PLATTFORM INDUSTRIE 4.0

ERGEBNISPAPIER

Umgang mit Sicherheitsrisiken industrieller Anwendungen durch mangelnde Erklärbarkeit von KI-Ergebnissen

Impressum

Herausgeber
Bundesministerium für Wirtschaft
und Energie (BMWi)
Öffentlichkeitsarbeit
11019 Berlin
www.bmwi.de

Redaktionelle Verantwortung
Plattform Industrie 4.0
Bertolt-Brecht-Platz 3
10117 Berlin

Gestaltung
PRpetuum GmbH, 80801 München

Stand
September 2019

Druck
BMWi

Bildnachweis
gettyimages
Degui Adil / EyeEm / S. 13
MEHAU KULYK/SCIENCE PHOTO LIBRARY / S. 15
Photographer is my life. / S. 4
seksan Mongkhonkhamsao / S. 8
Suebsiri Srithanyarat / EyeEm / S. 22
oxygen / S. 14
Yuichiro Chino / Titel, S. 10, S. 17

Diese und weitere Broschüren erhalten Sie bei:
Bundesministerium für Wirtschaft und Energie
Referat Öffentlichkeitsarbeit
E-Mail: publikationen@bundesregierung.de
www.bmwi.de

Zentraler Bestellservice:
Telefon: 030 182722721
Bestellfax: 030 18102722721

Diese Publikation wird vom Bundesministerium für Wirtschaft und Energie im Rahmen der Öffentlichkeitsarbeit herausgegeben. Die Publikation wird kostenlos abgegeben und ist nicht zum Verkauf bestimmt. Sie darf weder von Parteien noch von Wahlwerbern oder Wahlhelfern während eines Wahlkampfes zum Zwecke der Wahlwerbung verwendet werden. Dies gilt für Bundestags-, Landtags- und Kommunalwahlen sowie für Wahlen zum Europäischen Parlament.

Inhalt

1. Einleitung ... **3**

2. Entwurf, Training und vortrainierte Systeme ... **4**

2.1 KI-Anwendungen zum Schutz der Systeme ... 7

2.2 Über Security-Angriffe gegen AI-Anwendungen ... 8

3. Erklärbarkeit von KI-Entscheidungen ... **10**

3.1 Notwendigkeit von Erklärungen durch menschliche Argumentation ... 11

3.2 Komplexität des Erklärungsproblems ... 11

3.3 Forschungsansätze ... 12

3.4 Vorhandene Techniken ... 12

3.5 Beispiele ... 12

4. Angriffsvektoren auf KI bzw. mittels KI ... **14**

5. Sicherheitsrisiken bedingt durch Architekturen und Anwendungen ... **17**

5.1 Sicherheitsrisiken beim Einsatz von KI-Systemen ... 21
5.1.1 Folgerisiken: Produktionsausfälle und entsprechende Zusatzkosten ... 24
5.1.2 Sicherheitsrisiken beim Einsatz statischer KI-Systeme ... 24
5.1.3 Sicherheitsrisiken beim Einsatz dynamischer KI-Systeme ... 25

5.2 Sicherheitsmaßnahmen beim Einsatz von KI/Sicherheitsmaßnahmen durch KI ... 26

6. Abschließende Bemerkungen ... **27**

7. Grundbegriffe der KI ... **28**

8. Referenzen ... **30**

Abbildungsverzeichnis

Abbildung 1: Highlevel Phasenmodell ... 6

Abbildung 2: Training und vortrainierte Systeme zur Bilderkennung ... 7

Abbildung 3: Vier Quadranten-Darstellung der Freiheitsgrade einer KI-Anwendung ... 18

1. Einleitung

KI kann in verschiedenen Bereichen industrieller Entwicklung, Produktion und Fertigung sinnvoll eingesetzt werden. Dazu gehören

- KI in der **Produktionsplanung**, z. B. durch Sequenzieren
- KI in der **Produktion**, z. B. zur Überwachung von Zuständen
- KI in der **Prozessautomatisierung**, z. B. bei der Steuerung von Robotern
- KI in der **Qualitätssicherung**, z. B. zur optischen Bauteileprüfung
- KI in der **Logistik**, z. B. bei wegeoptimierten Abläufen

Um KI in der Industrie allerdings sicher und erfolgreich zum Einsatz zu bringen, sind einige Voraussetzungen zu erfüllen. Hierzu zählt: 1. Das Verständnis der Ziele, die KI erreichen oder eben nicht erreichen kann. 2. Die Kenntnis der Einflussfaktoren, die den Einsatz von KI-Systemen inhärent begleiten.

Grundlegende Aspekte zu diesem Themenfeld wurden bereits in einem Papier der Plattform Industrie 4.0 unter dem Titel „Künstliche Intelligenz (KI) in Sicherheitsaspekten der Industrie 4.0“ beschrieben und zur Hannover Messe Industrie 2019 veröffentlicht. Die vorliegende Publikation geht darüber hinaus, indem sie die Bedeutung von Erklärbarkeit der KI für Sicherheitsaspekte darstellt. Es geht dabei vorrangig um die Frage, wie menschliche Betrachter die Entscheidungen eines KI-Systems verstehen und etwaige verborgene Fehler in Architektur, Konfiguration und Training aufspüren können, um sie zu korrigieren: Was hat das System tatsächlich „gelernt“ und welche Einflussfaktoren waren in diesem Prozess ausschlaggebend?

Moderne Systeme Künstlicher Intelligenz (KI) entfernen sich in ihrer Funktionsweise immer weiter von den Vorstellungen, die bislang über die Funktion biologischer Gehirne bestanden. Beispielsweise erkennen Künstliche Neuronale Netze heute Muster in Bild, Ton, Text oder Video mit einer dramatisch höheren Zahl von Verschachtelungen in Neuronenschichten als es vom menschlichen Gehirn angenommen wird. Dadurch entstehen in zahlreichen Fällen übermenschliche Leistungsmerkmale im Hinblick auf Präzision und Treffsicherheit der Mustererkennung. Aber andererseits verlieren solche Systeme die Interpretierbarkeit mit menschlichem Erklärungsvermögen. Präzision und menschliche Nachvollziehbarkeit von Aussagen eines KI-Systems stehen zunehmend im Widerspruch.

Dieses Dokument richtet sich ausschließlich auf die Diskussion der Sicherheitsaspekte für Verfahren aus dem Bereich des Maschinellen Lernens für industrielle Anwendungen. In der heutigen Praxis sind dies vorwiegend Algorithmen des überwachten Lernens (Supervised Learning), in denen mittels Trainingsdaten eine Approximation erstellt wird, die in industriellen Prozessen zuverlässige Aussagen (Prädiktionen) erzeugen können.

Grundlegende anwendungsübergreifende Techniken werden dabei für das heute dominierende Gebiet der Künstlichen Neuronalen Netze (KNN) in Kapitel 2 in Kurzform erklärt, um Lesern auch ohne ausgeprägte KI-Kenntnis das Verständnis der industriellen Sicherheitsprobleme zu ermöglichen. Kapitel 3 behandelt das Problem der Erklärbarkeit und gibt einen kurzen Überblick der heutigen Technik. Es gibt (noch) keine einfachen Lösungen. Kapitel 4 beschreibt das derzeit wichtigste Verfahren, das Angreifer missbräuchlich verwenden, um KI gezielt zu täuschen. Auch hier existieren (noch) keine Problemlösungen ohne ausgeprägte Expertise von Spezialisten. Anwender müssen die Gefährdung jedoch verstehen, um ihre eigenen Risiken einzuschätzen. Kapitel 5 gibt umfassende praktische Ratschläge zur Risikominimierung im Einsatz von KI in der Industrie 4.0. Im Anhang in Kapitel 7 sind verwendete Grundbegriffe des Machine Learning, die im Text verwendet werden, kurz erklärt.

Der Einsatz von Künstlicher Intelligenz kann neue Sicherheitsprobleme aufwerfen. Gleichzeitig eignet sich KI in mancher Hinsicht auch als Basis neuartiger Waffen, die keineswegs nur gegen KI-basierte Systeme eingesetzt werden können, sondern eine insgesamt erweiterte Gefährdungslage bewirken. Das vorliegende Papier richtet sich an beide Seiten und liefert dazu Erklärungen und Hinweise für Betreiber.

2. Entwurf, Training und vortrainierte Systeme

Bereits vor dem Training werden beim Entwurf eines Machine Learning-Systems wesentliche Systemparameter vorausbestimmt. Wie bei einem klassischen System muss das Ein- und Ausgabeverhalten beschrieben werden, d. h. was sind Eingabeparameter (Syntax, Repräsentation, Normierung) und welches Ergebnis soll das System liefern. Gegebenenfalls ist es sinnvoll oder notwendig auf die Eingabedaten (klassische) Vorverarbeitungsschritte anzuwenden (z. B. Filterung, Diskretisierung). Durch solche vorbereitenden Schritte wird die Ergebnisqualität beabsichtigt oder unbeabsichtigt beeinflusst. Ausgehend davon wird ein geeignetes Lernmodell für das ML-System gewählt. Neben Neuronalen Netzen wurden und werden andere Modelle wie logistische Regression (Logit-Modelle), Support Vector Machines, Entscheidungsbäume, etc. genutzt. Im Zentrum der gegenwärtigen Entwicklung und dieses Textes stehen Deep Learning Modelle mit Künstlichen Neuronalen Netzen. Ein wesentlicher Teil des Entwurfs eines Neuronalen Netzes sind seine innere Struktur, d. h. die Anzahl der inneren Knoten (Neuronen), die Anzahl der Schichten, die Dichte der Vernetzung der Knoten, sowie die verwendete Transferfunktion. Obwohl diese Entscheidungen die Leistungsfähigkeit und Robustheit eines ML-Systems beeinflussen, sind sie für den späteren Anwender häufig nicht transparent. So ist beispielsweise über die verwendeten Modelle und Strukturen, auf denen ML-Funktionen großer Cloud-Service-Anbieter basieren, nur wenig Information verfügbar.

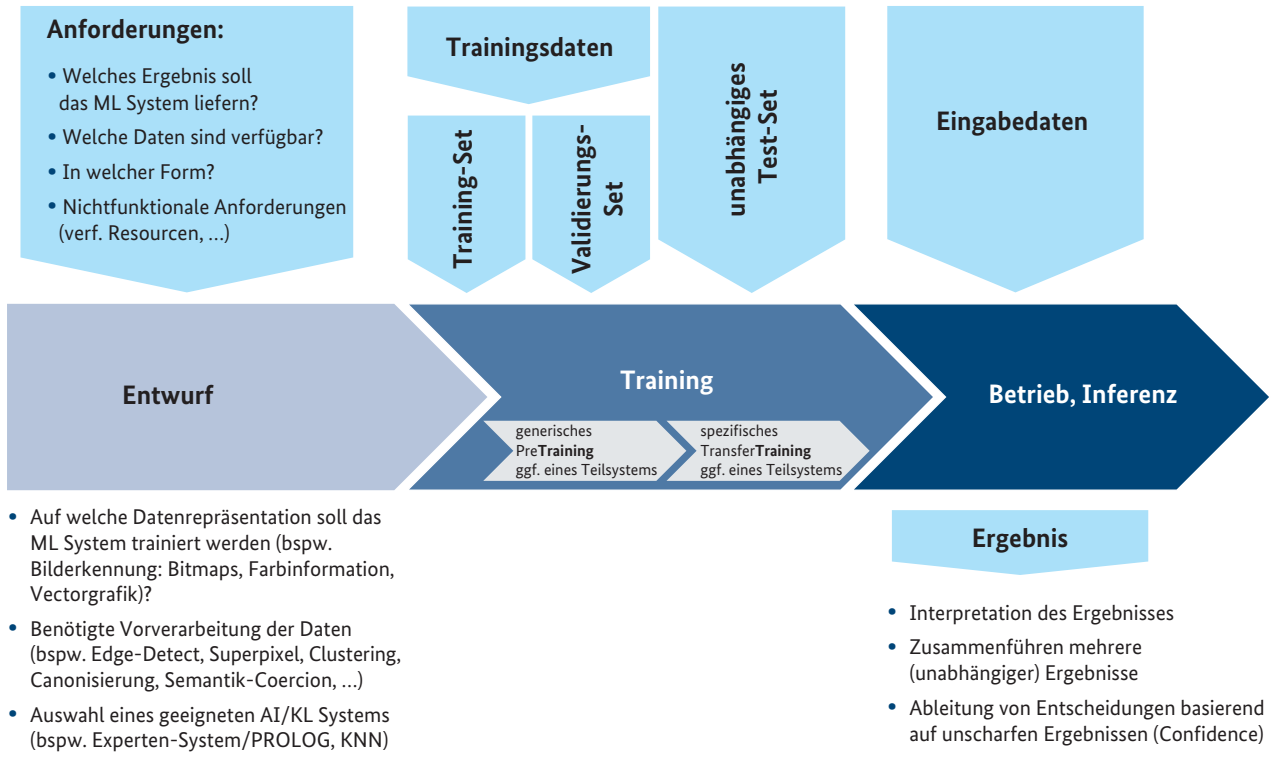
Von wenigen Ausnahmen abgesehen, benötigen alle Algorithmen des Machine Learning zur Bestimmung ihrer freien Parameter (Gewichte der jeweils eingesetzten Approximation) einen sogenanntes „Training". Dabei wird ausgehend von einer zufallsgesteuerten Initialisierung eine große Zahl von Fehlerkorrekturen dieser Parameter vorgenommen, bei denen die beobachtete Abweichung zwischen den „Labels" (Bezeichnung der Zugehörigkeit zu vorab definierten Klassen) von Daten mit den durch diese Daten erzeugten Ergebnissen verglichen werden. Dann werden die Gewichte so verändert, dass sich diese Abweichung verringert. Dieser Vorgang wird als Backpropagation bezeichnet, weil nach den Regeln der Differentialrechnung die Ableitung des Fehlers im Ergebnis rückwärts nach den Gewichten durch alle Neuronenschichten die Richtung für die Anpassung der Gewichte liefert: Der negative Gradient des Fehlers als Funktion der Gewichte zeigt in die Richtung des steilsten Abstiegs des Fehlers. Er wird also mit einer heuristisch festgelegten Schrittweite für die Gesamtheit der Gewichte zu deren Anpassung verwendet. Unter geeigneten Voraussetzungen an die Fehlerfunktion konvergiert dieser Gradientenabstieg gegen ein eindeutiges Minimum. Abbildung 1 stellt den Trainingsvorgang in einen Gesamtzusammenhang des Einsatzes von KI – von System-Entwurf über Training bis zum realen Einsatz. Dabei wird auch aufgezeigt, welche äußeren Einflussfaktoren in welcher Phase wichtig sind. Abbildung 2 versucht darauf aufbauend den Trainingsvorgang im Detail bildlich zu erklären.

Da moderne Systeme, insbesondere Neuronale Netze, sehr große Zahlen freier Parameter besitzen, muss ein sinnvoller Trainingsvorgang sich auf sehr große Zahlen von Trainingsdaten stützen, um diese Parameter zu bestimmen. Das bekannte AlexNet, mit dem 2012 der Siegeszug der CNN (Convolutional Neural Networks – Faltungsnetze) in der Bilderkennung begann, hatte rund 65 Millionen Gewichte zur Unterscheidung von 1.000 Kategorien, und es standen 1.000.000 Beispiele mit Labels zur Verfügung. Nach üblichen Vorstellungen über Statistik sind das offensichtlich viel zu wenige Beispiele. Deshalb sind diverse Techniken entwickelt worden, mit denen die Zahl der Beispiele erhöht werden kann (Data-Augmentation). Beispielsweise kann man Bilder um einige Pixel nach oben, unten oder seitwärts verschieben, ohne dass sich der inhaltliche Bezug zum Label ändert. Es entsteht aber schnell die Frage, wo die Grenze für diese Vervielfachung liegt und ab wann die Qualität des erworbenen Wissens leidet.

Die Trainingsdaten werden für jeden Schritt des Gradientenabstiegs in sogenannten Mini-Batches gebündelt, der Gradient wird also für den durchschnittlichen Fehler ermittelt und die Zusammensetzung der Batches erfolgt zufällig (siehe Abb. 2). Als Epoche wird der Teil des Trainings bezeichnet, in dem alle vorhandenen Daten einmal verwendet wurden. Danach wird der Vorgang wiederholt und nach jeder Epoche wird mit einem Teil der Daten, die vorher von den Trainingsdaten abgespalten wurden, eine Erfolgskontrolle durchgeführt.

Sobald sich dabei erkennen lässt, dass die ermittelte Genauigkeit auf diesen Validierungsdaten nicht weiter steigt, wird das Training beendet. Stagnation der Genauigkeit im Validierungsset ist ein Indikator für beginnendes Overfitting (siehe Erklärung in der Fachwortsammlung, Kapitel 7) – ab jetzt würde weiteres Training die Fähigkeit des Systems zur Verallgemeinerung der Klassifikation auf unbekannte Daten verschlechtern. Eine endgültige Prüfung anhand der ebenfalls vorher abgespaltenen Testdaten, die das System also auch in der Validierung noch nie sah, schließt das Training für die gewählten Hyperparameter (Schrittweiten, Normierungen, Fixierungen, u. ä.) ab. Weitere Trainingsläufe werden dann mit anderen Parametersätzen durchgeführt, bis ein akzeptables Ergebnis vorliegt.

Abbildung 1: Highlevel Phasenmodell

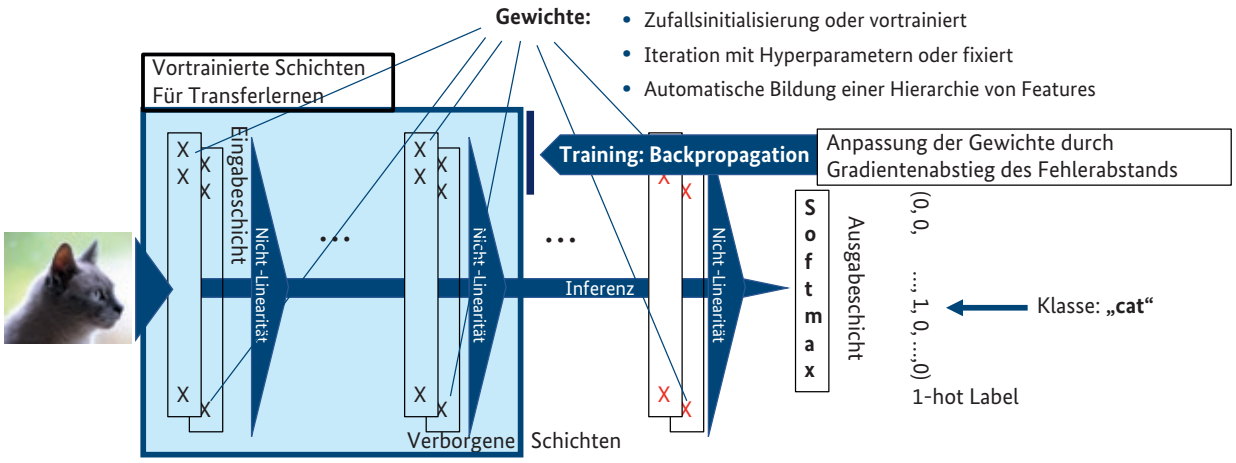


Quelle: Plattform Industrie 4.0

Vor über zehn Jahren entstand bereits die Technik des „Transfer-Lernens“ (transfer learning, Begriffserklärung in der Fachwortsammlung), mit der sich der chronische Mangel an Daten in vielen Situationen beheben lässt. Statt alle Gewichte eines KNN von Grund auf neu zu bestimmen, werden dabei die Gewichte der unteren Schichten eines bereits fertig trainierten KNN wiederverwendet. Nur die oberen Schichten werden für neu definierte Klassen neu bestimmt. Es ist heute sehr beliebt, derartige „vortrainierten“ Netze zu verwenden. Transfer-Lernen ist viel schneller, da ja nur ein Bruchteil der Gewichte neu trainiert werden muss, und vor allem reichen sehr viel weniger Daten aus, um hohe Trefferraten im Abschlusstest zu erreichen.

Im Open Source Bereich ist eine große Zahl vortrainierter Netze verfügbar geworden. Besonders wenn es darum geht, komplexe Inhalte zu erkennen, die komplexe Parameterstrukturen erfordern, aber gleichzeitig nur sehr wenig Beispielbilder vorhanden sind, kann Transfer-Lernen die Lösung sein. Ein typisches industrielles Beispiel ist die Qualitätskontrolle mit zerstörungsfreier Materialprüfung. Sofern komplexe Fehlerbilder in komplexen Werkstücken erkannt werden müssen, könnte es Jahrzehnte dauern, bis genügend Fehlerbilder vorliegen, um ein neuronales Netz zu trainieren. Wenn aus anderen Projekten schon vortrainierte Netze vorliegen, die mit Transfer Learning auf das Erkennen neuer Klassen trainiert werden können, ergeben sich neue Perspektiven.

Abbildung 2: Training und vortrainierte Systeme zur Bilderkennung. Andere Formen der visuellen Wahrnehmung (Szenenerkennung, Objekterkennung und Lokalisation sowie die Unterscheidung zwischen Klassifikation und Identifikation) werden in Kapitel 7 skizziert.



Daten für:

Training	Vermehrung durch Augmentierung (normalerweise viel weniger echte Daten als Gewichte im System – Statistikproblem) Ablauf in „Mini-Batches“ (wegen Stabilität der Konvergenz, typischer Umfang : 16-128 Elemente) und „Epochen“
Validierung	Kontrolle der Entwicklung der Präzision und rechtzeitiger Abbruch des Trainings, Variation der Hyperparameter
Test	Ausschließlich zur finalen Kontrolle der Präzision

Quelle: Plattform Industrie 4.0

Durch die Benutzung vortrainierter Netze entstehen Risiken, sobald nicht vollkommen transparent ist, woraus das Vortraining tatsächlich bestand. Was hat das neue Netz in seinen tiefen Schichten bereits gelernt? Welche „Biases“[1] stecken in diesen Gewichten? Kann der Lieferant, der mein ML-System implementiert, Aussagen über alle (!) Trainingsdaten machen, die mein Netz jemals gesehen hat? Gibt es Sicherheitsrisiken, mit denen mein System in seinem „früheren Leben“ vor dem Transfer schon infiziert worden ist? Solche Fragen sind Gegenstand des vorliegenden Papiers.

2.1 KI-Anwendungen zum Schutz der Systeme

Eine der bekanntesten Sicherheitsmaßnahmen zur Erhöhung der IT-Sicherheit in Firmennetzen sind Angriffserkennungssysteme, im Englischen Intrusion Detection Systems (IDS) genannt, angewendet auf Computersysteme und Rechnernetze. Sie können Firewalls ergänzen und in der erweiterten Ausführung als Intrusion Detection and Prevention Systems (IDPS) die Cyberangriffe stückweise automatisiert und aktiv verhindern.

Drei Ausprägungen werden heute am Markt angeboten: Host-basierte (HIDS), Netzwerk-basierte (NIDS) und hybride (HIDS) Angriffserkennungssysteme. Während HIDS Informationen aus Log-Dateien, Kernel-Dateien und Datenbanken analysieren, werden NIDS eingesetzt, um Daten-Pakete im Netz zu untersuchen, HIDS wiederum verbinden beide Prinzipien in einem Tool.

Für die genannten Systeme müssen zwei IDPS-Typen unterschieden werden: die Erkennung eines Missbrauchs gegenüber der Erkennung einer Anomalie. Für die Erkennung eines Missbrauchs werden aus der Modellierung eines Angriffs spezifische Pattern extrahiert, gegen die das System systematisch durchsucht wird. Hingegen werden Anomalien identifiziert, wenn das Verhalten des Systems signifikant vom ‚Normalen‘ abweicht. Daher muss eine erkannte Anomalie nicht notwendigerweise einen Missbrauch darstellen.

1 „Ein Bias bezeichnet allgemein Verzerrungseffekte. In der Statistik wird ein Bias als Fehler im Rahmen der Datenerhebung und -verarbeitung verstanden.“

IDPS, ergänzt mit KI-Funktionen, können gezielt mit Angriffsmustern, z. B. mit Denial of Service Angriffen (DoS) trainiert werden, umgekehrt aber auch mit sogenannten „Normalzuständen", um Anomalien zu erkennen. Im Allgemeinen stellt die sichere Erkennung dieser Normalzustände das schwierigste Problem solcher Verfahren dar, besonders dann, wenn sich normale Zustände dynamisch verändern können und das Auftreten neuer Muster normal ist.

Marktprognosen erwarten, dass ab 2020 neue Technologien und Methoden, wie Analytik, Maschinelles Lernen und verhaltensbasiertes Erkennen in den meisten IDPS-Tools integriert sind und am Markt angeboten werden.

Trotz dieses Fortschritts im Arsenal der Abwehrmaßnahmen sollten moderne IDPS-KI-Tools nicht als Universal-Lösung gegen Cyberangriffe betrachtet werden, schon gar nicht für alle Zukunft, sondern als neuen Meilenstein im „Hase-Igel-Wettlauf" zwischen intelligenteren Angreifern und Verteidigern.

2.2 Über Security-Angriffe gegen AI-Anwendungen

- Im Rahmen einer Sicherheitsanalyse von Systemen (Produkte, Anwendungen...) sollen im Wesentlichen zwei Fragen beantwortet werden:

1. Welche potenziellen Bedrohungen existieren?

2. Wie kann man eine potenzielle Bedrohung abwehren?

Zu 1): Typischerweise klassifiziert man Bedrohungen nach dem CIA-Prinzip aus der klassischen, datenzentrierten IT-Security nach CIA = engl. für (confidentiality, integrity, availability). Hinzu kommt noch die Authentizität (engl. authenticity).

Zu 2): Bedrohungen gegen die „confidentiality" werden klassisch durch Verschlüsselung oder Zugangskontrolle zu Daten (oder deren physikalischen Speicher und Verarbeitungssystemen) abgewehrt. „Availabiliy" sichert man durch Redundanzkonzepte und

Abschottung vor Angreifern, Unter „Integrity“ im Rahmen der Datenverarbeitung versteht man den möglichen Nachweis einer unbefugten Veränderung der Daten. Dies geschieht klassisch durch kryptographisch gesicherte Prüfsummen.

- Bei Bedrohungsanalysen von KI-Anwendungen wird ein erweiterter Integritätsbegriff erforderlich:

  - Daten-zentrische „IT-Security“ versteht unter Integrität den möglichen Nachweis einer unbefugten Veränderung von Daten (siehe oben). Im Rahmen von „Industrial Security“ muss der Integritätsbegriff auf die Funktionalität von Geräten und Systemen erweitert werden: Ein System ist dann „integer“, falls seine ausführbare Funktionalität nicht (unerkannt) verändert wurde; d. h. das System führt genau die Funktionen aus, welche von ihm erwartet werden und auch entsprechend dokumentiert sind. Das System soll nicht von Dritten unbemerkt manipuliert werden können.

- Bei KI-basierten Funktionen eines Gerätes oder Systems kann die Ausführung einer Funktion im Einzelfall nicht vorausgesehen bzw. beschrieben werden, da das System im Rahmen seiner internen Entscheidungsfindung nicht für jeden Fall „vorhersehbar“ ist. Über die klassische Integrität von Geräten und Systemen hinaus möchte man sicherstellen, dass die Entscheidungen

  AI1: unvoreingenommen (engl. „unbiased“) getroffen werden,

  AI2: festgesetzte Systemgrenzen nicht überschreiten und

- AI3: die „Frage“ des Fragestellers abhängig von den verfügbaren Daten „sinngemäß“ beantworten. Ein klassischer Verifikationstest gegen einzelne im Lastenheft festgelegte Funktions- oder Featuremerkmale ist meist nicht möglich bzw. nicht zielführend, da nicht offensichtlich ist bzw. gar nicht vorgegeben werden soll, „wie“ und aufgrund welcher Merkmale ein ML-System zu einer Klassifizierung gelangt.

Es stellt sich nun insbesondere die Frage, wie böswillige Angriffe gegen die Eigenschaften KI1–KI3 abgewehrt oder zumindest erkannt werden können. Können hier klassische Security-Methoden wie Prüfsummen, Verschlüsselung oder Redundanz angewendet werden? Sind bekannte Mechanismen ausreichend oder müssen neue Methoden entwickelt werden? Wie kann man insbesondere die Eigenschaften KI1 und KI3 testen?

- Beispiel aus der „Safety“: Ein KI-basiertes System soll sich in den Grenzen gewisser Erwartungen verhalten: Eine autonome Maschine/Roboter soll den Sicherheitsbereich nicht verlassen bzw. soll prinzipiell Kollisionen mit anderen Maschinen oder Menschen verhindern. Diese Verhaltenseigenschaft darf natürlich nicht durch böswillige Manipulation der zugrundeliegenden AI-Anwendung oder der Input-Daten beeinträchtigt werden.

- Frage: Ist „Manipulation der Input-Daten“ eine Bedrohung, gegen die man sich wehren kann? Gibt es (vorgeschaltete) Plausibilitätstests als Schutzmechanismen für KI-Anwendungen?

- Ähnliche Fragen sind aus dem Test (stark) nichtlinearer kryptographischer Funktionen mit sehr großer Definitions- und Wertemenge (> 2128) bekannt. Trotz der korrekten Verifikation vorgegebener Testvektoren kann sich später herausstellen, dass noch Fehler in der Implementierung vorlagen.

Zur Lösung der oben genannten Fragestellungen besteht weiterhin Forschungsbedarf.

3. Erklärbarkeit von KI-Entscheidungen

3.1 Notwendigkeit von Erklärungen durch menschliche Argumentation

Es ist heute ein strittiges Thema, inwieweit es notwendig, konzeptionell sinnvoll und technisch möglich ist, KI-Entscheidungen im Sinne einer Klassifizierung für menschliche Argumentation erklärbar oder interpretierbar zu machen. In manchen Anwendungsfeldern bestehen offensichtliche geschäftliche Rechtfertigungen für den Wunsch nach Erklärbarkeit: Warum wurde die Produktionsstraße durch einen autonomen Roboter zum Stillstand gebracht und wer trägt daran die Schuld? Warum wurde mein Kreditantrag abgelehnt, obwohl mein gefühlter Finanzstatus doch besser ist als der Durchschnitt? Es sind in der Regel Fragen nach Schuld, Haftung und der Korrektheit von Diagnosen, für die der Wunsch nach menschlich verständlicher Argumentation besteht. Andererseits kann davon ausgegangen werden, dass dieser Wunsch abnimmt und zumindest teilweise durch größeres Vertrauen in maschinelle Entscheidungen ersetzt wird.

Es wird immer mehr Bereiche geben, in denen Maschinen präziser, viel schneller und wesentlich zuverlässiger entscheiden. „Würden Sie sich eher einer Herzoperation durch einen renommierten Professor bei einem Sterberisiko von 8 Prozent oder durch einen Roboter mit Sterberisiko von 2 Prozent unterziehen? Wäre es dabei wichtig für Sie, dass der Professor im Gegensatz zum Roboter im negativen Fall Ihren Hinterbliebenen verständlich erklären könnte, woran der Eingriff scheiterte?“ Außerdem ist es heute in vielen wissenschaftsbasierten Praktiken vollkommen normal, sich auch in kritischen Bereichen auf statistische Erfahrung und nicht auf rationale Erklärung zu stützen. Medizin und Pharmazie liefern zahlreiche Beispiele. Die Wirkmechanismen gängiger Medikamente wie Aspirin wurden erst Jahrzehnte nach ihrem breiten Einsatz erforscht und die allgemeine Testpraxis der Pharmazie geht von Labortieren über spezielle Patientengruppen zu „normaler“ Medikation, aber nicht über rigorose Erklärung.

Unbestritten ist dagegen, dass erklärbare KI (explainable AI – XAI) Beiträge liefern kann, um zu beurteilen, was ein System tatsächlich gelernt hat und zu ergründen, ob inhaltliche Verzerrungen in den Trainingsdaten sich im Entscheidungsverhalten niederschlagen. Beispielsweise zeigten Sensitivitätsanalysen der Entscheidung im Hinblick auf die Pixel von Eingabebildern, dass ein Hauptkriterium eines erfolgreichen Bilderkennungssystems bei der bemerkenswert hohen Sicherheit in der Unterscheidung zwischen Wölfen und Huskies der Schnee im Hintergrund war. Natürlich sollte das System nicht lernen, dass ein wolfähnliches Tier auf Schnee wahrscheinlich ein Husky ist. Den Bereitstellern der Trainingsbilder war dieses Merkmal ihres Trainingssets auch sicherlich nicht bewusst. Das Beispiel zeigt aber sehr deutlich, dass Trainingsdaten gefährliche Bias-Eigenschaften haben können, die – gewollt oder ungewollt – technische, ethische und politische Fehlleistungen bewirken können. Solche Fehlleistungen können auch Gegenstand von Diskussionen im Kontext der DSGVO (GDPR) sein.

3.2 Komplexität des Erklärungsproblems

Erst die neuen Algorithmen des Machine Learning (ML) haben das Problem der abnehmenden Erklärbarkeit von KI-Entscheidungen erzeugt und bewusst gemacht. ML löst Klassifikationsprobleme durch Approximation: Beobachtungen werden durch eine „Hypothese“ angenähert. Mit Millionen von Beobachtungen kann man mathematische Funktionen mit Millionen von Parametern kalibrieren („trainieren“) – das ist die neue Qualität extrem komplexer Hypothesen (beispielsweise in Neuronalen Netzen). Während programmierte Entscheidungsbäume oder die Parameter der linearen Regression und anderer linearer Modelle noch in ihrer menschengemachten Konstruktion weitgehend erklärbar waren, trifft dies auf moderne Neuronale Netze mit über 20 bis in die Hunderte von Neuronenschichten nicht mehr zu. Menschen denken nicht in derart komplexen Argumentationsstrukturen und die Merkmale, die Neuronale Netze in diesen Schichten bilden, haben keine Entsprechungen in menschlicher Mustererkennung.

So entsteht das Problem, dass fortgeschrittene ML-Techniken übermenschliche Fähigkeiten in der Prädiktion entwickeln, aber gleichzeitig die Erklärbarkeit in menschlicher Argumentation verlieren. KI-Spezialisten können mit hochentwickelten Analysetechniken aufdecken, warum ihr System, in technischer Hinsicht welche Entscheidungen traf (Sensitivitätsanalysen). Insofern sind Neuronale Netze für KI-Spezialisten keine „Black-Box-Systeme“. Es geht bei XAI also vorrangig darum, die überlegene Leistungsfähigkeit moderner ML-Technik durch verbesserte Erklärbarkeit für domänenspezifische Anwender ohne KI-Expertise nutzbar zu machen. Es ist auch ein noch nicht klar beschriebenes Ziel auf dem Weg zu kontextsensitiver KI, in welcher Form menschliche Expertise und Argumentationsfähigkeit helfen könnte, die Abhängigkeit von exorbitanten Trainingsdatenmengen zu verringern und so Lernalgorithmen inspirieren könnte, die wie biologische Gehirne mit viel weniger Beispielen lernen.

3.3 Forschungsansätze

Es gibt seit langem Sensitivitätsanalysen bezogen auf Komponenten von Trainingsdaten: Welche Pixel in Bildern wirken sich in welchem Maß auf Klassifikationsentscheidungen aus. Da sowohl Eingaben als auch Klassifikation menschlich verständlich sind, liefern solche Techniken Aufschluss über verborgene Verzerrungen (Biases) in Trainingsdaten.

Die erwartete Modularisierung in ML-Technologie stellt eine Analogie zu der konzeptionellen Revolution der „strukturierten Programmierung“ aus den 70er Jahren dar, als „Spaghetti-Code“ von Software im heutigen Sinn abgelöst wurde.

Die größte konzeptionelle Herausforderung bleibt die Identifikation von erklärenden Merkmalen in einem offenbar intelligent agierenden System in menschlich verständliche Begriffsbilder und Argumentation. Wie kann eine Argumentation aus beispielsweise 24 Ebenen für Menschen erklärbar werden, die mehr als 3 bis 4 Ebenen intellektuell nicht erfassen können? Der derzeit verfolgte Ansatz besteht vorwiegend darin, weitere KI zur Erklärung von KI einzusetzen: Neuronale Netze, die Menschen erklären, was die intellektuell überlegenen Neuronalen Netze mit ihren Aussagen meinten.

3.4 Vorhandene Techniken

Grundsätzlich kann man versuchen, komplexe und daher auch sehr präzise Systeme als Black-Box zu benutzen und einfachere, aber dafür erklärbarere Systeme einzusetzen, um die Prädiktionen des präzisen komplexen Systems erklärbar zu machen. Das einfache System benutzt dabei die Klassifizierungen des komplexen Systems als Labels. Dieser Ansatz ist sehr umstritten. Er wird oft als „global“ bezeichnet, weil hier das System als solches, nicht eine einzelne Prädiktion, erklärbar gemacht werden soll.

Andere Techniken konzentrieren sich auf lokale Ereignisse im Sinne der Aussagen des Systems für spezifische Klassifikationen: „Warum machte das System diese Aussage für diese spezielle Eingabe“. Für diesen Bereich existieren inzwischen diverse verfügbare Software-Lösungen, die in realen Projekten eingesetzt werden:

- LIME: “Local Interpretable Model-Agnostic Explanations” kann zur Analyse beliebiger Black-Box-Systeme verwendet werden, um die Wichtigkeit von bestimmten Features für eine gegebene Prädiktion zu bestimmen. Es ist das derzeit verbreitetste Paket.
- “Shapley Value Explanations” beruhen auf dem spieltheoretischen Konzept des Shapley-Wertes, mit dem die gerechteste Methode zur Verteilung des Ertrags in einem kooperativen Spiel ermittelt wird. Es ist der derzeit aktuellste und wissenschaftlich fundierteste Ansatz, benötigt aber sehr große Rechenleistung, weil eine große Zahl von Ersatzmodellen trainiert werden muss. Das Paket „SHAP“ versucht die existierenden Konzepte mit dem Shapley-Wert-Ansatz durch Approximation mit realistisch möglichem Rechenaufwand vereinbar zu machen.
- Das LSTMVis-Projekt hat zum Ziel, die Erklärbarkeit von neuronalen Sequenzmodellen zu beschreiben, die durch sogenannte Rekurrente Neuronale Netze erzeugt wurden [1].
- Es existieren diverse andere Ansätze und Angebote, die aber grundsätzlich in diese führenden Methoden konzeptionell integrierbar sind.

3.5 Beispiele

- KI bei der Textanalyse: Erkennung einzelner Worte, deren Synonyme und des Kontextes auf Basis domainspezifischen Wissens für Rückschlüsse auf Inhalte bzw. beabsichtigte Aussagen oder Bewertungen. Einsatzfelder sind fachspezifisches Training und Support. Die KI vermittelt eventuell oberflächlich betrachtet eine gewisse semantische Erkennung – tatsächlich wird aber nicht die Erkennung der Bedeutung von Sachverhalten erlernt, sondern nur die Kontexte, in denen die Worte von Menschen zuvor verwendet wurden. Eine „höhere“ Intelligenz liegt diesem Einsatzfeld nicht zugrunde.
- KI-basierte Übersetzungssysteme: Überwiegend online verfügbar, basieren auf erlernten Textkorrelationen unterschiedlicher Fremdsprachen, die aus vorhandenem Übersetzungsmaterial bestehen. Werden ausschließlich hochwertige, zuvor von qualifizierten Menschen (Fachübersetzern) erzeugte Trainingstexte erlernt, können bei hinreichend breitem und hochwertigem Trainingstextangebot qualitativ gute Ergebnisse durch KI-Systeme erzeugt werden. Allerdings wird keine Qualitätserhöhung minderwertiger Trainingstexte durch KI-Systeme erreicht. Minderwertige Trainingstexte aus dem „Vortraining“ können sich dauerhaft negativ auswirken, wenn der aktuell zu übersetzende Text nicht im Detail erlernt wurde und der verwendete Algorithmus „unter allen Umständen“ einen „Übersetzungstext“ liefert. Bei Onlineangeboten

existiert ein qualitativer Graubereich. Eine „höhere, semantische“ Erkennung mittels KI liegt auch diesen Systemen nicht zugrunde.

- „Hochautomatisierte Roboter in Produktionen benötigen u. a. 3-D Umgebungsinformationen, damit z. B. Kollisionen in der Produktion verhindert werden. Zwei Informations-Elemente müssen dazu sehr schnell von der KI verarbeitet werden: a) Bildanalyse der Umgebung, b) Vorhersage der motorischen Bewegung, z. B. vom Roboter, vom Roboterarm aber auch vom Werkstück.
- KI in der intelligenten Automatisierung benutzt alle Daten für verfügbare Produktionsmittel, für zu bearbeitende Werkstücke, für notwendige Prozesse, aber auch für Qualitätskontrollen, Zwischenlagerbestand, Nachbearbeitung usw.

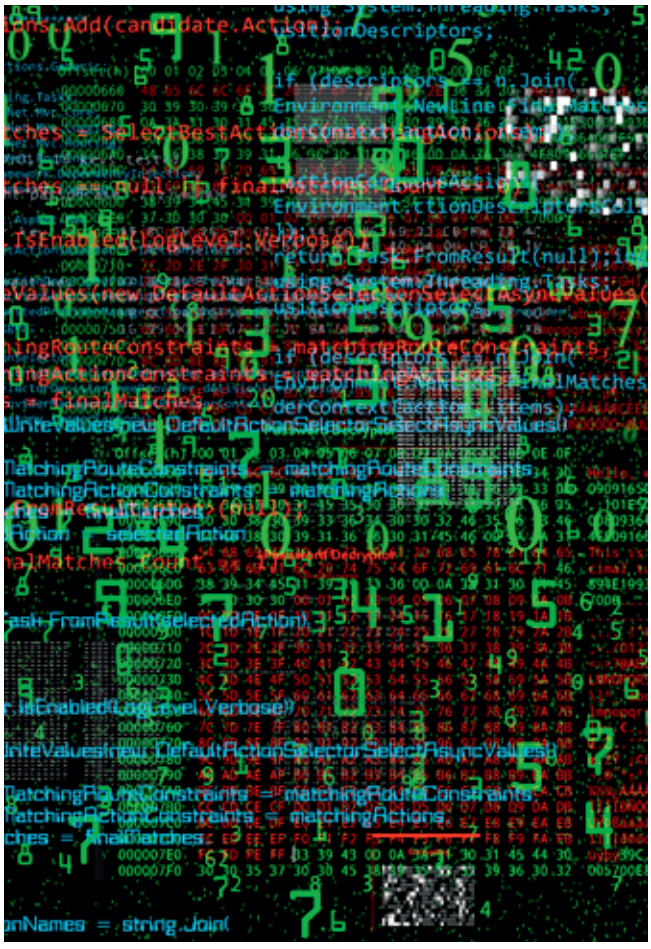
4. Angriffsvektoren auf KI bzw. mittels KI

KI-basierte Systeme unterliegen Risiken, die in früheren Systemen nicht existierten. Solche Risiken entstehen in Trainingsvorgängen mit undokumentierten oder menschlich unverständlichen Daten, durch missverstandene Architektur-, Prozess- oder Leistungsmerkmale und durch andere Struktureigenschaften, die Anwender zumindest teilweise nicht kennen. Aufklärung und Beratung ist geboten, damit Betreiber nicht unbewusst Risiken eingehen. Außerdem haben sich in den letzten fünf Jahren KI-basierte Angriffstechniken entwickelt, die besonders für KI-basierte Systeme gefährlich sein können, die aber auch die gesamte industrielle Security-Diskussion herausfordern und umfassendere Schutzkonzepte brauchen als sie bislang etabliert sind.

Gegnerische Beispiele (adversarial examples) sind Eingabedaten, die gezielt konstruiert werden, um Mustererkennungssysteme zu falscher Klassifikation zu führen. Das System erkennt etwas völlig anderes als es ein menschlicher Beobachter würde. Wie ist das möglich und welche Techniken werden von Angreifern eingesetzt? Dieses Kapitel verdeutlicht den Vorgang an dem bekannten und einfachen Beispiel des Modified NIST data set (abgekürzt MNIST-Datensatz) handgeschriebener Ziffern. Er wurde in den 90er Jahren durch die NIST-Behörde erstellt, um die Postsortierung in den USA zu automatisieren und besteht aus 60.000 28X28 Pixel 8-bit Graustufenbildern. Professionelle CNN erreichen Trefferraten von über 99 %, einfache Systeme wie logistische Regression kommen auf über 93 %. Praktisch alle neuen ML-Konzepte werden an diesem Datensatz erprobt, weil er so einfach handhabbar und problemlos zugänglich ist (vgl. Überblick in [2]).

Das Problem beginnt mit der Mächtigkeit der Modellierung durch ML: Die Bilder liegen offensichtlich in einem 28X28 = 784-dimensionalen Raum aus ganzen Zahlen mit Werten zwischen 0 und 255. Es gibt 256^{784} also mehr als 10^{1888} mögliche Bilder. Führt man sich vor Augen, dass die Physik schätzt, dass das sichtbare Universum „nur“ etwa 10^{80} Atome besitzt, dann erkennt man, dass die Zahl der möglichen MNIST-Bilder sehr (!) groß ist. „Normale“ Fotos aus einer Digitalkamera spannen noch dramatisch größere Räume auf. Zufällige Zusammensetzungen von Pixeln in MNIST-Bildern werden vom menschlichen Auge als Rauschen wahrgenommen, solche Bilder stellen nichts Erkennbares dar. Bilder, die ein Mensch als Ziffer (0, ..., 9) erkennt, bilden nur eine winzige Teilmenge – mit vielleicht einigen Millionen Elementen, sind aber verschwindend wenig gegenüber 10^{1888}.

Deshalb wäre es naheliegend, nach Bildern zu suchen, die ein mit MNIST trainiertes Netz als Ziffer erkennt, obwohl sie für Menschen gar nicht so aussehen. Dazu kann man den gleichen Algorithmus verwenden, mit dem die Gewichte des Netzes bestimmt wurden, der nun aber auf die 784 Pixel angewandt wird. Tatsächlich findet man so in großer Zahl Rauschbilder, denen das trainierte Netz hohe Wahrscheinlichkeiten zuordnet, dass es sich um eine bestimmte Ziffer handelt: Was für Menschen Rauschen ist, betrachtet das Netz mit hoher Zuversicht als 3, 4, 7, ... Mit einer leichten Änderung der Zielfunktion für den Gradientenabstieg kann man dann beispielsweise eine Suche nach Mustern starten, die als „3“ interpretiert werden, aber für Menschen aussehen als wäre es beispielsweise eine „7“. Man kann die Perversion auf die Spitze treiben und mit diesem Verfahren für jedes Bild aus dem MNIST-Testset andere Bilder entwickeln, die für Menschen identisch aussehen, vom trainierten Netz aber als jede andere Ziffern mit Zuversicht von mehr als 90 % klassifiziert werden.



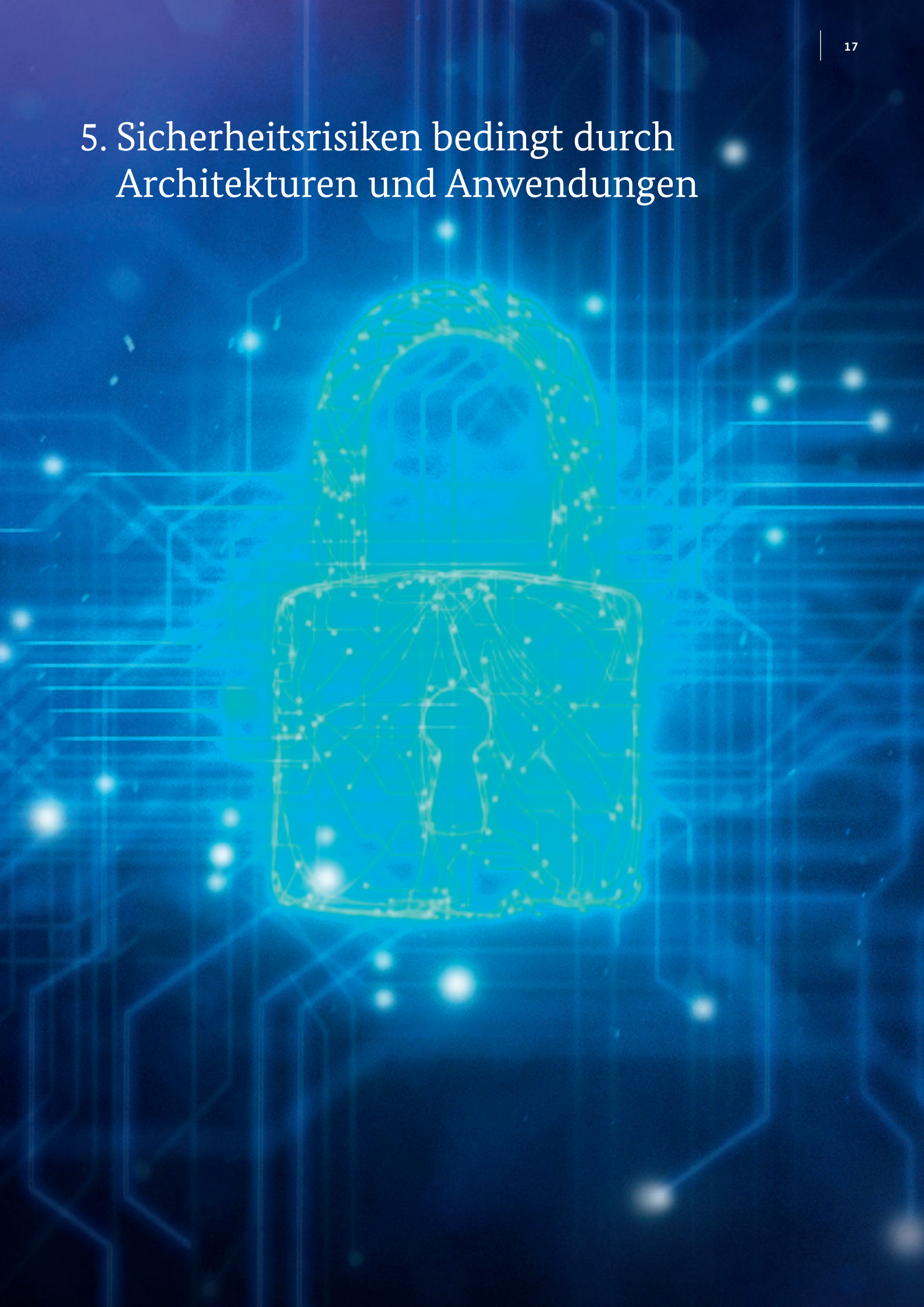
Es gibt Methoden, gegen solche Attacken zu kämpfen. Ein sehr einfacher Weg besteht darin, auf reine schwarz-weiß Pixel überzugehen. Die Fehlklassifikation entsteht im grauen Rauschen und ist damit größtenteils behoben. Das funktioniert aber nur in Bildern, die in dieser Darstellung noch Sinn ergeben: Bei Schrift ist das anwendbar, bei Fotos normalerweise nicht, besonders dann nicht, wenn die Auflösung gering ist und erst durch Grautöne der Inhalt plastisch wird. Eine andere Möglichkeit besteht darin, die falsch erkannten Bilder mit den korrekten Labels in das Training einzubeziehen. Auch diese Technik funktioniert. Aber um sie anzuwenden, müssen die gegnerischen Beispiele bekannt sein. Der Verteidiger wird aber normalerweise nicht wissen, mit welcher Zielfunktion der Angreifer seine Bilder erzeugt. Er kann also nur nach einem erfolgreich durchgeführten Angriff die erkannte Lücke schließen.

Noch dramatischer ist das Bedrohungsszenario, das aus der Transferierbarkeit gegnerischer Beispiele zwischen unterschiedlichen ML-Systemen entsteht [3]. Ein Beispiel, das ein gegebenes System täuscht, ist mit hoher Wahrscheinlichkeit auch in der Lage, ein vollkommen anderes ML-System zu täuschen: Beispiele, die ein bestimmtes, sagen wir, zweistufiges CNN täuschen, sind oft in der Lage, auch CNN mit mehr oder weniger oder anders konfigurierten Schichten zu täuschen und selbst einfachere Techniken, wie logistische Regression, oder sehr komplexe Techniken, wie ResNet oder Inception, sind durch die gleichen Angriffe gefährdet. Der Angreifer muss die Technologie seines Opfers nicht kennen, es kann für ihn eine Black-Box sein, aber er hat dennoch gute Chancen einen erfolgreichen Angriff zu starten.

Adversarial Examples können auf diese Weise gewissermaßen auf Vorrat produziert werden, um zu gegebenem Zeitpunkt zielgerichtete Zero-Day-Angriffe zu starten. Es ist ein extremer Fall von Asymmetrie zwischen Angreifer und Verteidiger, der die Angriffsvektoren, die im Verborgenen bereitstehen, nicht kennen kann.

Widerstand und Abwehr gegen adversarial examples ist eine Aufgabe für ML-Experten, die in der Lage sind, durch geeignete sehr speziell ausgerichtete Systemkonfiguration und Trainingsmaßnahmen einen individuell angepassten Rahmen zu entwickeln, der potentiellen Angreifern hohe Aufwände abverlangt, um erfolgreiche Angriffsbeispiele zu entwickeln. Die Literatur zu diesem Thema ist sehr umfangreich und komplex. Es geht vorrangig darum, in einer differenzierten Betrachtung Situationen für ML in speziellen Anwendungsfällen zu erzeugen, die Resistenz gegen adversarial examples garantieren können. Eine umfassende allgemeine Lösung des Problems ist noch Gegenstand der Forschung, aber für spezielle Anwendungen ist dies heute bereits möglich.

5. Sicherheitsrisiken bedingt durch Architekturen und Anwendungen



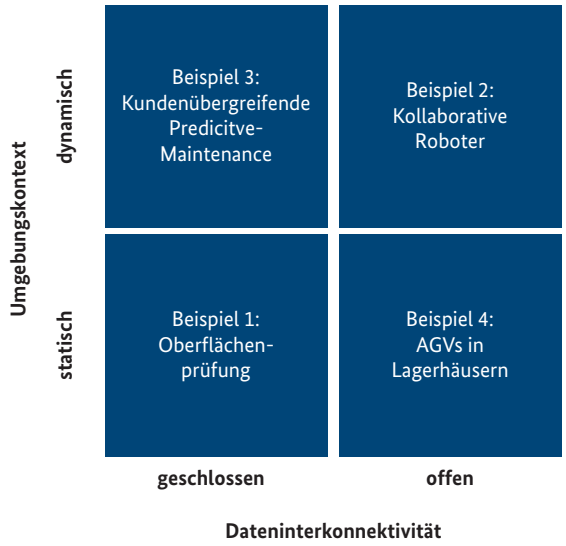
Verteilte industrielle Architekturen und Anwendungen stellen bekannte wichtige Eigenschaften der Industrie 4.0 dar. Das sichere Zusammenspiel von KI-Architekturen mit dieser neuen Infrastruktur ist allerdings überwiegend unerforscht. Sicherheitsrisiken, wie Wirtschaftsspionage, -sabotage und Datendiebstahl, sind in Reflektion der gewählten KI-Architektur und vorgesehenen Anwendung zu betrachten. KI-Anwendungen in der Produktion können hierfür unterschiedliche Klassifizierungen und Kriterien besitzen, wie

- **Statische** versus **dynamische** Anwendung der KI; dies bezieht sich auf die systemische Weiterentwicklung der KI-Software;

- **Geschlossene** bzw. **gekapselte** Anwendung mit nur einem Anwender bzw. einer Anwendung, versus **offene** Anwendung mit dynamischer Anzahl von Anwendern bzw. Anwendungen.

Für die Begriffe „Klassifizierung“ und „Kriterien“ könnte auch der Begriff „Freiheitsgrad“ bzw. allgemein „Komplexität“ der eingesetzten KI verwendet werden. Je nach Anzahl der Freiheitsgrade der KI-Anwendung sind unterschiedliche Bedrohungs-Szenarien zu berücksichtigen, was diese Fallunterscheidung wichtig macht.

Abbildung 3: Vier Quadranten-Darstellung der Freiheitsgrade einer KI-Anwendung. Nicht dargestellt sind KI in der autonomen Anwendung und beim Pooling.



Quelle: Plattform Industrie 4.0

Beispiel 1: Ein Beispiel für eine **statische**, **stationäre** und **geschlossene** Anwendung ist der Einsatz einer Kamera im Zuge einer vollautomatischen Qualitätskontrolle der Oberfläche eines Metallformteils, bevor dieses Freiformteil in die Lackier-Straße weitertransportiert wird. Die Kamera untersucht die Oberfläche vom Rohling nach möglichen Verformungen, die von der Soll-Form abweichen, wie eine Delle oder eine Welle, die beim Tiefziehen entstanden sein kann. Nach der Lackierung würden solche sehr geringe Verformungen sehr gut mit dem Auge sichtbar werden und das Freiformteil, z. B. ein Kotflügel, würde verworfen werden. Die KI-Software wertet das Kamerabild aus und vergleicht dieses gegen Referenzbilder. Die Referenzbilder sind durch ständiges Lernen permanent weiterentwickelt worden. Die Kamera und die KI-Software arbeiten dabei getrennt von anderen Produktions- bzw. Qualitätsschritten. Damit kann dieser Mess- und Entscheidungskreis als autonom von anderer HW und SW in der Produktionsumgebung betrachtet werden. Daten von anderen Kamerasystemen werden nicht benötigt, da sie an anderen Freiformflächen eingesetzt werden, bei denen andere Soll-Maße existieren. Im Gegensatz zur menschlichen Oberflächenkontrolle, kann das System aus Kamera und KI-SW deutlich genauere Ergebnisse liefern und kann eine gleichbleibend hohe Qualität erzeugen.

Geschlossene bzw. gekapselte Anwendung bedeutet in diesem Beispiel eine Routinetätigkeit der KI in der Produktion, die von einem einzigen Operator überwacht wird. Eine Kamera, eine Auswerte-Software und ein Bauteil bzw. Werkstück mit hoher Wiederholung in den geometrischen Abmessungen kennzeichnen diese Anwendung. Das Sicherheitsrisiko für Cyberangriffe mit dem Ziel der Wirtschaftsspionage bzw. -sabotage ist überschaubar, da der Täter und die Tat sehr schnell erkannt werden können.

Beispiel 2: Ein weiteres Beispiel für eine **dynamische**, räumlich ausdehnende und **offene** Anwendung ist das Zusammenschalten kollaborativ zusammenarbeitender Roboter und deren Steuerung über einen Leitstand. Das ist Zukunftsmusik. Mehrere Produktionsroboter in einer Autoproduktion tauschen über einem Leitstand, der eine KI-Software enthält, ständig Umgebungsdaten fast in Echtzeit aus. Die KI soll dabei mögliche Kollisionen von sehr schnell laufenden Roboter-Armen und zu bearbeitenden Werkstückteilen vorhersehen und durch Eingriffe in den Prozessablauf verhindern. Zudem können Programmierer neue Produktionsprozesse in jedem einzelnen Roboter einspielen, die wiederum der KI-SW als neuer Parametersatz übertragen werden muss. Die KI wird dabei mit vielen Robotern vernetzt, die jeweils unterschiedliche Prozesse in der Produktion anbieten. Die Prozessprogrammierung jedes Roboters wird durch mehrere Operatoren ständig optimiert. Der Mess- und Entscheidungskreis ist als offen zu bezeichnen, da die Zahl der beteiligten Produktionsmittel ständig variiert und daher einen dynamischen Faktor bildet, wie auch die Prozesse ständig durch Operatoren überarbeitet werden und daher ebenfalls hochgradig dynamisch sind.

Beispiel 3: Ein Beispiel für eine größtenteils **geschlossene** Anwendung einer KI, die jedoch im Vergleich zu den Beispielen 1 und 2 ein kontrolliertes, **dynamisches** Weiterlernen der KI im laufenden Betrieb mit entsprechender dynamischer Anpassung des Entscheidungsverhaltens beschreibt, ist die Nachklassifikation von Sicherheits-Alarmen aus einer (Cyber-)Security Anwendung durch einen Security-Experten. In diesem Fall werden vom Hersteller der Software Prüfalgorithmen ausgeliefert, die im Kontext der jeweiligen Einsatzumgebung oder des jeweiligen Unternehmens, welches die Software einsetzt, Fehlalarme produzieren können. Da die individuellen Einsatzumgebungsparameter stark voneinander abweichen können, kann der Hersteller keine KI-Parameter ausliefern, die das Ergebnis des Prüfalgorithmus im jeweiligen Umgebungskontext klassifizieren. Typische Klassifikationen solcher Alarme sind die Klassifikation nach echten Angriffsalarmen, Fehlalarmen, Fehl-Konfigurations-Alarmen, Unternehmens-Policy-Alarmen etc., die jeweils unterschiedliche Reaktionen erfordern. Durch die Nachklassifikation der Alarme durch einen Security-Experten entstehen im Laufe des Betriebes der Software viele „gelabelte" Datensätze, die direkt der KI zum dynamischen Nachlernen im laufenden Betrieb zugeführt werden und somit zur automatisierten Klassifizierung beitragen.

Das Beispiel 3 kann leicht auf das Beispiel 1 ausgeweitet werden, wenn bei Qualitätskontrollen durch einen Mitarbeiter eine andere Bewertung des Prüfergebnisses der KI entsteht und die abweichende Bewertung über eine entsprechende Software der KI des Prüfsystems als „gelabelter" Datensatz wieder zugänglich gemacht werden kann.

Beispiel 4: Der Einsatz von intelligenten, selbstnavigierenden Transportplattformen (Automated Guided Vehicles, AGV) in Lagerhäusern und verteilten Produktionslandschaften wie etwa im Flugzeugbau dient als gutes Beispiel für **offene** und zugleich vom Lernverhalten **statische** KI-Anwendungen. Die Offenheit ergibt sich durch den bedarfsorientierten Einsatz von mehr oder weniger teilnehmenden AGVs in den Anwendungen. So mag ein Lagerhaus im hektischen Weihnachtsgeschäft zusätzliche Einheiten in Betrieb nehmen, um die Kapazität der Auftragsabwicklung zu erhöhen. Eine weitere Dimension der Offenheit ergibt sich durch die Vielfalt der zu transportierenden Elementen. Besonders Logistikanwendungen mit Kontakt zu externen Parteien (z. B. von Kunden verpackte Waren) können auf unerwarteten Wegen von außen zugänglich werden. Grundsätzlich wird die unterliegende Intelligenz der einzelnen Transportplattformen nicht wesentlich modifiziert, es findet also kein Weiterlernen des Navigationssystems statt und kann daher als statisch gekennzeichnet werden. Dies ist nicht zuletzt auch den Anforderungen der funktionalen Sicherheit geschuldet, die eine Veränderung zertifizierter Systems ausschließen oder zumindest stark einschränken. Typische KI-Funktionen in diesem Bereich konzentrieren sich im Bereich der Navigation der Einzelplattformen und des Flottenmanagements. Es sollte aber nicht übersehen werden, dass auf der Ebene des Flottenmanagements zunehmend dynamische Lernprozesse realisiert werden. Hier ist eine Koexistenz von dynamischen und statischen Elementen in derselben Anwendung absehbar.

Die wesentlichen Unterschiede der KI-Anwendungen sind in den oben genannten Fällen somit wie in Tabelle 1.

Was bedeutet das für eine fehlerhafte oder gezielte Manipulation der KI, z. B. zum Zweck der Wirtschaftssabotage? Die KI in geschlossener Anwendung kann von einer einzelnen Person beschafft, in Betrieb genommen und kontrolliert werden. Im Falle einer Sabotage wäre der Verursacher sofort entlarvt und eine Gegenmaßnahme schnell einleitbar. In der offenen Anwendung müssten zunächst mögliche Ursachen untersucht werden, wie falsche Programmierung der Robotersteuerung, fehlendes Update von neuen Prozessparametern oder fehlende Mitteilung, dass ein weiterer Roboter in den Prozessablauf eingebunden ist. Ein möglicher Täter, der für einen Sabotagefall verantwortlich ist, wäre nur schwer, wenn überhaupt, ermittelbar. Die große Zahl der beteiligten Maschinen und Menschen und die hohe Komplexität würden den Saboteur vor Aufdeckung schützen.

Die folgende Gefährdungslage lässt sich aus Tabelle 2 ableiten.

Mit der oben gestellten Frage nach lokalen Auswirkungen geht eine weitere Dimension einher, nämlich die Beherrschbarkeit, ausgelöst durch den Einsatz von KI in der Cloud (d. h. Rechenzentren). In den Rechenzentren werden typischerweise KI-Anwendungen trainiert, deren Leistungsanforderungen aufgrund der verfügbaren höheren Rechenleistungen dort erfüllt werden können. In der Edge kommen dann die Inferenz-Systeme der KI zum Einsatz. In Ausnahmefällen können beide Prozesse je nach Komplexitätsanforderungen sowohl im Rechenzentrum als auch in der Edge (verteilte KI) durchgeführt werden.

Auch die Unterscheidung zwischen offener und geschlossener Daten-Interkonnektivität liefert Hinweise zu möglichen Gefährdungslagen s. Tabelle 3.

Tabelle 1

	KI in statischer Anwendung	KI in dynamischer Anwendung
Systemeigenschaften	eher geschlossen, kompakt	eher offen, komplex
Komplexität in der KI-Anwendung	eher gering	eher hoch
Benötigte Rechenleistung der KI-SW	eher gering	eher hoch
Beteiligte Produktionsmittel und Operatoren, die Inputs für die KI-SW liefern	eher wenige	eher viele

Tabelle 2

	Statische Anwendung	Dynamische Anwendung
Lokale Auswirkung im Fall von Sabotage	begrenzter	großflächiger
Durchschlagskraft eines Sabotagevorfalls	geringer	höher
Erkennbarkeit der Lage	eher möglich	eher unmöglich
Aufwand der Beseitigung nach Erkennung	eher geringer	eher höher

Tabelle 3

	Geschlossenes System	Offenes System
Erkennbarkeit des Angriffs	hoch, wahrscheinlich	schwierig, erfordert starke, komplexe Schutzmechanismen
Auswirkungen des Angriffs	eher gering, da Weiterverbreitung begrenzt ist	potenziell sehr hoch, da die weitere Ausbreitung nur schwer verhindert werden kann
Aufwand für die Beseitigung des Angriffs	eher gering	sehr hoch, korreliert mit dem Ausmaß des Befalls

Unterschied zwischen Cloud- und Distributed Edge-Anwendungen

- Intelligente Netzwerke in der Industrie fordern sehr schnelles Systemverhalten, welches Cloudanwendungen innerhalb begrenzter Latenzanforderungen nicht immer liefern können. Niedrige Latenzanforderungen sind oft entscheidend für industrielle Anwendungen wie automatische Steuerungen, koordinierte Roboterbewegungen (siehe Beispiel 2), Videoüberwachung für Produktqualitätsprüfungen (siehe Beispiel 1), Stromverteilung, Verhinderung von Fehlalarmen, etc.

- Die große Anzahl von Assets, die mit der Cloud verbunden sind, benötigt eine große Bandbreite für die Übertragung der Rohdaten, und die Cloud benötigt große Rechen- und Speicherressourcen, um dieses hohe Datenvolumen zu verarbeiten. Darüber hinaus enthalten die Rohdaten oft sensible Informationen und Angriffe auf diese Daten in der Cloud können den eingesetzten Unternehmen zudem ehebliche Schäden zufügen.

  Netzbetreiber gehen ebenfalls vermehrt dazu über, Intelligenz in die Edge zu verlagern, um Netze zu optimieren und Netzwerküberlastungen zu reduzieren. Distributed Edge Computing stellt aus Sicht eines Netzbetreibers eine erwünschte Topologie dar. Edge Computing gilt als eine verteilte offene Plattform am Netzwerkrand, nahe an den Dingen und Datenquellen, welche die Fähigkeiten von Netzwerken, Speichern und Anwendungen integriert und ebenfalls lokale Inferenz -Systeme unterstützt.

  Ergänzt wird diese Architektur durch Multi-Access Edge Computing (MEC), das Elemente der Informationstechnologie und der Telekommunikationsnetze vereint und Netzbetreibern ermöglicht, ihr Netz für autorisierte Dritte wie Anwendungsentwickler und Inhaltsanbieter zu öffnen.

  Neue KI-Systeme der Netzbetreiber optimieren diese Lastanforderungen und erfüllen die Anforderungen der Industrie 4.0 im Hinblick auf agile Konnektivität, Echtzeitdienste, Datenoptimierung, Anwendungsintelligenz, Sicherheit und Datenschutz deutlich besser.

  In dieser Situation ist es jedoch wichtig, auch hier die Herausforderung zu erkennen und die Funktionalitäten der KI für den Betrieb in der Cloud von denen für den Betrieb in der Edge an den Endgeräten, wie Smart Devices, zu trennen. Hochleistungs-Chipsätze und Edge-Computing-Plattformen sind erforderlich, um die performanten Algorithmen der künstlichen Intelligenz zur Inferenz zu implementieren und auszuführen. Wichtig ist auch die Unterstützung des Kontinuums des Wissensaustauschs zwischen den KI-Funktionen in der Edge und denen in der Cloud.

Steuerung von Netzwerken durch KI

In diesem Szenario kommt den Trainingsdatensätzen zur Merkmalserkennung unterschiedlicher Netzanforderungen besondere Bedeutung zu, da die Netz-KI als kritische Infrastruktur gilt, deren mögliches Fehlverhalten ähnliche Kollateralschäden wie autonome Systeme verursachen kann, die letztlich als Tradeoff gewertet werden müssen.

Aller Voraussicht nach dürfte die Überlegung, für die in Netzen typischen chaotischen Systemzustände den dritten bekannten Algorithmus – nämlich das Reinforcement Learning für diese Anforderungen – zu verwenden, daran scheitern, dass in kritischen Infrastrukturen das Trial-and-Error-Prinzip kaum umsetzbar sein dürfte. Der Reinforcement Learning-Algorithmus kommt dem Optimum Schritt für Schritt dadurch näher, dass er eine Vielzahl von Iterationen durchläuft, bei denen er bewährte Verhaltensmuster miteinander kombiniert und zufällig neue Verhaltensweisen ausprobiert.

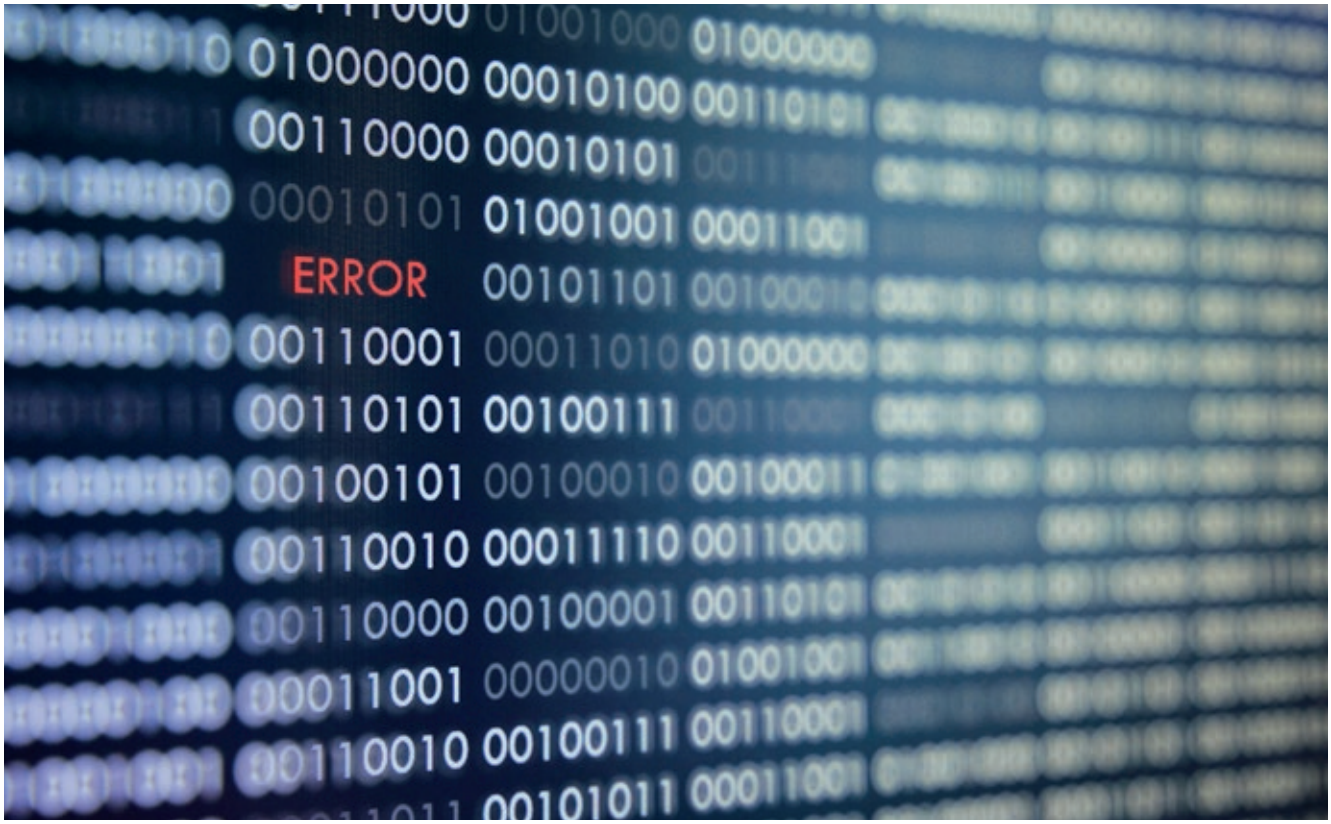
5.1 Sicherheitsrisiken beim Einsatz von KI-Systemen

Das Risiko beim Einsatz von KI-Systemen ist grundsätzlich die Gefahr von Fehlentscheidungen. Die Gründe für Fehlentscheidungen seitens der KI können verschiedenste Ursachen haben. Diese, zunächst von Security-Aspekten unabhängigen Probleme, können in der Standardliteratur zu KI nachgelesen werden [4]. Eine Relation dieser generellen Probleme zu Security Aspekten wird unter den jeweiligen Einzelpunkten diskutiert.

Die aufgezählten Gründe sind zusätzlich den jeweiligen Phasen des KI-Phasenmodells in Abbildung 1 zugeordnet.

Aufzählung der Gründe:

- **KI-Entwurf- und Trainingsphase, unklassifizierte Relevanz der Trainingsdaten:** Bei Trainingsdaten handelt es sich häufig um Massendaten, die idealerweise die Wertespektrum-Kombinationen für jeden von n Parametern



dieser Daten in Grobrastern abdecken, sowie entsprechend der Kombinatorik gelabelt sind. Besonders wichtig sind dabei auch Abgrenzungsdatensätze, bei denen das Labelling bei vergleichsweise geringfügigen Änderungen zu einer anderen Bewertung seitens der KI führen soll. Im oben genannten Beispiel 1 (Oberflächen-Qualitätskontrolle) wären hier Parameter für die Messung der ‚Rauheit' so zu bestimmen, bei denen die Qualitätskontrolle gerade noch den Wert ‚fehlerfrei' zurückliefern würde bzw. gerade schon den Wert ‚fehlerhaft' zurückliefern würde. Sehr häufig wird jedoch der Relevanz der Daten bzgl. der Abdeckung des Wertespektrums nicht genügend Aufmerksamkeit geschenkt. In diesem Falle wird das Wertespektrum einzelner Parameter entweder nicht vollständig abgedeckt oder geht in eine Richtung weit über den Zielkorridor hinaus und bleibt in die andere Richtung weit unterhalb des Maximal-/Minimal-Wertes des gewünschten Zielkorridors.

Neben der generellen Schwierigkeit, das n-dimensionale Raster der Parameter-Werte abzudecken, kann als Angriffsvektor zum Trainingszeitpunkt auch eine Manipulation dieser Daten erfolgen, so dass durch die Manipulation das Relevanzspektrum nicht abgedeckt wäre.

Als Resultat können zur Laufzeit der KI Fehlentscheidungen auftreten für die Grenzbereiche, oder gar Zufallsentscheidungen für Parameterwerte, die sich außerhalb derer der Trainingsdatensätze befinden.

- **KI-Trainingsphase, falsche Trainingsdaten, erzeugt durch bewusste Manipulation der KI:**

  - Vorsätzliches Manipulieren von Trainingsdaten zum Provozieren eines Fehlverhaltens wird mit dem Begriff „Poisoning" bezeichnet.

    Trainingsdaten bestimmen das Verhalten des KNN, werden die Trainingsdaten manipuliert, dann kann sich das NN anders verhalten als erwartet. Bei geschickter Manipulation ist die Manipulation den Trainingsdaten nicht anzusehen und auch das NN verhält sich scheinbar noch wie erwartet. Für bestimmte Eingabedaten wird allerdings ein unerwartetes oder gar unerwünschtes Verhalten auftreten.

    Das ist vergleichbar dem Verhalten von Software, die versteckten, schadhaften Code enthält. Die Software verhält sich in der Regel wie sie soll und der Schaden

ist nicht offensichtlich, unter bestimmten Bedingungen wird allerdings der Schadcode aktiv und schadet. Die Situation im Falle eines KNN dabei vergleichbar mit Closed Source. Es ist sehr schwer den Schadcode zu finden und das schadhafte Verhalten rechtzeitig zu entdecken, da der Source Code nicht analysiert werden kann, denn er liegt nicht vor.

- Das Auftreten von Fehlverhalten in einem deterministischen System mit mehreren Fehlerquellen ist extrem schwierig [5]. Das System verhält sich unter vermeintlich gleichen Bedingungen manchmal richtig und manchmal falsch. Eine 100 %ige Korrelation zwischen Eingabe und Fehlverhalten ist nicht gegeben. Solche Probleme lassen sich in der Regel nur durch systematisches Eingrenzen der Testbedingungen und die schrittweise Analyse des Systemverhaltens finden und beheben (Debugging).

  Für NN gilt die Analyseproblematik in verschärfter Weise, denn ihre Funktionsweise ist von außen betrachtet nicht streng deterministisch (e. g. falsches Erkennen einer Ziffer durch ein NN aufgrund von Pixelrauschen in Ziffernbildern ist für den menschlichen Betrachter nicht nachvollziehbar).

  Gängige Implementierungen von NN bieten eine Historie/Log der Eingabedaten. Mit Hilfe dieser Historie kann das Verhalten des NN reproduziert und analysiert werden. Dies wird u. a. verwendet, um Ursachen des Fehlverhaltens des NN zu identifizieren und zu beheben. Dieser Mechanismus kann auch zur Analyse und zum Eingrenzen von Poisoning verwendet werden.

- Die Gefahr von Poisoning wird prinzipiell auch durch föderiertes Machine Learning reduziert. Dabei werden die Modelle isoliert geschulter KNN in einem übergeordneten KNN-Modell aggregiert. Die einzelnen KNN der Föderation werden dann durch das aggregierte KNN-Modell ersetzt.

  Wird eines der Systeme mit "vergifteten" Lerndaten angegriffen, entfaltet dieses Fehlverhalten nur geringe Wirkung, da es von den korrekten Modellen überlagert wird.

  Föderiertes Machine Learning eignet sich nicht für alle Anwendungen neuronaler Netze.

- **Zwei Risiko-Typen sind zu betrachten:**

  - Die KI-Software wurde im Vortraining manipuliert und damit fehlerhaft in die Produktion eingeschleust;

  - Die KI-Software wird in der Produktion durch Cyberangriffe in der Entscheidung manipuliert; die Cyberangriffe können dabei durch eine einzige Produktionsanlage oder Qualitätsüberprüfungsanlage erfolgen, die ständig falsche Daten an die KI zurückmeldet und somit zu falschen Entscheidungen der KI führt.

- **KI-Trainingsphase, das KNN wird 'übertrainiert'** (Overfitting, siehe Grundbegriffe, Kapitel 7): Beim Training einer KI werden die gelabelten Datensätze häufig mehrfach in Lerniterationen der KI zugeführt, bis die Abweichung beim Testen mit Hilfe eines Validierungsdatensatzes gering wird. Im Falle des übertrainierten Systems, d. h. bei zu häufig durchgeführten Lerniterationen, lernt die KI die Ergebnisse der gelabelten Datensätze quasi auswendig. Das Bewertungsergebnis auf Basis der zugeführten Validierungsdaten, die das System aus dem Training nicht kennt, verschlechtert sich, weil das System beginnt, seine Generalisierungsfähigkeit zu verlieren. Die Überprüfung des Lernvorgangs mit Validierungsdaten liefert den Hinweis, wann das Training abgebrochen werden sollte, um Overfitting zu vermeiden. Bei im Produktivbetrieb auftretenden kleinen Parameterabweichungen im Vergleich zu den auswendig gelernten Regeln kann die KI nach Overfitting nicht mehr gut genug generalisieren und erzeugt Fehlentscheidungen.

- **KI-Trainingsphase, das KNN ist untertrainiert** (Underfitting, siehe Grundbegriffe, Kapitel 6): Die Trainingsdaten des KI-Systems sind zu wenige oder haben eine zu geringe Varianz ihrer Parameterwerte, so dass die KI im Testdatenbetrieb zwar gute Ergebnisse liefert, im Produktivbetrieb jedoch zu sehr generalisiert, die Komplexität der Aufgabe nicht erfassen kann und ggf. einseitig die ihr zugeführten Datensätze immer gleich klassifiziert und nur bei sehr großer Abweichung zu den Trainingsdaten anders klassifiziert.

- **KI-Trainings- und Betriebsphase, falsches dynamisches Lernen:** Im Zusammenhang mit Supervised Machine Learning, bei dem während der produktiven Nutzung des KNN durch Hinzufügen weiterer Daten noch dynamisch hinzugelernt wird (siehe Beispiel 3: Kontrolliertes dynamisches Weiterlernen), kann durch Falschbewertung einer gerade vorherrschenden aktuellen Situation

dauerhaft eine Fehlinterpretation ähnlicher Situationen herbeigeführt werden (Erzeugung von False Negatives/ Positives, s. u.). Dynamisches Lernen ist ein komplexes Thema, das im Allgemeinen nur durch hochspezialisierte menschliche Eingriffe mit hoher anwendungsbezogener Sachkenntnis sinnvoll implementiert werden kann. Die naive Vorstellung, dass intelligente Systeme im Betrieb immer weiter lernen und so immer besser werden, ist grundsätzlich falsch. Die ungefilterte Übernahme von Erkenntnissen des Systems in den eigenen Bestand an Grundwahrheiten kann zum Verlust der Erkennungsfähigkeit des KI-Systems führen.

- **KI-Betriebsphase, veränderte Umgebung im Vergleich zur Lernumgebung:** Grundlegende Risiken entstehen bei vortrainierten Systemen, wenn der Einsatz der KI in veränderten Umgebungen stattfindet, in denen früher oder später Situationen eintreten können, die während der Lernphase nicht in den Beispieldaten enthalten waren. Generalisierung durch KI bedeutet immer eine komplexe Form der Interpolation zwischen bekannten Daten aus bekannten Situationen. Ereignisse, die verglichen mit der Lernumgebung völlig anders sind, kann ein KI-System nicht handhaben.

Die beiden grundsätzlichen Arten von Fehlentscheidungen sind:

- **False Positives:** Die KI liefert bei Bewertungen der Kritikalität eines Zustandes eine hohe Kritikalität zurück, obwohl es sich um eine niedrige Kritikalität handelt. Im Zusammenhang mit Beispiel 1 (Qualitätsprüfung) würde die KI auf einen Produktionsfehler entscheiden, obwohl es keiner ist. In Beispiel 2 (kollaborativ arbeitende Roboter) würde die KI einen bevorstehenden Zusammenstoß der kollaborierenden Roboter- Produkt-Entitäten erkennen, der jedoch nicht stattfinden würde, und ggf. eine entsprechende Reaktion des Systems provozieren, die nicht nötig wäre oder schlimmstenfalls dann erst zu einem Zusammenstoß führen könnte. In Beispiel 3 werden (Cyber-Attacken-) Alarme erkannt, die keine sind.

- **False Negatives:** Die KI liefert bei Bewertungen der Kritikalität eines Zustandes eine niedrige Kritikalität zurück, obwohl es sich um eine hohe Kritikalität handelt. In Beispiel 1 würden vorhandene Qualitätsfehler nicht erkannt. In Beispiel 2 würde ein bevorstehender Zusammenstoß der involvierten Entitäten nicht oder zu spät erkannt. In Beispiel 3 wäre würde eine (Cyber-)Attacke nicht erkannt werden.

Beide Fälle können durch die oben aufgezählten Gründe zum Zeitpunkt des Lernens, oder durch die Veränderung der Umgebung im Produktionsumfeld entstehen.

5.1.1 Folgerisiken: Produktionsausfälle und entsprechende Zusatzkosten

Die Spanne der Folgerisiken aufgrund von Fehlentscheidungen seitens der KI kann als beliebig angenommen werden. In Abhängigkeit von den Prozessen eines Unternehmens und den Folgereaktionen können prinzipiell Produktionsstillstände, Fehlproduktionen oder Zerstörung von Maschinen erfolgen. Wenn z. B. aufgrund eines Fehlalarms (False Positive) automatisch die Industrie 4.0-Entität abgeschaltet wird und die Anzahl der Fehlalarme sehr hoch ist, entsteht möglicherweise eine Unmöglichkeit, die Produktion überhaupt wieder aufzunehmen.

Die auftretenden Sicherheitsrisiken in Produktionsumgebungen unterscheiden sich von der Art des Einsatzes und der Umgebung. Hier ist insbesondere zwischen dem Einsatz statischer und dynamischer KI-Systeme zu unterscheiden, wie oben beschrieben. Generell erhöhen sich die Sicherheitsrisiken bzgl. der Entscheidungen durch eine KI, je offener das System ist und je variabler sich die daraus entstehenden Parameterwerte ergeben.

Hinweis: Die Wahrscheinlichkeit, dass ein solches Risiko durch den Einsatz von KI eintritt, ist nicht zwangsläufig höher als durch die Prüfung fester Muster und die Verwendung anderer Überwachungsmechanismen. In offenen Systemen versagt die Prüfung durch feste Muster nahezu gänzlich, da diese keinerlei (oder nur die per Definition vorgegebenen) Änderungen von Parameterwerten und Kombinationen vorsehen.

5.1.2 Sicherheitsrisiken beim Einsatz statischer KI-Systeme

Unter Bezugnahme auf die o. g. Gründe von Fehlentscheidungen werden die Sicherheitsrisiken mit Bezug auf Beispiel 1 und teilweise Beispiel 3 diskutiert.

- **KI-Entwurf- und -Trainingsphase, unklassifizierte Relevanz der Trainingsdaten:** Im geschlossenen System werden Fehlentscheidungen in einer Startphase einer Produktion relativ schnell durch Nachkontrollen erkannt, so dass bei eventuellen Nicht-Abdeckungen der Parameter-

bereiche (sowohl durch Nicht-Beachtung der Relevanzbereiche als auch durch Manipulation der Trainingsdatensätze) entsprechend nachgelernt oder neu gelernt werden kann. In Beispiel 1 könnte man dann z. B. entsprechende Lerndatensätze hinzufügen, die die Grenzbereiche zwischen ‚fehlerhaft' und ‚fehlerfrei' abdecken bzw. besser beschreiben.

- **KI-Trainingsphase, falsche Trainingsdaten, erzeugt durch bewusste Manipulation der KI:** Im Falle von Beispiel 1 und Beispiel 3 würde eine Manipulation wahrscheinlich mit der Absicht durchgeführt, vorhandene Qualitätsprobleme nicht zu erkennen, um einen hohen Schaden zu provozieren (False Negatives), der erst spät bemerkt wird. Es entstehen die entsprechenden Folgerisiken wie oben beschrieben.

- **KI-Trainingsphase, das NN wird 'übertrainiert':** Im Falle von Beispiel 1 ist das Risiko von Fehlentscheidungen durch ein übertrainiertes System nicht zwangsweise erhöht, so lange die Lerndatensätze bereits alle auftretenden Fälle beinhalteten. Generell ist aber im Produktivbetrieb mit Abweichungen von den Lerndatensätzen zu rechnen, so dass mit Fehlentscheidungen zu rechnen ist, je größer die Varianz der echten Datensätze ist. Ein Sicherheitsrisiko entsteht speziell aus einer bewussten/unbewussten Manipulation der Einsatzumgebung, da dadurch die Anzahl der Fehlentscheidungen massiv in beide Richtungen steigt und die Produktivität sinkt.

- **KI-Trainingsphase, das NN ist ‚untertrainiert':** Im Falle von Beispiel 1 ist das Risiko des untertrainierten Systems gering, da dies schon bei der Inbetriebnahme einer Industrie 4.0-Anlage sofort wegen der hohen Anzahl von Fehlentscheidungen auffällt und nicht erst durch bewusste/unbewusste Manipulation der Anlage.

- **KI-Trainings- und Betriebsphase, falsches dynamisches Lernen:** Ein Sicherheitsrisiko besteht dann, wenn während des Zuführens weiterer gelabelter Datensätze bewusst falsch gelabelte Datensätze untergebracht werden, sei es durch einen Angreifer von außen oder durch einen Innentäter. Das Ziel wäre hier ebenfalls, vor allem False Negatives zu provozieren, damit die Manipulation möglichst spät entdeckt wird und der Schaden dadurch maximiert wird.

- **KI-Betriebsphase, veränderte Umgebung im Vergleich zur Lernumgebung:** Bei Beispiel 1 könnten u. a. folgende Umgebungsveränderungen im Vergleich zur Lernumgebung sowohl zu False Positives als auch zu False Negatives führen:

  - Veränderte Oberflächen von Werkstücken können zu Reflexionen führen, die die Bilderkennung blenden.

  - Die Lichtverhältnisse in der Produktion sind nicht mit dem „Lernlabor" identisch.

Bei Beispiel 3 weicht die Einsatzumgebung fast grundsätzlich von der Laborumgebung ab (unternehmensspezifische Umgebungen), so dass Fehlentscheidungen einer KI oder auch einer festen Regel immer vorhanden sind. Diese können im Laufe des Hinzulernens massiv reduziert werden.

5.1.3 Sicherheitsrisiken beim Einsatz dynamischer KI-Systeme

Die in diesem Kapitel diskutierten Sicherheitsrisiken sind zum einen dieselben wie die Risiken beim Einsatz statischer KI-Systeme, es kommen jedoch entweder weitere Qualitäten der diskutierten Risiken hinzu oder es können zusätzliche Risiken auftreten.

Unter Bezugnahme auf die o. g. Gründe von Fehlentscheidungen werden die Sicherheitsrisiken mit Bezug auf Beispiel 2 (kollaborativ arbeitende Roboter) und teilweise Beispiel 3 diskutiert.

- **KI-Entwurf- und Trainingsphase, unklassifizierte Relevanz der Trainingsdaten:** Im Vergleich zu geschlossenen Systemen sind die Grenzen der verschiedenen Parameterwerte eines offenen Systems nicht immer vollständig vordenkbar, die Anzahl der Parameter und deren Wertespektren oftmals größer, so dass zwangsweise die Grobraster der Abdeckung der Parameterwerte in den Trainingsdaten ebenfalls größer werden. D. h. die Gefahr der Nicht-Abdeckung von Parameterwerte-Bereichen steigt in offenen Systemen signifikant an. Auch eine bewusste Manipulation von Trainingsdaten kann nachträglich nur sehr schwer bemerkt werden, da für den Menschen die Einschätzung, warum sich die KI in eine bestimmte Richtung entschieden hat, auch aufgrund der riesigen Kombinatorik von Werten, noch unmöglicher wird als das schon bei geschlossenen Systemen der Fall ist.

- **KI-Trainingsphase, falsche Trainingsdaten, erzeugt durch bewusste Manipulation der KI:** Es treten dieselben zusätzlichen Risiken auf, wie bei manipulativer Veränderung der Umgebung.

- **KI-Trainingsphase, das KNN wird übertrainiert/untertrainiert:** Im Falle von Beispiel 2 ist die Gefahr des Über- als auch des Untertrainings im Produktivbetrieb gering, und damit auch die Gefahr der Ausnutzung dieser Aspekte durch einen Angreifer. Ein Über- als auch ein Untertraining der Anlage würden bereits im Testbetrieb erkannt, weil bereits hier die Dynamik des Systems zu Fehlreaktionen führen würde, und ein Nachlernen direkt getriggert werden muss.

- **KI-Trainings- und Betriebsphase, falsches dynamisches Lernen:** Es existiert das gleiche Szenario, wie bei statischen/geschlossenen KI-Systemen.

- **KI-Betriebsphase, veränderte Umgebung im Vergleich zur Lernumgebung:** In Beispiel 2 verändert sich die Umgebung durch die Dynamik des Systems per Definition massiv; beispielsweise werden weitere Roboter in der Produktion eingesetzt, die es zum Zeitpunkt des Vortrainings noch nicht gab. Es besteht daher ständig die Gefahr, dass die KI nicht generisch genug (vor-)trainiert wurde, um mit den neuen Situationen umzugehen, d.h. das Risiko von Zusammenstößen, wie in diesem Beispiel dargestellt, steigt. Ein Sicherheitsrisiko entsteht speziell, wenn im Produktivumfeld Maschinendaten manipuliert werden oder Daten von nicht vorhandenen Maschinen erzeugt und eingeschleust werden. Dadurch kann sowohl eine Provokation von False Positives (bevorstehender Unfall wird erkannt, obwohl nicht stattfindend) als auch False Negatives (echter bevorstehender Unfall wird nicht erkannt) stattfinden, je nachdem auf welche Effekte der Angreifer abzielt. Effekt 1 kann die dauerhafte Störung durch Fehlalarme sein, wodurch die Anlage zum Stillstand gebracht werden kann. Effekt 2 kann die Provokation eines sehr großen Unfalls sein, der nicht erkannt wird und zu einem großen finanziellen Schaden führen kann.

5.2 Sicherheitsmaßnahmen beim Einsatz von KI/Sicherheitsmaßnahmen durch KI

Für fast alle technischen Systeme gilt: Die Menge aller möglichen Eingaben kann nicht getestet werden, da sie zu groß ist.

Bei deterministischen Systemen testet man daher Randwerte und Extremwerte, eine definierte Reihe oder Abfolge von Testwerten sowie einige zufällige Eingaben (Monkey Test). Da dem System ein wohldefiniertes Verhalten unterstellt werden kann, kann mit diesen Tests auf das korrekte Verhalten für alle erlaubten Eingaben (Definitionsmenge) geschlossen werden.

Für KNN hat diese Methode nicht die gleiche Aussagekraft, da KNN mit bestimmten Wahrscheinlichkeiten auf Eingaben reagieren. Diese Eingaben variieren in der Regel allein aufgrund der vorgelagerten Sensorik (z.B. dasselbe Bild bei unterschiedlichen Umgebungsbedingungen, Licht, Temperatur und Rauschen des Bildsensors).

Dennoch greifen die gleichen Sicherheitsmaßnahmen deterministischer Systeme und äquivalent gilt daher bei der Nutzung von KNN zu beachten:

- Verwende nur vortrainierte KNN Modelle aus vertrauenswürdigen verifizierten Quellen und prüfe deren Integrität.

- Verwende eigene Trainingsdaten oder nur solche aus vertrauenswürdigen verifizierten Quellen.

- Prüfe das Verhalten des KNN nach jeder Änderung am System.

- Prüfe regelmäßig das Verhalten des KNN – möglichst auch unbedingt.

- Zusätzlich können Verhaltensstichproben durch menschliche Mitarbeiter verifiziert werden. Dies ist eine bewährte Strategie sehr großer IT-Serviceanbieter für die Prüfung und Schulung von KNN – e.g. „Mechanical Turk".

Das KNN kann auch zur Sicherheitsanalyse verwendet werden. Dazu wird einem KNN das „normale" Verhalten von Systemen trainiert. Dieses KNN überwacht dann die Systeme und meldet Anomalien. Erfolgreiche KNN-Systeme für die Sicherheitsüberwachung von IoT-Geräten wurden dazu als föderierte KNN implementiert. False Positives, also Fehlalarme, müssen bei Sicherheitsüberwachung möglichst vermieden werden. Das KNN erfordert zur Analyse von False Positives in der Regel eine Korrelationshistorie zwischen Rohdaten, Features und KI-Verhalten, so dass hier auch die Nachvollziehbarkeit von KI-Entscheidungen erfolgsentscheidend ist.

6. Abschließende Bemerkungen

Dieses Papier entstand mit dem Ziel und dem Anspruch, Chancen und Risiken, die aus dem Einsatz von Künstlicher Intelligenz im Hinblick auf Sicherheitsaspekte in der Industrie 4.0 erwachsen, zu erklären. Das Autorenteam verfolgte dabei das Ziel, den Stand der Technik zumindest in Grundzügen darzustellen und Ratschläge zur praktischen Umsetzung zu geben. Im Vordergrund standen dabei die modernen Aspekte der KI, die sich erst seit rund fünf Jahren ergeben haben, und in denen immer klarer wurde, wie groß die Bedeutung der Erklärbarkeit von KI geworden ist. In dieser Hinsicht geht das Papier auch deutlich über die erste Publikation hinaus, die im April 2019 veröffentlicht wurde [6].

Sobald der grundlegende Rahmen des maschinellen Lernens überschritten wird und diejenigen Einflüsse in den Fokus treten, in denen sich offenbart, dass die großartigen Leistungen des ML mit dem Verlust an Erklärbarkeit einhergehen, entsteht erheblicher Bedarf an kompetenter Beratung. Das vorliegende Papier kann nur in groben Zügen diesen Beratungsbedarf skizzieren. Es kann aber kein Ersatz für individuelle qualifizierte Dienstleistung sein. Für viele der angesprochenen Problembereiche gibt es heute (noch) keine universellen und gleichzeitig einfachen Antworten. Dies sind heute noch Bereiche weitergehender Forschung. Doch die tägliche Praxis ist, dass die meisten der angesprochenen Problemfelder adäquat adressiert werden können. Eine Risikobewertung muss im Einzelfall erfolgen.

7. Grundbegriffe der KI

Vortrainierte Systeme sind eine Technik, um **Transfer-Lernen** anzuwenden. Dabei werden Schichten am oberen Rand eines Neuronalen Netzes entfernt und durch neue Schichten mit anderen Klassen ersetzt. Nur diese neuen Schichten werden trainiert, während die unteren Schichten fixiert bleiben. Moderne tiefe Netze, insbesondere tiefe CNN zur Bilderkennung, bauen in ihrem Training eine Hierarchie von Parameterwerten in ihren Schichten auf – die automatisch erzeugten Features in unteren Schichten dienen zur Erkennung von Kanten und anderen Grundstrukturen. Sie werden in höheren Schichten zur Konstruktion komplexerer Features genutzt. Deshalb ist es naheliegend, statt in einem neuen Training alle Gewichte neu zu bestimmen, die unteren Schichten eines bereits trainierten Netzes zu übernehmen und nur die letzte(n) Schicht(en) neu zu bestimmen. Dieser Vorgang kommt mit vergleichsweise wenig Daten schnell zu guten Resultaten. Beispielsweise können metallische Oberflächen, die von Kugellagern und Kurbelwellen gelernt wurden als Basis für die Erkennung von Schrauben dienen, für die wenige Bilder der jeweiligen Schrauben nötig sind.

Der Vorgang der Einstellung der freien Parameter eines KI-Systems wird als **Training** bezeichnet. Dabei wird der Abstand zwischen den Schätzwerten, die das System liefert, und den dazu passenden Labels iterativ durch Anpassung dieser Parameter, die bei Neuronalen Netzen Gewichte heißen, verringert. Der Gradient dieses Abstands als Funktion der Gewichte zeigt in die Richtung des steilsten Anstiegs. Um die Gewichte zu verbessern, müssen also Schritte in die Richtung des nativen Gradienten vollzogen werden. Unter geeigneten Voraussetzungen über die Abstandsfunktion konvergiert das Verfahren des **Gradientenabstiegs** gegen Null. Wird aus den Parametern mehrerer KI-Systeme ein neuer gemeinsamer Parametersatz berechnet spricht man von föderiertem Lernen. **Föderiertes Lernen** bietet Sicherheits- und Privacy-Vorteile.

Die Kennzeichnung, die Trainingsdaten einer Klasse zuordnet, heißt **Label**. Um mit Labels rechnen zu können, werden diese in der sogenannten 1-Hot-Darstellung angegeben. In den durchnummerierten Klassen ist das Label der Klasse i ein Vektor, der in der Komponente i eine 1 und an allen anderen jeweils eine 0 enthält. Normiert man den Ergebnisvektor eines KI-Systems so, dass alle Koordinaten positiv sind und als Summe 1 ergeben, dann kann man die jeweiligen Werte als Wahrscheinlichkeiten auffassen, mit der eine Eingabe zu den entsprechenden Klassen gehört. Eine korrekte Schätzung, d. h. höchste Wahrscheinlichkeit bei dem tatsächlichen Label eines Elements des Test-Sets, heißt Treffer. (nach Top-1 Kriterium). Ist das tatsächliche Label unter den fünf höchsten Wahrscheinlichkeiten, spricht man von einem Treffer nach Top-5 Kriterium.

Visuelle Wahrnehmung wird in **Szenen-, Bild- und Objekt-Erkennung** unterteilt. Szenenerkennung richtet sich auf Darstellungen, die keine dominanten Objekte enthalten, beispielsweise einen Marktplatz, einen Hafen oder ein Fußballspiel. Bilderkennung richtet sich auf eine Darstellung mit einem einzigen klar dominierenden Objekt: Mein Auto, mein Boot, meine Katze. Objekterkennung und Lokalisierung richtet sich auf Darstellungen, die mehrere Elemente aus einer oder mehreren bekannten Klassen enthalten und diese durch Einrahmung in einem begrenzenden Rechteck markieren. Dies kann beispielsweise eine Abbildung einer Straßenkreuzung sein, in der alle Fußgänger, Radfahrer, PKW, Busse, u. ä. in beschrifteten **„Bounding Boxes“** eingeschlossen sind. **Gesichtserkennung** ist ein Spezialfall, im dem die Merkmale eines Gesichts als eigenständige Klassen trainiert wurden und in einem Portrait mit Bounding Boxes identifiziert werden. Es ist dann möglich aus den relativen Positionen im Portrait Zuordnungen zu Datenbeständen zu machen. So kann über die Aussage „das ist ein Gesicht“ hinausgehend ein bestimmtes Gesicht oder ein Mitglied einer bestimmten Gruppe identifiziert werden.

Anders als in der ersten Phase von KI (1956 bis Mitte der 80er Jahre) liegt der aktuelle Schwerpunkt (Machine Learning) auf der Extraktion von Strukturinformation aus **Datensätzen** mit dem Ziel der Zuordnung von Daten zu Begriffen oder numerischen Größen (Klassifikation, Regression), der Erkennung verborgener Merkmale oder Systematiken und der Entwicklung von Strategien zur Erreichung definierter Ziele. Der geschäftlich wichtigste Teil der KI besteht heute in den vielfältigen Varianten der Klassifikation: Zu einem bislang unbekannten Eingangssignal aus einem definierten Spektrum (Bild, Grafik, Ton, Wort, Text, Video, zeitliche Abfolge von numerischen Werten, ...) soll das KI-System Aussagen treffen, zu welcher Klasse von Daten das Eingangssignal mit höchster Wahrscheinlichkeit passt. Was ist beispielsweise auf einem 600X800 Pixel RGB-Bild abgebildet: Ein Auto, eine Katze, eine Schraube, eine Beule im Blech eines Kotflügels, ...? Die Fähigkeit, solche Fragen treffsicher zu beantworten erlangen KI-Systeme durch **überwachtes Lernen** (**Supervised Learning**). Grundlage für Supervised Learning sind Daten, die mit einem **Label**, also einer Kennzeichnung

des Inhalts versehen sind. Ausgehend von einer Einteilung der Daten in **Klassen**, in denen alle Datenelemente pro Klasse dasselbe Label tragen, „lernt" das KI-System durch den Aufbau einer Näherungsfunktion **(Approximation)** zu unterscheiden und korrekte Zuordnungen zu den definierten Klassen zu finden. Da sowohl der Lernvorgang als auch die spätere Erkennung reine Rechenoperation sind, müssen alle Daten in Zahlen umgewandelt werden. Beispielsweise kann man ein Rasterbild wie ein Foto als Matrix darstellen, deren Koeffizienten die Tripel der RGB-Werte der Farben der Pixel sind. Die Labels werden in der sogenannten **„1-hot"-Darstellung** als Vektoren dargestellt, die an der jeweiligen Nummer der Klasse, der dem Inhalt entspricht, eine 1 aufweist und sonst nur aus Nullen besteht. Damit kann das System rechnen: Wenn es einen Ausgabevektor erzeugt, der die Wahrscheinlichkeiten enthält, dass das Eingangsmuster den jeweiligen Klassen mit ihren Nummern entspricht, dann kann man diesen mit einem solchen Vektor vergleichen, der 100 % für eine bestimmte Klasse und 0 % für alle anderen angibt. Man kann das System danach so verändern, dass der Abstand zwischen eigener Schätzung und Label aus dem Datensatz (der Fehler der Schätzung) kleiner wird. Man benutzt dazu die **Kettenregel der Differentialrechnung** und verändert im Vektorraum der **Gewichte** der Fehlerfunktion alle Parameterwerte in Richtung des negativen Gradienten **(Gradientenabstieg)**. Dies definiert einen Weg, in dem der Fehler als Funktion der Gewichte in einem Iterationsprozess gegen Null konvergiert und nicht bekannte Labels korrekt geschätzt werden **(Vorhersage, Prädiktion)**. Das KI-System erinnert sich also nicht an vorher gesehen Labels sondern hat eine Methode der **Generalisierung** gefunden: Es muss nicht den Hund des Nachbarn wiedererkennen, um zu sagen: Das ist mit 99 %iger Wahrscheinlichkeit ein Hund.

Um den Umfang der Menge an Trainingsdaten zu erhöhen, wird oft eine sogenannte **Augmentierung** durchgeführt: Die einzelnen Beispiele werden dabei geringfügig verschoben, um zusätzliche Daten zu erzeugen, die zum selben Label passen. Drehung und die Addition von Rauschen sind weitere Möglichkeiten. Wichtig ist dabei, die relativen Häufigkeiten im Datenbestand nicht zu verändern, weil sonst zusätzliche Verzerrungen (Biases) entstehen.

Um mit einem fertig vorbereiteten Datensatz einen Trainingslauf durchzuführen, wird dieser zunächst in drei Teilmengen aufgeteilt: Das **Trainingset** (ca. 80 %), das **Validierungsset** und das **Testset** (jeweils ca. 10 %). Mit stochastischer Reihenfolge und Gruppierung werden alle Elemente des Trainingset genau einmal für den Gradientenabstieg benutzt – dies ist eine **„Epoche"** des Trainings. Nach jeder solchen Epoche wird dann der durchschnittliche Fehler auf dem Validierungsset ermittelt. Dabei kann man beobachten, dass dieser Fehler bis zu einem gewissen Punkt sinkt, dann jedoch wieder zu steigen beginnt. Gleichzeitig schrumpft der Fehler, den man im Trainingset ermitteln kann, weiter. Das bedeutet, dass das System dazu übergeht, die Trainingsdaten zu erinnern, aber gleichzeitig die Fähigkeit der Generalisierung, also das Erkennen der Klassen unbekannter Daten, zu verlieren beginnt. Dieser Zustand wird als **Overfitting** bezeichnet. Er tritt besonders dann auf, wenn die Zahl der Daten gering, die Zahl der Parameter im System aber hoch ist. Sobald der Fehler im Validierungsset steigt, wir der Trainingslauf abgebrochen.

Es gibt diverse heuristisch gewählte Parameter, die sogenannten **„Hyperparameter"**, die den Trainingslauf beeinflussen. Die Schrittweite **(„Lernrate")** im Gradientenabstieg und deren Veränderung während der Iteration, die zeitweise Fixierung der Gewichte einzelner Neuronen **(„Dropout")**, Umfänge periodisch durchgeführter **Regularisierungen** von Gewichten und ähnliche Parameter beeinflussen das Lernergebnis. Am Ende werden die Fehler aller ermittelten Varianten auf dem Testset bestimmt. Die darin enthaltenen Gewichte stellen das Endergebnis des Trainings dar, der verbleibende **Fehler auf dem Testset** ist das Maß für die erzielte Präzision der Klassifikation. Die Aufspaltung zwischen Test- und Validierungsset ist notwendig, um das System davon abzuhalten, die Hyperparameter zu lernen. Mit den Testdaten wird das System erst ganz am Ende des Vorgangs konfrontiert, sie sind also bis zum Schluss unbekannt. Je größer die Zahl der Hyperparameter, desto höher ist der Rechenaufwand der insgesamt notwendigen Trainingsläufe. So entstehen die oft verblüffend hohen Aufwände der Berechnung komplexer Netze mit vielen und breiten Schichten sowie mit vielen Kombinationsmöglichkeiten von Hyperparametern.

8. Referenzen

[1] https://arxiv.org/pdf/1606.07461.pdf

[2] Bild: https://pixabay.com/de/photos/katze-haustier-tier-grau-hauskatze-3261420/, (2015)
Copyright, Bildrechte https://pixabay.com/de/service/terms/, (2019)

[3] https://arxiv.org/pdf/1605.07277.pdf

[4] Stuart Russell, Peter Norvig „Artificial Intelligence, A modern Approach“, Global Edition, (2018)

[5] https://arxiv.org/pdf/1606.07461.pdf, (2017)

AUTORINNEN UND AUTOREN

Markus Heintel, Siemens AG | Dr. Detlef Houdeau, Infineon Technologies AG | Dr. Wolfgang Klasen, Siemens AG | Dr. Bernd Kosch (Leitung), Industrie-KI GmbH | Markus Ruppert, KOBIL Systems GmbH | Dr. Michael Schmitt, SAP SE | Thomas Walloschke, Fujitsu Technology Solutions GmbH | Dr. Thomas Wille, NXP Semiconductors Germany GmbH

Diese Publikation ist ein Ergebnis der UAG Künstliche Intelligenz der Arbeitsgruppe Sicherheit vernetzter Systeme der Plattform Industrie 4.0.

www.plattform-i40.de