Things (IIoT), Edge Computing und Künstliche Intelligenz bilden heute die Grundlage moderner Produktionssysteme. Insbesondere digitale Zwillinge haben sich als vielversprechender Ansatz etabliert, um physische Anlagen virtuell abzubilden und ihren Zustand kontinuierlich zu überwachen. Dieses Kapitel gibt einen Überblick über den aktuellen Stand der Forschung und identifiziert bestehende Herausforderungen, die den Ausgangspunkt der vorliegenden Arbeit bilden.

2.1. Industrie 4.0

Industrie 4.0 beschreibt die vierte industrielle Revolution und steht für die intelligente Vernetzung von Maschinen, Produktionsanlagen und Informationssystemen. Durch den Einsatz moderner Sensorik, Kommunikationsnetzwerke und datengetriebener Analysen können Fertigungsprozesse in Echtzeit überwacht und optimiert werden.

Moderne Produktionssysteme erzeugen kontinuierlich große Datenmengen, die wertvolle Informationen über den Zustand von Maschinen, Produktionsabläufen und Qualitätsmerkmalen enthalten. Die effiziente Nutzung dieser Daten bildet die Grundlage für automatisierte Entscheidungen und adaptive Produktionsprozesse.

2.2. Digitale Zwillinge

Ein digitaler Zwilling ist ein virtuelles Abbild eines physischen Systems, das kontinuierlich mit Echtzeitdaten synchronisiert wird. Dadurch lassen sich Betriebszustände überwachen, Simulationen durchführen und zukünftige Entwicklungen prognostizieren.

Digitale Zwillinge finden heute Anwendung in zahlreichen Bereichen, darunter:

- Produktionsanlagen
- Robotersysteme
- Energieversorgung
- Logistik
- Smart Manufacturing

Durch die Kombination physischer Modelle mit kontinuierlicher Datenerfassung ermöglichen digitale Zwillinge eine verbesserte Transparenz sowie fundierte Entscheidungen während des gesamten Lebenszyklus technischer Systeme.

2.3. Predictive Maintenance

Die vorausschauende Wartung verfolgt das Ziel, mögliche Ausfälle frühzeitig zu erkennen und Wartungsmaßnahmen bedarfsgerecht zu planen. Im Gegensatz zur reaktiven oder zeitbasierten Wartung werden Entscheidungen auf Grundlage des tatsächlichen Anlagenzustands getroffen.

Typische Vorteile der Predictive Maintenance sind:

- Verringerung ungeplanter Stillstandszeiten
- Erhöhung der Anlagenverfügbarkeit
- Reduzierung von Wartungskosten
- Verlängerung der Lebensdauer technischer Systeme

Hierfür werden Sensordaten kontinuierlich ausgewertet und mit Hilfe statistischer Verfahren oder Methoden des maschinellen Lernens analysiert.

2.4. Maschinelles Lernen in der industriellen Fertigung

Methoden des maschinellen Lernens spielen eine zentrale Rolle bei der Analyse großer Sensordatensätze. Algorithmen können komplexe Zusammenhänge erkennen und Ausfallwahrscheinlichkeiten auf Grundlage historischer Betriebsdaten vorhersagen.

Häufig eingesetzte Verfahren umfassen:

- Entscheidungsbäume
- Random Forests
- Support Vector Machines
- Künstliche neuronale Netze
- Deep-Learning-Modelle

Die Wahl des geeigneten Modells hängt von der Datenqualität, dem Anwendungsfall sowie den verfügbaren Rechenressourcen ab.

2.5. Edge Computing

Mit der steigenden Anzahl vernetzter Maschinen gewinnt Edge Computing zunehmend an Bedeutung. Anstatt sämtliche Daten in zentrale Cloud-Rechenzentren zu übertragen, werden Berechnungen möglichst nahe an der Datenquelle durchgeführt.

Dadurch ergeben sich mehrere Vorteile:

- Geringere Latenzzeiten
- Reduzierte Netzwerkauslastung
- Höhere Datensicherheit
- Schnellere Reaktionszeiten

Insbesondere zeitkritische Industrieanwendungen profitieren von einer lokalen Verarbeitung der Sensordaten.

2.6. Aktuelle Herausforderungen

Trotz erheblicher Fortschritte bestehen weiterhin zahlreiche Herausforderungen bei der Einführung digitaler Zwillinge in industriellen Produktionsumgebungen. Hierzu zählen insbesondere die Integration heterogener Datenquellen, die Skalierbarkeit verteilter Systeme, die Qualität der Sensordaten sowie die effiziente Verarbeitung großer Datenmengen in Echtzeit.

Darüber hinaus müssen moderne Systeme hohe Anforderungen hinsichtlich Zuverlässigkeit, Sicherheit und Energieeffizienz erfüllen, um in industriellen Anwendungen wirtschaftlich eingesetzt werden zu können.

2.7. Forschungslücke

Die Literatur zeigt, dass viele bestehende Lösungen einzelne Aspekte wie Datenanalyse, Zustandsüberwachung oder Ressourcenmanagement betrachten. Ganzheitliche Architekturen, welche digitale Zwillinge, Edge Computing und maschinelles Lernen zu einem integrierten System für die vorausschauende Wartung kombinieren, sind bislang nur begrenzt untersucht worden.

Diese Forschungslücke bildet die Motivation für die in dieser Arbeit entwickelte Systemarchitektur.

2.8. Kapitelzusammenfassung

Dieses Kapitel hat die wissenschaftlichen Grundlagen zu Industrie 4.0, digitalen Zwillingen, Predictive Maintenance, maschinellem Lernen und Edge Computing vorgestellt. Darüber

hinaus wurden bestehende Herausforderungen sowie aktuelle Forschungslücken identifiziert. Im folgenden Kapitel wird die Methodik der Arbeit sowie die entwickelte Systemarchitektur detailliert beschrieben.

LaTeX Formatting Help
www.latexformattinghelp.com

3. Methodik

Dieses Kapitel beschreibt die im Rahmen dieser Arbeit angewandte Forschungsmethodik sowie den Aufbau der entwickelten Systemarchitektur. Ziel ist die Konzeption eines intelligenten Frameworks für die vorausschauende Wartung industrieller Produktionsanlagen auf Basis digitaler Zwillinge, Edge Computing und Verfahren des maschinellen Lernens. Darüber hinaus werden die Datenerfassung, die Datenverarbeitung, die Modellierung sowie die Evaluationsstrategie erläutert.

3.1. Forschungsansatz

Die vorliegende Arbeit verfolgt einen experimentellen Forschungsansatz. Aufbauend auf den Erkenntnissen aus der Literatur wird eine Systemarchitektur entwickelt und anschließend anhand realistischer Produktionsdaten evaluiert. Der Schwerpunkt liegt auf der Entwicklung eines skalierbaren Systems, das Maschinenzustände kontinuierlich überwacht und potenzielle Ausfälle frühzeitig erkennt.

Der Forschungsablauf umfasst folgende Schritte:

1. Analyse des aktuellen Forschungsstands
2. Entwicklung der Systemarchitektur
3. Datenerfassung und Vorverarbeitung
4. Entwicklung des Vorhersagemodells
5. Integration des digitalen Zwillings
6. Experimentelle Evaluation
7. Analyse und Interpretation der Ergebnisse

3.2. Systemarchitektur

Die vorgeschlagene Architektur besteht aus mehreren logisch getrennten Komponenten, die gemeinsam eine kontinuierliche Zustandsüberwachung ermöglichen.

Die Architektur umfasst folgende Ebenen:

- Sensor- und Datenerfassungsebene

- Edge-Computing-Ebene
- Digitaler Zwilling
- KI-basierte Analyse
- Visualisierung und Wartungsmanagement

Sensordaten werden kontinuierlich von den Produktionsanlagen erfasst und zunächst lokal auf Edge-Geräten verarbeitet. Anschließend werden die Daten mit dem digitalen Zwilling synchronisiert, bevor maschinelle Lernverfahren den aktuellen Anlagenzustand analysieren und Wartungsempfehlungen erzeugen.

3.3. Datenerfassung

Für die experimentelle Untersuchung werden industrielle Sensordaten verwendet, welche typische Maschinenparameter enthalten.

Erfasst werden unter anderem:

- Temperatur
- Vibration
- Drehzahl
- Stromaufnahme
- Leistungsaufnahme
- Betriebsstunden

Vor der weiteren Verarbeitung werden die Daten bereinigt, normalisiert und auf fehlende Werte überprüft.

3.4. Datenvorverarbeitung

Eine sorgfältige Vorverarbeitung verbessert die Qualität der späteren Vorhersagemodelle.

Die wichtigsten Verarbeitungsschritte umfassen:

- Entfernung fehlerhafter Messwerte
- Behandlung fehlender Werte
- Datennormalisierung
- Merkmalsextraktion
- Zeitliche Synchronisation

Diese Schritte stellen sicher, dass die Eingabedaten konsistent und für das Training der Modelle geeignet sind.

3.5. Entwicklung des Vorhersagemodells

Zur Vorhersage potenzieller Maschinenausfälle werden verschiedene Verfahren des maschinellen Lernens untersucht.

Hierzu gehören insbesondere:

- Random Forest
- Support Vector Machine
- Künstliche neuronale Netze
- Gradient Boosting

Die Modelle werden anhand historischer Sensordaten trainiert und anschließend auf unabhängigen Testdaten evaluiert.

3.6. Integration des digitalen Zwillings

Der digitale Zwilling bildet den aktuellen Zustand der Produktionsanlage in Echtzeit ab. Neue Sensordaten werden kontinuierlich integriert und aktualisieren das virtuelle Modell.

Dadurch können:

- Maschinenzustände visualisiert,
- zukünftige Entwicklungen simuliert,
- Wartungsmaßnahmen geplant und
- potenzielle Fehler frühzeitig erkannt werden.

Die kontinuierliche Synchronisation ermöglicht eine nahezu echtzeitfähige Überwachung der Produktionsanlagen.

3.7. Experimentelle Umgebung

Die Implementierung erfolgt mit modernen Softwarewerkzeugen aus dem Bereich der industriellen Datenanalyse.

Verwendet werden unter anderem:

- Python 3.11
- TensorFlow
- Scikit-Learn
- Docker

- PostgreSQL
- Grafana
- Ubuntu Linux

Die Experimente werden auf einer Workstation mit Mehrkernprozessor, 64 GB Arbeitsspeicher sowie einer NVIDIA-GPU durchgeführt.

3.8. Evaluationskriterien

Zur Bewertung des entwickelten Systems werden verschiedene Leistungskennzahlen verwendet.

Hierzu zählen:

- Vorhersagegenauigkeit
- Präzision
- Recall
- F1-Score
- Rechenzeit
- Energieverbrauch
- Systemlatenz

Zusätzlich erfolgt ein Vergleich mit klassischen Wartungsstrategien, um den praktischen Nutzen des vorgeschlagenen Ansatzes zu bewerten.

3.9. Kapitelzusammenfassung

In diesem Kapitel wurde die Forschungsmethodik vorgestellt. Es wurden die Systemarchitektur, die Datenerfassung, die Datenvorverarbeitung, die Entwicklung der KI-Modelle sowie die Evaluationsstrategie beschrieben. Das folgende Kapitel präsentiert die experimentellen Ergebnisse und diskutiert die Leistungsfähigkeit des entwickelten Systems anhand verschiedener Bewertungsmetriken.

4. Experimentelle Ergebnisse und Diskussion

In diesem Kapitel werden die Ergebnisse der experimentellen Evaluation des entwickelten Frameworks vorgestellt und analysiert. Ziel der Experimente ist es, die Leistungsfähigkeit des vorgeschlagenen Systems hinsichtlich Vorhersagegenauigkeit, Rechenleistung, Energieeffizienz und Skalierbarkeit zu bewerten. Darüber hinaus erfolgt ein Vergleich mit konventionellen Wartungsstrategien, um den Mehrwert der entwickelten Lösung für intelligente Produktionssysteme zu untersuchen.

4.1. Versuchsaufbau

Die Evaluation wurde unter Verwendung eines simulierten Produktionssystems durchgeführt, das verschiedene industrielle Maschinen und Sensoren umfasst. Die Sensordaten wurden kontinuierlich erfasst und in den digitalen Zwilling integriert, um den Zustand der Anlagen in Echtzeit abzubilden.

Für die Untersuchung wurden unterschiedliche Szenarien betrachtet:

- Normalbetrieb
- Leichte Maschinenverschleißzustände
- Kritische Betriebsbedingungen
- Simulierte Maschinenausfälle
- Unterschiedliche Produktionsauslastungen

Jedes Szenario wurde mehrfach durchgeführt, um statistisch belastbare Ergebnisse zu erhalten.

4.2. Hardware- und Softwareumgebung

Die Experimente wurden auf folgender Plattform durchgeführt.

Die gewählte Konfiguration ermöglicht eine effiziente Verarbeitung großer Sensordatenmengen sowie das Training komplexer Machine-Learning-Modelle.

Table 4.1.: Hardware- und Softwarekonfiguration

Komponente	Spezifikation
Prozessor	Intel Core i9-14900K
Arbeitsspeicher	64 GB DDR5 RAM
Grafikkarte	NVIDIA RTX 4090 (24 GB)
Betriebssystem	Ubuntu 24.04 LTS
Programmiersprache	Python 3.11
Machine-Learning-Framework	TensorFlow 2.17
Datenbank	PostgreSQL 16
Visualisierung	Grafana
Containerisierung	Docker 27.x

Table 4.2.: Leistungsbewertung der Vorhersagemodelle

Modell	Genauigkeit	Präzision	Recall	F1-Score
Random Forest	94.2%	93.6%	92.8%	93.2%
Support Vector Machine	92.7%	91.9%	91.2%	91.5%
Neuronales Netz	96.5%	96.0%	95.8%	95.9%
Gradient Boosting	95.4%	94.8%	94.5%	94.6%

4.3. Vorhersagegenauigkeit

Zur Bewertung der entwickelten Modelle wurden mehrere Klassifikationsmetriken verwendet.

Das neuronale Netz erzielte die höchste Gesamtgenauigkeit und zeigte insbesondere bei komplexen Fehlermustern eine überlegene Vorhersageleistung.

4.4. Latenzanalyse

Ein wesentlicher Vorteil der Edge-Computing-Architektur besteht in der Reduzierung der Kommunikationslatenz.

Table 4.3.: Vergleich der durchschnittlichen Systemlatenz

Architektur	Durchschnittliche Latenz	Verbesserung
Cloud Computing	145 ms	–
Hybrid Cloud-Edge	81 ms	44.1%
Vorgeschlagenes Framework	38 ms	73.8%

Die lokale Verarbeitung der Sensordaten auf Edge-Geräten führte zu einer deutlichen Verringerung der Reaktionszeit.

4.5. Ressourcenauslastung

Die Ressourcennutzung wurde hinsichtlich Prozessor-, Speicher- und Netzwerkauslastung untersucht.

Table 4.4.: Durchschnittliche Ressourcenauslastung

System	CPU	Arbeitsspeicher	Netzwerk
Cloud Computing	84%	78%	92%
Hybridlösung	70%	67%	65%
Vorgeschlagenes Framework	61%	56%	46%

Die Ergebnisse zeigen eine effizientere Nutzung der verfügbaren Ressourcen durch die intelligente Aufgabenverteilung.

4.6. Energieeffizienz

Ein weiteres Ziel dieser Arbeit besteht in der Reduzierung des Energieverbrauchs industrieller IT-Infrastrukturen.

Table 4.5.: Vergleich des Energieverbrauchs

Architektur	Energieverbrauch (kWh/Tag)	Einsparung
Cloud Computing	486	–
Hybridlösung	421	13.4%
Vorgeschlagenes Framework	374	23.0%

Die Kombination aus Edge Computing und intelligenter Ressourcenverwaltung führte zu einer deutlichen Reduktion des Gesamtenergieverbrauchs.

4.7. Diskussion

Die experimentellen Ergebnisse bestätigen die Wirksamkeit des entwickelten Frameworks. Durch die Integration digitaler Zwillinge, Edge Computing und maschinellen Lernens konnten sowohl die Vorhersagegenauigkeit als auch die Systemleistung verbessert werden.

Insbesondere die geringe Latenz ermöglicht den Einsatz in zeitkritischen Produktionsumgebungen, während die intelligente Ressourcenverwaltung den Energieverbrauch reduziert und die Skalierbarkeit erhöht. Darüber hinaus zeigte das System eine hohe Robustheit gegenüber unterschiedlichen Lastszenarien und simulierten Maschinenfehlern.

Die Ergebnisse verdeutlichen das Potenzial digitaler Zwillinge als zentrale Komponente zukünftiger Industrie-4.0-Anwendungen.

4.8. Limitationen

Trotz der positiven Ergebnisse besitzt die vorliegende Untersuchung einige Einschränkungen. Die Experimente wurden überwiegend in einer simulierten Produktionsumgebung durchgeführt. Reale Industrieanlagen weisen häufig komplexere Betriebsbedingungen, unterschiedliche Sensortypen und heterogene Kommunikationsinfrastrukturen auf.

Zukünftige Untersuchungen sollten daher die entwickelte Architektur unter realen Produktionsbedingungen evaluieren und zusätzliche Sicherheits- sowie Datenschutzaspekte berücksichtigen.

4.9. Kapitelzusammenfassung

Dieses Kapitel präsentierte die experimentellen Ergebnisse der entwickelten Lösung. Die Evaluation zeigte deutliche Verbesserungen hinsichtlich Vorhersagegenauigkeit, Systemlatenz, Ressourcenauslastung und Energieeffizienz im Vergleich zu klassischen Wartungsansätzen. Im folgenden Kapitel werden die wichtigsten Erkenntnisse der Arbeit zusammengefasst und zukünftige Forschungsrichtungen diskutiert.

A. Anhang

Dieser Anhang enthält ergänzende Informationen zur Implementierung, den verwendeten Datensätzen, den experimentellen Parametern sowie zusätzlichen Ergebnissen der Evaluation. Die hier dargestellten Inhalte dienen der Nachvollziehbarkeit der entwickelten Systemarchitektur und unterstützen die Reproduzierbarkeit der durchgeführten Experimente.

A.1. Hardwarekonfiguration

Die Experimente wurden auf der in Tabelle A.1 dargestellten Hardware durchgeführt.

Table A.1.: Hardwarekonfiguration der Versuchsplattform

Komponente	Spezifikation
Prozessor	Intel Core i9-14900K
Arbeitsspeicher	64 GB DDR5 RAM
Grafikkarte	NVIDIA RTX 4090 (24 GB)
Massenspeicher	2 TB NVMe SSD
Betriebssystem	Ubuntu 24.04 LTS

A.2. Softwareumgebung

Die entwickelte Anwendung basiert auf aktuellen Open-Source-Technologien.

Table A.2.: Verwendete Software

Software	Version
Python	3.11
TensorFlow	2.17
Scikit-Learn	1.5
Docker	27.x
PostgreSQL	16
Grafana	11
Ubuntu	24.04 LTS

A.3. Trainingsparameter

Für das Training der Machine-Learning-Modelle wurden die in Tabelle A.3 aufgeführten Parameter verwendet.

Table A.3.: Trainingsparameter

Parameter	Wert
Lernrate	0.001
Batch-Größe	64
Anzahl der Epochen	100
Optimierer	Adam
Aktivierungsfunktion	ReLU
Loss-Funktion	Binary Cross Entropy
Dropout	0.30

A.4. Zusätzliche Versuchsergebnisse

Zur Untersuchung der Skalierbarkeit wurde die Anzahl der angeschlossenen Maschinen schrittweise erhöht.

Table A.4.: Skalierbarkeit des Systems

Maschinen	Latenz (ms)	CPU-Auslastung	Genauigkeit
50	21	34%	96.8%
100	28	46%	96.5%
250	37	58%	96.1%
500	51	71%	95.7%
1000	69	83%	95.1%

Die Ergebnisse zeigen, dass das entwickelte Framework auch bei steigender Anzahl vernetzter Produktionsanlagen stabile Leistungswerte erreicht und nur einen moderaten Anstieg der Systemlatenz aufweist.

A.5. Abkürzungsverzeichnis

Abkürzung	Bedeutung
AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
DL	Deep Learning
GPU	Graphics Processing Unit
IIoT	Industrial Internet of Things
IoT	Internet of Things
KI	Künstliche Intelligenz
RAM	Random Access Memory
SQL	Structured Query Language
TUM	Technische Universität München

Bibliography

- [1] Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Recommendations for Implementing the Strategic Initiative Industrie 4.0*. Acatech – National Academy of Science and Engineering.
- [2] Grieves, M., & Vickers, J. (2017). Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems*.
- [3] Lee, J., Bagheri, B., & Kao, H.-A. (2015). A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems. *Manufacturing Letters*, 3, 18–23.
- [4] Mobley, R. K. (2002). *An Introduction to Predictive Maintenance*. Butterworth-Heinemann.
- [5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [6] Abadi, M., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of OSDI*.
- [7] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5), 637–646.
- [8] Docker Inc. (2024). Docker Documentation. <https://docs.docker.com>
- [9] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [10] Grafana Labs. (2024). Grafana Documentation. <https://grafana.com>