Design-Leitlinien für „Roboter lernt vom Menschen“-Szenarien hinsichtlich Sicherheitsaspekten

des Kooperationsprojekts

CyberProtect

Baden-Württemberg
MINISTERIUM FÜR WIRTSCHAFT, ARBEIT UND WOHNUNGSBAU

Gefördert von Ministerium für Wirtschaft, Arbeit und Wohnungsbau Baden-Württemberg

Aktenzeichen Zuwendungsbescheid:
3-4332.62-FZI/53

Inhalt

Abbildungsverzeichnis

Änderungshistorie			
Rev.	Datum	Beschreibung	Autor
0.5	28.04.19	Erste Version	Erik Krempel
0.8	14.05.19	Update der Begriffe nach IEC 62443	Erik Krempel
0.9	26.07.19	Imitations Lernen und mögliche Angriffe	Atanas Tanev
1.0	30.07.19	Review, Layout und kleine Korrekturen	Erik Krempel

Abkürzungsverzeichnis und Glossar

A3C	Asynchronous Actor-Critic Agents: Eine Form des Lernens mittels RL
BC	Behavioral Cloning
CDG	Common dominant adversarial examples generation method
DDos	Distributed Denial-of-Service: Verteilter Angriff auf ein Rechnersystem mit dem Ziel dieses zu Überlasten
IEC	International Electrotechnical Commission
IL	Imitation Learning
IP	Intellectual Property
MDP	Markov Decision Process definiert
RGB-D	Red Green Blue-Depth: Kameras die neben den Farbdaten noch ein Tiefenbild liefern
RL	Reinforcement Learning
ROS	Robot Operating System: Ein verbreitetes Framework für Industrierobotik
SPA	Start point-based adversarial attack

Einleitung

Die vierte industriele Revolution und besonders Stichworte wie „Losgröße 1“ fordern von Produktionsanlagen, dass sie zukünftig deutlich schneller neu konfigurierbar sind, als dies heute geschieht. Ein aktueller Forschungsbereich um diese Flexibilität zu erreichen sind Systeme, die nicht mehr von Experten programmiert werden müssen, sondern auch durch einfaches „Vormachen“ neue Fähigkeiten lernen.

In diesem Dokument soll eine erste Betrachtung dieser Systeme aus dem Blickwinkel der Sicherheit vorgenommen werden. Untersucht wird, welche ggf. neuen Sicherheitsprobleme dadurch aufkommen und was Angreifer motiviert Systeme in der Produktion anzugreifen. Im Abschluss werden einige allgemeine Design-Leitlinien, wie zukünftig lernende Verfahren in die Produktion eingebettet werden können, vorgestellt.

Definitionen

Der im deutschen verwendete Begriff **Sicherheit** ist von seiner Bedeutung sehr breit und je nach Einsatzgebiet nicht eindeutig. Deshalb wird er in die drei, im englischen Sprachraum gebräuchlichen, Teilbereiche Safety, Security und Privacy unterteilt.

- **Safety:** Der Begriff Safety, oft auch funktionale Sicherheit genannt, umfasst unwillentliche Bedrohungen, wie sie durch zufällige Ereignisse (bspw. Erdbeben) oder Unachtsamkeit (bspw. bei der Entwicklung oder dem Betrieb von Systemen) entstehen. In unseren Szenarien bedeutet Safety insbesondere den Schutz eines Menschen vor Gefährdungen durch Funktion und Fehlfunktion eines technischen Systems.

- **Security:** Der Begriff Security betrachtet Gefahren, die von einem aktiven, intelligenten Angreifer ausgehen. Diese Gefahren können sowohl Zweck (schadensorientierter Angreifer) als auch Mittel (gewinnorientierter Angreifer) sein. In unseren Szenarien bedeutet Security insbesondere den Schutz eines Systems vor absichtlichen Angriffen durch Menschen.

- **Privacy:** Privacy wird oft als Teilmenge der Security betrachtet, die sich mit aktiven Angriffen gegen Persönlichkeitsrechte bzw. personenbezogene Daten beschäftigt. Moderne Ansätze im Datenschutz, etwa „Privacy by Design“ zielen allerdings auch darauf ab, gegenüber einer unbeabsichtigten Bedrohung personenbezogener Daten durch Unachtsamkeit wirksam zu sein. In unseren Szenarien bedeutet Privacy insbesondere Schutz vor der Verletzung der informationellen Selbstbestimmung.

Für die Privacy ist eine Datenverarbeitung nur dann relevant, wenn es sich um **personenbezogene Daten** von betroffenen Personen handelt. Klar ist, dass wenn ein System beispielsweise mit einer Kamera seine Umgebung beobachtet und sich darin auch Menschen aufhalten, es sich hierbei um personenbezogene Daten handelt. Gerade für die Produktion ist es auch typisch, dass versteckte personenbezogene Daten anfallen. Speichert ein System beispielsweise genau, wann welcher Mitarbeiter das System bedient, könnten diese Daten zu einer verbotenen Mitarbeiterüberwachung genutzt werden. Grundsätzlich ist die Erfassung und Verarbeitung personenbezogenen Daten nicht verboten. Unter dem Europäischen Datenschutzrecht ist es aber wichtig, die Datenverarbeitung zu erkennen, zu dokumentieren und entsprechend auszugestalten.

Szenario “Roboter lernen vom Menschen”

Intelligentes Verhalten automatisiert zu erzeugen ist eine sehr schwierige aber erstrebenswerte Aufgabe in der Robotik. Es erlaubt nicht nur eine intuitive Programmierung von Robotern, sondern erreicht zudem außerordentlich natürliche Bewegungsmuster, besonders für Einsatzgebiete, die mit dem Menschen zusammenarbeiten (Mensch-Roboter-Kollaboration). Beim sogenannten Imitation Learning (IL) beobachtet das System wie eine gegebene Aufgabe von einem Experten gelöst wird, versucht das Beobachtete zu generalisieren und anschließen nachzuahmen.

IL erleichtert die Roboterprogrammierung dadurch, dass kein explizites Model der Aufgabe benötigt wird. Stattdessen wird der Roboter durch das Vorwissen des Experten geführt. Entsprechend gilt, dass der Nutzen dieser Methode mit der Schwierigkeit der Aufgabe steigt und in manchen Fällen IL die einzige umsetzbare Lösung einer komplizierten Problemstellung ist. Daher wird es als Schlüsseltechnologie vieler moderner Roboteraufgaben angesehen (Schall).

Beim Einsatz von IL kommen verschiedene Designfragen auf:

- Wer ist der Experte?
- Wie werden Features repräsentiert?
- Was wird imitiert?
- Wie soll die Policy präsentiert, gelernt und verbessert werden?

Imitation Learning (IL)

In dieser Arbeit werden wir die folgenden Begrifflichkeiten nach (Hussen) verwenden:

- In IL löst ein Agent eine bestimmte Aufgabe, indem er eine Policy basierend auf Demonstrationen erlernt.
- Ein Agent beobachtet und interagiert mit der Umgebung.
- Eine Policy bildet einen Zustand auf eine Aktion ab.
- Eine statische Policy ignoriert den Zeitparameter und lernt eine Policy für alle Schritte in einer Sequenz
- Eine Demonstration ist eine Sequenz von Zuständen und Aktionen A (s_t, a_t).

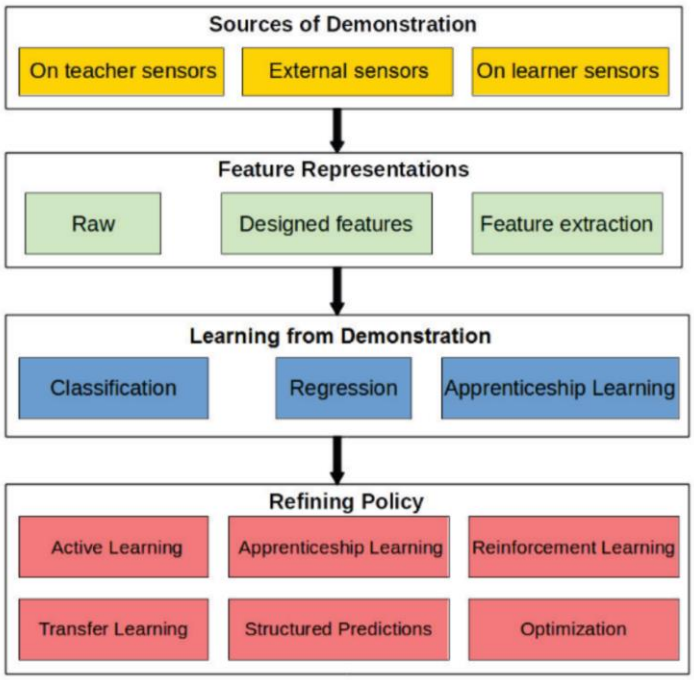

Abbildung 1: (Hussen) Vier Schritten von Imitation Learning

IL kann in vier Phasen eingeteilt werden, wie sie in Abbildung 1 dargestellt sind.

1. **Vorbereitung und Sensoren**
   Zunächst müssen die Daten der Demonstrationen durch verschiedene Sensoren aufgenommen werden.

2. **Feature Repräsentation**
   Die empfangenen Daten werden weiterverarbeitet, wobei die für das Verständnis der Aufgabe wichtigen Features extrahiert und abgebildet werden.

3. **Lernen aus Demonstration**
   Mit den extrahierten Features kann der Lernvorgang beginnen. Dabei werden Lernansätze wie Behaviour Cloning (Pajarinen) verwendet, welche es dem Roboter ermöglichen eine Aufgabe zu lernen und zu wiederholen.

4. **Verfeinerung der Policy**
   Die letzte Phase ist eng mit der vorigen verbunden. Es werden Lernprozesse

mit Reinforcement Learning Algorithmen (Sutton und Andrew) kombiniert, um das gelernte in der Anwendung ständig zu verbessern.

In der aktuellen Forschung ist der RL Ansatz eine der weitesten verbreiteten Methoden für die Thematik Lernfähige Roboter. Unsere Analyse bezüglich möglicher Sicherheitsprobleme in der Lernphase konzentriert sich auf bekannte Angriffe für verschiedene Reinforcement Learning Algorithmen.

Schritt 1: Vorbereitung und Sensoren

Um von einer Demonstration etwas zu lernen, müssen vor Beginn Informationen über den Lehrer/Experten, die Umgebung und den Zustand der Aufgabe gesammelt werden. Im Allgemeinen muss der Lehrer zuerst identifiziert werden. Dafür ist ein Modell des Lehrers notwendig. Zusätzlich müssen alle Objekte in dem Szenario mit ihrer Lage und ihren Konfigurationen identifiziert werden. Anhand all dieser Informationen kann dann ein Modell der Umgebung und Unterscheidungen der Aufgaben erstellt werden.

In der Praxis werden drei Arten von Sensoren benutzt:

1. Die erste Art von Sensoren sind direkt am Lehrer angebracht und sind auch als **„on teacher“** Sensoren bekannt. Diese können als aktiv oder passiv klassifiziert werden. Üblicherweise werden Sensoren zur Bewegungserkennung benutzt, beispielsweise in Form eines Handschuhs, eines Ganzkörperanzugs oder Markierungen wie in (Tung und Kak). Diese liefern akkurate Daten sowie Schätzungen der Körperhaltung und vereinfachen so den Vorverarbeitungsschritt.

2. Die zweite Art sind Sensoren für Lernende, oder **„on learner“** Sensoren genannt. Diese Sensoren befinden sich an dem Lernenden. Das Aufnehmen einer Demonstration mit dieser Art von Sensoren, stellt eine vielseitigste Lösung dar. Es ist üblich einen visuellen Input in RGB-D Format durch Tiefensensoren wie zum Beispiel Kinect (Microsoft) oder RealSense (Intel) Kameras aufzunehmen. Die IL Algorithmen erhalten die semantischen Informationen über die Aufgabe, welche aus den Momentaufnahmen jeder einzelnen Szene entnommen werden. In diesem

Kontext ist das Aktivitäten-Tracking eine der Kernherausforderungen dieser Herangehensweise. Eine menschliche Körperhaltung aus Aufnahmen des Skeletts abzuleiten ist immer noch Gegenstand der aktuellen Forschung, hierbei wurden in (Cao) wichtige Erkenntnisse präsentiert. Zusätzlich muss aus den verarbeiteten visuellen Sensorinformationen außerdem der Zusammenhang von Aktionen und Zuständen sowie die Umgebung selbst erkannt werden.

3. Zuletzt es auch weniger übliche Methoden wie das Benutzen von **externen** Sensoren. In diesem Fall werden kontextsensitive Informationen durch Beobachtungen von außerhalb gewonnen. Diese Sensoren sammeln Informationen über die gesamte Umgebung, möglicherweise auch über den Lernenden selbst. Das Hauptproblem hierbei ist, dass die Daten des Lerners vom Lernvorgang entkoppelt werden müssen.

Herausforderungen

Es gibt vier große Herausforderungen zu bewältigen, unabhängig von der Art der verwendeten Sensoren.

1. Da bei der Aufnahme von Daten Sensorfehler, Verzerrungen und Unschärfen häufig sind, müssen übliche Verfahren zur Verzerrungsreduktionen in der Vorverarbeitung angewendet werden, beispielsweise Filter.

2. Während der Ausführung der Aufgabe muss das Korrespondenzproblem gelöst werden. Da der Lehrer und der Lernende nicht zwangsweise die gleichen Bewegungsmöglichkeiten haben, müssen Aktionen entsprechend übersetzt werden. Dabei sind Freiheitsgrade, Gelenke und Bewegungsabläufe zu betrachten. Frühere Ansätze (Englert) versuchten die Bewegungsbahnen des Roboters mit den Demonstrationen anzugleichen. Neuere Arbeiten vereinfachen allerdings den Prozess aus Entwicklersicht, indem sie hauptsächlich auf End-to-End Ansätzen beruhen, wie zum Beispiel in (C. Finn).

3. Unterschiede zwischen den Aufgaben von Lehrer und Lerner müssen auch betrachtet werden. Diese ergeben sich aus den Interaktionen mit unterschiedlichen

Umgebungen und neuen Hindernissen. Um diesem Problem zu lösen ist eine gewissen Fähigkeit zur Generalisierung nötig.

4. Unzureichende Demonstrationen können dazu führen, dass die Tiefe der Aufgabe nicht in vollem Umfang aufgenommen und verstanden wird.

Schritt 2: Feature Repräsentation

Egal ob Kameras, Tiefensensoren oder verschiedene andere Systeme genutzt werden, typischerweise werden viele verschiedene Sensorarten gleichzeitig eingesetzt, hieraus entsteht das Problem von hoher Dimensionalität und hoher Korrelation zu den Sensordaten. Da die eingefangene Umgebung unmittelbar komplex ist, müssen für eine angemessene Repräsentation effiziente Vorverarbeitungsschritte eingesetzt werden, beispielsweise werden häufig Techniken zur Reduktion der Datendimensionalität verwendet. In diesem Fall vereinfacht es den Lernprozess, sich auf die wichtigen Aspekte zu konzentrieren und angemessene und gleichzeitig unterscheidbare Informationen zu vermitteln. Solche Merkmalsextraktionen können mit den folgenden Darstellungen erzielt werden:

- **Raw Features**, welche direkt im Imitation Learning Algorithmus verwendet werden. Aus diesem Grund sollte diese Methode nur angewendet werden, wenn die Daten selbst hoch relevant, unverzerrt, niedrig dimensioniert und nicht komprimierbar sind.

- **Designed Features**, die das Expertenwissen aus den Rohdaten für eine bestimmte Aufgabe extrahieren. Ein solches Beispiel ist (Billard und Mataric), bei dem relevante Markierungspositionen verfolgt werden, damit eine menschliche Armbewegung wiederhergestellt wird.

- **Extracted Features**, bei denen die automatische Verarbeitung von Rohdaten abgebildet ist, sodass der Zuordnungsprozess von korrelierten Features von höherer zu niedrigerer Dimension erfolgt.

Schritt 3: Lernen aus Demonstration

Laut (Osa) können beim Umgang mit Imitation Learning und bei der Reproduktion einer nahezu optimalen Policy für eine bestimmte Aufgabe unterschiedliche Lernparadigmen angewendet werden.

Zum einen lernt das Behaviour Cloning eine Policy direkt von Zuständen, Aktionen und Kontext um Eingaben zu steuern. Die Qualität dieses Ansatzes hängt direkt von der Gültigkeit der Zuordnung von Zuständen zu Aktionen ab. Auf der anderen Seite kann eine Policy durch Maximierung des Returns einer gegebenen Reward-Funktion gelernt werden. Eine solche Funktion ist normalerweise unbekannt und kann durch die Demonstrationen selbst ermittelt werden, indem Ansätze des Inverse Reinforcement Learning (Ng und Russell) verwendet werden.

Behaviour Cloning (BC)

Unter der Annahme, dass die Experten-Demonstrationen in Form von Zustands- und Aktionspaaren vorliegen, ist das Erlernen der direkten Ausrichtung zwischen dem tatsächlichen Zustand und der gegebenen Steuereingabe als BC zu bezeichnen. Ein solcher Cloning Algorithmus kann in wenigen Schritten nach (Pajarinen) beschrieben werden, der gleichzeitig mit dem Lernprozess verbunden ist.

Zunächst wird ein Satz von Sequenzen, die von einem bestimmten Experten gezeigt wurden, als Ressourcendaten gesammelt. Im nächsten Schritt ist ein Algorithmus für die Darstellung der Policy zusammen mit einer Zielfunktion zu definieren, die üblicherweise als Kostenfunktion definiert wird. Policies können auf verschiedenen Ebenen erlernt werden. Die Planung auf Aufgabenebene bietet Optionen, das heißt Aktionen über einen bestimmten Zeitraum (Sutton und Andrew), die eine Folge von Aktionen auf niedriger Ebene enthalten. Bei der Planung auf Ebene von Sequenzen wird ein Zustand direkt auf eine Sequenz abgebildet. Die Zielfunktion lernt, Fehler zwischen dem gegebenen und dem tatsächlichen Zustand zu minimieren, indem der Verlust der Kostenfunktion minimiert wird. Infolgedessen werden die definierten Parameter der Policy in jedem Schritt ständig weiterentwickelt und optimiert.

Es wird zwischen einem modellbasierten und einem modellfreien Behaviour Cloning unterschieden, wobei jedes das gleiche zu lösendem Problem behandelt, jedoch mit einem anderen Ansatz.

Schritt 4: Verfeinerung der Policy

Die Umgebung, in der Imitation Learning stattfindet, wird oft als Markov Decision Process definiert (MDP). Dies ist ein Tupel:

$$(S, A, T, R, I)$$

wobei S und A jeweils die Mengen der möglichen Zustände bzw. Aktionen sind.

Die Übergangsfunktion T realisiert eine Wahrscheinlichkeitsabbildung zwischen Zuständen bei einer bestimmten Aktion. Die Reward-Funktion R definiert eine Belohnungsfunktion für jeden Übergang von T, wobei der Reward nach dem Übergang direkt empfangen wird.

MDP stellt daher ein mathematisches Modell für sequenzielle Entscheidungsprozesse dar, bei denen die Ergebnisse ungewiss sind. Ein solcher Prozess kann optimiert werden, indem eine Reward-Funktion definiert wird, die in einem Lernalgorithmus, wie zum Beispiel Reinforcement Learning, natürlich angewendet werden kann.

Reinforcement Learning (RL)

RL ist ein Ansatz für maschinelles Lernen, der normalerweise einer Umgebung zugewiesen wird, in der ein beliebiger Agent Maßnahmen ergreift. Im Gegensatz zu überwachtem Lernen, insbesondere neuronalen Netzen, optimiert RL eine Policy zur Vorhersage einer Aktionsverteilung entsprechend der lösungsspezifischen Belohnung (Sutton und Andrew). Wie in Abbildung 2 dargestellt, interagiert der Agent mit der Umgebung, in der ausgehend vom aktuellen Zustand eine Aktion über die Policy ausgewählt und ausgeführt wird. Infolgedessen erhält der Agent dann eine Belohnung oder eine Strafe basierend auf der definierten Reward-Funktion.

Wenn für das bereitgestellte Aktionsmuster eine Belohnung erhalten wird, werden die Parameter der Policy entsprechend der Gradienten aktualisiert, sodass Aktionen mit

höheren Belohnungen in Zukunft bevorzugt werden. Wenn andererseits eine Strafe empfangen wird, ist die Aktualisierung gegen den Gradienten gerichtet, sodass Aktionsmuster, die zu negativen Belohnungen führen, allmählich herausgefiltert werden.

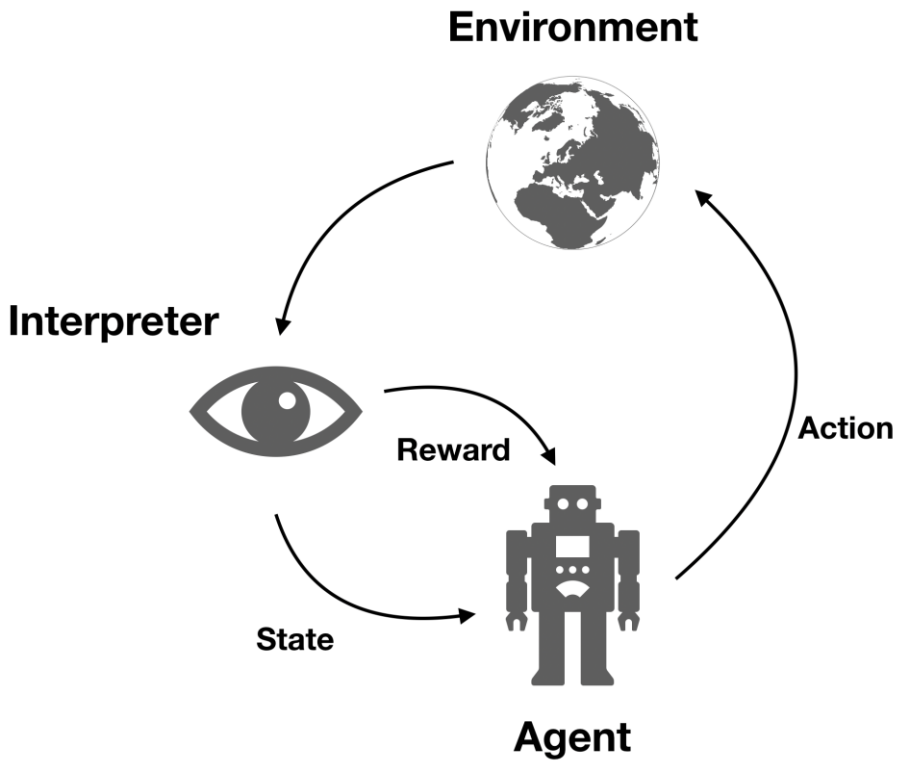

Abbildung 2: Der Lernprozess von Reinforcement Learning

Das Ziel des Agenten ist es, seine Policy so anzupassen, dass er möglichst viel Belohnung erhält. Zunächst erzeugt der Interpreter eine zufällige Ausgabe. Durch Zufall oder Anleitung werden gute Aktionsmuster gefunden und durch Aktualisierungen des Gradienten verstärkt.

Deep Reinforcement Learning (Deep RL)

Deep RL verbindet RL mit tiefen neuronalen Netzen, kombiniert damit die Vorteile beider und ermöglicht die Skalierung von Algorithmen. RL ermöglicht es dem Roboter, durch Trial-und-Error zu erforschen und gleichzeitig nach der optimalen Lösung zu suchen. Wobei Deep Learning uns hilft, mit unstrukturierten Umgebungen umzugehen

und die hohe Dimensionalität der realen Welt zu begreifen. Es werden die zwei Hauptstrukturen Q-Learning und Policy Gradienten (Sutton und Andrew) verwendet. Beim Q-Learning wird eine optimale Policy Q^* gefunden, die die erwartete aufsummierte Belohnung gemäß den Bellman Gleichungen maximiert.

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, a_0 = a\right]$$

Diese Gleichung ist jedoch nicht skalierbar, da jedes Zustand-Aktions-Paar berücksichtigt werden muss. Aus diesem Grund wird entweder der Suchraum quantisiert oder es werden Funktionsapproximatoren verwendet. Handelt es sich bei diesen Approximatoren um tiefe neuronale Netze, so spricht man von Deep Q-Learning (DQN) (Mnih, Kavukcuoglu und Silver). Eine andere Möglichkeit, RL und tiefe neuronale Netze zu koppeln, sind die Policy Gradienten. Dies ist eine Form des Gradientenaufstiegs über den Wert J einer Policy π mit den Parametern θ, wie in folgende Gleichung definiert.

$$J(\theta) = \mathbb{E}[\sum_{t\geq 0} \gamma^t r_t | \pi_\theta]$$

In einfachen Worten, wenn die Belohnung einer Sequenz hoch ist, erhöhen sich die Wahrscheinlichkeiten der zugrunde liegenden Aktionen, andernfalls verringern sie sich.

Systemmodell

Die betrachteten Szenarien stellen aus Sicht der der Mensch-Roboter-Interaktion völlig unterschiedliche Herausforderungen. Aus dem Blickwinkel von Safety, Security und Privacy erlauben sie jedoch, sie in ein abstraktes Szenario zu überführen in dem die weiteren Untersuchungen vorgenommen werden können. Ein davon abgeleitetes abstraktes Systemmodell ist in Abbildung 3 zu sehen.

Ein Werker arbeitet zusammen mit einem Roboter an einem Arbeitsplatz. Es sind keine trennenden Bauteile wie Gitter oder Zäune vorhanden womit vom Roboter eine mögliche Gefährdung für den Werker ausgeht. Ob Werker und Roboter gemeinsam an einem Werkstück arbeiten (Kollaboration), beide wechselseitig Aufgaben am selben Werkstück ausführen (Kooperation) oder völlig eigenständige Arbeiten ausführen, ist für das Szenario nicht relevant. Wichtig ist jedoch, dass beide multisensoriell erfasst werden. Das kann beispielsweise eine Kamera sein, die sicherstellt, dass kein Mensch im Arbeitsbereich des Roboters ist. Möglicherweise sind weitere Sensoren am Werker oder am Roboter und sammeln Daten über die durchgeführten Schritte um sie für IL zu nutzen. Genauso können aber auch einzelne Werkzeuge, beispielsweise eine Drehmomentschrauber, drahtlos Informationen über die ausgeführten Arbeiten versenden.

Absehbar ist zudem, dass zukünftig Roboter nicht nur als statische, sondern auch als bewegliche Akteure auftreten. Mit Autopickern, die benötige Bauteile aus dem Lager direkt bis zum Werker transportieren, sehen wir diese Entwicklung bereits heute. Für die mobilen Anwendungen müssen drahtlose Verfahren zur Kommunikation mit den Robotern verwendet werden. Dies stellt zusätzliche Anforderungen an Safety, Security und Privacy und wird deshalb in das Szenario aufgenommen.

Erfasste Daten werden teilweise direkt in einzelnen Komponenten, beispielsweise direkt im Roboter, aber auch gesammelt im System verarbeitet. Ebenfalls ist eine Übertragung von Teilen der Daten, beispielsweise für ein Predictive Maintenance, an externe Stellen denkbar.

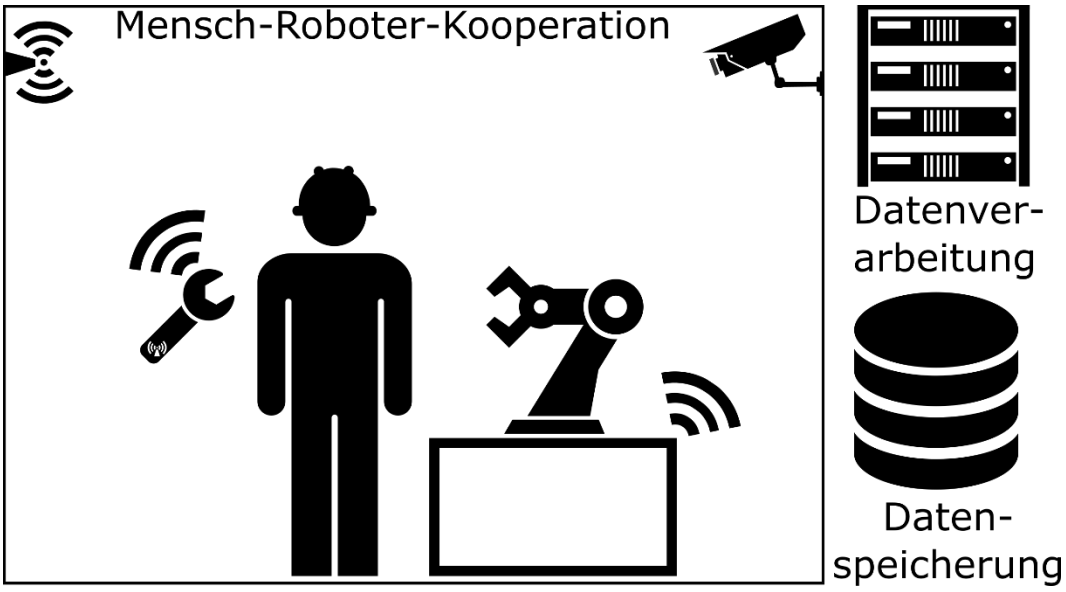

Abbildung 3: Systemmodell der Mensch-Roboter-Kooperation

Angreiferanalyse

Bisher existieren für die Mensch-Roboter-Interaktion hauptsächlich Untersuchungen zur Safety. Zu Beginn wird deshalb eine Angreiferanalyse erstellt die zusätzlich Privacy und Security betrachtet. Die Definition der genutzten Begriffe folgt dem IEC Standard 62443.

Ein **Angreifer** ist demnach eine Person oder eine Personengruppe, die beabsichtigt einen Vermögenswert zu missbrauchen oder zu beschädigen und damit eine Bedrohung verursachen. Eine Schwachstelle eines Systems ist eine Möglichkeit, die Sicherheit gezielt zu beeinträchtigen und damit die Gefahr zu erhöhen.

Der Begriff der **Angreiferziele** folgt nicht IEC 62443, sondern wurde neu definiert. Angreiferziele beschreiben, was einen Angreifer dazu bringt, ein System anzugreifen. Es ist unwichtig, warum ein Angreifer dies tut. Der Angriff kann sowohl ein Mittel sein, um beispielsweise mit gestohlenen Gütern einen Gewinn zu erzielen oder der Angriff kann der Zweck an sich sein, wenn der Angreifer nur das Ziel hat, möglichst viel Chaos zu stiften.

Während Bedrohungen und Schwachstellen sehr stark von einem System anhängen, lassen sich mögliche Angreifer und ihre Ziele bereits auf der Abstraktionsebene unseres allgemeinen Szenarios untersuchen. Abbildung 4 sammelt die **Angreifer** und welche **Angreiferziele** sie verfolgen.

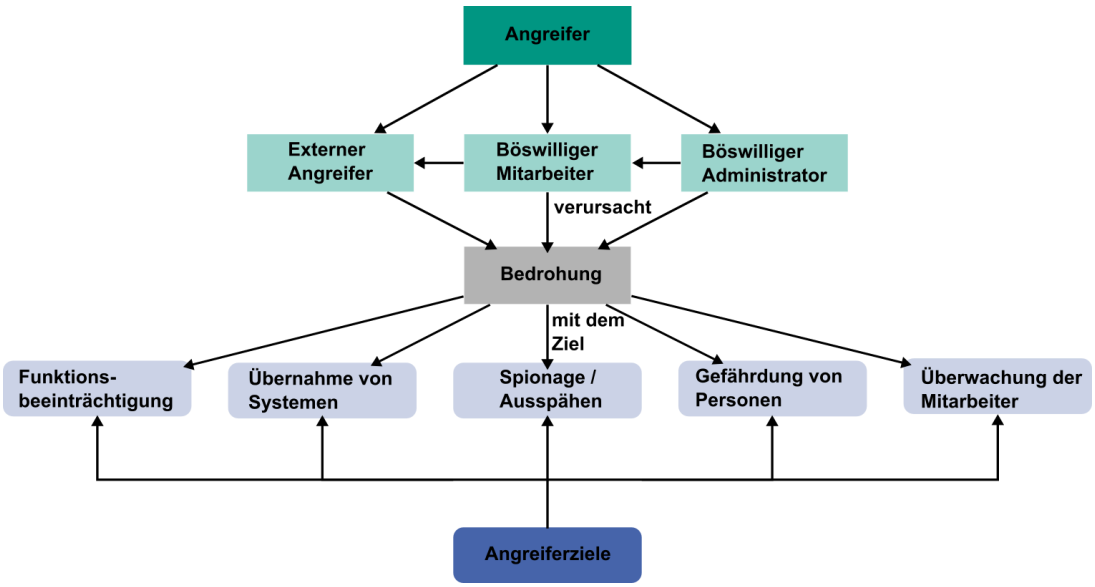

Abbildung 4: Angreifer, Bedrohungen und Angreiferziele in MRK

Angreifer

Das Szenario erlaubt uns, die möglichen Angreifer in drei Gruppen zu gliedern:

- **Externer Angreifer**
  Sind Anlagen mit dem Internet verbunden, können sie Ziel sowohl automatisierte ungezielte als auch gezielter Angriffe durch externe Angreifer werden. Der physikalische Zugang ist externen Angreifern verwehrt.

- **Böswilliger Mitarbeiter**
  Ein böswilliger Mitarbeiter kann die gleichen Angriffe nutzen, wie der externe Angreifer. Zusätzlich jedoch hat er zumindest einen eingeschränkten physikalischen Zugang zu den Systemen. Der böswillige Mitarbeiter ist in seiner Mächtigkeit durch ggf. Rechte- und Rollenkonzepte beschränkt, hat keinen Zugang zu Serverräumen.

- **Böswilliger Administrator**
  Der böswillige Administrator hat die höchste Stufe an Zugangsberechtigungen und stellt damit den mächtigsten Angreifer gegen das System dar. Er hat unbeschränkten physikalischen Zugang zu Anlagen und Serverräumen und ist nicht durch Rechte- oder Rollenkonzepte eingeschränkt. In diesem Modell werden dem böswilligen Administrator auch Angriffe zugeordnet, die dieser im Auftrag eines Vorgesetzten durchführt, beispielsweise, wenn er das System zur Überwachung der Mitarbeiter missbraucht.

Angreiferziele

Wie die Angreifer lassen sich auch die Angreiferziele in verschiedene Gruppen unterteilen. Diese sind nicht zwingend unabhängig voneinander. Es kann gut sein, dass ein Angreifer zuerst die Übernahme von Systemen verfolgt, um sein eigentliches Ziel zu erreichen.

- **Funktionsbeeinträchtigung**
  Der Angreifer möchte die Funktion des Systems in seine Sinne beeinträchtigen. Das kann zu einem Ausfall einer Anlage führen oder auch dazu, dass falsche oder fehlerhafte Produkte erzeugt werden. Als Grund kann reiner Vandalismus oder auch komplexere Ziele wie etwas Erpressung stehen.

- **Übernahme der Systeme**
  Ein Angreifer kann das Ziel haben, Systeme unbemerkt zu übernehmen, beispielsweise um zu einem zukünftigen Zeitpunkt eine Funktionsbeeinträchtigung auszulösen. Übernommene Systeme können aber auch Mittel zum Zweck sein. Das wäre der Fall, wenn übernommene Geräte für einen Distributed Denial-of-Service (DDoS) Angriff missbraucht werden.

- **Spionage / Ausspähen der Produktion**
  Ein Angriff kann das Ziel haben, vertrauliche Informationen aus der Produktion zu stehlen. Das kann sowohl Intellectual Property (IP) über die gefertigten

Produkte sein aber auch die aktuelle Auslastung oder Auftraggeber hinter den Produkten betreffen.

- **Gefährdung von Personen**
  Mittels Manipulation an Anlage kann ein Angreifer das Ziel verfolgen, Personen zu gefährden. Roboter können unkontrolliert bewegt, Sicherheitseinstellungen verändert oder für die Sicherheit der Produktion wichtige Parameter wie Temperaturen oder Drehzahlen verändert werden.

- **Überwachung der Mitarbeiter**
  Die Überwachung der Mitarbeiter stellt ein weiteres mögliches Angreiferziel dar. Dies kann beispielsweise von einem böswilligen Vorgesetzten ausgehen, der das System zur Leistungsüberwachung der Mitarbeiter missbraucht. Genauso können Systeme auch von Kollegen untereinander oder von externen Angreifern missbraucht werden um in die informationelle Selbstbestimmung von Mitarbeitern einzugreifen. Gerade durch die zunehmende Nutzung von Kameras steigt das Gefährdungspotential.

Adversarial Angriffe auf RL

Da die Deep RL Lernalgorithmen der bislang vielversprechendste Forschungsbereich für das Lernen in der Robotik ist, werden wir uns bei unserer Analyse genau auf die Sicherheitsaspekte dieses spezifischen Lernens konzentrieren. Ein Beispiel sind feindliche Angriffe, die den Agenten oder in unserem Fall den Roboter dazu veranlassen könnten, einige unerwünschte oder sogar gefährliche Entscheidungen zu treffen. Die möglichen Angriffe und ihre Vorgehensweise gegen RL sind im Allgemeinen in zwei Hauptgruppen unterteilbar (T. Chen):

- Als **White-Box-Angriffe**, bei denen der Angreifer Informationen über und Zugriff auf das Policy Netzwerk hat.

- Als **Black-Box-Angriffe**, bei denen dem Angreifer keine Informationen zur Verfügung stehen und kein vollständiger Zugriff auf das Policy Netzwerk gewährleistet ist.

In diesem Abschnitt werden wir einen Überblick über einige relevante Angriffe für RL Algorithmen geben. Beim aktuellen Stand der Technik gibt es verschiedene bekannte Angriffsansätze. Wir werden jedoch speziell auf die Angreife eingehen, die für Bahnplanungsprozesse von Robotern entwickelt und erfolgreich eingesetzt wurden. Im letzten Abschnitt werden mögliche Ansätze vorgestellt, um die Möglichkeiten eines Angreifers zu verringern oder einen robusteren und widerstandsfähigeren Lernprozess zu ermöglichen.

White-Box Angriffe

Start point-based adversarial attack (SPA)

Das von (Xiang, Niu und Liu) eingeführte SPA mit dem Schwerpunkt, einen Q-Learning RL Algorithmus anzugreifen, wird in einem Szenario der automatischen Pfadsuche verwendet. Sie erstellten ein Modell, das in Bezug auf bestimmte Eingabepunkte mit einiger Wahrscheinlichkeit entsprechende Ausgaben generieren kann. Bei ihrem Ansatz bestimmen vier Faktoren die Sicherheitsanfälligkeit, die sich auf das endgültige Ergebnis der Pfadplanung auswirken kann. In Bezug auf (Xiang, Niu und Liu) sind dies die Funktionen für die Energiepunktgravitation, Schlüsselpunktgravitation, die Weggravitation und den benutzten Winkel.

White-box based adversarial attack on DQN

Angeregt durch das oben diskutierte SPA haben (Bai, Niu und Liu) einen ähnlichen Ansatz für den Angriff auf den DQN Algorithmus (Mnih, Kavukcuoglu und Silver) zur Pfadfindung angesetzt. Es wurde eine Methode entwickelt, die die Schwachstellen des DQN Ansatzes erkennt. Es wurde festgestellt, dass die Angriffe häufig mit dem Gradienten des maximalen Q-Werts für jeden aufeinanderfolgenden Punkt des Pfades zusammenhängen. Je größer die Differenz der Q-Werte der einzelnen Punkte ist, desto stärker können sie angegriffen werden.

Common dominant adversarial examples generation method (CDG)

Durch die Analyse der Gradientenentwicklung eines A3C (Mnih, Badia und Mirza) RL Algorithmus entwickelte (Chen, Niu und Xiang) eine Methode, mit der feindliche Angriffe auf jeder Karte erzeugt werden können, die für Pfadplanungsaufgaben verwendet wird. Sie fanden heraus, dass das Hinzufügen von Hindernissen im Band mit dem höchsten Gradientenabstieg den A3C-Bahnplanungsprozess beeinflussen kann. Als Ergebnis zeigten (Chen, Niu und Xiang), dass die Genauigkeit der CDG-Methode relativ hoch ist, wenn sie den A3C Lernagenten angreift, der den optimalen Pfad finden soll.

Black-Box Angriffe

Adversarial attack on VIN

Da Black-Box-Angriffe ohne Zugriff auf das Policy Netzwerk definiert sind, bestand die einzige Information, die (Liu, Niu und Zhao) in ihrem Ansatz verwendete, darin, dass der Roboter für die Ermittlung des optimalen Pfades mithilfe des VIN-Algorithmus (Tamar, Wu und Thomas) geschult wurde. Sie haben einige Fälle abgeleitet, in denen der Agent schlecht gearbeitet hat und keine normale Pfadplanung aufwies. Durch das Hinzufügen von Hindernissen um die Wendeabschnitte auf dem Pfad ist es wahrscheinlicher, dass die Pfadplanung gestört wird. Ein anderer Fall ist das Hinzufügen von Hindernissen, die sich in der Nähe des Ziels befinden, was sich weniger auf die Änderung des Pfades auswirkt.

Abwehrstrategien für RL

Die Sicherheitslücken durch feindliche Angriffe auf RL Algorithmen kann auf verschiedenen Ebenen behoben werden. Nach (T. Chen) kann der Widerstand des Lernalgorithmus durch Modifizieren der Eingabe, der Zielfunktion und der Netzwerkstruktur verbessert werden.

Eingabemodifikation

Einer der häufigsten Ansätze zur Erhöhung der Robustheit eines Lernalgorithmus mit neuronalen Netzen besteht darin, das Netzwerk kontinuierlich zu trainieren und mit

neuen Arten von adversarial Beispielen zu versorgen, die auch als adversarial Training bezeichnet werden. (T. Chen) unterscheidet zwischen folgenden Beispielen:

- **Ensemble** adversarial Training von (Tramèr, Kurakin und Papernot),
- **Cascade** adversarial Training von (Na, Ko und Mukhopadhyay),
- **Principled** adversarial Training von (Sinha, Namkoong und Duchi),
- **Gradient Band-based** adversarial Training von (Chen, Niu und Xiang).

Darüber hinaus könnten die Merkmale der für eine bestimmte Aufgabe verwendeten Trainingsdaten, z.B. die Randomisierung der Größe von Trainingsbildern, wie von (Xie, Wang und Zhang, Adversarial examples for semantic segmentation and object detection.) vorgeschlagen, die Stärke des Angriffs verringern, indem sie eine höhere Robustheit schaffen. Zusätzlich zu diesem Ansatz führten (Guo, Rana und Cisse) eine Verteidigungsstrategie ein, indem sie die Trainingsdaten durch Transformation wie z.B. Minimierung der Gesamtvariant, Qualitätskomprimierung und Reduktion der Bit-Tiefe modifizierten. Eine andere Möglichkeit, die Eingabedaten zur Verbesserung der Robustheit gegen Angriffe zu verwenden, ist die Verwendung einer Methode zur Gradientenregulierung von (Ross und Doshi-Velez). Sie fanden heraus, dass durch Bestrafung des Netzwerks, wenn kleine Änderungen an den Eingabedaten signifikante Änderungen in der Ausgabe der Modellvorhersage hervorrufen, die Beständigkeit gegen feindliche Störungen erhöht wird.

Modifikation der Zielfunktion

Mehr Stabilität bei gegnerischen Angriffen kann durch eine Änderung der Struktur der für das Lernen verwendeten Zielfunktion erreicht werden. Ein Weg dies zu erreichen, wie (Zheng, Song und Leung) vorgeschlagen hat, ist einen Stabilitätsterm einzuführen, um den Lernalgorithmus zu ermutigen, eine ähnliche Ausgabe für Bilder verschiedener verrauschter Versionen zu erzeugen. Ähnlich dazu führt (Yan, Guo und Zhang) die Integration eines Regularisierungsterm ein, um die Abwehr zu optimieren, indem die ursprüngliche Zielfunktion mit einer modifizierten und skalierten Version von ihr kombiniert wird.

Modifikation der Netzwerkstruktur

Eine andere Möglichkeit, die Verteidigung gegen in der Literatur bekannte adversarial Angriffe zu erhöhen, besteht darin, die Struktur des Lernnetzwerks zu ändern und anzupassen. Ein von (Metzen, Genewein und Fischer) vorgeschlagenes Verfahren führt einen Ansatz ein, bei dem der Hauptstruktur ein Subnetz als Detektor hinzugefügt wird. Ein Detektor ist ein binärer Klassifikator, der die realen Daten von denen mit Störungen infizierten unterscheiden soll.

Angeregt durch den zuvor beschriebenen Black-Box-Angriff auf den VIN-Algorithmus (Tamar, Wu und Thomas) haben (Xie, Wang und Zhang, Mitigating adversarial effects through randomization.) einen Ansatz zur Erkennung der Angriffe auf Schwachstellen entwickelt. Er konzentriert sich auf den Pfandfindungsprozess, bei dem die Ergebnisse in vier Kategorien unterteilt werden: unerreichte Pfade, unveränderte Pfade, Pfadgabeln und Umleitungspfade.

Design-Leitlinien

Betrachtet man die Individualität der Mensch-Roboter-Kooperation und der Angriffe darauf wird schnell klar, dass es keine allgemeingültigen Lösungen zum Schutz der Systeme gibt. Es können jedoch allgemeine Design-Leitlinien abgeleitet werden, die generell beim Entwurf und Aufbau der Systeme genutzt werden.

Im betrachteten Szenario „Roboter lernt vom Menschen“ macht es Sinn, die Systeme in zwei unterschiedliche Phasen zu trennen. In der **Anlernphase** wird neues Wissen, etwa durch Vormachen eingebracht. In der **Produktionsphase** wird danach das Gelernte umgesetzt.

Das Verhalten des Systems in der Produktionsphase ist dabei deterministisch, sprich es wird nicht in der laufenden Produktion neues Verhalten gelernt. Diese Annahme steht dabei im Widerspruch dazu, dass IL Ansätze ihre Policy ständig verfeinern und damit die Qualität ihrer Lösung erhöhen. Aktuell ist nicht klar, wie solchen Anpassungen, insbesondere aus Sicht der Safety, im Betrieb einer Produktionsanlage

integriert werden können. Zumindest nach aktuellen Sicherheitsstandard muss die komplette Anlage nach einer Änderung neu auf ihre Safety-Eigenschaften zertifiziert werden.

In diesem Kapitel werden zuerst allgemeine Design-Leitlinien formuliert und danach Leitlinien, die nur für einzelne Phasen wichtig sind.

Allgemeine Design-Leitlinien

- **Transparenz der Datenerfassung**
  Es ist für den Mitarbeiter ersichtlich, ob und welche Daten über ihn erfasst werden und wie lange diese ggf. gespeichert werden.

- **Datenschutzkontrolle**
  Produktionssysteme müssen vor dem ersten Betrieb und nach jeder Veränderung auf ihre Datenschutzeigenschaften kontrolliert werden.

- **Einhaltung von Standards**
  Wo immer möglich sollte auf etablierter Standards zurückgegriffen werden. Dies gilt für Security, wo beispielsweise existierende Verschlüsselungsverfahren genutzt und keine eigenen entwickelt werden sollten und insbesondere für Safety. Dort gibt es eine Vielzahl etablierter Standards die die Entwicklung vereinfachen und gleichzeitig Voraussetzung für eine Zertifizierung des fertigen Systems sein können.

- **Graceful Degredation**
  Gerade die zunehmend nachgefragten drahtlosen Verfahren zur Datenübertragung können weder Verfügbarkeit noch Echtzeit garantieren. Dies muss im Design berücksichtigt werden. Würde beispielsweise eine Kamera einen Roboter informieren, wenn sich ein Mensch im Arbeitsbereich befindet, ist es entscheidend wie diese Information eingebettet wird. Um sicher gegenüber von Störungen zu sein, dürfte sich der Roboter nur bewegen, wenn er die Nachricht erhält, dass kein Mensch im Bereich ist. Das Ausbleiben einer

Nachricht darf nicht damit gleichgesetzt werden, dass kein Mensch in diesem Bereich ist.

- **Datenminimierung**
  Das System erfasst nur solche personenbezogene Daten, die für die Erfüllung der Aufgabe zwingend erforderlich sind.

- **Kontrolle der Datenübertragung**
  Werden von Systemen Daten an externe Stellen übertragen muss sichergestellt werden, dass diese Daten nicht in die Persönlichkeitsrechte der Mitarbeiter eingreifen und nicht zur Spionage gegenüber dem Unternehmen missbraucht werden.

Anlernphase

Die Anlernphase ist dadurch charakterisiert, dass hier eine intensive Beobachtung des Menschen und seiner Handlungen notwendig ist. Weiter werden Daten zumindest temporär gespeichert, bis die Anlernphase abgeschlossen ist. Beispielsweise könnte ein Mensch einer Maschine mehrfach vormachen, wie ein bestimmter Schritt zu erfolgen hat. Typischerweise wird, gerade bei komplexen Verfahren, die Maschine nicht sofort beginnen neu gelerntes auf dem Level der späteren Produktionsphase durchzuführen. In der Anlernphase wird so lange simuliert oder reduzierter Kraft und Beschleunigung gearbeitet, bis sichergestellt ist, dass das neue Gelernte korrekt ist. Je nach Anwendungsgebiet und Grad der Kooperation muss das Gelernte am Ende der Anlernphase formal auditiert werden, bevor es in die Produktion übergehen kann.

- **Benutzerautorisierung**
  Gerade in der Anlernphase ist es wahrscheinlich, dass nicht alle Mitarbeiter die gleichen Rechte haben. Beispielsweise könnte nur der Meister die Rechte haben, neue Verfahren einzulernen oder bestehende zu verändern. Um dies umzusetzen benötigen die System eine Möglichkeit ihre Benutzer zu authentifizieren und einzelnen Benutzern Rechte und Rollen zuzuordnen.

- **Auditierung von Änderungen**
  Immer wenn neue Schritte eingelernt oder bestehen verändert wurden, müssen die Änderungen auditiert werden. Hierbei muss sichergestellt werden, dass kein Produktionsschritt vorgenommen wird, durch den die Sicherheit des Produkts oder der Mitarbeiter gefährdet sind.

- **Transparenz der Datenerfassung**
  Über die normale Transparenz hinaus wird für den Mitarbeiter ersichtlich, welche Daten für das Anlernen zusätzlich erhoben werden und ob und wie lange diese ggf. gespeichert sind. Es ist für den Mitarbeiter ersichtlich, wann seine Handlungen für das Anlernen des Systems genutzt werden.

- **Zweckbindung**
  Die für das Anlernen erfassten personenbezogenen Daten werden nicht für andere Zwecke verwendet.

- **Weitergabe**
  Ohne Einwilligung der betroffenen Personen werden keine personenbezogene Daten mit externen Systemen geteilt. Dies bezieht sich nur auf die personenbezogenen Rohdaten. Trainierte neuronale Netze können weitergeben werden, wenn sie keine Rückschlüsse auf die Personen hinter den genutzten Rohdaten zulassen.

Produktionsphase

In der Produktionsphase wird ein angelerntes und einsatzbereites System für die Produktion genutzt. Kennzeichnend für die Produktionsphase ist, dass hier nur bereits auditierte Vorgänge ablaufen. Das System beobachtet weiterhin seine Nutzer, um beispielsweise die Kooperation zu ermöglichen oder Menschen nicht durch seine eigenen Bewegungen zu gefährden. In der Produktionsphase kann das System zwar sein eigenen Verhalten auf den Menschen anpassen, beispielsweise wird ein Roboterarm nicht in einen Bereich schenken, in dem sich ein Mensch befindet, dies

wird aber nicht gelernt. Das gesamte Verhalten des Systems ist bereits gelernt und vorhersehbar und wird in der Produktionsphase nur abgerufen.

- **Livesystem**
  Idealerweise werden erfasste personenbezogene Daten in der Produktionsphase live verarbeitet und nicht gespeichert. Ist dies nicht möglich muss die Speicherdauerbegrenzung definiert und technisch erzwungen werden.

- **Speicherdauerbegrenzung**
  Ist es für die Funktionalität unumgänglich, dass personenbezogene Daten gespeichert werden, muss bereits vor dem Erfassen definiert werden, wie lange diese gespeichert werden. Mit dem Erreichen der definierten Frist müssen Daten automatisch gelöscht werden.

Referenzen

Bai, X, et al. „Adversarial Examples Construction Towards White-Box Q Table Variation in DQN Pathfinding Training.“ *IEEE Third International Conference on Data Science in Cyberspace (DSC)*. 2018.

Billard, A. und M. J. Mataric. „Learning human arm movements by imitation:: Evaluation of a biologically inspired connectionist architecture.“ *Robotics and Autonomous Systems* nov 2001: 145-160.

C. Finn, P. Abbeel, and S. Levine. „Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.“ 2017.

Cao, Zhe, et al. „Realtime multi-person 2d pose estimation using part affinity fields.“ *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

Chen, T, et al. „Gradient band-based adversarial training for generalized attack immunity of a3c path finding.“ 2018b.

Chen, Tong. „Adversarial attack and defense in reinforcement learning-from AI security view.“ *Cybersecurity 2.1* (2019).

Englert, Peter, et al. „Addressing the Correspondence Problem by Model-based Imitation Learning.“ kein Datum.

Guo, C, et al. „ Countering adversarial images using input transformations.“ *International Conference on Learning Representations.* 2018.

Hussen, A et al. „A Imitation Learning: A Survey of Learning Methods A:2 A. Hussein et al.“ *ACM Computing Surveys*. 2017.

Intel. *Intel Realsense* . kein Datum. <https://www.intel.de/content/www/de/de/architecture-and-technology/realsense-overview.html>.

KASTEL Projekt. *Begriffsdefinitionen in KASTEL*. 27. September 2013. <https://www.kastel.kit.edu/downloads/Begriffsdefinitionen_in_KASTEL.pdf>.

Liu, J, et al. „A Method to Effectively Detect Vulnerabilities on Path Planning of VIN.“ *International Conference on Information and Communications Security.* 2017.

Metzen, JH, et al. „On detecting adversarial perturbations. .“ *CoRR*. 2017.

Microsoft. *Microsoft-Kinect*. kein Datum. <https://developer.microsoft.com/de-de/windows/kinect>.

Mnih, V, et al. „Asynchronous methods for deep reinforcement learning. In: International conference on machine learning.“ 2016.

—. „Human-level control through deep reinforcement learning.“ *Nature* (2015).

Na, T, JH Ko und S Mukhopadhyay. „Cascade adversarial machine learning regularized with a unified embedding.“ 2018.

Ng, A. und S. Russell. „Algorithms for inverse reinforcement learning.“ *Proc. of ICML*. 2000.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., & Peters, J. „An algorithmic perspective on imitation learning.“ *Foundations and Trends® in Robotics* 7 (2018): 1-179.

Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, T. Osa, J. Pajarinen, G. Neumann, J. Andrew Bagnell, P. Abbeel, and J. Peters. „An Algorithmic Perspective on Imitation Learning,.“ *Foundations and Trends ® in Robotics* (2018).

Ross, AS und F Doshi-Velez. „Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients.“ 2017.

Schall, Stefan. „Is imitation learning the route to humanoid robots?“ *Trends in cognitive sciences 3.6*. 1999.

Sinha, A, H Namkoong und J Duchi. „Certifiable distributional robustness with principled adversarial training.“ *International Conference on Learning Representations.* 2018.

Sutton, Richard S. und G. Barto Andrew. *Reinforcement learning: An introduction*. MIT Press, 2018.

Tamar, A, et al. „Value iteration networks.“ *Advances in Neural Information Processing Systems*. 2016.

Tramèr, F, et al. „Ensemble adversarial training: Attacks and defenses.“ *CoRR*. 2017.

Tung, C. und A. Kak. „Automatic learning of assembly tasks using a DataGlove system.“ *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative* . 1995.

Xiang, Y, et al. „A PCA-Based Model to Predict Adversarial Examples on Q-Learning of Path Finding.“ *IEEE Third International Conference on Data Science in Cyberspace (DSC)*. 2018.

Xie, C, et al. „Adversarial examples for semantic segmentation and object detection.“ *CoRR*. 2017.

—. „Mitigating adversarial effects through randomization.“ *International Conference on Learning Representations.* 2018.

Yan, Z, Y Guo und C Zhang. „Deepdefense: Training deep neural networks with improved robustness.“ *CoRR*. 2018.

Zheng, S, et al. „Improving the robustness of deep neural networks via stability training.“ *Proceedings of the ieee conference on computer vision and pattern recognition.* 2016.