

Schlussbericht

zu IGF-Vorhaben Nr. 20726 N

Thema

Cloud-basierte Assistenzsysteme zur Überwachung komplexer Produktionsanlagen und Anlagenkomponenten

Berichtszeitraum

01.06.2019 bis 31.05.2021

Forschungsvereinigung

Forschungsvereinigung Elektrotechnik beim ZVEI e.V. - FE
Lyoner Straße 9
60528 Frankfurt am Main

Forschungseinrichtung(en)

Fraunhofer Gesellschaft e.V.
Fraunhofer Kompetenzzentrum
Industrial Automation (IOSB)
Langenbruch 6
32657 Lemgo

Gefördert durch:

Inhaltsverzeichnis

Zusammenfassung.....	3
1 Gegenüberstellung der durchgeführten Arbeiten und des Ergebnisses mit den Zielen.....	4
Arbeitspakete.....	5
Arbeitspaket 1: Analyse und Anforderungsdefinition (2 PM).....	6
Ziele.....	6
Durchgeführte Arbeiten.....	6
Erzielte Ergebnisse.....	6
Arbeitspaket 2: Datenerfassung (4,5 PM).....	7
Ziele.....	7
Durchgeführte Arbeiten.....	7
Erzielte Ergebnisse.....	7
1. Datenerfassungskomponenten.....	7
2. Simulationsmodelle.....	8
3. Testdatensätze.....	10
Arbeitspaket 3: Condition Monitoring System.....	11
Arbeitspaket 3.1: Algorithmen zur Anomalie-Erkennung.....	11
Ziele.....	11
Durchgeführte Arbeiten.....	11
Erzielte Ergebnisse.....	12
SmartThreshold.....	12
Hidden Markov Modelle.....	12
Long Short-Term Memory (LSTM) Netze.....	15
Automaten.....	19
Petri-Netze.....	20
AP3.2 Fehlerursachen-Erkennung.....	27
Ziele.....	27
Durchgeführte Arbeiten.....	27
Erzielte Ergebnisse.....	27

AP3.3 Web-basierte Benutzerschnittstelle.....	30
Ziele.....	30
Durchgeführte Arbeiten.....	30
Erzielte Ergebnisse.....	30
AP3.4 Cloud-basiertes Modell-Lernen.....	33
Ziele.....	33
Durchgeführte Arbeiten.....	33
Erzielte Ergebnisse.....	33
Arbeitspaket 4: Evaluierung, Dokumentation und Projektmanagement.....	34
Arbeitspaket 4.1: Evaluierung.....	34
Ziele.....	34
Durchgeführte Arbeiten.....	34
Erzielte Ergebnisse.....	34
Arbeitspaket 4.2: Dokumentation.....	35
Ziele.....	35
Durchgeführte Arbeiten.....	35
Erzielte Ergebnisse.....	35
Arbeitspaket 4.3: Projektmanagement und Transfer.....	35
Ziele.....	35
Durchgeführte Arbeiten.....	35
Erzielte Ergebnisse.....	36
Projektergebnisse.....	36
2. Verwendung der Zuwendung.....	37
3. Notwendigkeit und Angemessenheit der geleisteten Arbeit.....	37
4. Darstellung des wissenschaftlich-technischen und wirtschaftlichen Nutzens der erzielten Ergebnisse insbesondere für KMU sowie ihres innovativen Beitrags und ihrer industriellen Anwendungsmöglichkeiten.....	38
5. Plan zum Ergebnistransfer in die Wirtschaft.....	40

Zusammenfassung

Ziel des Projektes war die Entwicklung eines Cloud-basierten Condition Monitoring Systems (CMS). Das CMS kann in komplexen Produktionsanlagen verschiedener Branchen (u.a. in der Holzverarbeitenden, der Automobil- und der Stahl-verarbeitenden Industrie) eingesetzt werden. Durch generische Prozessmodelle und maschinelle Lernverfahren kann es an verschiedene Produktionsanlagen und Kunden-spezifische Einsatzszenarien adaptiert werden. Das in dem Projekt entwickelte CMS besteht aus einem Modul zur Datenerfassung sowie aus selbstlernenden Modulen zur Anomalie-Erkennung und zur Diagnose. Offene wissenschaftliche Fragestellungen bei der Umsetzung des CMS ergaben sich hinsichtlich der Skalierung existierender selbstlernender Anomalie-Erkennungsverfahren auf komplexe Produktionsanlagen sowie die Entwicklung selbstlernender Diagnosemethoden zur Identifikation von Fehlerursachen.

Zu Projektbeginn wurden die Anforderungen an ein Cloud-basiertes CMS erfasst und dokumentiert. In der ersten Projekthälfte wurden weiterhin die Datenerfassung und die Anomalie-Erkennungsverfahren entwickelt. Darauf aufbauend wurde in der zweiten Projekthälfte zunächst die Fehlerursachen-Erkennung untersucht. In der letzten Projektphase wurden die Entwicklung der Benutzerschnittstellen, die Auslagerung rechenintensiver Prozesse in die Cloud und eine systematische Evaluation der entwickelten Methoden durchgeführt.

Durch das Projekt werden insbesondere KMU, denen die Ressourcen für Kunden-individuelle Implementierungen und Konfigurationen fehlen, in die Lage versetzt, CMS wirtschaftlich zu entwickeln und anzuwenden.

Das Forschungsziel wurde erreicht.

1 Gegenüberstellung der durchgeführten Arbeiten und des Ergebnisses mit den Zielen

Projektziele

Übergeordnetes Ziel des Projektes CLArA war es, ein selbstlernendes CMS für komplexere Produktionsanlagen und insbesondere auch für Anlagen des Sondermaschinenbaus zu entwickeln. Das entwickelte System kann sich durch den Einsatz von maschinellen Lernverfahren automatisch an existierende Anlagen anpassen sowie Anomalien und Fehlerursachen auf der Grundlage einer gelernten Repräsentation des Systemverhaltens detektieren. Dies wird dadurch erreicht werden, dass in einer Trainingsphase Analyse- und Diagnosemodelle aus historischen Prozessdaten gelernt und während des Betriebs des CMS zur Anomalie-Erkennung und zur Diagnose von Fehlerursachen verwendet werden (siehe Abbildung 1).

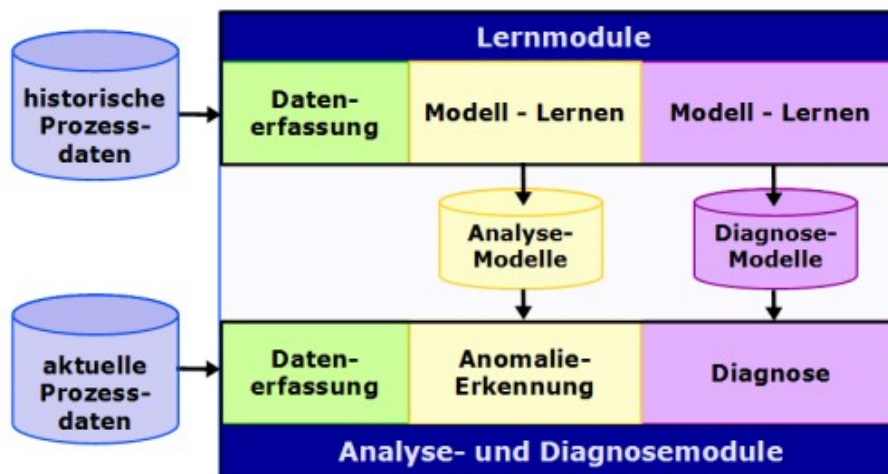


Abbildung 1: Selbstlernendes Condition Monitoring System für komplexe Produktionsanlagen: Datenerfassung, Anomalie-Erkennung und Diagnose

Auf diese Weise wird es dem Anwender ermöglicht, das CMS ohne aufwendige Programmierung und Konfiguration in verschiedenen Anlagen einzusetzen. Dadurch werden die Entwicklungs- und Inbetriebnahmekosten für CMS erheblich reduziert und es wird insbesondere ein wirtschaftlicher Einsatz in kundenspezifischen Anlagen, die nur in kleinen Stückzahlen produziert werden, erreicht. Im Einzelnen wurden bei der Realisierung des selbstlernenden CMS die folgenden Fragestellungen untersucht:

- 1 Wie können Prozessdaten in heterogenen Produktionsanlagen standardisiert erfasst werden?
- 2 Wie kann eine zuverlässige Anomalie-Erkennung in komplexen Produktionsanlagen durchgeführt werden?
- 3 Wie können Fehlerursachen mittels eines selbstlernende Diagnosemoduls automatisch identifiziert werden?

Zur Realisierung des selbstlernenden Condition Monitoring Systems wurden daten-getriebene Methoden zur Anomalie-Erkennung und zur Identifikation von Fehlerursachen erforscht sowie in charakteristischen Anwendungsszenarien aus unterschiedlichen Branchen erprobt.

Arbeitspakete

Die einzelnen Arbeitspakete des Projektes sind in **Abbildung 2** dargestellt.

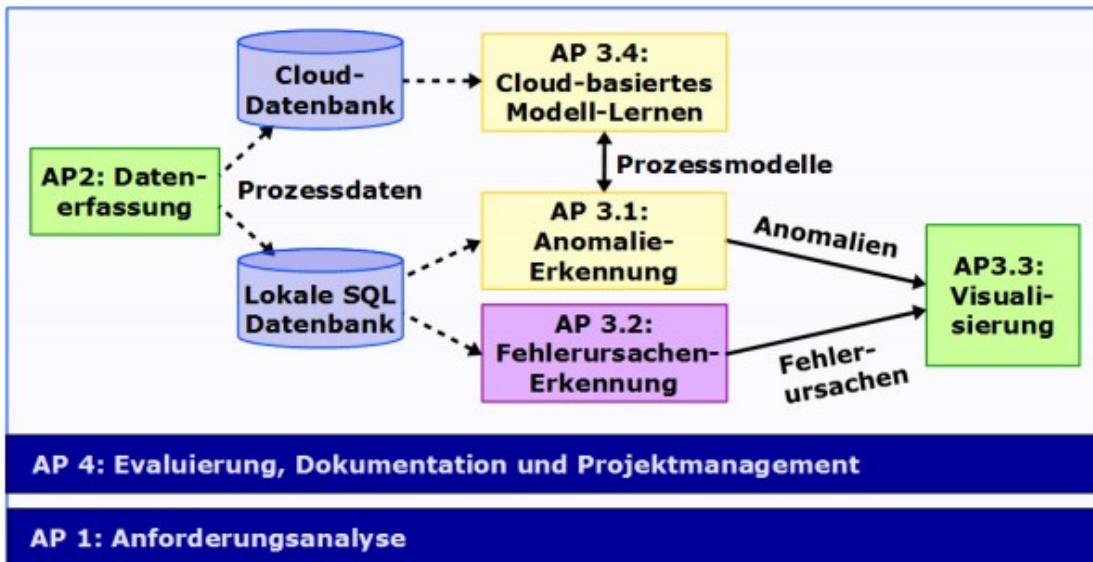


Abbildung 2: Arbeitspakete und Weiternutzung von Ergebnissen

A

AP 1 und AP 4 umfassen Arbeitsschritte, von denen alle im Projekt entwickelten Komponenten betroffen sind. In AP 2 und AP 3 sollen jeweils Teilkomponenten des in Abbildung 1 dargestellten CPS realisiert werden. Die Weiternutzung von Ergebnissen zwischen diesen APs ist in Abbildung 4 in Form von Pfeilen dargestellt.

Die Ziele, die durchgeführten Arbeiten und die Ergebnisse für die einzelnen Arbeitspakete sind in den folgenden Abschnitten dargestellt.

Arbeitspaket 1: Analyse und Anforderungsdefinition

Ziele

Ziel dieses Arbeitspakets war es, die Anforderungen an das CMS und die erforderliche Infrastruktur zu analysieren und zu definieren. Dabei sollten sowohl die Sicht der Anwender des CMS als auch die Sicht der Komponenten-Entwickler und Systemintegratoren berücksichtigt werden. Als Ergebnis sollte ein Dokument mit (i) Anlagenbeschreibungen, (ii) Anforderungen für die Einsatzszenarien, (iii) Beschreibungen der Messsignale und Kommunikationsschnittstellen, (iv) typischen Fehlersituationen sowie Wartungszyklen erstellt werden.

Durchgeführte Arbeiten

In Arbeitspaket 1 wurden die Anforderungen wie geplant mit den Projektpartnern abgestimmt und in einem Dokument festgehalten.

Im Rahmen der Analyse und Anforderungsdefinition wurden drei Beispielanlagen für die Evaluierung des Condition Monitoring Systems ausgewählt. In der betrachteten Verpackungsanlage des Unternehmens Becker wird Karton in vier Prozessschritten zugeschnitten und zur Verpackung eines Produktes verwendet. Typische Sensordaten, welche zum Condition Monitoring herangezogen werden können, sind die Signale von Lichttastern, Positions- und Belegungsmessungen sowie elektrische Leistungsdaten. Typische Fehlersituationen ergeben sich durch falsch positionierte oder verdrehte Produkte. Die zweite Beispielanlage ist eine Elektro-Schlacke-Umschmelzanlage der Deutschen Edelstahlwerke, welche die Produktion hochwertiger Stähle und Sonderlegierung durch das Umschmelzen von Stahlblöcken in einem Schlackebad ermöglicht. Als typische Fehlersituationen ergeben sich in diesem Anwendungsfall Druckabfälle, welche beispielsweise durch Undichtigkeiten oder durch einen unterbrochenen Argonfluss entstehen und sich negativ auf den Reinheitsgrad des raffinierten Stahls auswirken. Als dritter Anwendungsfall wurde ein Lackaustragungsband des Unternehmens Venjacob betrachtet. In diesem Anwendungsfall werden neben Messgeräten für elektrische Leistungsdaten ein Beschleunigungssensor sowie Temperatursensoren verwendet. Als Fehlersituation wurde hier die Abnutzung des Bandes betrachtet.

Erzielte Ergebnisse

Als Ergebnis der Analyse- und Anforderungsspezifikation wurde ein Dokument mit Beschreibungen der drei Anlagen sowie Anforderungen an die Einsatzszenarien erstellt, welches allen Interessenten auf Anfrage zur Verfügung gestellt wird. In diesem Dokument werden weiterhin die Messsignale und Kommunikationsschnittstellen beschrieben sowie typische Fehlersituationen spezifiziert. Neben den drei ausgewählten Anwendungsszenarien wurden drei weitere Datensätze spezifiziert, welche zur systematischen Evaluierung des Condition Monitoring Systems dienen.

Arbeitspaket 2: Datenerfassung

Ziele

In AP 2 sollte die für die Entwicklung der Analyse- und Diagnosemethoden grundlegende Datenerfassung realisiert werden. Dazu sollten insbesondere die folgenden Prozessdaten der Beispielanlagen von Becker und Venjakob mittels vorhandener Datenerfassungskomponenten auf Basis von OPC UA integriert, synchronisiert und gespeichert werden: Daten unmittelbar an die Steuerungen angeschlossener Sensoren (mittels S7/S5-OPC-UA-Server), Daten einer Ventilinsel mit OPC UA Server, Profinet-Daten (mittels am Fraunhofer IOSB entwickelter Profinet Datalogger mit OPC UA Schnittellen), Steuerungsdaten (mittels vorhandener OPC UA Schnittstellen der Steuerungen). Die bei der Datenerfassung erforderliche Informationsmodellierung sollte auf das notwendige Minimum beschränkt werden und soweit möglich auf der Basis vorhandener OPC UA Companion Standards erfolgen. Für in dem Projekt untersuchte Module und Anlage, für die solche Standards insbesondere in Hinblick auf den Anwendungsfall Condition Monitoring noch nicht existieren bzw. sich als unzureichend erweisen, sollten eigene Companion Standards definiert werden und in den Standardisierungsprozess von OPC UA einfließen. Die Daten aus den einzelnen Datenquellen sollten unter Verwendung von OPC UA an eine SQL-Datenbank sowie an die Cloud-basierte Infrastruktur angebunden werden. Sowohl die SQL-Datenbank als auch die Cloud-basierte Infrastruktur waren am Fraunhofer IOSB bereits zu Projektbeginn vorhanden und verfügen über die erforderlichen Kommunikationsschnittstellen. Im Rahmen von AP 2 sollten vor dem ersten Meilenstein bereits erste Testdaten aufgenommen werden, die in AP3.1 zur Algorithmen-Entwicklung genutzt werden können. Als Ergebnis des Arbeitspaketes sollten die für die Umsetzung des Projektes erforderlichen Datenerfassungskomponenten mit geeigneten Informationsmodellen installiert werden. Die erfassten Testdaten der Anwendungsszenarien sollten sich in der SQL-Datenbank und in der Cloud-basierten Infrastruktur befinden.

Durchgeführte Arbeiten

Im Rahmen des zweiten Arbeitspaketes wurden Daten in den drei Anwendungsfällen plangemäß erfasst und zur Auswertung gespeichert. Weiterhin wurden zwei Simulationsmodelle zur Erzeugung von Simulationsdaten für die systematische Entwicklung der Anomalie-Erkennung erstellt.

Erzielte Ergebnisse

1. Datenerfassungskomponenten

Der **OPC UA Connector** dient dazu, OPC UA Daten in einer InfluxDB zu speichern. Der Connector läuft in einem Docker Container, der mit den Verbindungsdaten für den OPC UA Server und die InfluxDB konfiguriert werden kann. In einer Node-Set Konfiguration kann spezifiziert werden, dass der OPC UA Connector sich nur für dort angegebene Knoten des OPC UA Namensraums registrieren soll. Alternativ ist es möglich, die Änderungen aller OPC UA Variablen in der Influx DB zu speichern.

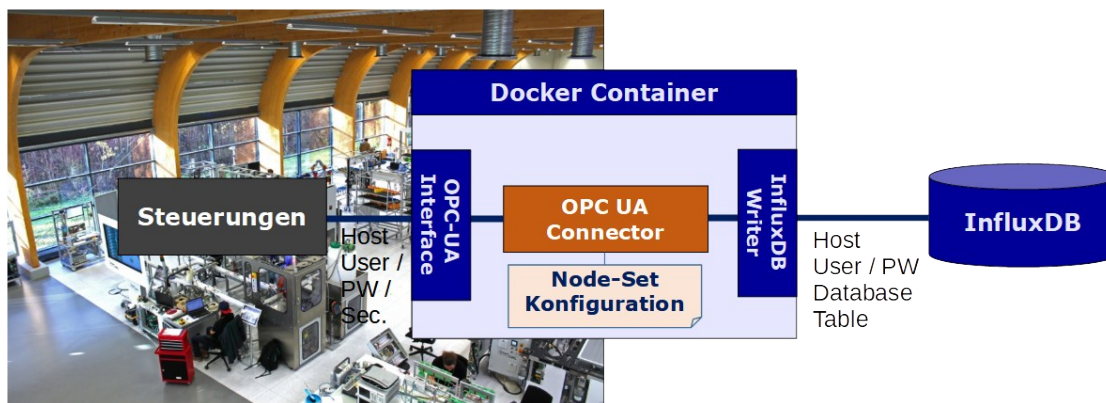


Abbildung 3: OPC UA Connector für das entwickelte Condition Monitoring System

Der **InfluxDB Writer** ermöglicht es, CSV-Daten in der InfluxDB abzuspeichern. Mit dem **InfluxDB Reader** können Daten in der InfluxDB in Form von CSV-Dateien abgespeichert werden.

In den exemplarischen Anwendungsfällen der Unternehmen Becker und Venjacob werden die Daten in Form von Log-Dateien erfasst. Eine solche Log-Datei enthält für jede Signaländerung eine eigene Zeile, in welcher der Timestamp der Signaländerung (bestehend aus Datum und Änderung), der Name des geänderten Signals sowie der neue Wert des Signals gespeichert ist. Der **Log-File Konverter** erzeugt aus den Log-Dateien jeweils CSV-Dateien, in denen zu einem Timestamp alle aktuellen Signalwerte gespeichert werden.

2. Simulationsmodelle

Im Rahmen des Projektes wurden zwei Simulationsmodelle implementiert, welche es ermöglichen, die entwickelten Verfahren zur Anomalie-Erkennung systematische zu evaluieren.

Tanksystem

Das entwickelte Simulationsmodell für ein Tanksystem ist in Abbildung 4 dargestellt. Der Simulationsablauf besteht – vereinfacht dargestellt – darin, das zunächst Tank 3 über Tank 1 und 2 befüllt wird. Anschließend werden die Flüssigkeiten vermischt, erhitzt und wieder abgekühlt. Am Ende des Zyklus wird Tank 3 über Tank 4 geleert. Tank 1 Tank 2 verfügen über eine eigene Regelung, auf deren Basis die Pumpen angeschaltet werden, sobald ein definierter Füllstand unterschritten ist. Das entwickelte Simulationsmodell bildet einen typischen hybriden Prozess mit diskreten Signalen (Schaltsignale für Ventile und Pumpen) sowie kontinuierlichen Signalen (z.B. Temperatur- und Füllstandmessungen). Weiterhin wurde ein Python-Skript erstellt, welches es ermöglicht, das Simulationsmodell mit verschiedenen Parametern auszuführen und auf diese Weise Testdatensätze zu erzeugen, die in Form von CSV-Dateien abgespeichert werden.

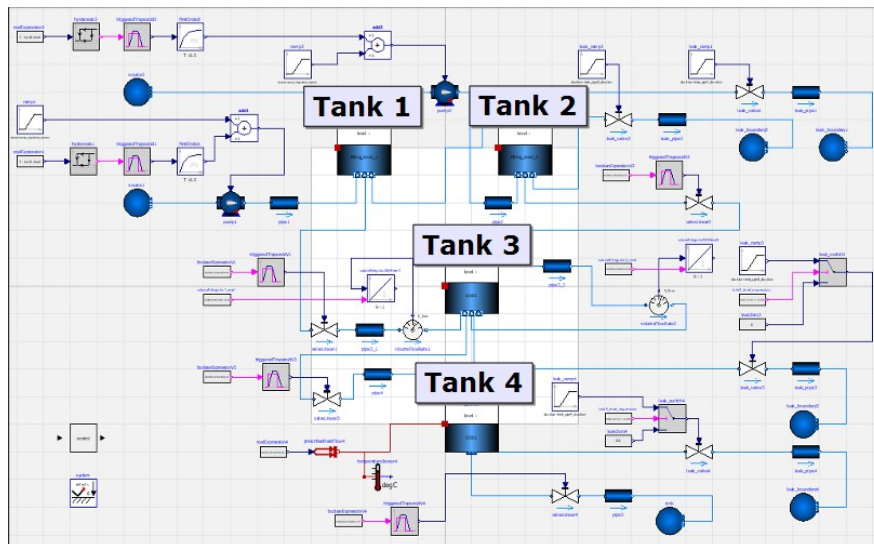


Abbildung 4: Simulationsmodell für ein 4-Tank-System

Transportsysteme

Um die Anomalie-Erkennung in diskreten Ereignissystemen systematisch evaluieren zu können, wurden Simulationsmodelle für drei Transportsysteme unterschiedlicher Komplexität erstellt (Modell A: 29 Förderbänder, Modell B: 12 Förderbänder, Modell C: 6 Förderbänder). Abbildung 5 zeigt exemplarisch das Simulationsmodell des Transportsystems mit 29 Förderbändern.

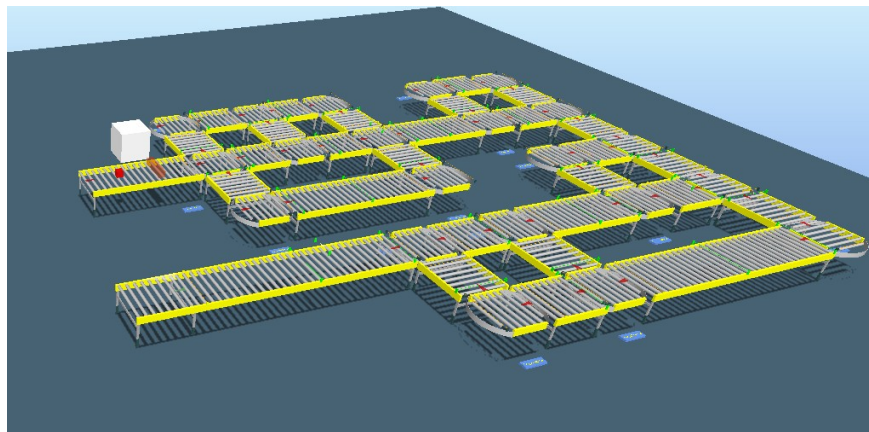


Abbildung 5: Simulationsmodell des Transportsystems mit 29 Förderbändern

In der Simulation bewegen sich Behälter auf verschiedenen Routen durch das jeweilige Transportsystem. Die einzelnen Förderbänder der Transportsysteme sind jeweils mit einem Encoder zur Positionsmessung sowie mit zwei Lichtschranken zur Detektion der Behälter ausgestattet. Die Sensordaten werden während der Simulation mittels einer Logging-Komponente in Form einer CSV-Datei abgespeichert. Während der Simulation ist es möglich, verschiedene Fehlerfälle zu erzeugen (z.B. Sensorausfall, Rotation oder Schlupf).

3. Testdatensätze

Als Ergebnis der Datenerfassung liegen die folgenden Datensätze vor, welche allen Projektpartnern und Interessierten zur Verfügung stehen:

1. Testdatensatz zur Verpackungsanlage des Unternehmens Becker:

16 Log-Files mit Signalen der vier Teilprozesse

- Insgesamt 47 Sensorsignale: Lichttaster, Positionen (z.B. von Gurtbändern), Belegungen (z.B. von Ablageplätzen), Stromaufnahmen, ...
- Diverse Fehlermeldungen: falsch positionierte oder verdrehte Produkte

2. Testdatensatz zum Elektro-Schlacke-Umschmelzprozess des Unternehmens DEW:

CSV-Dateien mit den Signalen der Elektro-Schlacke-Umschmelzanlage

- 22 Signale: Zur Erkennung der Anomalien wird hauptsächlich der gemessene Druck benötigt
- 459 Datensätze, teilweise mit Anomalien im Druckverlauf

3. Testdatensatz zur Lackierstraße des Unternehmens Venjacob

- Log-Files von fünf Tagen (Zeitraum: 11.06. – 24.09.2019)
- 14 Signale: Beschleunigungssignal, Temperaturmessungen, elektrische Leistungsdaten des Antriebs
- Fehlersituation: Abnutzung des Lackaustragungsbandes

4. Testdatensatz für ein Simulationsmodell des Unternehmens Venjacob

- 9 Signale: Positions- und Encoder-Signale der Förderbänder, ID des Gegenstandes am Wareneingang
- 12 Datensätze, teilweise mit Sensorfehlern oder fehlerhaften Längen der transportierten Gegenstände

5. Testdatensatz mit Simulationsdaten für hybride Prozesse

- 130 Datensätze mit Normalverhalten
- 80 Datensätze mit Fehlern (drei Fehlertypen: defekte Tanks, blockierte Ventile, Sensorfehler)
- Simulationsmodell und Python-Skripte zur automatischen Testdatengenerierung

6. Testdatensatz mit Simulationsdaten für Batch-Prozesse

- 4 Teil-Datensätze (unterschiedliche Anfangsbedingungen und Simulationsparameter),
- 400 Batches mit Normalverhalten, 22.000 Batches mit fehlerhaftem Verhalten (15 Fehlerarten) je Teil-Datensatz

7. Testdatensatz mit den Sensordaten verschiedener Simulationsdurchläufe in drei Transportsystemen unterschiedlicher Komplexität, für jedes Transportsystem

- 150 fehlerfreie Simulationsdurchläufe, in denen sich jeweils nur ein Behälter gleichzeitig durch das Transportsystem bewegt
- 200 fehlerfreie Simulationsdurchläufe, in denen sich 10-40 Behälter gleichzeitig durch das Transportsystem bewegen
- Jeweils 25 Simulationsdurchläufe für sechs verschiedene Fehlertypen, in denen sich 10-20 Behälter gleichzeitig durch das Transportsystem bewegen.

Arbeitspaket 3: Condition Monitoring System

Im Rahmen von AP 3 wurden Algorithmen zur Anomalie-Erkennung und zur Fehlerursachen-Erkennung entwickelt (AP 3.1 und AP 3.2). Weiterhin sind in AP 3 eine Web-Applikation und eine Cloud-basierte Infrastruktur für das CMS entstanden, in welche die Algorithmen integriert wurden (AP 3.3 und AP 3.4).

Arbeitspaket 3.1: Algorithmen zur Anomalie-Erkennung

Ziele

In AP 3.1 sollte ein Produkt-übergreifend wirksamer Erkennungsmechanismus für Anomalien bzw. Auffälligkeiten des Prozessverhaltens einer Produktionsanlage realisiert werden. Dabei sollen wie in Abschnitt 3.2 motiviert die folgenden Arbeitsschritte durchgeführt werden:

- 1 Realisierung des Condition Monitoring Frameworks (Anwendungslogik, Zugriff auf die Prozessdatenbank, Integration existierender Anomalie-Erkennungsverfahren auf Basis hybrider Automaten und tiefer neuronaler Netze) unter Berücksichtigung vorhandener Referenzarchitekturen wie VDMA 24582, ggf. Erarbeitung von Vorschlägen zur Weiterentwicklung der verwendeten Standards
- 2 Untersuchung von Verfahren zur Dimensionsreduktion (Hauptkomponentenanalyse, Faktorenanalyse, Auto-Encoder, etc.)
- 3 Skalierung der Prozessmodelle und Lernverfahren für hybride Automaten und tiefe neuronale Netze
- 4 Parallelisierung der Lernverfahren.

Die Migration und Evaluierung der parallelisierten Lernverfahren in der Cloud-basierten Infrastruktur des Fraunhofer IOSB INA erfolgte in AP 3.4. Das Ergebnis von AP 3.1 sind Modell-basierte Anomalie-Erkennungsverfahren.

Durchgeführte Arbeiten

Im Laufe des Projektes hat sich herausgestellt, dass die verschiedenen Anwendungsszenarien unterschiedliche Ansätze zur Anomalie-Erkennung erfordern. Während es sich bei den Transportsystemen der Unternehmen Venjacob und Becker überwiegend um diskrete Ereignissysteme handelt, werden in den exemplarischen Anwendungsszenarien des Unternehmens DEW vorwiegend kontinuierliche Sensorsignale verwendet. Daher wurden in dem Projekt sowohl Ansätze für kontinuierliche als auch für diskrete Systeme erarbeitet. Für kontinuierliche Systeme wurden drei Verfahren implementiert:

- SmartThreshold
- Hidden Markov Modell (HMM)
- Long Short-Term Memory (LSTM) Netz

Weiterhin wurde die Kombination von HMMs und LSTMs mit Variational Autoencodern untersucht, welche in den betrachteten Anwendungsszenarien jedoch zu keinen signifikanten Verbesserungen führte.

Für diskrete Ereignissysteme wurden Verfahren auf Basis der folgenden Modelle entwickelt:

- Automaten
- Petri-Netze

Nachfolgend werden die Projektergebnisse zu den einzelnen Verfahren beschrieben.

Erzielte Ergebnisse

SmartThreshold

SmartThreshold ist ein einfaches Baseline-Verfahren. Hier werden in einer Trainingsphase für jedes Sensorsignal die erfassten Minima und Maxima festgehalten. Falls sich ein Sensorsignal während der Betriebsphase des CMS außerhalb dieser Grenzwerte befindet, wird eine Anomalie angezeigt.

Hidden Markov Modelle

Die Anomalie-Erkennung mit Hidden Markov Modellen geht auf einen Ansatz zurück, der in [WJN2015] entwickelt wurde. Das Verfahren wurde im Rahmen des Projektes in Python implementiert und um ein dynamisches CUSUM Verfahren zur Residuenanalyse erweitert. Die verwendeten Hidden Markov Modelle (HMMs) ermöglichen es, sowohl diskrete Prozessphasen als auch kontinuierliches Prozessverhalten in den einzelnen Prozessphasen zu beschreiben. Dabei wird die Annahme getroffen, dass nur die kontinuierlichen Signale beobachtbar sind. Diese werden durch multivariate Gaußverteilungen repräsentiert, deren Parameter von der jeweils aktuellen Prozessphase abhängen. Die Parameter des HMMs können mit einem EM-Algorithmus wie dem Viterbi-Algorithmus oder dem Baum-Welch-Algorithmus ermittelt werden (siehe [WJN2015] für Details). Die resultierenden Modelle sind in Abbildung 6 für den zweidimensionalen Fall dargestellt, wobei „Signal 1“ und „Signal 2“ die beiden Komponenten des Signals sind. Man erkennt, dass die Signale durch das Modell stückweise approximiert werden, wobei der Modellierungsfehler in Form einer Standardabweichung σ ebenfalls gelernt wird. Die Fehlererkennung beruht auf der statistischen Analyse der Residuen zwischen Modell und Beobachtungen.

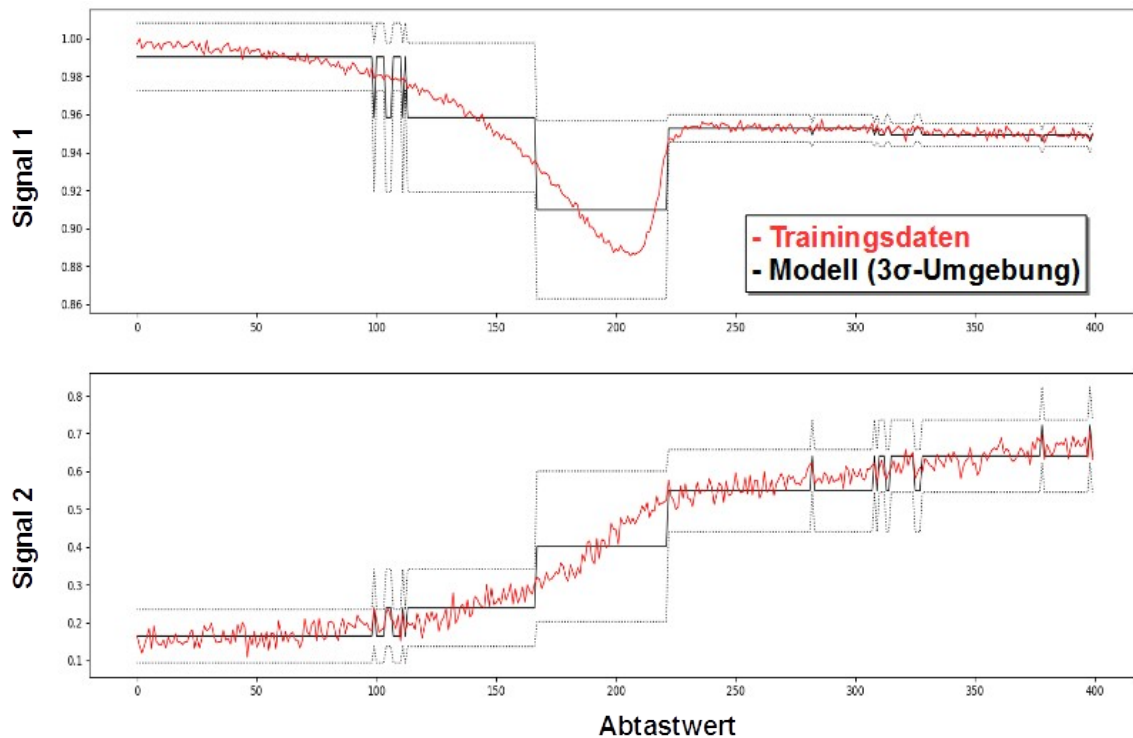


Abbildung 6: Trainingsdaten und gelerntes HMM für 2-dimensionale Messwerte

In [WJN2015]¹ wurde zu diesem Zweck die Likelihood der Beobachtungen in Bezug auf das gelernte Prozessmodell ausgewertet. Dieser Ansatz wurde im Rahmen des Projektes verbessert. Hierzu wurde ein neues Analyseverfahren (Dynamic CUSUM) für die Residuen von Hidden Markov Modellen entwickelt, welches auf den klassischen CUSUM-Verfahren für einfache Gauß-Verteilungen und Kalman Filter basiert (siehe z.B. [Granjon2013]²). Eine Veröffentlichung der Details des Dynamic CUSUM Verfahrens ist im Rahmen eines Artikels mit dem Arbeitstitel „Robust Data-Driven Fault Detection in Industrial Batch Processes based on a Stochastic Hybrid Process Model“ geplant.

Der Hidden Markov Modell basierte Ansatz wurde auf dem Datensatz des Unternehmens DEW sowie für das 4-Tank-System evaluiert.

Ergebnisse für die Elektro-Schlacke-Umschmelzanlage des Unternehmens DEW

In der des Unternehmens DEW Elektro-Schlacke-Umschmelzanlage wurde der Druckverlauf als zu untersuchendes Signal verwendet. Zeitschritte mit fehlenden Werten wurden entfernt und das Signal normalisiert. Außerdem wurden ein Moving-Average-Filter und ein Windowing auf die Zeitreihen angewandt. Es wurden 50 Sequenzen ohne Anomalie zum Training des Modells genutzt. Weiterhin wurden die verbliebenen 61 normalen Sequenzen, sowie 61 anormale Sequenzen zum Testen der Anomalieerkennung genutzt. Abbildung 11 zeigt exemplarisch die Prädiktion des HMMs für einen normalen Druckverlauf. In der Abbildung sind der tatsächliche

1 [WJN2015] S. Windmann, F. Jungbluth, O. Niggemann: „A HMM-Based Fault Detection Method for Piecewise Stationary Industrial Processes,“ in IEEE International Conference on Emerging Technologies and Factory Automation, Luxembourg, Sep. 2015.

2 [Granjon2013] P. Granjon: P. Granjon, “The CuSum algorithm - a small review,“ hal-00914697, 2013.

Signalverlauf rot und die 3σ -Umgebung um den vorhergesagten Signalverlauf. Es ist zu erkennen, dass der vorhergesagte Signalverlauf die 3σ -Umgebung nicht verlässt. Insgesamt ergab sich ein F1-Score von 0.976. Das Verfahren weist eine TP-Rate von 100.0 % und eine FP-Rate von 4.9 % auf.

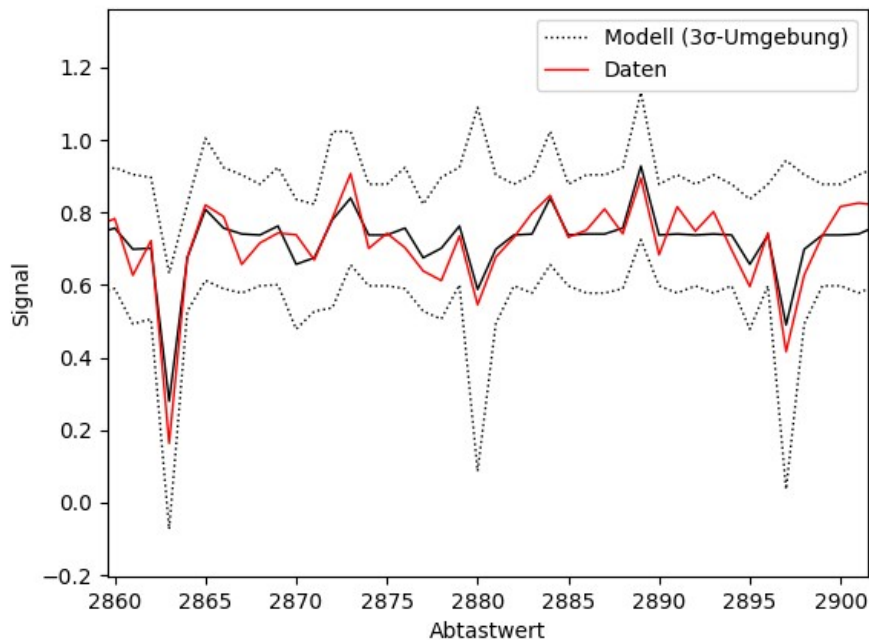


Abbildung 7: HMM-basierte Prädiktion des Druck-Signals des DEW Datensatzes

Ergebnisse für das 4-Tank-System

Für den Test des HMMs auf den Daten der in Abbildung 5 dargestellten Tanksystem-Simulation wurden die Signale tank3.level, tank4.level und temperatureSensor4.T sowie die Ableitungen der Signale tank1.level und tank2.level verwendet. Außerdem wurden tank3.level und tank4.level durch einen Schwellenwert binarisiert um das Verfahren auch für binäre Signale zu testen. Genauso wie in dem exemplarischen Anwendungsfall des Unternehmens DEW wurden die Signale normalisiert, sowie eine Moving-Average-Filter und eine Windowing-Funktion auf die Signale angewandt. 20 Sequenzen wurden als Trainingsdaten verwendet. Der Test erfolgte auf 70 normalen und 70 anormalen Sequenzen. Das HMM erreicht einen F1-Score von 0,78, bei einer TP-Rate von 78,6 % und einer FP-Rate von 21,4 %. In Abbildung 8 und Abbildung 9 ist zu sehen, dass das HMM binäre Signale ebenso prädizieren kann wie Werte-kontinuierliche (dargestellt sind genauso wie in Abbildung 7 jeweils der tatsächliche Signalverlauf als rote Linien sowie die 3σ -Umgebung als schwarze Linien).

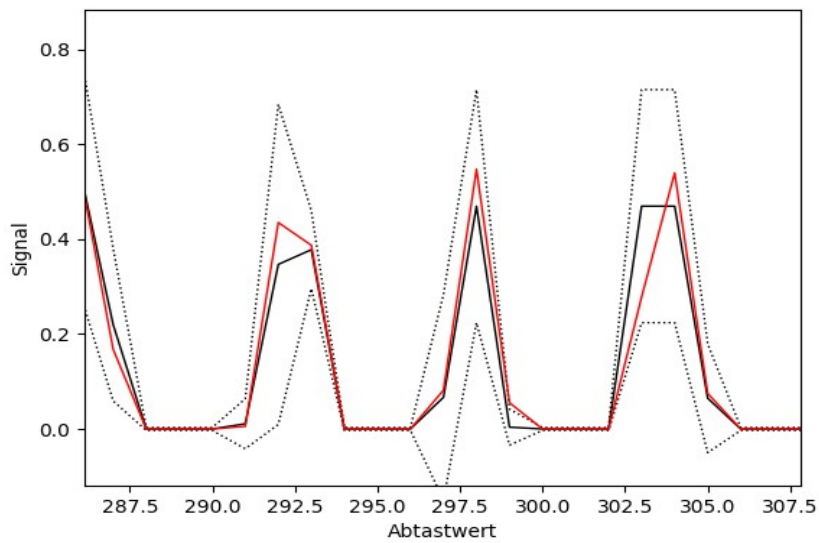


Abbildung 8: Prädizierter Signalverlauf für ein Werte-kontinuierliches Signal des Tanksystems

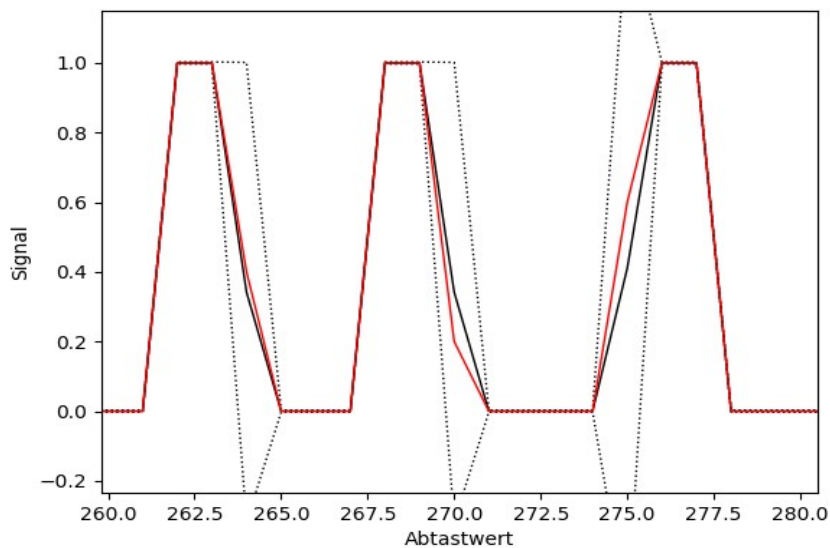


Abbildung 9: Prädizierter Signalverlauf für ein binäres Signal des Tanksystems

Long Short-Term Memory (LSTM) Netze

Das im Rahmen des Projektes implementierte LSTM basiert auf der Veröffentlichung [EN2019]³, in welcher LSTM-Netzen zur Prognose von Energieverbräuche verwendet und auf dieser Basis einer Modell-basierte Anomalie-Erkennung durchgeführt wird. Der allgemein Einsatzbereich von LSTMs sind dynamische Systeme mit einem inneren Zustand. Die beobachtbaren

3 [EN2019] B. Eiteneuer, N. Hranisavljevic, and O. Niggemann, "Dimensionality Reduction and Anomaly Detection for CPPS Data using Autoencoder," in 2019 IEEE International Conference on Industrial Technology (ICIT), 2019, pp. 1286–1292.

Ausgangsgrößen des Systems sind nicht nur von den Eingängen abhängig, sondern auch von der Historie des Systems, dem Zustand. Klassische Anomalie-Erkennung auf dynamischen Systemen erfordert den Entwurf eines Prozessmodells und somit eine exakte mathematische Formulierung der Vorgänge des Prozesses.

Ähnlich wie das Hidden Markov Modell, soll auch dieser Ansatz ein stochastisches Prozessmodell aus den aufgezeichneten Daten anlernen. Im Gegensatz zu ersterem soll das Modell jedoch nicht nur die Emissionen in Abhängigkeit des Zustands abbilden, sondern auch die Abhängigkeit von den Eingangsgrößen.

Bei LSTMs handelt es sich um eine Erweiterung sogenannter Rekurrenter Neuronaler Netze. Rekurrente Neuronale Netze haben im Gegensatz zu einfachen Neuronalen Netzen einen inneren Zustand, sodass sie temporale Zusammenhänge lernen können. Das Netz besteht aus mehreren LSTM-Schichten gefolgt von voll-vernetzten Schichten. Die letzte Schicht bildet die Ausgangs-schicht. Es werden sowohl der Erwartungswert der Ausgangsgrößen als auch die Standardabweichung prädiziert.

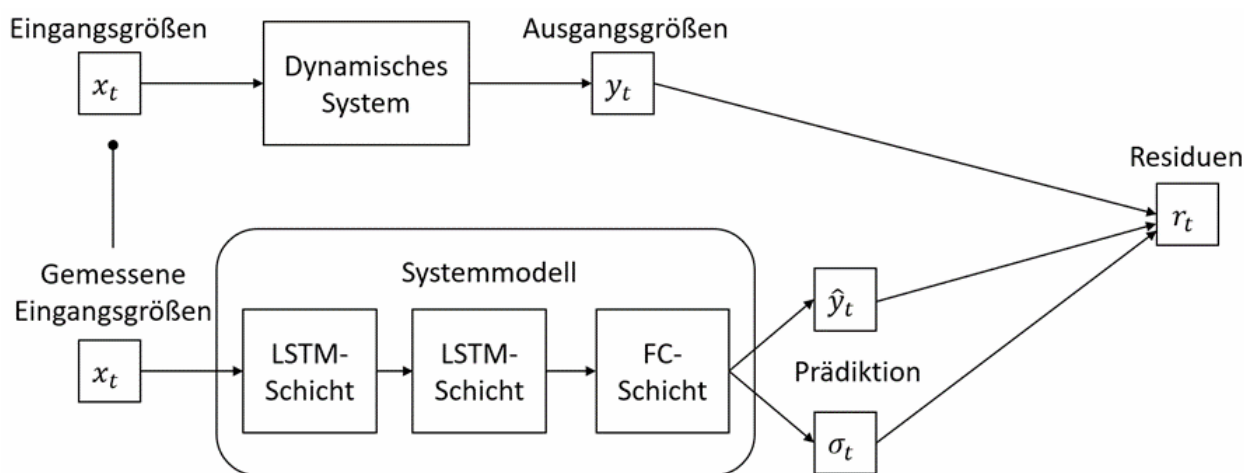


Abbildung 10: Funktionsprinzip des LSTM-basierten Ansatzes

Zur Simulation des Systemverhaltens werden die Eingangsgrößen des Prozesses gemessen und in das Systemmodell eingespeist. Die Anomalie-Erkennung basiert auf den Residuen zwischen den tatsächlichen Ausgangsgrößen des Prozesses und den mit dem Systemmodell prädizierten Ausgangsgrößen (Abbildung 14). In der Veröffentlichung [EN2019] werden Anomalien alleine durch das Überschreiten des Schwellenwertes ermittelt. Hierdurch werden jedoch nur Punktanomalien erkannt. Um konsekutive Abweichungen zu erkennen, wird in diesem Projekt stattdessen ein Filter eingesetzt. Eine Anomalie wird erkannt, wenn multiple aufeinander folgende Überschreitungen des Schwellenwertes festgestellt werden.

Das Training des Neuronalen Netzes erfolgt mit dem Backpropagation-Algorithmus sowie dem stochastischem Optimierungsverfahren Adam [KIN14]. Zusätzlich wird eine Hyperparameter-optimierung durchgeführt, sowohl des Neuronalen Netzes mittels einer Zufallssuche als auch des Schwellenwertes durch eine Rastersuche. Da für jede Parameterkonfiguration ein Netz trainiert werden muss, erfolgt die Optimierung verteilt auf mehrere Prozessoren. Das Ziel der Hyperparameteroptimierung ist die Maximierung der F1-Metrik.

Ergebnisse für die Elektro-Schlacke-Umschmelzanlage des Unternehmens DEW

In der Elektro-Schlacke-Umschmelzanlage des Unternehmens DEW bildet der Druckverlauf die Ausgangsgröße. Da aus den verbliebenen Größen keine nützliche Information zur Rekonstruktion des Druckverlaufs abgeleitet werden konnten, wurde nur die Zeit als Eingangsgröße des LSTMs verwendet. Im Gegensatz zu der Vorverarbeitung für das HMM ist es für das LSTM nicht notwendig, Zeitschritte mit fehlenden Datenpunkten zu entfernen. Stattdessen werden diese maskiert, wodurch sie keinen Einfluss auf das Training des LSTMs haben. Für das Training wurden 40 Sequenzen ohne Anomalie verwendet, sowie zehn zusätzlichen Sequenzen zur Validierung. Weiterhin wurden 40 Sequenzen mit Anomalien und die für das Training verwendete Sequenzen zur Optimierung des Schwellenwertes verwendet. Der Test erfolgte auf 61 normalen und 61 anormalen Sequenzen. Hierbei ergab sich ein F1-Score von 1.0, eine TP-Rate von 100.0 % und eine FP-Rate von 0.0 %. In Abbildung 14 ist zu sehen, dass die normalen Sequenzen selten starke Abweichungen aufweisen, die außerhalb des als normal gelernten Intervalls liegen. Diese werden durch einen Filter erfasst. Das anormale Signal liegt größtenteils unterhalb der unteren Detektionsschwelle für die Anomalie-Erkennung.

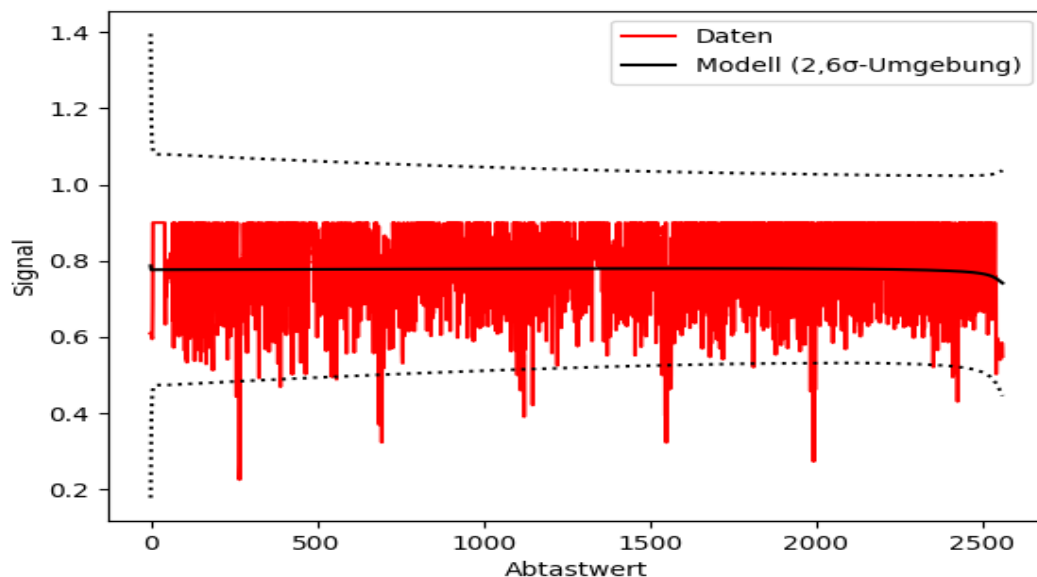


Abbildung 11: LSTM-basierte Prädiktion des Druck-Signals für eine Sequenz ohne Anomalien

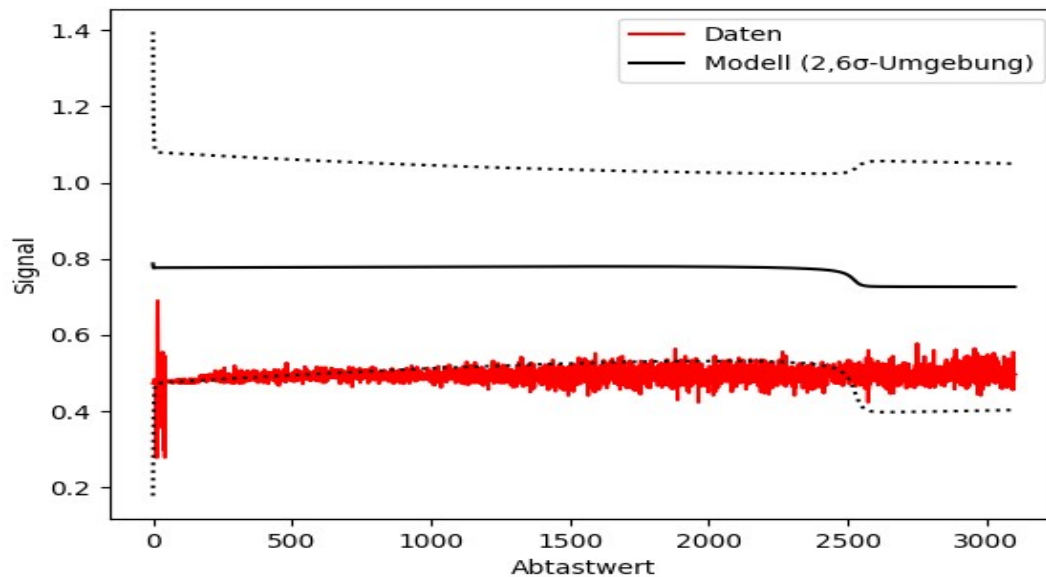


Abbildung 12: LSTM-basierte Prädiktion des Druck-Signals für eine Sequenz mit Anomalien

Ergebnisse für das 4-Tank-System

Das LSTM-Verfahren wurde außerdem auf dem Tanksystem-Datensatz getestet. Hierbei wurden dieselben Signale wie durch das HMM als Ausgangssignale verarbeitet. Als Eingangssignale verwendet das LSTM-Verfahren die vier Ventilstellungen `valveLinear1.opening` bis `valveLinear4.opening`. Auch in diesem Test wurden die Differenzenbildung (für Signal 1 und 2), Binarisierung (für Signal 3 und 4) und Normalisierung angewandt. Das Training erfolgte auf 15 Trainingssequenzen mit fünf Validierungssequenzen. Der Test erfolgte auf 80 normalen und 80 anormalen Sequenzen. Dabei konnte ein F1-Score von 0,88 erzielt werden, mit einer TP-Rate von 80 % und einer FP-Rate von 1,4 %, welcher oberhalb des F1-Scores für das HMM liegt (vgl. den vorangehenden Abschnitt zum HMM). In Abbildung 15 ist zu sehen, dass das LSTM in der Lage ist, sowohl binäre als auch Werte-kontinuierliche Signale zu lernen (sowohl mit als auch ohne Binarisierung liegen die tatsächlichen Signalverläufe überwiegend in der 2σ – Umgebung der prädizierten Signalverläufe).

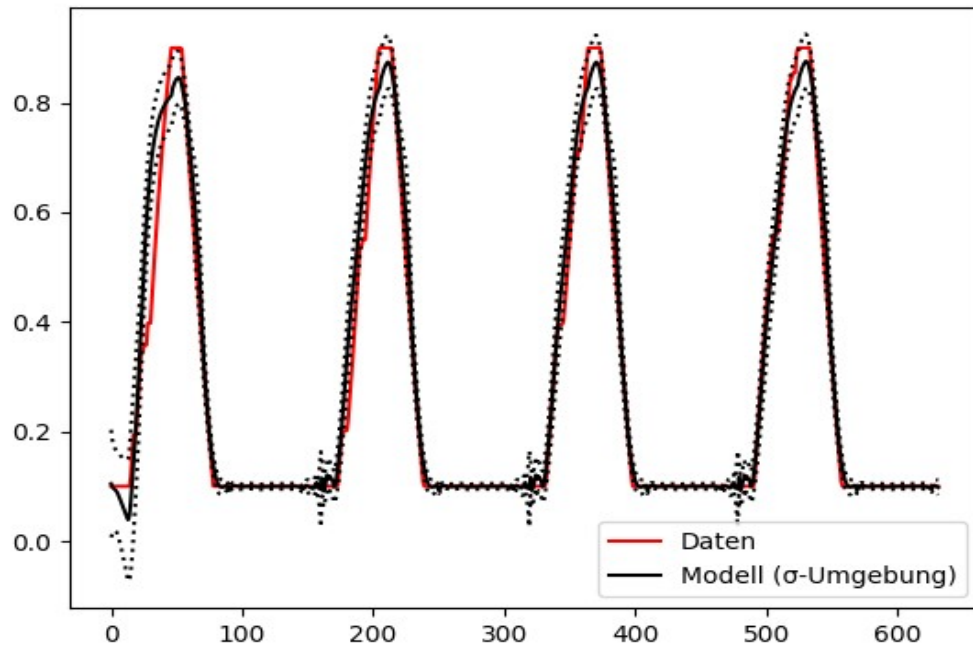


Abbildung 13: Prädizierter Signalverlauf für ein Werte-kontinuierliches Signal des Tanksystems

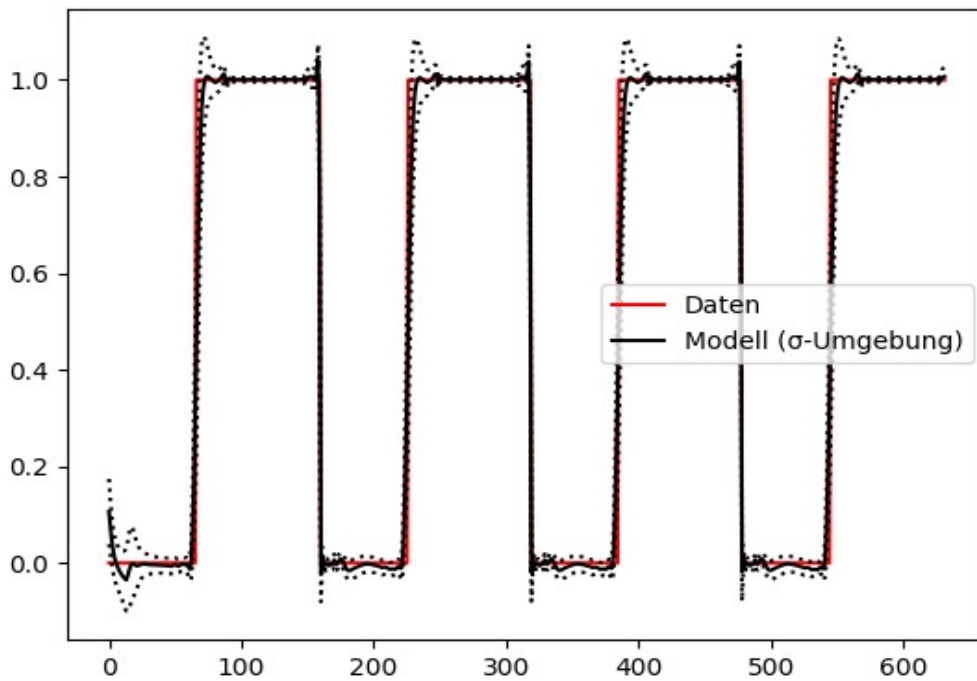


Abbildung 14: Prädizierter Signalverlauf für ein binäres Signal des Tanksystems

Automaten

Als Baseline für die Anomalie-Erkennung in diskreten Ereignissystemen wurden bereits am Fraunhofer IOSB-INA existierende Automaten-Lernverfahren in Python implementiert [Maier2014].

Das Grundprinzip dieses Ansatzes wird im Folgenden anhand des in Abbildung 15 dargestellten elementaren Fördersystems mit den beiden Lichtschranken L1 und L2 erläutert. Eine detaillierte Beschreibung des Verfahrens findet sich in [Maier2014].

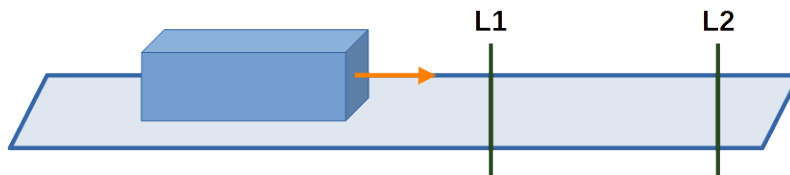


Abbildung 15: Elementares Fördersystem

Die Automaten, die mittels des OTALA Algorithmus aus historischen Sensordaten gelernt werden, bestehen, wie in Abbildung 16 dargestellt, aus Zuständen und Transitionen.

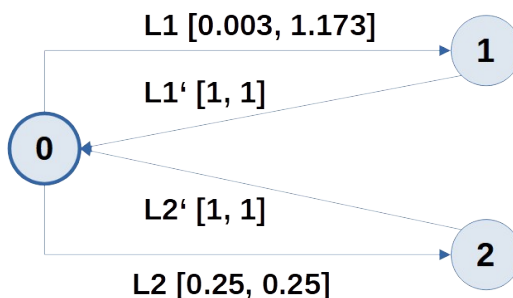


Abbildung 16: Automaten-Darstellung des elementaren Fördersystems

Im Beispiel gibt der Zustand an, welche Lichtschranke gerade aktiv ist (0: keine Lichtschranke aktiv, 1: Lichtschranke L1, 2: Lichtschranke L2). Die Zustandsübergänge werden durch Ereignisse ausgelöst, welche innerhalb von Zeitschranken auftreten, die sich aus den historischen Sensordaten ergeben (im Beispiel wird erwartet, dass im Zustand 0 innerhalb des Zeitfensters [0.003, 1.173] das Ereignis L1 auftritt). Die Anomalie-Erkennung basiert auf der Überprüfung, inwieweit die richtigen Ereignisse innerhalb der gelernten Zeitfenster aufgetreten sind. Die Anomalie-Erkennung auf der Grundlage von Automaten wurde auf einem ersten Testdatensatz des Unternehmens Venjacob erfolgreich evaluiert (erreichter F1-Score: 100%).

Petri-Netze

Bei komplexeren Systemen und insbesondere dann, wenn Parallelität auftritt, stößt die Verwendung von Automaten zur Anomalie-Erkennung allerdings an ihre Grenzen. Aus diesem Grund wurde in dem Projekt einer neuer Ansatz zur Daten-getriebenen Anomalie-Erkennung auf Basis von Petri-Netzen entwickelt (eine Veröffentlichung dieses Ansatzes ist in einem Artikel mit dem Arbeitstitel „Learning Petri Net Models from Sensor Data of Conveying Systems based on the Merging of Prefix and Postfix Trees“ geplant). In dem Ansatz wird zunächst ein Petri-Netz aus historischen Daten gelernt.

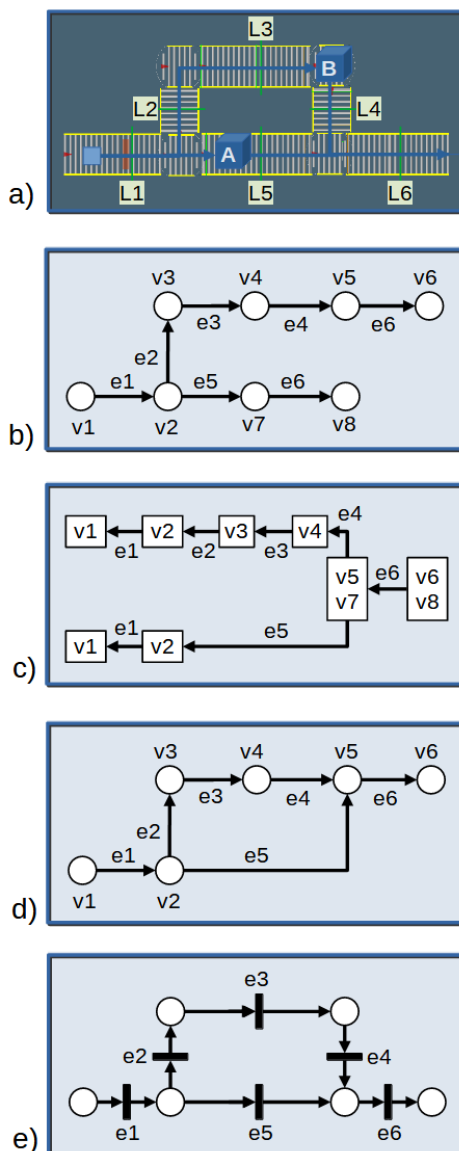


Abbildung 17: Schritte zum Lernen des Petri-Netzes aus Ereignissequenzen

Dazu werden die folgenden Schritte durchgeführt, welche in Abbildung 17 exemplarisch für ein kleines Fördersystem (siehe Abbildung 17 a) dargestellt sind:

- Lernen der Struktur des diskreten Ereignissystems aus fehlerfreien Simulationdurchläufen, in denen sich jeweils nur ein Behälter gleichzeitig durch das Transportsystem bewegt (hierzu wird für jeden Behälter die Ereignissequenz mitgeschnitten und in Form eines Prefix-Trees festgehalten, siehe Abbildung 17 b); unter zusätzlichen Modell-Annahmen (z.B. der Annahme, dass jedes Ereignis in einem Prozess - hier: dem Transportvorgang eines einzelnen Behälters - nur einmal auftritt, ist es alternativ möglich, den Prefix-Tree aus Simulationdurchläufen mit mehreren parallelen Prozessen abzuleiten
- Ableiten eines Postfix-Trees, in dem Ereignissequenzen mit gleichen End-Sequenzen zusammen abgespeichert werden (Abbildung 17 c)

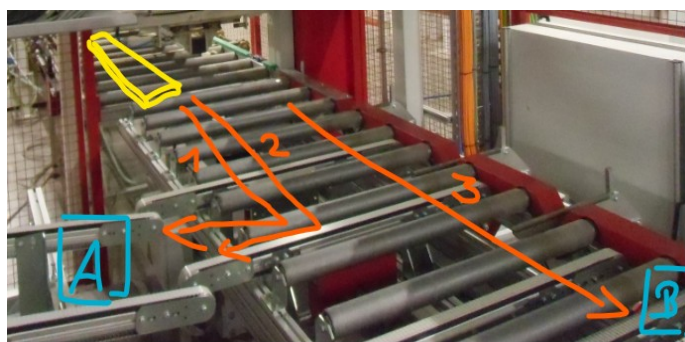
- Sukzessives Mergen von Pfaden des Präfix-Trees unter Berücksichtigung des Postfix-Trees, Herstellung der Determiniertheit des resultierenden Graphen (Abbildung 17 d)
- Transformation des Graphen in ein Petri-Netz (Abbildung 17 e)
- Ermittlung zeitlicher Bedingungen an den Transitionen des Petri-Netzes aus fehlerfreien Simulationsdaten, in denen sich mehrere Behälter gleichzeitig durch das Transportsystem bewegen

Eine einfache Parallelisierung des Verfahrens ist möglich, indem jeweils für einen einzelnen Simulationsdurchlauf ein Präfix-Tree entsprechend Abbildung 17 b) erstellt wird. Die Präfix-Trees für die einzelnen Simulationsdurchläufe können anschließend elementar zu einem globalen Präfix-Tree verschmolzen werden. Hierzu wird einer der lokalen Präfix-Trees als Startpunkt für den globalen Präfix-Tree verwendet. Anschließend werden die verbleibenden lokalen Präfix-Trees durchlaufen und die mit den durchlaufenen Pfaden korrespondierenden Ereignissequenzen sukzessive in den globalen Präfix-Tree eingefügt.

Zur Anomalie-Erkennung werden das Systemverhalten mit dem gelernten Petri-Netz simuliert und auf diese Weise unerwartete Ereignisse oder Verletzungen von Zeitschranken detektiert. Im Folgenden sind die Ergebnisse der Petri-Netz-basierten Anomalie-Erkennung in verschiedenen Use Cases dargestellt.

Exemplarische Use Cases des Unternehmens Becker

Das Unternehmen Becker hat zur Evaluierung des Condition Monitoring Systems die Sensordaten eines Fördersystems mit Verzweigungen bereitgestellt. Aufgrund fehlerhafter Trainingsdaten enthalten die für diesen Anwendungsfall generiert Präfix-Trees teilweise Pfade, die das Normalverhalten der Anlage nicht korrekt abbilden. Beispielsweise erstrecken sich einige Pfade in dem Präfix-Tree, der für das in Abbildung 18 dargestellte Modul 3 des Fördersystems erstellt wurde, über zwei Prozess-Sequenzen, da aufgrund von Sensorfehlern einzelne Startsignale von Prozess-Sequenzen in den Trainingsdaten fehlen.



Diskrete Ereignisse
1: LT_Start_Pos
2: Winkeluebergabe_belegt
3: P3_LT_Stopp
4: Puffer_belegt
5: ZielA_belegt
6: LT_Auslauf
7: ZielB_belegt

Abbildung 18: Modul 3 des Fördersystems

Die fehlerhaften Pfade sind in Abbildung 19 rot dargestellt (die Namen der in dem Modul auftretenden diskreten Ereignisse können Abbildung 18 entnommen werden). Um Prozessfehler in den Trainingsdaten zu kompensieren, wird ein Pruning des Präfix-Trees durchgeführt, wobei selten durchlaufende Pfade abgeschnitten werden. Der Präfix-Tree nach dem Pruning sowie der mit dem in Abbildung 17 skizzierten Verfahren generierte Graph sind in Abbildung 20 dargestellt.

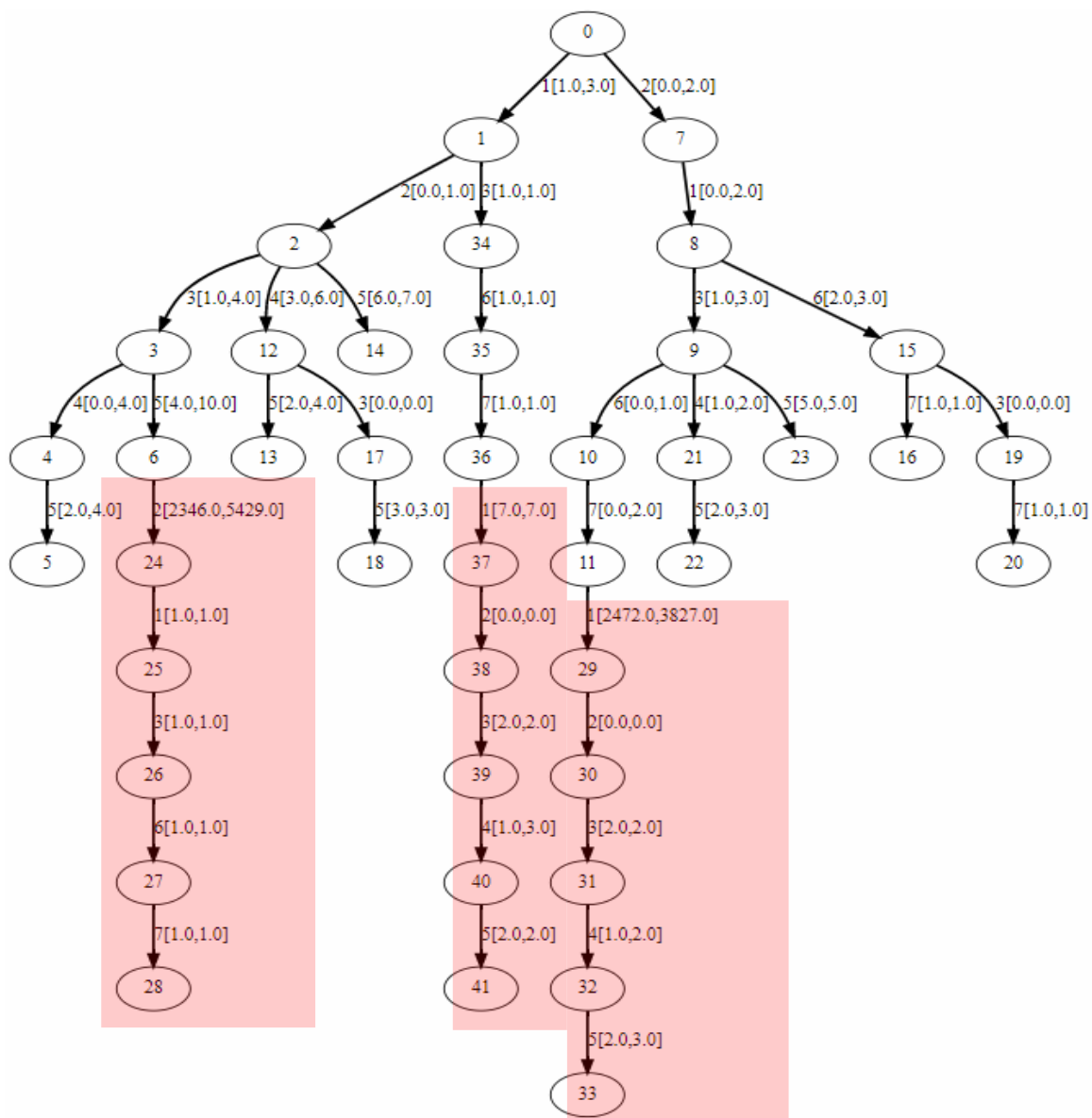


Abbildung 19: Präfix-Tree für den Use Case des Unternehmens Becker

Zur Evaluierung der Anomalie-Erkennung wurden 50 fehlerfreie Prozesssequenzen sowie Testdatensätze mit 10 Fehlerfällen, welche durch Signalverzögerungen gekennzeichnet sind, verwendet. Ein typischer Fehlerfall ist in Abbildung 21 dargestellt. Die 10 Fehlerfälle wurden von dem Condition Monitoring System korrekt als solche erkannt. Weiterhin wurden keine Fehler auf den fehlerfreien Testdaten erkannt. Während des Normalbetriebs der Anlage ist allerdings zu berücksichtigen, dass häufig Verzögerungen und fehlende Signale (z.B. ein fehlendes Startsignal) auftreten. Solche Anomalien, die in den untersuchten 50 fehlerfreien Prozesssequenzen nicht vorhanden sind, führen potentiell zu vielen Fehlermeldungen des Condition Monitoring Systems.

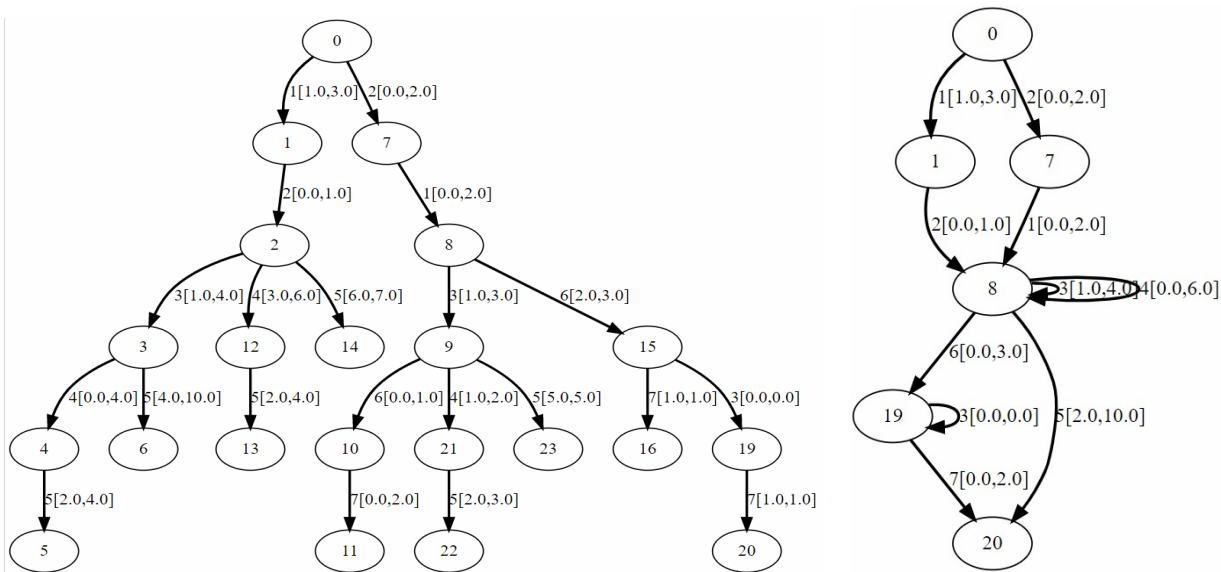


Abbildung 20: Präfix-Tree nach dem Pruning und gelernter Graph

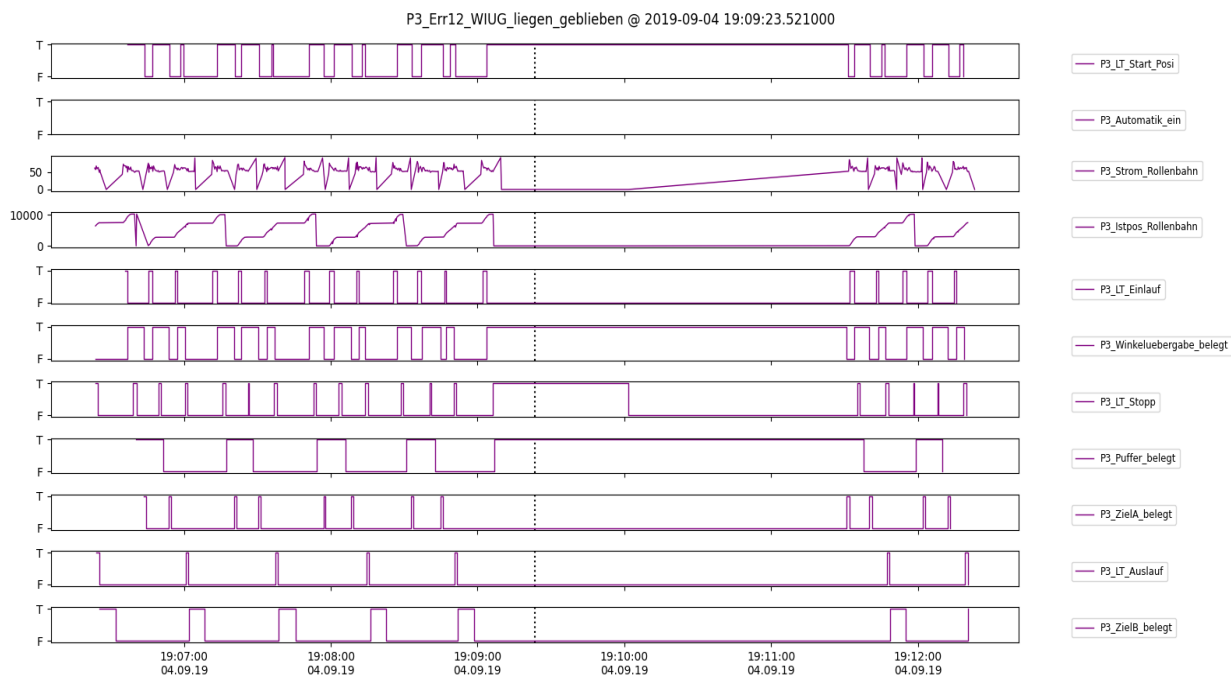


Abbildung 21: Typischer Fehlerfall im Use Case des Unternehmens Becker

Exemplarische Use Cases des Unternehmens Venjacob

Zur Evaluierung der Petri-Netz-basierten Anomalie-Erkennung hat das Unternehmen Venjacob CSV-Dateien zur Verfügung gestellt, welche mit dem in Abbildung 8 dargestellten Simulationsmodell generiert wurden. In dem Anwendungsfall ist zu berücksichtigen, dass bereits beim Lernen der Struktur des Petri-Netzes mehrere Werkstücke gleichzeitig transportiert werden. Aus diesem Grund sind Heuristiken erforderlich, um die Sensorsignale eindeutig den

einzelnen Transportprozessen zuordnen zu können. Konkret wird in dem implementierten Verfahren die Annahme getroffen, dass jedes Event in einem Transportvorgang nur einmal vorkommt, ein doppeltes Event also jeweils dem nächsten Transportvorgang zugeordnet werden kann. Die Struktur eines für diesen Anwendungsfall gelernten Petri-Netzes ist in Abbildung in Form des gerichteten Graphen, aus dem das Petri-Netz generiert wird, dargestellt. Es ist zu erkennen, dass die gelernten Petri-Netze aufgrund der geringen Komplexität des Anwendungsfalls eine vergleichsweise einfache Struktur aufweisen. Die parallelen Pfade kommen durch die unterschiedlichen Längen der Werkstücke, welche zu unterschiedlichen Aktivierungsreihenfolgen von Sensoren führen, zustande. Um die Skalierbarkeit des Ansatzes zu evaluieren, wurde das Lernverfahren für Petri-Netze, wie im nächsten Abschnitt dargestellt, zusätzlich auf ein komplexeres Transportsystem angewendet.

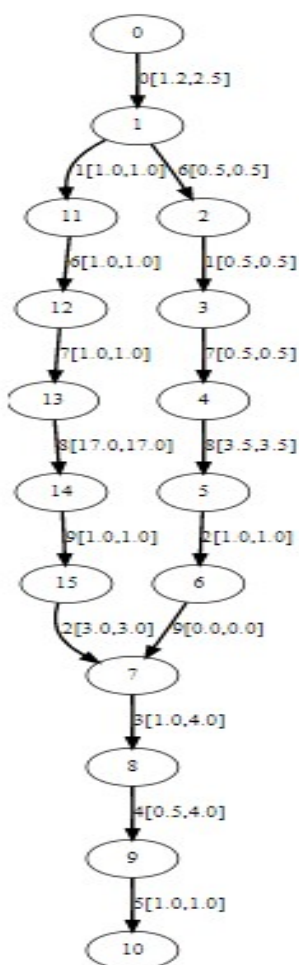


Abbildung 22: Struktur des Petri-Netzes für den Use Case des Unternehmens Venjacob

Komplexeres Transportsystem

Für das Beispiel aus Abbildung 12 ergibt sich durch die Anwendung des entwickelten Lernverfahrens das in Abbildung 23 dargestellte Petri-Netz. In dem Beispiel wurden für jedes Förderband zwei Transitionen gelernt, die den Lichtschranken am Anfang und Ende des Förderbandes entsprechen.

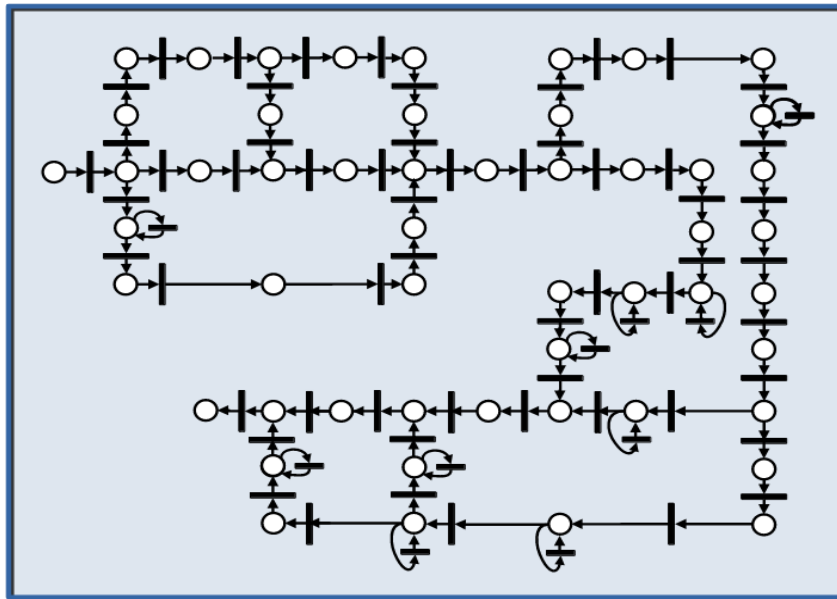


Abbildung 23: Struktur des gelernten Petri Netz für das Transportsystem aus Abbildung 12

Um die Anomalie-Erkennung mit Petri-Netzen systematisch zu evaluieren, wurde ein Testdatensatz für die ersten 12 Förderbänder des in Abbildung 12 dargestellten Transportsystems verwendet. Zum Lernen der Struktur des Petri-Netzes wurden 20 fehlerfreie Simulationsdurchläufe herangezogen, in denen sich jeweils nur ein Behälter gleichzeitig durch das Transportsystem bewegt. Um die zeitlichen Bedingungen an den Transitionen des Petri-Netzes zu ermitteln, wurde auf 20 weitere Simulationsdurchläufe mit jeweils 20 Behältern zurückgegriffen. Die Evaluierung wurde auf 100 fehlerfreien Simulationsdurchläufen und 75 fehlerbehafteten Simulations-durchläufen mit jeweils 20 Behältern durchgeführt. Dabei wurden drei Fehlertypen mit gleicher Häufigkeit berücksichtigt (Sensorausfälle, Schlupf und falsche Länge der Behälter) Insgesamt konnte ein F1-Score von 0,85 erzielt werden.

AP 3.2 Fehlerursachen-Erkennung

Ziele

Ziel der Fehlerursachen-Erkennung war es, den erkannten Anomalien mögliche Ursachen zuzuordnen. Die Fehlerursachen-Erkennung soll einem unerwarteten Event, welches als Anomalie detektiert wird, abhängig vom Systemzustand bei Auftreten des Symptoms beispielsweise die mögliche Ursache „Sensorausfall“ zu. Als mögliche Ursachen für ein verzögertes Ereignis könnten „Schlupf“ oder fehlerhafte Abmaße der transportierten Gegenstände vorliegen.

Durchgeführte Arbeiten

In dem Projekt wurde zur Fehlerursachen-Erkennung ein Ansatz auf der Grundlage von Case-Based Reasoning entwickelt. Im ersten Schritt werden die Fehlersymptome (falsche Ereignisse oder verletzte Zeitschranken) ermittelt und in einer Datenbank abgespeichert. Darauf aufbauend wurde ein Verfahren entwickelt, welches den Symptomen auf Basis von Case-based Reasoning Fehlerbeschreibungen zugeordnet werden. Die Evaluierung der Fehlerursachen-Erkennung wurde in den im Projekt betrachteten Transportsystemen durchgeführt.

Erzielte Ergebnisse

Der entwickelte Ansatz basiert auf einem Repository, in dem zu den in der Vergangenheit aufgetretenen Anomalien jeweils eine Beschreibung des Symptoms und der Fehlerursache abgelegt sind. In diskreten Ereignissystemen wird das Symptom durch folgende Informationen charakterisiert:

- Systemzustand beim Auftreten der Anomalie (in Petri-Netzen die aktuelle Markierung zu diesem Zeitpunkt)
- Ereignis, welches die Anomalie hervorgerufen hat
- Bei der Verletzung von Zeitschranken zusätzlich die erwarteten Zeitschranken für das Ereignis und der Zeitpunkt, zu dem das Ereignis tatsächlich eingetreten ist.

Als Fehlerursache wird jeweils eine Kategorie eingetragen (bei Transportsystemen kommen z.B. die Kategorien „Sensorausfall“, „Schlupf“, „Falsche Länge“ in Betracht).

Der Ablauf der Fehlerursachen-Erkennung ist in Abbildung 24 dargestellt.

Als Ergebnis der Anomalie-Erkennung ergeben sich Symptome, deren Ursachen zunächst unbekannt sind. Mögliche Ursachen werden in einem Retrieval-Schritt ermittelt, in dem ein jeweils aufgetretenes Symptom mit den im Repository gespeicherten Symptomen verglichen wird. Anschließend werden die möglichen Ursachen entsprechend der Ähnlichkeit der Symptome sortiert und dem Benutzer präsentiert.

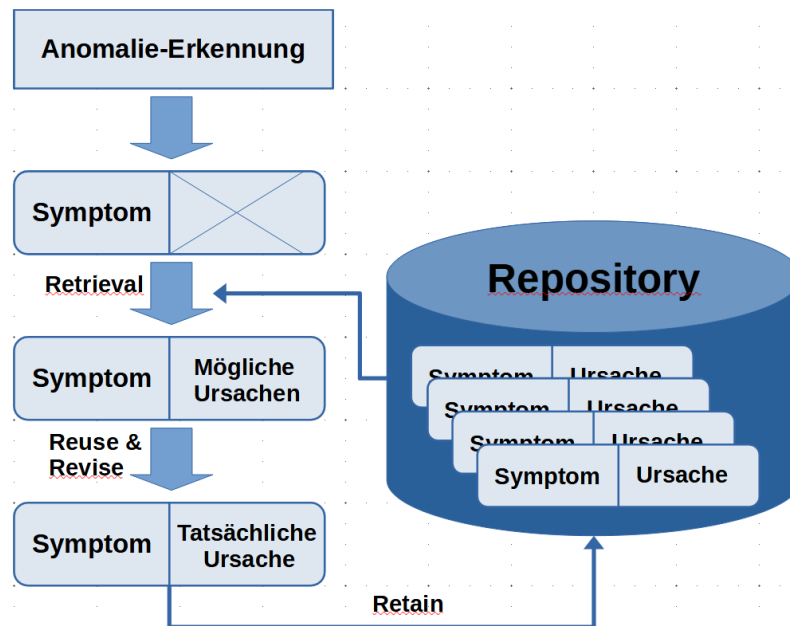


Abbildung 24: Case-Based Reasoning - Grundprinzip

Grundlage für diesen Ansatz ist ein geeignetes Ähnlichkeitsmaß, welches von der Beschaffenheit der Symptome im Condition Monitoring System abhängt. Wird die Anomalie-Erkennung auf der Grundlage von Petri-Netzen durchgeführt, kann das Ähnlichkeitsmaß als Produkt der drei folgenden Faktoren berechnet werden, wobei der dritte Faktor nur berücksichtigt wird, wenn durch die Anomalie Zeitschranken verletzt werden:

- **Ähnlichkeit der Systemzustände (Markierungen):** Ausgangspunkt der Ähnlichkeitsberechnung ist eine vereinfachte Darstellung der Markierung in Form von Mengen, welche angeben, welche Stellen eines Petri-Netzes zum Zeitpunkt der Anomalie belegt sind. Die Markierung $M = \{1, 5, 8\}$ gibt beispielsweise an, dass sich Tokens an den Stellen 1, 5 und 8 des Petri-Netzes befinden. Als Ähnlichkeitsmaß zwischen zwei Markierungen $M1$ und $M2$ wird $\max\{\text{eps}, J(M1, M2)\}$ verwendet, wobei $J(M1, M2) = (M1 \cap M2) / (M1 \cup M2)$ den Jaccard-Koeffizienten der Mengen $M1$ und $M2$ bezeichnet, welcher durch die Konstante $\text{eps} \ll 1$ nach unten begrenzt wird, um zu verhindern dass der erste Faktor des Ähnlichkeitsmaßes 0 wird.
- **Ähnlichkeit der Ereignisse:** Der zweite Faktor des Ähnlichkeitsmaßes wird auf den Wert 1 gesetzt, wenn die Symptome auf ein identisches Ereignis zurückzuführen sind, ansonsten auf den Wert $\text{eps} \ll 1$.
- **Art der Zeitüberschreitung:** Für zeitliche Anomalien wird weiterhin die Art der Zeitüberschreitung in dem Ähnlichkeitsmaß berücksichtigt. Falls in beiden Symptomen eine obere Zeitschranke überschritten oder eine untere Zeitschranke unterschritten wird, wird der dritte Faktor des Ähnlichkeitsmaßes auf den Wert 1 gesetzt. Ist die Art der Zeitüberschreitung bei den beiden Symptomen unterschiedlich, wird für den dritten Faktor des Ähnlichkeitsmaßes ein kleiner Wert $\text{eps} \ll 1$ angenommen.

Auf die beschriebene Weise wird für jeden im Repository existenten Fall ein Ähnlichkeitswert berechnet. Anschließend werden die Ähnlichkeitswerte für eine Fehlerursache jeweils

aufaddiert und die Fehlerursachen mit den höchsten kumulierten Ähnlichkeitswerten dem Benutzer als mögliche Fehlerursachen angezeigt.

Der Benutzer kann mit dieser Zusatzinformation den Fehler beheben und die tatsächliche Fehlerursache eingeben bzw. eine der vorgeschlagenen Fehlerursachen bestätigen (Reuse & Revise). Abschließend wird die vom Benutzer angegebene Fehlerursache zusammen mit dem aufgetretenen Symptom im Repository gespeichert. Auf diese Weise wird die Fall-Datenbank mit jedem aufgetretenen Symptom erweitert.

Die Ergebnisse der Fehlererkennung sind in Abbildung 25 und Abbildung 26 dargestellt. Abbildung 25 zeigt, dass die drei Fehlerarten „Sensor failure“, „Slipped container“ und „Incorrect dimensioned container“ in 93%, 81%, and 99% der Fälle korrekt klassifiziert werden.

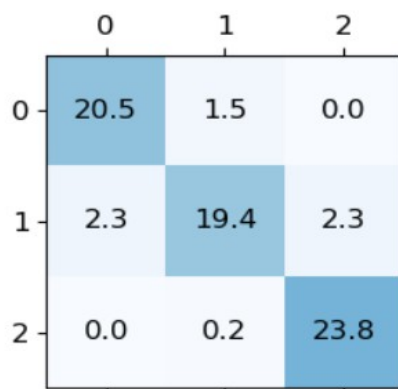


Abbildung 25: Confusion Matrix für die Fehlerarten "Sensor failure" (Typ 0), "Slipped container" (Typ 1) und "Incorrect dimensioned container" (Typ 2)

Aus Abbildung 26 geht hervor, dass der Anteil der richtig klassifizierten Fehlerfälle mit zunehmender Größe des Repositories zunimmt.

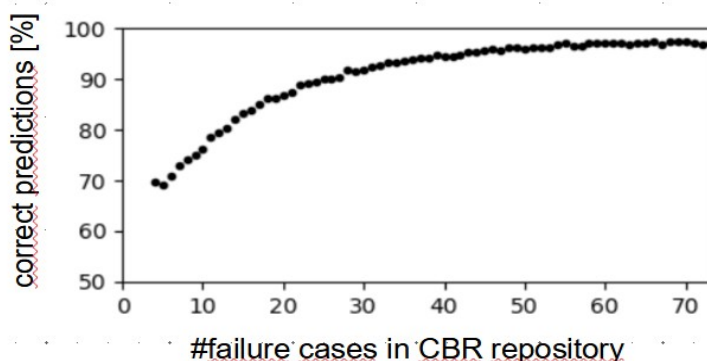


Abbildung 26: Zunahme der richtig klassifizierten Fehler mit der Anzahl der Fehlerfälle im CBR Repository

AP 3.3 Web-basierte Benutzerschnittstelle

Ziele

Zur Anzeige von Anomalien und Fehlerursachen sollte eine geeignete Benutzerschnittstelle entwickelt werden. Die Stellmöglichkeiten in der GUI sollten auf das wesentliche Minimum reduziert werden, so dass sie ohne aufwendige Konfiguration für verschiedene Anlagen eingesetzt werden kann. Beispielsweise sollte keine aufwendige Auswahl von Prozessdaten bei der Anomalie-Erkennung notwendig sein. Mögliche Anomalien sollten z.B. in Form einer Ampel angezeigt werden, wobei zwischen normalem Prozessverhalten sowie leichten und schwerwiegenden Anomalien unterschieden werden kann. Darüber hinaus sollten aufgrund identifizierter Fehlerursachen Handlungsanweisungen für den Operator generiert werden.

Durchgeführte Arbeiten

In diesem Arbeitspaket wurde eine Web-Applikation entwickelt, welche das Lernen der für das Condition Monitoring erforderlichen Prozessmodelle sowie eine Modell-basierte Anomalie-Erkennung ermöglicht.

Erzielte Ergebnisse

Die entwickelte Web-Applikation, welche unter dem im Kapitel „Projektergebnisse“ angegebenen Link verfügbar ist, besteht aus mehreren Web-Pages, die im folgenden beschrieben werden. In Abbildung 27 ist die Web-Page zur Konfiguration der Modell-Erstellung dargestellt.

The screenshot shows a web page titled "Modell-Erstellung" with a green header. The main content area has a blue background. It contains the following elements:

- Name des zu erstellenden Modells:** A text input field containing "ModelA".
- Konfiguration der Datenquelle:** A table with the following rows:

Typ:	InfluxDB
Host:	localhost
Port:	8086
User:	admin
DB:	clara_train
Tabelle:	sensordata
- Zeitraum für die Modell-Erstellung:** A table with the following rows:

Beginn:	2021-01-01 00:00:00
Ende:	2021-05-19 09:15:00
- Modell-Erstellung starten:** A button.
- Footer:** A link: "Über den folgenden Link können Sie zum Condition Monitoring wechseln, ohne ein neues Modell zu erstellen: [Condition Monitoring](#)".

Abbildung 27: Konfiguration der Modell-Erstellung

Auf der Web-Page können der Name des Modells, eine Datenquelle sowie der Trainingszeitraum konfiguriert werden. Weitere Parameter wie der Modelltyp (PETRINET, HMM, LSTM, etc.) oder die Anzahl der Sensordaten, die gleichzeitig von der Datenquelle übertragen

werden, sind in einer Datei konfigurierbar. Nach dem Start der Modell-Erstellung wird dem Benutzer der Status des Lernvorgangs angezeigt. Das erstellte Modell wird in einer SQL-Datenbank abgelegt. Beim Start des Condition Monitorings müssen die Datenquelle angegeben und ein zuvor erstelltes Modell ausgewählt werden (siehe Abbildung 28). Weitere Parameter können wiederum in der Datei **gui_config.py** spezifiziert werden. Hier kann insbesondere angegeben werden, inwieweit das Condition Monitoring online oder für Demonstrationen offline durchgeführt werden soll. Wenn das Condition Monitoring offline durchgeführt wird, ist es darüber hinaus möglich, einen Startzeitpunkt sowie eine zeitliche Verzögerung beim Condition Monitoring zu spezifizieren. Weiterhin ist die Anzahl der Sensordaten, die gleichzeitig von der Datenquelle übertragen werden, konfigurierbar. Darüber hinaus kann das Diagnosemodul in der Konfigurationsdatei aktiviert werden. Während des Condition Monitorings werden der Systemzustand in Form einer Ampel sowie weitere Informationen zu aufgetretenen Anomalien in einem Textfeld angezeigt (Abbildung 29). Wenn der Diagnose-Modus aktiviert ist, werden dem Benutzer darüber hinaus mögliche Ursachen für die Anomalien präsentiert, welche von dem Diagnosemodul ermittelt werden. In dem Textfeld „Tatsächliche Ursache“ wird zunächst autoamtisch die wahrscheinlichste Ursache eingetragen. Diese kann vom Benutzer bestätigt oder durch eine andere Ursache ersetzt werden. Anschließend kann die Anomalie-Erkennung durch Betätigen des Buttons „Anomalien behoben“ fortgesetzt werden.

Konfiguration der Datenquelle	
Typ:	InfluxDB
Host:	localhost
Port:	8086
User:	admin
DB:	clara_eval
Tabelle:	sensordata

Ausgewähltes Modell für die Anomalie-Erkennung:

ModelA

Um neue Modelle zu erstellen, können Sie den folgenden Link verwenden: [Modell-Erstellung](#)

Abbildung 28: Starten des Condition Monitorings

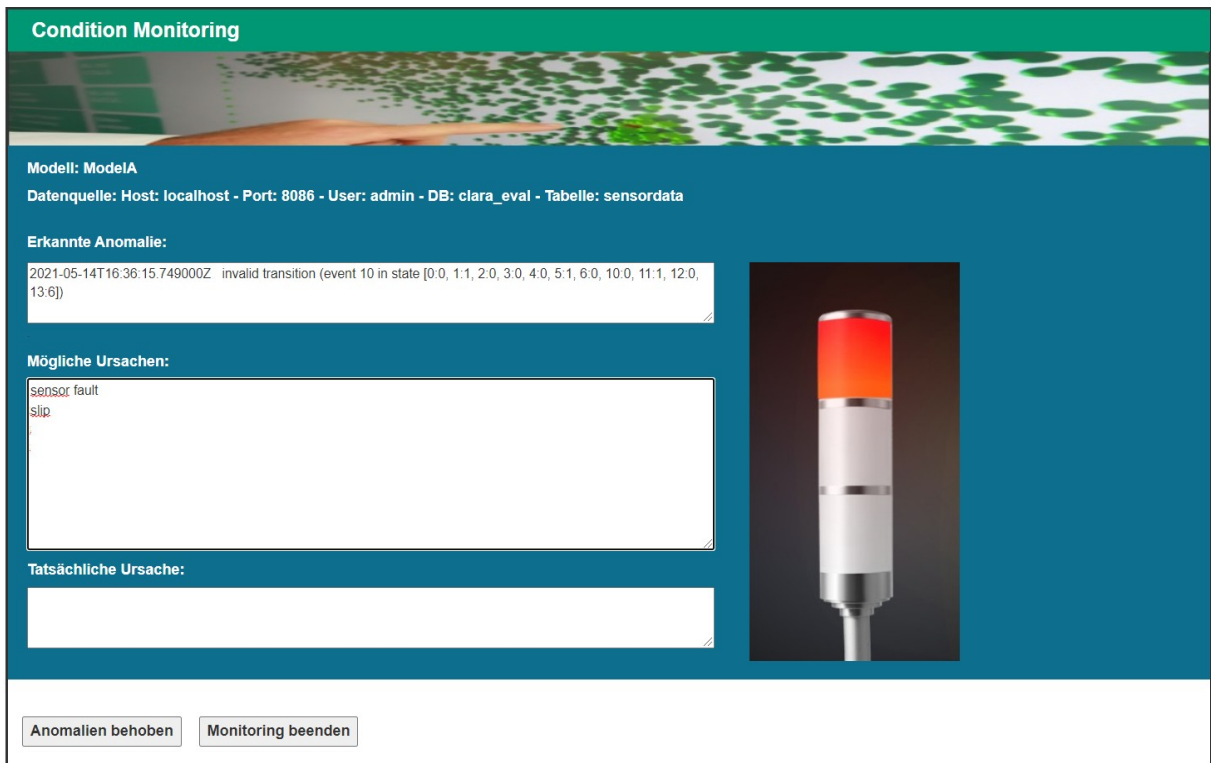


Abbildung 29: Condition Monitoring

Die Architektur der Web-Applikation ist in Abbildung 30 skizziert.

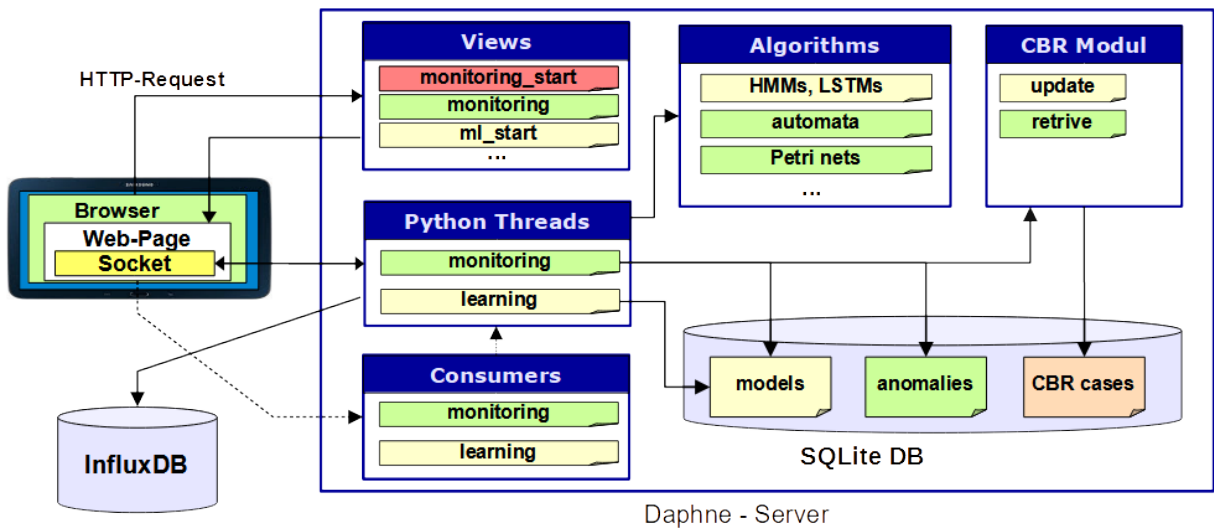


Abbildung 30: Architektur der Web-Applikation

Die Views dienen der Erstellung einzelner Web-Pages. Je nach aufgerufener URL wird ein anderes Python-Skript ausgeführt (*monitoring_start*, *monitoring*, *ml_start*, etc.), welches das Rendern der Web-Page entsprechend einem dafür erstellten Template anstößt. Die Web-Pages kommunizieren über WebSockets mit Python-Threads, in denen die Logik des Condition Monitoring Systems implementiert ist. Consumer dienen zur Erzeugung dieser Threads. Zum

Modell-Lernen wird ein Learning-Thread aufgerufen, der auf historische Daten in einer InfluxDB zugreift und unter Verwendung einer Algorithmen-Bibliothek ein Prozessmodell erstellt (z.B. die oben beschriebenen Automaten oder Petri-Netze oder die im vorangehenden Berichtszeitraum entwickelten HMMs oder LSTMs). Die gelernten Prozessmodelle werden in einer SQLite DB abgelegt und während des Condition Monitorings geladen. Beim Condition Monitoring auftretende Anomalien werden in einer zweiten SQLite DB gespeichert. Weiterhin existiert ein Diagnosemodul (CBR-Modul) welches im Fehlerfall die wahrscheinlichsten Fehlerursachen auf Basis eines Repositories ermittelt (retrieve). Die tatsächliche Fehlerursache für den jeweiligen Fehlerfall wird zur Aktualisierung des Repositories verwendet. Das Repository des Diagnosemoduls wurde als SQLite Datenbank realisiert (CBR cases).

AP 3.4 Cloud-basiertes Modell-Lernen

Ziele

In AP3.4 sollten rechenintensive Prozesse, d.h. insbesondere die Methoden zum Modell-Lernen, in die in AP2.2 entwickelte Cloud-basierte Infrastruktur ausgelagert werden. Dazu sollten die in AP 3.1 bereits parallelisierten Verfahren zum Modell-Lernen an die Schnittstellen der Cloud-Umgebung angepasst und in der Cloud-basierten Infrastruktur des Fraunhofer IOSB-INA evaluiert werden.

Durchgeführte Arbeiten

Im Rahmen von AP3.4 wurde ein Cloud-basiertes Infrastrukturkonzept entwickelt, welches insbesondere ein Cloud-basiertes Modell-Lernen ermöglicht. Das Konzept wurde zunächst lokal getestet. Anschließend wurde die Migration in die Cloud durchgeführt.

Erzielte Ergebnisse

Eine Übersicht über das Infrastrukturkonzept ist in Abbildung 31 dargestellt. Das Konzept stellt eine logische Trennung der drei Hauptkomponenten Datenerfassung, Datenhaltung und Anwendungslogik sicher. Die Komponenten befinden sich in separaten Docker-Containern, die über Microservices verbunden sind, um Isolation und Skalierbarkeit zu gewährleisten. Das Fraunhofer IOSB-INA stellt einen leistungsfähigen und rechenstarken Nvidia DGX-1-Cluster zur Verfügung, der genügend Ressourcen bietet, um hochintensive Berechnungen in parallelen Threads durchzuführen. Der Cluster ist im lokalen Netzwerk des Instituts gekapselt und nach Industriestandards abgesichert. Der Zugriff darauf ist über ein sicheres VPN möglich. Alternativ kann die Infrastruktur auch On-Edge implementiert werden, wie es einige Industriepartner entsprechend ihrer Infrastruktur- und Sicherheitsvorschriften wünschen.

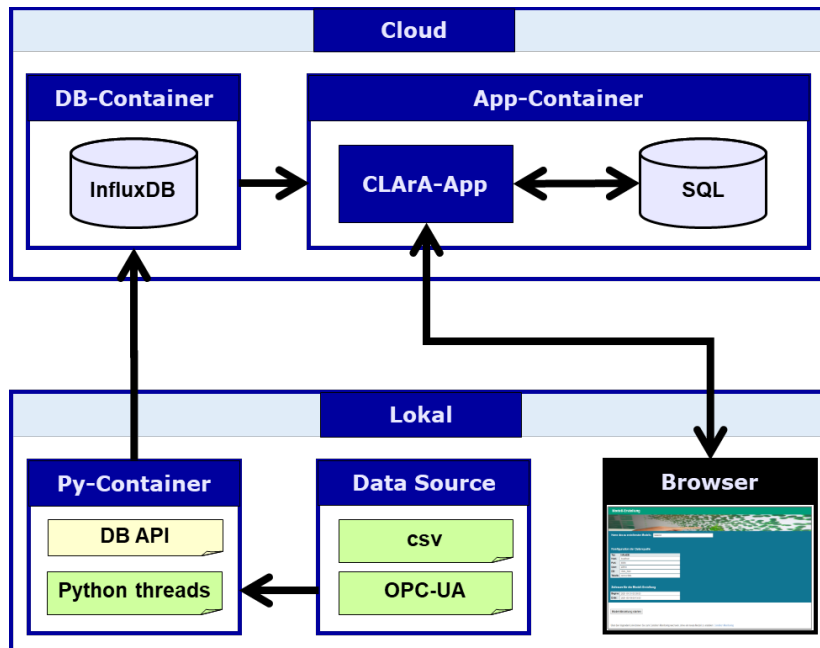


Abbildung 31: Cloud-basiertes Infrastruktur-Konzept für das parallelierte Modell-Lernen

Arbeitspaket 4: Evaluierung, Dokumentation und Projektmanagement

Arbeitspaket 4 umfasst Projekt-übergreifende Tätigkeiten, welche in den folgenden Abschnitten dargestellt sind.

Arbeitspaket 4.1: Evaluierung

Ziele

In AP4.1 sollte das CMS mit Testdaten der Unternehmen Becker, DEW und Venjakob systematisch evaluiert werden. Die Ergebnisse sollten in einem Testbericht festgehalten und im Rahmen des projektbegleitenden Ausschusses diskutiert werden.

Durchgeführte Arbeiten

Im Rahmen von AP4 wurde das CMS mit den Testdaten der Unternehmen Becker, DEW und Venjakob systematisch evaluiert. Die durchgeführten Evaluierungen sind im Rahmen von Arbeitspaket 3.1 und Arbeitspaket 3.2 dargestellt.

Erzielte Ergebnisse

Systematische Evaluierung der Condition Monitoring Verfahren auf Testdatensätzen.

Arbeitspaket 4.2: Dokumentation

Ziele

Die in dem Projekt gewonnenen Erkenntnisse sollten durch das Fraunhofer IOSB-INA dokumentiert werden.

Durchgeführte Arbeiten

Im Rahmen von AP4.2 wurde eine Dokumentation der Projektergebnisse erstellt und den Mitgliedern des Projekt-begleitenden Ausschusses zur Verfügung gestellt. Darüber hinaus wurden wissenschaftliche Veröffentlichungen zu den Projektergebnissen erstellt.

Erzielte Ergebnisse

Als Ergebnis von AP 4.2 liegt eine Dokumentation vor, welche allen Interessenten auf Anfrage zur Verfügung gestellt wird. Weiterhin sind folgende Publikationen mit Bezug zu dem Projekt erstellt worden:

- *Kaja Balzereit, Alexander Diedrich, Jonas Ginster, Stefan Windmann, Oliver Niggemann: „An Ensemble of Benchmarks for the Evaluation of AI Methods for Fault Handling in CPPS“, IEEE International Conference on Industrial Informatics (INDIN), 2021*
- *Stefan Windmann: „Data-Driven Fault Detection in Industrial Batch Processes based on a Stochastic Hybrid Process Model“, conditionally accepted for: IEEE Transactions on Automation Science and Engineering*
- *Stefan Windmann: „Learning Petri Net Models from Sensor Data of Conveying Systems based on the Merging of Prefix and Postfix Trees“, accepted for: at Automatisierungstechnik.*

Zwei weitere Veröffentlichungen zur Petri-Netz-basierten Diagnose von Prozessfehler sowie zur Anomalie-Erkennung auf Basis von LSTMs sind in Arbeit.

Arbeitspaket 4.3: Projektmanagement und Transfer

Ziele

Das Projektmanagement durch das IOSB sorgt für einen reibungslosen Projektablauf und eine angenehme Kommunikation zwischen den Partnern. Die Zwischenergebnisse wurden regelmäßig zu den Projektpartnern transferiert.

Durchgeführte Arbeiten

Im Rahmen von AP 4 wurden ein Kick-Off-Meeting sowie Telefongespräche und Treffen zur inhaltlichen und organisatorischen Abstimmung der Projekthalte durchgeführt. Weiterhin wurden die Projektergebnisse den Projektpartnern über OpenProject zur Verfügung gestellt.

Erzielte Ergebnisse

Das Projekt wurde erfolgreich umgesetzt und über diverse Maßnahmen ein Ergebnistransfer an die Projektpartner durchgeführt (vgl. Transferplan). Die wesentlichen Projektergebnisse sind zusammenfassend im nächsten Abschnitt dargestellt.

Projektergebnisse

Im Rahmen des IGF-Vorhabens CLArA sind die folgenden Softwarekomponenten und Datensätze entstanden:

AP1: Anforderungsanalyse

- Anlagenspezifikationen

AP2: Datenerfassung

- Software-Komponenten zur Datenerfassung:
 - OPC UA Connector für die InfluxDB
 - InfluxDB Writer und InfluxDB Reader
 - Log-File Konverter
- Simulationsmodell eines 4-Tank-Systems
- Simulationsmodelle von Transportsystemen
- Testdatensätze (siehe Abschnitt Fehler: Verweis nicht gefunden)

AP3.1: Anomalie-Erkennung

- Algorithmen zum Modell-Lernen und zur Anomalie-Erkennung:
 - SmartThreshold
 - Verbesserte Implementierung eines HMM-basierten Ansatzes
 - LSTM-basierte Anomalie-Erkennung
 - Anomalie-Erkennung mit Automaten, Lernverfahren für Automaten
 - Anomalie-Erkennung mit Petri-Netzen, Lernverfahren für Petri-Netze

AP3.2: Fehlerursachen-Erkennung

- Diagnosemodul auf der Grundlage von Case-based Reasoning

AP3.3: Visualisierung

- Web-Applikation mit Visualisierungskomponenten für die Modell-Erstellung, das Condition Monitoring und die Fehlerursachen-Erkennung

AP3.4: Cloud-basierte Infrastruktur

- Cloud-basiertes Infrastruktur-Konzept für das parallelisierte Modell-Lernen

Die Softwarekomponenten befinden sich in einem Git-Repository, auf das über die folgende URL zugegriffen werden kann: <https://ina-project.iosb-ina.fraunhofer.de/git/clara.git>.

Auf die Testdatensätze, die sich in der OwnCloud befinden, kann über zu diesem Zweck versendete Links zugegriffen werden.

2. Verwendung der Zuwendung

Angaben zu den aus der Zuwendung finanzierten Ausgaben für Personenmonate des wissenschaftlich-technischen Personals gemäß Beleg über Beschäftigungszeiten (Einzelansatz A.1 des Finanzierungsplans), für Geräte (Einzelansatz B des Finanzierungsplans) und für Leistungen Dritter (Einzelansatz C des Finanzierungsplans):

- **Wissenschaftlich-technisches Personal (Einzelansatz A.1 des Finanzierungsplans)**

HPA-A:

2019: 8,11 PM | 2020: 12,01 PM | 2021: 8,89 PM | Gesamt: 29,01 PM

Angaben für Gerätebeschaffung oder Leistungen Dritter wurden nicht geltend gemacht.

3. Notwendigkeit und Angemessenheit der geleisteten Arbeit

Der erste wissenschaftliche Mitarbeiter und die Hilfskraft wurden planmäßig eingesetzt. Die Aufwände für den zweiten wissenschaftlichen Mitarbeiter wurden teilweise von der ersten in die zweite Projekthälfte verschoben worden, da in dieser Phase mehr Testdaten vorhanden waren, so dass ein sinnvollerer Mitteleinsatz erfolgen konnte. Die geleistete Arbeit entsprach im Übrigen dem begutachteten und bewilligten Antrag und war daher für die Durchführung des Vorhabens notwendig und angemessen.

4. Darstellung des wissenschaftlich-technischen und wirtschaftlichen Nutzens der erzielten Ergebnisse insbesondere für KMU sowie ihres innovativen Beitrags und ihrer industriellen Anwendungsmöglichkeiten

Industrielle Produktionsanlagen und insbesondere auch Anlagen des Sondermaschinenbaus sind durch eine steigende Komplexität gekennzeichnet, die die Konfiguration und Bedienung solcher Anlagen zunehmend erschwert. Daher entstehen immer mehr intelligente Assistenzsysteme, die die Anlagenbetreiber dabei unterstützen, Fehler und Anomalien in industriellen Prozessen frühzeitig zu erkennen, Prozessabläufe zu optimieren und die Ressourceneffizienz der Produktionsanlagen zu verbessern. Das Wachstumspotential für intelligente Assistenzsysteme liegt laut einer Studie vom BBC Research⁴ bei 30% pro Jahr. Dies ist besonders deswegen interessant, weil laut acatech⁵ die Gewinnspanne beim Absatz von Anlagen lediglich bei zwei bis drei Prozent liegt, die nachgelagerten Dienstleistungen allerdings Margen von bis zu 20 Prozent erwirtschaften können.

Insbesondere die Überwachung der Funktionalität, d.h. das Erkennen von Komponentenausfällen und schleichenden Veränderungen des Anlagenverhaltens ist von essentieller Bedeutung für die Anlagenbetreiber. Stillstandszeiten von Maschinen und Anlagen sind in der Regel mit erheblichen Kosten für die Anlagenbetreiber verbunden. So kostet beispielsweise der Ausfall einer Spritzgießmaschine im Mittel zwischen 10.000€ und 40.000€ pro Stunde. Ist eine defekte Maschine Teil einer Produktionskette, z.B. im Automobilbau, können die Kosten eines Ausfalls sogar schnell in den Millionenbereich gehen. Um Maschinen- und Anlagenausfälle zu vermeiden, werden in Deutschland jährliche direkte Ausgaben für Instandhaltungsmaßnahmen im Umfang von 250 Milliarden Euro, d.h. von etwa acht Prozent des deutschen Bruttoinlandproduktes, vorgenommen. Diese Kosten können durch den Einsatz von Condition Monitoring Systemen (CMS) erheblich reduziert werden.

In dem durchgeführten Projekt wurden Lösungen entwickelt, welche den Einsatz von CMS insbesondere im Sondermaschinenbau erleichtern. Zum einen wurden neue Condition Monitoring Methoden erarbeitet, welche dazu beitragen, den Anlagenzustand in verschiedenen Betriebssituationen hinreichend genau zu beurteilen. Zum anderen wurden Hürden bei der Implementierung und Inbetriebnahme überwunden, welche durch manuelle Konfigurationen und Anpassungen an unterschiedliche Anlagen entstehen. Dies wurde durch benutzerfreundliche Visualisierungskomponenten, eine einheitliche Datenanbindung sowie den Einsatz maschineller Lernverfahren erreicht. Das entwickelte System passt sich automatisch an existierende Anlagen an und detektiert Anomalien und Fehlerursachen auf der Grundlage einer gelernten Repräsentation des Systemverhaltens. Hierzu werden in einer Trainingsphase Analyse- und Diagnosemodelle aus historischen Prozessdaten gelernt und während des Betriebs des CMS zur Anomalie-Erkennung und zur Diagnose von Fehlerursachen verwendet. Durch die

4 A. McWilliams, "Smart Machines: Technologies and Global Markets," tech. rep., BBC RESEARCH, 2014.

5 C. Vornholt, ed., Smart Maintenance für Smart Factories: Mit intelligenter Instandhaltung die Industrie 4.0 vorantreiben. acatech - Deutsche Akademie der Technikwissenschaften, 2015.

entwickelte Lösung werden die Entwicklungs- und Inbetriebnahmekosten für CMS erheblich reduziert und insbesondere ein wirtschaftlicher Einsatz in kundenspezifischen Anlagen, die nur in kleinen Stückzahlen produziert werden, erreicht.

KMU aus unterschiedlichen Industriezweigen entlang der gesamten Wertschöpfungskette können einen Nutzen aus den Ergebnissen des Vorhabens ziehen:

- Für Anwender ergibt sich durch das Projekt die Möglichkeit, CMS ohne aufwendige Programmierung und Konfiguration in verschiedenartige Produktionsanlagen, die in unterschiedlichen Branchen wie der Holzverarbeitenden Industrie, der Automobilindustrie oder der Stahlverarbeitenden Industrie eingesetzt werden, zu integrieren. Dies stellt insbesondere für KMU einen echten Mehrwert dar, da diese oft nicht über die erforderlichen Ressourcen verfügen, um CMS für unterschiedliche Produktionsanlagen zu implementieren und zu konfigurieren.
- Für Systemintegratoren stellen die Projektergebnisse einen wesentlichen Beitrag zum Know-how-Aufbau im Bereich Cloud-basierter CMS dar. Dadurch können sie die Projektergebnisse in ihre Produkte integrieren und auf diese Weise Umsatzwachstum erzielen. Durch das Projekt wird es Systemintegratoren ermöglicht, das vorhandene Produktportfolio um Produkte im Bereich Cloud-basierter CMS zu erweitern. Darüber hinaus können vorhandene Private und Public Cloud Lösungen von Systemintegratoren perspektivisch auf Condition Monitoring Lösungen für industrielle Anwendungen ausgeweitet werden.
- Komponentenhersteller bekommen durch die Projektergebnisse einen besseren Einblick in die Anforderungen der Anwender bzgl. der Datenanalyse. Gerade KMUs, deren Wissen über eine Gesamtanlage oder einen Produktionsprozess vergleichsweise gering ist, werden so in die Lage versetzt, neue (datenbasierte) Produkte und Services auf Komponentenbasis zu entwickeln. Aus Sicht der Komponentenhersteller ist es dabei insbesondere von Interesse zu verstehen, mit welchen informationstechnischen Infrastrukturen sich Komponenten zukünftig verbinden müssen, um höheren Hierarchieebenen Daten bereitstellen zu können. Komponentenherstellern können die Ergebnisse des Projektes bei der Produktentwicklung und der Entwicklung zukünftiger Produkt-Roadmaps berücksichtigen. Ein Nutzen für KMUs ergibt sich insbesondere auch in der Betrachtung einer einheitlichen Integration heterogener Datenquellen und der Formulierung zukünftiger Anforderungen an die eigenen Produkte auf Basis der in dem Projekt gesammelten Erfahrungen.

5. Plan zum Ergebnistransfer in die Wirtschaft

Durchgeführte Transfermaßnahmen während der Projektlaufzeit:

Maßnahme	Ziel	Ort/Rahmen	Zeitraum
Bereitstellung von Zwischenergeb. an PA und Interessierte	Transfer von Zwischenergebnissen	Arbeitspaket 4.3	Projektbegleitend
Projektbegleitender Ausschuss (PA)	Überwachung der Projektziele und Beschluss weiterführender Maßnahmen	1. Lemgo (Kick-Off) 2. Rheda-Wiedenbrück 3. Webkonferenz 4. Webkonferenz 5. Webkonferenz (Projektabschluss)	10.07.2019 18.02.2020 21.07.2020 09.02.2021 26.05.2021

Transfermaßnahmen nach Projektabschluss:

Maßnahme	Ziel	Ort/Rahmen	Zeitraum
Wissenschaftliche Veröffentlichung auf Fachkonferenzen	Wissenschaftliche Erkenntnisse werden auf Fachkonferenzen veröffentlicht	1. K. Balzereit, A. Diedrich, J. Ginster, S. Windmann, O. Niggemann: „An Ensemble of Benchmarks for the Evaluation of AI Methods for Fault Handling in CPPS“, INDIN 2021. 2. T. Westerhold, S. Windmann: „LSTM-based fault detection“, designated for: ETFA 2022.	1. Juli 2021 2. geplant für September 2022
Workshops / Know-how-Transfer	Übertragung des Know-hows in die Unternehmen des PA	Video-Schulungen mit den einzelnen Mitgliedern des Projektbegleitenden Ausschusses	Juli 2021
Lehre und Weiterbildung	Vorstellung von Projektergebnissen (z.B. neuer Algorithmen) in Vorlesungen	Vorlesungen des Antragstellers an der TH OWL	Wegen Professur-Wechsel voraussichtlich nicht durchführbar
Vorstellung von Ergebnissen in Standardisierungsorganisationen	Weiterentwicklung von Informationsmodellen und Referenzarchitekturen für CMS	AutomationML e.V., OPC Foundation, VDMA 24582, etc.	Entfällt, da im Projekt keine standardisierungsrelevanten Ergebnisse entstanden sind
Abschlussbericht	Zusammenstellung der Ergebnisse	Bericht und Präsentation	November / Dezember 2021
Evaluierung bei Anwendern	Transfer der Ergebnisse an Unternehmen außerhalb des PA	Gespräche mit Betreibern großer Produktionsanlagen	Innerhalb von 1-2 Jahren nach Projektabschluss
Integration in Leitsysteme	Integration der Ergebnisse in existierende Leitsysteme	Gespräche mit Herstellern von Leitsystemen	Innerhalb von 1-2 Jahren nach Projektabschluss
Information, Beratung, Qualifikation	Projektergebnisse werden an den Mittelstand vermittelt	Schulungen und Demonstrationen in der SmartFactoryOWL	Innerhalb von 1-2 Jahre nach Projektabschluss
Demonstration der Ergebnisse	Darstellung der Verwertbarkeit der Technologie	Industriemessen (z.B. Hannovermesse)	Im Jahr nach Projektende
Wissenschaftliche	Wissenschaftliche Erkenntnisse werden in	1. S. Windmann: „Data-Driven Fault Detection in Industrial Batch	1. Einreichung der finalen Version erfolgt

Veröffentlichung in Zeitschriften	Magazinen, Journals, etc. veröffentlicht	Processes based on a Stochastic Hybrid Process Model“, conditionall accepted for: T-ASE. 2. S. Windmann: „Learning Petri Net Models from Sensor Data of Conveying Systems based on the Merging of Prefix and Postfix Trees“, accepted for: at Automatisierungstechnik 3. S. Windmann: „Petri Net Based Diagnostic Approach for Concurrent Discrete Event Systems“, designated for: at Automatisierungstechnik.	im Dezember 2021 2. Veröffentlichung ist im November 2021 angenommen worden, Datum der Veröffentlichung steht noch nicht fest. 3. Einreichung erfolgt im Dezember 2021
Vorstellung in Gremien	Wissenstransfer	Frühjarssitzung 2022 der ZVEI-Forschungsgemeinschaft Automation	Frühjahr 2022

Erläuterungen zur Realisierbarkeit der noch geplanten Maßnahmen

Im Folgenden wird dargelegt, wie die Realisierbarkeit der noch geplanten Maßnahmen einzuschätzen ist. Die Berücksichtigung der Projektergebnisse in der Lehre an der TH OWL und die Vorstellung von Ergebnissen in Standardisierungsorganisationen sind aus den in der Tabelle aufgeführten Gründen voraussichtlich nicht realisierbar. Die Evaluierung bei Anwendern, die Integration in Leitsysteme und die Information, Beratung und Qualifikation innerhalb von 1-2 Jahren nach Projektende erscheint realistisch, da diesbzgl. bereits konkrete Gespräche mit interessierten Unternehmen laufen. Die Demonstration der Projektergebnisse auf Industriemessen hängt davon ab, inwieweit dies Corona-bedingt möglich ist. Die in der Tabelle aufgeführten wissenschaftlichen Veröffentlichungen sind mit Ausnahme der Veröffentlichung zur LSTM-basierten Fehlererkennung bereits erstellt und teilweise angenommen worden. Es erscheint realistisch, auch die Projektergebnisse zur LSTM-basierten Fehlererkennung wie geplant in 2022 zu veröffentlichen.