

Digitalisierung implementieren

KI-Einsatz in KMU: Einstiegshürden ausräumen

I. Heider, H. Yu, N. Krischke, B. Wirth, A. Puchta, J. Fleischer

Anwendungen künstlicher Intelligenz (KI) bieten für die Produktionstechnik enorme Potenziale. Diese sind vor allem in kleinen und mittelständischen Unternehmen (KMU) nicht umfassend ausgeschöpft. Ein Grund dafür ist, dass die Umsetzung von KI-Projekten Ressourcen benötigt, welche die KMU oft nicht eigenständig bereitstellen können. Vorgestellt wird ein Konzept, um das Deployment von KI-Modellen im Produktionsumfeld zu begleite. Es ist einsetzbar unter verschiedenen Randbedingungen und mit geringem Eigenentwicklungsanteil.

STICHWÖRTER

Künstliche Intelligenz, Industrie 4.0

Clearing entry hurdles for AI deployment in SMEs – Artificial intelligence for German SMEs

Artificial intelligence (AI) holds enormous potential for application in the field of production engineering. SMEs, in particular, are not ready to fully exploit these opportunities. One reason is that implementing AI projects requires resources that SMEs often are not able to provide on their own. This paper presents an approach to accompany the deployment of AI models in the production environment. It can be applied with various boundary conditions and a low proportion of in-house development.

1 Einleitung

Die Möglichkeiten KI (künstliche Intelligenz)-Anwendungen in der Produktionstechnik einzusetzen sind vielfältig. Von der Erkennung von Anomalien in der Zerspanung, der Qualitätskontrolle von Zwischenprodukten auf Basis von Bilddaten bis hin zur KI-basierten Bestimmung von Werkzeugverschleiß ist praktisch kein Anwendungsfall undenkbar.

Vor allem in kleinen und mittelständischen Unternehmen (KMU) stehen den Möglichkeiten und dem enormen Potenzial von KI-Anwendungen jedoch Einstiegshürden gegenüber. Vom ersten Konzept zur Lösung eines Problems mittels KI bis hin zur Implementierung im Produktivbetrieb sind Kompetenzen aus unterschiedlichsten Fachbereichen nötig. Das technische Verständnis für die Problemstellung liegt dabei hauptsächlich bei Technikern und Ingenieuren. Zur Erfassung, Aufzeichnung und Speicherung der relevanten Daten sind die Fähigkeiten eines Data Engineers notwendig. Ein Data Scientist wiederum hat zur Aufgabe, das technische Problem in ein KI-Problem zu übersetzen und dieses zu lösen. Zuletzt werden die Fähigkeiten des Software Engineers benötigt, um die KI-Anwendung in ein funktionsfähiges Softwareprodukt zu überführen [1].

Diese Aufteilung von Aufgabenbereichen und Rollen bei der Durchführung von KI-Projekten fällt unter den Begriff MLOps (Machine Learning Operations). MLOps ist eine Sammlung von Methoden und Best Practices für die KI-Entwicklung und hat zudem das Ziel, aufzuzeigen, welche personellen und technischen Ressourcen aufgebracht werden müssen, um eine KI-Anwendung nicht nur zu entwickeln, sondern auch zum Einsatz zu bringen [1]. Die höchsten Einstiegshürden für deutsche Unternehmen auf

dem Gebiet der künstlichen Intelligenz ergeben sich aus mangelnden personellen, finanziellen und technischen Ressourcen sowie unzureichender Datenverfügbarkeit [2]. Vor allem KMU fehlen die Ressourcen für eine eigene KI-Implementierung. Während beim Einsatz von KI-Anwendungen in der Produktionstechnik Potenziale ungenutzt bleiben, ist die Auswahl frei verfügbarer (Open-Source)-Technologien für eine KI-basierte Lösung von Fragestellungen der Produktionstechnik nach aktuellem Stand der Technik enorm.

Dieser Beitrag legt dar, welche Funktionalitäten benötigt werden und welche Technologien prinzipiell genutzt werden können, um mit begrenzten Ressourcen eine eigene prototypische Implementierung von KI-Anwendungen zu ermöglichen. Es wird ein Ansatz vorgestellt, der etablierte Open-Source-Bibliotheken zu einem Framework kombiniert, welches den Einsatz eines trainierten KI-Modells für verschiedene Anwendungsfälle und unter unterschiedlichen Randbedingungen möglich macht. Der Schwerpunkt liegt auf dem Einsatz (dem Deployment) einer KI-Anwendung. Abschließend wird die Implementierung des Frameworks in einem konkreten Anwendungsfall aus der additiven Fertigung illustriert.

2 KI-Einsatz im Feld: Deployment-Konzept

2.1 Grundkonzept für ein KI-Deployment

Das Modell ist Herzstück einer einsatzfähigen KI-Anwendung. Um einen Mehrwert aus den Aussagen des Modells erzielen zu können, ist eine Infrastruktur notwendig, innerhalb welcher das Modell agieren kann. Rohdaten müssen aufgenommen, vorverarbeitet und gegebenenfalls abgespeichert werden, bevor sie dem



Bild 1. Übersicht über die Funktionsbereiche beim KI-Deployment. *Grafik: wbk*

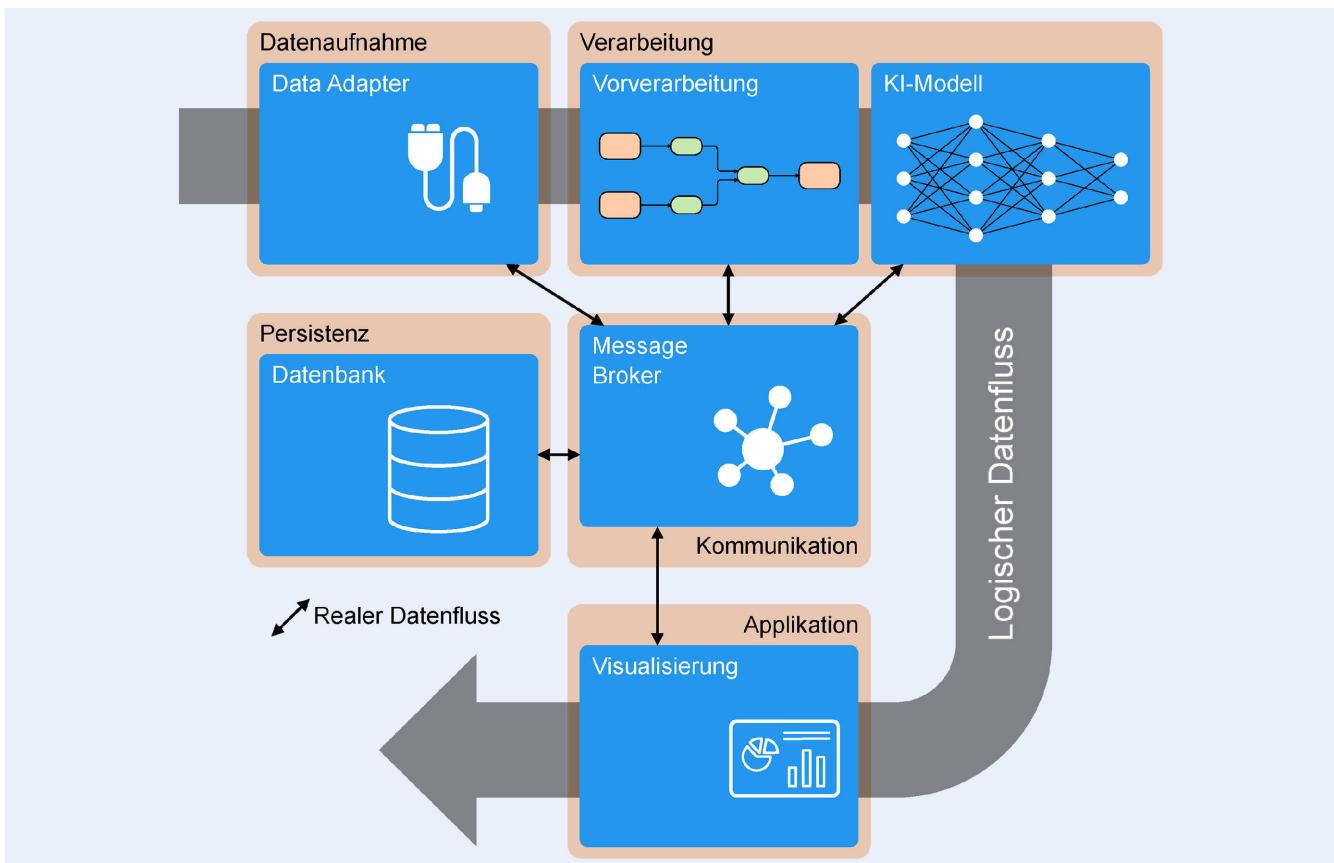


Bild 2. Konzeptionelle Systemarchitektur. *Grafik: wbk*

Modell zugeführt werden. Anschließend kann die Aussage des Modells an einen Nutzer übergeben werden oder fließt wieder in den Produktionsprozess ein. In der Produktionstechnik bietet sich die Nutzung von Zeitreihendaten für den Einsatz in KI-Anwendungen an. Beispiele sind Motorströme, Achspositionen und Temperatursignale, die oftmals über die Maschinensteuerung verfügbar sind, sodass keine zusätzliche Sensorik aufgerüstet werden muss. Für den Einsatz einer KI-Anwendung auf Basis von Zeitreihendaten sind die in **Bild 1** dargestellten Funktionsbereiche erforderlich.

Diese fünf Funktionsbereiche dienen zur Strukturierung der Systemkomponenten, um ein Framework für ein Deployment zu bilden. Der folgende Abschnitt erläutert die verschiedenen Funktionsbereiche und ihr Zusammenspiel. In Kapitel 2.2 werden die Umsetzung der Systemkomponenten und des Gesamtkonzeptes beschrieben sowie die Auswahl der eingesetzten Tools erklärt.

Die Datenaufnahme bezeichnet das Erfassen von Rohdaten aus einer Datenquelle, wie etwa Sensor- oder Steuerungsdaten. Bedingt durch die Vielfalt an Sensor- und Maschinenschnittstellen werden verschiedene Formate und Protokolle zur Übertragung von Rohdaten verwendet. Eine effektive Datenaufnahme sollte in

der Lage sein, eine Vielzahl von Kommunikationsstandards abzudecken.

Der Funktionsbereich Kommunikation umfasst die Verteilung von Daten zwischen den Systemkomponenten. Die über die Datenaufnahme erfassten Rohdaten werden über die Kommunikation an andere Komponenten weitergeleitet. In diesem Kontext bieten broker-basierte Implementierungen eine flexible Lösung. Dabei müssen die Komponenten, welche Daten konsumieren beziehungsweise publizieren, nicht direkt miteinander kommunizieren, sondern tauschen ausschließlich Daten mit einem zentralen Knotenpunkt, dem Broker aus (siehe Bild 2). Durch den Einsatz eines Brokers wird das Problem der gegenseitigen Identifikation der Kommunikationspartner gelöst. Zudem entfallen Aufwände bei der Erweiterung des Systems, da lediglich die Kommunikation mit dem Broker konfiguriert werden muss [3].

Der dritte Funktionsbereich, die Verarbeitung, umfasst das eigentliche KI-Modell sowie die Vorverarbeitung. Die Vorverarbeitung, auch als Preprocessing bekannt, umfasst alle Schritte zur Umwandlung von Rohdaten in eine Form, die von einem Modell verarbeitet werden kann. Die Daten liegen als Ergebnis der Vorverarbeitung, als sogenannte Features, vor. Beispiele für die Datenvorverarbeitung sind die Datenbereinigung, die Skalierung

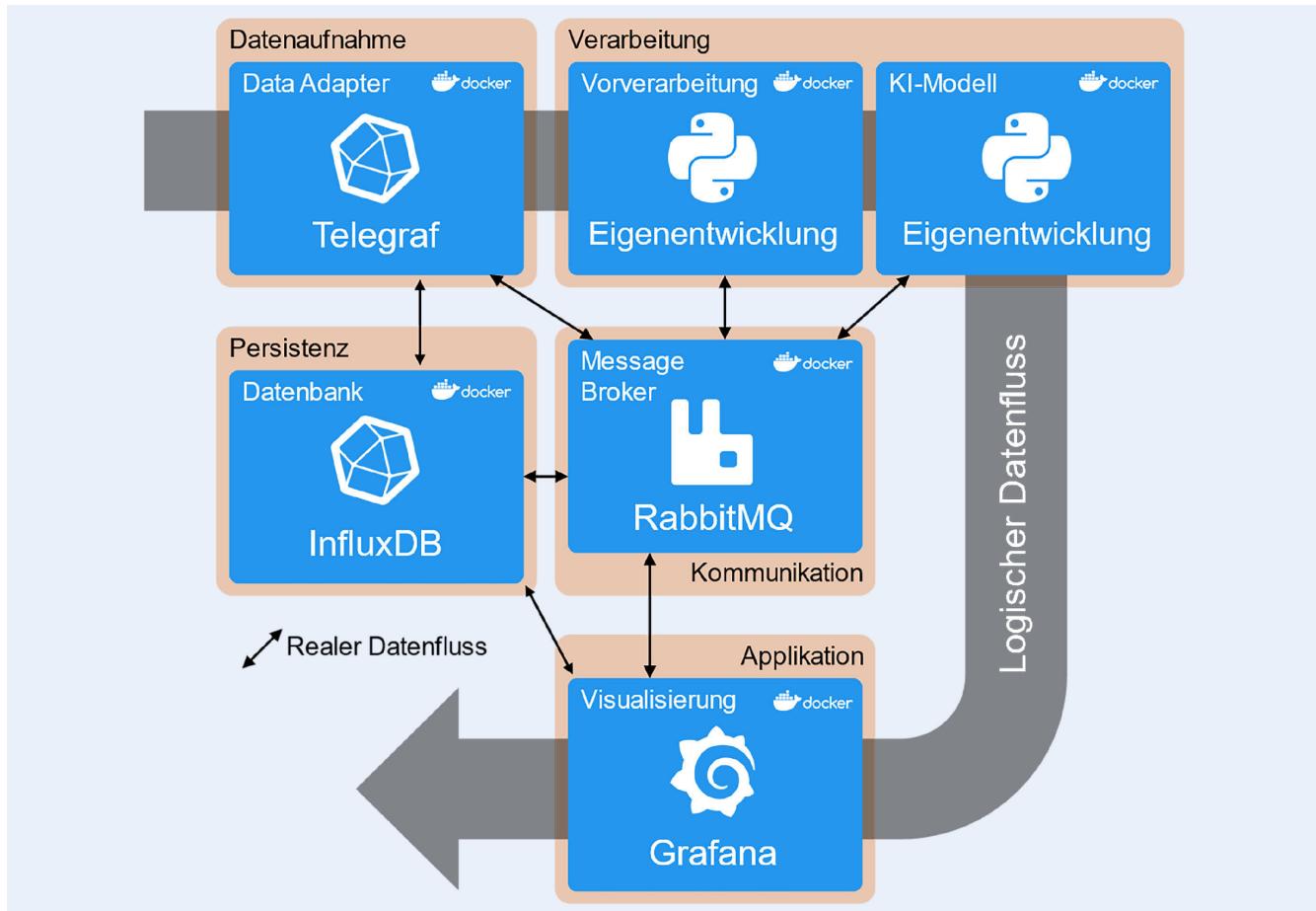


Bild 3. Systemarchitektur des Lösungsvorschlags. *Grafik: wbk*

und die Normalisierung. Unter Persistenz versteht man das Abspeichern von Daten. Typischerweise werden Rohdaten, Features und Modellaussagen abgespeichert, um die Nachverfolgbarkeit von Ergebnissen und kontinuierliches Modelltraining zu erlauben. Hier kommen Datenbanken zum Einsatz, wobei zwischen unterschiedlichen Datenbankkonzepten unterschieden werden kann. Für die Arbeit mit Zeitreihendaten, wie im hier behandelten Anwendungsfall, bieten Zeitreihendatenbanken, im Gegensatz zu relationalen Datenbanken, Vorteile [4]. Der als Applikation bezeichnete Funktionsbereich liefert den eigentlichen Mehrwert der KI-Anwendung. Er setzt die Aussage eines Modells in für den Nutzer wertvolle Information um. Im einfachsten Fall kann dies eine Visualisierung sein, wie zum Beispiel eine Anzeige, die den aktuellen Zustand einer Maschine oder eines Prozesses bewertet. Ein weiteres Beispiel ist die Ausgabe einer Benachrichtigung bei Erreichen eines bestimmten Verschleißzustandes einer Komponente. Komplexere Applikationen können ein Not-Halt oder eine KI-basierte Prozessregelung sein.

Bild 2 zeigt, wie die vorgestellten Funktionsbereiche angeordnet werden können, um ein Framework für den KI-Einsatz zu bilden. Der graue Pfeil zeigt die chronologische beziehungsweise logische Abfolge, in welcher der Informationsfluss von der Datenaufnahme bis zur Nutzung des KI-Outputs erfolgt. Die schwarzen Pfeile zeigen den tatsächlichen Datenfluss zwischen den Systemkomponenten und dem Broker. Nicht dargestellt ist die Datenquelle, die je nach Anwendungsfall unterschiedlich ist.

2.2 Lösungsvorschlag und genutzte Tools

Der Aufbau des Deployment-Frameworks und die Wahl der genutzten Tools sind in **Bild 3** dargestellt.

- Folgende Anforderungen motivieren dessen Gestaltung:
- KI-Deployment unter geringem Eigenentwicklungsanteil
- Anbindung an unterschiedliche Datenquellen
- Einsatz auf verschiedenen Zielsystemen

Die in Bild 3 aufgeführten Komponenten sind jeweils als eigene Services in Docker Containern ausgeführt. Diese Modularität der Komponenten ermöglicht eine Wiederverwendung über mehrere Anwendungsfälle hinweg. Ferner wird so ermöglicht, dass bei der (Neu-)Entwicklung einzelner Komponenten andere Komponenten nicht berücksichtigt werden müssen. Das Vorgehen motiviert sich zudem aus dem Ziel, den Einsatz auf verschiedenen Zielsystemen, wie Edge Devices oder Entwickler-PCs, zu ermöglichen. Docker Container umfassen den Programm-Code sowie benötigte Bibliotheken. Ihre Ausführbarkeit ist auf allen Betriebssystemen gewährleistet, die einen Linux-Kernel bereitstellen, wie etwa Windows ab Windows 10 [5].

InfluxDB ist eine performante Zeitreihendatenbank, die auf die Aufnahme großer Datenströme sowie schnelles Lesen und Schreiben von Daten ausgelegt ist. Anwendungsfälle erstrecken sich vom Laborbetrieb, über Großforschung bis hin zu Applikationen von Finanzdienstleistern. Von InfluxDB (aber auch anderen Datenbanken) abdeckte Funktionalitäten sind, neben dem Speichern von Daten, Aggregationen und Downampling sowie

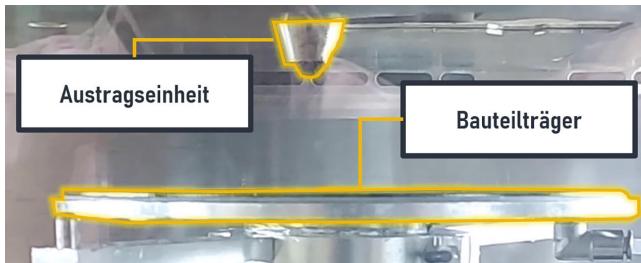


Bild 4. Bauraum des „freeformer 300-3X“ Austragseinheit und Bauteilträger. *Grafik: wbk*

das Löschen alter Daten in regelmäßigen Abständen. InfluxDB hat sich als IoT-Datenbank etabliert [6]. Die Wahl von InfluxDB fällt ferner durch die Verfügbarkeit von Telegraf, einem angegliederten Tool zur Datenaufnahme, das im sogenannten Tick Stack mit InfluxDB integriert ist. Telegraf enthält eine Vielzahl von Datenadapters für gängige Kommunikationsstandards wie OPC UA und MQTT und bietet die Möglichkeit, eigene Plugins für die Datenaufnahme zu einzubinden [7]. Neben der Integration gängiger Protokolle per Standard-Plugin ist somit die Anbindung weiterer Kommunikationsstandards möglich, was auch die Einbindung von Brownfield-Anlagen ermöglicht.

Die Wahl von RabbitMQ als Message Broker motiviert sich auf ähnliche Weise durch eine einfache Implementierung und die Popularität unter Nutzern [8]. Die Erweiterung der vorgestellten Gesamtarchitektur um weitere Komponenten, wie ein zusätzliches KI-Modell oder eine weitere Visualisierung, bringt bei der Nutzung eines Message Brokers auf Seiten der Kommunikation keinen Entwicklungsaufwand mit sich. Auf Seiten der erweiterten Komponente ist die Implementierung eines RabbitMQ Clients aufwandsarm. Im Kontext des vorgestellten Frameworks wird RabbitMQ mit dem Protokoll AMQP genutzt. Es sind aber auch Implementierungen mit anderen Standards, wie MQTT möglich [8].

Grafana ist ein gängiges Tool zur Visualisierung von Zeitreihendaten. Die Integration verschiedener Datenquellen ist einfach gestaltet. Die Anbindung an InfluxDB ist standardmäßig möglich, sodass kein Entwicklungsaufwand anfällt. Die Gestaltung eines individuellen Dashboards erfolgt über die grafische Benutzeroberfläche [7].

Während das Framework stark auf vorgefertigte Lösungen setzt, gibt es Komponenten, die für jeden Anwendungsfall mit Eigenentwicklung und somit mit Aufwand einhergehen. Die Datenvorverarbeitung sowie das KI-Modell sind individuelle Komponenten. Unter Nutzung des Frameworks beschränkt sich die für den KI-Einsatz erforderliche Eigenentwicklung im Idealfall auf diese beiden Komponenten.

3 Anwendungsfall additive Fertigung

Das aufgezeigte Framework wird beispielhaft anhand eines realen Anwendungsfalls aus der additiven Fertigung demonstriert. Als Versuchsgerät dient der „Arburg freeformer 300-3X“. Kennzeichnend für dessen Einrichtungsprozess ist das manuelle Einsetzen und Ausrichten der Austragseinheit. Beim manuellen Ausrichten der Austragseinheit durch einen unerfahrenen Bediener kann es zur Fehlausrichtung kommen. Fehlausrichtungen von geringem Ausmaß haben keinen Einfluss auf die Prozessstabilität, können aber zu einer Beeinträchtigung der Bauteilqualität führen, etwa zu

einer Beeinträchtigung der Oberflächengüte. Derartige Mängel können ausschließlich manuell und erst nach Abschluss eines Bauauftrags identifiziert werden. Da Bauaufträge sich über viele Stunden erstrecken können, ist eine automatisierte Erkennung einer Fehlausrichtung der Austragseinheit vielversprechend.

Die konkret betrachtete Fragestellung ist die Identifikation der Fehlausrichtung zwischen der Auftrageinheit und dem Bauteilträger anhand von Anomalien in Zeitreihendaten. **Bild 4** zeigt die beiden Komponenten im Bauraum des freeformers.

Ein Prozessparameter, der Aufschluss über die Fehlausrichtung geben kann, ist der Volumenstrom des Druckmaterials, der sogenannte Materialaustrag. Weitere Prozessparameter, die Aufschluss über die Fehlausrichtung der Austragseinheit sowie generell über Anomalien geben können, sind etwa Motorströme der Linearachsen. Aus Gründen der Übersichtlichkeit wird im Folgenden nur der Parameter Materialaustrag betrachtet. Ein Ansatz, um die korrekte Ausrichtung der Austragseinheit vor Beginn des Bauauftrags sicherzustellen, ist die Überwachung des Materialaustrags beim Drucken eines einschichtigen Referenzbauteils.

In einer Versuchsreihe wurden Referenzbauteile bei korrekt ausgerichteter Austragseinheit (Offset 0,0 mm) sowie bei einem axialen Offset von +/-0,2 mm gedruckt. Die Unterschiede in der Beschaffenheit der Bauteiloberfläche zeigt **Bild 5** (oben).

Die untere Reihe in Bild 5 führt die zugehörigen Verläufe des Materialaustrags auf. Zum Vergleich sind diese auf einen Wertebereich zwischen 0 und 1 skaliert. Die Anzahl der dabei aufgezeichneten Werte beträgt 200. Der Offset von +0,2 mm bedeutet einen zu großen Abstand zum Bauteilträger. Dieser hat zur Folge, dass die Bauteiloberfläche porös wird. Bei einem Offset von -0,2 mm ist der Abstand zwischen Austragseinheit und Bauteilträger zu gering, sodass sich Rillen auf der Bauteiloberfläche bilden.

Die Umsetzung des Anwendungsfalls im freeformer zeigt auf, dass durch die Nutzung des vorgestellten Frameworks der Implementierungsaufwand für KI-Anwendungen maßgeblich verringert werden kann. Im vorliegenden Falle wurden nur Preprocessing und KI-Modell anwendungsfallspezifisch entwickelt. Die weiteren Systemkomponenten wurden durch das Anpassen von Konfigurationsdateien eingerichtet. Die Datenaufnahme wird durch das Telegraf Standard-Plugin für OPC ermöglicht. Zur Einrichtung müssen nur die Adresse des OPC-Servers der Maschinensteuerung sowie die Adressen der zu betrachtenden Knoten angegeben werden. Die Speicherung von Daten basiert auf dem bereits implementierten Funktionsbaustein der Datenbank. Dabei handelt es sich um eine InfluxDB-Implementierung, die standardmäßig mit Telegraf integriert werden kann. Eine gesonderte Programmierung fällt nicht an.

Durch die beschriebene RabbitMQ-Implementierung wird die Kommunikation zwischen den Systemkomponenten abgewickelt: Anpassungen sind auch hier nicht nötig. Das Preprocessing umfasst das Ausschneiden relevanter Zeitschnitte im Verlauf des Materialaustrages, ein Resampling der Werte im Zeitschnitt auf eine feste Anzahl (hier: 200) sowie die Skalierung der Werte auf das Intervall [0,1] (Bild 5). Als KI-Modell wird ein Autoencoder verwendet, der mit Materialaustragsdaten von Referenzbauteilen bei korrekter Ausrichtung trainiert wurde. Der Autoencoder ist ein einfaches und robustes Modell zur Anomalieerkennung in Zeitreihendaten.

Die Entwicklung der beiden Komponenten Preprocessing und KI-Modell stellt einen Aufwand dar, der für die KI-Implementierung betrieben werden muss. Das Framework bietet jedoch alle

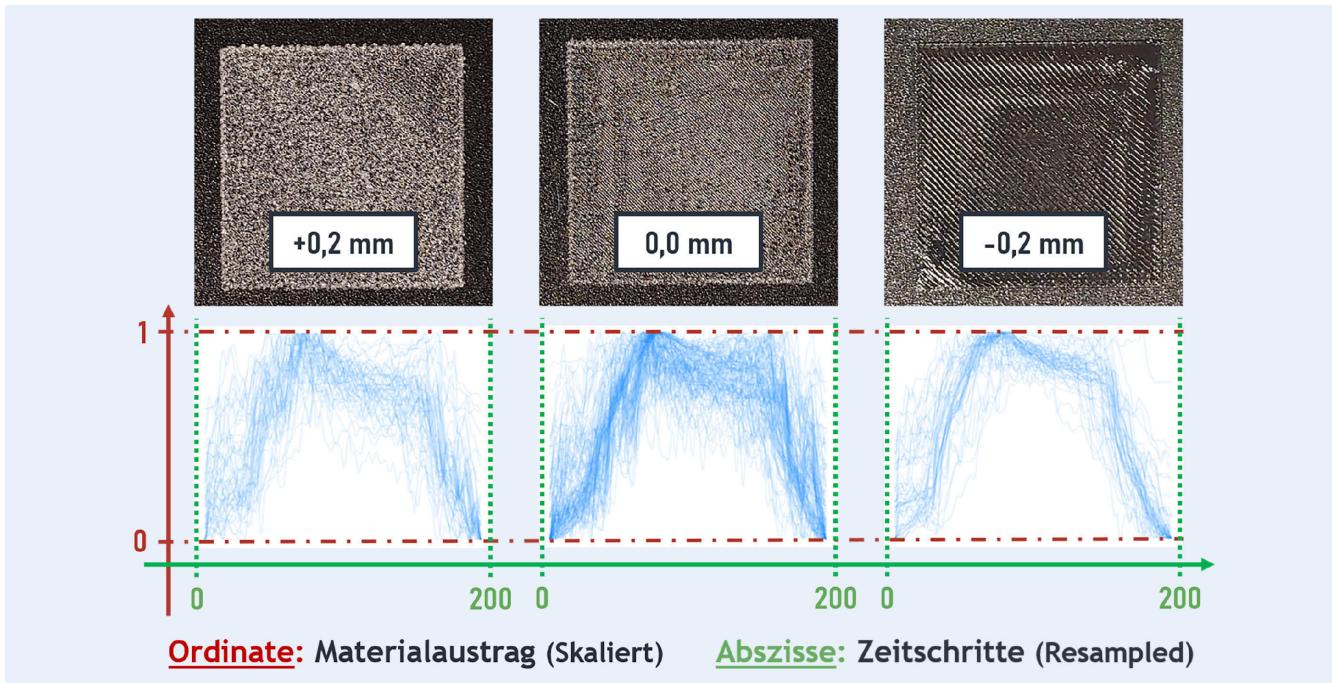


Bild 5. Einschichtige Referenzbauteile bei variablem Offset der Austragseinheit (oben). Verlauf des Materialaustauschs beim Herstellen der Referenzbauteile (unten). Skalierung des Materialaustauschs auf den Wertebereich [0,1] sowie Resampling der Abtastung auf 200 Werte. *Grafik: wbk*



Bild 6. Visualisierung der Anwendung im Grafana-Dashboard. Zeitlicher Verlauf des Materialaustausgs und KI-Bewertung des Drucks im zeitlichen Verlauf: zum Zeitpunkt t a) und Zeitpunkt t+1 b). *Grafik: wbk*

Rahmenbedingungen für den Einsatz der KI-Anwendung „im Feld“. Anpassungen sind dabei zu beachten, erfordern aber keine Neuentwicklung. Das Ergebnis der KI-Anwendung zeigt **Bild 6**.

Der Funktionsbereich der Applikation wird hier durch eine Visualisierung abgedeckt. Die oberen Graphen in Bild 6 zeigen den zeitlichen Verlauf des Materialaustausgs beim Drucken der Referenzbauteile. In den unteren Hälften ist die Bewertung der Druckvorgänge durch das KI-Modell aufgeführt. Bild 6 a) zeigt den Zustand der Maschine zum Zeitpunkt t. Der erratische Verlauf des Materialaustausgs lässt auf eine fehlerhaft eingesetzte Austragseinheit schließen. Dies wird vom KI-Modell erkannt und auf dem Dashboard angezeigt. Nach Beheben des Problems wird ein weiteres Referenzbauteil zum Zeitpunkt t+1 gedruckt (Bild 6 b)). Der Gut-Zustand ist wiederhergestellt, das KI-Modell bestätigt dies.

Die beispielhafte Implementierung soll veranschaulichen, wie eine KI-Anwendung mit möglichst geringem Aufwand und Res-

sourceneinsatz entwickelt werden kann. Bei der Generierung der Trainingsdaten wurde pragmatisch vorgegangen. Im Zuge der Entwicklungen einer KI-Anwendung für den produktiven Betrieb ist gegebenenfalls umfassendere Vorarbeit nötig. Im vorliegenden Versuch wurden insgesamt 65 Referenzbauteile gedruckt, davon 60% mit korrekt ausgerichteter Austragseinheit sowie jeweils 20% je Offset-Stufe. Wenngleich der Umfang des Trainings- und Testdatensatzes gering ist, kann eine je nach Anforderung bereits brauchbare Präzision erreicht werden. Die Precision beziehungsweise Positive Predictive Value (also die Präzision der Identifikation der Gut-Teile) beträgt 100 %. Der Negative Predictive Value, die Präzision der Identifikation von Ausschussteilen, beträgt 91 %.

4 Ausblick

Das vorgestellte Konzept und dessen Implementierung können Ausgangspunkte für die weitere Entwicklung und Untersuchung

von KI-Anwendungen sein, die auf eine funktionierende Kommunikationsarchitektur und Datenpipeline angewiesen sind.

Der Anwendungsfall zeigt, dass der Aufwand bei der Implementierung einer KI-Anwendung auf die wesentlichen Aspekte des KI-Modells und des Preprocessing reduziert werden kann. Beim Übergang auf einen anderen Anwendungsfall fallen diese erneut an. Die anderen Funktionsbereiche der Datenaufnahme, Kommunikation und Persistenz können jedoch ohne umfassende Neuentwicklung wiederverwendet werden. Das Konzept wurde im Rahmen des Forschungsprojektes AutoLern entwickelt und wird weiterhin ausgebaut. In Kooperation mit den beteiligten Forschungs- und Industriepartnern werden Funktionalitäten entwickelt, die den KI-Einsatz unter wechselnden Randbedingungen erlauben. Ziel ist ein KI-Einsatz mit anhaltender Modellvalidität.

- [3] Garcés-Erice, L.; Bauer, D.; Scotton, P.: A flexible and scalable message broker for sensor network integration. Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE. Dublin Ireland, 2009, pp. 1–9
- [4] Petrik, D. et al.: Anforderungen für Zeitreihendatenbanken im industriellen IoT. In: Meinhardt, S.; Wortmann, F. (Hg.): IoT – Best Practices. Wiesbaden: Springer Fachmedien (Edition HMD) 2021, pp. 339–377
- [5] Bhat, S.: Practical Docker with Python. Berkeley, CA: Apress Berkeley 2022
- [6] Mostafa, J.; Wehbi, S.; Chilangaryan, S. et al.: SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things. Proceedings of 34th International Conference on Scientific and Statistical Database Management. Copenhagen Denmark, 2022, pp. 1–11
- [7] Chakraborty, M.; Kundan, A. P.: Monitoring Cloud-Native Applications. Berkeley, CA: Apress Berkeley 2021
- [8] Li, R.; Yin, J.; Zhu, H. et al.: Verification of RabbitMQ with Kerberos Using Timed Automata. Mobile Networks and Applications 27 (2022) 5, pp. 2049–2067

FÖRDERHINWEIS

Förderorganisation: Bundesministerium für Bildung und Forschung; Förder-Projekt-Kennung: 02P20A025

DANKSAGUNG

Dieses Forschungs- und Entwicklungsprojekt wird durch das Bundesministerium für Bildung und Forschung (BMBF) im Förderprogramm „Lernende Produktionstechnik – Einsatz künstlicher Intelligenz (KI) in der Produktion (ProLern)“ in der Fördermaßnahme AutoLern – Selbstlernende Werkzeugmaschinen für eine hocheffiziente Produktion gefördert und vom Projektträger Karlsruhe (PTKA) betreut. Die Autoren danken dem Ministerium für die Förderung.



manuel heider, M.Sc.
Foto: wbk

Huiyan Yu, M.Sc.

Nikolai Krischke, M.Sc.

Benjamin Wirth, M.Sc.

Alexander Puchta, M.Sc.

Prof. Dr.-Ing. Jürgen Fleischer

Karlsruher Institut für Technologie (KIT)
wbk Institut für Produktionstechnik – Forschungsfabrik
Rintheimer Querallee 2, 76131 Karlsruhe
Tel. +49 172 141 1977
manuel.heider@kit.edu
www.wbk.kit.edu

L iteratur

- [1] Kreuzberger, D.; Kühl, N.; Hirschl, S.: Machine Learning Operations (MLOps): Overview, Definition, and Architecture. IEEE Access 11 (2023), pp. 31866–31879
- [2] Berg, A; Bitkom e.V.: Künstliche Intelligenz – Wo steht die deutsche Wirtschaft?. Stand: 2022. Internet: www.bitkom.org/sites/main/files/2022-09/Charts_Kuenstliche_Intelligenz_130922.pdf. Zugriff am 11.08.2023

L I Z E N Z



Dieser Fachaufsatz steht unter der Lizenz Creative Commons Namensnennung 4.0 International (CC BY 4.0)