Nils Röttger · Gerhard Runze · Verena Dietrich

Basiswissen

KI-Testen

Qualität von und mit KI-basierten Systemen

Aus- und Weiterbildung zum
Certified Tester AI Testing
- Foundation Level Specialist
- nach ISTQB®-Standard

dpunkt.verlag

Nils Röttger hat an der Universität in Göttingen Informatik studiert. Bereits während des Masterstudiums lag sein Schwerpunkt im Themengebiet Softwaretest und Qualitätssicherung, in dem er seit über 15 Jahren tätig ist. Seit 2008 arbeitet er bei der imbus AG in Möhrendorf, aktuell als Seniorberater und Projektleiter. Er ist u.a. für die fachliche Aus- und Weiterbildung sowie den Bereich Mobile Testing verantwortlich und als Scrum Master im internen KI-Team tätig. Außerdem beschäftigt er sich immer wieder mit neuen Themen und berichtet darüber in vielen Vorträgen, zuletzt insbesondere mit Bezug zur künstlichen Intelligenz.

Dr. Gerhard Runze hat an der Friedrich-Alexander-Universität Erlangen-Nürnberg Elektrotechnik studiert und dort im Bereich digitaler Signalverarbeitungsalgorithmen promoviert. Er hat über viele Jahre als Entwickler, Projekt- und Testteamleiter in der Telekommunikationsindustrie in klassischen und agilen Projekten gearbeitet. Seit 2015 ist er bei der imbus AG als Testmanager, Trainer für ISTQB®-Schulungen und Seniorberater für Qualitätssicherung von KI, Embedded Software und agiles Testen tätig. Seit 2020 ist er zudem Product Owner für KI-Themen und hält Schulungen zum Certified Tester AI Testing.

Verena Dietrich hat an der Friedrich-Alexander-Universität Erlangen-Nürnberg Integrated Life Sciences studiert. In ihrem Masterstudium wählte sie Vorlesungen aus den Bereichen Bioinformatik, biologisch inspirierte Algorithmen und maschinelles Lernen. Von 2019 bis 2021 war sie bei der imbus AG als Softwaretesterin und Trainerin für die A4Q-Schulung KI-Testen tätig. Als Mitglied im KI-Team hat sie mit Nils und Gerhard die Arbeiten an diesem Buch begonnen und auch nach ihrem Wechsel in die Bioinformatik mit großem Engagement fortgeführt.

Nils Röttger · Gerhard Runze · Verena Dietrich

Basiswissen KI-Testen

Qualität von und mit KI-basierten Systemen

**Aus- und Weiterbildung zum
Certified Tester AI Testing –
Foundation Level Specialist nach ISTQB®-Standard**

dpunkt.verlag

Nils Röttger
nils.roettger@imbus.de

Gerhard Runze
gerhard.runze@imbus.de

Verena Dietrich
dietrich.verena@web.de

Lektorat: Christa Preisendanz
Lektoratsassistenz: Julia Griebel
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: inpunkt[w]o, Wilnsdorf *(www.inpunktwo.de)*
Herstellung: Stefanie Weidner, Frank Heidt
Umschlaggestaltung: Helmut Kraus, *www.exclam.de*

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

Fachliche Beratung und Herausgabe von dpunkt.büchern zum Thema »ISTQB® Certified Tester«:
Prof. Dr. Andreas Spillner · *Andreas.Spillner@hs-bremen.de*

ISBN:
Print 978-3-86490-947-4
PDF 978-3-96910-993-9
ePub 978-3-96910-994-6

1. Auflage 2024

Wieblinger Weg 17
69123 Heidelberg

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *hallo@dpunkt.de*.

Einleitung

In diesem Kapitel erzählen wir, wie es dazu kam, dieses Buch zu schreiben. Außerdem geben wir dir eine Art Bedienungsanleitung an die Hand, um dieses Buch zu lesen und damit zu arbeiten: Wo steht was, warum machen wir dies oder jenes? Und wir beschreiben das Zusammenspiel zwischen dem ISTQB®-Lehrplan [GTB 2021] und diesem Buch. Fachliche Inhalte findest du ab Kapitel 1.

Wie es dazu kam, dieses Buch zu schreiben

Als die Idee entstand, dieses Buch zu schreiben, haben wir alle drei bei der imbus AG in Möhrendorf gearbeitet. imbus beschäftigt sich seit vielen Jahren u.a. mit der Qualitätssicherung (QS) von IT-Systemen. Um immer auf dem Laufenden zu bleiben, gibt es jedes Jahr neue interne Projekte, die sich mit neuen Technologien, neuen Methoden, neuen Testvorgehen etc. befassen. Vor etwa fünf Jahren wurde die Idee geboren, sich in diesem Rahmen intensiv dem Thema »künstliche Intelligenz« (KI) zu widmen. Kurz zuvor hatte Nils sich mit dem Thema Ethik im Kontext von QS für KI auseinandergesetzt. Nils wollte sein Wissen folglich in dieses Projekt einbringen. Gerhard hatte in seinem Berufsleben immer mal wieder mit KI zu tun und war ebenfalls von Beginn an dabei. Verena hatte sich bereits während ihres Studiums mit dem Thema künstliche Intelligenz beschäftigt. Ihre Expertise in diesem Bereich passte also sehr gut ins Team. Das interne Forschungsprojekt hatte und hat bis heute das Ziel, in das Thema Testen von und Testen mit KI einzutauchen. Wir wollten und wollen neue Methoden finden und etablieren, um erfolgreich KI-Systeme qualitätssichern zu können. Unsere Zusammenarbeit lief gut und wir schrieben gemeinsam Artikel oder hielten Vorträge auf Konferenzen. Verena und Gerhard übernahmen dann auch die KI-Schulung zum »A4Q AI and Software Testing« [A4Q 2019], die seit Dezember 2020 bei imbus angeboten wurde. Als dann im ISTQB® (International Software Testing Qualifications Board, *istqb.org*) der neue Lehrplan zum KI-Testen erstellt wurde und dessen Review anstand, engagierte sich Gerhard dort.

Wir flachsten immer mal wieder darüber, ein gemeinsames Buch zum Thema KI und QS zu schreiben. Aus diesen Scherzen wurde dann irgendwann Ernst. Wir hoffen, dass das Ergebnis nicht nur uns, sondern auch dir gefällt.

Wie verhält sich das Buch zum Lehrplan des ISTQB®?

Ende des Jahres 2021 hat das ISTQB® den Lehrplan zum »Certified Tester AI Testing« (CT-AI) veröffentlicht [ISTQB 2021a]. Dieser beinhaltet im Wesentlichen zwei Themen:

- Das Testen *von* KI-basierten Systemen
- Das Testen *mit* KI-basierten Systemen, also die Testunterstützung durch KI-basierte Systeme

Das Buch deckt beide Bereiche ab. Da sich unser beruflicher Alltag aber viel häufiger um den Test *von* KI-basierten Systemen dreht, konzentrieren wir uns in diesem Buch – wie auch der Lehrplan selbst – darauf. Hier werden wir die einzelnen Passagen mit Praxisbeispielen aus unserem Projektalltag untermalen, viele Übungen beschreiben und Lösungsbeispiele aufzeigen.

Wir haben den Anspruch, dass dieses Buch dabei hilft, eine Schulung für den CT-AI mit anschließender Zertifizierungsprüfung erfolgreich zu absolvieren, und zugleich ein Nachschlagewerk für den Arbeitsalltag von Testerinnen und Testern darstellt. Häufig ist ein Lehrplan allein nicht geeignet, um den Kurs im Selbststudium zu absolvieren. In diesem Fall kann das Buch als Begleitmaterial zu einer Schulung oder zum Selbststudium verwendet werden. Das Buch wurde von uns so konzipiert, dass es die Inhalte des Lehrplans abdeckt, diese entsprechend mit weiteren Informationen anreichert sowie die im Lehrplan genannten Übungen aufzeigt und dabei hilft, sie zu meistern. An manchen Stellen erschien es uns sinnvoll, zusätzliche Übungen einzubauen.

Aufgrund unserer Erfahrungen als Trainerin und Trainer sowohl für den ISTQB® CT-AI als auch dessen Vorgänger »A4Q AI and Software Testing« [A4Q 2019] sind wir sicher, dass dieses Buch eine gute Unterstützung bei der Vorbereitung auf die Zertifizierungsprüfung sein wird. Um die eigenen Kenntnisse zu überprüfen, empfehlen wir die Bearbeitung der zusammen mit dem Lehrplan veröffentlichten ISTQB®-CT-AI-Übungsfragen [ISTQB 2023]. Die originale Veröffentlichung des Lehrplans ist in Englisch verfasst. Das German Testing Board (GTB) hat im darauffolgenden Jahr sowohl den Lehrplan als auch die Übungsfragen in deutscher Sprache veröffentlicht (vgl. [GTB 2021]; [ISTQB 2023]).

Die Kapitel in diesem Buch stimmen mit den zugehörigen Kapiteln des ISTQB®-CT-AI-Lehrplans überein. An einigen Stellen haben wir in den (Unter-)Kapiteln selbst Exkurse eingefügt. Diese ändern jedoch nichts an der Übereinstimmung der einzelnen Abschnitte mit dem Lehrplan. Die Schlüsselbegriffe für die einzelnen Kapitel führen wir in Englisch und Deutsch auf.

Das ISTQB®- und das Certified-Tester-Ausbildungsschema

Das International Software Testing Qualifications Board (ISTQB®) ist eine global agierende Organisation, die das erfolgreichste Ausbildungs- und Zertifizierungsschema zum Testen von Software entwickelt hat. Es ist in regionalen bzw. nationalen Boards organisiert. Im deutschsprachigen Raum sind dies das Austrian Testing Board (ATB), das German Testing Board (GTB) und das Swiss Testing Board (STB). Das Ausbildungsschema des ISTQB® ist in drei Stufen organisiert: Foundation Level, Advanced Level und Expert Level. Innerhalb jeder Stufe gibt es Lehrpläne zu mehreren Themenbereichen rund um den (Software-)Test, die von ehrenamtlich tätigen Autorinnen und Autoren aus Industrie und Forschung erstellt werden. Neben diesen drei Stufen gibt es weitere Spezialmodule und Module zum Thema »Testen in agilen Projekten«.

Weitere Informationen zur Organisation, dem Ausbildungs- und Zertifizierungsschema sowie den Lehrplänen finden sich auf der Webseite des ISTQB® [URL: ISTQB] sowie auf den Seiten der nationalen Testing Boards (vgl. [URL: ATB]; [URL: GTB]; [URL: STB]).

Für den ISTQB® »Certified Tester Foundation Level« (CTFL) wurde im August 2023 ein neuer Lehrplan in der Version 4.0 – zunächst in englischer Sprache – veröffentlicht [GTB 2023]. In diesem sind beispielsweise Begriffe wie *Testorakel* entfallen. Der aktuelle CT-AI-Lehrplan bezieht sich noch auf den CTFL in der Version 3.1. Für dich als Leserin oder Leser ist es daher wichtig, zumindest für die Zertifikatsprüfung die Begrifflichkeiten der Version 3.1 des CTFL parat zu haben. Wir beziehen uns im Buch immer auf den CTFL 3.1 [GTB 2018].

Schlüsselbegriffe und Keywords

Zu Beginn jedes Kapitels listen wir die Schlüsselbegriffe und die dazugehörigen Begriffe aus dem Lehrplan auf. Zusätzliche Begriffe, die uns wichtig erscheinen, ergänzen diese Auflistung.

Manche Begriffe verwenden wir im Projektalltag in der Regel in Englisch. Ein Beispiel ist *Bias*. Der Begriff wird auch im Deutschen oft verwendet, der deutsche Begriff *Verzerrung* hingegen eher weniger. Wir haben versucht, diesem Umstand im Buch gerecht zu werden, und verweisen dann bei solchen Begriffen auf den Projektalltag. In einer möglichen Zertifikatsprüfung solltest du auf jeden Fall den deutschen Begriff kennen.

Notwendige Vorbereitungen für die praktischen Übungen

Dieses Buch enthält zu vielen Kapiteln auch praktische Übungen. In den meisten veranschaulichen wir die Lerninhalte an Codebeispielen in der Programmiersprache *Python*. Die Aufgaben und Lösungen befinden sich auf GitHub unter *https://github.com/KI-Testen/Uebungen*, unserem KI-Testen-Repository. Wir verwenden hauptsächlich *Jupyter Notebooks*[1], da diese es uns ermöglichen, den Code für dich anschaulich zu dokumentieren, und du ihn interaktiv ausführen kannst. *Jupyter Notebooks* finden nicht nur an (Hoch-)Schulen, sondern auch in Lehrmaterialien häufig Verwendung.

Um unsere Jupyter Notebooks nutzen zu können, benötigst du Zugriff auf eine JupyterLab-Umgebung. Wir empfehlen die Installation auf deinem eigenen PC oder Laptop. Dann kannst du darauf Python und die notwendigen Bibliotheken selbst installieren. Eine Anleitung dafür findest du in unserem GitHub-Repository.

Wenn du auf deinem PC nichts installieren kannst oder willst, gibt es verschiedene Möglichkeiten, auf bereits installierte JupyterLab-Umgebungen im Web zuzugreifen, wie z. B.:

- GitHub Account: über *https://github.com/codespaces* einloggen und mit dem Jupyter Notebook-Template einen Codespace anlegen.
- Mit Google-Konto: auf *https://colab.research.google.com/* mit deinem Google Account einloggen.
- Auf Jupyter.org: ohne Login für Versuche ein JupyterLab über *https://jupyter.org/try-jupyter* starten.

Du musst in jeder dieser Varianten die Dateien aus unserem Git-Repository manuell dort hineinkopieren. Achte dabei darauf, die Verzeichnisstruktur beizubehalten. Wir können jedoch nicht garantieren, dass diese Umgebungen immer verfügbar sind und mit unseren Übungen funktionieren.

Weitere Hinweise

Anrede

Wir arbeiten meist in jungen und agilen Teams, in denen wir Produkte mitentwickeln und testen. In diesen steht das Team im Mittelpunkt und der Kommunikationsstil ist so angepasst, dass sich in der Regel alle Teammitglieder duzen. Die-

1. Einen Einstieg in Jupyter Notebooks und wie man damit arbeitet, findest du auf dieser Webseite: *https://jupyter-tutorial.readthedocs.io/de/latest/index.html*.

sen Kommunikationsstil haben wir auch für dieses Buch übernommen und duzen daher die Leserin und den Leser. Selbstverständlich kann uns auch jeder duzen. Wir hoffen, dass du dich durch diesen Kommunikationsstil nicht unangemessen angesprochen oder sogar angegriffen fühlst. Sollte dies doch der Fall sein, möchten wir uns dafür entschuldigen.

Geschlechtergerechte Sprache

Wie bereits in den vorangehenden Abschnitten zu sehen war, versuchen wir im gesamten Buch, eine geschlechtergerechte Sprache zu verwenden. Wir nutzen sowohl die weibliche als auch die männliche Form. Zum Beispiel schreiben wir die Nutzerin und der Nutzer. Auf die Verwendung von Gendersternchen oder Ähnlichem verzichten wir an dieser Stelle. Wir tun dies ausschließlich aus Gründen der besseren Lesbarkeit. Wir beabsichtigen damit keine Diskriminierung von Menschen unterschiedlichen Geschlechts. Sollten sich manche durch diese Entscheidung diskriminiert fühlen, möchten wir uns dafür entschuldigen und versichern, dass dies nicht unsere Absicht ist.

Praxisbeispiele

Die Praxisbeispiele stammen überwiegend aus unserer beruflichen Praxis. Um den Schutz der Kunden und deren Geschäftsgeheimnisse zu wahren, mussten wir abstrahieren oder kleinere Anpassungen vornehmen.

Einige Praxisbeispiele haben keinen direkten Bezug zu Projekten mit unserer Beteiligung. Sie sind, wie ein Spamfilter für E-Mails, aber Beispiele, die uns und vielleicht auch dir im täglichen Leben begegnen.

Die Praxisbeispiele sind folgendermaßen gekennzeichnet:

Praxisbeispiel 0–1: Beispielthema (ggf. Erweiterung)

So stellen wir Praxisbeispiele dar. Bei diesem Kasten handelt es sich um ein Anschauungsbeispiel, das zeigen soll, wie wir Situationen aus unserem (oder einem vergleichbaren) Projektalltag im weiteren Verlauf des Buches beschreiben.

Englische Abbildungen

Wir haben echte Screenshots eingefügt, um das Geschriebene bildlich zu verdeutlichen. Es gibt ein Bild nur mit englischem Inhalt. Da dieses Bild selbsterklärend ist und den Text ergänzen soll, wurde auf eine Übersetzung verzichtet.

Übungen

Wir empfehlen dringend, die im Buch enthaltenen Übungen durchzuführen und nicht nur zu lesen. Diese selbst durchzuführen, fördert wesentlich das Verständnis der Lerninhalte. Theoretische Kenntnisse wirken nur unterstützend für die praktische Anwendung. Die Übungen helfen dir, Ideen zu generieren und festzulegen, wie du mit der aktuellen Aufgabenstellung im Projektkontext umgehen kannst. Vielleicht findet sich für die Übungen eine Sparringspartnerin oder -partner aus der Testing Community oder unter den Arbeitskollegen.

Alle Aufgaben und Lösungen zu den Übungen findest du im online verfügbaren GitHub-Repository (*https://github.com/KI-Testen/Uebungen*).

Hinweis zum Glossar

Solltest du im Buch auf einen dir unbekannten Begriff stoßen, empfehlen wir dir, direkt in unserem Glossar am Ende des Buches nachzuschlagen. Wir haben darin alle Begriffe, die unserer Meinung nach zum besseren Verständnis beitragen, aufgelistet und mit dem domänenspezifischen Glossar des CT-AI-Lehrplans [GTB 2021] und dem allgemeinen Glossar des ISTQB® [URL: Glossar] abgeglichen. Bei den Begriffen und/oder Definitionen kann es andere Formulierungen geben, die Inhalte stimmen jedoch aus unserer Sicht überein. Sollten in diesem Buch weitere unbekannte Wörter oder Begriffe auftauchen, empfehlen wir dir als erste Anlaufstelle die beiden genannten Quellen.

Quellen und Links

Quellen und Links wurden von uns zuletzt im August 2023 überprüft. Sollten sich nach diesem Datum Änderungen ergeben haben, sind diese nicht im Buch berücksichtigt.

Danksagungen

Wir möchten Danke sagen: Ein großes Dankeschön geht an Anna, Brigitte und Felix für ihre Unterstützung, ihre Geduld und ihre Nachsicht während der letzten fast zweieinhalb Jahre.

Danke auch an Florian Fieber und Binia Sonnen für ihre vielen Anmerkungen und Tipps, die das Buch qualitativ deutlich aufgewertet haben. Ebenso bedanken wir uns bei Andreas Spillner, Mario Winter und Jonas Mönnich für ihre wertvollen Anmerkungen sowie bei unseren zahlreichen Kolleginnen und Kollegen, mit denen wir viele Gespräche über das Thema KI und damit auch über das Buch geführt haben.

Zu guter Letzt geht unser Dank auch an den dpunkt.verlag und Christa Preisendanz für die gute Betreuung während des Schreibens.

Inhaltsübersicht

1	Einführung in KI	1
2	Qualitätsmerkmale KI-basierter Systeme	35
3	Maschinelles Lernen (ML) – ein Einstieg	59
4	ML-Daten – ein Einstieg	85
5	Funktionale Leistungsmetriken – ein Einstieg	113
6	Neuronale Netze und Testen	127
7	Testen KI-basierter Systeme im Überblick	139
8	Testen KI-spezifischer Qualitätsmerkmale – ein Einstieg	169
9	Methoden und Verfahren für das Testen KI-basierter Systeme	193
10	Testumgebungen für KI-basierte Systeme	223
11	Einsatz von KI für Tests	231

Anhang 249

A	Abkürzungen	251
B	Glossar	255
C	Verzeichnis der Praxisbeispiele	273

D Verzeichnis der Übungen 275

E Verzeichnis der Exkurse 277

F Literaturverzeichnis 279

Index 291

Inhaltsverzeichnis

1 Einführung in KI 1

1.1 Definition von KI und der KI-Effekt 1
1.2 Schwache KI, starke KI und die künstliche Superintelligenz 3
1.3 KI-basierte Systeme und klassische Systeme 6
1.4 KI-Techniken 9
1.4.1 **Exkurs:** KI-Techniken im Detail 10
1.5 KI-Entwicklungs-Frameworks 15
1.6 Hardware für KI-basierte Systeme 18
1.7 KI als Service (AI as a Service, AIaaS) 22
1.8 Vortrainierte Modelle 25
1.9 Normen, Vorschriften und KI 29
1.9.1 **Exkurs:** Liste einiger Normen und Standards mit KI-Bezug 33

2 Qualitätsmerkmale KI-basierter Systeme 35

2.1 Flexibilität und Anpassbarkeit 35
2.2 Autonomie von Systemen 38
2.3 Evolution 40
2.4 Bias 42
2.4.1 **Exkurs:** Weitere Arten des Bias 44
2.5 Ethik 46
2.6 Seiteneffekte und Reward Hacking 49
2.6.1 Seiteneffekte 50
2.6.2 Reward Hacking 50

2.7 Transparenz, Interpretierbarkeit und Erklärbarkeit 51
2.8 Funktionale Sicherheit und KI 56

3 Maschinelles Lernen (ML) – ein Einstieg 59

3.1 Arten des maschinellen Lernens (ML) 59
3.1.1 Überwachtes Lernen 60
3.1.2 Unüberwachtes Lernen 63
3.1.3 Bestärkendes Lernen 65
3.1.4 **Exkurs:** Das Wissen einer KI – der Unterschied zwischen Korrelation und Kausalität 67
3.2 ML-Workflow 69
3.2.1 **Exkurs:** Alternative Workflows 76
3.3 Auswahl einer Art von ML 76
3.3.1 Übung: Wahl der passenden ML-Art 78
3.4 Faktoren, die bei der Auswahl von ML-Algorithmen eine Rolle spielen 79
3.5 Overfitting und Underfitting 81
3.5.1 Overfitting 82
3.5.2 Underfitting 83
3.5.3 Übung: Demonstration von Overfitting und Underfitting . . 84

4 ML-Daten – ein Einstieg 85

4.1 Datenvorbereitung als Teil des ML-Workflows 85
4.1.1 Datenbeschaffung 87
4.1.2 Vorverarbeitung der Daten 88
4.1.3 Merkmalsermittlung 91
4.1.4 Herausforderungen bei der Datenvorbereitung 93
4.1.5 Übung: Datenvorbereitung für ML 94
4.2 Trainings-, Validierungs- und Testdatensätze 95
4.2.1 Übung: Identifizieren von Trainings- und Testdaten und Erstellen eines ML-Modells 99
4.2.2 **Exkurs:** Aufteilungsmethoden für Trainings- und Validierungsdaten 99
4.3 Probleme mit der Datensatzqualität 102

4.4 Datenqualität und ihre Auswirkungen auf das ML-Modell 104
4.4.1 Übung: Aspekte der Datenqualität 108
4.5 Datenkennzeichnung für überwachtes Lernen 108
4.5.1 Ansätze zur Datenkennzeichnung 110
4.5.2 Falsch gekennzeichnete Daten in Datensätzen 111

5 Funktionale Leistungsmetriken – ein Einstieg 113

5.1 Konfusionsmatrix 113
5.1.1 Übung: Metriken einsetzen 116
5.2 Zusätzliche funktionale Leistungsmetriken von ML für Klassifikation, Regression und Clusterbildung 116
5.3 Beschränkungen der funktionalen Leistungsmetriken von ML 120
5.4 Auswahl funktionaler Leistungsmetriken von ML 122
5.4.1 Übung: Evaluieren eines erstellten ML-Modells 125
5.5 Benchmark-Suiten für ML 125

6 Neuronale Netze und Testen 127

6.1 Neuronale Netze 127
6.1.1 Übung: Training eines neuronalen Netzes 134
6.2 Überdeckungsmaße für neuronale Netze 135

7 Testen KI-basierter Systeme im Überblick 139

7.1 Spezifikation KI-basierter Systeme 139
7.2 Teststufen für KI-basierte Systeme 143
7.2.1 Eingabedatentest 144
7.2.2 ML-Modelltest 145
7.2.3 Komponententest 148
7.2.4 Komponentenintegrationstest 148
7.2.5 Systemtest 149
7.2.6 Abnahmetest 151
7.3 Testdaten zum Testen KI-basierter Systeme 151
7.4 Testen auf Automatisierungsbias in KI-basierten Systemen 154
7.5 Dokumentieren einer KI-Komponente 155
7.6 Testen auf Konzeptdrift 160
7.7 Auswahl einer Testvorgehensweise für ein ML-System 162

8 Testen KI-spezifischer Qualitätsmerkmale – ein Einstieg 169

8.1 Herausforderungen beim Testen selbstlernender Systeme 169
8.2 Test von autonomen KI-basierten Systemen 174
8.3 Testen auf algorithmischen, stichprobenartigen und unangemessenen Bias 176
8.4 Herausforderungen beim Testen probabilistischer und nichtdeterministischer KI-basierter Systeme 180
8.5 Herausforderungen beim Testen komplexer KI-basierter Systeme . . 181
8.5.1 Übung: Herausforderungen bei der Verwendung eines künstlichen neuronalen Netzes 183
8.6 Testen der Transparenz, Interpretierbarkeit und Erklärbarkeit KI-basierter Systeme 183
8.6.1 Übung: Erklärbare KI 186
8.7 Testorakel für KI-basierte Systeme 186
8.8 Testziele und Akzeptanzkriterien 189
8.8.1 Übung: Akzeptanzkriterien 191

9 Methoden und Verfahren für das Testen KI-basierter Systeme 193

9.1 Gegnerische Angriffe und Datenverunreinigung 194
9.1.1 Gegnerische Angriffe 194
9.1.2 Datenverunreinigung 197
9.2 Paarweises Testen 200
9.2.1 Übung: Paarweises Testen 205
9.3 Vergleichendes Testen 206
9.4 A/B-Testen 209
9.5 Metamorphes Testen 211
9.5.1 Übung: Metamorphes Testen 214
9.6 Erfahrungsbasiertes Testen von KI-basierten Systemen 215
9.6.1 Checklisten für den Test von KI-basierten Systemen 217
9.6.2 Übung: Exploratives Testen und explorative Datenanalyse (EDA) 219
9.7 Übersicht und Auswahl von Testverfahren für KI-basierte Systeme 219
9.7.1 Übersicht der Verfahren 220
9.7.2 Übung: Verfahren für das Testen KI-basierter Systeme 222

10 Testumgebungen für KI-basierte Systeme 223

10.1 Besonderheiten von Testumgebungen für KI-basierte Systeme 223

10.2 Virtuelle Testumgebungen zum Testen KI-basierter Systeme 226

11 Einsatz von KI für Tests 231

11.1 KI-Techniken fürs Testen 231

11.1.1 Algorithmische Methoden, mit denen KI unterstützt 233

11.1.2 Übung: Der Einsatz von KI bei Tests 235

11.2 Einsatz von KI zur Analyse gemeldeter Fehler 235

11.3 Einsatz von KI für die Testfallgenerierung 237

11.4 Einsatz von KI für die Optimierung von Regressionstestsuiten 239

11.5 Einsatz von KI für die Fehlervorhersage 240

11.5.1 Übung: Aufbau eines Fehlervorhersagesystems 242

11.6 Einsatz von KI zum Testen von Benutzungsschnittstellen 242

11.6.1 Einsatz von KI zum Testen über die GUI 243

11.6.2 Einsatz von KI zum Testen der GUI 244

11.7 **Exkurs:** ChatGPT als Teammitglied? 246

11.7.1 Übung: ChatGPT zur Testfallgenerierung 247

11.7.2 Mehrwert von großen Sprachmodellen im Test 248

Anhang 249

A Abkürzungen 251

B Glossar 255

C Verzeichnis der Praxisbeispiele 273

D Verzeichnis der Übungen 275

E Verzeichnis der Exkurse 277

F Literaturverzeichnis 279

Index 291

1 Einführung in KI

»Dinge wahrzunehmen ist der Keim der Intelligenz.«
Laotse

In diesem Kapitel erklären wir allgemein den Begriff der künstlichen Intelligenz (KI) und die damit in Verbindung stehenden Systeme. Wir vermitteln, wie der Begriff entstand und sich bis heute entwickelt hat, aber auch was heutige künstlich intelligente Systeme charakterisiert und sie von konventioneller Software unterscheidet. Ebenso werfen wir einen Blick darauf, wie und mit welchen Frameworks KI-basierte Systeme entwickelt werden und was bei deren Betrieb zu beachten ist: als Service in der Cloud, auf dedizierter Hardware oder im regulatorischen Umfeld.

KI-spezifische Schlüsselbegriffe aus dem Lehrplan:
AI as a Service (AIaaS), KI-Entwicklungs-Framework (AI development framework), KI-Effekt (AI effect), KI-basiertes System (AI-based System), künstliche Intelligenz (KI, artificial intelligence), neuronales Netzwerk (neural network), Deep Learning (DL), tiefes neuronales Netzwerk (deep neural network), starke KI (general AI), Datenschutzgrundverordnung (DSGVO, General Data Protection Regulation, GDPR), maschinelles Lernen (ML, machine learning), schwache KI (narrow AI), vortrainiertes Modell (pre-trained model), künstliche Superintelligenz (super AI), technologische Singularität (technological singularity), Transfer-Lernen (transfer learning)

Weitere Schlüsselbegriffe in diesem Kapitel:
keine

1.1 Definition von KI und der KI-Effekt

Der Begriff der *künstlichen Intelligenz (KI)* ist in unserem Leben fast allgegenwärtig geworden. Wir haben ihn schon oft in den Nachrichten gehört, in Artikeln und Büchern dazu gelesen oder Filme darüber gesehen. So verwundert es nicht, wenn viele von sich sagen, den Begriff *KI* verstanden zu haben. Wir ordnen dann gerne KI als aktuelle *Hightech-Innovation* ein.

Neu ist die Idee einer von Menschen geschaffenen Intelligenz aber keinesfalls. Schon weit vor dem heute beobachtbaren gesellschaftlichen Hype führte die wissenschaftliche Forschung den Begriff der künstlichen Intelligenz ein. Er geht auf die 1950er-Jahre zurück, als Forscherinnen und Forscher der damals bereits wachsenden technischen Möglichkeiten ein besonderes Ziel vor Augen hatten: Rechenmaschinen zu bauen, die den Menschen und seine Fähigkeiten imitieren können. Als wichtigstes Ereignis wird heute der Workshop am Dartmouth College in New Hampshire im Jahr 1956 angesehen, der erstmals eine Studie zu künstlicher Intelligenz zum Thema hatte.[1]

Seitdem beschäftigt sich die Forschung in Theorie und Praxis mit diesem Ziel. Mit den Fortschritten in der mathematischen Modellierung, den technischen Möglichkeiten und neuen potenziellen Anwendungsgebieten hat sich auch die Definition des Begriffs der künstlichen Intelligenz ständig weiterentwickelt. Im derzeitigen Sprachgebrauch versteht man Folgendes darunter:

Künstliche Intelligenz (KI): Die Fähigkeit eines technischen Systems, Wissen und Fertigkeiten zu erwerben, zu verarbeiten, zu entwickeln und anzuwenden.[a]

a. Aus der englischen Definition des technischen Reports ISO/IEC TR 29119-11 [ISO/IEC TR 29119-11].

Der Wandel, dem die Definition des KI-Begriffs bis heute unterworfen ist, erklärt sich vor allem aus einer veränderten Wahrnehmung technischer Systeme und deren Leistungen. Insbesondere ist die Grenze zwischen konventioneller Software und Systemen mit künstlicher Intelligenz nicht festgeschrieben, sondern ebenfalls einem Wandel unterworfen.

Noch in den 1970er-Jahren konnten mäßig geübte Schachspielerinnen und -spieler die damaligen Schachcomputer besiegen. Ein Programm, das in der Lage gewesen wäre, einen Schachweltmeister zu schlagen, schien dagegen unmöglich. Das Schachspiel ist ein Paradebeispiel für eine ganze Reihe von Problemen, die trotz weniger Regeln und begrenztem Spielfeld eine hohe strategische Komplexität aufweisen. Als 1997 dann *Deep Blue* den damals amtierenden Schachweltmeister Garri Kasparow besiegte, wurde dies als Siegeszug der künstlichen Intelligenz gefeiert. Aus heutiger Sicht wird die damalige Software jedoch von vielen nicht mehr als KI bezeichnet. Deep Blue nutzte damals eine *Brute-Force*-Methode, also eine auf purer Rechenleistung beruhende Methode, um den nächsten Zug zu ermitteln. Ein Erwerben oder Weiterentwickeln der eigenen Fertigkeiten, wie nach der aktuellen Definition einer KI, war hier gar nicht vorgesehen.

Mit dem Begriff der künstlichen Intelligenz verbinden wir durchaus eine Art von Bewunderung für eine unerwartete Leistung. Insbesondere, wenn man diese

1. *https://en.wikipedia.org/wiki/Dartmouth_workshop*, abgerufen am 08.08.2023.

nicht ohne Weiteres technisch, einfach und nachvollziehbar erklären kann. Eine kleine Anekdote dazu: Selbst die Entwicklerinnen und Entwickler von Deep Blue waren damals über bestimmte Züge ihres Programms überrascht. Später fanden sie heraus, dass einer diese Züge auf einen Programmierfehler zurückzuführen war, der die Maschine zunächst in eine Endlosschleife versetzte. Ein Notfallmechanismus hat dann die Zugsuche abgebrochen und per Zufall einen beliebigen ausgewählt. Just dieser völlig unorthodoxe Zug hatte Kasparow damals aus dem Konzept gebracht, weil er dahinter eine besondere Genialität vermutete. Das Blatt wendete sich und Kasparow verlor [Silver 2012].[2]

Ebenso zählte die Fachwelt *Expertensysteme* der 1970er- und 1980er-Jahre lange Zeit zu den Systemen mit künstlicher Intelligenz. Diese arbeiten mit klar beschriebenen Regeln, z.B. Wenn-dann-Bedingungen, sowie mit in Datenbanken gespeichertem Expertenwissen. Meist können sie komplizierte Fragestellungen aus einem bestimmten Wissensgebiet gut beantworten. Schon die ersten Expertensysteme zeigten, trotz einfach strukturierter Regelwerke, Fähigkeiten, die man damals nur menschlichen Fachexpertinnen und -experten zutraute. Heute werden die damaligen Expertensysteme nicht mehr zur KI gezählt, moderne Expertensysteme hingegen schon.

Die Verschiebung dieser Grenze, also die Wahrnehmung, ob man ein System als künstliche Intelligenz ansieht oder nicht, bezeichnet man als *KI-Effekt* [URL: Wikipedia]. Im Laufe der Zeit hat sich diese Einschätzung in der Gesellschaft verändert und sie verändert sich vermutlich auch weiterhin. Diesem Umstand trägt der alternative Begriff *Lernende Systeme* Rechnung, der sich in den vergangenen Jahren gerade in der Forschung etabliert hat.

In diesem Buch benutzen wir den Begriff *KI* sehr oft. Damit meinen wir die sehr weit gefasste Menge an Systemen im Sinne der oben genannten Definition. In den seltensten Fällen ist jedoch eine reine *KI-Software* allein im Einsatz. Meistens ist sie in konventionelle Software[3] eingebettet. Um dies zu betonen, werden wir diese dann auch als *KI-basierte* Systeme bezeichnen. Im weiteren Verlauf des Buches müssen wir andererseits aber auch präziser werden und sprechen dann z.B. von *logikbasierten Systemen* (siehe Abschnitt 1.4) oder *Systemen des maschinellen Lernens* (ML-Systemen, ab Kap. 3).

1.2 Schwache KI, starke KI und die künstliche Superintelligenz

Viele kennen die künstlichen Superintelligenzen in den Science-Fiction-Filmen wie *2001 – Odyssee im Weltraum* oder *Terminator*. Sie greifen nicht nur auf übermenschliches Wissen zu, sondern sind sogar fähig, die Menschheit auszulöschen. Wenn du dir aber anschaust, wozu KI heute in der Lage ist, dann wird schnell

2. *https://en.wikipedia.org/wiki/Deep_Blue_(chess_computer)#Deep_Blue_versus_Kasparov*, abgerufen am 08.08.2023.
3. In Projekten verwenden wir synonym auch den Begriff *klassische Software*.

klar, dass diese Superintelligenzen derzeit nicht realisierbar sind. KI, so wie sie heute eingesetzt und entwickelt wird, ist lediglich in der Lage, spezifische Aufgaben in einem limitierten Kontext auszuführen. Wir sprechen hier von *begrenzter* oder *schwacher* KI, da ihr Einsatzgebiet auf einen sehr begrenzten Problemraum beschränkt ist.

Schwache KI: KI, die auf eine einzelne klar definierte (engl. well-defined) Aufgabe konzentriert ist, um ein spezifisches Problem zu lösen.[a]

a. Übersetzt aus [ISO/IEC TR 29119-11].

Solche begrenzten KIs zeigen uns immer wieder beeindruckende Spielkünste in strategischen Spielen, wie z.B. *AlphaZero,* veröffentlicht von der Google-Firma DeepMind im Dezember 2017 [Silver et al. 2017]. AlphaZero kann innerhalb von wenigen Stunden eigenständig Spielstrategien verschiedener Brettspiele nur anhand der Spielregeln erlernen. Damit zeigt AlphaZero Fähigkeiten, die bei Weitem die eines Menschen übersteigen. Dennoch betrachten wir dieses System als begrenzte KI, da sie mit ihren Fähigkeiten auf das sehr spezifisches Aufgabenfeld der strategischen Brettspiele beschränkt ist.

In unserem Alltag finden wir alle mittlerweile viele Situationen, in denen schwache KIs Aufgaben übernehmen: Spamfilter, die unerwünschte E-Mails aussortieren, Sprachassistenten, die unsere Telefonate entgegennehmen, Bordcomputer in unseren Autos, die uns mitteilen, welcher Kundenservice ansteht. All diese KIs zeigen ihr Können nur in ihrem definierten Aufgabenbereich. Für neue Aufgaben außerhalb ihres vorgesehenen Einsatzgebiets sind sie nicht entwickelt.

Stell dir vor, du trainierst eine KI mit sehr vielen Hundebildern, um auf Bildern von Hunden deren Rasse erkennen zu können.[4] Wenn du der KI Hundebilder präsentierst, funktioniert diese erstaunlich gut. Was passiert aber, wenn auf dem Bild kein Hund zu sehen ist? Dafür ist die KI nicht trainiert. Sie versucht dennoch, das Bild einer Hunderasse zuzuordnen, obwohl es sich bei dem Bild gar nicht um einen Hund handelt. Die offensichtlich begrenzte KI bleibt in ihrem eng gefassten Aufgabenbereich.

Die offene Forschungsplattform *OpenAI* hat mit *GPT-3* im Juni 2020 eine KI zur Verarbeitung von Texten auf einem noch nicht dagewesenen Level veröffentlicht [Brown et al. 2020]. Seit Anfang 2023 ist die neuere Version GPT-3.5 als Chatbot *ChatGPT* für alle kostenlos verfügbar. GPT-3 ist ein durch *Deep Learning* trainiertes neuronales Netz[5], das zur Textverarbeitung im weitesten Sinne eingesetzt werden kann. Das Einsatzgebiet für GPT-3 reicht von der Erstellung von journa-

4. Auf der offenen Plattform Kaggle findest du dafür geeignete Datensätze, z.B.: *https://www.kaggle.com/jessicali9530/stanford-dogs-dataset*.
5. Wir verwenden abweichend vom Syllabus im gesamten Buch den Begriff *neuronales Netz* anstelle von *neuronalem Netzwerk*, siehe dazu unseren Hinweis zu Beginn von Kapitel 6.

listischen Artikeln über Gedichte bis hin zu Programmcode. Sind damit die Grenzen der begrenzten KI hin zur generellen KI überschritten?

Starke KI: KI, die über das gesamte Spektrum der kognitiven Fähigkeiten ein dem Menschen vergleichbares intelligentes Verhalten zeigt. [a]

a. Übersetzt aus [ISO/IEC TR 29119-11].

In der Definition der *starken* KI, angelehnt an die ISO/IEC TR 29119-11:2020 (der Lehrplan spricht hier von der *allgemeinen* KI), wird die Summe der menschlichen Fähigkeiten ins Zentrum gestellt. Im Gegensatz zur begrenzten KI sprechen wir von einer generellen KI, wenn die Fähigkeiten nicht auf eine Aufgabenstellung beschränkt ist. Eine starke KI hat weitreichende Fähigkeiten, vergleichbar mit denen eines Menschen, wie beispielsweise Begründungen zu formulieren, die Umwelt zu verstehen, Schlussfolgerungen zu ziehen und danach zu handeln.

Auch wenn die Fähigkeiten von GPT-3 so mächtig erscheinen, ist GPT-3 auf rein textuelle Aufgaben beschränkt und nicht in der Lage, wie ein Mensch Sachverhalte zu verstehen, sondern ahmt Formulierungen aus den Trainingsdaten, wie im Training gelernt, lediglich nach. Daher sprechen wir bei GPT-3 nicht von einer generellen KI. Gleiches gilt für die bessere Version GPT-4. Im Kontext der gegebenen Definitionen und nach Stand von August 2023 wurde noch keine starke KI realisiert, kurz: Es gibt keine.

Die oberste Stufe der künstlichen Intelligenz ist die der *künstlichen Superintelligenz*. Darin werden KIs zusammengefasst, die das Verhalten von uns Menschen nicht nur nachahmen können, sondern unsere Fähigkeiten durch die Verarbeitung von gewaltigen Datenmengen sogar übertreffen. Dazu nutzen sie einen praktisch unlimitierten Datenspeicher und Zugang zu allem menschlichen Wissen, wie es z.B. im Internet zu finden ist. Der Punkt, an dem die KI von der generellen KI zur künstlichen Superintelligenz übergeht, wird auch als *technologische Singularität* beschrieben.

Künstliche Superintelligenz: Eine KI, deren Fähigkeiten die Fähigkeiten des Menschen überschreiten bzw. weit übertreffen.

Auch wenn diese künstlichen Superintelligenzen bis jetzt nur in Filmen existieren, hielt Stephen Hawking sie in der Zukunft für realisierbar und warnt vor deren Fähigkeiten.

> *»Ich fürchte, dass die künstliche Intelligenz den Menschen insgesamt ersetzen könnte. Wenn Menschen Computerviren entwerfen, wird jemand eine künstliche Intelligenz entwerfen, die sich selbst verbessert und vermehrt. Das wird eine neue Lebensform sein, die den Menschen überragt.«*
>
> Stephen Hawking, 2017

1.3 KI-basierte Systeme und klassische Systeme

»KI ist auch nur eine Software.« So hört man es des Öfteren. Das ist sicher richtig, aber KI-Systeme werden nicht wie *klassische Software* (auch *konventionelle Software* genannt) entwickelt. Denke an klassische Softwaresysteme, hier haben Menschen die darin enthaltenen logischen Zusammenhänge und Abläufe hergeleitet. Diese werden in Form von Anweisungen wie *Wenn-dann-sonst-Konstrukten* (engl. if-then-else) und Schleifen im Programmcode festgehalten. Die dafür verwendeten Programmiersprachen werden auch als *imperative Programmiersprachen* bezeichnet. So entscheiden die Entwicklerinnen und Entwickler, bei welchem Input sich die Software wie verhalten soll.

Für Menschen ist es im Normalfall relativ einfach, mit entsprechendem Vorwissen die logischen Zusammenhänge zu verstehen. Sie können nachvollziehen, wie die Softwareeingaben in Ausgaben[6] umgewandelt werden und warum sich das System so verhält, wie es sich verhält. Selbst, wenn diese klassischen Softwaresysteme sehr komplex sind, sind doch Menschen mit Vorwissen in der Lage, diese Software zu verstehen und den Grund für ein bestimmtes Verhalten zu ermitteln.

KI-basierte Systeme, die durch *maschinelles Lernen* (engl. machine learning, ML) trainiert werden, erlernen während der Trainingsphase das gewünschte Verhalten. Um diese Systeme zu trainieren, gibt es eine Vielzahl von Techniken, auf die wir in Abschnitt 1.4 genauer eingehen. Unabhängig von der gewählten Technik, wird in der Trainingsphase eines KI-basierten Systems anhand von erlernten Datenmustern bestimmt, wie es in Zukunft auf neue Daten reagieren soll. Die zugrunde liegenden logischen Zusammenhänge werden nun nicht mehr von Entwicklerinnen oder Entwicklern vorgegeben, sondern automatisiert aus den Datenmustern durch das ML abgeleitet (für eine detaillierte Erklärung des maschinellen Lernens siehe Kap. 3). Du kannst dir leicht vorstellen, dass die Qualität der Daten einen wesentlichen Einfluss darauf hat, wie gut die KI in der Trainingsphase lernt. Wenn die Lernphase mit Daten, die keine repräsentativen Datenmuster und Merkmale beinhalten, stattgefunden hat, lernt die KI während der Trainingsphase unzureichende oder sogar falsche Regeln (siehe die Abschnitte 2.4 und 3.5).

Während bei der klassischen Softwareentwicklung die Entwicklerinnen und Entwickler direkten Einfluss auf die Logik der Software haben, haben sie bei KI-basierten Systemen nur indirekt Einfluss auf deren Verhalten, nämlich lediglich über die Wahl des Inputs während des Trainings und die Auswahl der KI-Technik und des dazugehörigen ML-Algorithmus (siehe Abb. 1–1). Ein *ML-Algorithmus* ist eine mathematische Vorgehensweise, die zur Erstellung eines ML-Modells aus einem Trainingsdatensatz verwendet wird.

---

6. Im Folgendem schreiben wir oft Input und Output. Diese Begriffe werden meist im KI-Kontext verwendet.

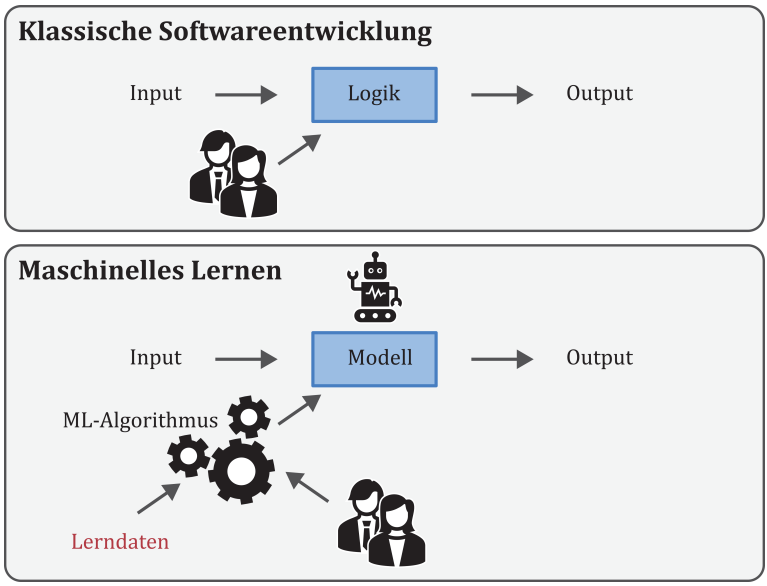

Abb. 1–1 *Klassische und KI-basierte Software im Vergleich*

KI-basierte Systeme haben im Vergleich zu klassischen Systemen meist den Nachteil, dass die logischen Regeln für Menschen weniger leicht bis überhaupt nicht nachvollziehbar sind – je nach Komplexität der KI-Technik (siehe Abschnitt 2.7). Daher spricht man bei KI-Software oft auch von einer *Blackbox*.

Um den Unterschied zwischen der Entwicklung von KI-basierten und klassischen Systemen noch deutlicher zu machen, schauen wir uns ein Praxisbeispiel aus dem Bereich des maschinellen Lernens an.

Praxisbeispiel 1–1: Testframework für eine Heizungssteuerung

Im Rahmen dieses Projekts ging es um funktionale Tests einer Heizungssteuerung (Smarthome-Steuerung). Auf einem Touchdisplay konnten die Raumtemperatur sowie viele andere Einstellungen der Heizung vorgenommen werden. Im Projekt musste für verschiedene Testszenarien bewertet werden, ob das Touchdisplay die richtigen Informationen anzeigt. Dafür wurde im Test der Smarthome-Steuerung jede Minute das Touchdisplay fotografiert und das Foto in einer Datenbank abgelegt. Mithilfe des KI-Systems sollte der Testaufwand für die manuelle Auswertung der Fotos reduziert werden. Es wurde ein Testframework entwickelt, das den Inhalt der Bilder automatisiert erkennen und auswerten sollte. Das Framework setzte sich aus klassischen und KI-Softwarekomponenten zusammen.

Ein Teil des Touchdisplays stellte eine Temperaturanzeige mit vier Sieben-Segment-Ziffern dar (siehe Abb. 1–2). Wir wählten für die sehr einfach aufgebaute Anzeige einen klassischen Softwareansatz für das Auslesen der Temperaturanzeige: Wir überlegten uns,

→

welche Merkmale auf der Anzeige ausschlaggebend sind und welche Logik im Programmcode angewandt werden musste, um festzulegen, um welche Ziffer es sich an den einzelnen Positionen handelt. Dies gestaltete sich recht aufwendig, da die Software auch bei leicht variiertem Winkel des Fotos oder sich änderndem Lichteinfall fähig sein sollte, die Anzeige abzulesen. Nach langem Tüfteln mit den Belichtungszeiten und den Kontrasteinstellungen hatten wir eine zuverlässige Testautomatisierung entworfen. Diese war in der Lage, die auf der Temperaturanzeige abgebildete Zahl mit einem Sollwert zu vergleichen.

Die Anzeige enthielt neben der Temperaturanzeige auch deutlich kompliziertere Elemente, wie Warnsymbole, ein Zeichen für den Kalibrierungsmodus und eine Batteriezustandsanzeige. Der Aufwand mit der oben beschriebenen Herangehensweise wurde dadurch ungleich größer. Da die Logik im Programmcode zu komplex zu werden drohte, entschieden wir uns für einen ML-Ansatz. Wir wählten ein neuronales Netz, das sich für diese Klassifikationsaufgabe eignete. Um die KI zu trainieren, verwendeten wir den selbst erstellten Datensatz von Bildern dieser Heizungssteuerung. Anhand dieser Trainingsdaten lernte die KI, die Merkmale und Muster zu erkennen und sie den verschiedenen Symbolen zuzuordnen. Allerdings stellten wir beim Testen der KI fest, dass diese nicht die benötigte Genauigkeit hatte. Warum die KI so schlechte Ergebnisse lieferte, konnten wir anhand des Programmcodes der KI nicht mehr erkennen. Die Logik im neuronalen Netz war für uns Menschen nicht mehr nachvollziehbar.

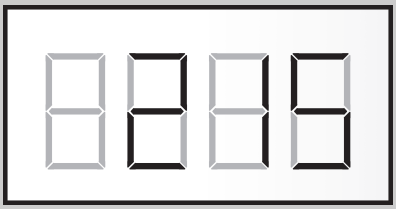

Abb. 1–2 *Sieben-Segment-Anzeige auf dem Display einer Heizungssteuerung*

Praxistipp

Auch wenn KIs beeindruckende Erfolge erzielt haben, haben sie den großen Nachteil, dass ihre Logik nur schwer bis gar nicht mehr nachvollziehbar ist. Es gibt Ansätze, die versuchen, KIs erklärbar zu machen (siehe Abschnitt 2.7). Bei sensiblen Aufgabenstellungen kann es dennoch sinnvoll sein, von der Verwendung einer KI ganz abzusehen und auf eine klassische Software zurückzugreifen. Diese ist möglicherweise aufwendiger in der Entwicklung der inneren Logik, der Entscheidungsprozess der Software ist dafür aber im Detail nachvollziehbar. Testerinnen und Tester können in einer solchen Software auch sehr einfach Reviews vornehmen und Whitebox-Tests durchführen. Reviews und Whitebox-Tests in einer KI sind im Gegensatz dazu sehr schwierig bis unmöglich.

Bei der Abgrenzung von konventionellen und KI-basierten Systemen musst du dir im Klaren darüber sein, dass sich in der Praxis die Wahrnehmung über KI-basierte Systeme in stetem Wandel befindet. Systeme, die man heute als KI-basierte Systeme bezeichnet, werden in Zukunft vielleicht den konventionellen Systemen zugeordnet (siehe Abschnitt 1.1).

1.4 KI-Techniken

Im vorangegangenen Abschnitt haben wir den wesentlichen Unterschied zwischen KI-basierter und konventioneller Software beschrieben. Insbesondere haben wir dort das maschinelle Lernen als eine der am weitverbreitetsten Entwicklungsmethoden erwähnt. Doch das ist nicht die einzige Methode. In Abbildung 1–3 zeigen wir eine Auswahl an *KI-Techniken*[7] (der ISTQB®-Lehrplan spricht hier gar von *Technologien*).

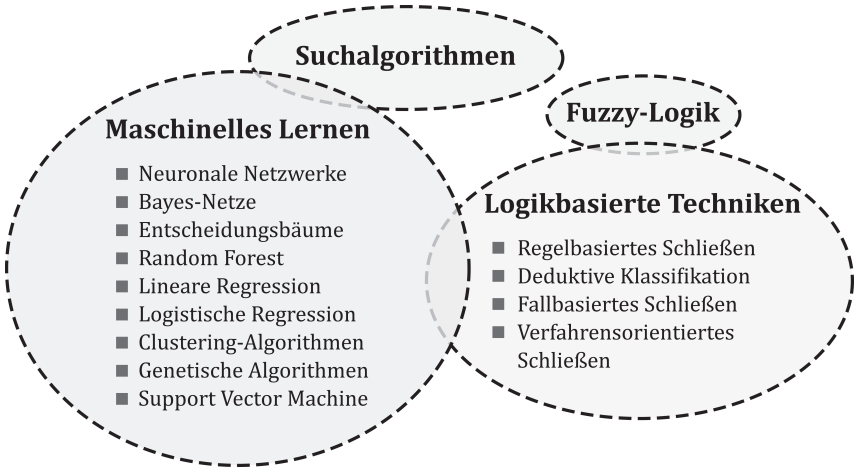

Abb. 1–3 *Eine Auswahl an KI-Techniken, mit denen KI-Komponenten entwickelt werden können.*

Wenn man eine KI-Komponente entwickeln will, muss man meist frühzeitig eine KI-Technik und einen der dazugehörigen ML-Algorithmen auswählen. Erst danach kann man mit dem Training beginnen. In Abschnitt 3.4 gehen wir näher darauf ein, welche Faktoren für die Auswahl eines Algorithmus zum Tragen kommen.

Jede KI-Technik hat ihre eigenen Vorzüge, aber auch Nachteile. Daher ist es nicht ungewöhnlich, dass man nach einem wenig erfolgreichen Training die Wahl des Algorithmus überdenken und mit einem vermutlich besser geeigneten Algorithmus das Training erneut beginnen muss.

Es kommt nicht selten vor, dass man in einem KI-basierten System auch mehr als eine einzige KI-Technik verwendet. Stell dir beispielsweise ein System vor, das eine Reisebuchung über einen Sprachdialog ermöglichen soll. Man wird es dort vermutlich mit neuronalen Netzen zur Spracherkennung zu tun haben. Die Auswahl des Reiseangebots könnte hingegen durch einen intelligenten Suchalgorithmus unterstützt werden.

7. Technisch Interessierte finden einen umfassenden Einblick in [Russell & Norvig 2020] oder eine Übersicht zu vielen weiteren KI-Techniken in [Wahlster et al. 2022, ab S. 42].

1.4.1 Exkurs: KI-Techniken im Detail

Im Folgenden gehen wir auf die in Abbildung 1–3 gezeigten KI-Techniken näher ein und beschreiben auch in groben Zügen die ein oder andere Technik, damit du diese besser einordnen kannst.

Suchalgorithmen

Im ersten Moment würdest du wahrscheinlich keine KI hinter einer automatisierten Suche vermuten. Gerade intelligente Suchalgorithmen haben eine enorme Bedeutung erlangt. Dabei darfst du nicht nur an die klassische (z. B. lineare) Suche durch eine Liste von Elementen denken. Vielmehr kommen oft spezialisierte Suchalgorithmen zum Einsatz, wie etwa die folgenden:

- **Heuristische Suchalgorithmen** finden beispielsweise nicht alle mathematisch korrekten Treffer, ermitteln dafür aber diejenigen, die im aktuellen Kontext am besten passen – und das in erstaunlich kurzer Zeit. Denke hier nur an die Suche bei *Google*.
- **Optimierende Suchen** finden innerhalb großer Parameterräume sehr effizient Kombinationen dieser Parameter, die eine möglichst niedrige Kostenfunktion ergeben – etwa bei Routenplanern.
- **Stark spezialisierte Suchen** nach Zeichenketten, etwa zum Auffinden ähnlicher Wörter oder Begriffe, sind ebenfalls Beispiele dieser KI-Techniken.

Suchalgorithmen zählt man verallgemeinert zu den Problemlösungsstrategien und Optimierungsverfahren.

Logikbasierte Techniken

Auf Logik basierende Techniken sind alle dadurch gekennzeichnet, dass sie Erfahrungswissen explizit maschinenlesbar erfassen und durch Regeln miteinander in Beziehung setzen:

- Das **regelbasierte Schließen** (Regelmaschinen) greift in der klassischen Form auf Methoden der Aussagenlogik oder Prädikatenlogik zurück. Diese sind in der Lage, die Korrektheit einer neuen Aussage anhand elementarer bekannter Aussagen und Regeln zu deren Verknüpfung entweder zu beweisen oder zu widerlegen.
- **Deduktive Klassifikatoren** leiten beim sogenannten deduktiven Schließen neue Aussagen aus den bisher bekannten ab. Sie können dadurch beispielsweise Begriffe oder Gegebenheiten in Kategorien, Klassen oder Unterklassen einordnen. Aktuell gewinnen derartige Systeme wieder an Bedeutung, denn sie erlau-

ben es, die seit einigen Jahren entstehenden und über das Internet geteilten Wissensdatenbanken von Begriffen und Bedeutungen (Ontologien) als Datengrundlage zu nutzen. Sie können damit zu mehr wissenschaftlicher Erkenntnis beitragen und sogar im Kampf gegen COVID-19 helfen [Smith 2020].

- Sowohl **fallbasiertes Schließen** (fallbezogene Argumentation) als auch **verfahrensorientiertes Schließen** (prozedurale Argumentation) erlauben es, die Lösungen für neue Problemsituationen durch den Vergleich mit bereits bekannten Problemen und den dafür bekannten Lösungsverfahren abzuleiten.

Fuzzy-Logik

Neben der einfachen binären Logik (wahr oder falsch) existieren auch sogenannte **mehrwertige Logiken,** die in Verfahren des logischen Schließens angewandt werden. Diese verwenden für eine Aussage neben den Alternativen *wahr* und *falsch* auch Wahrheitswerte dazwischen. Beispielsweise können damit in einer Nachbildung der Logik digitaler Schaltkreise auch Zustände zwischen den beiden sonst verwendeten Spannungspegeln *High* und *Low* modelliert werden. Gerade in der Schaltungstechnik erweist sich das als sehr hilfreich.

Die sogenannte *Fuzzy-Logik* erlaubt darüber hinaus sogar beliebig viele Wahrheitswerte. Dennoch kann man auf diesen Werten weiterhin logische Operationen wie z.B. ∧ (Und), ∨ (Oder), ¬ (Nicht) anwenden.

Maschinelles Lernen

Die derzeit wohl prominenteste Kategorie von KI-Techniken ist *maschinelles Lernen*. Ab Kapitel 3 widmen wir uns ausgiebig diesem Themenbereich. Im Folgenden veranschaulichen wir gleich an mehreren Stellen das Geschriebene anhand des Praxisbeispiels einer Wetterstation.

Praxisbeispiel 1–2: Wetterstation

Unsere Aufgabe ist es, für eine kleine ortsfeste Wetterstation eine Regenvorhersagefunktion zu programmieren. Unsere Wetterstation liefert täglich Messdaten zu Luftdruck, Temperatur und relativer Luftfeuchtigkeit an ihrem Standort. Aus diesen Messdaten soll unsere Regenvorhersage abgeleitet werden.

Zum maschinellen Lernen zählt eine Vielzahl von Techniken. In Projekten begegnen wir öfters den folgenden:

- **Neuronale Netze** haben aktuell eine große Verbreitung. Daher werden wir in Abschnitt 6.1 noch genauer auf diese eingehen.

- Die **Bayes-Netze** und **Bayes-Klassifikatoren** nutzen mathematische Herangehensweisen, die auf einer statistischen Betrachtung der vorhandenen Trainingsdaten einschließlich der jeweils erwarteten Ergebnisse (als Annotationen[8]) beruhen. Auf Basis der beobachteten Wahrscheinlichkeiten in den Trainingsdaten kann für eine neue Situation auf die (aus Trainingssicht) wahrscheinlichste Ergebnisklasse geschlossen werden. Das Praxisbeispiel 1–3 skizziert die Grundidee dieser KI-Technik für eine Wetterstation.

Praxisbeispiel 1–3: Wetterstation (Erweiterung)

Für die Wetterstation aus Praxisbeispiel 1–2 wollen wir vorhersagen, ob es in den kommenden drei Stunden am Ort regnen wird oder nicht. Dies wollen wir statistisch betrachten: Größen wie Temperatur, Luftdruck und Luftfeuchtigkeit der letzten Stunden an diesem Ort stehen vermutlich damit im Zusammenhang; der Statistiker sagt, sie korrelieren mit dem Ereignis Regen. Je stärker diese Korrelation in der Statistik ausgeprägt ist, desto sicherer ist die Prognose auch für neue Messwerte der Wetterstation. Dabei fließen alle Parameter, hier die Wetterverhältnisse, zugleich in die Bewertung ein.

- **Entscheidungsbäume** nutzt man meist ebenfalls für eine Klassifikation wie etwa im obigen Beispiel zur Wetterprognose. Dabei zieht man in gleicher Weise annotierte Trainingsdaten heran. Der erzeugte Klassifikator berücksichtigt jedoch die einzelnen Einflussparameter Schritt für Schritt in Form von Wenn-dann-Bedingungen. So entsteht eine baumartig verzweigte Kaskade von Entscheidungen, an deren Ende die jeweils als am wahrscheinlichsten ermittelte Klasse benannt wird (siehe Abb. 1–4). Je nachdem, welche Strategie bei der Verzweigung oder Baumtiefe angewendet werden soll, kann man aus einer Reihe von konkreten Algorithmen wählen, z. B. *CART*, *ID3* oder *C4.5* [Knuth 2021].

  Das **Random-Forest**-Verfahren geht noch einen Schritt weiter und nutzt gleich mehrere unabhängig voneinander trainierte Bäume zur Entscheidungsfindung.
- Die **Support Vector Machine** (SVM) ist eine weitere Klassifikationsmethode, die ebenfalls auf bereits annotierten Trainingsdaten beruht. Das Grundprinzip besteht darin, dass der SVM-Klassifikator für z. B. in zwei Klassen eingeteilte Trainingsdaten eine möglichst gute Trennlinie im Parameterraum der zugehörigen Eigenschaften findet. Um eine solche Grenze in Form einer *Hyperebene* (engl. hyperplane) finden zu können, muss man ggf. durch Transformationen künstlich neue Parameter (Dimensionen) erzeugen.

8. Annotationen sind Zusatzinformationen zu (Trainings-)Daten, die mit diesen gespeichert werden, z. B. das erwartete Klassifikationsergebnis.