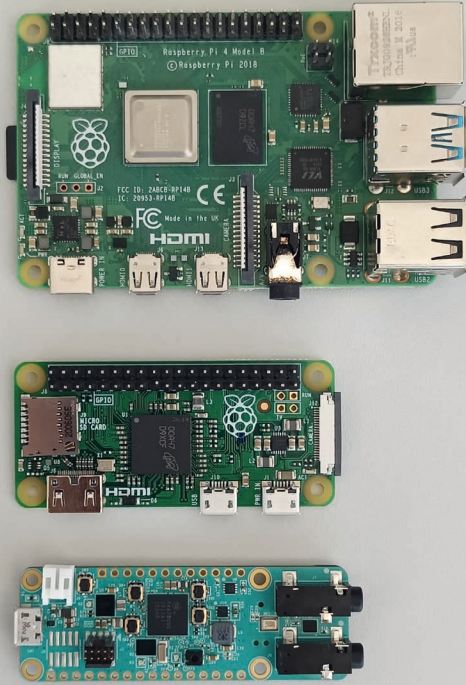


## Tiny-ML – Mikrocontroller erkennen Anomalien

Energie- und ressourcenschonende Lösungen für die erfolgsorientierte Digitalisierung im Mittelstand



**Bild 1:** Drei typische SBCs, oben ein Raspberry Pi, in der Mitte dessen abgespeckte Variante Raspberry Pi zero, unten das Evaluationsboard des MAX78000.

Der erwartete Wertbeitrag von KI in Höhe 488 Milliarden EUR [1] im Jahr 2025 für die deutsche Industrie steht im krassen Missverhältnis zu den Ergebnissen einer Studie im Auftrag des Wirtschaftsministeriums: Nur rund 6 % der deutschen Unternehmen nutzen KI; die meisten davon Großunternehmen und Startups. Im klassischen Mittelstand ist sicherlich ein großes Potenzial vorhanden - aber wie schon bei Industrie 4.0/Big Data

klemmt die Säge bei der Einführungs- und Umsetzungsunterstützung für die ersten Gehversuche. Der nachfolgende Artikel will Hilfe zur Selbsthilfe geben, zeigt bewährte Umsetzungsmethoden auf, vermittelt Praxiserfahrung und versucht, auf bereits bekannte Fallstricke aufmerksam zu machen. Denn gerade in der Zustandsüberwachung und vorausschauenden Wartung bieten sich Einsatzfelder für KI im industriellen Umfeld, die eine schnelle Amortisation versprechen.

### Was ist KI für den Mittelstand?

Bei KI denken viele erst einmal an Chat-GPT und ähnliche Systeme (Bild 2). Diese Systeme sind sehr mächtig - haben aber auch einen entsprechenden Hunger nach Energie. Nach einer US-amerikanischen Studie benötigt allein das Training eines einzelnen dieser KI-Modelle mindestens so viel elektrische Energie wie 100 Haushalte. Bild 2 zeigt ein Foto, das niemals gemacht wurde. KI hat den real existierenden Unternehmer Jeff Bezos

in eine fiktive Szene kurz vor dem Launch seiner Website amazon.com eingebettet. Das Spiel Mars Marauder (unten rechts) gab es nie; das iPhone (unten links, kaum erkennbar) wurde 2007 vorgestellt - und damit 12 Jahre nach dem behaupteten Fototermin.

Doch auch kleinere KI-Modelle, wie sie z. B. in der mittelständischen Industrie zum Einsatz kommen können, benötigen Strom. Je mehr KI-Modelle auf Desktop-PCs zum Einsatz kommen, desto schneller werden sie sich zu einem weiteren Lastfaktor für die Energieversorgung entwickeln - und der Stromverbrauch zu einem Kostenfaktor.

### Ressourcen- und energieeffiziente KI-Lösungen

Aus diesen Überlegungen heraus ging das Labor für hardwarenahe Digitalisierung an der Technischen Hochschule Deggendorf (THD) der Frage nach, wie sich ressourcen- und energieeffiziente KI-Lösungen entwickeln und umsetzen lassen. Der Fokus liegt dabei auf eng begrenzten Anwendungen, welche auf vergleichsweise kleinen und preiswerten Rechnern oder Mikrocontrollern laufen können. Entsprechend werden solche Lösungen nachfolgend als "Tiny" bezeichnet. Gemeinsam mit dem Unternehmen Elec-Con technology aus Passau, Hersteller digital geregelter Stromversorgungen, untersucht die THD, welche speziellen Möglichkeiten, aber auch Einschränkungen sich aus dem Einsatz solcher Modelle ergeben.

### Schon Künstliche Intelligenz oder nur Maschinelles Lernen?

Prinzipiell zielt Künstliche Intelligenz (KI) darauf, Maschinen zu befähigen, angemessen und vorausschauend in ihrer Umgebung zu agieren, also auch bis zu einem gewissen Grad Sinneseindrücke zu erfassen und darauf zu reagieren, Informationen aufzunehmen und zu verarbeiten, als auch Wissen zu speichern, Sprache zu verstehen und zu erzeugen sowie per-

spektivisch Probleme zu lösen und Ziele zu erreichen.

Bereits anhand dieser Definition wird klar, dass so ein Ansatz viel zu mächtig ist, um damit vergleichsweise einfache Herausforderungen zu lösen, etwa zu erkennen, dass ein gerade gefertigtes Teil in bestimmten Eigenschaften vom erwarteten Ergebnis abweicht.

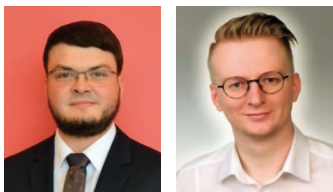
### Maschinelles Lernen

Daher fokussiert sich dieser Artikel auf das Teilgebiet Maschinelles Lernen (ML) der Künstlichen Intelligenz. Um ein fehlerhaft gefertigtes Teil sicher identifizieren zu können, muss die Maschine viele, viele Gut-Teile "gesehen" und erfasst haben. Durch diesen iterativen Prozess lernt das System, dass bestimmte Abweichungen tolerabel sind (etwa, weil in der Realität Fertigungstoleranzen nicht vermeidbar sind). Damit sind auch bereits die Kernbereiche des maschinellen Lernens umrissen. Aus den Daten, mit denen das System "gefüttert" wird, etwa Kamerabilder einer bestückten Leiterplatte, entwickeln entsprechende Algorithmen ein statistisches Modell. Das System lernt also nicht einfach Bilder auswendig, sondern es versucht, in jedem neuen Kamerabild bereits gelernte Muster und Gesetzmäßigkeiten zu erkennen.

### Unbekannte Daten selbständig beurteilen

Ziel ist es, das System zu befähigen, neue, bisher unbekannte Daten selbständig zu beurteilen. Also z. B. in einem neuen Kamerabild zu erkennen, ob ein Widerstand auf einer Leiterplatte nur ver-rutscht aber beidseitig kontaktiert ist (Gut-Teil), oder aber z. B. der Widerstand fehlt oder ein "Grabstein" produziert worden ist (Bild 3). In diesem Fall aktiviert das System z. B. die automatisierte Aussonderung und leitet die Baugruppe mit der Info über die identifizierten Fehler weiter zur Nacharbeit.

Bild 3 zeigt den Grabsteineffekt: Wenn sich beim Reflow-Löten kleine Bauteile aufstellen, ist ein Teil der Anschlüsse nicht kontaktiert. Die-



Autoren:

Andreas Federl (li), M.Sc.

TH Deggendorf

Markus Böhmisch (re), M.Sc.

Entwicklungsingenieur

Elec-Con technology GmbH

www.elec-con.com

ser Fehler tritt in der Elektronikfertigung häufig auf. Da Widerstände vergleichsweise klein sind (bei der Bauform 0402 etwa 1 mm lang und 0,5 mm breit), sind die Fehler nicht immer leicht zu erkennen; die Suche für das menschliche Auge ist ermüdend. Daher kommt für solche Analysen immer öfter KI zum Einsatz.

## Leistungsfähigkeit

Hat man also, wie gerade skizziert, eine bestimmte Aufgabe identifiziert, die per Tiny-ML gelöst werden soll, stellt sich die Frage nach der erforderlichen Leistungsfähigkeit. Ein System, das zwingend hochaufgelöste Bilder mehrerer Videokameras verarbeiten muss, benötigt zweifellos mehr (Rechen-)Leistung, als ein System, das lediglich Standbilder zur Beurteilung heranzieht. Häufig stellt man bei dieser Betrachtung fest, dass ein Embedded-Rechner oder ein entsprechend ausgestatteter Mikrocontroller für die Aufgabe völlig ausreicht.

## Weiterführende Überlegungen

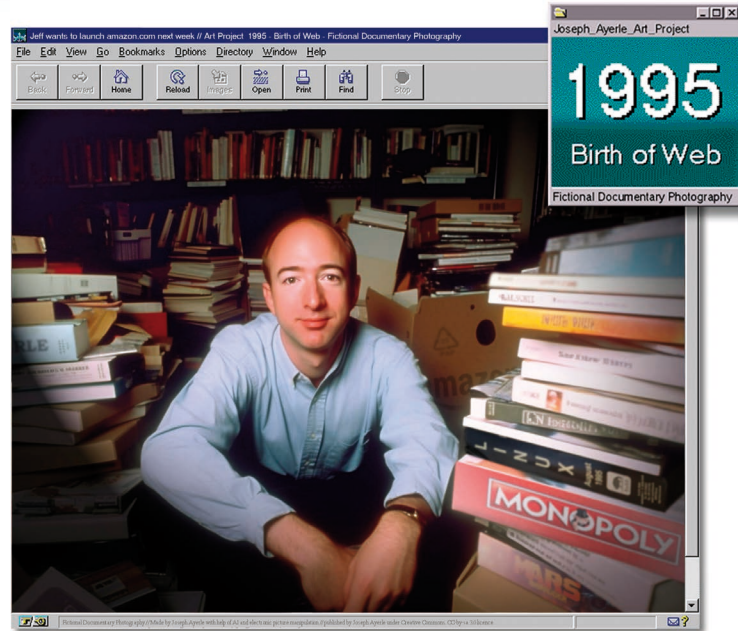
Wenn es gelingt, Tiny-ML direkt auf einem Embedded-Rechner oder Mikrocontroller zu realisieren, ergeben sich daraus einige Vorteile.

- **Ressourcen- und Energieverbrauch:** Tabelle 1 vergleicht die wesentlichen Parameter einiger grundsätzlich geeigneter Systeme mit einem Desktop-PC.
- **Datensicherheit und Datenschutz:** Bei ausreichender Aus-

stattung mit Rechenleistung bleiben die gesammelten Daten auf dem System. Weder für die Auswertung noch für das Training des Systems müssen rückverfolgbare oder auf einzelne Personen beziehbare Rohdaten weitergegeben werden. Stattdessen gibt ein entsprechend leistungsfähiges System nur hoch aggregierte und anonymisierte Daten an übergeordnete Systeme weiter.

- **Cybersicherheit:** KI-Systeme sind beliebte Ziele für Hackerangriffe, und diese können durchaus sehr subtil sein, etwa indem ein böswilliger Angreifer dem System bestimmte Botschaften "beibringt". Embedded Systeme sind prinzipbedingt viel weniger angreifbar. Zum einen wegen fehlender oder eingeschränkter Netzwerkanbindung, zum anderen aber auch, weil darauf z. B. Betriebssysteme zum Einsatz kommen, für die es kaum entsprechende Schad- oder Angriffswerkzeuge gibt.

Bei einem entsprechenden Systemzuschnitt ist es technisch auch nicht möglich, dieses bzw. das angeschlossene Netzwerk über die Sensoren anzugreifen [2]. Mittels eines KI-Ansatzes kann bei bestimmten Aufgaben sogar komplett auf Sensoren verzichtet werden. Das reduziert die Systemkosten und kann die Cybersicherheit deutlich verbessern. So gelingt es etwa, mittels Tiny-ML das Strom-Zeit-Profil eines Verbrauchers auszuwerten und darin Anomalien zu identifizieren.



**Bild 2:** Dieses Foto ist niemals gemacht worden. KI hat den real existierenden Unternehmer Jeff Bezos in eine fiktive Szene eingebettet. © gemeinfrei

Nur ein Beispiel dafür ist das zuverlässige Erkennen eines beschädigten CPU-Lüfters einer Hochleistungs-Grafikkarte.

## Zustandsüberwachung und vorausschauende Instandhaltung

Im industriellen Umfeld sind Applikationen im Bereich der Zustandsüberwachung und der vorausschauenden Wartung häufige erste Einsatzgebiete von KI-Verfahren, weil sie in aller Regel wenig komplex sind, aufgrund vermiedener ungeplanter Maschinenstillstände aber eine kurze Amortisationszeit versprechen.

Gerade wenn es darum geht, unscharfe Muster oder Anomalien zu erkennen, kann Tiny-ML seinen Vorteil ausspielen. Damit lassen sich auch Prognosen über die nahe Zukunft treffen, basierend auf den gesammelten Daten und erlernten Mustern. Ein Beispiel ist das Erkennen des bevorstehenden Ausfalls eines Lagers. Rechtzeitig vor dem ungeplanten Stillstand lässt sich der Austausch des Lagers in die Wartungsschicht verschieben und sich so die Effizienz von Wartungsprozessen erheblich steigern [3].

## Anwendung von KI in eingebetteten Systemen

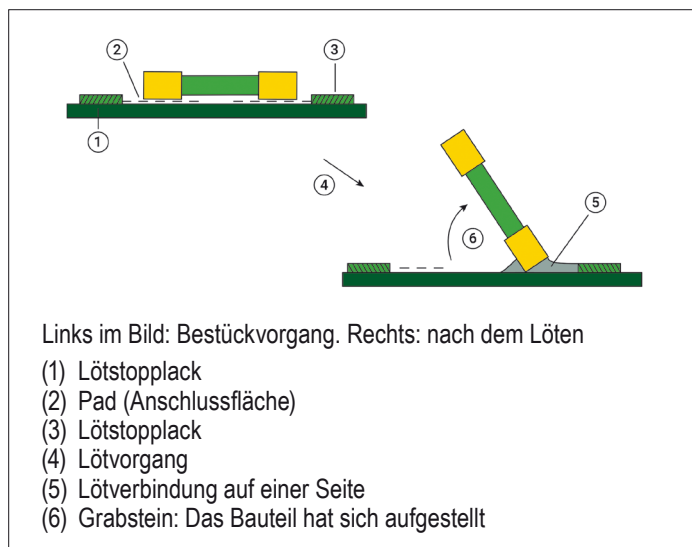
Als Plattform für eingebettete Systeme dienen häufig SBCs [4].

Beispiele für solche Systeme sind Raspberry Pi, BeagleBone oder Arduino. Diese kleinen Universalrechner kommen auch in der Automatisierung, der Robotik, zur Anbindung von Geräten an das IoT (Internet der Dinge) oder in der Lehre zum Einsatz. Sie sind günstig, vielseitig verwendbar sowie einfach zu programmieren.

Die permanente Arbeit einer weltweiten Community sorgt dafür, dass es für viele Applikationen bereits Beispiellösungen gibt; man das Rad also nicht immer vollständig neu erfinden muss.

## Spezieller SBC

Je höher die Stückzahl eines Geräts ist, desto mehr lohnt sich die Entwicklung eines speziellen SBC mit applikationsspezifischer Programmierung für genau diese Anwendung. Teilweise werden solche Systeme um dafür optimierte Mikroprozessoren bzw. Mikrocontroller herum aufgebaut. So sind etwa in modernen Kraftfahrzeugen bis zu hundert solcher eingebetteter Systeme im Einsatz; dort allerdings häufig als Steuergeräte bezeichnet. Das Aufgabenspektrum reicht von der komplexen und mit mehreren Mikrocontrollern bestückten Motorsteuerung über Bremssysteme, Navigations- und Entertainment-Systeme bis hin zum elektronischen Blinker-Relais.



**Bild 3:** Grabsteineffekt: Wenn sich beim Reflow-Löten kleine Bauteile aufstellen, ist ein Teil der Anschlüsse nicht kontaktiert. © Elec-Con

## Leistungsdaten in Vergleich

	PC	Raspberry Pi 2 (B)	Raspberry Pi (Zero)	Arduino Nano	MAX78000FTHR
<b>CPU</b>	Intel	ARM Cortex A7	ARM Cortex A7	megaAVR	ARM Cortex M4 + Neural Network Accelerator
<b>Taktfrequenz [GHz]</b>	2...5	0,6	1	0,016	0,1
<b>Anzahl Kerne</b>	1...8	4	1	1	2
<b>Busbreite [bit]</b>	64	32	32	8	32
<b>Arbeitsspeicher [GB]</b>	4...32	4	0,512	0,000032	0,00064
<b>Rechenleistung [GFLOPS]</b>	200...400	4...5	0,2...0,3	0,0013	
<b>Stromaufnahme (typ.) [W]</b>	60	5	1,3	0,17	0,25
<b>Raumbedarf (typ) [l]</b>	5	0,12	0,03	0,01	0,02
<b>Energiebedarf [8] [kWh]</b>	5.256	438	113	15	22
<b>Energiekosten [9] [EUR]</b>	1.300	110	28	4	6
<b>Preis (typ.) [EUR]</b>	1.000	60	30	20	60

Tabelle 1: Vergleich der Leistungsdaten unterschiedlicher Rechner-Plattformen

### Tiny-ML-Applikationen

Sowohl SBCs als auch Mikroprozessoren sind wegen ihrer Flexibilität sowie ihrer kleinen Abmessungen ideal für Tiny-ML-Applikationen geeignet. Sie sind in der Lage, komplexe Berechnungen durchzuführen, während sie gleichzeitig wenig Energie verbrauchen (siehe Tabelle 1). Zwar können preiswerte SBCs nicht mit den Rechenleistungen von PCs mithalten; dies ist aber auch nicht zwingend erforderlich: Viele KI-Modelle können so optimiert werden, dass sie trotz begrenzter Ressourcen anspruchsvolle Aufgaben erfüllen können. Durch die geschickte Anpassung der KI-Modelle an die begrenzten Ressourcen von eingebetteten Systemen können diese mit geringem Energieverbrauch und minimaler Datenverarbeitung effektiv arbeiten. So lässt sich ein leistungsfähiges KI-System mit Ressourceneffizienz und Datensparsamkeit kombinieren.

### Energie- und Ressourceneffizienz

Aus Tabelle 1 ist ersichtlich, dass man für den Preis eines PCs rund 50 Arduino-Boards bekommt - die zusammen etwa den halben Stromverbrauch des PCs haben. Gerade eine geringe Stromaufnahme ist ein wichtiges Kriterium für Geräte und Systeme, die aus dezentralen Energiequellen wie Solarzellen, Batterien oder Akkus versorgt werden. Entscheidend dabei ist der sparsame Umgang mit Daten, und zwar aus folgenden Gründen:

- **Beschränkte Ressourcen:** Eingebettete Systeme sind sparsam ausgestattet und daher in aller Regel nicht in der Lage, große Datenmengen zu speichern oder zu verarbeiten. Entsprechend kommen hier ML-Modelle zum Einsatz, welche selbst wenig Speicherplatz benötigen und mit vergleichsweise kleinen Datensätzen effektiv arbeiten können.
- Hier sei angemerkt, dass für eine effiziente Zustandsüberwachung teilweise ein sehr tiefes Systemverständnis erforderlich ist, um die relevanten Daten zu identifizieren und die für eine sinnvolle Auswertung minimal erforderliche Datenrate festzulegen.
- **Energieeffizienz:** Häufig arbeiten eingebettete Systeme in Umgebungen, in denen Energie eine begrenzte Ressource ist. Das Verarbeiten großer Datenmengen erfordert mehr Rechenleistung und damit mehr Energie. Entsprechend führen daten- und rechensparsame Modelle zu einem reduzierten Energieverbrauch.
- **Datenschutz:** Nicht selten erfassen eingebettete Systeme persönliche oder in anderer Weise sensible Daten. Datensparsamkeit kann dazu beitragen, die Menge der gesammelten und gespeicherten Daten zu minimieren und lokal zu halten und so die Privatsphäre der Nutzer zu schützen.

### KI-Modelle lassen sich komprimieren

Wie erreicht man im KI-Umfeld einen sparsamen Umgang mit Daten?

Zum einen durch die Komprimierung des Modells. Dazu kommen etwa folgende Techniken zum Einsatz:

- **Quantisierung:** Codieren der Parameter bzw. Gewichte in niedrigerer Präzision. Etwa, indem diese nicht - wie während des Trainings - als 32-Bit- oder 64-Bit-Gleitkommazahlen dargestellt werden, sondern z. B. als Festkommazahlen oder ganzzahlig mit entsprechender Skalierung. Denn gerade für kleine Mikrocontroller-Architekturen, wie etwa Cortex M3, die nicht über eine Gleitkomma-Recheneinheit verfügen, stellen solche mathematischen Operationen einen hohen Rechenaufwand mit entsprechendem Energiebedarf dar.
- **Pruning:** Bei diesem Verfahren sucht man gezielt nach unwichtigen Gewichtungen. Werden solche gefunden, werden sie entfernt; teilweise lassen sich auch komplette Feature-Kanäle entfernen (z. B. extrem seltene Ereignisse). Anschließend trainiert der Entwickler das Modell neu, um den vom Pruning verursachten Genauigkeitsabfall zu kompensieren.
- **Wissensdestillation:** Hier versucht man, das Wissen eines komplexen, tief strukturierten Modells möglichst ohne Verlust der Gültigkeit auf ein kleineres Modell zu skalieren. Hintergrund dafür ist, dass das umfangreiche Modell zwar sehr viel mehr Details verarbeitet, dieses im konkreten Fall aber nicht benötigt wird. Entsprechend kann das kompaktere

Modell auf eine weniger leistungsfähige Hardware-Plattform portiert werden.

### Edge-Computing

Ein weiterer Ansatz zum Sparen von Daten und Energie ist Edge-Computing: Anstatt Rohdaten zur Verarbeitung an einen zentralen Server zu senden, findet die Datenverarbeitung direkt auf dem eingebetteten System statt. Da weniger Daten (ggf. drahtlos) übertragen werden müssen, sinkt der Energieaufwand für die Datenübertragung. Gleichzeitig gewinnt man an Sicherheit und garantiert die Privatsphäre (siehe oben).

### KI-Integration in Mikrocontrollern und Einplatinenrechnern

Der Raspberry Pi (siehe Tabelle 1) ist ein Beispiel für einen kostengünstigen, kompakten Einplatinenrechner, der sich zunehmend auch in industriellen Applikationen findet. Diese SBCs lassen sich im KI-Umfeld auch gut als Knoten in einem verteilten KI-System nutzen. Jeder Knoten führt dabei individuelle Aufgaben aus und trägt zur Gesamtleistung des Systems bei. Die Programmierung erfolgt häufig in Python (siehe Kasten).

### KI-Rechenwerke

Mittlerweile gibt es erste Mikrocontroller auf dem Markt, welche nicht nur CPUs beinhalten, sondern auch spezifische KI-Rechenwerke. Mit solchen Bausteinen lassen sich leistungsfähige, verteilte KI-Sys-

teme mit geringstem Energieverbrauch umsetzen. Ein Beispiel ist der MAX78000, der dafür entwickelt wurde, neuronale Netzwerke auszuführen [5]. Seine CNN-Beschleunigungseinheit [6] ermöglicht es, KI-Inferenzen [7] mit nur Mikrojoule an Energieverbrauch durchzuführen. Tabelle 2 vergleicht wesentliche Eigenschaften dieses Mikrocontrollers mit einem Raspberry Pi.

Der MAX78000 ist als System-on-a-Chip (SoC) konzipiert und kann Updates seines neuronalen Netzwerks im laufenden Betrieb vornehmen. Dies macht ihn zu einem flexiblen und gut anpassbaren Werkzeug für eingebettete KI-Anwendungen. In der Praxis werden die neuronalen Netzwerke zunächst in gängigen Tools, wie etwa PYTorch (auf dem PC) trainiert, bevor sie mit den von Maxim bereitgestellten Werkzeugen auf den Mikrocontroller portiert werden.

## Ausblick

Vorbeugende Instandhaltung und Zustandsüberwachung sind gute Einsteiger-Applikationen für den Einsatz von KI im Mittelstand. Mittels ML-Verfahren lassen sich z. B. schleichende oder typische Veränderungen in Strom-Zeit-Profilen erkennen, ohne dass es zusätzlicher Sensoren etc. bedarf. Für Einsteiger ist die große Community bei den verbreiteten SBCs ein nicht zu unterschätzender Einsatzbeschleuniger. Für fortgeschrittene Anwender macht es Sinn, sich mit den Möglichkeiten spezieller KI-Mikrocontroller vertraut zu machen. Generell empfehlen die Autoren, mit smarten Ansätzen zu arbeiten und damit mit kleinen, energie- und ressourceneffizienten Lösungen. Für das Erkennen von sich abzeichnenden Schäden durch Verschleiß, blockierter Lüfter oder fehlerhafter Druckvorgänge ist sicherlich kein High-End-IPC als Recheneinheit erforderlich.

## Wer schreibt:

Andreas Federl ist wissenschaftlicher Mitarbeiter sowie Doktorand im Labor für hardwarenahe Digitalisierung an der Fakultät Elektrotechnik und Medientechnik (EMT) der Technischen Hochschule Degendorf (THD). Schwerpunkt seiner Arbeit ist die Forschung im Bereich der Stromversorgungstechnik, Hardware für Embedded Sys-

## Vergleich Raspberry Pi / MAX 78000FTHR

	Raspberry Pi	MAX 78000FTHR
Weit verbreitet	+++	--
Große Community	+++	--
Vielfältige Anschlussmöglichkeiten	++	+
Rechenleistung / Speicher	+	-
KI-Rechenleistung	+	++
KI-Flexibilität [10]	-	+++
Energieeffizienz / geeignet für mobile Applikationen	-	++
Einfache Programmierung	+	-
Spezialisierte KI-Hardware	-	+++

Tabelle 2: Vergleich Raspberry Pi / MAX 78000FTHR

tems und Anwendung von künstlicher Intelligenz im Bereich der Elektrotechnik.

Markus Böhmisches ist Entwicklungingenieur bei der Elec-Con technology GmbH. Er ist Spezialist für digital geregelte DC/DC-Wandler. Ein weiterer Schwerpunkt seiner Arbeit sind Simulation sowie die EMV-Optimierung von kundenspezifischer (Leistungs-)elektronik.

## Referenzen

- [1] Quelle: Forschungsbeirat der Plattform Industrie 4.0 (acatech)
- [2] Nach Ansicht des TÜV Nord Cert sind "Sensoren in der Automation derzeit das gefährlichste Einfallstor für Hacker".

- [3] <https://www.nist.gov/publications/key-elements-contextualize-ai-driven-condition-monitoring-systems-towards-their-risk>
- [4] SBC: Single Board-Computer oder Einplatinenrechner. Diese enthalten auf einer Leiterplatte einen voll funktionsfähigen Rechner mit Mikroprozessor, Arbeitsspeicher, Ein-/Ausgabe-Bausteinen (IOs), gelegentlich auch Massenspeicher.
- [5] <https://www.stg-maximintegrated.com/en/products/microcontrollers/MAX78000.html#online-ds>
- [6] CNN ist die Abkürzung von Convolutional Neural Network, zu Deutsch etwa „faltendes neuronales Netzwerk“. Dies ist ein

künstliches neuronales Netz. Sein Konzept ist von biologischen Prozessen inspiriert und eignet sich insbesondere für maschinelles Lernen.

- [7] Eine Inferenz ist bei einem neuronalen Netzwerk ein einzelner schlussfolgernder Denkvorgang bzw. ein Prozess, bei dem aus gegebenen Eingangsdaten und gelerntem Wissen Schlüsse gezogen werden.
- [8] bei zehn Jahren Dauerbetrieb
- [9] bei Stromkosten von 25 ct/kWh
- [10] Der MAX78000 unterstützt verschiedene Gewichtsformate für neuronale Netze und erlaubt Aktualisierungen der KI-Netzwerke im laufenden Betrieb ◀

## Python

Die weit verbreitete und leicht zugängliche Programmiersprache Python ist zu einer bevorzugten Sprache für die Entwicklung von KI-Anwendungen geworden. Beispielsweise kann die Modellbildung mithilfe der freien Software-Bibliothek Scikit-Learn erfolgen, in welcher zahlreiche Modelle aus dem Bereich des maschinellen Lernens zur Verfügung stehen. Weitere Vorteile von Python sind:

- Die einfache und intuitiv lesbare Syntax erleichtert den schnellen Einstieg und ermöglicht erfahrenen Programmierern, sich auf KI-Algorithmen und -Konzepte zu fokussieren.
- Python bietet eine breite Palette von Bibliotheken und Frameworks, die speziell für KI entwi-

ckelt wurden. Die bekanntesten sind NumPy, Pandas, TensorFlow, Keras, PyTorch und Scikit-Learn. Diese Bibliotheken bieten vorgefertigte Funktionen und Algorithmen für Aufgaben wie maschinelles Lernen, neuronale Netze, Datenanalyse und -manipulation, Bildverarbeitung etc.

- Es gibt eine große und weltweit aktive Community, welche ständig neue Bibliotheken, Frameworks und Tools für KI entwickelt und verbessert. Dadurch stehen Anwendern eine Fülle von Ressourcen, Tutorials, Dokumentationen und Beispielcodes zur Verfügung. Die Zusammenarbeit und der Austausch von Wissen in der Python-Community sind auch für KI-Entwickler äußerst wertvoll.

- Python ist plattformunabhängig; Programme in Python sind auf unterschiedlichsten Betriebssystemen wie Windows, macOS oder Linux lauffähig. Diese Flexibilität ist besonders wichtig, da KI-Anwendungen auf verschiedenen Plattformen und Geräten implementiert werden müssen, von Servern bis hin zu eingebetteten Systemen.
- Nahtlose Integration von anderen Programmiersprachen und Tools: Im Bedarfsfall kann auf leistungsstarke Bibliotheken und Frameworks aus anderen Sprachen zurückgegriffen werden. So lassen sich komplexe Berechnungen in Python dadurch beschleunigen, dass diese als optimierte Funktionen in C oder Fortran programmiert werden.