Vertrauen ist schlecht, Kontrolle ist besser

Johannes Fähndrich und Dirk Labudde

Bias, Diskriminierung und Intransparenz wird komplexen Modellen der Künstlichen Intelligenz (KI) nachgesagt. Komplexe Modelle sind für Menschen schwer zu verstehen. Durch das fehlende Verständnis lässt sich die Nachvollziehbarkeit von „Black Box“ Modellen nur selten erhöhen. Dieser Sachverhalt führt dazu, dass Richter meist nicht in der Lage sind, die Ergebnisse komplexer Modelle in Gerichtsverfahren zu verwenden. Wegen der mangelnden Verwendbarkeit von komplexen Modellen werden diese noch nicht ausreichend in die Forensik integriert. Nachvollziehbarkeit kann auf drei Komponenten heruntergebrochen werden: **Reproduzierbarkeit** erlaubt es, die Vorgänge, Entscheidungen oder Ergebnisse wiederholt zu generieren. Dadurch kann der Einfluss von Parametern untersucht werden. **Rückverfolgbarkeit** erlaubt es, die Prozesse der Erstellung, des Trainings, der verwendeten Daten, deren Einfluss und der Verwendung eines komplexen Modells zu dokumentieren. **Verständnis** erlaubt, die Ausgaben eines komplexen Modells zu begreifen und intellektuell zu erfassen. Zur intellektuellen Erfassung von komplexen Zusammenhängen oder Sachverhalten ist meistens eine Vereinfachung dessen für den Menschen hilfreich. Zur Vereinfachung untersuchen Ansätze der verifizierten KI Modelle der KI auf Nachvollziehbarkeit.

Aufbauend auf der Arbeit von Sanjit et al. 2022, werden Anwendungen forensischer KI-Modelle diskutiert, die sich mit der „Gleichheit der Waffen“, „Be- und entlastendes Ermitteln“ beschäftigen. Gesondert wird auf die Problematik Bias eingegangen. Es soll ein Beitrag geleistet werden, der das Verständnis für diese Modelle erhöht und die Attraktivität von KI-Modellen in der Forensik verbessert. Damit soll das Vertrauen in KI gestärkt werden.

Wir werden in dieser Publikation als Beispiel ein System nutzen, welches Elemente des Maschinellen Lernens bereithält. Anforderung war, dass dieses System noch nicht in Anwendung, aber in der Diskussion ist. Gezielt wurde hier ein kontroverses Beispiel genommen, damit die nötige Diskussion und deren Dimensionen klarwerden. Das Beispiel ist die Anwendung von Gesichtserkennung. Dabei lassen sich für die Wahl des Beispiels viele Parameter bestimmen. Beispielsweise, ob ein Algorithmus oder die implementierte Anwendung untersucht werden soll. Je mehr Anwendertiefe vorhanden ist, desto schwieriger ist die Verifikation des Systems, da immer mehr Fehlerquellen hinzukommen können. Beispielsweise ist ein Algorithmus in Pseudocode

einfacher zu überprüfen als seine Implementierung, da Implementierungsdetails, das verwendete Paradigma oder Einzelheiten der Programmiersprache und des Compilers/Interpreters, die zu Erstellung einer Software Library nicht in Betracht gezogen werden müssen. Jedoch ist eine Software-Library einfacher zu verifizieren als eine Software, die diese verwendet, da hier nicht der Einsatz einbezogen werden muss. Der Einsatz einer Library umfasst die Hardware, auf der diese ausgeführt wird, sowie deren Architektur (Verteilt, in Memory, Persistenz, ...). Eine Software ist einfacher zu verifizieren als ein System im Deployment (zur Laufzeit), denn auch hier werden Details abstrahiert, die eine Verifikation erschweren. Beispielsweise, auf welchen Daten ein System trainiert wurde oder welche Fehler ein Nutzer bei den Eingaben macht.

In unserem Fall gehen wir von der schwierigsten dieser Varianten aus und nehmen an, dass wir für unsere Software nur die Deployte Umgehungen zur Verfügung haben und keinen Zugriff auf die Quellen und Daten, die das System verwendet.

Verifizierte Systeme

Das Feld der Verifikation von formalen Systemen wird schon einige Zeit beforscht [2]. Dabei werden die Eigenschaften eines Systems formal dargestellt und durch einen Beweis die Eigenschaften des Systems repräsentiert. Durch die Konstruktion solcher Beweise lernen wir etwas über die Systeme und deren Eigenschaften. Dies ist der Fall bei sicherheitsrelevanten Systemen wie digitale Flugkontrollsysteme [3]. Durch die zunehmende Verwendung von Methoden des Maschinellen Lernens (als Teil der Künstlichen Intelligenz) in sicherheitsrelevanten Systemen, wird die Anforderung für formale Verifikation nun auch an Lernende Systeme gestellt [2]. Die Autoren von [1] unterscheiden die Verhaltensüberprüfung eines Systems in drei Teile:

Als Erstes gibt es einen Verifikationsansatz. Dabei werden in Abb. 1 dargestellt drei Eingaben in ein Verifikationssystem gegeben:

1. Das System, das verifiziert werden soll. In unserem Fall eine Software für Gesichtserkennung.
2. Die Spezifikation, die das System erfüllen soll, beispielsweise Fehlerraten für die Gesichtserkennung und Zeitbedingung, in denen das System antworten sollte.
3. Die Umgebung, in der das System läuft. In unserem Beispiel die Anwendung und Beispieldaten sowie die Ausführungsumgebung.

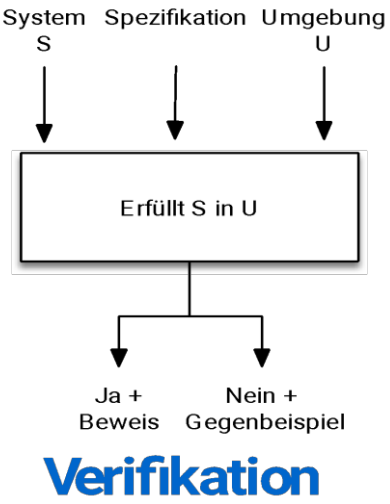

Abb. 1: Schematische Darstellung eines Verifikationsansatzes. (aus [1])

Die Art der Verifikation:

Die in Abb. 1 dargestellte Verifikation führt das System aus und untersucht unter der Ausführungsumgebung, mit den in der Spezifikation gegebenen Parametern zu beweisen, dass die Spezifikation erreicht wird, oder zeigt, mit welchen Parametern das System nicht den Spezifikationen entspricht (Gegenbeispiel).

So werden nicht nur der Algorithmus oder die Software überprüft, sondern auch die Ausführungsumgebung und ob diese fehleranfällig und leistungsstark genug ist.

Nachteile der Methodik:

Nachteil ist die Notwendigkeit einer detaillierten Spezifikation. Ohne die entweder abstrakte oder umfangreiche Spezifikation kann nicht vollumfänglich verifiziert werden, was einen Beweis fast unmöglich macht. Eine abstrakte Beschreibung der Spezifikation erlaubt eine kompakte Darstellung, kann aber zu hohen Aufwänden bei der Verifikation führen. Beispielsweise, wenn Parameterwerte nicht eingeschränkt werden können. Eine umfangreiche Spezifikation benötigt Kenntnis über das System und die gesamten Umgebungen in denen es verwendet wird. Dies zu erstellen, ist meist aufwändig.

Vorteile der Methodik:

Vorteil ist, dass eine Verifikation meist automatisch durchgeführt werden kann. Methoden wie Fuzzing [4] und Laufzeitverifikation [5] lassen auch die Ausführlichkeit der Spezifikation variieren. Was die Nachteile

des Systems aufwiegt. Vorteil ist, dass eine Verifikation das System zur Laufzeit überprüfen kann und damit Veränderungen in der Umgebung zur Laufzeit ermöglicht.

Beispiel:

In unserem Beispiel könnte also die Verifikation mit einer Menge Testdaten in unterschiedlicher Qualität sein. Mit einer Ground-truth sowie unterschiedliche Fälle, wie beispielsweise unterschiedliche Ethnien, verschiedene Altersgruppen, Zwillinge, und verschiedene Umgebungsvariationen wie mit Bart, mit Mütze, Beleuchtungsvarianten und Bildformate, mit Teil verdeckten Gesichtern und unterschiedlichen Kompressionsverfahren und Parametern.

Zu beachten ist hier, dass die ethischen Fragestellungen, die das System erfüllen soll, in der Spezifikation und der Umgebung mit aufgenommen werden müssen. Sind beispielsweise die Testdaten gebiast, kann eine Verifikation nicht stattfinden.

Zweitens kann die Funktionsweise eines Systems durch Synthese untersucht werden. Wie Abb. 2 darstellt. Braucht es dazu das System selbst nicht als Eingabe, sondern das System wird synthetisiert. Dabei wird Code anhand der Spezifikation generiert [6]. Hierfür gibt es viele Arten, in denen eine Spezifikation vorliegen kann. Softwarespezifikation gerade zur Laufzeit ist ein eigenes Forschungsfeld [10]. Die Anwendung großer Sprachmodelle vereinfacht die Spezifikation nochmal [9].

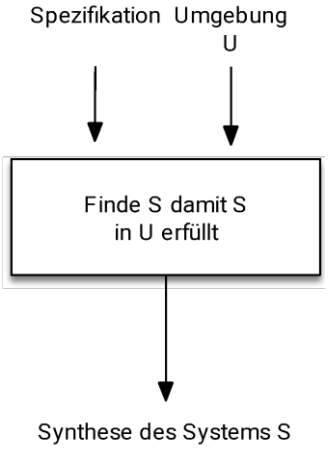

Abb. 2: Abstrakte Darstellung einer Synthese. (aus [1])

Art der Synthese: Generierung von Code ist je nach Spezifikation mehr oder weniger abstrakt. Wenn die Spezifikation detailliert ist, kann besser zugeschnittener Code generiert werden. Testgetriebenes Entwickeln [7] kann als ein solches Beispiel gesehen werden. Auch moderne Softwarearchitekturen wie Loosly-Coupled System unter der Verwendung von Service-Level-Agreements erlauben eine einfachere Synthese. Die Generierung von Code kann je nach Abstraktion auch in anderen Feldern beforscht werden [8]. Idee ist dabei, dass das generierte System die Spezifikation in der gegebenen Umgebung erfüllt.

Nachteil: Eine ausreichend genaue Spezifikation muss vorhanden sein. So müssen Klassendiagramme oder Entität-Relation-Diagramme Teil der Spezifikation sein, um Datentypen und Datenbanken zu synthetisieren. Event- und Schnittstellenbeschreibungen müssen vorhanden sein. Sowie für die Planung ein Start- und End-Zustand.

Vorteil:

Vorteil ist, mit einer ausreichend genauen Spezifikation kann ein System fast vollautomatisch erstellt werden. Auch die Adaption und Wartbarkeit eines synthetisierten Systems sind vorteilhaft, da durch Einpassung der Umgebung oder der Spezifikation eine neue Version des Systems synthetisiert werden kann.

Beispiel:

In unserem Beispiel könnte die Gesichtserkennungssoftware durch eine Verarbeitungspipeline synthetisiert werden, wobei alle Komponenten die Anforderungen der Spezifikation z. B. in Blick auf den Datenschutz erfüllen. Dadurch kann die Software modular aufgebaut und Aufgaben können gekapselt werden. So sind Fragen, wie:

- Wo liegen die Daten?
- Wie wurden die Daten aufbereitet?
- Auf welchen Daten werden die Komponenten mit Maschinellem Lernen trainiert?
- Welche Komponente bietet welchen Dienst an?
- Wie können Komponenten verbessert bzw. ersetzt werden?
- Wo liegt die Verantwortung der Komponenten?

von Bedeutung.

In diesem Beispiel wird die Reihenfolge, die Datenformate, die verwendeten Komponenten, die SLAs und so weiter so synthetisiert, dass die Kette von Komponenten am Ende die Anforderungen erfüllt. Synthetisierte Software hat den Vorteil, dass viel von dem, was auto-

matisiert erstellt wird, schnell anpasst werden kann. Eine Adaption an neue Komponenten oder Dienstanbieter wird damit vereinfacht. r. Diese Adaption erlaubt eine höhere Ausfallsicherheit und Resilienz.

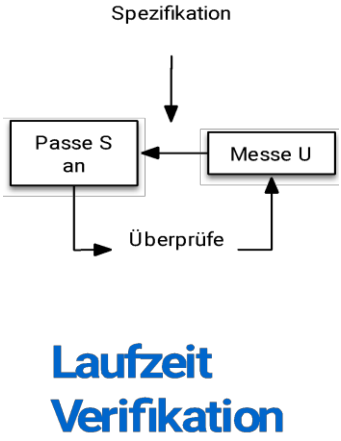

Abb. 3: Abstrakte Darstellung einer Laufzeitverifikation. (aus [1])

Art der Laufzeitverifikation:

Die dritte Art der Erzeugung überprüfbarer Systeme verwendet einen höheren Automatisierungsgrad. Die Automatisierung ist nicht nur die Zusammenstellung und Synthese von Software, sondern interagiert direkt mit der Anwendungsumgebung zur Laufzeit. Idee ist dabei, dass ein System so adaptiert wird, dass es sich zur Laufzeit an die Anwendung anpasst. Ein Beispiel für solche Systeme sind Planer. Hier wird autonom über die zu verwendenden Schritte entschieden, die benötigt werden, um ein gegebenes Ziel (hier Spezifikation genannt) erreicht wird. Dabei werden zur Laufzeit die vorhandenen Komponenten gesucht und eingebunden und so ein System zur Laufzeit synthetisiert.

Was bei einem synthetisierten System gewöhnlich zur Designzeit passiert, wird nun in die Laufzeit übertragen. Die Adaption zur Laufzeit erlaubt mehr Flexibilität, bringt aber auch mehr Komplexität mit sich [12,13]. In Abhängigkeit, welche Flexibilität zur Laufzeit gewünscht ist, werden verschiedene Arten der Adaption des Systems notwendig. Je nach Adaptionsart sind verschiedene Arten der Laufzeitverifikation notwendig.

Nachteil:

Die Flexibilität, die hier entsteht, ermöglicht die Anwendung eines Systems in vielen Kontexten, macht jedoch die Spezifikation der Lauf-

zeitumgebung notwendig. Hier entstehen Fragestellungen wie:

- Wann wird ein System adaptiert?
- Was wird als Erfolg/Fehlschlag gemessen?
- Wie wird adaptiert?
- Welche Parameter sind adaptierbar?
- Wie oft wird eine Zielerreichung überprüft?
- Welchen Grad der Autonomie überlassen wir dem System?
- Welche Metriken für dessen Bewertung zur Laufzeit finden Anwendung?
- Wo sind Adaptionspunkte im System?

beantwortet.

Diese Fragestellungen machen ein System, welches zur Laufzeit verifiziert wird, komplexer. Die Komplexität wird bei den beiden vorherigen Arten der Verifikation zur Designzeit entschieden und umgesetzt. Dies macht die resultierenden Systeme unflexibler, aber weniger komplex.

Vorteil:

Entscheidungen, die zur Laufzeit getroffen werden, erlauben es, auf Veränderungen im Ausführungskontext einzugehen. Durch die Veränderungen zur Laufzeit kann zwar das Verhalten verifiziert werden, da die Adaption zur Laufzeit an Einschränkungen (engl. Constrains) gebunden ist, jedoch müssen die Einschränkungen spezifiziert werden. Bei der Spezifikation der Einschränkungen sollte auf den Grad der Autonomie des Systems eingegangen werden. Die Grade der Autonomie sind nicht nur von Technologie und Kosten abhängig, sondern auch vom Anwendungskontext und der rechtlichen Grundlage. Autonomie von Systemen ist ein Forschungsbericht der sogenannten „starken KI“ (engl. Strong AI) [17], und kann auf verschiedene Weise interpretiert werden [16].

Beispiel:

Für unser Beispiel der Gesichtserkennung in der Polizeiarbeit kann ein System, welches eine Laufzeitverifikation durchführt, die verwendeten Dienste oder, wenn möglich, deren SLAs überprüft, um so ein System zu adaptierten, damit die in der Spezifikation aufgeführten Bedingungen erfüllt werden. Wird ein System beispielsweise in einem Anwendungskontext mit einer gewissen Rechtslage eingesetzt, kann darauf geachtet werden, dass diese rechtlichen Grundlagen erfüllt werden. Außerdem können die Qualität der verwendeten Dienste (manchmal auch Agenten und deren Aktionen genannt) zur Laufzeit überprüft und angepasst werden. Bei einer Gesichtserkennung kön-

nen so verschiedene Dienste kombiniert werden, die auf verschiedenen Datengrundlagen aufbauen, sowie Spezialisierungen, wie Gesichtserkennung mit Maske, Brille oder Bart. Damit haben wir die Arten der Kontrolle und den Stand der Technik erhoben. Hier wäre eine strukturierte Literaturübersicht wünschenswert, da die Verwendung von Systemen mit Ansätzen des Maschinellen Lernens sich schnell ausbreitet und eine rechtssichere und für die Gesellschaft nützliche Anwendung schnellstmöglich auch in der Polizeiarbeit Einzug halten sollte.

Leider ist hier die Polizeiarbeit noch nicht so weit und eine Diskussion über Ethik und Recht für Anwendungen mit Ansätzen des Maschinellen Lernens ist noch nicht Teil des Diskurses in der Praxis. Hier könnte eine frühzeitige Diskussion bei der Umsetzung bzw. Erstellung von modernen Systemen helfen. Eine wissenschaftliche Diskussion ist schon Teil der der akademischen Community [18], Wir haben jedoch noch keine Übertragung in die Praxis feststellen können. Für die Polizeiarbeit bleibt es zum gegenwärtigen Zeitpunkt fraglich, welche Arten der Verifikation einen Königsweg darstellen könnte. Durch den Schritt der freien Beweiswürdigung vor Gericht, müssen Ergebnisse, die auf diese Art entstanden sind, diskutiert werden.

Eine Frage des Vertrauens

Vor Gericht ist eine Spur oder ein gefundenes Indiz immer unter Bewertung der verwendeten Methode für dessen Gewinnung zu betrachten. Dies gilt für Methoden und Methodiken der klassischen Forensik, wie auch für Methoden aus dem Bereich der Digitalen Forensik. Der Weg einer vor Gericht anerkannten Methode geht über Standardisierung. Dabei muss jeder Einzelschritt der Methode wissenschaftlichen Standards genügen. Neben der wissenschaftlichen Betrachtung muss die Ebene des Eingriffs in Grundrechte des Beschuldigten beachtet werden. Diese zwei Perspektiven vermitteln Vertrauen und werden durch Richtlinien abgesichert. Somit ergeben sich gerade für biometrische Verfahren, welche für die Personenfeststellung genutzt werden, eine nachvollziehbare wissenschaftliche Methode und Methodik, welche von rechtlichen Bedingungen flankiert werden. Ein gesonderter Punkt ist die Beschreibung der Qualität der sogenannten Tatortprobe und der Vergleichsprobe. Bei starken Abweichungen kann es zu Individual Gutachten, anstatt der standardisierten Methode, kommen.

Arten der Kontrolle

Wir haben die verschiedenen Arten der Kontrolle über Software besprochen und diese in den Kontext der Polizeiarbeit, für die Ge-

sichtserkennung, als biometrische Methode, gebracht. Je nach Software und ihrer Parameter verursacht eine Verifikation mehr oder weniger Kosten [19]. Um Kosten der Verifikation zu minimieren, wollen wir hier die notwendige Kontrolle diskutieren. Um ein Indiz vor Gericht zu verwenden, muss das Vertrauen in dessen Aussagekraft ausreichend sein. Was genau ausreichend ist, wird in Bezug zur notwendigen Kontrolle gesetzt. Um bei unserem Beispiel zu bleiben, könnte eine Software zur Gesichtserkennung zwar exakte Ergebnisse liefern, aber die verwendeten Daten im Verfahren sind Teil einer gewohnheitsrechtlichen Regel im Strafprozess, wenn die Daten nicht nach rechtsstaatlichen Kriterien erhoben wurden.

Die Frage ist, welche Art von Kontrolle braucht es. Hier wird meist argumentiert, dass die Daten, die für das Training der Verfahren verwendet wurden, auch deren Ausgabe bestimmen. Durch eine Manipulation der Trainingsdaten soll so die Ausgabe des Systems manipuliert werden [20]. Dies ist ähnlich zum menschlichen Handeln. Wird ein Experte vor Gericht geladen, sollte dessen Referenzen und seine Qualifikation überprüft werden. Es wird dabei also darauf geachtet, welche Daten (Lehrmaterialien, Zertifikate, Abschlüsse, Berufserfahrungen,) dazu frühen, dass die Aussage (z. B. dem vorläufigen Gutachten) des Experten vertraut werden kann. Die Kontrolle, die dieses Vertrauen erzeugt, wird dabei meist vor der Beauftragung des Experten durchgeführt, und durch die Verteidigung in der Hauptverhandlung überprüft.

Hier eine formale Verifikation durchzuführen, wäre ein aufwändiges Verfahren. Das Ergebnis eines Experten kann beispielsweise durch weitere Experten bestätigt oder widerlegt werden. Für ein System mit der Verwendung von Maschinellem Lernen (ML-System) könnte diese ähnlich sein.

Eine der wissenschaftlichen Diskussionen um den Umgang mit ML-Systeme ist eine Zertifizierung [21]. Diese kann (wie auch beim Menschen) in zeitlichen Abständen wiederholt werden. Diese Wiederholung ist sinnvoll, da lernende Systeme sich weiterentwickeln können. Systeme, die sich entwickeln, können auf der anderen Seite sogar einen Effekt erleiden, der „Catastrophic Forgetting“ genannt wird [22]. Dabei werden alte gelernte Informationen durch neue überschrieben. Was zu anderen Ausgaben führen kann. Somit stellt sich die Frage: Wenn ein Ergebnis eines ML-Systems für Gericht Anwendung findet und Jahre später sich weiterentwickelt hat und nun zu einem anderen Ergebnis kommen würde, wie dieses System dann erneut Anwendung findet.

In unserem Beispiel könnte unser Gesichtserkennungssystem für gewisse Altersgruppen, Bildqualität oder Ethnie eine gewisse Güte von Ergebnis liefern. Beispielsweise könnte ein Match einem Kamerabild zugeordnet werden. Später, nachdem das ML-System mehr Daten zum Lernen hatte, oder algorithmisch angepasst wurde, ist eine genauere Erkennung möglich, die einen Verurteilten entlastet. Hier stellen sich nun weitere Fragen wie:

- Wie oft müssen die Ergebnisse eines ML-Systems überprüft werden?
- Wie wird bei solchem mit anderen Verfahren umgegangen?
- Wann dürften sich Systeme weiterentwickeln?
- Wie kann hier ein Standard definiert werden?

Wenn von einem Menschen (Sachverständigen) ein Gutachten, welches sich auf dem gegenwärtigen Wissensstand befindet, in einem Verfahren eingebracht wird, kann auch später mit mehr Wissen des Experten, bzw. durch den Fortschritt auf dem speziellen Gebiet, ein Gutachten anders ausfallen. Hier stellen sich ähnliche Fragen: Wird das Gutachten dann neu erstellt? Wo sind hier die rechtlichen und ethischen Hürden? Ein Experte kann hier als die dritte Art der Verifikation zur Laufzeit interpretiert werden. Da ein deterministisches Verhalten für die anderen Arten der Verifikation kaum möglich ist.

Die Frage, die wir hier also stellen ist: Welches Vertrauen setzen wir hier in Menschen und ab wann kann einem ML-System ein ähnliches Vertrauen entgegengebracht werden? Vielleicht ist hier ein alternativer Ansatz, eine ML-Verfahren nicht als Sachbeweis zu sehen. Denn ein ML-System wurde auf Daten trainiert (ausgebildet) und nutzt das so erlernte Wissen, um nun Informationen zu interpretieren. Also wie ein Mensch, der auf einer Videoaufnahme ein Gesicht wieder erkennt, und damit beispielsweise einen Tatverdächtigen in diesem Video zuordnet. So findet auch nach dem Training ein Wiedererkennen bei unserem Beispiel mit Gesichtserkennung statt. Die Fehlerrate und der Konfidenzbereich für eine so getroffene Aussage hängt dann vom ML-System (dem Experten), den Trainingsdaten (seiner Ausbildung) und den gegebenen Daten ab.

Eine objektive Bewertung kann nun verschiedene Teile des Systems bewerten. Dabei kann argumentiert werden, dass nur das Klassifikationsergebnis zählt. Es kann aber auch argumentiert werden, dass die Art, wie das Klassifikationsergebnis zustande kam, gewertet werden soll. Wir versuchen hier eine Einordnung der verschiedenen Ebenen der Kontrolle:

Kontrolle der Trainingsdaten: Hier wird aus dem Bereich des Data-Mining und der Datenanalyse das Thema Datenqualität beforscht. Datenqualität kann gemessen und verbessert werden. Daten können Biased sein, sie können normalisiert werden und sie können verrauscht sein. Je nachdem welche Anforderungen an das System gestellt werden, können Trainingsdaten generiert werden, oder es können statistische Methoden wie N-Folde Corssvalidation [24] verwendet werden, um deren Nutzung zu verbessern [23]. Hier gibt es mehrere Probleme, Overfitting [25] and the Curse of dementionality [26] sind nur zwei Beispiele. Hier ist jedoch fraglich, ob nicht oder überhaupt aus Daten mit niedriger Qualität etwas gelernt werden kann. Mit niedriger Qualität ist nicht die Datenqualität gemeint, sondern verrauschte Daten, die Intra-Klassen Distanz vergrößern und die Inter-Klassen Distanz verkleinern. Also für unser Argument: Auch Menschen können aus Biased Daten das Richtige lernen, wenn die falschen oder Biased Teile der Daten ignoriert werden. Diese Fähigkeit sprechen wir Automatisierungen von Statistik (unserer „schwachen KI“) ab.

Kontrolle der Algorithmen: Die Art, wie ein Algorithmus oder genauer eine Methode des Maschinellen Lernen funktioniert, kann zertifiziert werden. Ähnlich wie die Methode eines DNA-Verfahren zertifiziert werden kann. Allerdings kommt es auf die Inputs des Algorithmus an, was die Ausgabe liefert. So ist das Verfahren für DNA-Analysen so zertifiziert, dass sie vor Gericht Verwendung finden können. Dabei ist jedoch das Ergebnis des Verfahrens davon abhängig, was für DNA-Material in das Verfahren gegeben wird. Ist der Input zu Biased dann wird die Aussage des Verfahrens auch die nötige Qualität fehlen [27]. Wir argumentieren, dass selbst die Zertifizierung eines Verfahrens nicht die Eigenschaften eines ML-Systems kontrollieren kann, die am Ende die Ausgaben gerichtsfest machen, da die Trainingsdaten und die Daten als Eingabe die Ausgabe beeinflussen.

Kontrolle der Systemevaluation (Performance): Die objektivste Kontrolle eines Systems ist der Output. Dabei wird nicht diskriminiert, wie der Algorithmus funktioniert, noch auf welchen Daten trainiert wurde, sondern nur die Aussage des Systems ist entscheidend. Ähnlich wie bei einem Experten werden irrelevante Merkmale ignoriert. Merkmale, die ignoriert worden sind, beispielsweise die Herkunft des Algorithmus, seine Methode, die Erfinder, in welcher Sprache er implementiert wurdes, usw. Dies wird alles auf das Ergebnis übertragen. Wenn die Programmiersprache, einen konzeptionellen Fehler hat, was zu einem Fehler in der Aussage des ML-System führt, wird dies mit als Fehler des ML-Systems gewertet. Als Fehler des Systems wird die Verwendung von Biased Trainingsdaten ebenfalls bewertet. Durch das Inter-

pretieren des Ergebnisses des ML-Systems kann durch diese Evaluation der Ergebnisse nur schwer auf die Gründe des Fehlers zurückgeschlossen werden. Die ausgeführte Kontrolle über einem ML-System bestimmt, wo wir Parameter ändern können, um die Aussage zu verändern. Dies funktioniert meist gut bei Systemen, die wir verstehen. Warum ein System funktioniert, kann bei komplexen Systemen schwer zu verstehen sein und damit auch der Zusammenhang zwischen Änderungen von Parametern und den Änderungen in der Aussage.

Die Diskussion über die Kontrolle von Systemen wird damit verbunden, dass wir Menschen nur kontrollieren können, was wir auch verstehen. Verständnis für komplexere ML-Systeme ist nicht in der Tiefe vorhanden. Das Forschungsfeld, das Verständnis für ML-Systeme beforscht, untersucht, wie Systeme sich selbst erklären können [28]. Dies wird „Explainable AI“ (erklärbare Künstliche Intelligenz) genannt. Dieses beschränkt sich jedoch darauf, die Aussage mit der Eingabe eines Systems zu erklären und nicht die Trainingsdaten mit zu untersuchen [29]. Hier könnte hinterfragt werden, ob für eine Verwertung vor Gericht eine Erklärung notwendig ist, da ein Verständnis bei den Beteiligten wahrscheinlich ausbleibt. So wie auch ein DNA-Test (der genetische Fingerabdruck) nicht verstanden wird, sondern auf dessen Korrektheit vertraut wird.

Vertrauen in neue Methoden zu schaffen, benötigt die dritte Art der Kontrolle: also die Korrektheit der Aussage. In unserem Beispiel mit der Gesichtserkennung: Wenn die erkannten Personen über eine andere Methode den Bildern zugeordnet werden können, oder genügend Testdaten vorhanden sind, um beispielsweise einen erwarteten Fehleranteil von 6-Sigma zu erreichen. Hier ist also die ausschlaggebende Frage: Was kann getan werden, um neuen Verfahren Vertrauen zukommen zu lassen?

Anwendung in der Forensik

Die Forensik ist eine Wissenschaft, die sich Methoden und Techniken anderer Fachgebiete zu Nutze macht. So bedient sie sich der Molekulargenetik, um Unterschiede auf der DNA (Desoxyribonukleinsäure) zur Identifizierung von Tätern und Opfern zu nutzen. Sogenannte genetische Fingerabdrücke werden mittels Elektrophorese erstellt, um Tatortspuren mit Datenbankeinträgen bzw. mit den genetischen Fingerabdrücken eines Tatverdächtigen zu vergleichen. Bei der Elektrophorese wird zur Trennung von DNA-Fragmenten die Wirkung eines elektrischen Feldes auf geladene Moleküle ausgenutzt. In der Forensik geht es prinzipiell immer um einen Vergleich von Mustern, sei

es den Abdruck durch die Papillarleisten der Fingerbeere, oder um die DNA. Der menschliche Bauplan entscheidet, wie wir aussehen und wie unser Stoffwechsel funktioniert. Sie ist zu einem Großteil in allen Menschen identisch. Dennoch unterscheiden sich gewisse Sequenzen voneinander. Gerade in Bereichen, die keine genetische Information tragen, in den sogenannten „nicht-kodierenden DNA-Sequenzen“, gibt es Unterschiede. Dort findet man häufig Wiederholungssequenzen, die „Short Tandem Repeats“ (STR). Diese kurzen, sich wiederholenden Basenpaarsequenzen werden in der Abstammungsanalyse und in der Forensik als Marker verwendet, wobei die Anzahl der Wiederholungen gemessen wird. Die Anzahl der Wiederholungen der STR-Wiederholungen ist bei jedem Menschen einzigartig.

Mit Hilfe der STR-Analyse, Polymerase-Kettenreaktion und Ektrophorese können innerhalb kürzester Zeit und mit wenigen manuellen Schritten Vorauswahlen zur Identifizierung von Personen getroffen werden.

Für den genetischen Fingerabdruck betrachtet man bestimmte STR-Bereiche der DNA und untersucht, wie oft sich die Sequenzen dort wiederholen. Aktuell werden bei der STR-Analyse bis zu 19 Bereiche untersucht, um eine hinreichende statistische Aussagekraft zur Identifizierung zu erlangen. Das Muster, die Anzahl der Wiederholungen jedes Abschnittes, hat eine Wahrscheinlichkeit von 10-19, dass zwei Personen einen identischen genetischen Fingerabdruck besitzen. Die Folge "AGG" etwa kann bei einem Menschen 3-mal, bei einem anderen aber 5-mal am selben Ort in der DNA auftreten. Zu beachten ist, dass beim genetischen Fingerabdruck keine Gene (codierende Anteile der DNA) untersucht werden, sondern Abschnitte im nicht-codierenden Anteil der DNA. Der Name ist also etwas irreführend. Man verwendet auch die Begriffe DNA-Analytik oder DNA-Profile. Das zu analysierende Muster ergibt sich aus der Verwendung sogenannter STR-Marker, die ein standardisiertes System ergeben. Die Verwendung von standardisierten Marker-Systemen macht eine nationale und internationale Verwendung durch Behörden möglich.

Durch die Verwendung als standardisiertes Verfahren und einer juristischen Würdigung hat sich über viele Jahre hinweg das Vertrauen in diese Identifizierungsmethode entwickelt. auf der anderen Seite gibt es auch Methoden, wie die morphologische Haaranalytik, welches durch wissenschaftliche Untersuchungen das Vertrauen vor Gericht verloren hat. [30,31,32,33,34]

Jedoch unterliegen standardisierte Methoden vor Gericht nicht nur einer wissenschaftlichen Evaluierung, um Vertrauen aufzubauen. In

der heutigen Zeit müssen Experten und deren Labore zertifiziert werden. Diese Zertifizierung wird bei DNA-Laboren über sogenannte Ringversuche durchgeführt. Auch die Gutachten müssen sich einem Nachweis ihrer Sachkunde unterziehen. Nur so ist gewährleistet, dass die Qualität der Analyse und deren Interpretation langfristig gesichert werden kann.

Ein Blick in die Zukunft

Ein ML-System muss kein Sachbeweis sein. Eher ein Experte mit einer Aussage, der anhand von gegebenen Daten ein Analyseergebnis präsentiert. Hier sollten also die gleichen Anforderungen wie an ein Gutachten bzw. ein Bericht eines Experten gestellt werden.

Je nach Art der Kontrolle, die wir über ein ML-System ausüben wollen, werden unterschiedliche Formalisierungen notwendig. In [1] werden ein paar der Zusammenhänge zwischen Formalisierung und den dazugehörigen Herausforderungen abgeleitet. Diese sind in Abb. 4 zusammengefasst.

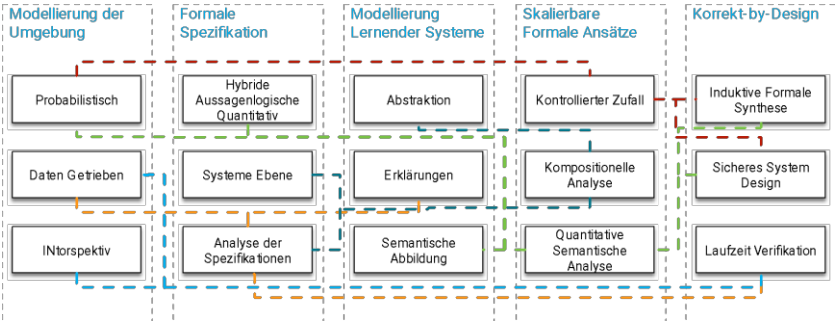

Abb. 4: Zusammenfassung der Herausforderungen in der Verifikation von ML-Systemen. (aus [1])

Wir machen hier die Einschränkung auf „schwache KI“, was wir hier ML-Systeme nennen, da eine Verifikation von „starker KI“ mit autonomem Verhalten weitere Arten der Kontrolle voraussetzen würde. So beispielsweise eine Überprüfung, Veränderung der Zielsetzung der KI-Systeme. Wenn die Zielsetzung eines Systems durch das System selbst verändert werden kann, entsteht autonomes Verhalten. Autonomes Verhalten bedeutet auch, dass Verantwortung für die Aktionen des Systems übernommen werden muss. Die eigenen Ziele zu setzen und dann auf deren Erfüllung hinzuarbeiten, auch wenn diese langfristig sind, ist mit den hier beschriebenen Methoden noch nicht kontrollierbar. Hier sind nicht nur technische Fragestellungen zu klären (beispielsweise die Priorisierung von Ziele oder der Umgang mit Konflik-

ten in Zielen), sondern auch Fragen der „starken KI“, wie braucht es dazu ein Bewusstsein, welche Entscheidungen werden unter Bedingungen wie Ressourcenbegrenzung getroffen oder gibt es eine ethische Dimension in der Bewertung von Zielen und der Planung diese zu erreichen. So kommen wir zu der Frage, ob ein hochkomplexes System überhaupt noch verstanden werden kann. Unter der Annahme, dass einem Zuwachs an kognitiver Fähigkeit, technisch kaum Grenzen gesetzt sind, könnte ein evolutionärer Schritt sein, dass hier Systeme entstehen, die so komplex sind, dass sie von Menschen nicht mehr verstanden werden. So wie vielleicht eine Ameise kognitiv nicht in der Lage ist, weltpolitische Tätigkeiten von Menschen zu verstehen.

Abb. 4 zeigt die Zusammenfassung der Überlegungen aus [1]. Hier werden die verschiedenen Herausforderungen der Modellierung von beweisbarer Software zusammengefasst dargestellt. Welche der Arten von „Korrekt-bei-Design“ jedoch vor Gericht verwendbar sind, muss sich noch zeigen. Vor Gericht verwendbar sind wohl die Systeme, die besser verstanden werden können. Am besten verständlich sind die Systeme, die am wenigsten dynamisch sind. Am wenigsten dynamisch sind Systeme mit „sicherem Systemdesign“ Dies ist jedoch mit den Lernenden Systemen schwierig, da das System so entwickelt werden muss, dass das trainierte System die Überprüfbaren Eigenschaften hat. Durch ein Training hängt die Aussage des Lernens von den Trainingsdaten ab. Je dynamischer das System ist (Synthese oder noch dynamischer Laufzeitverifikation), desto schwerer ist dessen Verhalten zu verstehen. Je schwerer ein System zu verstehen ist, desto weniger kann es kontrolliert werden. Je weniger ein System kontrolliert werden kann, desto weniger Vertrauen wird dessen Ergebnissen entgegengebracht. Neben den technischen Herausforderungen aus Abb. 4 gibt es also noch die Frage des Vertrauens und dessen Zusammenhang mit Kontrolle. Des Weiteren ist die Frage, ob einem System ähnlich vertraut werden kann wie einem Experten. Wissenschaftlich könnte über die Fehlerrate eines Systems über die Zeit Vertrauen geschaffen werden. Vertrauen könnte dann entstehen, wenn die erwartete Fehlerrate klein genug ist.

Außer dem Vertrauen in die Aussagen eines Systems (vertrauen in das ML-System) gibt es noch andere Eigenschaften, die ein ML-System erfüllen muss, um vor Gericht eingesetzt werden zu können. Zu dessen Eigenschaften gehört, dass ein ML-System die Eigenschaft besitzt, sowohl entlastend als auch belastend eingesetzt zu werden („Waffengleichheit“ als Anspruch auf rechtliches Gehör vor Gericht (Art. 103 Abs. 1 GG)).

In unserem Beispiel ist damit fraglich, wie Gesichtserkennung sowohl entlastend als auch belastend eingesetzt werden kann. Hier kann wieder das Argument der „Sachbeweise“ verwendet werden, um das ML-System wie einen Experten zu interpretieren, der sowohl ent- als auch belastende Fakten erstellt. Vertrauen muss also doch sein. Kontrolle ist nur ein Teil, um Vertrauen schaffen zu können. Irgendwann werden Systeme so komplex, dass sie nur noch schwer kontrolliert werden können.

Referenzen:

[1] Seshia, S. A., Sadigh, D., & Sastry, S. S. (2022). Toward verified artificial intelligence. *Communications of the ACM*, *65*(7), 46-55.

[2] Owre, Sam, John M. Rushby, and Natarajan Shankar. "PVS: A prototype verification system." *Automated Deduction—CADE-11: 11th International Conference on Automated Deduction Saratoga Springs, NY, USA, June 15–18, 1992 Proceedings 11*. Springer Berlin Heidelberg, 1992.

[3] Rushby, J., & von Henke, F. (1991). Formal verification of algorithms for critical systems. *ACM SIGSOFT Software Engineering Notes*, *16*(5), 1-15.

[4] Beyer, D., & Lemberger, T. (2017). Software Verification: Testing vs. Model Checking: A Comparative Evaluation of the State of the Art. In *Hardware and Software: Verification and Testing: 13th International Haifa Verification Conference, HVC 2017, Haifa, Israel, November 13-15, 2017, Proceedings 13*(pp. 99-114). Springer International Publishing.

[5] Leucker, M., & Schallhart, C. (2009). A brief account of runtime verification. *The journal of logic and algebraic programming*, *78*(5), 293-303.

[6] Ritz, S., Pankert, M., & Meyr, H. (1992, January). High level software synthesis for signal processing systems. In *Proceedings of the international conference on application specific array processors* (pp. 679-680). IEEE Computer Society.

[7] Perelman, D., Gulwani, S., Grossman, D., & Provost, P. (2014). Test-driven synthesis. *ACM Sigplan Notices*, *49*(6), 408-418.

[8] Fähndrich, J., Masuch, N., Yildirim, H., & Albayrak, S. (2014). Towards automated service matchmaking and planning for multi-agent systems with OWL-S–approach and challenges. In *Service-Oriented Computing–ICSOC 2013 Workshops: CCSA, CSB, PASCEB, SWESE, WESOA, and PhD Symposium, Berlin, Germany, December 2-5, 2013. Revised Selected Papers 11* (pp. 240-247). Springer International Publishing.

[9] Vaithilingam, P., Zhang, T., & Glassman, E. L. (2022, April). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi conference on human factors in computing systems extended abstracts* (pp. 1-7).

[10] Tamura, G., Villegas, N. M., Müller, H. A., Sousa, J. P., Becker, B., Karsai, G., ... & Wong, K. (2013). Towards practical runtime verification and

validation of self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II: International Seminar, Dagstuhl Castle, Germany, October 24-29, 2010 Revised Selected and Invited Papers* (pp. 108-132). Springer Berlin Heidelberg.

[11] Park, N., & Parker, A. C. (1988). Sehwa: A software package for synthesis of pipelines from behavioral specifications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *7*(3), 356-370.

[12] Bencomo, N., France, R. B., Cheng, B. H., & Aßmann, U. (Eds.). (2014). *Models@ run. time: foundations, applications, and roadmaps* (Vol. 8378). Springer.

[13] Lehmann, G., Blumendorf, M., Trollmann, F., & Albayrak, S. (2011). Meta-modeling runtime models. In *Models in Software Engineering: Workshops and Symposia at MODELS 2010, Oslo, Norway, October 2-8, 2010, Reports and Revised Selected Papers 13* (pp. 209-223). Springer Berlin Heidelberg.

[14] Trollmann, F., Fähndrich, J., & Albayrak, S. (2018, May). Hybrid adaptation policies: towards a framework for classification and modelling of different combinations of adaptation policies. In *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems* (pp. 76-86).

[15] Ben-Sassi, N., Dang, X. T., Fähndrich, J., Görür, O. C., Kuster, C., & Sivrikaya, F. (2018). Service Discovery and Composition in Smart Cities. In *Information Systems in the Big Data Era: CAiSE Forum 2018, Tallinn, Estonia, June 11-15, 2018, Proceedings 30* (pp. 39-48). Springer International Publishing.

[16] Hrabia, C. E., Masuch, N., & Albayrak, S. (2015). A metrics framework for quantifying autonomy in complex systems. In *Multiagent System Technologies: 13th German Conference, MATES 2015, Cottbus, Germany, September 28-30, 2015, Revised Selected Papers 13* (pp. 22-41). Springer International Publishing.

[17] Yampolskiy, R. V. (2012, April). AI-complete, AI-hard, or AI-easy–classification of problems in AI. In *The 23rd Midwest Artificial Intelligence and Cognitive Science Conference, Cincinnati, OH, USA*.

[18] Andreotta, A. J. (2021). The hard problem of AI rights. *Ai & Society*, *36*(1), 19-32.

[19] Matichuk, D., Murray, T., Andronick, J., Jeffery, R., Klein, G., & Staples, M. (2015, May). Empirical study towards a leading indicator for cost of formal software verification. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering* (Vol. 1, pp. 722-732). IEEE.

[20] Ma, Y., Jun, K. S., Li, L., & Zhu, X. (2018). Data poisoning attacks in contextual bandits. In *Decision and Game Theory for Security: 9th International Conference, GameSec 2018, Seattle, WA, USA, October 29–31, 2018, Proceedings 9* (pp. 186-204). Springer International Publishing.

[21] Cihon, P., Kleinaltenkamp, M. J., Schuett, J., & Baum, S. D. (2021). AI certification: Advancing ethical practice by reducing information asymmetries. *IEEE Transactions on Technology and Society*, *2*(4), 200-209.

[22] Kemker, R., McClure, M., Abitino, A., Hayes, T., & Kanan, C. (2018, April). Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).

[23] Zhu, X., & Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *The Artificial Intelligence Review*, *22*(3), 177.

[24] Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, *76*(3), 503-514.

[25] Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, *44*(1), 1-12.

[26] Lerman, L., Poussier, R., Bontempi, G., Markowitch, O., & Standaert, F. X. (2015). Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis). In *Constructive Side-Channel Analysis and Secure Design: 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers 6* (pp. 20-33). Springer International Publishing.

[27] Matus, K. J., & Veale, M. (2022). Certification systems for machine learning: Lessons from sustainability. *Regulation & Governance*, *16*(1), 177-196.

[28] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, *58*, 82-115.

[29] Samek, W., Wiegand, T., & Müller, K. R. (2017). *EXPLAINABLE ARTIFICIAL INTELLIGENCE: UNDERSTANDING, VISUALIZING AND INTERPRETING DEEP LEARNING MODELS*.

[30] Bioanalytik: F. Lottspeich (Herausgeber), J.W. Engels; Spektrum Akademischer Verlag, 3. Aufl. 2012

[31] Forensik in der digitalen Welt: *Moderne Methoden der forensischen Fallarbeit in der digitalen und digitalisierten realen Welt*: D. Labudde, M. Spranger; Springer Spektrum; 1. Aufl. 2017

[32] Bioinformatik im Handlungsfeld der Forensik: D. Labudde, M. Mohaupt, Springer Spektrum; 1. Aufl. 2018

[33] DNA Fingerprinting Technique in Forensic Investigation: *Application of DNA Fingerprinting Technique in Forensic Investigation*: Case Studies: A. Mishra, LAP LAMBERT Academic Publishing (2019)

[34] Fundamentals of Forensic DNA Typing: J.M. Butler, Academic Press (2009)