Lokale Sensorintelligenz per KI

Das Internet der Dinge (IoT) löste den Boom für drahtlose Sensoren aus. Nun kommt der lokale Einsatz der künstlichen Intelligenz hinzu. Dadurch verlagert sich der Schwerpunkt der IoT-Anwendungsentwicklung weg von der Cloud in Richtung Sensorik. Der damit verbundene Einsatz von Maschine Learning (ML)-Algorithmen direkt im Sensor oder in unmittelbarer Nähe der Sensorik ermöglicht neue Anwendungen.

Die Problematik

Es gibt zahlreiche Gründe, warum ein permanenter Sensor-to-Cloud-Datenstrom für viele Anwendungen keine gute Idee ist. Immer größere Datenmengen, unzureichende Bandbreite, lange Latenzzeiten, Datenschutzprobleme und relativ hohe Betriebskosten sind nur einige Beispiele. Trotz allem hat sich die Cloud in den vergangenen Jahren als Datenverarbeitungsplattform für IoT-Sensordaten etabliert, weil dort hochentwickelte, bedienerfreundliche und gut dokumentierte Werkzeuge und Möglichkeiten zur Datenverarbeitung existieren. Neben den Machine Learning-Funktionen aus dem Bereich der künstlichen Intelligenz (AI: Artificial Intelligence) sind ortsunabhängige Benutzerzugriffe auf die in der Cloud gespeicherten oder dort erzeugten Daten besonders wichtig. Des Weiteren bietet eine Cloud nahezu unbegrenzte Ressourcen in Bezug auf Speicherkapazität und Rechenleistung (z. B. für das Erzeugen von komplexen Machine-Learning-Modellen), die darüber hinaus sehr professionell betreut werden und sogar Lastschwankungen automatisch ausbalancieren können.

Lokaler Einsatz

Der lokale Einsatz von ML-Algorithmen für Edge-AI-Sensoren verfolgt einen völlig anderen Lösungsansatz: Sensordaten sollen direkt vor Ort verarbeitet werden, um (vielfach sogar in Echtzeit) die jeweils erforderlichen Informationen zu erzeugen und zur automatischen Entscheidungsfindung zu nutzen. Eine solche Edge-AI-Lösung unterscheidet sich allerdings vom Edge Computing, das vielfach auch als Edge-Cloud-Einsatz bezeichnet wird. Hier geht es in der Regel darum, einfach Cloud-Dienste auf entsprechend leistungsfähiger Hardware direkt On-Prem vor Ort zu nutzen. Teilweise werden dabei sogar die gleichen Softwareschnittstellen (APIs) wie in den jeweiligen Cloud-Anwendungen verwendet. Edge Computing ist in erster Linie ein Hardware- bzw. Plattformthema. Edge-AI-Anwendungen verfolgen eine etwas andere Zielsetzung. Es geht primär darum, leistungsfähige KI-Algorithmen auch im kleinsten Sensor-Mikrocontroller oder in unmittelbarer Nähe der Sensoren auf einem Gateway zur Echtzeitdatenanalyse und automatischen Entscheidungsfindung nutzen zu können. Für solche Lösungen müssen Hardware und Software sorgfältig aufeinander abgestimmt werden.

Hybride Lösungen

Es sind allerdings auch verschiedene hybride Lösungen möglich, in denen Edge AI-Lösungen die per Machine Learning-Algorithmen-Einsatz erzeugten Metadaten direkt an eine Cloud senden oder ein Edge-AI-Sensor als modulare Lösung eingesetzt wird (Sensorik mit Datenschnittstelle zu einem Gateway, auf dem ein zum Sensor gehörender Docker-Container mit den Sensordatenanalysealgorithmen läuft). Eine sehr dynamische Edge-AI-Entwicklung, besonders in Bezug auf Innovationen, Anwendungsvielfalt und den Bedarf an geeigneten Halbleiterchips, ist aber zunächst einmal im Bereich der drahtlosen IoT-Sensorik mit Cloud-Integration zu erwarten.

Drahtlose Sensoren mit lokaler KI

Die Systemintegration von drahtlosen Sensoren in IoT-Lösungen ist besonders in industriellen Anwendungen recht anspruchsvoll. Eine große Herausforderung ist nach wie vor der jeweilige Protokollstack. Er muss nicht nur für den Sensor selbst, sondern auch für das jeweils zum Einsatz kommende Gateway zur Verfügung stehen (Bild 1). Dabei sind neben den Funktionen und Schnittstellen auch die jeweiligen Lizenzmodelle zu beachten. Im Rahmen einer Entscheidungsfindung sollten sogar die hinter dem Code stehenden Entwickler bewertet werden. Schließlich erfordert eine industrielle Funksensorlösung auch für die Protokollsoftware qualifizierten Support für die kommenden 10 bis 15 Jahre. Eine Funkfirmware als Binärobjekt (Blob) aus einer intransparenten Quelle, wie für Wi-Fi- oder BLE-Implementierungen im Consumer-Bereich üblich, ist nicht in jedem Fall eine gute Wahl.

IT- bzw. IoT-Security

Ein weiterer wichtiger Themenkomplex für die Wireless-Sensorik einer Edge-AI-Lösung ist die IT- bzw. IoT-Security. Dazu gehören zunächst einmal die Authentizi-

Autor:
Klaus-Dieter Walter, CEO
SSV Software Systems GmbH
https://ssv-embedded.de

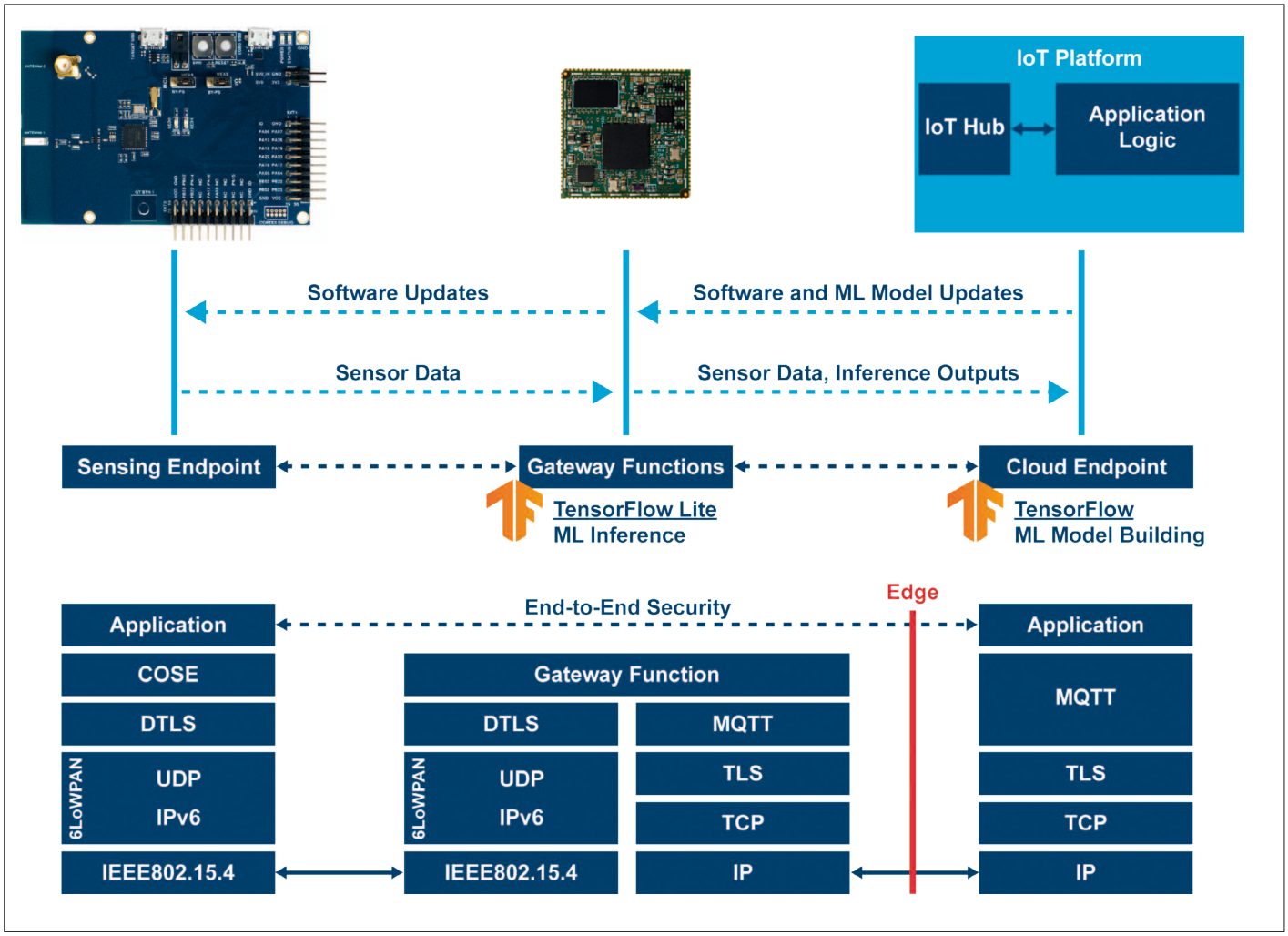

Bild 1: Viele IoT-Anwendungen bestehen aus drei elementaren Funktionsblöcken: Der Sensorik, den Gateway-Funktionen sowie einer Cloud, z. B. als IoT-Plattform mit verschiedenen Diensten. Sensoren und Gateway waren bisher in erster Linie als Datenlieferant für die Cloud konzipiert. Nun kommt durch die lokale Sensorintelligenz, also die Edge-AI-Komponente, eine weitere anspruchsvolle Aufgabe hinzu: dazu gehört die Echtzeitsensordatenanalyse direkt vor Ort an der Sensordatenquelle (beispielsweise als Machine-Learning-Inferenz per TensorFlow Lite).

tät, Vertraulichkeit und Integrität der Sensordaten. Eine IoT-Anwendung am anderen Ende benötigt die 100-%ige Sicherheit, dass die erhaltenen Daten auch tatsächlich von dem Sensor stammen, dessen Absenderkennung als Quellangabe in den Sensordaten zu finden ist und dass diese Daten auf dem Übertragungsweg nicht verändert wurden. Bei der erforderlichen Ende-zu-Ende-Sicherheit wird häufig übersehen, dass die Sicherheitsmechanismen eines von MQTT- oder HTTPS-Verbindungen genutzten TLS-Protokolls in der Praxis architekturbedingt an mehr als einem Punkt „aufgebrochen" werden. Dadurch existieren mehrere Bereiche zwischen Sensor und Anwendung, in denen Sensordatenmanipulation oder andere unerwünschte Eingriffe erfolgen können. Um derartige Risiken weitestgehend auszuschalten, ist aber nicht gleich eine Blockchain erforderlich. In der Praxis reicht es aus, wenn die Sensordaten direkt an der Quelle oder in unmittelbarer Nähe mit einer digitalen Signatur versehen werden, die von der Anwendung am anderen Ende in jedem Fall überprüft wird.

Updates Over-The-Air (OTA)

Ein weiteres Security-Thema sind die technischen und organisatorischen Voraussetzungen für Over-The-Air (OTA) Updates. Ein unidirektionaler Kommunikationspfad zur Messdatenübertragung vom Sensor bis in die Cloud reicht somit nicht mehr aus. In Gegenrichtung, also von der Cloud bis in den Sensor müssen von Zeit zu Zeit bzw. bei Bedarf Software-Updates verschickt werden. Das ist wegen der Gefahren einer missbräuchlichen Nutzung eine hochsensible Aufgabenstellung. Aber auch Zuverlässigkeitsaspekte sind hier zu beachten: bricht die Funkdatenübertragung zwischen Sensor und Gateway während eines Update-Vorgangs ab, muss der IoT-Sensor auf jeden Fall noch die „alte" Softwareversion nutzen können. Dafür sollte der interne

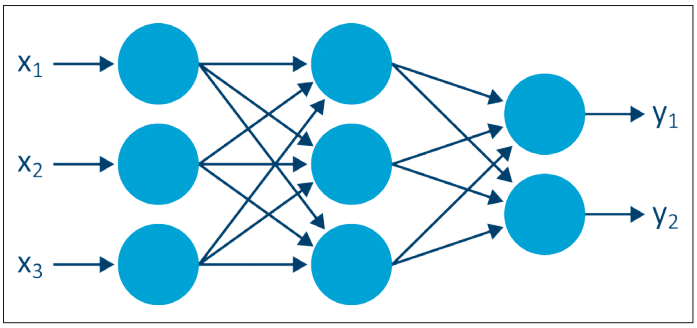

Bild 2: Ein ML-Modell für Sensordaten lässt sich durch ein neuronales Netzwerk realisieren. Es arbeitet als Mapping-Funktion und verknüpft die Eingangsparameter X per Regressions- oder Klassifizierungsalgorithmen mit den Ausgangswerten Y. Die Machine-Learning-Modellbildung erfolgt in einer speziellen Lernphase mit Hilfe geeigneter Trainingsdaten, die vorab in einer Datei erfasst werden und deren Merkmalsvektorformat die Anzahl der Eingänge des neuronalen Netzwerks bestimmt (siehe Beitrag unter [1]).

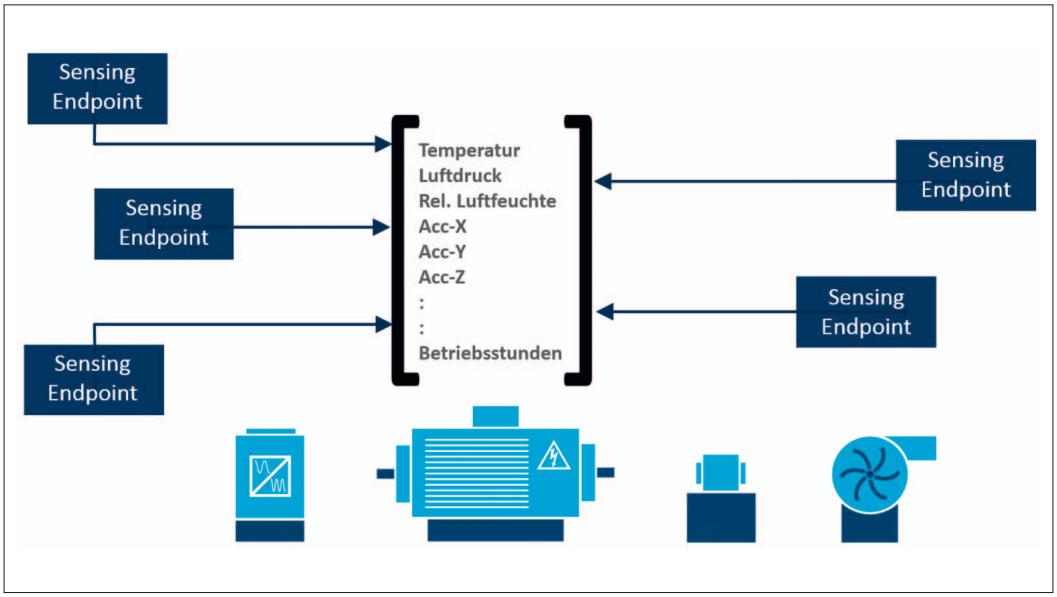

Bild 3: Typische Anwendungen für drahtlose Sensoren plus Machine Learning-Algorithmen sind Zustandsüberwachungen aller Art (Condition Monitoring). Dafür wird eine Umgebung (z. B. eine Maschine, Anlage oder die gesamte Smart Factory) mit möglichst vielen Sensoren ausgestattet, die zusammen jederzeit ein qualitativ hochwertiges Datenabbild der Realität liefern. Datentechnisch werden die Messwerte aller Sensoren zum Zeitpunkt t in einem Merkmalsvektor zusammengefasst und durch ein neuronales Netzwerk klassifiziert.

Flash des Sensormikrocontrollers in einen A- und B-Bereich aufgeteilt werden. (A/B-Boot- bzw. Update-Konzept: gestartet wird die Software jeweils aus einem Bereich. Das Update erfolgt in den jeweils anderen Bereich. Erst wenn ein Software-Update vollständig und ohne Fehler durchgelaufen ist, werden die Bereiche gewechselt und ein Neustart der Mikrocontrollersoftware ausgelöst, um die neue Softwareversion zu starten.)

In die Kommunikationsbeziehungen für Software-Updates sind die Gateway-Funktionen einer Anwendung eingebunden. Sie dienen nicht nur als OTA-Update-Proxy für das Sensor-Update, sondern benötigen auch selbst Funktions- und Sicherheits-Updates aus der Cloud. Durch den Einsatz von ML-Algorithmen zur Sensordatenanalyse in Echtzeit innerhalb der Gateway-Funktionen werden auch Updates für das dabei zum Einsatz kommende Machine-Learning-Modell benötigt (ML Model Updates). Solche Edge-AI-Lösungen nutzen Modelle, die durch eine Machine-Learning-Trainingsphase (ML Model Building) in der Cloud entstehen.

Sensor plus ML-Modell

Die künstliche Intelligenz einer Edge-AI-IoT-Applikation wird durch Supervised Machine Learning-Methoden gebildet. Zum Sensor gehört dann jeweils ein ML-Modell für die Inferenzphase (beispielsweise eine Echtzeitdatenanalyse mittels einer zuvor erlernten und zur jeweiligen Aufgabenstellung passenden Mustererkennung). Den dafür zum Einsatz kommenden ML-Algorithmus kann man sich als Mapping-Funktion vorstellen: zu den jeweiligen Eingangsparametern X werden über ein mathematisches Regressions- oder Klassifizierungsverfahren die passenden Ausgangswerte Y geliefert. Der Zusammenhang zwischen X und Y wird von der Mapping-Funktion aus zuvor erfassten Trainingsdaten erlernt (siehe Beitrag „Sensordaten zur KI-Wissensbildung“ in PC&Industrie 3-2020 [1]). Das mathematische Mapping-Verfahren lässt sich durch ein neuronales Netzwerk realisieren (siehe Bild 2).

Die ML-Modellbildung mit einem neuronalen Netzwerk ist auf Grund der zahlreichen Iterationen ein rechenintensiver Vorgang, der mit einer ausreichend großen Menge an Trainingsdaten erfolgen sollte. Darüber hinaus benötigt ein ML-Modell hin und wieder einige weitere Lernphasen, um auf Veränderungen in der Inferenzumgebung zu reagieren und die Inferenzfehlerquote zu optimieren. Des Weiteren muss die aktuelle Modellversion allen Sensing-Endpunkten einer Edge-AI-Anwendung zentral zur Verfügung stehen. Durch diese unterschiedlichen Anforderungen ist eine Cloud oder ein gleichwertiger On-Prem-Service in den meisten Fällen der beste Ort für Modellbildung, Modellpflege und Modellspeicherung.

Anwendungsbeispiel Condition Monitoring

Typische Anwendungen für drahtlose Sensoren plus Machine Learning-Algorithmen sind Zustandsüberwachungen aller Art (Condition Monitoring). Dafür wird eine Umgebung (z. B. eine Maschine, Anlage oder die gesamte Smart Factory) mit möglichst vielen Sensoren ausgestattet, die zusammen jederzeit ein qualitativ hochwertiges Datenabbild der Realität liefern. Datentechnisch werden die Messwerte aller Sensoren zum Zeitpunkt t in einem Merkmalsvektor zusammengefasst und durch ein neuronales Netzwerk klassifiziert.

Hierzu ein Beispiel: Der Betreiber eines Maschinenparks möchte für eine größere Werkzeugmaschine eine Echtzeitinformation an eine MES-Software übermittelt bekommen, aus der hervorgeht, in welchem der folgenden Zustände sich die Maschine jeweils befindet:

1) Maschine wird nicht benötigt (Versorgungsspannung = Aus).
2) Maschine ist im Standby-Zustand (Versorgungsspannung = Ein).
3) Maschine ist aktiv und produziert.
4) Es liegt eine Rüstphase vor.
5) Geplanter Halt, die Maschine wartet auf etwas.
6) Ungeplanter Halt, Maschine ist defekt (z. B. Not-Aus-Schalter betätigt).
7) Maschine zeigt ein auffälliges Betriebsverhalten (z. B. eine Unwucht).

Um diese sieben Zustände möglichst genau voneinander unterscheiden zu können, sind verschiedene Sensoren erforderlich. Die Zustände 3) und 7) sind mit Hilfe einer mehrdimensionalen Vibrations- bzw. Stromsensorik ausreichend sicher bestimmbar. Aus den Messwerten dieser Sensoren lässt sich auch ableiten, ob ein Zustand ungleich 3) oder 7) vorliegt, also irgendwas aus dem Bereich 1), 2), 4), 5) oder 6). Um zwischen den Zuständen 1), 2), 4), 5) und 6) differenzieren zu können, sind weitere Sensoren erforderlich. Evtl. kann man 1), 2) und 6) über einen optischen Farbsensor bestimmen, der an der Maschinenampel befestigt wird und RGB-Farbwerte bestimmen kann (ist die Versorgungsspannung der Maschine ausgeschaltet, ist auch die Maschinenampel inaktiv). In der Rüstphase 4) ist die Tür zum Maschineninnenraum geöffnet, evtl. befindet sich sogar eine Person im inneren Arbeitsbereich der Maschine. Um den Zustand der Tür zum Arbeitsbereich zu bestimmen, wäre ein binärer Türsensor erforderlich. Der Zustand 5) ist dann jeweils der inaktive Wartezustand ohne Rüstphase und Not-Aus (Maschinenampel nicht Rot, aber aktiv) usw. ◄

Externe Quellenangabe

[1] Download Link zu „Sensordaten zur KI-Wissensbildung“ in PCIndustrie 3-2020:
https://www.beam-verlag.de/fachartikel-aus-pc-industrie/sensoren/