

Editorial

Liebe Leserinnen und Leser,

herzlich willkommen zur bereits vierten Ausgabe der *ibIS* - Informatische Bildung in Schulen.

In dieser Ausgabe erwarten euch und Sie ein Update zu den **“Bildungsstandards Informatik für die Sekundarstufe 1”**: 17 Jahre nach Veröffentlichung der ersten Fassung dieser Empfehlungen der Gesellschaft der Informatik war eine Überarbeitung und Aktualisierung dringend notwendig, die in mehrjähriger Arbeit durch einen Arbeitskreis angegangen wurde. Im Januar 2025 wurde die neue Fassung nun verabschiedet. Einige Mitglieder des Arbeitskreises stellen das Endprodukt, seine Genese und Leitlinien der Weiterentwicklung in komprimierter Form vor.

Auch finden Sie nach dem Inhaltsverzeichnis die **Termine der durch die GI-Fachgruppen in den Bundesländern organisierten Informatiklehrkräftetage**, die eine spannende Mischung aus hochrangigen Vorträgen zu Themen der Informatik und Informatikdidaktik, praxisorientierten Workshops und viel Möglichkeit zur Vernetzung bieten. Besonders hinweisen möchten wir aber natürlich auch auf die diesjährige **Tagung “Informatik und Schule - INFOS 2025”**, die in der Schweiz unter dem Motto “Grenzen überwinden - voneinander Lernen” stattfinden wird.

Der Kern der Ausgabe sind wie immer die Artikel, die die Vielfalt informatischer Bildung in Schulen widerspiegeln. So schildert Tobias Bahr Implikationen aus zwei wissenschaftlichen **Untersuchungen zum interdisziplinären Informatikunterricht** aus Baden-Württemberg. Reinhild Kokula und andere stellen Fallbeispiele mit Hilfe des “Saftladens” OWASP JuiceShop zur **handlungsorientierten Erarbeitung von Angriffsszenarien in der IT-Sicherheit** vor. Und natürlich geht es auch wieder um Künstliche Intelligenz: Janne Mesenhöller & Katrin Böhme führen als Impuls in **ethische, rechtliche und soziale Implikationen (ELSI) von KI** ein, während Annika Eickhoff-Schachtebeck und Kerstin Strecker verschiedene konkrete **Verfahren des maschinellen Lernens für den Informatikunterricht vergleichen und diskutieren**. Außerdem steht Python in dieser Ausgabe hoch im Kurs: Joachim Kutzera et al. stellen mit der Plattform **CodingLabs ein JupyterLab-basiertes Werkzeug** vor, während Gerhard Röhner mit **GuiPy eine für den Unterricht konzipierte Python-IDE** mit vielfältigen Features für die Modellierung für den Informatikunterricht präsentiert.

Wir freuen uns bereits auf die nächste Ausgabe im Herbst und laden euch und Sie herzlich ein, sich mit Beiträgen zu beteiligen. Gerne stehen wir auch für Fragen zur Beitragseinreichung unter info@informatischebildung.de zur Verfügung.

Viel Freude beim Lesen wünscht das Redaktionsteam der *ibIS* – Informatische Bildung in Schulen

Andreas
Peer
Sandra
Tilman

Impressum

Bezug der Zeitschrift

Die Zeitschrift **ibis** erscheint kostenfrei online unter www.informatischebildung.de.

Alle Beiträge stehen dort und zusätzlich in der *Digitalen Bibliothek* der *Gesellschaft für Informatik* unter dl.gi.de als Open-Educational-Ressources zur Verfügung.

Zusätzlich werden zu verschiedenen Anlässen Printausgaben produziert, z. B. zur 20. GI-Fachtagung „Informatik und Schule“ 2023, die vor Ort an der Veranstaltung bezogen werden können.

Beitragseinreichung

Beiträge zur **ibis** können jederzeit online über die Webseite www.informatischebildung.de eingereicht werden. Dort finden Sie auch den Beitragsaufruf und alle Informationen zur Einreichung.

Für Fragen steht die Redaktion gerne zur Verfügung.

Bibliographische Angaben

ISSN (Print): 2941-7538

ISSN (Online): 2941-7546

DOI-Namensraum: 0.18420/ibis-JG-NR-ART (JG: Jahrgang, NR: Heftnummer, ART: Artikelnummer)

Herausgeber

Gesellschaft für Informatik e. V.
Fachausschuss Informatische Bildung an Schulen
Fachgruppe Didaktik der Informatik

Wissenschaftszentrum, Ahrstraße 45, 53175 Bonn

Redaktion

Dr. Andreas Grillenberger (*verantwortlicher Redakteur*)

Prof. Dr. Tilman Michaeli

Prof. Dr. Sandra Schulz

Dr. Peer Stechert

www.informatischebildung.de

info@informatischebildung.de

Beirat

Alisch, Sven · Anthes, Jaqueline · Arnold, Peter · Barkmin, Mike · Bergner, Nadine · Best, Alexander · Bockelberg, Stefanie · Brinkmeier, Michael · Burk, Steffen · Batur, Fatma · Diethelm, Ira · Dorn, Julian · Gallenbacher, Jens · Geldreich, Katharina · Gramm, Andreas · Grillenberger, Mareen · Hellmig, Lutz · Hempel, Tino · Hennecke, Martin · Hielscher, Michael · Hildebrandt, Claudia · Kastl, Petra · Losch, Daniel · Otto, Torsten · Puhlmann, Hermann · Pöhner, Nicolai · Rau, Thomas · Reher, Jan · Reinold, Klaus · Romeike, Ralf · Rücker, Michael · Schmidt, Pascal · Spalteholz, Wolf · Strecker, Kerstin

Autorinnen und Autorinnen dieser Ausgabe

Bahr, Tobias · Burk, Steffen · Hellmig, Lutz · Kokula, Reinhild · Kufner, Han-Min · Kutzera, Joachim · Leesch, Benedikt · Pasternak, Arno · Reinold, Klaus · Röhner, Gerhard · Scholz, Julian · Schunder, Toni · Wich, Sebastian · Winter, Philipp · Witte, Leonie

Gestaltung und Satz

Dr. Andreas Grillenberger

Urheberrecht / Lizenzen

Die Artikel in dieser Zeitschrift unterstehen dem Urheberrecht und der Verantwortung der jeweiligen Autorinnen und Autoren. Die Redaktion überprüft die Beiträge ebenfalls auf erkennbare (Urheber-)Rechtsverletzungen, die Verantwortung für solche liegt aber weiterhin bei den Autorinnen und Autoren.

Die Autorinnen und Autoren erteilen der **ibis** eine nicht-exklusive Veröffentlichungslizenz. Am Ende jedes Artikels wird die Lizenz genannt, unter der ein Artikel zur Verfügung steht.

Alle von der Redaktion erstellten Teile der Zeitschrift unterstehen der Lizenz CC BY-NC 4.0. Als Attribution wird die Nennung des Namens der Zeitschrift und der Webadresse www.informatischebildung.de gefordert.

Bildquellen

Titelgrafik: Erstellt von Jona Krücken mit DALL-E unter Nutzung des Prompts: *An abstract digital painting representing computer science education. The artwork features vibrant neon colors and geometric shapes forming circuit-lines“*

Grafiken in den Artikeln: Bildquelle gem. Angabe an den Grafiken bzw. Lizenz am Ende des Artikels

Inhalt

Editorial.....	1
Impressum.....	2
Veranstaltungstermine.....	5

Impulse

<i>Burk, S.; Hellmig, L.; Pasternak, A.; Röhner, G.:</i> Neue Zeiten, neue Standards: Neue Empfehlung der Gesellschaft für Informatik für den Informatikunterricht in Sekundarstufe I	7
<i>Mesenhöller, J. & Böhme, K.:</i> Thematisierung ethischer, rechtlicher und sozialer Implikationen (ELSI) Künstlicher Intelligenz im Informatikunterricht.....	11

Aus der Wissenschaft für die Praxis

<i>Bahr, T.:</i> Interdisziplinärer Informatikunterricht: Ergebnisse zweier empirischer Untersuchungen aus Baden-Württemberg	25
<i>Kutzer, J., Witte, L., Schunder, T.:</i> Python-Programmierung im Informatikunterricht mit Jupyter Notebooks	35

Praxisbeiträge

<i>Röhner, G.:</i> GuiPy – eine Python-IDE für die Schule.....	41
<i>Kokula, R., Kufner, H., Leesch, B., Reinold, K., Scholz, J., Wich, S., Winter, P.:</i> So ein Saftladen: IT-Sicherheit in handlungsorientierten Fallbeispielen	51
<i>Eickhoff-Schachtebeck, A., Strecker, K.:</i> Zum Stellenwert von Verfahren des maschinellen Lernens im allgemeinbildenden Informatikunterricht.....	63

Veranstaltungstermine



Fachausschuss Informatische Bildung in Schulen (FA IBS)

22. - 24.09.2025: Informatik und Schule - INFOS 2025 (Stoos Lodge, Stoos, Schweiz)

Didaktik der Informatik (FG DDI)

24. - 26.03.2025: Doktorandenkolloquium (Universität Halle)

21.09.2025: Doktorandenkolloquium (PH Schwyz, Goldau, Schweiz)

Bayerische Informatiklehrkräfte (FG BIL)

09.10.2025: 17. Informatiklehrerinnen und -lehrertag Bayern 2025 - ILTB 2025 (Universität Bayreuth)

Hessische und Rheinland-Pfälzische Informatiklehrkräfte (FG HRPI)

02.06.2025: Rheinland-Pfälzischer Schulinformatiktag (Uni Koblenz)

29.09.2025: Hessischer Schulinformatiktag (tba)

Informatische Bildung in Mecklenburg-Vorpommern (FG IBMV)

05.04.2025: 16. Landestagung der Informatiklehrerinnen und -lehrer (Universität Rostock)

06.10.2025: HILF! 2025 (Universität Rostock)

Informatische Bildung in Niedersachsen und Bremen (FG IBNB)

25.03.2025: till - Tag der Informatiklehrerinnen und -lehrer (Landesinstitut für Schule, Bremen)

Informatische Bildung in NRW (FG IBN)

31.03.2025: 21. Informatiktag NRW (RWTH Aachen University)

Informatische Bildung in Sachsen und Thüringen (FG IBiSaTh)

02.04.2025: Sächsischer Schulinformatiktag (TU Dresden und online)

Schleswig-Holsteiner und Hamburger Informatik-Lehrerinnen und -Lehrer (FG SH-HILL)

15.11.2025: 20. Informatik-Fachtagung / Landesfachtag Informatik (Städtisches Gymnasium Bad Segeberg, Schleswig-Holstein)

Verband zur Förderung des MINT-Unterrichts (MNU)

01. - 04.05.2025: MNU-Bundeskongress (Informatikangebot insbesondere am 03.05.) (Bochum)

Neue Zeiten, neue Standards: Neue Empfehlung der Gesellschaft für Informatik für den Informatikunterricht in Sekundarstufe I

Burk, S.; Hellmig, L.; Pasternak, A.; Röhner, G.

DOI: 10.18420/ibis-03-01-02

Bildungsstandards: GI verabschiedet neue Empfehlung für den Informatikunterricht

Am 31.01.2025 hat das GI-Präsidium den vom Arbeitskreis „Bildungsstandards Sekundarstufe I“ erarbeiteten Entwurf nach kurzer Beratung und Diskussion als **Empfehlung**¹ für den Informatikunterricht verabschiedet. Sie ersetzt die „**Grundsätze und Standards für die Informatik in der Schule**“² vom 24.01.2008. Letztere hatten in der Vergangenheit großen Einfluss auf die Erarbeitung von Curricula in den Bundesländern. Mit den Jahren wuchs der Bedarf für eine Überarbeitung der Standards. Die stürmische Entwicklung der Informatik macht vor der Schule nicht Halt und führt zwangsläufig zu notwendigen Veränderungen und Anpassungen der Curricula. Mit der neuen Empfehlung hat die GI zeitgemäße Bildungsstandards formuliert, an denen sich die Bundesländer bei der Überarbeitung ihrer Curricula orientieren können.

Bildungsstandards³ definieren Kompetenzen, die Schülerinnen und Schüler bis zu einer be-

¹https://informatikstandards.de/fileadmin/FA/IBS/user_upload/Bildungsstandards_Sek_I_2025__pdf-A_.pdf

²https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/bildungsstandards_2008.pdf

³<https://www.iqs.gv.at/themen/nationale-kompetenzerhebung/grundlagen-der-nationalen-kompetenzerhebung/bildungsstandards-und-kompetenzorientierter-unterricht>

stimmten Jahrgangsstufe erreichen sollen. Sie legen fest, was sie in einem Fach wissen und können sollen. Im Unterschied zur Orientierung an Inhalten und Methoden in früher üblichen Lehrplänen sind Bildungsstandards kompetenzorientiert. Sie basieren auf wissenschaftlichen Erkenntnissen und pädagogischen Zielsetzungen und beschreiben Kompetenzen, die Schülerinnen und Schüler in Bezug auf Inhalte und fachliche Methoden entwickeln sollen. Es geht nicht darum, was und wie gelernt werden soll, sondern was die Schülerinnen und Schüler am Ende der Sekundarstufe I können sollen.

Das Kompetenzmodell der Bildungsstandards

Den neuen Bildungsstandards liegt ein Kompetenzmodell zu Grunde, das aus einer Verknüpfung von jeweils fünf Prozess- und Inhaltsbereichen sowie drei Anforderungsbereichen besteht.

In den Prozessbereichen wird beschrieben, auf welche Art und Weise die Schülerinnen und Schüler mit Fachinhalten umgehen sollen. Die Prozessbereiche sind „Modellieren und Implementieren“, „Begründen und Bewerten“, „Strukturieren und Vernetzen“, „Kommunizieren und Kooperieren“ und „Darstellen und Interpretieren“. Für jeden Prozessbereich sind zwischen sieben und neun Kompetenzen formuliert. „Die Schülerinnen und Schüler testen eine Implementierung nach vorgegebenen Kriterien“ ist ein Beispiel dieser Kompetenzen.

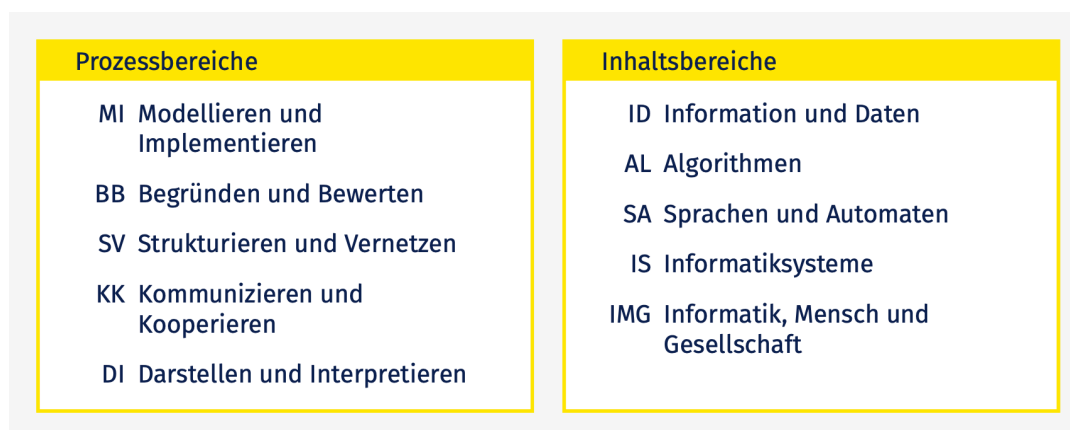


Abbildung 1: Prozess- und Inhaltsbereiche der Bildungsstandards Informatik

Durch die Inhaltsbereiche wird festgelegt, in welchen Gebieten der Informatik die Schülerinnen und Schüler Kompetenzen erwerben und über welches fachliche Wissen und Können sie dazu in diesen Gebieten verfügen sollen. Sie heißen „Information und Daten“, „Algorithmen“, „Sprachen und Automaten“, „Informatiksysteme“ und „Informatik, Mensch und Gesellschaft“. Die Anzahl der Kompetenzen eines Inhaltsbereichs beträgt zwischen vier bei „Sprachen und Automaten“ und zwölf bei „Information und Daten“. Beispielsweise ist „Die Schülerinnen und Schüler beschreiben den Einfluss der verwendeten Daten bei Anwendung eines Verfahrens des maschinellen Lernens“ eine Kompetenz aus dem Inhaltsbereich „Information und Daten“.

Die in den Prozess- und Inhaltsbereichen beschriebenen Kompetenzen werden im Verbund erworben und angewendet. Prozessbezogene Kompetenzen werden in Auseinandersetzung mit informatischen Inhalten erworben und Fachinhalte über prozessbezogene Aktivitäten erschlossen.

Die Anforderungsbereiche beschreiben unterschiedliche kognitive Ansprüche in den drei Stufen „Reproduktion“, „Reorganisation und Transfer“ sowie „Reflexion und Problemlösung“. Sie entsprechen den „[Einheitlichen Prüfungsanforderungen in der Abiturprüfung Informatik](#)“⁴ der Kultusministerkonferenz von 2004. Aufgrund ihrer allgemeinen und jahrgangsunabhängigen Formulierung sind sie auch in der Sekundarstufe I anwendbar. Sie werden im Kompetenzmodell zur Strukturierung der Prozessbereiche benutzt. Die Strukturierung der Inhaltsbereiche erfolgt nach den Jahrgangsstufen 5 bis 6 und 7 bis 10.

Das selbe Kompetenzmodell liegt den [GI-Bildungsstandards für die Sekundarstufe II](#)⁵ von 2016 zugrunde. Die Empfehlungen für den Primarbereich von 2019 beschränken sich auf die Prozess- und Inhaltsbereiche. Durch die Nutzung eines einheitlichen Kompetenzmodells beschreiben die Standards der Gesellschaft für Informatik konsistent die Qualifikationsziele für den allgemeinbildenden Informatikunterricht.

Von PISA zu den Bildungsstandards Informatik

Eine Folge der PISA-Untersuchungen Anfang der 2000er Jahre war die Entwicklung von Bildungsstandards für die Fächer Deutsch, Englisch, Mathematik und später für die Naturwissenschaften. Damit verband sich die Hoffnung auf eine bundesweite Standardisierung und Qualitätsentwicklung in der schulischen Bildung. In Folge dessen wurden die Bildungspläne in den Bundesländern kompetenzorientiert formuliert und jeweils Zentralabiturre eingeführt.

Zugleich legt die Erarbeitung von Bildungsstandards für ausgewählte Fächer den Schluss nahe, die verbleibenden Fächer – wie Informatik – als zweitrangig zu betrachten. Das führte in der fachdidaktischen Gemeinschaft zu Diskussionen und zum Impuls, eigene Standards der GI zu veröffentlichen. Damit wurde der Anspruch des Schulfaches Informatik unterstrichen, unverzichtbarer Teil der Allgemeinbildung zu sein. Diese Diskussion mündete im Jahr 2005 in einer Konstituierung einer Arbeitsgruppe, die daraufhin einen Entwurf für derartige Bildungsstandards formulierte.

Dieser Entwurf wurde in zahlreichen bundesweiten, fachöffentlichen Diskussionen weiter ausgearbeitet und schlussendlich 2008 von der GI unter dem Titel „Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I“ verabschiedet.

Wenig Erfahrung mit Informatikunterricht in der Sekundarstufe I und unterschiedliche Vorstellungen zu den Zielen eines allgemeinbildenden Informatikunterrichts bewogen die Autorinnen und Autoren seinerzeit, die Standards mit einer Darstellung zu „Grundsätzen eines guten Informatikunterrichts“ einzuleiten. Hervorgehoben wurde, dass aus Gründen der Chancengleichheit ein qualifizierter Informatikunterricht für alle Schülerinnen und Schüler unabdingbar ist. Dies wurde durch die Formulierung von Mindeststandards anstelle von Regelstandards unterstrichen.

Die Situation hat sich in den letzten zwei Jahrzehnten geändert: Zwischenzeitlich ist Informatik in der Sekundarstufe I zumindest in einigen Bundesländern Pflichtfach geworden. Damit erübrigt sich die Notwendigkeit, dieses Fach vor unterschiedlichen Adressaten zu rechtfertigen.

⁴ https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01_EPA_Informatik.pdf

⁵ https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/Bildungsstandards_SII.pdf

Neue Zeiten, neue Standards.

Für die Neuauflage der Bildungsstandards Informatik für die Sekundarstufe I durch die Gesellschaft für Informatik im Jahr 2025 spielten mehrere Überlegungen eine Rolle.

Als wohl weitreichendste Wirkung der Bildungsstandards Informatik hat sich bislang der Einfluss auf die Arbeit der Bildungsplankommissionen in den Bundesländern erwiesen. Die Neuauflage der Bildungsstandards trägt diesem Umstand Rechnung und adressiert darum ausdrücklich Bildungsplankommissionen als primäre Zielgruppe.

Anders als noch 2008 bedarf es keiner Rechtfertigung eines Pflichtfachs Informatik für alle Schülerinnen und Schüler. Die Empfehlung der Ständigen Wissenschaftlichen Kommission der KMK aus dem Jahr 2022, das Fach Informatik in der Stundentafel der Sekundarstufe I mit mindestens vier und mittelfristig sechs Unterrichtsstunden zu verankern, unterstreicht dies eindrücklich.

Mit zunehmenden Erfahrungen im Informatikunterricht der Sekundarstufe I konnte die Angemessenheit der 2008 oftmals sehr ambitionierten, als Mindeststandards deklarierten Kompetenzbeschreibungen kritisch beurteilt und in Teilen relativiert werden, besonders für Schüle-

rinnen und Schüler, die ihren Schulabschluss am Ende der Sekundarstufe I anstreben. Der Arbeitskreis konnte dankbar auf eine Vielzahl kritisch-konstruktiver Hinweise von Expertinnen und Experten aus Schulpraxis und Wissenschaft zurückgreifen, die in mehreren Workshops auf der INFOS 2023, den Fachdidaktischen Gesprächen zur Schulinformatik in Königstein zwischen 2021 und 2024 sowie über eine Online-Umfrage erhoben wurden.

Großen Raum nahm die Prüfung der Standards auf Aktualität und Vollständigkeit unter Berücksichtigung des Bildungswertes für Schülerinnen und Schüler ein. Wichtige und aktuelle Themen wie Internet, Kryptologie oder Künstliche Intelligenz seien exemplarisch als neue bzw. ausgebauten Aspekte der Bildungsstandards genannt. Die gesellschaftliche Bedeutung informatischer Bildung wurde durch eine differenzierte Beschreibung von Kompetenzen im Inhaltsbereich „Informatik, Mensch und Gesellschaft“ und die Einbeziehung von Aspekten der Bildung für nachhaltige Entwicklung (BNE) berücksichtigt.

Wir gehen davon aus, dass die neuen Bildungsstandards für die Sekundarstufe I einen ebenso positiven Einfluss wie die bisherigen Empfehlungen auf die Lehrpläne für Informatik in den Bundesländern haben und den Unterricht im Schulfach Informatik in den nächsten Jahrzehnten prägen werden.



Abbildung 2: Titel der Bildungsstandards Informatik für die Sekundarstufe I

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Steffen Burk
Studienseminar Darmstadt
burk.steffen@sts-gym-da.eu

Lutz Hellmig
Universität Rostock
lutz.hellmig@uni-rostock.de

Arno Pasternak
TU Dortmund
arno.pasternak@cs.tu-dortmund.de

Gerhard Röhner
Dieburg
groehner@t-online.de

Thematisierung ethischer, rechtlicher und sozialer Implikationen (ELSI) Künstlicher Intelligenz im Informatikunterricht

Mesenhöller, J. & Böhme, K.

DOI: 10.18420/ibis-03-01-03

Zusammenfassung

Seit dem Launch von ChatGPT-3.5 im November 2022 hat die Nutzung Künstlicher Intelligenz – sowohl gesamtgesellschaftlich als auch im Bildungskontext – stark zugenommen. Neben technologischen Aspekten rücken daher vermehrt auch ethische, rechtliche und soziale Implikationen (ELSI) dieser neuartigen Interaktion zwischen Mensch und Maschine in den Fokus. Der Informatikunterricht bietet verschiedene Möglichkeiten, diese zu thematisieren. Der vorliegende Impuls umfasst daher Gestaltungsvorschläge für eine sechsstündige Unterrichtssequenz (drei Doppelstunden) zu diesem Thema und richtet sich schulformübergreifend an (Informatik-)Lehrkräfte der Jahrgangsstufen 7 bis 10. Er zeigt auf Basis multimedialer Lernmaterialien auf, wie die Auseinandersetzung mit ethischen Herausforderungen spezifischer KI-Anwendungen als Ausgangspunkt für die Thematisierung dahinterliegender maschineller Lernverfahren genutzt werden kann.

Einleitung

Künstliche Intelligenz (KI) ist mittlerweile allgegenwärtig und somit auch im Schulkontext präsent. Derzeit geben 74% der 14- bis 20-Jährigen an, KI-Systeme zu nutzen, zum Teil sogar täglich (Vodafone Stiftung Deutschland, 2024, S. 10f.). Fragt man die Jugendlichen, welche KI-Systeme das hauptsächlich sind, so gibt der Großteil Large-Language-Modelle (LLM) wie ChatGPT an. Die Lernenden äußern die Erwartung, dass der Einsatz von KI den Unterricht in den kommenden fünf Jahren stark verändern wird, auch gaben 86% der fast 1600 Befragten darüber hinaus an, dass die Nutzung von KI in ein paar Jahren so selbstverständlich sein wird, wie die Nutzung des Smartphones (Vodafone Stiftung Deutschland, 2024, S. 7f.). Die schulbezogene Nutzung von LLM erfolgt derzeit jedoch vor allem zu Recherchezwecken und zum Großteil ohne Begleitung oder Unterstützung von Lehrkräften (Vodafone Stiftung Deutschland, 2024). In Anbetracht möglicher *Halluzinationen* von LLM – also Outputs, in denen überzeugend klingende, aber inhaltlich ungenaue oder völlig erfundene Informationen generiert werden (Ye et

al., 2023) – ist diese Kombination als potentiell problematisch zu bewerten.

Durch die Nutzung von KI in Schule und Unterricht rückt daher neben den technologischen und anwendungsbezogenen Aspekten auch die *gesellschaftlich-kulturelle Perspektive* und die Frage nach möglichen *Auswirkungen* dieser Nutzung (Brinda et al., 2016) in den Mittelpunkt. Sowohl gesamtgesellschaftlich als auch speziell im Schulkontext gewinnen die sogenannten ethischen, rechtlichen und sozialen Implikationen (*Ethical, Legal and Social Implications*, ELSI, Boden et al., 2021) dementsprechend zunehmend an Bedeutung. Diese sollten auch aktiv im Unterricht behandelt werden, um Schüler:innen ein Verständnis für die Potenziale und Risiken der Nutzung von KI-Anwendungen zu vermitteln. Das 2021 vorgeschlagene und 2024 verabschiedete KI-Gesetz der EU-Staaten zur Regulierung von KI, in dem auch Regularien zur Nutzung von KI im Schulkontext enthalten sind, unterstreicht die aktuelle Bedeutsamkeit dieser ethischen und rechtlichen Fragestellungen (*„Einigung Der EU-Staaten: KI-Gesetz Ist Endgültig Beschlossen,“* 2024; Europäisches Parlament, 2023).

Obwohl es Aufgabe aller Fächer ist, Schüler:innen in der Ausbildung ihrer digitalen Kompetenzen zu unterstützen (Brinda et al., 2016; Sekretariat der Ständigen Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland [KMK], 2016) bietet jedes Fach spezifische Möglichkeiten zum Ausbau dieser Kompetenzen (KMK, 2021). Dies gilt insbesondere für den Informatikunterricht, der als eigenständiger Lernbereich vertieft das technologische Hintergrundwissen liefern kann, um zentrale ELSI im Kontext zunehmender KI-Nutzung näher zu beleuchten.

Der vorliegende Impuls zeigt auf, wie ELSI in den Informatikunterricht eingebunden werden können. Da das Fach Informatik in den meisten Bundesländern noch immer kein Pflicht-, sondern ein Wahlfach ist (Gesellschaft für Informatik e.V., 2024; Stifterverband, 2023), können aber auch Lehrkräfte anderer Fächer wie bspw. Ethik, Philosophie, Politische Bildung oder Gemeinschaftskunde die hier skizzierten Impulse und Materialien für ihren Unterricht nutzen. In diesen Fächern könnte der Fokus dann weniger auf

der technologischen sondern stärker auf der gesellschaftlich-kulturellen Perspektive konkreter KI-Anwendungen liegen.

ELSI im Kontext von KI

ELSI befasst sich mit potenziellen ethischen, rechtlichen und sozialen Herausforderungen, die neuartige wissenschaftliche und technologische Entwicklungen sowohl auf gesellschaftlicher als auch individueller Ebene mit sich bringen können (Juengst, 1991).

Das Konzept ELSI stammt ursprünglich aus der medizinischen Forschung und erlangte während der Entschlüsselung der menschlichen DNA an Bedeutung. Im Fokus standen dabei Themen wie Datenschutz, Urheberrecht und informierte Einwilligung.

Im Kontext von KI umfasst ELSI Herausforderungen wie die Frage nach ausreichend Datenschutz und Datensicherheit (Mantelero, 2018; Stahl & Wright, 2018) oder Unklarheiten bezüglich des Urheberrechts von KI-generierten Texten sowie damit möglicherweise einhergehende Plagiate (Ständige Wissenschaftliche Kommission der Kultusministerkonferenz [SWK], 2024). Auch ethische Aspekte wie Neutralität und Fairness von KI-Systemen fallen unter ELSI. Ein sehr bekanntes Beispiel ist hier das Phänomen der *algorithmischen Diskriminierung*, bei dem KI-Anwendungen Vorurteile und Ungleichheiten aus den zugrunde liegenden Trainingsdaten übernehmen und dadurch in der Gesellschaft vorherrschende Diskriminierungsstrukturen reproduzieren oder sogar verstärken (Akter et al., 2021; Buolamwini, 2017; Raji et al., 2020).

Informatikunterricht als geeigneter Ort für die Auseinandersetzung mit ELSI

Die Thematisierung von ELSI im Informatikunterricht bietet die Möglichkeit, Schüler:innen einen Zugang zu den gesellschaftlichen und individuellen Auswirkungen von KI-Technologien zu eröffnen. Lehrkräfte haben an dieser Stelle die Möglichkeit, den Schüler:innen zunächst die erforderlichen technologischen Grundlagen

von KI zu vermitteln und anschließend die gesellschaftlich-kulturellen Aspekte spezifischer KI-Anwendungen zu adressieren. Ein alternativer didaktischer Zugang ist, durch die Betrachtung der gesellschaftlich-kulturellen Perspektive einer spezifischen KI-Anwendung das Interesse und die Motivation der Schüler:innen für dahinterliegende technische Prozesse zu wecken. So kann ein tieferes technisches Verständnis der Schüler:innen gefördert werden. Im Folgenden wird ein Ansatz vorgestellt, der die zweite Herangehensweise wählt.

Am Beispiel der ethischen Herausforderungen von Gesichtserkennungssoftware wird in sechs Unterrichtsstunden für Schüler:innen der Jahrgangsstufen 7 bis 10 das übergeordnete Kompetenzziel *Wechselwirkungen zwischen Informatiksystemen, Mensch und Gesellschaft beurteilen – Anwendungen erfassen und Auswirkungen abschätzen* adressiert (vgl. bspw. Berliner Senatsverwaltung für Bildung, Jugend und Familie [SenBJF] & Ministerium für Bildung, Jugend und Sport des Landes Brandenburg [MBJS], 2015)

Orientiert am Rahmenlehrplan Informatik Berlin-Brandenburg (SenBJF & MBJS, 2015, S. 15) sowie Mecklenburg-Vorpommern (Apsel & Brandt, 2019, S. 11) wird konkret auf die folgenden Lernziele hingearbeitet: Schüler:innen können

- aktuelle Entwicklungen bewerten, aus ihnen mögliche Trends ableiten und Auswirkungen auf die Zukunft beschreiben
- Chancen und Risiken moderner technologischer Entwicklungen für eine demokratische Gesellschaft bewerten
- angemessen auf Risiken bei der Nutzung von Informatiksystemen reagieren.

Wie Lehrkräfte Schüler:innen dabei unterstützen können, diese Lernziele in Bezug auf KI-bezogene ELSI im Informatikunterricht zu erreichen, wird nachfolgend beispielhaft dargelegt. Dazu wird die folgende Struktur vorgeschlagen:

Unterrichtssequenz: ELSI als Ausgangspunkt für die Beschäftigung mit der technologischen Perspektive von maschinellen Lernverfahren

Unterrichts- stunde	Thema/Schwer- punkt	Zielsetzung	Materialien	Übergeordnete Lernziele
1 (optional)*	Einführung: Was ist Künstliche Intelligenz?	Die Schüler:innen entwickeln ein grundlegendes Verständnis von KI, lernen die Unterschiede zwischen starker und schwacher KI kennen und können die Funktionsweise von KI-Systemen grob skizzieren.	<ul style="list-style-type: none"> Text „Was ist KI?“ vom KI-Campus Digitales Tool (z. B. Mentimeter) oder analoges Medium (z. B. Tafel, Plakat) 	Schüler:innen können <ul style="list-style-type: none"> aktuelle Entwicklungen bewerten, aus ihnen mögliche Trends ableiten und Auswirkungen auf die Zukunft beschreiben. Chancen und Risiken moderner technologischer Entwicklungen für eine demokratische Gesellschaft bewerten. angemessen auf Risiken bei der Nutzung von Informationssystemen reagieren.
2	Algorithmische Diskriminierung in Gesichtserkennungssoftware als Beispiel für ethische Herausforderungen der KI-Nutzung	Die Schüler:innen lernen anhand eines konkreten Beispiels die Problematik algorithmischer Diskriminierung kennen, verstehen deren Ursachen und diskutieren erste Ansätze, wie solche Probleme vermieden werden könnten.	<ul style="list-style-type: none"> Text „Coded Bias – Können Maschinen diskriminieren?“ vom KI-Campus Digitales Tool (z. B. Mentimeter) oder analoges Medium (z. B. Tafel, Plakat) 	
3	Trainieren eines KI-Modells auf Basis der Teachable Machine	Die Schüler:innen erhalten einen praktischen Einblick in maschinelles Lernen, erleben die Schritte des Modelltrainings und testen die Funktionalität selbst trainierter Modelle. Sie reflektieren die Herausforderungen und potenziellen Verzerrungen maschineller Klassifikationssysteme.	<ul style="list-style-type: none"> Teachable Machine (browserbasiert, Endgeräte mit Kamera, allerdings keine Tablets!) Digitales Tool (z. B. Mentimeter) oder analoges Medium (z. B. Tafel, Plakat) 	
4	Maschinelle Entscheidungsverfahren	Die Schüler:innen erarbeiten vertieft, wie maschinelle Lernsysteme Entscheidungen treffen, indem sie ein Entscheidungsbaummodell erstellen.	<ul style="list-style-type: none"> Lernmaterial zum „Gute-Äffchen-Böse-Äffchen-Spiel“ von AI Unplugged Bilder der Affen, Beschreibung der Aufgabe (digital oder analog) 	
5	Erarbeitung und Präsentation der Arten maschinellen Lernens	Die Schüler:innen erarbeiten in Kleingruppen eine der drei Hauptarten des maschinellen Lernens (überwachtes, unüberwachtes, verstärkendes Lernen). Anschließend präsentiert pro Lernverfahren eine Gruppe ihre Arbeitsergebnisse und die Mitschüler:innen ergänzen.	<ul style="list-style-type: none"> Karten oder Tafel für Gruppenfindung und Leitfragen Webseite Computing Education – Maschinelles Lernen Endgeräte mit Internetzugang (Tablets, Laptops oder Smartphones) Präsentationsmaterial (Papier und Flipchartmarker oder digitales Tool wie PowerPoint) 	

Unterrichtsstunde	Thema/Schwerpunkt	Zielsetzung	Materialien	Übergeordnete Lernziele
6	Zusammenfassung und Rückbezug zu ELSI	Die Schüler:innen fassen die wichtigsten Inhalte der Unterrichtssequenz zusammen, reflektieren ethische, rechtliche und soziale Herausforderungen im Kontext von KI und formulieren Handlungsoptionen für einen verantwortungsvollen Umgang mit KI-Systemen.	<ul style="list-style-type: none"> • Diskussionsleitfragen (digital oder analog) • ggf. Audio-Interview KI-Campus: Interview zu algorithmischer Diskriminierung • ggf. Technisches Endgerät mit Internetzugang (Tablet, Laptop) sowie Lautsprecher, um Audio-Interview abzuspielen • Materialien zur Ergebnissicherung (z. B. digitales Padlet) 	

* Unterrichtsstunde 1 kann optional geplant werden, falls noch kein Basiswissen zum Thema KI vorhanden ist. Wird die Stunde nicht benötigt, können bspw. die maschinellen Lernverfahren in Stunde 5 vertiefend behandelt und ggf. um Deep Learning erweitert werden.

Unterrichtsstunde 1: Einführung: Was ist Künstliche Intelligenz? (optional)

Auch wenn die Beschäftigung mit ELSI als Ausgangspunkt dient, um dahinterliegende maschinelle Lernverfahren tiefergehend zu thematisieren, müssen die Schüler:innen dennoch ein grundlegendes Verständnis des Konzeptes KI haben, um die Frage, welche Auswirkungen die Nutzung von KI auf unsere Gesellschaft haben kann (Brinda et al., 2016), beantworten und ELSI der KI-Nutzung verstehen zu können. Ist dieses Basiswissen noch nicht vorhanden, so bietet sich bspw. der Text „Was ist Künstliche Intelligenz?“ vom KI-Campus als thematische Einführung an (KI-Campus, 2024). Der Text ist für die 7. bis 10. Jahrgangsstufe geeignet und Schüler:innen benötigen keine informatischen Vorkenntnisse, um ihn verstehen zu können. In dem Lernmaterial wird der Begriff KI grundlegend erklärt, starke von schwacher KI abgegrenzt sowie der Themenkomplex der maschinellen Entscheidungssysteme angeschnitten. Nach dieser Unterrichtsstunde sollen die Lernenden in der Lage sein, „grundlegend zu beschreiben, wie ein algorithmenbasiertes KI-System arbeitet/lernt“ (KI-Campus, 2024). Hierfür kann folgende Strukturierung genutzt werden:

Zielsetzung:

Die Schüler:innen entwickeln ein grundlegendes Verständnis von KI, lernen die Unterschiede zwischen starker und schwacher KI kennen und

können die Funktionsweise von KI-Systemen grob skizzieren.

1. Einstieg: Brainstorming „Was ist Künstliche Intelligenz?“ (~ 5 Minuten)

- Die Lehrkraft startet mit der Frage: „Was fällt euch ein, wenn ihr den Begriff ‚Künstliche Intelligenz‘ hört?“
- Die Schüler:innen äußern ihre Ideen, die von der Lehrkraft mittels eines digitalen Tools (z. B. Mentimeter) oder analog an der Tafel/auf einem Plakat gesammelt werden. Falls nötig, ergänzt die Lehrkraft Beispiele wie Gesichtserkennung, Chatbots oder autonome Fahrzeuge. Ziel ist die Aktivierung von Vorwissen und Interesse für das Thema.

2. Erarbeitung: Einzelarbeit mit dem Text „Was ist Künstliche Intelligenz?“ (~ 25 Minuten)

- Die Schüler:innen lesen den Text „Was ist Künstliche Intelligenz?“ vom KI-Campus in Einzelarbeit. Sie bearbeiten dabei folgende Leitfragen kurz schriftlich:
 - Was ist der Unterschied zwischen schwacher und starker KI?
 - Wie „lernt“ ein KI-System?
 - Wo wird KI im Alltag bereits eingesetzt?
- Die Lehrkraft kann während der Bearbeitung unterstützen, indem sie Verständnisfragen klärt.

3. Ergebnissicherung und Diskussion (~ 10 Minuten)

- Die Antworten der Schüler:innen auf die Leitfragen werden im Plenum besprochen. Dabei werden die Konzepte schwache und starke KI sowie grundlegende Funktionsweisen wie das Black-Box-Prinzip thematisiert.
- Als Übergang zu ethischen Fragestellungen wird eine Diskussion angestoßen durch die Frage: „Welche Verantwortung hat der Mensch in der Entwicklung und im Umgang mit KI-Systemen?“

4. Abschluss (~ 5 Minuten)

- Die Lehrkraft fasst die zentralen Erkenntnisse der Stunde zusammen und gibt einen kurzen Ausblick auf die nächste Stunde, in der ein Beispiel für die praktischen Auswirkungen von KI auf die Gesellschaft (algorithmische Diskriminierung) behandelt wird.

Materialien

- Text „Was ist Künstliche Intelligenz?“
- Digitales Tool (z. B. Mentimeter) oder analoges Medium (z. B. Tafel, Plakat)

Unterrichtsstunde 2: Algorithmische Diskriminierung in Gesichtserkennungssoftware als Beispiel für ethische Herausforderungen der KI-Nutzung

In der zweiten Unterrichtsstunde wird mit Hilfe des Textes „Coded Bias - Können Maschinen diskriminieren?“ des KI-Campus (KI-Campus, 2024) eine zentrale ethische Herausforderung des KI-

Einsatzes anhand eines konkreten Beispiels thematisiert. Das Lernmaterial wurde für Schüler:innen der 7. bis 10. Jahrgangsstufe entworfen und behandelt das Problem der algorithmischen Diskriminierung.

In dem Material des KI-Campus wird die Forschung der schwarzen Informatikerin Joy Buolamwini thematisiert. Diese entdeckte, dass ihr Gesicht von einer Gesichtserkennungssoftware nur dann erkannt wird, wenn sie eine weiße Maske trägt. Buolamwini fand heraus, dass die Trainingsdaten der verwendeten Gesichtserkennungssoftware überwiegend Bilder von weißen Männern enthielten, während Bilder von Frauen, insbesondere schwarzen Frauen, stark unterrepräsentiert waren (Buolamwini, 2017; Raji et al., 2020). Als Konsequenz war das System nicht in der Lage, die Gesichter schwarzer Frauen korrekt zu erkennen. Die Unterrichtsstunde kann wie folgt strukturiert werden:

Zielsetzung:

Die Schüler:innen lernen anhand eines konkreten Beispiels die Problematik algorithmischer Diskriminierung kennen, verstehen deren Ursachen und diskutieren erste Ansätze, wie solche Probleme vermieden werden könnten.

1. Einstieg: Bezug zur Lebenswelt (~ 10 Minuten)

- Die Lehrkraft eröffnet die Stunde mit der Frage: „Wer von euch nutzt die Gesichtserkennung auf dem Smartphone, um es zu entsperren?“
- Im Anschluss wird gefragt: „Habt ihr euch schon einmal gefragt, ob diese Technik für alle Menschen gleichermaßen gut funktioniert?“

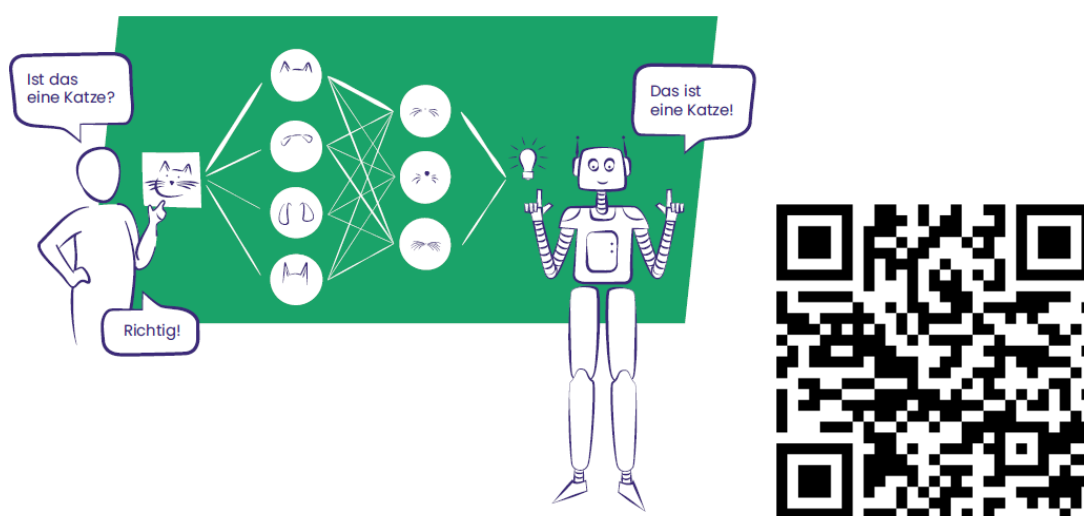


Abbildung 1: Text „Was ist KI?“ vom KI-Campus sowie weitere Hinweise zum Einsatz im Unterricht (QR-Code) (KI-Campus/CC BY-SA 4.0)

- Die Antworten der Schüler:innen werden gesammelt, entweder digital (z. B. in einem Padlet), analog an der Tafel oder auf einem Poster. Ziel ist es, die Schüler:innen für die Alltagsrelevanz des Themas zu sensibilisieren und mögliche Vorurteile oder Ungleichheiten zu hinterfragen.

2. Erarbeitung: Einzelarbeit mit dem Text „Coded Bias - Können Maschinen diskriminieren?“ (~ 20 Minuten)

- Die Schüler:innen lesen in Einzelarbeit das Lernmaterial „Coded Bias – Können Maschinen diskriminieren?“. Dabei sollen sie folgende Leitfragen beantworten, um den Text gezielt zu bearbeiten:
 - Warum hat die Gesichtserkennungssoftware bei Joy Buolamwini versagt?“
 - „Was hat das mit den Trainingsdaten des Systems zu tun?“
 - „Welche Auswirkungen könnten solche Probleme in anderen Bereichen haben?“

3. Austausch und Diskussion: Reflexion im Plenum (~ 10 Minuten)

- Die Klasse diskutiert gemeinsam die Frage: „Können Maschinen diskriminieren?“
- Zu Beginn der Diskussion wird auf die gesammelten Antworten aus dem Einstieg zurückgegriffen. Die Lehrkraft moderiert die Diskussion und lenkt die Aufmerksamkeit darauf, dass Maschinen nicht absichtlich diskriminieren, sondern dass dies auf die zugrunde liegenden Trainingsdaten und Algorithmen zurückzuführen ist.
- Die Lehrkraft stellt weitere Impulsfragen:

- „Welche Verantwortung tragen Entwickler:innen solcher Systeme?“
- „Wie könnte man verhindern, dass solche Diskriminierungen auftreten?“

4. Abschluss und Ausblick (~ 5 Minuten)

- Die Stunde endet mit einem kurzen Fazit, das von den Schüler:innen zusammengetragen wird. Dabei weist die Lehrkraft darauf hin, dass solche ethischen Herausforderungen auch bei anderen KI-Anwendungen auftreten können.
- Die Stunde wird mit einem Ausblick auf die nächste Unterrichtsstunde (praktisches Arbeiten mit der Teachable Machine) beendet.

Materialien:

- Lernmaterial „Coded Bias – Können Maschinen diskriminieren?“
- Digitales Tool (z. B. Padlet) oder analoges Medium (z. B. Tafel/Plakat) zur Sammlung von Ideen und Diskussionsergebnissen

Unterrichtsstunde 3: Trainieren eines KI-Modells auf Basis der Teachable Machine

Ein praxisnaher Zugang zu der Funktionsweise maschineller Lernverfahren kann durch die Nutzung der Teachable Machine von Google (Google, 2024a) erfolgen. Diese ermöglicht es Schüler:innen, selbst einfache KI-Modelle zu erstellen und zu trainieren.

Die Teachable Machine knüpft als schwache KI und sehr einfach zu programmierendes maschinelles Entscheidungssystem an den Text des KI-Campus „Was ist Künstliche Intelligenz?“ an. Sie ist ein browserbasiertes KI-Tool, das den Trainingsprozess des maschinellen Lernens für Lai-

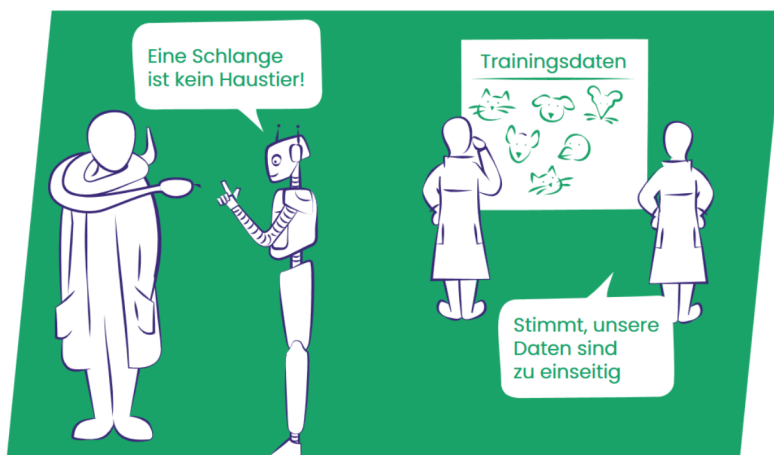


Abbildung 2: Text „Coded Bias – Können Maschinen diskriminieren?“ vom KI-Campus sowie weitere Hinweise zum Einsatz im Unterricht (QR-Code) (KI-Campus/CC BY-SA 4.0)

Abbildung 3: Teachable Machine und ausführliche Beschreibung und Demonstration der Teachable Machine (QR-Code)

en erfahrbar macht. Nutzer:innen können im Browser ein Modell mithilfe eigener Daten trainieren, welches Bilder, Audios oder Bewegungen kategorisiert (Google, 2024a). Da das zugrundeliegende neuronale Netz bereits vortrainiert ist, ist der Trainingsprozess mit wenig Daten und geringem Rechenaufwand umsetzbar (Google, 2024b). Eine umfassende Demonstration der Teachable Machine finden Sie hier:

- [Teachable Machine](#)
- [Ausführliche Beschreibung und Demonstration der Teachable Machine](#)

Nachfolgend wird erläutert, wie die Teachable Machine im Unterricht genutzt werden kann, um die hinter KI-Systemen (wie bspw. Gesichtserkennungssoftware) liegenden maschinellen Lernverfahren besser verstehen zu können.

Zielsetzung:

Die Schüler:innen erhalten einen praktischen Einblick in maschinelles Lernen, erleben die Schritte des Modelltrainings und testen die Funktionalität selbst trainierter Modelle. Sie reflektieren die Herausforderungen und potenziellen Verzerrungen maschineller Klassifikationssysteme.

1. Einführung in die Teachable Machine und Ablauf der Stunde (~ 5 Minuten)

- Die Lehrkraft erklärt die Funktion der Teachable Machine als einfaches Werkzeug für maschinelles Lernen. Sie erläutert, wie Daten genutzt werden, um ein Modell zu trainieren, und skizziert den Ablauf der Stunde. Ziel ist es, einen Überblick über die bevorstehende praktische Arbeit zu geben und die Motivation zu fördern.

2. Auswahl und Aufnahme der zu klassifizierenden Objekte (~ 10 Minuten)

- Die Schüler:innen arbeiten in Kleingruppen und wählen Objekte oder Gesten, die sie klassifizieren möchten. Es bietet sich an, dass die unterschiedlichen Kleingruppen verschieden komplexe Klassifikationen wählen:
 - Einfache Klassifikationen:
 - Objekte wie Federmäppchen und Geodreiecke oder Handgesten wie Fingerzeigen und High Five unterscheiden.
 - Komplexere Klassifikationen:
 - Zwei Klassen mit mehreren Unterobjekten (z. B. Tier vs. kein Tier, Obst vs. Gemüse).
 - Oberklassen mit Unterklassen (z. B. Emotionen anhand von Bildern, Musikstile anhand von Geräuschen, Sportarten anhand von Körperposen).

3. Trainieren des Modells (~ 5 Minuten)

- Die Schüler:innen trainieren das Modell mithilfe der aufgenommenen Daten. Dabei analysiert die Teachable Machine die Bilder, Posen oder Audiospuren und generiert ein Klassifikationsmodell.

4. Testen des Modells (~ 10 Minuten)

- Die Schüler:innen testen ihr Modell mit den bereits aufgenommenen und neuen Daten. Sie notieren die Ergebnisse und vergleichen, welche Klassifikationen korrekt waren und wo Fehler auftraten. Dies fördert ein erstes Verständnis für die Funktionsweise und Grenzen solcher Systeme.

5. Reflexion und Rückmeldung (~ 15 Minuten)

- Im Plenum diskutieren die Schüler:innen ihre Ergebnisse:
 - Welche Klassifikationen funktionierten gut, und wo gab es Fehlklassifikationen?
 - Was könnten die Gründe für die Fehler sein (z. B. unzureichende Daten, Verzerrungen)?
- Die Schüler:innen formulieren Hypothesen, die die Lehrkraft auf einem Whiteboard oder einer Tafel sammelt. Ziel ist es, ein Bewusstsein für die Herausforderungen des maschinellen Lernens zu entwickeln und die Erkenntnisse auf andere KI-Systeme zu übertragen.

Materialien:

- **Teachable Machine** (browserbasiert, Geräte mit Kameras erforderlich, kein Tablet!)
- Whiteboard oder digitale Alternative (für Hypothesen und Diskussionsergebnisse)

Zum Abschluss der Unterrichtsstunde bietet es sich an, den Übergang zur *Unterrichtsstunde 4: Verschiedene Arten maschineller Lernverfahren* herzustellen. Die Lehrkraft kann bspw. darauf hinweisen, dass die Schüler:innen in dieser Stunde durch ein praktisches Beispiel (das „Gute-Äffchen-Böse-Äffchen-Spiel“) die Entscheidungsfindung bei der Klassifikation von maschinellen Lernverfahren spielerisch nachvollziehen werden.

Unterrichtsstunde 4: Maschinelle Entscheidungsverfahren

Zielsetzung der vierten Unterrichtsstunde ist es, dass die Schüler:innen vertieft erarbeiten, wie maschinelle Lernsysteme Entscheidungen treffen, indem sie ein Entscheidungsbaummodell erstellen.

Zum Einstieg in maschinelle Entscheidungsverfahren bietet es sich an, die Schüler:innen eine analoge Übung durchführen zu lassen. Eine besonders anschauliche Methode ist das „Gute-Äffchen-Böse-Äffchen-Spiel“ von Lindner & Seegerer (o.D.). In diesem Szenario schlüpfen die Schüler:innen in die Rolle von Unterstützer:innen einer Tierpflegerin, die mithilfe von Bildern und spezifischen Kriterien vorhersagen möchte, welche Affen potenziell beißen könnten. Ziel ist es, auf Basis der gegebenen Daten einen Entscheidungsbaum zu erstellen, der die Arbeitsweise eines KI-Systems modellhaft darstellt.

Während der Bearbeitung analysieren die Schüler:innen die Bilder der Affen und überlegen sich Kriterien wie Haltung des Mundes oder der Augen, die für eine Vorhersage herangezogen werden könnten. Diese Kriterien ordnen sie schrittweise an, um so Entscheidungswege zu erarbeiten, die letztlich in einen Entscheidungsbaum münden. Diese Methode veranschaulicht auf einfache und intuitive Weise, wie KI-Systeme strukturierte Daten verarbeiten und auf Grundlage von Trainingsdaten Entscheidungen treffen.

Das Szenario bietet den Vorteil, dass es den Schüler:innen eine konkrete Vorstellung davon vermittelt, wie maschinelle Lernprozesse funktionieren, ohne dass technische Vorkenntnisse erforderlich sind. Daher eignet sich das Spiel auch für den Einsatz in anderen Fächern. Für die Unterrichtsstunde kann die folgende Strukturierung gewählt werden:

1. Einstieg: Einführung in maschinelle Entscheidungsverfahren (~ 10 Minuten)

- Die Lehrkraft stellt das „Gute-Äffchen-Böse-Äffchen-Spiels“ kurz vor und zeigt, wie die Schüler:innen mithilfe von Kriterien (z. B. Fellfarbe, Größe) auf Basis eines Entscheidungsbaums Entscheidungen treffen können, was der Verarbeitung von Daten von KI-Systemen ähnelt.

2. Arbeitsphase: Durchführung des „Gute-Äffchen-Böse-Äffchen-Spiels“ (~ 25 Minuten)

- Die Schüler:innen arbeiten in Kleingruppen. Jede Gruppe erhält vorbereitete Materialien (Bilder der Affen, Beschreibung der Aufgabe).
- Die Gruppen analysieren die Bilder und erarbeiten Kriterien, die für eine Vorhersage relevant sein könnten. Sie erstellen schrittweise einen Entscheidungsbaum.
- Die Lehrkraft unterstützt bei Fragen.

3. Reflexionsphase: Präsentation der Entscheidungsbäume und Gruppenfindung (~ 10 Minuten)

- Präsentation: Jede Gruppe stellt kurz ihren Entscheidungsbaum vor und erklärt, wie sie die Kriterien gewählt und geordnet hat. Eventuelle Rückfragen werden von Mitschüler:innen und der Lehrkraft beantwortet.

Materialien:

- Lernmaterial „Gute-Äffchen-Böse-Äffchen-Spiel“ von [AI Unplugged](#)
- Bilder der Affen, Beschreibung der Aufgabe (digital oder analog)

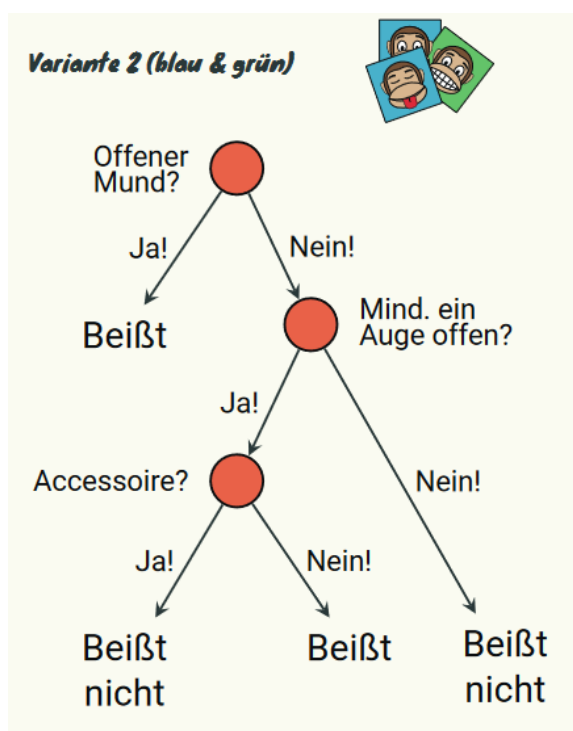


Abbildung 4: Unterrichtsmaterial „AI Unplugged“, welches in fünf Sprachen verfügbar ist und neben dem Gute-Äffchen-Böse-Äffchen-Spiel weitere analoge Spiele zum Thema KI enthält (QR-Code)
(Stefan Seegerer, Annabel Lindner/CC BY-SA 4.0)

Unterrichtsstunde 5: Erarbeitung und Präsentation der Arten maschinellen Lernens

Nachdem die Schüler:innen im „Äffchenspiel“ praktisch nachvollzogen haben, wie maschinelle Systeme Entscheidungen treffen, und in Unterrichtsstunde 3 mit der Teachable Machine ganz praxisnah erlebt haben, wie Trainingsdaten ein Modell formen und zu Klassifikationen führen, können diese Beispiele nun genutzt werden, um die verschiedenen Arten maschineller Lernverfahren besser zu verstehen. Die Teachable Machine hat dabei bereits grundlegende Elemente des überwachten Lernens veranschaulicht. Dieses Wissen bildet die Grundlage, um die drei Hauptmethoden – überwachtes Lernen (*Supervised Learning*), unüberwachtes Lernen (*Unsupervised Learning*) und verstärkendes Lernen (*Reinforcement Learning*) – systematisch zu erschließen.

Zielsetzung: Die Schüler:innen erarbeiten in Kleingruppen eine der drei Hauptarten des maschinellen Lernens (überwachtes, unüberwachtes, verstärkendes Lernen). Anschließend präsentiert pro Lernverfahren eine Gruppe ihre Arbeitsergebnisse und die Mitschüler:innen ergänzen.

In der fünften Unterrichtsstunde erarbeiten die Schüler:innen in Kleingruppen eine der drei Hauptarten des maschinellen Lernens (überwachtes, unüberwachtes, verstärkendes Ler-

nen) mithilfe des digitalen Textes „So Lernen Maschinen!“ auf der Webseite Computing Education – Maschinelles Lernen. Anschließend präsentieren sie die Ergebnisse und teilen ihr Wissen mit den anderen Gruppen.

1. Einstieg: Einführung in die Arbeitsphase (~ 5 Minuten)

- Zu Beginn der Stunde teilt die Lehrkraft die Klasse in Kleingruppen (3-5 Personen) ein, wobei sich jede Kleingruppe jeweils mit einer Art des maschinellen Lernens (überwachtes Lernen, unüberwachtes Lernen, verstärkendes Lernen) beschäftigen soll.
- Die Lehrkraft erläutert dann kurz den Ablauf der Unterrichtsstunde:
 - Jede Gruppe arbeitet mit dem Text der Webseite Computing Education – Maschinelles Lernen und orientiert sich dabei an den drei Leitfragen (s. Schritt 2). Am Ende der Stunde wird jede Art des maschinellen Lernens von einer Kleingruppe vorgestellt.

2. Arbeitsphase: Gruppenarbeit mit der Webseite (~ 25 Minuten)

- Die Schüler:innen nutzen die Webseite Computing Education – Maschinelles Lernen und orientiert sich dabei an den folgenden Leitfragen:

- Was zeichnet die Methode aus?
- Wie funktioniert sie?
- Wo findet sie Anwendung?
- Die Schüler:innen lesen jeweils nur den Abschnitt zu dem maschinellen Lernverfahren, dem sie zugeordnet wurden.
- Jede Gruppe notiert sich die wichtigsten Punkte, um am Ende der Stunde kurz die Gruppenergebnisse zu präsentieren (Plakat, Folie oder mündliche Präsentation). Die Lehrkraft unterstützt bei Fragen oder technischen Problemen und überwacht den Zeitrahmen.

3. Präsentationsphase: Vorstellung der Ergebnisse (~ 15 Minuten)

- Pro Lernverfahren präsentiert jeweils eine Gruppe ihre Ergebnisse in ca. 3 Minuten.
- Nach jeder Präsentation gibt es Zeit für Rückfragen oder Ergänzungen durch die Mitschüler:innen oder die Lehrkraft.

Materialien:

- Karten oder Tafel für Gruppenfindung und Leitfragen
- Webseite [Computing Education – Maschinelles Lernen](#)
- Geräte mit Internetzugang (Tablets, Laptops oder Smartphones)
- Materialien für Präsentationen (Papier, Marker oder digitale Tools wie PowerPoint)

Weitere Informationen zur Thematisierung der verschiedenen Lernverfahren sind auch bei Dorn (2024) zu finden.

Unterrichtsstunde 6: Zusammenfassung und Rückbezug zu ELSI

Nach der Erarbeitung der drei Hauptarten maschinellen Lernens in der fünften Unterrichtsstunde bietet die abschließende sechste Stunde die Möglichkeit, das Gelernte zu verknüpfen. Hierbei wird ein Rückbezug zu ELSI hergestellt, um die zuvor behandelten technischen Prinzipien kritisch im gesellschaftlichen Kontext zu reflektieren. Durch eine moderierte Diskussion und die Formulierung von Handlungsempfehlungen erhalten die Schüler:innen die Gelegenheit, ihre Erkenntnisse zu vertiefen und praxisnahe Handlungsoptionen für einen verantwortungsvollen Umgang mit KI-Systemen zu entwickeln.

Zielsetzung:

Die Schüler:innen fassen die wichtigsten Inhalte der Unterrichtssequenz zusammen, reflektieren ethische, rechtliche und soziale Herausforderungen im Kontext von KI und formulieren Handlungsoptionen für einen verantwortungsvollen Umgang mit KI-Systemen.

1. Einstieg: Rückblick auf die Unterrichtsreihe (~ 5 Minuten)

- Die Lehrkraft fasst die zentralen Inhalte der vorherigen Stunden knapp zusammen.
- Die Schüler:innen ergänzen spontan, welche Inhalte oder Themen sie besonders interessant fanden oder welche offenen Fragen sie noch haben.

2. Arbeitsphase: Diskussion und Rückbezug zu ELSI (~ 25 Minuten)

- Die Lehrkraft moderiert eine Diskussion, bei der die Schüler:innen Verbindungen zwischen den erlernten technischen Prinzi-



Abbildung 5: Unterrichtsmaterial „So Lernen Maschinen!“ (QR-Code) (Stefan Seegerer, Tilman Michaeli, Sven Jatzlau / CC BY 4.0)

pien und den ethischen Herausforderungen herstellen.

- Leitfragen für die Diskussion:
 - Was passiert, wenn unvollständige oder verzerrte Daten in ein KI-System eingespeist werden?
 - Welche Risiken könnten KI-Systeme in sensiblen Bereichen wie Schule oder Strafverfolgung haben?
 - Wie kann man verhindern, dass KI-Systeme diskriminieren?
 - Wer trägt Verantwortung für Entscheidungen von KI-Systemen?
- Bei Bedarf wird ein vorher von der Lehrkraft ausgewählter Ausschnitt aus dem Audio-Interview des KI-Campus: „Interview zu algorithmischer Diskriminierung“ als zusätzlicher Input genutzt.
 - Die Lehrkraft spielt den Ausschnitt aus dem Interview ab.
 - Anschließend reflektieren die Schüler:innen die Inhalte des Interviews und beziehen diese in die Diskussion ein.

3. Abschluss und Ausblick (~ 10 Minuten)

- Ergebnissicherung: Die Schüler:innen formulieren abschließend eine kurze Handlungsempfehlung oder einen „Leitsatz“ für einen verantwortungsvollen Umgang mit KI. Dies kann in Form eines kurzen Satzes erfolgen, z. B.:
 - „KI-Systeme sollten immer mit vielfältigen Daten trainiert werden, um Diskriminierung zu vermeiden.“

- „Entwickler:innen tragen die Verantwortung, KI-Systeme transparent und fair zu gestalten.“

Materialien:

- Diskussionsleitfragen (digital oder analog)
- ggf. Interview des KI-Campus: „[Interview zu algorithmischer Diskriminierung](#)“
- ggf. technisches Endgerät mit Internetzugang (Tablet, Laptop) sowie Lautsprecher, um Interview abzuspielen
- Materialien zur Ergebnissicherung (z. B. ein digitales Padlet oder Papier und Stifte für die Leitsätze)

Hinweis

Die hier skizzierte Unterrichtssequenz ist potentiell schulformübergreifend für die Jahrgangsstufen 7 bis 10 geeignet. Natürlich sind die Lernvoraussetzungen und -bedürfnisse je nach konkreter Lerngruppe – und natürlich auch innerhalb der jeweiligen Lerngruppen – sehr unterschiedlich, so dass eine differenzierende Anpassung und Konkretisierung der unterbreiteten Vorschläge erforderlich ist. Dies kann entweder über eine Differenzierung der Lernziele und Lerninhalte oder über eine Differenzierung der Methoden, also der Arbeits- und Sozialformen, Materialien und Medien erfolgen (Gehrer & Nusser, 2020). Besonders wichtig ist eine sprachensible Ausgestaltung der Lerngelegenheiten. Auch hierfür können KI-basierte Tools zum Einsatz kommen, bspw. um die hier vorgeschlagenen informierenden Texte auf unterschiedlichen sprachlichen Anforderungsniveaus zu entlasten oder um Aufgabenstellung sprachlich zu differenzieren. Dies kann bspw. bedeuten, dass zu den jeweiligen Leitfragen

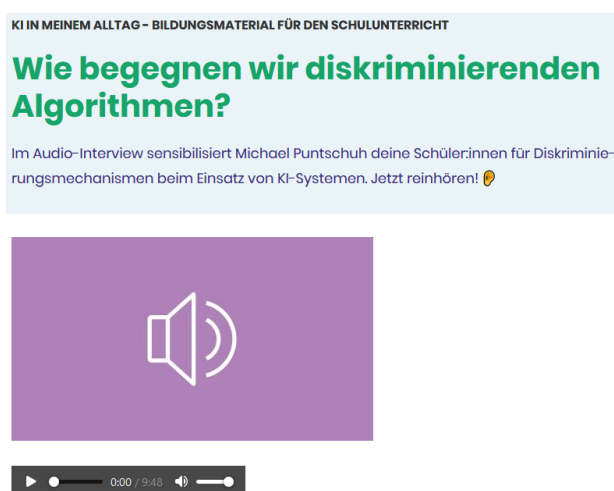


Abbildung 6: Audio-Interview „Wie begegnen wir diskriminierenden Algorithmen?“ (QR-Code) (KI-Campus / CC BY-SA 4.0)

sprachliche Scaffolds in Form von Satzanfängen oder Lückentexte vorbereitet werden.

Fazit und Ausblick

Die vorgelegten Materialien bieten Lehrkräften umfassende Möglichkeiten, spannende ELSI-bezogene Aspekte als Ausgangspunkt zu nutzen, um mit den Schüler:innen KI-Systeme und ML-Verfahren zu thematisieren. Die Einbindung dieser Aspekte in den Informatikunterricht und die Vermittlung grundlegender technischer Kenntnisse kann Schüler:innen dazu befähigen, sich an der aktuellen Diskussion um den Einsatz von KI sowie die damit verbundenen ethischen Herausforderungen kompetent zu beteiligen.

Je nach Vorwissen der Schüler:innen, zeitlicher Unterrichtskapazität und Flexibilität des jeweiligen Rahmenlehrplans sollten die hier zunächst nur oberflächlich angeschnittenen maschinellen Lernverfahren und damit einhergehende ELSI in weiteren Unterrichtsstunden vertieft werden (vgl. bspw. Dorn, 2024). Ein mögliches Anschlusssthema für weitere Unterrichtsstunden könnte die Beschäftigung mit der Funktionsweise und den Auswirkungen von LLM wie ChatGPT sein. Diese Systeme werfen für das schulische Lehren und Lernen umfassende ELSI auf. Dazu zählen bspw. Aspekte wie der Umgang mit Halluzinationen in LLM-Ausgaben sowie Unklarheiten hinsichtlich des Urheberrechts KI-generierter Texte und damit möglicherweise einhergehende Plagiate (SWK, 2024). Soziale Herausforderungen umfassen bspw. das Risiko der Verstärkung von Bildungsungleichheiten, so ist der Zugang zu (generativen) KI-Systemen wie LLM ist nicht für alle Schüler:innen gleichermaßen gegeben, was bestehende digitale Bildungsdisparitäten (Digital Divide) weiter verschärfen könnte (KMK, 2024). Dies betrifft sowohl den Zugang zu KI-Anwendungen als auch die Kompetenz, diese lernförderlich und kritisch-reflektiv einzusetzen.

Darüber hinaus könnte die Beschäftigung mit *Deep Fakes* ein spannendes Anschlusssthema sein, um weitere ELSI zu thematisieren. Deep Fakes sind täuschend echte, aber manipulierte Inhalte wie Videos oder Audios und können potenziell schwerwiegende gesellschaftliche Folgen haben. Themen wie die Verbreitung von Desinformation, die Untergrabung von Vertrauen in digitale Inhalte oder die Risiken im Zusammenhang mit Persönlichkeitsrechten und Datenschutz könnten hier in den Fokus rücken (Pawelec, 2024; Pawelec & Bieß, 2021).

Für einen didaktisch, pädagogisch und fachinhaltlich zielführenden Einsatz der hier beschriebenen Lernmaterialien ist es entscheidend, dass (Informatik-)Lehrkräfte über ausreichende KI-Kompetenz verfügen, um diese Unterrichtsmaterialien (selbst-)wirksam einsetzen zu können. Obwohl KI bereits Einzug in die Schulen gehalten hat (vgl. Befunde der Vodafone Stiftung Deutschland, 2024), bestehen bei Lehrkräften mitunter Vorbehalte oder Zweifel, auch in Bezug auf ihre eigenen Kompetenzen. Auf die Frage, was sie sich für den Einsatz von KI in der Schule wünschen, nennen Lehrkräfte neben Aspekten wie Datenschutz und Datensicherheit daher am häufigsten den Bedarf an *Weiter- und Fortbildungen* (Böhme & Mesenhöller, 2024).

Um diesem Bedarf gerecht zu werden, gibt es derzeit zahlreiche – auch bildungspolitische – Initiativen (vgl. KMK, 2024; SWK, 2024), die sich dem Thema der Lehrkräfteprofessionalisierung im Kontext der Nutzung von KI in der Schule widmen. Ein Beispiel aus der empirischen Bildungsforschung ist das KISS-Projekt, das sich dieser Herausforderung im Kontext des sprachlichen Lernens annimmt und ab Herbst 2025 Open Educational Resources in Form eines Massive Open Online Courses (kurz MOOC) für Lehrkräfte zur Verfügung stellt (kiss-pro.de).

Literatur

- Akter, S., McCarthy, G., Sajib, S., Michael, K., Dwivedi, Y. K., D'Ambra, J. & Shen, K. N. (2021). Algorithmic bias in data-driven innovation in the age of AI. *International Journal of Information Management*, 60, 102387. <https://doi.org/10.1016/j.ijinfomgt.2021.102387>
- Apfel, M. & Brandt, M. (2019). Rahmenplan für die Sekundarstufe I - Informatik und Medienbildung. https://www.bildung-mv.de/export/sites/bildungsserver/downloads/unterricht/rahmenplaene_allgemeinbildende_schulen/Informatik/RP_INFO_AHR_5-10.pdf
- Berliner Senatsverwaltung für Bildung, Jugend und Familie & Ministerium für Bildung, Jugend und Sport des Landes Brandenburg. (2015, 18. November). Rahmenlehrplan für die Jahrgangsstufen 1 - 10 der Berliner und Brandenburger Schulen, Teil C - Informatik. https://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche_Fassung/Teil_C_Informatik_2015_11_10_WEB.pdf
- Boden, A., Liegl, M. & Büscher, M. (2021). Ethische, rechtliche und soziale Implikationen (ELSI). In

- Sicherheitskritische Mensch-Computer-Interaktion (S. 185–205). Springer Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-32795-8_9
- Böhme, K. & Mesenhöller, J. (2024). Meine Kollegin, die KI – Wie die Nutzung von Künstlicher Intelligenz das schulische Lehren und Lernen verändert. In S. Schork (Hrsg.), *Vertrauen in Künstliche Intelligenz: Eine multi-perspektivische Betrachtung* (S. 79–99). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-43816-6_5
- Brinda, T., Diethelm, I., Gemulla, R., Romeike, R., Schöning, J. & Schulte, C. (2016). Dagstuhl-Erklärung: Bildung in der digitalen vernetzten Welt. <https://doi.org/10.13140/RG.2.1.3957.2245>
- Buolamwini, J. A. (2017). Gender shades: intersectional phenotypic and demographic evaluation of face datasets and gender classifiers. Massachusetts Institute of Technology, School of Architecture and Planning. <https://dspace.mit.edu/handle/1721.1/114068>
- Dorn, J. (2024). Künstliche Intelligenz im Informatikunterricht. Informatische Bildung in Schulen. Vorab-Onlinepublikation. <https://doi.org/10.18420/IBIS-02-02-02>
- Einigung der EU-Staaten: KI-Gesetz ist endgültig beschlossen (21. Mai 2024). [tagesschau.de. https://www.tagesschau.de/ausland/europa/ki-gesetz-eu-102.html](https://www.tagesschau.de/ausland/europa/ki-gesetz-eu-102.html)
- Europäisches Parlament. (2023, 17. April). Berichtigung: des in erster Lesung am 13. März 2024 festgelegten Standpunkts des Europäischen Parlaments im Hinblick auf den Erlass der Verordnung (EU) 2024/ des Europäischen Parlaments und des Rates zur Festlegung harmonisierter Vorschriften für künstliche Intelligenz und zur Änderung der Verordnungen (EG) Nr. 300/2008, (EU) Nr. 167/2013, (EU) Nr. 168/2013, (EU) 2018/858, (EU) 2018/1139 und (EU) 2019/2144 sowie der Richtlinien 2014/90/EU, (EU) 2016/797 und (EU) 2020/1828 (Verordnung über künstliche Intelligenz). https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138-FNL-COR01_DE.pdf
- Gehrer, K. & Nusser, L. (2020). Binnendifferenzierender Deutschunterricht und dessen Einfluss auf die Lesekompetenzentwicklung in der Sekundarstufe I. *Journal for educational research online*, 12(2), 166–189. <https://doi.org/10.25656/01:20976>
- Gesellschaft für Informatik e.V. (2024). Informatikmonitor - Angebot des Informatikunterrichts. <https://informatik-monitor.de/2023-24/laendervergleich>
- Google. (2024a). Teachable Machine - Bring einem Computer bei, deine eigenen Bilder, Töne und Posen zu erkennen. <https://teachablemachine.withgoogle.com>
- Google. (2024b). Teachable Machine - Was genau passiert beim maschinellen Lernen? <https://teachablemachine.withgoogle.com/faq#Weitere-Informationen>
- International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822), 860–921. <https://doi.org/10.1038/35057062>
- Juengst, E. T. (1991). The Human Genome Project and bioethics. *Kennedy Institute of Ethics journal*, 1(1), 71–74. <https://doi.org/10.1353/ken.0.0139>
- KI-Campus. (2024). Können Maschinen diskriminieren? <https://ki-campus.org/texts/diskriminierung>
- Lindner, A. & Seegerer, S. AI Unplugged: Wir ziehen künstlicher Intelligenz den Stecker. Aktivitäten und Unterrichtsmaterial zu künstlicher Intelligenz ohne Strom. <https://www.aiunplugged.org/german.pdf>
- Mantelero, A. (2018). AI and Big Data: A blueprint for a human rights, social and ethical impact assessment. *Computer Law & Security Review*, 34(4), 754–772. <https://doi.org/10.1016/j.clsr.2018.05.017>
- Parker, L. S., Sankar, P. L., Boyer, J., Jean McEwen, J. D. & Kaufman, D. (2019). Normative and conceptual ELSI research: what it is, and why it's important. *Genetics in medicine : official journal of the American College of Medical Genetics*, 21(2), 505–509. <https://doi.org/10.1038/s41436-018-0065-x>
- Pawelec, M. (2024). Wenn der Schein trügt – Deepfakes und die politische Realität - Auf einen Blick. <https://www.bpb.de/lernen/bewegtbild-und-politische-bildung/555994/auf-einen-blick/>
- Pawelec, M. & Bieß, C. (2021). Deepfakes: Technikfolgen und Regulierungsfragen aus ethischer und sozialwissenschaftlicher Perspektive (1. Auflage). *Kommunikations- und Medienethik: Bd. 16*. Nomos Verlagsgesellschaft mbH & Co. KG. <http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1955149>
- Raji, I. D., Gebru, T., Mitchell, M., Buolamwini, J., Lee, J. & Denton, E. (2020). Saving Face: Investigating the Ethical Concerns of Facial Recognition Auditing. In A. Markham, J. Powles, T. Walsh & A. L. Washington (Hrsg.), *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (S. 145–151). ACM. <https://doi.org/10.1145/3375627.3375820>
- Sekretariat der Ständigen Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland. (2016). Strategie der Kultusministerkonferenz „Bildung in der digitalen Welt“: Beschluss der Kultusministerkonferenz vom 08.12.2016. <https://www.kmk.org/fileadmin/Dateien/>

pdf/PresseUndAktuelles/2017/Digitalstrategie_KMK_Weiterbildung.pdf

Sekretariat der Ständigen Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland. (2021). Lehren und Lernen in der digitalen Welt: Die ergänzende Empfehlung zur Strategie "Bildung in der digitalen Welt". <https://www.kmk.org/aktuelles/artikelansicht/lehren-und-lernen-in-der-digitalen-welt-kultusministerkonferenz-verabschiedet-ergaenzende-empfehlung.html>

Sekretariat der Ständigen Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland. (2024). Handlungsempfehlung für die Bildungsverwaltung zum Umgang mit künstlicher Intelligenz in schulischen Bildungsprozessen: Themenspezifische Handlungsempfehlung (Beschluss der Bildungsministerkonferenz vom 10. Oktober 2024). https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2024/2024_10_10-Handlungsempfehlung-KI.pdf

Stahl, B. C. & Wright, D. (2018). Ethics and Privacy in AI and Big Data: Implementing Responsible Research and Innovation. *IEEE Security & Privacy*, 16(3), 26–33. <https://doi.org/10.1109/MSP.2018.2701164>

Ständige Wissenschaftliche Kommission der Kultusministerkonferenz. (2024). Large Language Models und ihre Potenziale im Bildungssystem: Impulspapier der Ständigen Wissenschaftlichen Kommission der Kultusministerkonferenz. <http://dx.doi.org/10.25656/01:28303>

Stifterverband. (2023, 12. Januar). Informatikunterricht: Deutschland hat großen Nachholbedarf [Pressemitteilung]. https://www.stifterverband.org/pressemitteilungen/2023_01_12_informatikunterricht

Vodafone Stiftung Deutschland. (2024). Pioniere des Wandels: Wie Schüler:innen KI im Unterricht nutzen möchten. <https://www.vodafone-stiftung.de/wp-content/uploads/2024/03/Pioniere-des-Wandels-wie-Schueler-innen-KI-im-Unterricht-nutzen-wollen-Jugendstudie-der-VS-2024.pdf>

Ye, H., Liu, T., Zhang, A., Hua, W. & Jia, W. (2023, 13. September). Cognitive Mirage: A Review of Hallucinations in Large Language Models. <http://arxiv.org/pdf/2309.06794v1>

Für die Verfassung dieses Beitrags wurde ChatGPT-4o als Formulierungshilfe genutzt.

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Janne Mesenhöller und Katrin Böhme

Universität Potsdam

Inklusionspädagogik mit dem Schwerpunkt Sprache

mesenhoeller@uni-potsdam.de

katrin.boehme@uni-potsdam.de

Interdisziplinärer Informatikunterricht: Ergebnisse zweier empirischer Untersuchungen aus Baden-Württemberg

Bahr, T.

DOI: 10.18420/ibis-03-01-04

Zusammenfassung

Informatikunterricht ist in Deutschland in den Bundesländern unterschiedlich in Bildungspläne integriert. In Baden-Württemberg wurde seit 2018 ergänzend zum Pflichtfach Informatik in Klasse 7, das Wahlpflichtfach Informatik, Mathematik, Physik (IMP) eingeführt. Hiermit ergeben sich verschiedene Fragestellungen hinsichtlich der interdisziplinären Umsetzung sowie der Interessen, Kurswahlentscheidungen und beruflichen Orientierung der Schüler:innen. Der interdisziplinäre Kontext bietet die Möglichkeit Physical Computing, KI und Design Thinking in den Unterricht zu integrieren. In einer qualitativen Interviewstudie wurden $N_1 = 21$ IMP-Lehrpersonen u. a. zu den o. a. Aspekten befragt. Weiterhin wurden in einer quantitativen und qualitativen Fragebogenstudie $N_2 = 336$ IMP-Schüler:innen am Ende der 10. Klasse zu ihren Interessen, der Vernetzung und weiteren Aspekten befragt. Die Ergebnisse liefern einen interessanten Einblick in die interdisziplinäre Umsetzung von IMP und können Hinweise zur Bildungsplanarbeit sowie zur unterrichtlichen Umsetzung interdisziplinärer Fächer liefern.

Einleitung

In der zunehmenden digitalen Transformation (OECD, 2019a) und aufgrund der veränderten Anforderungen an zukünftige Arbeitnehmer:innen im 21. Jahrhundert (OECD, 2019a) werden informatische Kompetenzen immer bedeutsamer (SWK, 2022). Daraufhin wurde in verschiedenen Bundesländern Informatik verpflichtend für alle Schüler:innen eingeführt (Hellmig et al., 2023). Die meisten Bundesländer haben Bildungspläne, die auf den Bildungsstandards der Gesellschaft für Informatik e.V. (GI) (2008) basieren. In Baden-Württemberg wurde 2018 Informatik in Klasse 7 mit einer Wochenstunde verpflichtend eingeführt (MJKS, 2018a). Weiterführend haben Schüler:innen die Möglichkeit, zwischen verschiedenen Wahlpflichtfächern für die Klassen 8 bis 10 zu wählen. Diese vierstündigen Fächer können eine dritte Fremdsprache, Sport, Musik, Naturwissenschaft und Technik (NwT), oder Informatik, Mathematik, Physik (IMP) sein. Letzteres baut auf den Prozess- und

Inhaltsbereichen von Klasse 7 auf und ergänzt die Informatik um „die physikalischen und mathematischen Grundlagen informatischer Systeme“ (MJKS, 2018b, S. 4). Theoretisch adressiert dieser Bildungsplan somit die Forderung verschiedener Expert:innen interdisziplinäre Inhalte in den Schulunterricht zu integrieren (OECD, 2019b).

Gleichzeitig sieht sich die Informatik mit weiteren Herausforderungen konfrontiert. So existiert in der Informatik im schulischen und hochschulischen Bereich sowie auf dem Arbeitsmarkt ein Gender Gap (Happe et al., 2020; Kompetenzzentrum Technik-Diversity-Chancengleichheit e. V., 2023). Interdisziplinärem Unterricht wird das Potential attestiert durch hands-on Projekte z. B. Physical Computing oder Making (vgl. Dittert et al., 2023; Przybylla, 2018) und andere Kontexte z. B. Nachhaltigkeit und Umwelt (Bahr & Zinn, 2023; Spieler, 2023) mehr Schülerinnen für die Informatik zu begeistern. Mit der skizzierten Ausgangssituation stellten sich im Bezugsfeld des interdisziplinären Fachs IMP verschiedene Fragestellungen, die basierend auf den Ergebnissen vorangegangener Arbeiten des Autors, mit diesem Beitrag in der Rubrik aus der Wissenschaft für die Praxis in Teilen beantwortet, aufgearbeitet sowie in den Kontext interdisziplinären Informatikunterrichts diskutiert und eingeordnet werden sollen, um Erkenntnisse für die schulische Praxis und Bildungsadministration abzuleiten. Dieser Beitrag greift primär auf die Ergebnisse der Dissertation des Autors (Bahr, 2024a) zurück.

Hintergrund und Forschungsstand

Nachfolgend wird der theoretische Hintergrund und Stand der Forschung hinsichtlich interdisziplinären Informatikunterrichts beschrieben. Ein Vergleich zwischen disziplinärem und interdisziplinärem Informatikunterricht wird hier nicht vorgenommen. Expert:innen der GI (2008) sowie der Ständigen Wissenschaftlichen Kommission der Kultusministerkonferenz (SWK) (2022) sprechen sich u. a. aufgrund der Bildungsgerechtigkeit für ein Pflichtfach Informatik für alle Schüler:innen ab der Sekundarstufe I aus. Weiterhin zeigt eine kürzlich erschienene Analyse der Bildungspläne in den USA, dass Fä-

cher, die Informatik nur in Teilen adressieren (integrated Computing genannt), Schüler:innen nicht ausreichend auf den Informatikunterricht in höheren Klassenstufen vorbereiten und somit ein geringes Vorwissen vorhanden sein kann (Margulieux et al., 2024, S. 18). Daher versucht dieser Beitrag, basierend auf den Ergebnissen, ein Beschreibungs- und Erklärungswissen zu interdisziplinären Wahlfächern, Arbeitsgemeinschaften, außerschulischen Aktivitäten und der Integration von interdisziplinären Kontexten in das Fach Informatik abzuleiten.

Vernetzungsmöglichkeiten

Im amerikanischen Raum wurde der K-12 Informatik Referenzrahmen von verschiedenen Expert:innen entwickelt (K-12 Computer Science Framework Steering Committee, 2016). Hier werden u. a. Vernetzungsmöglichkeiten bei den prozessbezogenen Kompetenzen zwischen der **Informatik**, der **Mathematik**, den **Naturwissenschaften und den Ingenieurwissenschaften** genannt (Abbildung 1). Zwischen **Informatik und Mathematik** werden das *Entwickeln und Abstrahieren*, das *Nutzen kollaborativer Tools* und das *präzise Kommunizieren* genannt. Das *Kommunizieren mit und über Daten* und das *Erstellen von Artefakten* werden als Schnittmengen der **Informatik und den Natur- und Ingenieurwissenschaften** identifiziert. Gemeinsame Prozesse zwischen **allen beteiligten Wissenschaften** sind nach dem Referenzrahmen das *Definieren von Problemen*, das *Modellieren*, *Computational Thinking* sowie das *Bewerten und Begründen*. Explizite Vernetzungsmöglichkeiten zwischen den Inhaltsbereichen werden nicht aufgeführt.

Als weitere Kategorisierung für interdisziplinäre MINT-Fächer (Mathematik, Informatik, Natur-

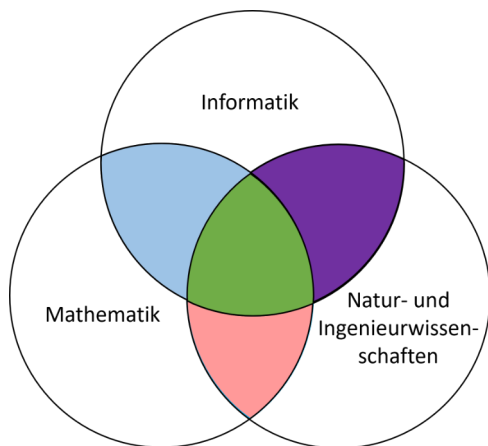


Abbildung 1: Beziehungen zwischen Informatik, Mathematik, Natur- und Ingenieurwissenschaften (übersetzt nach K-12 Computer Science Framework, 2016, S. 72 / CC BY-NC-SA 4.0)

wissenschaften und Technik) können die Definitionen von Labudde (2014) herangezogen werden. Für den fächerübergreifenden, naturwissenschaftlichen Unterricht definiert Labudde (2014) drei Unterbegriffe: *fachüberschreitend* (es wird eine Verbindung von einem Fach z. B. der Informatik zu einem anderen z. B. Mathematik hergestellt, hier könnte das binäre Zahlensystem als ein Beispiel genannt werden), *fächerverbindend* (in zwei oder mehr Fächern wird zur gleichen Zeit ein Thema erarbeitet, z. B. Kryptographie aus Sicht der Informatik mit dem RSA-Verfahren, aus Sicht der Mathematik mit den mathematischen Grundlagen wie modulo-Rechnung sowie aus Sicht der Physik mit Quantencomputern (Lee et al., 2023)) und *fächerkoordinierend* (es steht ein Problem oder eine Frage im Mittelpunkt, z. B. das optimale Auslösen eines Fallschirms (Welker, 2020)).

Der IMP-Bildungsplan (MJKS, 2018b) bietet auf Seite der inhaltsbezogenen Kompetenzen verschiedene Anhaltspunkte zur Vernetzung der drei beteiligten Fächer. In Klasse 8 wird im Mathematik-Teil bei den *mathematischen Grundlagen der Kryptologie* (Binärsystem, Primfaktorzerlegung und dem euklidischem Algorithmus) und der Graphentheorie auf die Inhaltsbereiche des Informatik-Teils des Bildungsplans verwiesen. Beim Mathematik-Teil der 9. Klasse finden sich die Verweise auf die Informatik in den gleichen Bereichen wieder. Inhaltsbezogenen Kompetenzen der Physik bei der die Inhalte der Informatik aufgeführt sind, werden in Klasse 8 nur bei den „*Funktionen optischer Geräte*“ und „*Aspekten der analogen und digitalen Bildfassung*“ aufgeführt (MJKS, 2018b, S. 34). *Geometrie* sowie *Erde und Weltall* haben keinen expliziten Verweis auf andere Inhalte der Informatik. Im Physik-Teil in Kl. 9 wird bei der *Elektrodynamik und Informationsverarbeitung* und *computergestützten Physik* auf die Informatik verwiesen. In Klasse 10 wird im Bereich *numerische Verfahren der Mechanik* und bei der *Transitmethode zum Nachweis von Exoplaneten* auf die Informatik verwiesen. Somit gibt es nach dem Bildungsplan Vernetzungsmöglichkeiten zwischen den Fächern aber auch fachisolierte Inhaltsbereiche. Die prozessbezogenen Kompetenzen werden im Bildungsplan nicht direkt vernetzt.

Empirische Erkenntnisse zu interdisziplinärem Informatikunterricht

Nach den Analysen von Vegas et al. (2021) und Bocconi et al. (2022) wird Informatik international hauptsächlich als eigenständiges Fach unterrichtet. Durch Physical Computing sowie

Computational Thinking findet in manchen Bereichen interdisziplinärer Unterricht statt. Empirische Erkenntnisse zur Umsetzung interdisziplinärer MINT-Fächer liegen bislang nur in geringem Maß vor (Bahr, 2024b). Die Unterrichtsvorbereitung, Abstimmung im Kollegium, organisatorische Schwierigkeiten und die Bildungspläne werden von verschiedenen Studien als Herausforderungen interdisziplinären Unterrichts genannt (Bastian et al., 2000; Häsing, 2009; Stübig et al., 2006). Gleichzeitig wird von Mokhonko et al. (2014) für das interdisziplinäre Profulfach NwT in Baden-Württemberg ein Zusammenhang zwischen dem Qualifikationsprofil der unterrichtenden Lehrpersonen und der Intensität der verschiedenen behandelten Themen geschlossen.

Nachfolgend soll nach dem kurzen Überblick zur Umsetzung eine Übersicht zu erprobten Unterrichtsmaterial, in dem Informatik mit Kontexten oder Inhalten anderer Fächer vernetzt wird, gegeben werden.

In dem mit Design Thinking entwickeltem und erprobtem Unterrichtsmaterial IT2School (Diethelm & Schaumburg, 2016) wurde u. a. Physical Computing, d. h. die Erstellung eines interaktiven Objektes oftmals in einer Kombination aus **Software und Hardware** (Przybylla, 2018), integriert. Schüler:innen können u. a. mit dem MocoMoco, dem Calliope Mini und später mit dem Arduino-Board erste Produkte umsetzen. Weiterhin wird **3D-Druck** mit einem Modul in den Informatikunterricht integriert. Empfohlen werden hierbei die Unterrichtsmethoden Design Thinking und die Projektmethode. Zusätzlich werden in den Unterrichtseinheiten unplugged (Informatik ohne Computer) Aktivitäten integriert.

Weiterhin existiert mit dem Themenbereich **Data Science** eine direkte Verbindung zwischen der Informatik und Mathematik (Grillenberger, 2019). Hierfür wurde von Grillenberger (2019) ein Kompetenzmodell für den Informatikunterricht entwickelt sowie erprobtes Material zum Thema Data Mining veröffentlicht.

Anknüpfend an das Sammeln und Analysieren von Daten wurde verschiedenes Unterrichtsmaterial im Kontext **Künstliche Intelligenz** für Schüler:innen verschiedener Altersstufen entwickelt und erprobt. Dieses umfasst z. B. die Verbindung des MIT App Inventors mit Daten aus der realen Welt in interdisziplinären Kontexten (Ravi et al., 2024).

Oben aufgeführte Themen finden sich auch im OER-Online-Schulbuch *inf-schule* wieder (Jochum et al., 2017). Somit lässt sich zusammenfassen, das basierend auf dem oben skizzierten

Unterrichtsmaterial, praxiserprobte Möglichkeiten existieren Informatik im Kontext (Diethelm et al., 2011) und vernetzt mit Inhalten anderer Fächer im Schulunterricht umzusetzen.

Forschungsstand zum Gender Gap im Kontext interdisziplinärer Schulinformatik

Verschiedene Literaturanalysen deuten an, dass interdisziplinäre Projekte das Interesse von Schülerinnen an Informatik fördern könnten (Happe et al., 2020). Demnach soll das weitverbreitete Programmieren lernen durch eine Schritt für Schritt Anleitung ohne realen Lebensweltbezug und/oder Einbettung in ein Projekt oder iterativ entstehendes Produkt, d. h. „Programmieren lernen um Programmieren zu können“ nicht alle Schüler:innen gleichermaßen ansprechen (Crick, 2017). Wie oben skizziert existieren bereits verschiedene Ansätze interdisziplinäre Projekte in das Fach Informatik zu integrieren um somit ggfs. das Interesse an Informatik sowie das Selbstkonzept der Schüler:innen in Informatik und gleichzeitig die informatischen Kompetenzen der Schüler:innen zu fördern (Diethelm & Schaumburg, 2016). Als Empfehlungen für die Förderung von Schülerinnen werden nach den Analysen des Kompetenzzentrums Technik-Diversity-Chancengleichheit (2023) und Happe et al. (2020) u. a. (1) der Besuch von außerschulischen Lernorten (z. B. Girls' Digital Camps), (2) die Ermöglichung der Teilnahme an Informatik-Wettbewerben (z. B. den Informatik-Biber), (3) Problemstellungen mit einem realen Lebensweltbezug sowie (4) die Priorisierung von Design Thinking und Problemlösen über das eigentliche Programmieren genannt. Außerdem fällt auf, dass außerschulisch Lernorte in ihren Workshops oft interdisziplinäre Themen (z. B. Digital Art oder Roboterprojekte) integrieren (Spieler, 2023).

Forschungsziel und methodisches Vorgehen

In dem oben aufgeführten Kontext war das Forschungsziel der Dissertation (Bahr, 2024a) die Generierung eines systematischen Beschreibungswissens zu den Merkmalen der Lernenden sowie der unterrichtlichen Umsetzung und Vernetzung der Lehrenden des Profulfachs Informatik, Mathematik, Physik (IMP). Dieser Beitrag aus der Rubrik „von der Wissenschaft für die Praxis“ geht insbesondere auf die Vernetzung aus Sicht der Schüler:innen und Lehrpersonen, der unterrichtlichen Umsetzung der Lehrpersonen sowie auf Geschlechterunterschiede ein, um Implikationen für die schuli-

sche Praxis und Bildungsadministration aus den gewonnenen Daten abzuleiten.

Die Erhebungen wurden zwischen November 2021 und Juli 2022 in Baden-Württemberg durchgeführt. Insgesamt nahmen $N_1 = 21$ IMP-Lehrpersonen (männlich = 15, weiblich = 6; Alter: $M = 41$ Jahre, $SD = 7.06$ Jahre) und $N_2 = 336$ ($m = 236$, $w = 88$, divers = 12) IMP-Schüler:innen am Ende der 10. Klasse teil. 31 Schulen aus allen vier Regierungsbezirken von Baden-Württemberg nahmen an der Befragung teil.

Die Daten wurden quantitativ mittels Fragebogen und qualitativ mit offenen Fragen und Interviews erfasst. Weitere Informationen zur Methodik und den Gütekriterien können in (Bahr, 2024a) nachgelesen werden.

Ergebnisse

Die Ergebnisse der durchgeführten Studien und weitere noch nicht veröffentlichte Daten werden nachfolgend aus Sicht der Lehrpersonen und Schüler:innen dargestellt.

Umsetzung, Vernetzung und Optimierungsbedarf aus Sicht der Lehrpersonen

IMP wird wie alle Profulfächer vier Stunden pro Woche unterrichtet. Die Schwerpunkte liegen dabei in Klasse 8 bei der Informatik, in Klasse 9 in Physik und in Klasse 10 in der Mathematik mit zwei Wochenstunden. Den anderen Fachanteilen wird nach dem Stoffverteilungsplan in dem jeweiligen Jahr eine Stunde pro Woche zugewiesen. Nach Aussage der Befragten wird dies zu ca. 53 % im Doppelstundenmodell umgesetzt. Dabei hat ein Fach (z. B. Informatik in Klasse 7) eine Doppelstunde pro Woche. Die anderen beiden Fächer (z. B. Mathematik und Physik in Klasse 7) werden ebenfalls in Doppelstunden, aber nur für ein Halbjahr unterrichtet. Dies hat, laut den Befragten, zur Folge, dass manche Inhalte des Bildungsplans nicht miteinander vernetzt werden können, da sie z. B. erst ein halbes Jahr später unterrichtet werden. Ca. 47 % der Befragten geben an, IMP als ein Fach mit einem fächerkoordinierenden Ansatz zu unterrichten. Dieses Modell wird entweder von zwei Lehrpersonen, die sich untereinander eng abstimmen, oder von einer Lehrperson mit allen drei Fakultäten umgesetzt.

Da IMP parallel zum regulären Mathematik- und Physikunterricht unterrichtet wird und dort auch Vernetzungen im Bildungsplan aufgeführt werden, identifizieren die Lehrpersonen Abstimmung im Kollegium als wichtigen Punkt, um Redundanzen und heterogenes Vorwissen der

Schüler:innen zu vermeiden. Nach Aussagen der Befragten gibt es jedoch nur selten Absprachen in Fachschaften oder andere systematische Absprachen (24 %). Primär finden Absprachen sporadisch (62 %) oder in manchen Fällen gar nicht statt (14 %).

Gleichzeitig nennen die Teilnehmenden Themengebiete des Bildungsplans wie die Astronomie die nur schwer zu vernetzen sind, als problematisch:

„Ja, ich war etwas erstaunt. Da ist eben die Astronomie in IMP eingeflossen ist. Die war im letzten Bildungsplan in NwT. Wir waren ja zu einigen Fortbildungen zusammengesessen. Und da hat sich zu allen der Verdacht aufgetreten, nachdem sie in NwT raus ist, musste die Astronomie halt in IMP rein, damit sie da ein Stück ihre Existenzberechtigung weiter hat. [...] Das ist jetzt Teil des Physik Plans. Und dann muss man das eben umsetzen. Da finde ich die Vernetzung schwierig.“ (Lehrperson 4).

Zusammengefasst werden als positive Beispiele für die Möglichkeiten der Vernetzung der drei beteiligten Fächer in der Kategorie **Informatik und Mathematik** die Themen Kryptologie, Graphentheorie, stückweise definierte Funktionen und Zahlensysteme genannt. In der Kategorie **Informatik und Physik** werden die Elektronik, Informationsverarbeitung und Optik genannt. Letztlich werden Simulationen, Numerik und die computergestützte Physik als Themen in denen **alle drei Fächer** miteinander vernetzt werden von den Befragten genannt.

„Die Informatik und Mathematik lassen sich an ganz vielen Stellen sehr gut vernetzen. Zum Beispiel bei den Graphen-Algorithmen. Wenn man in der Mathematik den Graphen als Datenstruktur kennenlernt. Man macht sich Gedanken über zum Beispiel kürzeste-Wege-Algorithmen. Diese Algorithmen macht man erst an dem Graphen. Anschließend kann man in der Informatik implementieren, erweitern, Effizienz-Betrachtung, usw. machen. (...) Ich habe schon über die Aussagenlogik gesprochen. Die ist ein großes Thema ab Klasse neun, wo man sich eben ganz intensiv mit Boolescher Algebra in der Mathematik beschäftigt. Das passt natürlich super, es würde auch gut mit der Physik zusammenpassen, wenn man eben in der Physik die Transistoren behandelt usw. Auf Gatter-Ebene kann ich das in der Informatik sehr schön übernehmen. Das ist eben

der üblichen Gang vom Halbaddierer, über Volladdierer bis hin zur CPU und vieles mehr.“ (Lehrperson 19).

Nach der Betrachtung den Vernetzungsmöglichkeiten stellt sich die Frage wie die Lehrpersonen das Fach IMP umsetzen. Hier geben alle Befragten an, das entwickelte Unterrichtsmaterial vom Zentrum für Schulqualität und Lehrerfortbildung (ZSL) in Baden-Württemberg zu nutzen. Dieses Material wurde von der Zentralen Planungsgruppe (ZPG) des IMP Bildungsplans basierend auf vorher erprobten Unterrichtsmaterial u. a. von Gallenbacher (2017) (in Klasse 8 bei *Daten und Codierung*) und inf-schule (Jochum et al., 2017) (in Klasse 8 bei *Algorithmen*, Klasse 9 *Daten und Codierung*) entwickelt oder selbst erprobt. Nachfolgend werden die Unterrichtsgänge kurz beschrieben. Im Informatik-Teil in Klasse 8 wurde der *Algorithmen*-Teil mit dem MIT-App Inventor erprobt. Im *Rechner und Netze*-Teil in Klasse 8 und 10 wird mit Filius (Freischlad, 2010) gearbeitet. Das Material im *Informationsgesellschaft und Datensicherheit*-Teil in allen Klassen wurde selbst entwickelt. Ebenso das Material im *Algorithmen*-Teil der Klassen 9 und 10. Vernetzungen werden nur **zum mathematischen Teil** mit der „Bestimmung des ggT mit dem euklidischen Algorithmus, das Sieb des Eratosthenes zur Bestimmung von Primzahlen oder das Finden der Primfaktorzerlegung einer Zahl“ (Eisenmann, 2018, S. 15) genannt. Der Physikteil nennt im Unterrichtsgang der Klassen 8, 9 und 10 nur Vorwissen der Schüler:innen in Informatik aber keine expliziten Vernetzungen. Ein Scratch Programm kann in Klasse 8 zu Visualisierung genutzt werden. GeoGebra und Excel werden in Klasse 9 und 10 eingesetzt sind jedoch keine Inhalte der Informatik, sondern als Werkzeuge zu verstehen. Der Mathematik-Teil enthält im *Geometrie*-Teil keine Verweise. Sonst werden Bezüge zur Informatik hergestellt. Im Teil *Funktionen im Sachkontext* wird beim Thema Erfassung von Umrissen mithilfe von Schrittweitensteuerung ein Projekt das **fachüberschreitend oder fächerverbindend mit der Informatik** durchgeführt werden kann empfohlen. Außerdem geben die Befragten an, dass im IMP-Unterricht im Vergleich zu NwT weniger Projekte umgesetzt werden.

„Die NwT als handlungsorientierte Disziplin und die IMP eher als doch kognitiv-wissenschaftlich-geartete Disziplin.“ (Lehrperson 20).

Neben den gewonnenen Erfahrungen aus der ersten Umsetzung der Lehrpersonen wird wei-

terer Optimierungsbedarf genannt (81 %). Darunter fallen Änderungen am Bildungsplan hinsichtlich der Vernetzung der Fächer (57 %), Absprachen im Kollegium (19 %), sowie die Geschlechterverteilung der Schüler:innen.

„Eben durch den Alltagsbezug. (...) Die zweite Herausforderung ist dadurch auch für Mädchen attraktiv zu machen. Was bei uns eben bisher eher weniger Mädchen sind. (...) weil ich mir gut vorstellen kann, dass da eine höhere Hemmschwelle ist. Bei Mädchen zum einen, wenn sie nachher wissen sie sind dann im Kurs, wo vielleicht fast nur Jungs sind aber auch, weil ja die Ermutigung, in Richtung Informatik zu gehen (...).“ (Lehrperson 2).

Vernetzung aus Sicht der Schüler:innen

Vernetzungen zwischen **Informatik und Mathematik** sehen die Teilnehmenden primär in der Kryptologie (35 %), Aussagenlogik (23 %) und beim Programmieren (20 %).

„Im Matheteil haben wir den RSA-Algorithmus kennengelernt und ich habe diesen selbstständig programmiert.“ (Schüler:in 255)

Bei den Bereichen **Informatik und Physik** werden die elektrotechnischen Grundlagen (31 %), computergestützte Physik (28%) und numerische Methoden der Mechanik (21 %) genannt. Ein Beispiel ist:

„Technische Informatik (Schaltkreise) und Halbleiter (Physik: PN-Übergang, Dioden, Transistoren, Logische Gatter)“ (Schüler:in 303).

Die computergestützte Physik (25 %) und Aussagenlogik (28 %) werden als Vernetzungsmöglichkeiten **aller drei Fächer** gesehen. Hier wird zum Beispiel Physical Computing (3 %) aufgeführt:

„Mit einem Calliope Mini Microcontroller + Sensoren wurden physikalische Messungen vorgenommen und damit Werte berechnet (z.B. Temperatur in Grad Celsius)“ (Schüler:in 79).

Nach Aussage der Schüler:innen wird primär Java als textbasierte Programmiersprache un-

terrichtet (76 %) gefolgt von Python (22 %). Anderen Programmierumgebungen die verwendet werden sind Arduino, Open Roberta Lab und der MIT App Inventor (13 %).

Geschlechterunterschiede bei den Schülerinnen

Die am häufigsten genannten Kurswahlmotive der Schüler:innen sind deren Fachinteressen hinsichtlich einer der drei beteiligten Fächer und Selbstkonzepte (Bahr & Zinn, 2023). Beim Interesse an IMP, Mathematik, der Motivation im IMP-Unterricht sowie der beruflichen Orientierung im Bereich Naturwissenschaften und Technik gibt es keine geschlechtsspezifischen Unterschiede. Signifikante Unterschiede gibt es bei dem Interesse an Informatik, Physik und dem fachspezifischem Selbstkonzept an IMP mit geringen Effektstärken. Hier haben Schüler im Mittel höhere Werte als Schülerinnen (Bahr & Zinn, 2023). In einer Folgestudie (Bahr, 2024d) mit einer größeren Stichprobe (N = 665 IMP-Schüler:innen) haben die Effekte jedoch nur eine geringe statistische Power (unter 80 %). Somit lassen sich einzig bei den Kurswahlentscheidungen Geschlechterunterschiede als statistisch signifikant sicher feststellen. Insgesamt wählen ca. 70 % Schüler das Profulfach IMP.

Diskussion und Implikationen

Die Ergebnisse hinsichtlich der Vernetzung der Lehrenden und Lernenden decken sich mit den aufgeführten Themen im Bildungsplan. Vernetzungsmöglichkeiten (z. B. Kryptologie) zwischen **Informatik und Mathematik** werden häufig gesehen. Vernetzungsmöglichkeiten zwischen **Informatik und Physik** liegen primär nur bei elektrotechnischen Themen. Bei **allen drei Fächern** in den Bereichen Numerik und computergestützte Physik. Nach den Angaben der Schüler:innen wird Physical Computing (12,5 %) eher selten in den IMP-Unterricht integriert. Gleichzeitig geben die Lehrpersonen an, IMP nicht projektbasiert oder „hands-on“ zu unterrichten (Bahr, 2024b). Als Optimierungsbedarf nennen manche Lehrpersonen den Gender Gap bei der Kurswahl. Primär wählen Schüler das Fach IMP. Somit entsteht für die Kurswahl in der Oberstufe bereits eine stereotype Vorauswahl. Geschlechtersensibler Informatikunterricht wird zwar im Bildungsplan als didaktischer Hinweis aufgeführt (MKJS, 2018b), lässt sich im Unterrichtsmaterial und der berichteten Unterrichts-umsetzung wenig wiederfinden (Eisenmann, 2018; Welken, 2018).

Dabei hat IMP, im Vergleich zu anderen Profulfächern, das Potential die beruflichen Interessen der Schüler:innen im MINT-Bereich zu fördern (Bahr et al., 2024). Nichtsdestotrotz zeigen die empirischen Ergebnisse, dass es im Sinne der Bildungsgerechtigkeit und Teilhabe an der digitalen Welt einen verpflichtenden Informatikunterricht für alle Schüler:innen benötigt. So berichten einige Lehrpersonen davon nicht genügend Zeit für die Inhalte der Informatik zu haben (Bahr, 2024b) und erste Ergebnisse des Informatik-Fachwissens der IMP-Schüler:innen deuten an, dass viele Schüler:innen ein geringes Fachwissen haben (Bahr, 2024e).

Aus den empirischen Ergebnissen der eigenen Studien und dem Forschungsstand lassen sich somit nachfolgende Implikationen ableiten:

1. Informatikunterricht sollte geschlechtersensibler gestaltet sein.

Hier kann beispielsweise die Checkliste von Spieler (2017) ein Ansatzpunkt sein, um alle Schüler:innen zu erreichen. Weiterhin nennen die Lehrpersonen die Kommunikation mit den Schüler:innen und Eltern bei der Profulfachwahl als entscheidend, Informatik als attraktiv für alle zu präsentieren (kreativ, projektbasiert, mit realen Lebensweltbezug).

2. Interdisziplinärer Informatikunterricht sollte kontextorientiert erfolgen.

Über eine stärkere Kontextorientierung (z. B.: Nachhaltigkeit, Umwelt, Biologie, Making, Medizin) besteht die Möglichkeit den diversen (geschlechtsspezifischen) Interessenprofilen der Schüler:innen gerecht zu werden (Bahr & Zinn, 2023). IMP wurde primär von Schülern aufgrund ihrer Interessen gewählt. Um den Anteil an Schülerinnen in der Oberstufe zu erhöhen, sollten jedoch auch Kontexte gewählt werden, die in einem stärkeren Maße den Schülerinnen entsprechen (Elster, 2007; Happe et al., 2020). Gleichzeitig wäre die Nutzung erprobter Ansätze wie Informatik im Kontext (Diethelm et al., 2011) bei der Entwicklung von neuem Unterrichtsmaterial für interdisziplinäre Fächer sinnvoll. Außerdem sollten interdisziplinäre Fächer die Vernetzungsmöglichkeiten im Bildungsplan besser nutzen. Idealerweise enthält der Bildungsplan des interdisziplinären Fachs direkte Verweise bei prozessbezogenen und inhaltsbezogenen Kompetenzen und das erprobte Material geht vertieft auf die Vernetzungsmöglichkeiten (z. B. nach Labudde (2014)).

3. Sollte Informatik als interdisziplinäres Fach angeboten werden, so benötigt es auch eine entsprechende Lehrpersonenbildung.

Durch die Integration von Themen wie Data Literacy (Grillenberger, 2019), KI (Ravi et al., 2024) und Physical Computing (Przybylla, 2018) werden vielfältige Anforderungen an die Qualifikationsprofile von Lehrpersonen gestellt. Da Qualifikationsprofile von Lehrpersonen mit der Behandlungsintensität zusammenhängen können (Mokhonko et al., 2014), erscheint es sinnvoll Lehrpersonen auf interdisziplinäre Fächer vorzubereiten (Zinn et al., 2018).

Ausblick

Nach der Einführung des einwöchigen Pflichtfachs Informatik in Klasse 7 und des Wahlpflichtfachs IMP soll in den nächsten Jahren das Pflichtfach „Informatik und Medienbildung“ von Klasse 5 bis 11 an allgemeinbildenden Gymnasien in Baden-Württemberg verpflichtend eingeführt werden. IMP wird eingestellt. Dafür soll in NwT, aufbauend auf den Kompetenzen in Informatik, Physical Computing ausgebaut werden, wodurch mittels Projektarbeit ein handlungsorientierter Bezug zur Informatik hergestellt werden soll. Der interdisziplinäre Informatikunterricht bleibt in Baden-Württemberg damit erhalten. Gleichzeitig gibt es in anderen Bundesländern (z. B. in Hessen mit dem Fach „Digitale Welt“ (Meinel et al., 2024)) und anderen Ländern (z. B. „Digitale Grundbildung“ in Österreich (Bocconi et al., 2022) oder CS+X-Fächer in den USA (Margulieux et al., 2024)) ebenfalls interdisziplinäre Ansätze. Entsprechend Margulieux et al. (2024) ist es hierfür ratsam, das Vorwissen der Schüler:innen in diesen Fächern nicht zu überschätzen. Gleichzeitig existiert wissenschaftlich erprobtes Unterrichtsmaterial zu verschiedenen Bereichen (Diethelm et al., 2011; Diethelm & Schaumburg, 2016; Przybylla, 2018; Ravi et al., 2024), dessen Einsatz in Anbetracht der Ergebnisse der Dissertation sinnvoll erscheint. Denn das primäre Ziel des interdisziplinären Informatikunterrichts sollte es sein, die Breite und Vielfalt der Informatik darzustellen, allgemeinbildende Problemlösefähigkeiten zu fördern und mehr Schüler:innen für Informatik zu begeistern.

Quellen

- Bahr, T. (2024a). Das interdisziplinäre Profulfach Informatik, Mathematik, Physik (IMP) – Eine explorative Forschungsstudie zur unterrichtsfachlichen Umsetzung, den Lehrenden und Lernenden des Profulfachs. Dissertation. <http://elib.uni-stuttgart.de/handle/11682/15536>
- Bahr, T. & Zinn, B. (2023). Gender Differences in the New Interdisciplinary Subject Informatik, Mathematik, Physik (IMP)—Sticking with STEM? *Educ. Sci.* 13 (5), S. 478. <https://doi.org/10.3390/educsci13050478>
- Bahr, Tobias (2024b). Interdisziplinärer Informatikunterricht – zwischen Chance und Herausforderung. *Journal of Technical Education (JOTED)*. <http://dx.doi.org/10.18419/opus-15235>
- Bahr, T., Brändle, M. & Zinn, B. (2024). STEM Career Choices for K–12 Students and the Influencing Factors—A Comparison of Students in Different Support Programs. *Journal for STEM Educ Res.* <https://doi.org/10.1007/s41979-024-00129-w>
- Bahr, T. (2024c). Where is the Interdisciplinary? Insights of Interdisciplinary STEM Students! Proceedings of 2024 IEEE Global Engineering Education Conference (EDUCON), Kos, Greece. <https://doi.org/10.1109/EDUCON60312.2024.10578648>
- Bahr, T. (2024d). An Analysis of Gender Differences of Interdisciplinary STEM Students' Interest, Motivation, Self-Concept and Vocational Orientations. 2024 IEEE Global Engineering Education Conference (EDUCON), Kos, Greece. <https://doi.org/10.1109/EDUCON60312.2024.10578573>
- Bahr, T. (2024e). Piloting of a Computer Science Content Knowledge Test for 10th Grade Students. 2024 IEEE Global Engineering Education Conference (EDUCON), Kos, Greece. <https://doi.org/10.1109/EDUCON60312.2024.10578675>
- Bastian, J., Combe, A., Gudions, H., Herzmann, P. & Rabenstein, K. (2000). Profile in der Oberstufe. Fächerübergreifender Projektunterricht in der Max-Brauer-Schule Hamburg. PB-Bücher, 39, (1). Bergmann + Helbig, Hamburg
- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, K., Horvath, M., Jasutė, M., Malagoli, C., Masiulionytė-Dagienė, V., Stupurienė, G. (2022). Reviewing Computational Thinking in Compulsory Education: State of Play and Practices from Computing Education. Publications Office of the European Union, Luxembourg. ISBN 978-92-76-47208-7
- Crick, T. (2017). Computing education: An overview of research in the field. London: Royal Society. <http://cronfa.swan.ac.uk/Record/cronfa43589> (zuletzt geprüft 21.01.2025)
- Diethelm, I., Koubek, J. & Witten, H. (2011). Inik—Informatik im Kontext: Entwicklungen, Merkmale und Perspektiven. *LOG IN*, 31(2), 97–104
- Diethelm, I., Schaumburg, M. (2016). IT2School – Development of Teaching Materials for CS Through Design Thinking. Brodник, A., Tort, F. (eds) *Informatics in Schools: Improvement of Informatics*

- Knowledge and Perception. ISSEP 2016. Lecture Notes in Computer Science, vol 9973. Springer, Cham. https://doi.org/10.1007/978-3-319-46747-4_16
- Dittert, N., Daeglau, M., Pancratz, N. & Diethelm, I. (2023). Breaking Gender Barriers in Computer Science: Exploring the Impact of Digital Fabrication Workshops in Smart Environments. 16th International Conference on Informatics in Schools, ISSEP 2023, Local Proceedings (ISSEP2023), Lausanne, Switzerland. <https://doi.org/10.5281/zenodo.8431891>
- Eisenmann, M. (2018). Vertiefung Algorithmen. Unterrichtsgang mit Hintergrundinformationen. Lehrerfortbildung Baden-Württemberg. (Hrsg.) Zentrum für Schulqualität und Lehrerbildung (ZSL), Stuttgart. https://lehrerfortbildung-bw.de/u_matnatech/imp/gym/bp2016/fb1/2_i2_alg/1_hintergrund/2_verlauf/01_alg_unterricht_hintergrund.pdf (zuletzt geprüft 01.08.2024)
- Elster, D. (2007). In welchen Kontexten sind naturwissenschaftliche Inhalte für Jugendliche interessant. Plus Lucis.
- Freischlad, S. (2010). Entwicklung und Erprobung des Didaktischen Systems Internetworking im Informatikunterricht (Vol. 3). Universitätsverlag Potsdam. <http://pub.ub.uni-potsdam.de/volltexte/2010/4185>
- Gallenbacher, J. (2017). Abenteuer Informatik. IT zum Anfassen für alle von 9 bis 99 – vom Navi bis Social Media. Springer Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-63739-5>
- Gesellschaft für Informatik e. V. [GI] (2008). Grundsätze und Standards für die Informatik in der Schule Bildungsstandards Informatik für die Sekundarstufe I. LOG IN 28 (150/151). https://informatikstandards.de/fileadmin/GI/Projekte/Informatikstandards/Dokumente/bildungsstandards_2008.pdf (zuletzt geprüft 01.08.2024)
- Grillenberger, A. (2019). Von Datenmanagement zu Data Literacy: Informatikdidaktische Aufarbeitung des Gegenstandsbereichs Daten für den allgemeinbildenden Schulunterricht. Dissertation, Freie Universität Berlin
- Grillenberger, M. (2023). Why and How to Teach Physical Computing: Research and Practice in Computer Science Education at Secondary Schools. Keane, T., Fluck, A.E. (eds) Teaching Coding in K-12 Schools. Springer, Cham. https://doi.org/10.1007/978-3-031-21970-2_15
- Große-Bölting, G., Gerstenberger, D., Gildehaus, L., Mühling, A. & Schulte, C. (2021). Identity in K-12 Computer Education Research: A Systematic Literature Review. ICER 2021, Proceedings of the 17th ACM Conference on International Computing Education Research (S. 169–183). Association for Computing Machinery. <https://doi.org/10.1145/3446871.3469757>
- Happe, L., Buhnova, B., Koziolok, A. & Wagner, I. (2020). Effective measures to foster girls' interest in secondary computer science education. Education and Information Technologies, 26(3), 2811–2829. <https://doi.org/10.1007/s10639-020-10379-x>
- Häsing, P. (2009). Fächerübergreifender Unterricht in der gymnasialen Oberstufe aus Sicht der Lehrenden. Eine qualitative Studie. <https://doi.org/10.25656/01:3340>
- Haselmeier, K., Humbert, L., Killich, K. & Müller, D. (2019). Interesse an Informatik und Informatikselbstkonzept zu Beginn der Sekundarstufe I des Gymnasiums. In A. Pasternak (Hrsg.), Informatik für alle (S. 99–108). Gesellschaft für Informatik. <https://doi.org/10.18420/infos2019-b6>
- Hellmig, L., Schieckoff, B., Schwarz, R., Süßenbach, F. (2023). INFORMATIK-MONITOR 2023/24. (Hrsg.) Gesellschaft für Informatik e. V., Stifterverband für die Deutsche Wissenschaft e. V., Berlin. https://informatik-monitor.de/fileadmin/GI/Projekte/Informatik-Monitor/Informatik-Monitor_2023-24/PDF-Versionen/Informatik-Monitor_2023-24_Final.pdf (zuletzt geprüft 01.08.2024)
- Jochum, H., Becker, K., Zimnol, M., Jonietz, D., Schneider, O., & Froitzheim, M. (2017). Ein Durchgang durch das Grundfach Informatik (nur) mit dem elektronischen Schulbuch inf-schule.de. Informatische Bildung zum Verstehen und Gestalten der digitalen Welt. PISSN: 1617-5468. ISBN: 978-3-88579-668-8. pp. 345-347, Oldenburg
- K-12 Computer Science Framework Steering Committee (2016). K-12 computer science framework. ACM. <https://dl.acm.org/doi/abs/10.1145/3079760> (zuletzt geprüft 21.01.2025)
- Kompetenzzentrum Technik-Diversity-Chancengleichheit e. V. (2023). Mädchen und Frauen in die Informatik: Aktivierungspotenziale und Erfolgsfaktoren Handlungsempfehlungen Bildung. https://www.kompetenzz.de/statische_inhalte/Handlungsempfehlungen_Bildung.pdf (zuletzt geprüft 01.08.2024)
- Labudde, P. (2014). Fächerübergreifender naturwissenschaftlicher Unterricht – Mythen, Definitionen, Fakten. Zeitschrift für Didaktik der Naturwissenschaften, 20(1), 11–19. <https://doi.org/10.1007/s40573-014-0001-9>
- Lee, S., Kim, Y., Jeong, S. & Park, S. (2023). Teaching Quantum Computing at a Middle School. 16th International Conference on Informatics in Schools, ISSEP 2023, Local Proceedings (ISSEP2023),

Lausanne, Switzerland. Zenodo. <https://doi.org/10.5281/zenodo.8432008>

Margulieux, L. E., Liao, Y. C., Anderson, E., Parker, M. C., & Calandra, B. D. (2024). Intent and Extent: Computer Science Concepts and Practices in Integrated Computing. *ACM Trans. Comput. Educ.* <https://doi.org/10.1145/3664825>

Meinel, C., Galbas, M., Dengel, A., & Wendlandt, M. (2024). Konzeption eines integrativen Schulfaches „Digitale Welt“ für hessische Schulen (Nr. 160). <https://doi.org/10.25932/publishup-63911>

Ministerium für Kultus, Jugend und Sport Baden-Württemberg [MKJS] (2018a). Bildungsplan Aufbaukurs Informatik. https://bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_SEK1_INF7.pdf (zuletzt geprüft: 01.08.2024)

Ministerium für Kultus, Jugend und Sport Baden-Württemberg [MKJS] (2018b). Bildungsplan zum Profilmfach Informatik, Mathematik, Physik (IMP). http://bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_IMP.pdf (zuletzt geprüft: 01.08.2024)

Mokhonko, S., Stefanica, F. & Nickolaus, R. (2014). NwT-Unterricht: Herausforderungen bei der Einführung eines neuen Faches im Spiegel einer aktuellen Bestandsaufnahme. *Journal of Technical Education (JOTED)*, 2 (1). <https://doi.org/10.48513/joted.v2i1.28>

OECD (2019a). An OECD Learning Framework 2030. In: Bast, G., Carayannis, E.G., Campbell, D.F.J. (eds) *The Future of Education and Labor. Arts, Research, Innovation and Society*. Springer, Cham. https://doi.org/10.1007/978-3-030-26068-2_3

Organisation for Economic Co-operation and Development [OECD] (2019b), PISA 2018 Ergebnisse (Band I). Was Schülerinnen und Schüler wissen und können, PISA, wbv Media, Bielefeld, <https://doi.org/10.1787/1da50379-de>

Przybylla, M. (2018). From embedded systems to physical computing: Challenges of the “digital world” in secondary computer science education. Doctoral dissertation, Universität Potsdam.

Ravi, P., Parks, R., Masla, J., Abelson, H., & Breazeal, C. (2024). "Data comes from the real world": A constructionist approach to mainstreaming K12 data science education. *Proceedings of the 2024 ACM Virtual Global Computing Education Conference V. 1 (SIGCSE Virtual 2024)*, 271–274. Association for Computing Machinery. <https://doi.org/10.1145/3649165.3703623>

Spieler, B. (2017). Checkliste: Empfehlungen für einen gendersensiblen Informatikunterricht. https://bernadette-spieler.de/wp-content/uploads/2017/02/Checkliste_BSpieler.pdf (zuletzt geprüft 01.08.2024)

Spieler, B. (2023). Empfehlungen für gendersensible MINT-Angebote für Schülerinnen am Beispiel der Schweiz. Jeanrenaud, Y. (eds) *Teaching Gender in MINT in der Pandemie*. Edition Fachdidaktiken. Springer VS, Wiesbaden. https://doi.org/10.1007/978-3-658-43375-8_5

Ständige Wissenschaftliche Kommission der Kultusministerkonferenz [SWK] (2022). Digitalisierung im Bildungssystem: Handlungsempfehlungen von der Kita bis zur Hochschule. Gutachten der Ständigen Wissenschaftlichen Kommission der Kultusministerkonferenz (SWK). <http://dx.doi.org/10.25656/01:25273>

Stübiger, F., Ludwig, P., Bosse, D., Gessner, E. & Lorberg, F. (2006). Bestandsaufnahme zur Praxis fächerübergreifenden Unterrichts in der gymnasialen Oberstufe im Bundesland Hessen. <https://doi.org/10.25656/01:3336>

Vegas, E., Hansen, M. & Fowler, B. (2021). BUILDING SKILLS FOR LIFE: How to expand and improve computer science education around the world. Brookings. <https://www.brookings.edu/articles/building-skills-for-life-how-to-expand-and-improve-computer-science-education-around-the-world/> (zuletzt geprüft 01.08.2024)

Welker, H. (2020). Numerische Verfahren der Mechanik. Lehrerfortbildung Baden-Württemberg. (Hrsg.) Zentrum für Schulqualität und Lehrerbildung (ZSL), Stuttgart. https://lehrerfortbildung-bw.de/u_matnatech/imp/gym/bp2016/fb3/p02_nvm/1_hintergrund/2_verlauf/00_nvm_unterrichtsgang.pdf (zuletzt geprüft 01.08.2024)

Zinn, B., Tenberg, R., Pittich, D. (2018). Technikdidaktik. Eine interdisziplinäre Bestandsaufnahme. Franz Steiner Verlag. <https://doi.org/10.25162/9783515119429>

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Tobias Bahr

Universität Stuttgart
Institut für Erziehungswissenschaft
Abteilung Berufspädagogik Schwerpunkt
Technikdidaktik (BPT)

bahr@ife.uni-stuttgart.de

Python-Programmierung im Informatikunterricht mit Jupyter Notebooks

Kutzera, J, Witte, L, Schunder, T.

DOI: 10.18420/ibis-03-01-05

Zusammenfassung

In diesem Artikel beleuchten wir die Einsatzmöglichkeiten JupyterLab-basierter Lernumgebungen im Informatikunterricht am Beispiel der Plattform Coding Labs. Zunächst geben wir einen Überblick über die Nutzung von JupyterLab im Bildungsbereich und stellen die Lernplattform Coding Labs vor. Anschließend beschreiben wir spezifische Gestaltungsmöglichkeiten für Lehrmaterialien innerhalb von JupyterLab. Abschließend fassen wir unsere Erkenntnisse aus Praxis-tests an Schulen zusammen. Dabei wurde deutlich, dass besonders die Möglichkeit der Kombination von theoretischen Unterrichtsinhalten mit praktischen Programmieraufgaben in JupyterLab-Umgebungen einen erheblichen Mehrwert bietet. Lehrkräfte hoben zudem positiv hervor, dass die geräteunabhängige Erreichbarkeit es den Schüler*innen erlaubt, Aufgaben und Mitschriften von zu Hause aus aufzurufen und zu bearbeiten.

Einleitung

Jupyter Notebooks stellen eine gute Möglichkeit dar, um Programmierwissen interaktiv und anwendungsorientiert an Lernende zu vermitteln. Dennoch werden sie von Lehrkräften im Schulalltag bislang nur selten verwendet. Im Forschungsprojekt Coding Labs wurde eine JupyterLab-basierte Plattform zum Programmieren Lernen entwickelt, die besonders für den Einsatz an Schulen und Universitäten geeignet ist.

In diesem Artikel stellen wir die Eigenschaften und Nutzungsmöglichkeiten von JupyterLab im Informatikunterricht am Beispiel der Coding Labs-Plattform vor. Ein besonderer Fokus liegt

dabei auf den Möglichkeiten bei der Erstellung und Gestaltung von Lehrmaterialien. So erhalten Lehrkräfte einen praxisnahen Einblick in die Funktionsweise von Jupyter Notebooks und können auf dieser Grundlage bewerten, ob die Nutzung von Coding Labs oder einer anderen JupyterLab-Umgebung eine sinnvolle Ergänzung für ihren Unterricht darstellt.

JupyterLab im Bildungsbereich

JupyterLab ist eine browserbasierte Umgebung zur Anzeige, Erstellung und Bearbeitung interaktiver Dokumente, sogenannter Jupyter Notebooks. Die JupyterLab-Umgebung erlaubt die gleichzeitige Nutzung mehrerer Notebooks in Tabs und enthält eine einfache Dateiverwaltung im Browser. Diese kann zum Hochladen und Organisieren von Notebooks und anderen Inhalten in einer Ordnerstruktur genutzt werden. Zudem unterstützt JupyterLab Login-Systeme, so dass Nutzer*innen individuelle Accounts mit persönlichem Dateiablageort erhalten. Die Umgebung kann direkt auf jupyter.org ohne Login ausprobiert werden¹. Neben Coding Labs² gibt es weitere Anbieter, die JupyterLab samt Login-System zur Verfügung stellen³. Darüber hinaus kann JupyterLab auch lokal gehostet werden. Ein Beispiel für die Ansicht einer JupyterLab-Umgebung befindet sich in Abbildung 1.

Jupyter Notebooks bestehen aus einzelnen Abschnitten, den sogenannten Zellen ("Cells"). Diese beinhalten entweder Programmcode, Medien oder in HTML oder Markdown formatierten Text

¹ <https://jupyter.org/try>

² <https://beta.codinglabs-projekt.de>

³ <https://hszg-jupyter.inf.hszg.de>



Abbildung 1: Die JupyterLab-Umgebung von Coding Labs

(Rule et al., 2018). Dank einer Vielzahl installierbarer Kernels unterstützt JupyterLab unterschiedliche Skript- und Programmiersprachen, wobei Python am häufigsten genutzt wird. Durch die Möglichkeit zur Kombination von Code und Text in den Notebooks eignen diese sich hervorragend für die Vermittlung von Coding Skills. Dabei können sie gleichzeitig als Lehr- und Übungsmaterial sowie als Programmierumgebung und Mitschrift für die Lernenden fungieren.

Dennoch sind Jupyter-basierte Umgebungen im Bildungs- und insbesondere im Schulbereich bislang nur wenig verbreitet. Mögliche Gründe sind hier mangelnde Bekanntheit oder Unsicherheiten hinsichtlich der technischen Implementierung. Auch rechtliche Bedenken, insbesondere bezüglich der DSGVO-Konformität, stellen eine potenzielle Hürde dar (Kutzera et al., 2023). Dennoch wurde der Einsatz von Jupyter Notebooks im Schulbereich bereits mehrfach beschrieben, beispielsweise in einem Praxisbericht zu einem Kurs für Datenanalyse im Informatikunterricht (Podworny et al., 2022). Bovermann (2024) stellt zwei Bibliotheken für die Darstellung von Graphen in Jupyter Notebooks vor und erläutert deren Potenzial für den Informatikunterricht. Eine weitere Ausarbeitung diskutiert den Einsatz von Gamification-Ansätzen in JupyterLab zur Förderung des Lernprozesses (Brocker et al., 2022).

Die Plattform Coding Labs

Coding Labs ist eine kostenlose, browserbasierte Plattform zum Programmieren mit Python, die auf JupyterLab basiert. Nach dem Einloggen steht den Nutzenden eine persönliche JupyterLab-Umgebung samt Dateiverwaltung zur Verfügung. Das Account-System erlaubt es ihnen, von verschiedenen Endgeräten aus in ihrer eigenen Programmierumgebung zu arbeiten, Änderungen zu speichern und ihre Arbeit jederzeit fortzusetzen. Die Plattform wurde speziell für den Einsatz an Schulen und Universitäten sowie für lebenslanges Lernen entwickelt. Sie wird in Deutschland gehostet und ist DSGVO-konform, was insbesondere für den Einsatz in der Schule wichtig ist.

Der Einsatz von JupyterLab im Informatikunterricht

Der Einsatz von JupyterLab-Umgebungen im Informatikunterricht ist besonders dann empfehlenswert, wenn Inhalte mithilfe textbasierter Programmiersprachen vermittelt werden sollen. Bezogen auf das Kompetenzmodell des Bildungsstandards Informatik für die Sekundarstufe II der Gesellschaft für Informatik (GI) e.V. (Röhner 2016),

ist die Nutzung von JupyterLab besonders für die Lehre der Inhaltsbereiche „Algorithmen“ und „Informationen und Daten“ geeignet. Für andere Inhaltsbereiche wie „Sprachen und Automaten“ kann die Umgebung zusätzlich um spezielle Bibliotheken wie ipyturtle erweitert werden.

Erstellung von Lehrmaterial: wieviel Zeit steht der Lehrkraft zur Verfügung?

JupyterLab-basierte Plattformen lassen sich mit wenig Aufwand für den Unterricht vorbereiten. Allerdings erfordert das Gestalten von passendem Lehrmaterial einen gewissen Arbeitsaufwand. Lehrkräfte, die bereits Materialien in Form von Jupyter Notebooks zur Verfügung haben, können diese direkt verwenden. Lehrmaterialien in anderen Formaten (z. B. HTML- oder Textdateien) müssen hingegen manuell in Jupyter Notebooks übertragen und ggf. angepasst werden.

Für Lehrkräfte mit wenig Einarbeitungszeit gibt es online eine wachsende Auswahl von OERs (Open Educational Resources) im Jupyter Notebook-Format. Allerdings sollte Zeit für die Suche, Sichtung und Anpassung des Materials an den Lehrplan und den Kenntnisstand der Schüler*innen eingeplant werden. Auch im Rahmen des Coding Labs-Projekts wurde frei verfügbares und individuell anpassbares Kursmaterial erstellt, welches sich für den Einstieg in Python eignet⁴.

Wer sich intensiver mit der Erstellung von Lehrinhalten für JupyterLab-Plattformen auseinandersetzen möchte, hat durch die Kombination von Markdown- und Python-Zellen in den Jupyter Notebooks eine Vielzahl gestalterischer Möglichkeiten. Diese werden in den folgenden Abschnitten näher erläutert.

Jupyter Notebooks ansehnlich gestalten: Möglichkeiten innerhalb der Markdown-Zellen

Jupyter Notebooks ermöglichen die Formatierung von Inhalten mithilfe von Markdown und vereinfachtem HTML. Dadurch lassen sich u.a. Überschriftenhierarchien, farbliche Hervorhebungen und Strukturelemente wie Trennlinien und visuell abgesetzte Boxen erzeugen. Grafiken können entweder per Drag-and-Drop direkt ins Notebook eingefügt oder über das HTML-Element `` eingebunden werden. Während eingebettete Bilder fest im Notebook-Quelltext eingebunden werden, sodass sie nicht getrennt gespeichert werden müssen, können über `` eingebundene Grafiken flexibel skaliert werden.

⁴ <https://github.com/KMI-KPZ/Coding-Labs-Lerninhalte>

Zusätzlich gibt es weniger bekannte Gestaltungsmöglichkeiten, die jedoch besonders für den Unterrichtseinsatz relevant sind. Mit HTML `<details>` Elementen lassen sich Notebook-Bereiche ausblenden, welche erst durch Anklicken sichtbar werden. So können beispielsweise Aufgabenlösungen verborgen werden. Zur abwechslungsreicheren Gestaltung von Text und Überschriften können Unicode-Emojis verwendet werden. HTML `` Elemente können zum Erzeugen von gesondert formatierten Boxen und Feldern verwendet werden. Die Elemente können farblich hervorgehoben und als Container für Exkurse und Tipps genutzt werden. Außerdem erlaubt das Einbinden von LaTeX-Code die Verwendung mathematischer Formeln. Alle Gestaltungsmöglichkeiten sind in Abbildung 2 noch einmal aufgezeigt und kurz erklärt.

Gestalten von Code-Aufgaben und Tools: Möglichkeiten innerhalb der Python-Zellen

Bei der Erstellung von Programmieraufgaben können die Code-Zellen entweder leer gelassen werden oder bereits Code enthalten, den die Schüler*innen vervollständigen müssen. Aufgabenanweisungen können in Form von Python-Kommentare eingefügt werden, z. B. «`## Ergänze hier deinen Code`». Da Code-Zellen versehentlich gelöscht oder verändert werden können, muss dabei sichergestellt sein, dass der existierende Code wiederhergestellt werden kann. Eine Möglichkeit ist ein über der Code-Zelle platziertes `<details>`-Element mit einer Beschriftung wie «Code zum Wiederherstellen anzeigen», welches den Code nach Anklicken anzeigt, so dass die Schüler*innen den ursprünglichen Code bei Bedarf wieder in die Zelle kopieren können.

Eine weitere wichtige Funktion von Jupyter Notebooks ist hier zudem das Sperren von Zellen. Für jede Zelle (sowohl Python als auch Markdown) kann festgelegt werden, ob sie editierbar sein soll oder nicht. Dies ist insbesondere bei Python-Zellen mit vorgegebenem Übungscode wichtig, sofern der Code nicht verändert werden soll.

JupyterLab enthält zudem umfangreiche Möglichkeiten zur Visualisierung und Interaktion. Für die Visualisierung mathematischer Funktionen und für das Plotten von Ergebnissen aus Datenanalysen hat sich z. B. die Bibliothek `Matplotlib`⁵ bewährt.

Soll ein Spiel oder eine interaktive Anwendung entwickelt werden, eignen sich insbesondere die Bibliotheken `ipycanvas` und `ipywidgets`. `ipycanvas` gewährt Zugriff auf die Canvas-API im Browser und erlaubt das direkte Zeichnen von Pixeln, Linien und Formen. Dabei sind verschiedene Darstellungsformen möglich, von Retro-Pixelgrafiken bis hin zu Skizzenoptik. Zudem können PNG-Grafiken auf dem Canvas eingebettet und animiert werden.

Mit der Library `ipywidgets` steht eine große Auswahl interaktiver Elemente für die Eingabe und Ausgabe zur Verfügung, z.B. Buttons, Sliders, Checkboxen oder Labels. Diese sind mit Python-Funktionen verknüpft, die immer dann ausgeführt werden, wenn auf ein Element geklickt oder ein Slider bewegt wird. Darüber hinaus lassen sich diese Elemente gezielt je nach Bedarf manipulieren, z. B. durch Ein- oder Ausblenden, oder durch Deaktivieren.

Weiterhin ist über die Code-Zellen auch die Einbindung von YouTube-Videos möglich. Diese werden angezeigt, sobald die Code-Zelle ausgeführt wurde. Der entsprechende Code ist in Ab-

⁵ <https://matplotlib.org>



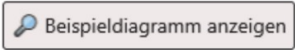
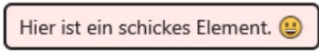
Designelement	Code	Details	Ziel
Unicode-Emojis  Willkommen	Markdown	Direktes Einfügen von Unicode-Übersichtsseiten	Elegante und einfache Methode, um z. B. Überschriften ansprechend zu gestalten
Grafik (PNG, JPG) 	Markdown, HTML	Einbindung durch Drag+Drop oder Verlinkung via HTML <code></code> tag	Grafiken können Teil der Lehrmaterialien aber auch Übungsaufgabe sein, da Schüler*innen Grafiken einfach selbst einfügen können
HTML <code><details></code> Element 	HTML	Verstecken von Hinweisen und Lösungen, die erst nach Klicken auf Button sichtbar sein sollen	Schüler*innen bekommen die Gelegenheit, erst selbst nach der Lösung zu suchen, bevor Hinweise angezeigt werden
HTML <code></code> Element 	HTML	Erstellen von grafisch ansprechenden und flexibel konfigurierbaren Designelementen	Auflockerung des Lehrmaterials, Schaffen von Abwechslung und Strukturierung
Mathematische Formeln $f(x) = \int_{-\infty}^{\infty} e^{-x^2} dx$	Markdown	Markdown unterstützt die Einbindung von LaTeX-Code	Verwendung der mathematischen Notationen, wie sie auch im Mathematikunterricht Einsatz finden

Abbildung 2: Markdown-basierte Gestaltungselemente und ihre Anwendungsbereiche (eigene Darstellung)

bildung 3 zu finden, die eine Übersicht der auf Python basierenden Gestaltungselemente bietet.

Im Projekt Coding Labs wurden verschiedene Libraries speziell für Lehrkräfte entwickelt, die bei der Erstellung eigener Lernmaterialien verwendet und angepasst werden können. Die Libraries sind auf GitHub verfügbar⁶.

So erlaubt die Library CL_canvas die Verwendung von ipycanvas-Funktionen mit weniger Code, so dass der Einstieg in das Handling von Grafik und Buttons mit ipycanvas vereinfacht wird. Zusätzlich wurde die Library sherlock.holmes entwickelt, die es Lehrkräften erlaubt, in ihre Übungen detaillierte Code-Überprüfungen zu integrieren, die von den Schüler*innen selbstständig ausgeführt werden können. So entsteht ein motivierendes Moment, da die Schüler*innen direktes Feedback darüber erhalten, welche Teile ihres Codes bereits funktionieren und welche noch verbessert werden müssen. Die Library drwatson erlaubt das Einbinden von kleinen Multiple-Choice Quizen in die Übungsaufgaben, mit denen die Schüler*innen ihr Wissen selbst testen können. Im GitHub-Repository befinden sich detaillierte Anleitungen zur Verwendung der hier genannten Tools, sowie eine Übersicht zu den in den vorherigen Absätzen beschriebenen gestalterischen Möglichkeiten.

Erfahrungen aus der Praxis

Während der Entwicklung der Coding Labs-Plattform wurde diese von einigen Lehrkräften im Unterricht getestet. Die Erkenntnisse aus diesen Tests sind nicht nur für Coding Labs, sondern für alle JupyterLab-Plattformen relevant.

Allgemein kommen Schüler*innen gut mit JupyterLab-Plattformen und Jupyter Notebooks zurecht. Von den Lehrkräften wird besonders geschätzt, dass die Jupyter Notebooks gleichzeitig als Lehrmaterial und persönliche Mitschrift für die Schüler*innen fungieren. So kann neu erlangtes Theoriewissen in den ausführbaren Code-Zellen sofort praktisch angewendet werden. Zudem können die Kapitel des Lehrmaterials problemlos aufeinander aufbauend gestaltet werden, da Code aus vorherigen Kapiteln für die Schüler*innen einfach aufrufbar bleibt und somit direkt wiederverwendet werden kann.

Zusätzlich ermöglicht der browserbasierte Zugang der Lehrkraft, den Schüler*innen direkt auf der Plattform zu lösende Hausaufgaben aufzugeben. Die bearbeiteten Aufgaben können in der nächsten Unterrichtsstunde ohne weitere Zwischenschritte vom Schul-PC aus aufgerufen werden.

Ein genereller Nachteil der Plattformen ist die Abhängigkeit von einer stabilen Internetverbindung, da JupyterLab-Umgebungen sonst nicht zuverlässig funktionieren. Sofern nicht lokal gehostet, ist ein „Offline“ Arbeiten, z. B. im Rahmen von Kurzkontrollen, nicht möglich.

⁶ <https://github.com/KMI-KPZ/Coding-Labs-Lerninhalte>

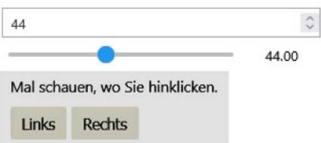
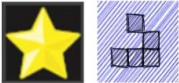

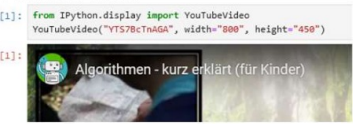
Designelement	Library	Details	Ziel
Interaktive-Elemente 	ipywidgets	Ipywidgets sind mit definierbaren Funktionen verbunden, welche bei Aktivierung der Elemente ausgeführt werden	Slider, Buttons, Radiobuttons und andere interaktive Elemente als Nutzereingabe und -Ausgabe für Pythonprogramme
Bilder und Pixelgrafiken 	ipycanvas	Erstellen, Anzeigen, Ausblenden und Animieren von Grafiken und einzelnen Pixeln	Motivierende Übungsaufgaben, z. B. Erstellen grafisch ansprechender Spiele
Turtle-Grafiken 	ipyturtle	Viele Libraries, z. B. Turtle, existieren in einer Jupyter-kompatiblen Variante	Verwenden von etablierten Lernmitteln für Themen wie Automaten
YouTube-Video 	ipython	Codzeile muss zum Anzeigen einmalig ausgeführt werden	Einbindung von weiterführenden Inhalten in Videoform

Abbildung 3: Python-basierte Gestaltungselemente und ihre Anwendungsbereiche (eigene Darstellung)

Fazit und Ausblick

JupyterLab-Plattformen wie Coding Labs stellen ein geeignetes Werkzeug für das Erlernen der Python-Programmierung und die Lehre der Themen Algorithmen und Datenstrukturen im Informatikunterricht dar. Durch das Account-System erhalten Schüler*innen eine persönliche JupyterLab-Umgebung. Dort sind sowohl Lehrmaterial als auch Übungen und Mitschriften stets verfügbar und können im Unterricht und zu Hause eingesehen und bearbeitet werden.

Das Erstellen von Lehrmaterial für JupyterLab-basierte Plattformen ist mit einem gewissen Zeitaufwand verbunden. Allerdings kann bereits in anderen textbasierten Formaten existierendes Material einfach an die Plattform angepasst werden und durch die Möglichkeiten, die Jupyter Notebooks bieten, z. B. die direkt ausführbaren Code-Zellen, sinnvoll erweitert werden. Einmal für die Nutzung in JupyterLab angepasstes Material kann einfach verteilt, wiederverwendet und auf allen JupyterLab-basierten Plattformen genutzt werden.

Auch jenseits des Informatikunterrichtes ist ein Einsatz von JupyterLab-basierten Plattformen denkbar: So könnten im Mathematikunterricht mithilfe der Bibliothek Matplotlib interaktive Visualisierungen mathematischer Funktionen erstellt werden, bei denen die Schüler*innen Variablen über Schieberegler anpassen können.

Somit ermöglichen JupyterLab-Plattformen eine nahtlose Verknüpfung von theoretischem Wissen und praktischer Anwendung, indem sie beides in einer interaktiven Umgebung vereinen und jederzeit zugänglich machen – sowohl im Informatikunterricht als auch in anderen Fachbereichen.

Förderhinweis

Das Projekt Coding Labs wird im Rahmen der Maßnahme „Initiative Nationale Bildungsplattform“ (Förderkennzeichen 16INB2008 ff.) vom Bundesministerium für Bildung und Forschung (BMBF) gefördert und vom Projektträger VDI-VDE IT betreut. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autor*innen.

Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 28.11.2024.

Bovermann, Klaus (2024): Graphen im Unterricht Mit Hilfe von Python und Jupyter-Notebooks, IBIS Jg. 2 (2024), Nr. 1

Brocker, Annabell; Judel, Sven; Roepke, Rene; Mihailovska, Nikol und Schroeder, Ulrik (2022): Gamifying JupyterLab to Encourage Continuous Interaction in Programming Education, in Kiili et al.: Games and Learning Alliance, GALA 2022, Lecture Notes in Computer Science, vol 13647

Kutzera, Joachim; Wöhlert, Romy; Friedrich, Julia; Römer, Vanita und Richter, Patrick (2023): Future Learning in a Collaborative "Laboratory" Environment - Requirements to Build Up Future Skills, in Camarinhamatos et al.: Collaborative Networks in Digitalization and Society 5.0. PRO-VE 2023. IFIP Advances in Information and Communication Technology, vol 688

Podworny, Susanne; Hüsing, Sven und Schulte, Carsten (2022): A PLACE FOR A DATA SCIENCE INTRODUCTION IN SCHOOL: BETWEEN STATISTICS AND PROGRAMMING, Statistics Education Research Journal (IASE/ISI), 21(2), Article 6

Röhner, Gerhard (2016): Bildungsstandards Informatik für die Sekundarstufe. Beilage zu LOG IN, 36. Jg. (2016), Heft Nr. 183/184.

Rule, Adam; Tabard, Aurélien und Hollan, James D. (2018): Exploration and Explanation in Computational Notebooks, CHI 2018, April 21–26, 2018, Montreal, QC, Canada

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC SA 4.0 zur Verfügung.

Kontakt

Joachim Kutzera, Leonie Witte, Toni Schunder
 Institut für Angewandte Informatik (InfAI) e.V.
 Leipzig
 kutzera@infai.org

GuiPy – eine Python-IDE für die Schule

Röhner, G.

DOI: 10.18420/ibis-03-01-06

Zusammenfassung

Im Informatikunterricht der Sekundarstufe II ist die objektorientierte Modellierung bundesweit verankert. Zur Unterstützung erfolgreicher Lehr- und Lernprozesse stellt die Python-IDE GuiPy hilfreiche Modellierungswerkzeuge bereit, mit denen grundlegende objektorientierte Konzepte veranschaulicht und erarbeitet werden können. Mit dem Struktogrammeditor können Schülerinnen und Schüler Algorithmen modellieren. Der GUI-Designer gibt ihnen eine leicht nutzbare Möglichkeit, ihre eigenen Projekte mit einer modernen und vorzeigbaren grafischen Benutzeroberfläche zu versehen. Im Artikel werden neben grundsätzlichen Anmerkungen zu Python die Modellierungswerkzeuge von GuiPy vorgestellt.

Python in der Schule

Mit einer kurzen Internetrecherche findet man zahlreiche Entwicklungsumgebungen für Python. Daher stellt sich natürlich die Frage, wieso eine weitere Python-IDE (engl. Integrated Development Environment) überhaupt sinnvoll ist. Diese Frage soll in diesem Artikel beantwortet werden. Im Kern geht es darum, dass eine IDE für die Schule Lehr- und Lernprozesse so gut wie möglich unterstützen soll. Der Schwerpunkt liegt also nicht bei professionellen Ansprüchen, sondern bei didaktischen Funktionalitäten. Diese werden im Kontext von Modellierungswerkzeugen dargestellt und unterstützen somit Informatiklehrkräfte, die GuiPy als schulgeeignete Python-IDE einsetzen.

Für Programmieranfänger sind blockbasierte Sprachen und IDEs wie *Scratch*, *Snap!* oder *App Inventor* sehr gut geeignet, da sie damit erfolgreich kreative Animationen, Geschichten und Spiele realisieren können. Blöcke wählt man einfach aus einer Palette aus, das Vokabular einer Computersprache muss man nicht kennen, sie basieren auf Wiedererkennung statt auf Erinnerung. Blöcke helfen Lernenden, Code ohne grundlegende Fehler zusammenzustellen, denn zwei inkompatible Konzepte haben keine verbindenden Teile, syntaktische Fehler werden somit vermieden (Weintrop&Grover, 2020). Spätestens zu Beginn der Oberstufe stellt sich die Frage, ob man zu einer textbasierten Sprache wechselt. Sicherlich ist es so, dass sich Schüle-

rinnen und Schülern mit dem Wechsel zu einer textbasierten Programmiersprache eine neue Welt erschließt und ihre Kompetenzen so erweitert, dass sie auf Programmieranforderungen in Ausbildung oder Studium besser vorbereitet sind.

Doch welche textbasierte Sprache soll es sein? In den letzten zwanzig Jahren wurden bevorzugt Delphi und Java in der Oberstufe und im Abitur eingesetzt. Delphi bzw. Lazarus als IDEs für die von Niklaus Wirth entwickelte Lehrsprache Pascal werden in der universitären Ausbildung nicht verwendet. Das stellt junge Lehrkräfte, die an eine Schule mit Delphi als Programmiersprache kommen, vor erhebliche Herausforderungen. Java wird an Universitäten häufig gelehrt. Allerdings ist Java syntaktisch überladen, im Wesentlichen objektorientiert und macht es den Anfängerinnen und Anfängern oft unnötig schwer, einfache Dinge wie z.B. eine Texteingabe zu erledigen. Python hingegen hat eine einfache und gut lesbare Syntax, kann objektorientiert und prozedural genutzt werden, erfordert weniger Code als z. B. Java und Programmbausteine können ganz leicht interaktiv getestet werden (Bovermann, 2024). Daher ist Python sehr beliebt und lässt sich vielfältig sowie bereits in jüngeren Jahrgangsstufen gut einsetzen. (Jugendwettbewerb Informatik, 2025).

Man kann zwar auf der Konsole in Python programmieren, aber der Einsatz einer integrierten Entwicklungsumgebung (IDE) ist empfehlenswert, weil diese vielfältige Unterstützung beim Programmieren gibt. Die existierenden Python-IDEs haben einen klaren Fokus auf dem Schreiben und Debuggen von Python-Programmen, also beim Implementieren. Das Modellieren wird hingegen nicht unterstützt. Da die Bildungsstandards Informatik für die Sekundarstufe II der GI (GI, 2016) den Prozessbereich „Modellieren und Implementieren“ ausweisen, sollten für die Unterrichtspraxis auch geeignete Modellierungswerkzeuge zur Verfügung stehen. Hier setzt die grundsätzliche Idee von GuiPy an. Es ist eine Python-IDE für die Schule, die bei der Objektorientierung neben dem Implementieren auch das Modellieren durch vielfältige Werkzeuge unterstützt.

Klasseneditor

Mit dem Klasseneditor stellt GuiPy ein Werkzeug zur Verfügung, das die Schülerinnen und Schüler beim Modellieren einer Klasse hilft und ihnen die konkrete Implementierung einer Klasse abnimmt. Das entlastet von syntaktischen Details und konzentriert die Modellierung auf die Abstraktionsprozesse, welche für die Modellierung essentiell sind, nämlich die problemadäquate Auswahl und Gestaltung von Attributen und Methoden.

Auf der Registerkarte *Attribute* lässt sich die Sichtbarkeit für ein Attribut auswählen. Ist eine get- oder set-Methode ausgewählt, so wird beim Anlegen eines Attributs die jeweilige Methode automatisch angelegt. In der Konfiguration des Klassenmodellierers wählt man die gewünschte Voreinstellung aus. Beim Modellieren muss man darüber nachdenken, welche get- bzw. set-Methoden gebraucht werden und sinnvoll sind. Den Kilometerstand z. B. darf man zur Vermeidung einer Tachomanipulation nicht mit einer Set-Methode verändern können, das darf nur die Methode `fahren(Strecke: float)`.

Python ist eine Programmiersprache mit dynamischer Typisierung, d. h. für Variablen, Attribute oder Parameter müssen keine Datentypen angegeben werden. Trotzdem stehen im Klassenmodellierer Auswahlfelder für Datentypen bereit, denn es soll einerseits möglich sein, Klassen gemäß UML-Standard zu modellieren und andererseits sind Programme und Methoden einfacher zu lesen, zu verstehen und zu debuggen, wenn Datentypen angegeben sind. Wer

Typisierung nicht möchte, kann in der Konfiguration die Auswahl von Datentypen verbergen.

Ab Python 3.5 stehen mit PEP 484 (engl. Python Enhancement Proposal, van Rossum, 2014) die sogenannten *Type Hints* zur Verfügung, welche statische Typisierung durch eine Syntaxerweiterung ermöglicht. Hier ist beispielsweise eine einfache Funktion, bei der der Datentyp für den Parameter und Rückgabebetyp deklariert ist:

```
def toString(num: int) → str:
    return str(num)
```

Trotz dieser Typisierung findet zur Laufzeit keine Typprüfung statt. Dafür sind zusätzliche Werkzeuge wie z. B. *mypy* vorgesehen. In der Konfiguration lässt sich *mypy* als Werkzeug einrichten.

In der linken Strukturansicht des Klasseneditors wählt man Attribute oder Methoden zur Bearbeitung aus. Per Drag&Drop lässt sich auch die Reihenfolge ändern, was sich auch auf den im Hintergrund erzeugten Quelltext auswirkt. Mit *Übernehmen* werden Änderungen wirksam, *Neu* schafft Platz für ein neues Attribut oder eine neue Methode.

Parallel zur Modellierung wird im Hintergrund der Pythoncode für die modellierte Klasse erstellt. Alle get- und set-Methoden, sowie der Konstruktor sind am Ende fertig implementiert. Die anderen Methoden müssen später im Editor implementiert werden. Schließt man den Klasseneditor, so wird der erzeugte Pythoncode ausgeführt. Hat man den Klasseneditor von einem UML-Fenster aus aufgerufen, so wird als

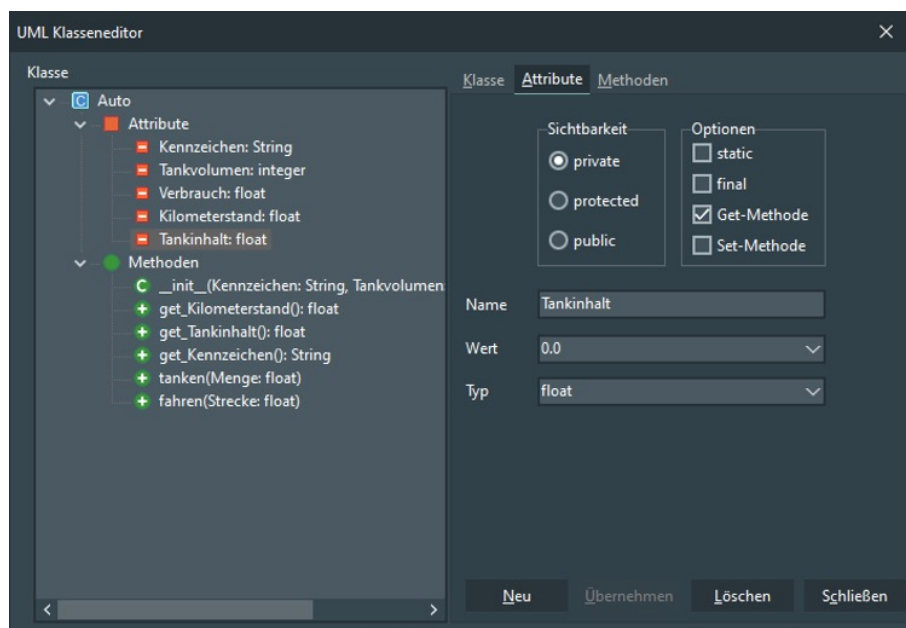


Abbildung 1: Der Klasseneditor mit geöffneter Registerkarte Attribute

Ergebnis das Klassendiagramm dargestellt, ansonsten der erzeugte Pythoncode in einem Editorfenster.

UML-Fenster

Klassen und Klassenbeziehungen im UML-Fenster

Ein UML-Fenster stellt ein oder mehrere Klassen als Klassendiagramm grafisch dar. GuiPy erkennt automatisch Assoziationen zwischen den Klassen, sofern statische Typisierung benutzt wird. Im Beispiel hat die Klasse `Person` das Attribut `Konten` mit dem Typ `list[Konto]`. Daraus ermittelt GuiPy, dass eine Aggregation zur Klasse `Konto` besteht. Ohne statische Typisierung modelliert man Assoziationen über den Befehl *Verbinden mit* im Kontextmenü einer Klasse. Nachträglich lassen sich Assoziationen per Doppelklick editieren, so dass man die Art einer Assoziation ändern, sowie zusätzliche Angaben wie Rollen oder Multiplizitäten angeben kann.

Im Kontextmenü einer Klasse stehen einige Optionen zur Einstellung der Anzeige von Sichtbarkeit, Parameter und Datentypen bereit. Dies ist für die bessere Übersicht bei einem Diagramm mit vielen Klassen oder zur Gestaltung eines Aufgabenblatts hilfreich.

Hat man mehrere Klassen in jeweils eigenen Dateien modelliert, so verwendet man den Befehl *Diagramm aus geöffneten Dateien* um ein gemeinsames Klassendiagramm für alle geöffneten py-Dateien zu erzeugen. Alternativ dazu

gibt es noch den Befehl *Öffne Ordner*, der aus allen Klassendateien eines Ordners ein Klassendiagramm erstellt. Für ein Pythonprojekt, das man beispielsweise im Internet gefunden hat, kann man so ganz einfach ein Klassendiagramm erstellen und damit einen Überblick über die Projektstruktur erhalten.

Auf Basis der Klassenbeziehungen lässt sich automatisch ein passendes Layout berechnen. Wählt man mehrere Klassen aus, so richtet man diese über das Kontextmenü zueinander aus. Für die Verwendung in einem Arbeitsblatt, einer Dokumentation, einer Lernumgebung oder Webseite speichert man das fertige Klassendiagramm als skalierbare SVG-Grafik ab.

In einem UML-Fenster können Schülerinnen und Schüler somit ein Klassendiagramm aus mehreren Klassen und Klassenbeziehungen modellieren. Das hat aus didaktischer Perspektive den Vorteil, dass sie sich selbst aktiv handelnd mit Klassen und Klassenbeziehungen auseinandersetzen. Es geht nicht nur um vorgegebene Klassendiagramme, sondern sie können auch eigene Klassendiagramme leicht erstellen. Gegenüber auf Papier modellierten Klassen hat das UML-Fenster als Werkzeug den Vorteil, dass Entwürfe einfach bearbeitet werden können und man professionelle Ergebnisse produziert. Durch die Integration des UML-Fensters in die GuiPy-IDE können Schülerinnen und Schüler sowie Lehrkräfte innerhalb einer Umgebung mit Klassendiagrammen und Pythoncode arbeiten. Da kein externer UML-Editor notwendig ist, wird der Unterricht entlastet.

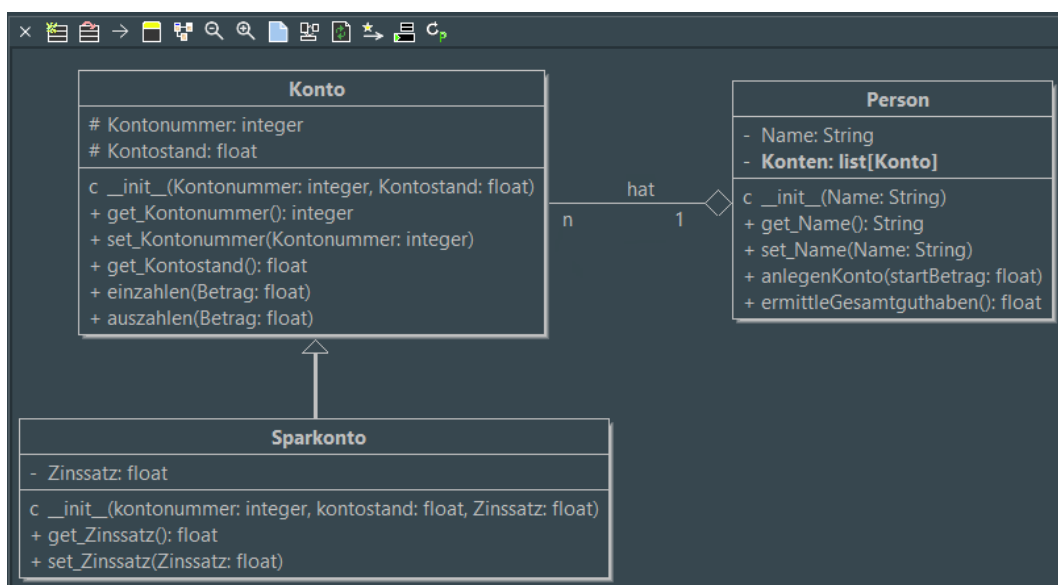


Abbildung 2: Klassen und Klassenbeziehungen im UML-Fenster

Objekte im UML-Fenster


Aus der Literatur ist bekannt, dass beim objektorientierten Ansatz Lernende Schwierigkeiten haben zwischen Klassen und Objekten zu unterscheiden (Eckerdal, 2004). Da dies aber grundlegende Konzepte des OO-Ansatzes sind, muss man didaktische Ansätze entwickeln, um diese Schwierigkeiten zu meistern. Mit sprachlichen Erklärungen alleine, lässt sich das Problem nicht lösen. Auch die Umsetzung auf der Programmierenebene reicht nicht aus. Beispielsweise hilft die Erstellung eines Objekts der Klasse `Auto` mittels `meinAuto = Auto('DA RR 1050', 60, 7.3)` mit den Parameterwerten für Kennzeichen, Verbrauch und Tankkapazität nicht wirklich, das Objektkonzept zu verstehen.

Erst das Zusammenspiel von Erklärungen, Visualisierungen und selbstständige Auseinandersetzung mit konkreten Beispielen hilft, die Hürden bei der Unterscheidung von Klassen und Objekten zu überwinden. Aus diesem Grunde bieten didaktisch gestaltete Werkzeuge wie `GuiPy`, `JavaEditor` bzw. `BlueJ` Visualisierungen anhand eigener Beispiele an. Im Kontextmenü einer Klasse ruft man den Konstruktor auf, um ein Objekt zu erzeugen. Eventuell erforderliche Parameterwerte werden abgefragt. Erstellt man zu einer Klasse mindestens zwei Objekte, so lassen sich im kombinierten Klassen- und Objektdiagramm gut verständlich Unterschiede und Gemeinsamkeiten zwischen Klassen und Objekten in einer Unterrichtssituation besprechen.

Wir haben eine Klasse `Auto`, aber zwei `Auto`-Objekte und weitere könnten erstellt werden. Bei den Objekten werden nur die Attribute und deren Werte angegeben, die sich zwischen den Objekten unterscheiden. Bei der Klasse `Auto`

werden stattdessen die Attribute und deren Datentypen sowie der Konstruktor und die Methoden angegeben. Man erkennt, dass die Objekte auf Basis der Klasse `Auto` erstellt sind.

Aus dem Kontextmenü der erstellten Objekte kann man die Methoden der Klasse aufrufen. Da der Klasseneditor get- und set-Methoden automatisch implementiert, lassen sich diese interaktiv testen, ohne von der Modellierungsebene zur Programmierenebene wechseln zu müssen.

Die Methoden `tanken(Menge: float)` und `fahren(Strecke: float)` müssen im Editorfenster der Klasse `Auto` implementiert werden. Der Test der Implementierung erfolgt im UML-Fenster. Beim Wechsel in das UML-Fenster wird die Klasse mit violettem Kopf als ungültig dargestellt, weil Quellcodeänderungen vorliegen und diese noch nicht vom Python-Interpreter verarbeitet wurden. Durch das Ausführen des UML-Fensters über das grüne Dreieck wird die Klasse gültig, aber die Objekte gehen verloren, weil sie nicht mehr zur geänderten Klasse gehören. Zur Entlastung des Unterrichts öffnet das Icon  ein Fenster, das die interaktiven Aktionen protokolliert. Über dessen Ausführen-Symbol lassen sich die Objekte schnell wieder herstellen. Nach jedem Aufruf einer Methode wird der aktuelle Objektzustand dargestellt, eine Inspektion wie in `BlueJ` ist nicht erforderlich. So kann man die Wirkung von Methodenaufrufen auf den Objektzustand leicht erkennen und die Implementierung der Methoden testen.

Unterrichtliche Erfahrungen zeigen, dass mit diesem Ansatz auch schwächeren Schülerinnen und Schülern der Zugang zu den Konzepten Klasse und Objekt gut gelingt. Insbesondere werden sie auch durch das Protokollfenster beim Übergang vom Modellieren zum Imple-

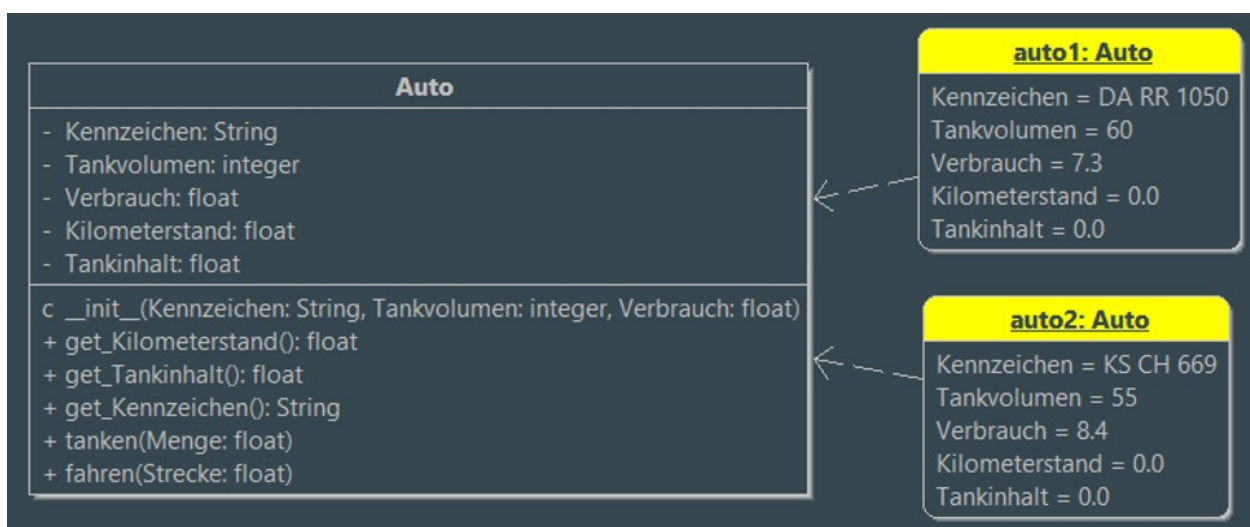


Abbildung 3: Kombiniertes Klassen- und Objektdiagramm zur Klärung des Unterschieds zwischen Klasse und Objekt

mentieren unterstützt. Bei interaktiven Aufrufen von Konstruktoren und Methoden wählt man wie bei blockbasierten Sprachen nur aus bzw. füllt eine Tabelle für die Parameterwerte aus, benötigt aber keine Syntaxkenntnisse. Im Protokollfenster können sie dann sehen, wie Konstruktor- und Methodenaufrufe in Python implementiert werden.

Dynamische Datenstrukturen im UML-Fenster

Wenn im Unterricht dynamische Datenstrukturen wie Listen oder Bäume behandelt werden, so bietet das UML-Fenster eine ausgezeichnete Darstellung, in der alle Referenzen zwischen den Objekten sichtbar sind. Im Beispiel wird das für eine einfach verkettete Liste mit drei Listenelementen gezeigt, welche mit `verketteteListe1 = VerketteteListe([4,13,2])` erzeugt wurde.

Jede Objektreferenz wird als Pfeil dargestellt, wodurch die lineare Struktur der Liste deutlich wird. Das Attribut `Anfang` der Klasse `VerketteteListe` verweist auf den Kopf der Liste und mit `Nächster = None` ist `knoten3` das Ende der Liste. Bei einer doppelt verketteten Liste, werden auch die Pfeile rückwärts dargestellt. In dieser Umgebung können die Schülerinnen und Schüler einzelne Methoden der Klasse `VerketteteListe` implementieren und testen. Nach z. B. dem Programmieren der Methode `löschen(Daten)` und dem erneuten Erstellen der Liste mit Hilfe des Protokollfensters kann man beim Objekt `verketteteListe1` die Methode `löschen(13)` aufrufen. Das sich ergebende Objektdiagramm zeigt, ob korrekt gelöscht wurde oder nicht. Das Ergebnis muss allerdings interpretiert und die Implementierung möglicherweise überarbeitet werden.

Aber warum sollte man die lineare Liste als dynamische Datenstruktur überhaupt behandeln, wenn Python den in der Sprache eingebauten Datentyp `Liste` hat? Die linear verkettete Liste kann man als einfaches Einführungsbeispiel für eine dynamische Datenstruktur ansehen, die mit Hilfe von Objektreferenzen realisiert wird. Damit werden grundlegende Kompetenzen für die Nutzung und Anwendung des Referenzkonzepts erworben. Auf dieser Basis können weitere Datenstrukturen wie zum Beispiel Stapel, Warteschlange, binärer Suchbaum oder Graph verstanden und realisiert werden.

Mit der Darstellung einer dynamischen Datenstruktur als Objektdiagramm und das einfache Testen einer Methodenimplementierung durch Aufruf im UML-Fenster mit anschließender visueller Überprüfung des Ergebnisses wird für die Schülerinnen und Schüler eine sehr große Zugänglichkeit zum Thema dynamische Datenstrukturen geschaffen. Sie können selbständig implementieren, Testergebnisse reflektieren, Fehler finden sowie korrigieren und damit die Kompetenzen erwerben, die für die Nutzung von Referenzstrukturen nötig sind.

Die didaktische Konzeption des UML-Fensters geht weit über das hinaus, was BlueJ zu bieten hat. Es steht ein vollwertiger Editor für Klassendiagramme zur Verfügung, mit dem alle Klassenbeziehungen modelliert werden können. Objekte werden in einem gemeinsam Klassen- und Objektdiagramm dargestellt, wobei alle Attributwerte direkt einzusehen sind. Die erzeugte Objektstruktur wird aus den Beziehungen zwischen den Objekten vollständig repräsentiert. Die hohe Transparenz und einfachen Zugänglichkeit ermöglicht den Schülerinnen und Schüler einen erfolgreichen Einstieg in die objektorientierte Modellierung und Programmierung.

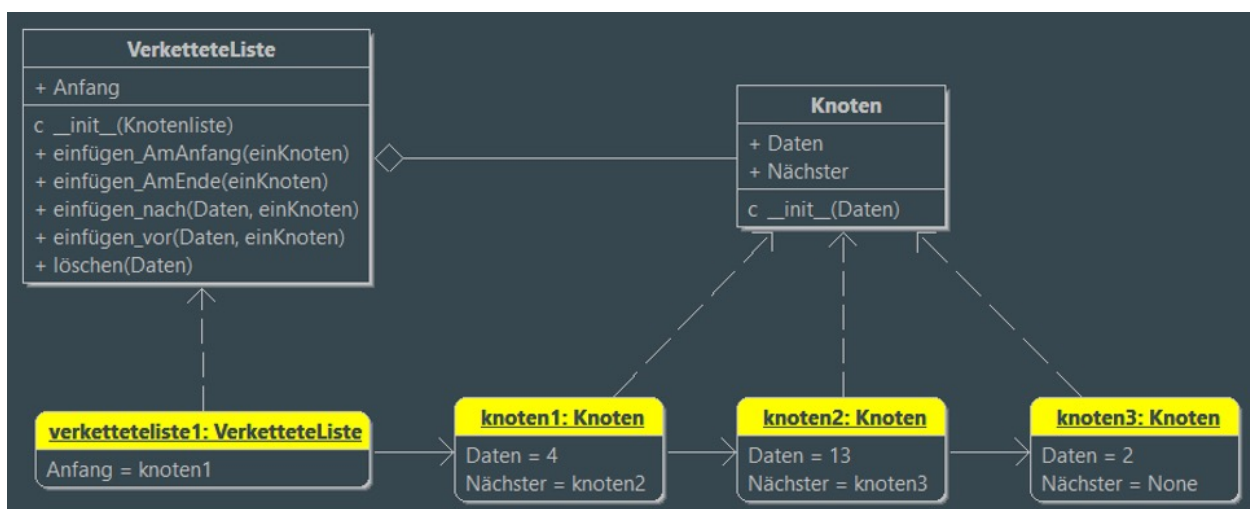


Abbildung 4: Darstellung einer 3-elementigen Liste als Objektdiagramm

Struktogrammeditor

Indem man dem Implementieren eine Modellierungsphase voranstellt, kann man auf einer abstrakten und dem eigenen Denken entsprechenden Ebene arbeiten. Insbesondere muss man beim Modellieren nicht auf syntaktische Details bzw. Möglichkeiten einer Programmiersprache Rücksicht nehmen. In einem Top-Down- oder Bottom-Up-Ansatz kann ein Algorithmus zur Lösung eines Problems schrittweise entwickelt werden. Der Einsatz eines Struktogrammeditors unterstützt die Lernenden beim Modellieren, weil die algorithmischen Grundstrukturen aus einer Palette ausgewählt und Änderungen im Entwurf leicht durchzuführen sind. Sie erhalten im Gegensatz zu einer handschriftlichen Herangehensweise ein vorzeigbares und bearbeitbares digitales Ergebnis, wodurch auch ihre Kompetenzen beim Umgang mit digitalen Werkzeugen erweitert werden.

Der Algorithmus zum Fahren einer Strecke könnte in einem ersten Entwurf so aussehen:

Er ist sprachlichen und gedanklichen Strukturen angelehnt, welche beim Nachdenken über das Fahren in den Sinn kommen können. Es ist

Im nächsten Schritt wird der modellierte Algorithmus verfeinert. Die bisherigen Formulierungen werden mit den Attributen Tankinhalt, Kilometerstand und Verbrauch der Klasse Auto in Beziehung gesetzt und in Berechnungen überführt. Umgangssprachliche und mehrdeutige Formulierungen wie z.B. „Reicht dieser Verbrauch?“ werden in eindeutige Anweisungen bzw. Ausdrücke überführt.

In dieser Form kann der Algorithmus mit dem entsprechenden Befehl aus dem Kontextmenü in eine Python-Funktion übertragen werden. Der erzeugte Quellcode bildet die algorithmische Struktur des Algorithmus richtig in Python-Kontrollstrukturen ab, muss in der Regel aber syntaktisch überarbeitet werden. Für Schülerinnen und Schüler, die sich beim Programmieren schwertun, ist die automatische Übertragung eines Algorithmus in Programmcode eine gute Hilfe. Platziert man den Cursor in die Kopfzeile einer Pythonfunktion oder markiert man einige Pythonzeilen im Codeeditor, so kann man umgekehrt auch Pythoncode in ein Struktogramm übertragen.

Der Struktogrammeditor stellt auf einer Symbolleiste alle Struktogrammelemente bereit.

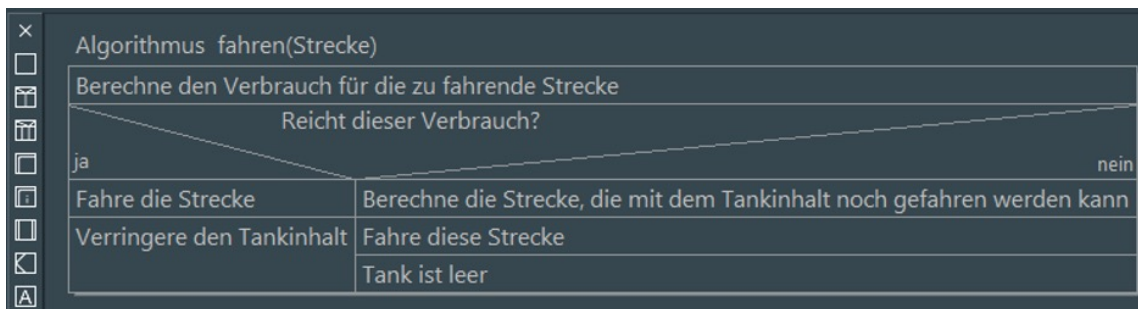


Abbildung 5: Umgangssprachlicher Entwurf für den fahren-Algorithmus

auch schon eine wesentliche Modellierungsentscheidung getroffen worden, nämlich wie mit Strecken umgegangen wird, die die aktuelle Reichweite überschreiten: gar nicht erst losfahren oder fahren bis der Tank leer ist. Auf dieser Ebene ist es aber noch nicht nötig, sich Gedanken über einzelne Berechnungen zu machen.

Diese können ähnlich wie in Scratch per Drag&Drop benutzt werden, um ein Struktogramm zu erstellen. Durch Abziehen nach unten lassen sich ganze Blöcke ablösen und dann wieder neu zusammensetzen. Nach einem Doppelklick gibt man Text direkt in ein Strukturelement ein. Die Return-Taste erzeugt mehrzeiligen

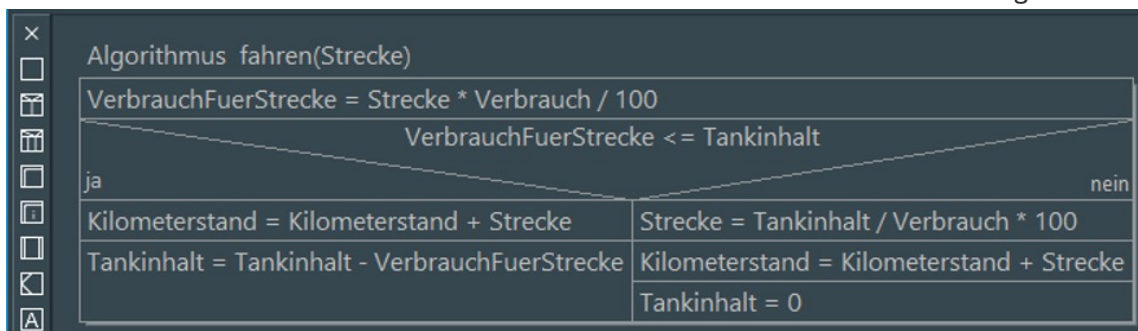


Abbildung 6: Implementierbarer Entwurf des fahren-Algorithmus

Text, was bei Fallunterscheidungen oft sinnvoll ist. In der Konfiguration von GuiPy gibt es eine Seite für den Struktogrammeditor, auf der unter anderem Texte angepasst werden können.

Über das Kontextmenü eines fertigen Struktogramms schaltet man in den Puzzlmodus um. Im Puzzlmodus zerlegt eine Lehrkraft das Struktogramm in einzelne Puzzleteile. Der weitere Befehl *Speichere Puzzle Dateien* erzeugt Struktogrammpuzzles der Schwierigkeitsstufen *leicht, mittel, schwer* und *sehr schwer*. Schülerinnen und Schüler können sich eine für sie passende Schwierigkeitsstufe auswählen und ihre Lösung mit dem Puzzle-Symbol überprüfen.

Sequenzdiagramme

Mit einem Sequenzdiagramm wird der zeitliche Ablauf einer Interaktion modelliert. Jedes beteiligte Objekt erhält eine abwärts verlaufende gestrichelte Lebenslinie als Zeitachse. Die Kommunikation der Objekte wird durch horizontale Pfeile zwischen den Lebenslinien dargestellt, welche jeweils eine Nachricht übermitteln. Handelt es sich um Objekte als Exemplare von Klassen, so entsprechen Methodenaufrufe den Nachrichten.

Das Beispiel zeigt das Sequenzdiagramm für eine mögliche Interaktion im UML-Fenster, bei der zunächst ein Auto-Objekt erzeugt, erfolglos gefahren, betankt und dann erfolgreich gefahren wird. Das sind die zugehörigen Methodenaufrufe:

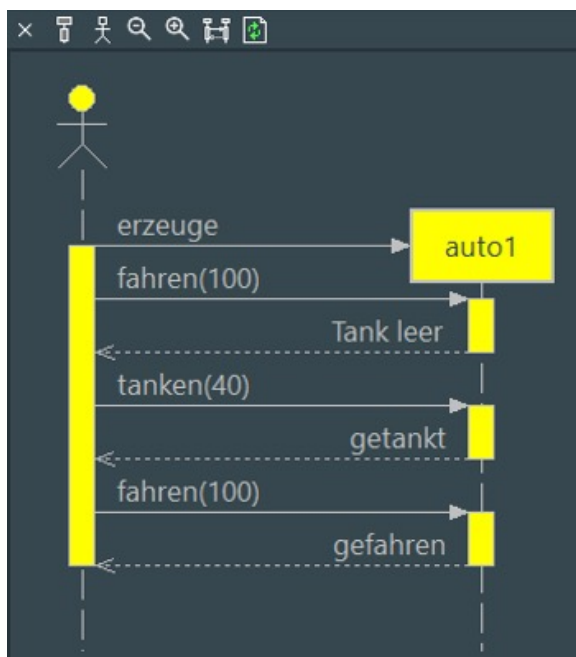


Abbildung 7: Sequenzdiagramm zum Erzeugen, Tanken und Fahren eines Autos

```

auto1 = Auto('DA RR 1023', 60, 7.3)
auto1.fahren(100)
auto1.tanken(40)
auto1.fahren(100)
  
```

Lebenslinien erzeugt man mit den betreffenden Symbolen der Symbolleiste. Über das Kontextmenü einer Lebenslinie erstellt man Nachrichten zu anderen Objekten.

Sequenzdiagramme können Kommunikationsvorgänge, wie sie beispielsweise bei der Client-Server-Kommunikation auftreten, sehr gut darstellen. Für die objektorientierte Modellierung in der Schule haben sie keine so große Relevanz, weswegen sie in vielen Bundesländern nicht zum Curriculum gehören.

GUI-Designer

Scratch verfügt eine Bühne und Greenfoot über eine Welt, in der man mit vorgegebenen oder abgeleiteten Grafikobjekten arbeiten kann. Das hat den Vorteil, dass die Entwicklung interaktiver Animationen und Simulationen vielfältig unterstützt wird, von der Gestaltung der Bühne bzw. Welt bis hin zur interaktiven Ablaufsteuerung. GuiPy verfolgt einen anderen Ansatz, bei dem die Beschränkungen durch eine Bühne bzw. Welt aufgehoben und durch eine vollkommen freie Gestaltung der Benutzeroberfläche ersetzt wird. Dieser Ansatz wird durch den GUI-Designer ermöglicht, mit dem man interaktiv Widgets aus Paletten in ein GUI-Formular platziert und deren Attribute im Objektinspektor konfiguriert, wobei synchron dazu der entsprechende Pythoncode automatisch erzeugt wird. Grafische Benutzeroberflächen direkt durch Programmierung zu erzeugen, wie es beispielsweise bei *inf-schule.de* vorgeschlagen wird (Becker, 2023), erfordert sehr viel Detailwissen über Attribute und Methoden der Widgets und ist somit ein sehr mühsamer und nicht zielführender Ansatz.

Python verfügt von Hause aus über die beiden Grafikbibliotheken *Tkinter* und *TTK*. Sie stehen in Python direkt zur Verfügung. Die modernere Grafikbibliothek *Qt* muss zusätzlich installiert werden, wofür GuiPy das Werkzeug *Pakete mit PIP installieren* bereitstellt. Für *Qt* benötigt man das Paket *PyQt6*. Weitere Installationshinweise finden Sie auf der GuiPy-Homepage *gui.py.de* (Röhner, 2024b). Auf der Registerkarte *Programm* wählen Sie *Tkinter/TTK* oder *Qt* aus, wenn Sie eine GUI-Anwendung erstellen wollen. *Qt* als moderne Bibliothek mit konsistenter

Schnittstelle für die Ein- und Ausgabe wird empfohlen.

Die wichtigsten Widgets stehen auf der Registerkarte *Qt Base* bereit. Per Drag&Drop platziert man sie auf dem GUI-Formular. Die Attribute eines Widgets bearbeitet man im Objektinspektor. Eine didaktische Besonderheit besteht darin, dass er nicht wie bei professionellen IDEs üblich die komplette Liste aller Attribute anzeigt, sondern zunächst nur die wichtigsten Attribute, sodass die Schülerinnen und Schüler deutlich entlastet werden. Bei Bedarf kann man sich mehr bzw. alle Attribute anzeigen lassen. Platziert man mehrere Buttons auf einem Formular, so werden diese fortlaufend durchnummeriert, z.B. `pushButton1`, `pushButton2`, usw. Bei mehr als drei bis vier Buttons wird es dann im Quellcode unübersichtlich, weil die Nummerierung keinen Bezug zur Bedeutung eines Buttons hat. Da die Schülerinnen und Schüler jedem Button eine Beschriftung geben, wird dieses Problem didaktisch so gelöst, dass der Name eines Buttons aus der eingegebenen Beschriftung abgeleitet wird.

Der visuelle Entwurf eines GUI-Formulars mit Widgets wird automatisch als Pythoncode in die Methode `create_widgets` übertragen, wobei statt eines Layout-Managers das absolute Layout benutzt wird. Standardmäßig sind `create_widgets` und der Konstruktor eingeklappt, so dass der Programmcode für die Schülerinnen und Schüler leicht überschaubar bleibt und oft nur aus den Ereignismethoden besteht.

Diesem Beispiel einer Aufzug-Simulation liegt als Fachkonzept die Klasse `Aufzug` zugrunde. Sie muss importiert und im Konstruktor ein Aufzug-Objekt erzeugt werden, mit dem dann im Weiteren gearbeitet wird. Die Ein- und Ausgabe erfolgt über die Widgets, die Verarbeitung im Objekt `einAufzug` (EVA-Prinzip). Widgets stellen Methoden für die Ein-/Ausgabe bereit. Die wenigen relevanten Methoden sind auf der GuiPy-Homepage gui.py.de dokumentiert. Dort findet man auch hilfreiche Anwendungsbeispiele. Gemäß dem Model-View-Controller-Konzept aktualisiert die Ereignismethode `bWeiterfahren_clicked` als Controller das Modell `einAufzug` und zeigt anschließend über die Widgets im GUI-Formular als View mit `setText`, `setValue` und `setChecked` den aktuellen Zustand des Aufzugs.

Debugging

Beim Programmieren der Ereignismethoden können leicht Fehler gemacht werden. Beispielsweise fehlerhafte Groß-/Kleinschreibung bei Variablen und Attributen, falsche Bezeichnung von Widgets, Aufrufe mit Datentyp `int` statt `String` oder vergessene Klammern (Henning, Michaeli, 2023). Da Python interpretiert und nicht compiliert wird, treten solche Fehler erst während der Programmausführung auf, also typischerweise wenn bei GUI-Programmen eine Ereignisprozedur ausgeführt wird. Die Fehlermeldungen des Python-Interpreters sind in der Regel nicht hilfreich.

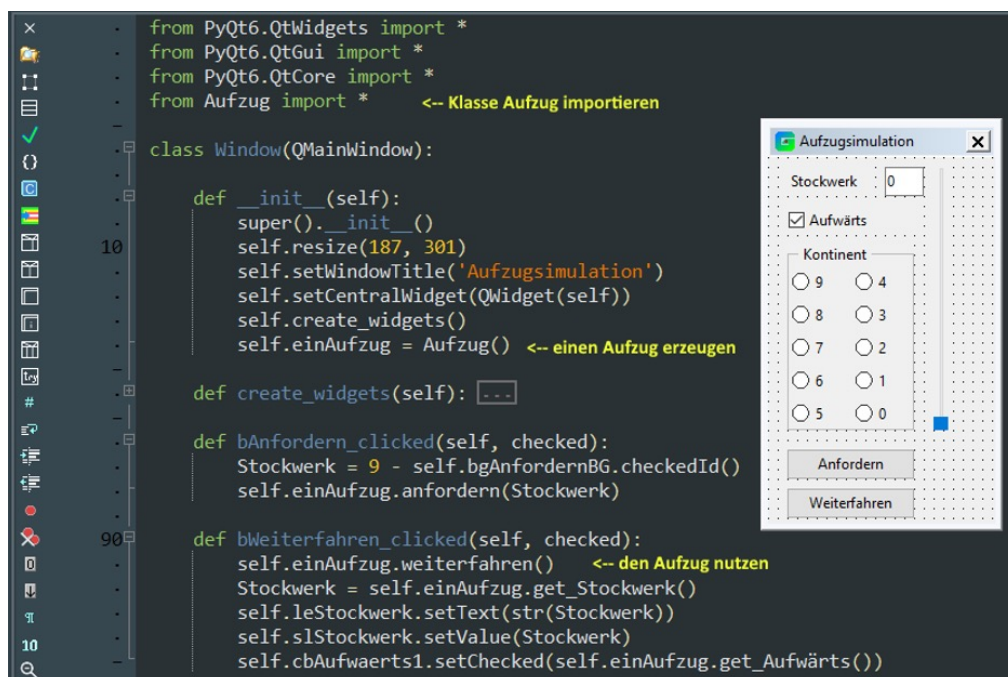


Abbildung 8: Ein- und Ausgabe in einem GUI-Formular

Man benötigt eine funktionierende Debuggingstrategie, um Fehler in GUI-Programmen zu finden und zu beheben. Dazu setzt man bei einer nicht funktionierenden Ereignismethode einen Haltepunkt auf die erste Anweisung nach dem Methodenkopf und führt dann das Programm im Debuggingmodus aus. Über die GUI-Oberfläche klickt man auf den Button, dem die Ereignismethode zugeordnet ist. Der Debugger hält dann das Programm am Haltepunkt an. Jetzt führt man schrittweise das Programm aus. Wird an der nächsten Anweisungszeile angehalten, war die vorherige fehlerfrei. Ansonsten enthält die Zeile, an der nicht mehr angehalten wurde, einen Fehler. Hier sollte man nach den typischen oben genannten Fehler Ausschau halten.

Python und OOP

Die Realisierung des Klassenkonzepts in Python weist einige Schwächen auf, die man beim Programmieren bald bemerkt. Der Zugriff auf ein Attribut benötigt mindestens ein `self`. Setzt man das Geheimnisprinzip konsequent um und gibt allen Attributen die Sichtbarkeit `private` so muss jeder Zugriff auf ein Attribut sogar mit `self.__` eingeleitet werden, was aufwändig ist. Gemäß dem Zen von Python (Peters, 2004) gilt allerdings: *Explicit is better than implicit*. Mit der expliziten Angabe von `self` werden Kollisionen mit implizit angelegten lokalen Variablen oder mit gleichnamigen Parametern vermieden. In Java kann man in solchen Fällen `this` explizit benutzen, ansonsten kann man `this` weglassen.

In Python gibt es trotz *Explicit is better than implicit* keine Möglichkeit, Attribute explizit zu deklarieren. Sie werden implizit angelegt, wenn eine Wertzuweisung stattfindet. Das findet typischerweise im Konstruktor statt, kann aber

auch in einer normalen Methode passieren. Bei explizitem Anlegen würde man die Attribute einer Klasse direkt erkennen können. Außerdem könnte man die Sichtbarkeit festlegen und müsste diese nicht bei jedem Zugriff angeben.

Bei rekursiv definierten Klassen gibt es Probleme mit der Typisierung. In diesem Beispiel weist der Python-Interpreter darauf hin, dass der Typ `Tree` des Parameters `left` nicht definiert ist (van Rossum, 2014).

```
class Tree:
    def __init__(self, left: Tree, right: Tree):
        self.left = left
        self.right = right
```

Man kann dieses Problem durch die Anweisung `from __future__ import annotations` beheben, weswegen GuiPy diese Anweisung bei Bedarf automatisch ergänzt.

KI-Schnittstelle

Künstliche Intelligenz wird das Programmieren auch in der Schule verändern, denn sie hat das Potential jedem Schüler und jeder Schülerin individuelle Rückmeldungen zu geben. Wenn man sich bei OpenAI oder Google Gemini einen API-Schlüssel erzeugt, kann man in GuiPy mit KI-Unterstützung arbeiten. Das Chat-Fenster ist eine in GuiPy integrierte Alternative zu einer browserbasierten KI-Nutzung. Der KI-Assistent steht im Quelltextfenster zur Verfügung. Markiert man Sourcecode, so hat man im Kontextmenü des Editorfenster diese drei Optionen: *Erkläre*, *Fehler beheben* und *Optimieren*.

```
def anfordern(self, Stockwerk: int):
    """
    Verwaltet Aufzugsanforderungen basierend auf der aktuellen Fahrtricht

    Diese Methode fügt eine neue Aufzugsanforderung hinzu, abhängig davon
    Anforderungen in Fahrtrichtung werden sofort hinzugefügt, entgegen de

    Args:
        Stockwerk (int): Das gewünschte Stockwerk der Aufzugsanforderung.
    """
    if self.__Aufwärts:
        if Stockwerk > self.__Stockwerk:
            self.__AnforderungenAktuell.append(Stockwerk)
        else:
            self.__AnforderungenGemerkt.append(Stockwerk)
    else:
```

Abbildung 9: KI-Assistent: Erklärung des markierten Sourcecodes

Sonstiges

GuiPy basiert auf dem open source project PyScripiter, das Kiriakos Vlahos seit 2005 als Python-IDE entwickelt (Vlahos, 2024). Daher verfügt GuiPy über eine Fülle von Funktionen für die Programmierung u. a. Editor mit Syntaxhervorhebung, Code-Faltung, Codevervollständigung, Syntaxprüfung während der Eingabe, integrierter Interpreter und Debugger, Variablenfenster, externe Tools uvm. GuiPy erweitert PyScripiter um die in diesem Artikel dargestellten Modellierungswerkzeuge, welche vom JavaEditor (Röhner, 2024a) übernommen und an die Sprache Python angepasst wurden. Zusätzlich kann in GuiPy die Sichtbarkeit von Menüs, Registerkarten und Symbolen konfiguriert werden. Auch gibt es Unterstützung für die Installation und Konfiguration in einem schulischen Netzwerk.

Ein besonderer Dank gilt Peter Ehrlich, der mit seinen unermüdlichen Rückmeldungen zu Fehlern und Problemen maßgeblich zur Verbesserungen von GuiPy beiträgt.

Quelltexte

Die im Beitrag referenzierten Quelltexte werden auf der ibis-Webseite zur Verfügung gestellt: <https://t1p.de/l25rl>



Literatur

Alle Webseiten/Links wurden zuletzt geprüft am 01.08.2024.

Becker, K. (2023). GUI-Entwicklung mit tkinter. https://inf-schule.de/software/gui/entwicklung_tkinter

Bovermann, K. (2014). Graphen in der Schule mit Python und Jupyter-Notebooks. IBIS Jg. 2, Nr. 1

Eckerdal, A. (2004): On the understanding of Object and Class. <https://www2.it.uu.se/research/publications/reports/2004-058/2004-058-nc.pdf>

Gesellschaft für Informatik, 2016: Bildungsstandards Informatik für die Sekundarstufe II

Hennig, H. & Michaeli, T. (2023). Entwicklung eines Prozessmodells für Diagnose- und Intervention von Lehrkräften beim Debugging, S. 157-166. In: Informatikunterricht zwischen Aktualität und Zeitlosigkeit, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2023

Jugendwettbewerb Informatik (2025). Trainingsaufgaben Python. <https://jwinf.de/contest/training>

Peters, T. (2004). PEP 20 – The Zen of Python. <https://peps.python.org/pep-0020/>

Röhner, G. (2024a). Java-Editor. <https://javaeditor.org>

Röhner, G. (2024b). GuiPy. <https://gui.py.de>

van Rossum, G. (2014). PEP 484 – Type Hints. <https://peps.python.org/pep-0484/>

Vlahos, K. (2024). PyScripiter. <https://github.com/pyscripiter>

Weintrop, D. & Grover, S. (2020). JavaScript, Python, Scratch, or Soemthing Else? Navigating the Busting World of Introductory Programing Languages. In Grover, S.: Computer Science in K-12. An A to Z handbook on teaching programming. S.99-112.

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Gerhard Röhner

groehner@t-online.de

So ein Saftladen: IT-Sicherheit in handlungsorientierten Fallbeispielen

Kokula, R., Kufner, H., Leesch, B., Reinold, K., Scholz, J., Wich, S., Winter, P.

DOI: 10.18420/ibis-03-01-07

Zusammenfassung

IT-Sicherheit ist ein Thema, das immer dann in den öffentlichen Diskurs kommt, wenn Systeme gehackt oder anderweitig gestört werden. Auch in den schulischen Lehrplänen gewinnt es zunehmend an Bedeutung. Der Artikel skizziert eine Annäherung an das Thema in Form von Fallbeispielen auf der Basis des OWASP Juice Shop.

Einleitung

IT-Sicherheit ist ein für Gesellschaft und Wirtschaft, aber auch für den privaten Anwender relevantes Thema, das zunehmend an Bedeutung gewinnt. Die nachfolgende Abbildung stellt wesentliche Risiken zusammen, denen IT-Systeme ausgesetzt sind.

Insbesondere die Verhinderung und Abwehr gezielter Angriffe ist in Zeiten von Cyberkriminalität und Cyberkrieg von globaler Relevanz. So schreibt das Bundesamt für Sicherheit in der Informationstechnik (BSI 2024):

„[...] ganz Deutschland ist aufgerufen, eigene Angriffsflächen zu ermitteln und zu



Abbildung 1: Gefahren für IT-Systeme (Brichzin et al. 2024)

schützen. Das ist in historisch gewachsenen IT-Landschaften eine große Herausforderung, aber notwendig, denn die Angreifer suchen beständig nach neuen Angriffsvektoren.“ (BSI 2024, S. 8)

Zur Vermeidung von erfolgreichen Angriffen ist die Schaffung eines breiten gesellschaftlichen Problembewusstseins ein zentraler Faktor:

„Im alltäglichen Umgang mit IT-Systemen ist Awareness eine elementare Sicherheitsmaßnahme. Das bedeutet zunächst, dass ein Problembewusstsein für Cyber-Sicherheit geschaffen werden muss. Darauf aufbauend kann man eine Verhaltensänderung hin zu sicherem digitalen Umgang erreichen. Security Awareness Maßnahmen sind dann erfolgreich, wenn sie die Zielgruppen befähigen und den einzelnen Menschen für mehr Cyber-Sicherheit motivieren.“ (BSI 2025)

Die allgemeinbildende Relevanz des Themas Informationssicherheit zeigt sich auch darin, dass das Thema Einzug in die aktuellen Lehr- und Bildungspläne findet. Exemplarisch seien

hier die Pläne von Nordrhein-Westfalen, Mecklenburg-Vorpommern und Bayern zitiert:

„Die Schülerinnen und Schüler [...] untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen.“ (QUA-LiS 2024)

„Die Schülerinnen und Schüler [...] konzipieren Maßnahmen zur Realisierung von Datensicherheit für konkrete Anwendungsfälle, insbesondere Zugriffskontrolle.“ (Institut für Qualitätsentwicklung 2019)

„Die Schülerinnen und Schüler [...] analysieren exemplarisch ein Informatiksystem (z. B. Smartphone-App, Smarthome-System, Informatiksystem eines Unternehmens) hinsichtlich der Umsetzung wichtiger Schutzziele der Informationssicherheit und bewerten das Erreichen dieser Ziele. [Sie] beschreiben verschiedene Arten der Gefährdung eines Informatiksystems und analysieren ein mögliches Angriffsszenario. [Sie] erläutern Maßnahmen, die die Informationssicherheit gewährleisten sollen. In diesem Kontext werden ihnen technische und wirtschaftliche Grenzen bewusst. [Sie] erörtern verschiedene Perspektiven einer Fragestellung der Informationssicherheit, z. B. Offenlegung oder Nichtoffenlegung von Schwachstellen. Dabei berücksichtigen sie individuelle und gesellschaftliche Auswirkungen.“ (ISB 2025)

Es wird deutlich, dass man sich nicht auf eine gesellschaftlich-kulturell-wirtschaftliche und anwendungsbezogene Perspektive beschränken kann, um ein Bewusstsein für die Gefährdungsszenarien von IT-Systemen zu erreichen. Um ein Verständnis für die zugrundeliegenden Wirkprinzipien zu erzeugen, muss auch - zumindest exemplarisch - der technologische Hintergrund aufgezeigt werden; es ist dazu notwendig, dass die Lernenden Fälle auch aus der Perspektive des Angreifenden analysieren. Dabei besteht das Problem, dass nach den sogenannten „Hackerparagraphen“ BGB §202a und §202c das unbefugte Ausspähen und allein die Vorbereitung darauf strafbare Handlungen sind, wofür die Lernenden im Rahmen der Unterrichtssequenz auch sensibilisiert werden sollten. Experimente müssen also in einem System stattfinden, das den Zugriffsversuch explizit zulässt. Ein solches System ist der nachfolgend vorgestellte OWASP Juice Shop. Der Fokus der Angriffe (vgl. Abb. 1) liegt dabei auf den besonders relevanten Bereichen Diebstahl, Datenspionage und Manipulation von IT-Systemen.

OWASP steht für Open Worldwide Application Security Project. Es handelt sich um eine gemeinnützige Stiftung, die sich für die Verbesserung der IT-Sicherheit einsetzt und die ihre Projekte frei, offen und kostenlos bereitstellt. Unter anderem führt OWASP mit der „OWASP Top 10“ eine anerkannte, regelmäßig aktualisierte Liste der kritischsten Angriffsszenarien für Webanwendungen (OWASP Top Ten, 2025). Der Juice-Shop (OWASP Juice Shop, 2025), der im Rahmen der nachfolgenden Fallbeispiele vorgestellt wird, bietet einen „Spiel- und Experimentierplatz“, um sich praktisch mit gängigen

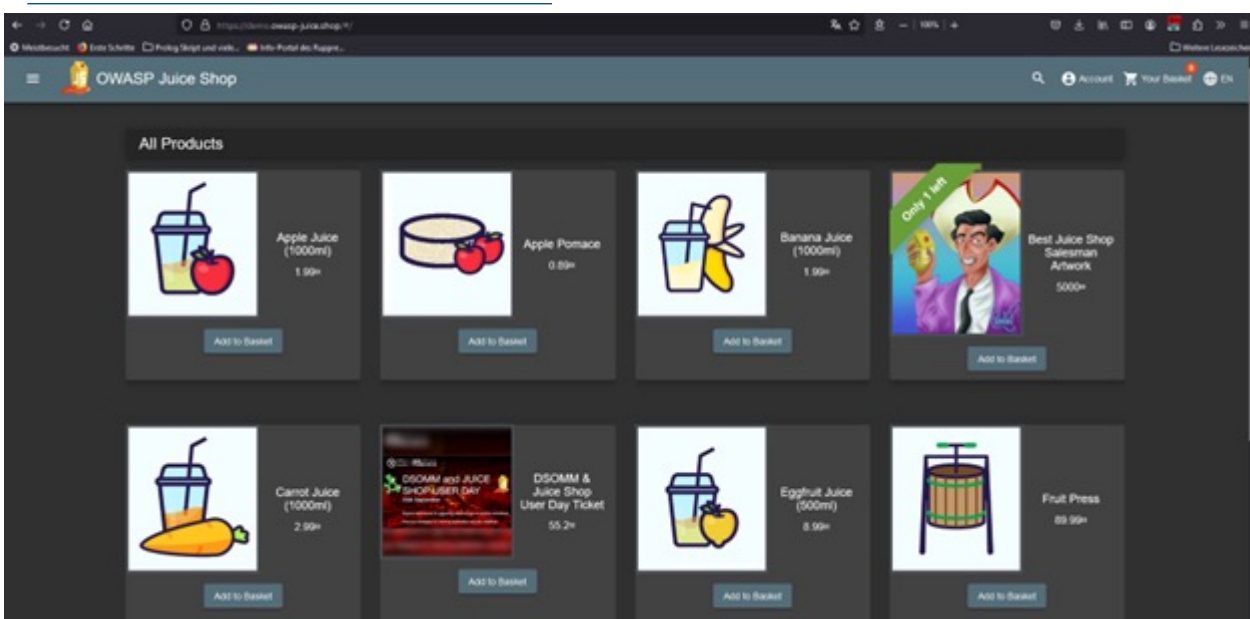


Abbildung 2: demo.owasp-juice.shop (OWASP Foundation/ CC BY-SA 4.0)

Angriffsszenarien vertraut zu machen. Unter <http://demo.owasp-juice.shop> steht eine regelmäßig aktualisierte Deployment-Test-Version ohne garantierte Verfügbarkeit bereit. Eine weitere Instanz findet man unter <https://juice-shop.herokuapp.com>. Für intensivere Auseinandersetzung empfiehlt es sich, auf Basis von node.js eine lokale Instanz bzw. einen eigenen Juice-Shop-Server zu installieren.

Die nachfolgend beschriebenen Fallbeispiele sind für den konkreten Unterrichtseinsatz konzipiert, z. B. in Form einer Expertenarbeit in Kleingruppen. Je nach Kursgröße können bestimmte Fallbeispiele weggelassen bzw. weitere ergänzt werden. Jedes Szenario steht dabei exemplarisch für einen typischen Angriffsvektor, der so oder in ähnlicher Form bei echten Cyberangriffen verwendet wurde. Ziel ist es, dass Schülerinnen und Schüler erkennen, dass die Vektoren sehr vielfältig sein können und Angreifer sich typischerweise über eine gefundene Schwachstelle Zugang zu weiteren Teilen der Systeme verschaffen können.

Angriffsszenario 1: Anmelden als Admin mittels SQL-Injection

Beschreibung des Szenarios: Bei Webanwendungen werden oft im Frontend Eingabefelder benutzt, die im Backend in SQL-Statements übersetzt werden. Das Beispiel zeigt eine typische Anmeldemaske und die dadurch ausgelöste SQL-Abfrage zur Überprüfung der Logindaten.

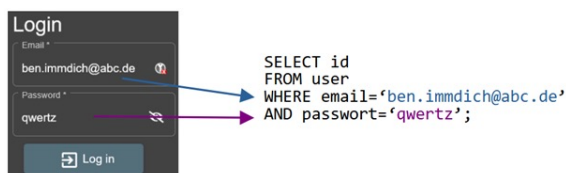


Abbildung 3: Anmeldemaske und beispielhafte Übersetzung in SQL-Anweisung (Klaus Reinold/ OWASP Foundation/ CC BY-SA 4.0)

ten.

Angriffsvektor: Werden die eingegebenen Daten aus dem Frontend direkt und ungeprüft in einen SQL-Befehl eingesetzt, so kann ein Angreifer dies gezielt ausnutzen. Dazu nutzt man Elemente der Syntax von SQL, beispielsweise das Apostroph zur Klammerung von Textdaten, das Semikolon zur Abtrennung zweier SQL-Befehle oder einen doppelten Bindestrich, um den Rest einer Zeile als Kommentar zu kennzeichnen. Trägt man nun im Feld der E-Mail-Kennung die auf den ersten Blick sinnlose Phrase ' OR TRUE; -- ein, so wird erst bei genauerer

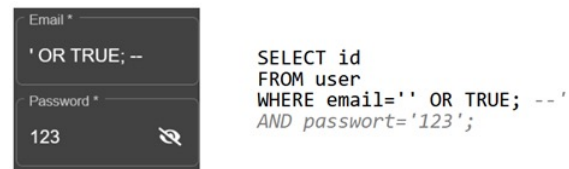


Abbildung 4: SQL-Injection (Klaus Reinold/OWASP Foundation/ CC BY-SA 4.0)

Betrachtung des entsprechenden SQL-Befehls klar, was man dadurch erreicht:

Es entsteht eine SQL-Abfrage mit einer Bedingung, bei der der erste Teil (email= ' ') wohl für keinen Benutzer wahr ist. Durch die Verknüpfung OR TRUE entsteht aber insgesamt eine Bedingung, die für alle Datensätze zutrifft. Durch den doppelten Bindestrich wird der Rest der Abfrage auskommentiert, also irrelevant. Man erhält somit eine Ergebnistabelle, die alle Benutzer-IDs umfasst. Es lässt sich vermuten, dass das Entwicklungsteam aufgrund der Eindeutigkeit von Benutzernamen davon ausgegangen ist, dass höchstens ein Datensatz in der Ergebnistabelle vorhanden sein kann. Die Anmeldung erfolgt daraufhin für den ersten Datensatz der Benutzertabelle, welches häufig – so auch hier – ein Administrator ist. Jackpot für den Angreifer, er ist nun als privilegierter Nutzer im System angemeldet, ganz ohne die Eingabe eines richtigen Passworts.

Fazit: Statt sich am System anzumelden und damit Zugang zu erlangen, könnte der Angreifer auch die Verfügbarkeit des Systems angreifen und ' OR TRUE; SHOW TABLES; -- ausprobieren, um die Tabellenbezeichner zu ermitteln. Mit dem TRUNCATE-Befehl könnte man diese Tabellen dann leeren. Die Folge wäre eine Katastrophe für den Anbieter, da alle seine Kunden in der Tabelle gelöscht würden und sich niemand mehr am System anmelden könnte.

Zum Schutz vor SQL-Injections nutzt man heute standardmäßig Prepared Statements. Deren Grundidee ist es, SQL-Befehl und Daten nicht in einer Zeichenkette an den SQL-Interpreter zu übergeben, sondern Befehl und Daten zu trennen. Erst nach Interpretation des Befehls werden dann die Daten eingefüllt und der Befehl ausgeführt. Trotz seit langem bekannter Gegenmaßnahmen stehen Injections aber immer noch in der Liste der gängigsten und erfolgreichsten Angriffsszenarien.

Angriffsszenario 2: „Security through obscurity“

Beschreibung des Szenarios: In der IT-Sicherheit gibt es verschiedene Ansätze, um Systeme und Informationen vor unbefugten Zugriffen zu schützen. Um sich komplizierte Sicherheitsmechanismen zu sparen, ist ein sehr einfacher Ansatz „Security through obscurity“: Dabei wird die Sicherheit eines Systems dadurch gewährleistet, dass der genaue Ort von Daten unbekannt ist oder ein verwendeter Algorithmus geheim gehalten wird. Dass dies kein sinnvoller Ansatz ist, besagt das Kerckoffsche Prinzip (Koch 2024), nach welchem die Sicherheit eines Systems niemals von der Geheimhaltung des Designs abhängen darf.

Dies hat mehrere Gründe:

- Ein geheimer Speicherort kann gefunden werden und ein geheimer Algorithmus kann durch Reverse Engineering rekonstruiert werden.
- Sollte ein Speicherort oder Algorithmus bekannt werden, ist es deutlich schwieriger, zu einem alternativen Speicherort oder Algorithmus zu wechseln, als einen neuen Schlüssel zu erstellen.
- Je mehr Fachleute sich mit einem Sicherheitsverfahren beschäftigen, desto eher werden Schwachstellen in diesem entdeckt.

Das Einführungsbeispiel in diesem Szenario ist die sogenannte Steganographie. Dabei wird In-

formation so verborgen, dass diese für Dritte bei einer oberflächlichen Betrachtung nicht zu sehen ist. Es lassen sich beispielsweise mehrere Bilder zu einer Datei kombinieren, von denen im Browser nur eines angezeigt wird.

Angriffsvektor: Dieses Verfahren findet Anwendung in der „About us“-Seite des „Saftladens“. Betrachtet man mit Hilfe der Entwicklerwerkzeuge den Quellcode der Webseite im Browser genauer (STRG+UMSCHALT+I oder F12), so stellt man fest, dass unerwartet eines der sieben Bilder bei gleichen Proportionen eine abweichende Dateiendung besitzt: *Bild 5* ist vom Typ *png*. Speichert man diese Datei, indem man den Pfad aus dem Quellcode kopiert und in einem neuen Tab aufruft, so kann man sie danach genauer untersuchen.

Mit frei verfügbaren Programmen wie *OpenStego*¹ lassen sich durch Steganographie überlagerte Bilder wieder trennen. Öffnet man hier die Datei *Bild 5*, so findet man das Bild eines Charakters aus der Fernsehserie „Rick and Morty“. In diesem Beispiel ist das geheime Bild eher ein versteckte Hommage; in anderen Fällen könnte an dieser Stelle ebenso ein Passwort stehen. Mit *OpenStego* lassen sich mit wenigen Klicks auch eigene Dateien in anderen Dateien verstecken.

Fazit: Der Ansatz „Security through obscurity“ ist mit einem Schlüssel unter der Fußmatte eines Hauses zu vergleichen. Solange niemand

¹<https://www.openstego.com>

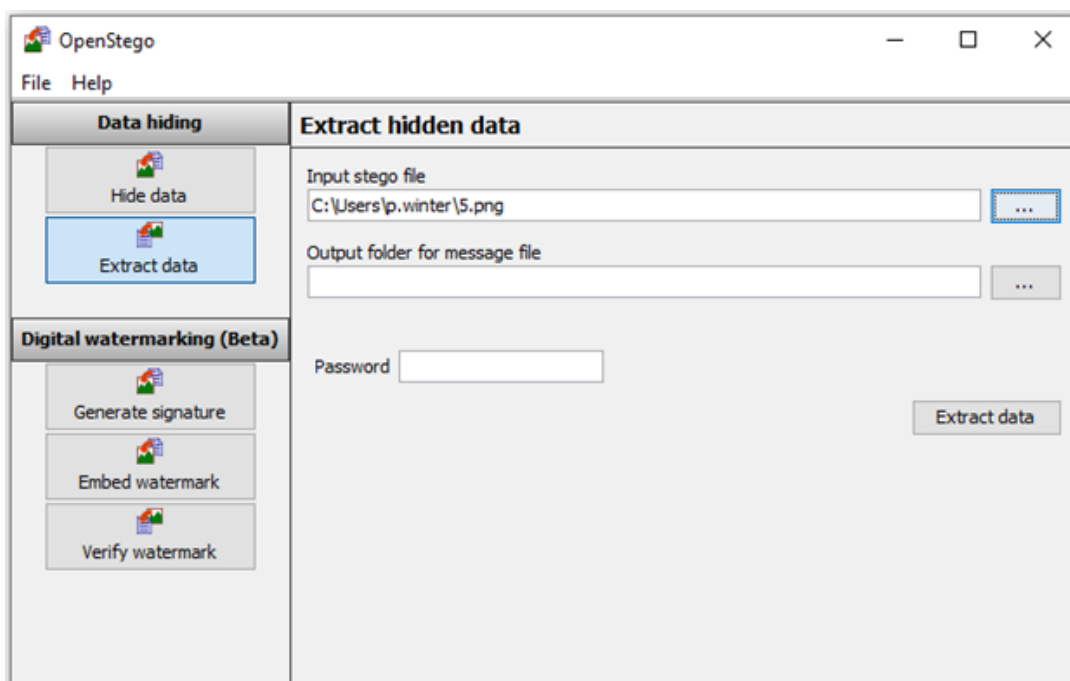


Abbildung 5: Screenshot OpenStego (Philipp Winter/ CC BY-SA 4.0; Samir Vaidya GPL2.0)

genauer danach sucht, ist das Haus sicher. Weiß man aber, wonach man suchen muss, sieht es schnell ganz anders aus. Der „Saftladen“ nutzt an vielen Stellen solche vermeintliche „Sicherheitsmaßnahmen“ und es werden in den folgenden Szenarien noch weitere betrachtet.

Angriffsszenario 3 – Inhalte und Struktur des Webservers auslesen

Beschreibung des Szenarios: Im Saftladen lassen sich Artikel erwerben, indem man sie zunächst in den Warenkorb legt. Vor dem Abschluss des Kaufs bietet sich die Möglichkeit, einen Coupon-Code einzugeben, der einen prozentualen Nachlass gewährt. Der Social-Media-Account des Shops veröffentlicht regelmäßig solche Codes, um Kunden anzulocken, ein beispielhafter Code ist `k#pDlG7sn`. Nun wäre es interessant, selbst einen möglichst hochwertigen Rabattcoupon zu erzeugen. Dazu benötigt man Kenntnis über die Struktur der Rabattcoupons. Oft sind nicht über einen Link zugängliche Dateien auf dem Server eine Quelle für solche Informationen. Es wird bei der Bearbeitung dieser Sicherheitslücke wiederum deutlich, dass „Security Through Obscurity“ keine sinnvolle Strategie zum Schutz einer Website ist.

Angriffsvektor: Grundsätzlich kann eine Website als Ordnerstruktur gesehen werden, in der die verschiedenen Ordner und Dateien jeweils über eine eindeutige URL erreichbar sind. Um eventuell ungesicherte Dateien an bestimmten URLs zu finden, lohnt es sich, bei der Navigation durch die Seite einen gezielten Blick auf die Pfade zu werfen – so wird auf der Seite About Us ein Link zu [Check out our boring terms of use if you are interested in such lame stuff](#) angegeben. Nach Klick auf den Link wird die Datei <https://juice-shop.herokuapp.com/ftp/legal.md> geöffnet, worin uninteressanter Text steht. Jedoch zeigt die URL, dass es einen Ordner namens `ftp` auf dem Server gibt, der die Datei `legal.md` enthält. Eingabe der URL ohne den Dateinamen eröffnet einen Blick in das entsprechende Verzeichnis (Abb. 6), das offensichtlich nicht für den öffentlichen Zugriff gedacht ist.

Fazit: Fehlkonfigurationen beim Webserver erlauben es mit relativ wenig Aufwand, Außenstehenden einen Einblick in die Inhalte des Webservers zu erhalten. Diese Einblicke in Struktur und Inhalte können dann Ausgangspunkt für weitere Aktivitäten sein.

Angriffsszenario 4 – Fälschen von Rabattcoupons

Beschreibung des Szenarios: Beim Betrachten der auf dem Server hinterlegten Daten scheinen die Dateien `coupons_2013.md.bak` und `package.json.bak` Hinweise zur Coupongenerierung zu enthalten. Leider erscheint bei Aufruf dieser Dateien einer Fehlermeldung, dass der Zugriff nur auf Dateien im Format `.md` und `.pdf` erlaubt ist. Dieser Schutz lässt sich durch Einsatz des sogenannten *Poison Null Byte* umgehen.

Angriffsvektor: Die Länge einer Zeichenkette wird entweder durch die explizite Angabe der Länge oder durch die Verwendung eines speziellen Symbols am Ende des Strings erreicht. Letzteres ist als Null Byte (`%00`) bekannt. Dieses Null Byte kann verwendet werden, um die Beschränkung auf bestimmte Dateitypen zum Download zu umgehen. Dazu wird die URL der Datei so angepasst, dass ein Null Byte nach der eigentlichen Dateiendung eingefügt und danach eine passende Endung ergänzt wird, welche die Dateiendung in den „Augen“ des Servers ändert und so die Sperre umgeht. So wird aus

```
~/ftp/coupons_2013.md.bak
```

per Null Byte

```
~/ftp/coupons_2013.md.bak%00.md
```

Diese Herangehensweise funktioniert jedoch noch nicht, da URLs ein eigenes Encoding haben – das `%`-Symbol (vom Null Byte `%00`) muss daher auch mit der Zeichenkette `%25` encodiert werden. Der Aufruf von

```
~/ftp/coupons_2013.md.bak%2500.md
```

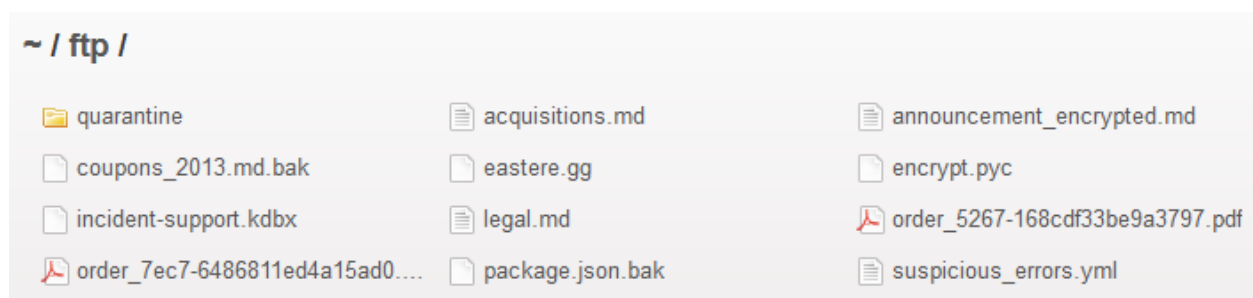


Abbildung 6: Ein Blick hinter die Kulissen des OWASP Juice Shops (Han-Min Kufner/ CC BY-SA 4.0)

hat den Download der Datei zur Folge. Analog kann für andere interessante Dateien im Ordner verfahren werden.

Die Datei package.json.bak ist von großem Wert, da sie offenlegt, welche Codebibliotheken bei der Entwicklung der Website verwendet wurden. Für die Coupongenerierung ist es naheliegend, eine Bibliothek für Hashing-Algorithmen oder andere kryptografische Verfahren zu ver-

im Admin-Bereich einsehen; als eingeloggter Admin agiert man quasi wie ein normaler User. Allerdings kann man sich trotzdem diesen Zugang erschleichen, z.B. um in diesem Szenario 5-Sterne-Feedback zu löschen und dadurch den Shop schlechter darzustellen.

Angriffsvektor: Zum Admin-Bereich kann man nur Zugriff erhalten, indem man sich wie in Szenario 1 beschrieben als Admin anmeldet. Der

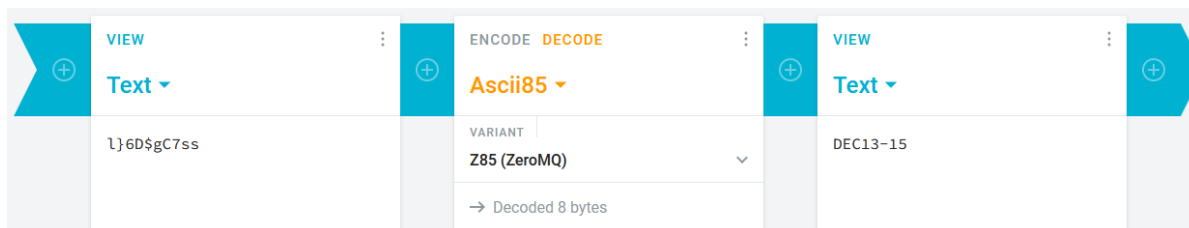


Abbildung 7: Nutzung eines Online-Tools zur Codierung/Decodierung mit dem z85-Verfahren <https://cryptii.com/pipes/z85-encoder> (Han-Min Kufner/ CC BY-SA 4.0)

muten. Ein Blick auf die Liste der verwendeten Pakete enthält auch z85, ein Codierungsverfahren.

Probeweises Dekodieren eines Coupons, z.B. „l}6D\$gC7ss“ aus der Coupon-Backup-Datei mit dem z85-Verfahren ergibt die Zeichenkette „DEC13-15“.

Weitere Coupon-Decodierungen verdeutlichen, dass der Code mit einem Monatskürzel und dem zugehörigen Jahr beginnt und nach dem Bindestrich angibt, wie viel Prozent der Rabatt betragen soll. Nun lassen sich nach Herzenswunsch Coupons erstellen, so z.B. ein Coupon für den aktuellen Monat (DEC24 für den Autor) mit 99% Nachlass. Dazu kodiert man den String „DEC24-99“ und erhält: l}6D#g+yZF.

Fazit: Bei diesem Szenario zeigt sich erneut, dass ein bloßes Verstecken von kritischer Struktur keine nachhaltige Strategie zur Sicherung einer Website sein kann. Auch wenn solche Information die Seite nicht unmittelbar kompromittieren muss, so lassen sich dadurch oft wichtige weitere Angriffsvektoren identifizieren und es wird zunehmend schwieriger, alle Eventualitäten abzusichern.

Angriffsszenario 5: Unbefugter Zugriff auf den Admin-Bereich

Beschreibung des Szenarios: Das Abtrennen des Admin-Bereichs vom Administrator-Account kann bereits schützen, da ein Angreifer, der sich mit Angriffsszenario 1 erfolgreich als Admin eingeloggt hat, weniger Schaden anrichten kann. Im Juice Shop kann man die Liste der registrierten User und das Käufer-Feedback nur

Bereich ist zwar nicht als Verweis anklickbar im Menü hinterlegt; man kann jedoch die URL einfach erraten, durch Vergleich mit den URLs verschiedener Bereiche. Um zum Administrationsbereich zu gelangen, wird das Stichwort „administration“ an den Teil „/#/“ angehängt. Nun

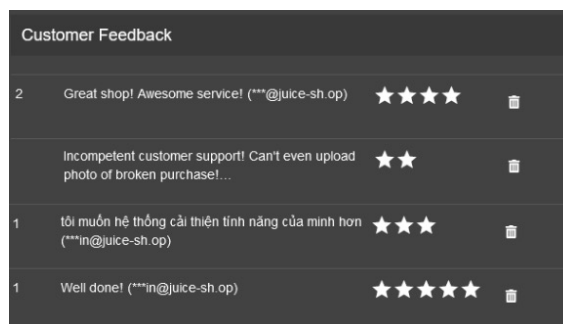


Abbildung 8: Ausschnitt des Customer-Feedbacks (Reinhild Kokula/ OWASP Foundation/ CC BY-SA 4.0)

sieht man links die Tabelle der registrierten User und rechts die Tabelle „Customer Feedback“. Durch Klicken auf die Mülltonne kann man ungewolltes Feedback nun löschen und dadurch das Szenario lösen.

Fazit: Gängige Maßnahmen sind hier der Schutz des Administratorpassworts (Angriffsszenario 1), der Verzicht auf direkte Verweise zum Adminbereich im Menü (siehe Angriffsszenario 2) sowie das Vermeiden von gängigen URL-Endungen („.../#/administration“). Das Entwicklungsteam geht also davon aus, dass diejenigen, die als Administrator eingeloggt sind und die URL zum Administrationsbereich kennen, dies auch rechtmäßig tun. Allerdings ist die URL zu vorhersehbar, um tatsächlich Schutz zu gewährleisten. Eine einprägsame URL wie „.../#/

noitartsnimda“ (administration rückwärts) ist aber nicht zielführend, da dies wieder Security through obscurity darstellt (siehe Angriffsszenario 2). Bereits das Wählen einer komplizierteren Zieladresse, z.B. ein zufällig generierter String, kann das Finden des Administrationsbereichs erheblich erschweren. Zusätzlich könnte der Zugang durch ein weiteres, entsprechend gegen Angriffsszenario 1 abgesichertes Passwort eingeschränkt werden.

Angriffsszenario 6: Manipulierte Bewertungen

Beschreibung des Szenarios: Klickt man auf den Menü-Button in der linken oberen Ecke und dann auf den Menüpunkt "Customer Feedback",

Abbildung 9: Feedback-Formular (Julian Scholz/ OWASP Foundation/ CC BY-SA 4.0)

so gelangt man auf die Feedback-Seite. Wenn man nicht eingeloggt ist, wird das Feedback automatisch unter dem Namen „anonymous“ abgeschickt. Ziel dieses Angriffsszenarios ist es, das Feedback unter dem Namen eines bereits registrierten Benutzers abzuschicken.

Angriffsvektor: Beim Drücken des Submit-Buttons wird ein POST-Request ausgeführt, der folgende Information als JSON-Daten überträgt. Dieser kann mittels des Netzwerkanalyse-Tabs über die Entwicklerwerkzeuge ausgelesen werden:

```
{
  "captchaID": 7
  "captcha": "5"
  "comment": "Das ist ein Test. "
  "rating": 5
}
```

Bei diesem POST-Request werden keine Benutzerinformationen mitgesendet. Nun betrachtet man die JSON-Daten der Antwort auf diesen POST-Request:

```
{
  "status": "success",
  "data": {
    "id": 7,
    "comment": "Das ist ein Test",
    "rating": 5,
    "updateAt": "2024-12-09T13:07:38.035Z",
    "createAt": "2024-12-09T13:07:38.035Z",
    "UserId": null
  }
}
```

In der Antwort befindet sich zusätzlich die UserID. Nun betrachtet man den Quelltext der Webseite mit den Entwicklerwerkzeugen genauer. Zur betreffenden Stelle im Quelltext, die untersucht werden soll, gelangt man über einen Rechtsklick auf das Formular und anschließend über den Menüpunkt Untersuchen des Kontextmenüs. Hier die betreffende Zeile:

```
<input id="userId" class="ng-untouched
ng-pristine ng-valid" ngcontent-nnm-
c23="" hidden="" type="text">
```

Durch das Schlüsselwort "hidden" wird dieses Eingabefeld dem Benutzer jedoch nicht angezeigt (siehe Angriffsszenario 2). Manipuliert man die Zeile und entfernt das Schlüsselwort "hidden", so erscheint das Eingabefeld im Feedbackformular.

Jetzt kann Feedback unter einer anderen Benutzeridentität versendet werden. Betrachtet man

Abbildung 10: Feedback-Formular mit ID-Feld (Julian Scholz/ OWASP Foundation/ CC BY-SA 4.0)

nochmals die Daten des POST-Requests, so wird tatsächlich die User-ID mit übergeben:

```
{
  "UserID": "1",
```

```

"captchaID": 7,
"captcha": "5",
"comment": "Das ist ein Test.",
"rating": 5
}
    
```

Fazit: Aus Sicherheitsgründen sollte vor der Veröffentlichung des Feedbacks geprüft werden, dass wirklich nur Daten des aktuell angemeldeten Benutzers manipuliert werden (Integrität). Da das bei der Entwicklung aber schnell einmal vergessen wird, können Schwachstellen entstehen.

Angriffsszenario 7: Umgehen eines CAPTCHAs

Beschreibung des Szenarios: Webanwendungen sind täglich unerwünschten automatisierten Angriffen durch Bots ausgesetzt, beispielsweise dem automatisierten Ausfüllen von Formularen oder Erstellen von sogenannten „Fake-Konten“. Diese Angriffe können mit CAPTCHAs verhindert werden. CAPTCHA (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part - Ein vollständig automatisierter öffentlicher Turing-Test zur Unterscheidung zwischen Computern und Menschen) ist ein Verfahren, welches sicherstellen soll, dass eine Interaktion mit einem System von einem Menschen und nicht von einem automatisierten Bot durchgeführt wird. CAPTCHAs werden eingesetzt, um jegliche Art von automatisiertem Missbrauch, einschließlich Brute-Force-Angriffen, zu stoppen. Sie funktionieren, indem sie einen Test präsentieren, der für Menschen *leicht* zu bestehen, für Computer jedoch *schwer* zu bewältigen ist. Daher kann man mit einiger Si-

cherheit feststellen, ob ein Mensch eine Anfrage tätigt.

Angriffsvektor: Ziel des Angriffs soll es sein, innerhalb von 10 Sekunden zehn oder mehr Kundenfeedbacks möglichst automatisiert zu übersenden. Das Customer Feedback Formular (vgl. Abbildung 9) enthält ein CAPTCHA, um es vor Missbrauch zu schützen. In diesem Fall muss der Benutzer eine einfache Rechenaufgabe lösen, bevor er das Feedback abschicken kann. Das manuelle Abschicken eines Feedbacks inklusive Lösung des CAPTCHAs in weniger als zwei Sekunden stößt hierbei schon an seine Grenzen. Die Herausforderung dieses Szenarios ist daher, den Automatisierungsschutz zu überwinden.

Dies gelingt mit den folgenden Schritten:

1. Vorbereitung: Öffnen Sie über das Hauptmenü das Customer Feedback-Formular und die Entwicklerwerkzeuge.

```

{
  "captchaId": 8,
  "captcha": "8+5-5",
  "answer": "8"
}
    
```

2. Wechseln Sie in den Tab Netzwerk (oder Netzwerkanalyse).
3. Sie sollten eine GET-Anfrage an die Adresse <https://juice-shop.herokuapp.com/rest/captcha/> bemerken, die das CAPTCHA für das Feedback-Formular abrufen. Die HTTP-Antwort sieht ähnlich aus wie:

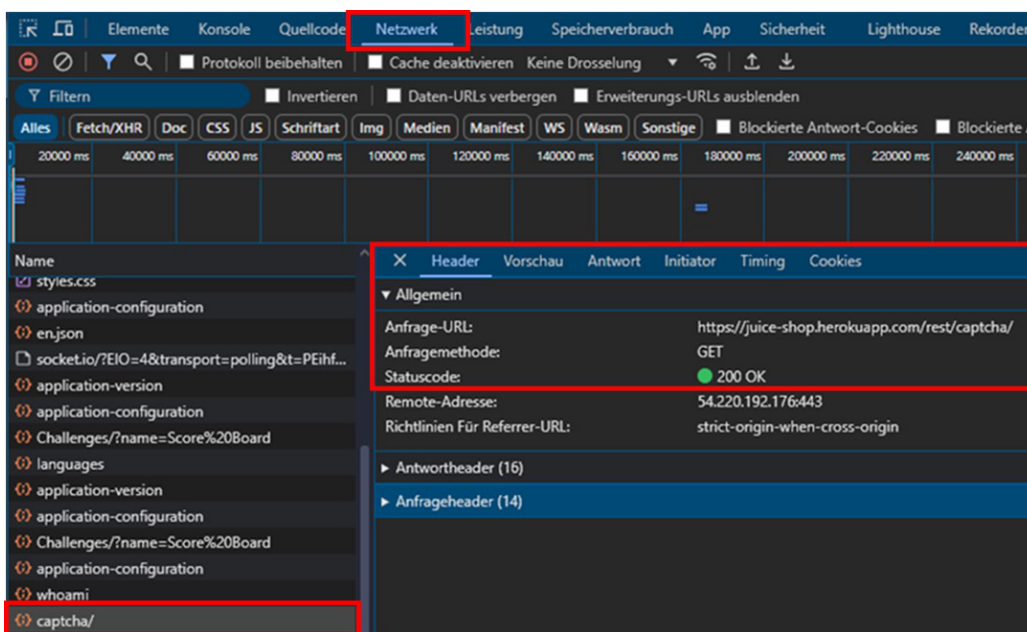


Abbildung 11: Entwicklerwerkzeuge im Browser (Benedikt Leesch/CC BY-SA 4.0)

```

{
  captchaId: 9,
  captcha: "10",
  comment: "Testkommentar
           (anonymous)",
  rating: 3
}

```

4. Füllen Sie das Formular aus und senden Sie es ab. Beobachten Sie währenddessen den Netzwerk-Tab. Die CAPTCHA-ID und die Lösung werden zusammen mit dem Feedback als POST-Anfrage (Feedbacks/) übermittelt:
5. Währenddessen wurde ein neues CAPTCHA mit neuer Aufgabe abgerufen.
6. Kopieren Sie nun die POST-Anfrage als Fetch:
Rechtsklick Feedbacks/ -> Kopieren -> Als fetch kopieren
7. Wechseln Sie nun in den Tab Konsole. Dort können Sie den kopierten Fetch einfügen und mit Enter bestätigen. Der Server akzeptiert die *POST-Anfrage*, obwohl die CAPTCHA Lösung nicht mehr zu der aktuellen Rechnung passt. Wiederholen Sie dies mehrmals und die Aufgabe ist gelöst.
8. *Automatisierung*: Schreiben Sie ein kurzes Javascript, welches mit einer Wiederholung automatisiert die *POST-Anfrage* an den Server schickt. Führen Sie das Skript im

Tab *Konsole* aus. Sie können das folgende Beispiel als Vorlage verwenden.

Tipp: Fügen Sie das Skript in einem Editor zusammen.

Fazit: Dieses Beispiel zeigt: CAPTCHAs sind nicht unbesiegbar. Das Entwicklungsteam muss sicherstellen, dass CAPTCHAs nicht wiederverwendet werden können. Aber auch aktuellere CAPTCHAs, die auf Bildverarbeitung setzen, müssen aufgrund der rasanten Weiterentwicklung der Künstlichen Intelligenz in Zukunft neu gedacht werden.

Angriffsszenario 8: Anfragen vor dem Senden abfangen

Beschreibung des Szenarios: In den Angriffsszenarien 4 und 6 wird über die Entwicklerwerkzeuge eine bereits gesendete Anfrage im Netzwerk-Tab ausgelesen und anschließend über die Konsole eine modifizierte Version erneut gesendet. Wollte man nun eine Bewertung des Onlineshops manipulieren, so müsste man zunächst mindestens einmal eine 1-Sterne-Bewertung abgeben, um den Aufbau und Inhalt der Anfrage auslesen zu können; selbst einen Stern hat dieser wortwörtliche „Saftladen“ nicht verdient! Das Ziel ist es, die Webseite dazu zu bringen, eine ungültige Bewertung zu akzeptieren, ohne zuvor eine gültige Anfrage gesendet zu haben. Die übliche Bewertungsskala (1 bis 5 Sterne) wird dabei durch eine manipulierte Anfrage umgangen. Gerade bei der Vorberei-

```

const sendFeedback = async () => {
  for (let i = 0; i < 10; i++) {
    try {
      const response = await fetch(...
                                  ...);
      if (response.ok) {
        console.log(`Anfrage ${i + 1}:Erfolgreich gesendet!`);
      } else {
        console.error(`Anfrage ${i + 1}:Fehler ${response.status}`);
      }
    } catch (error) {
      console.error(`Anfrage ${i + 1}: Fehler`, error);
    }
  }
};
sendFeedback();

```

Ersetzen Sie **fetch** durch die kopierte Post-Anfrage.

Das Skript wiederholt die Anfrage zehnmal.

//Ersetzen

Error-Handling: Fehler oder ungültige Antworten werden in der Konsole ausgegeben.

Quelltext 1: Automatisierung des CAPTCHA-Angriffs

tung von größeren Angriffen möchte man vor-

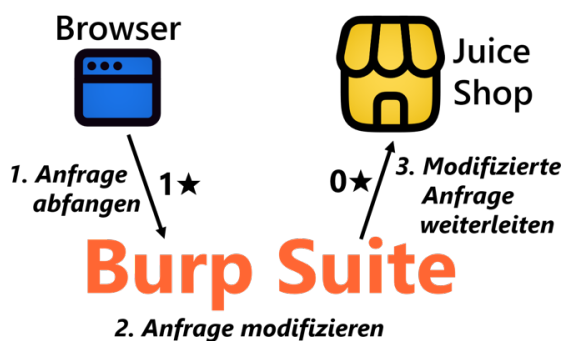


Abbildung 13: Improper Input Validation Angriff: Datenfluss der Anfrage (Sebastian Wich/ CC BY-SA 4.0)

her „unter dem Radar“ bleiben und das Abschicken von nicht notwendigen Anfragen in jedem Fall vermeiden. Im Folgenden werden die in Abbildung 13 gezeigten Schritte zur Ausführung eines Angriffs genauer erläutert.

Angriffsvektor: Mit dem Programm *Burp Suite*² wird der Datenverkehr zwischen dem Browser und der Webseite abgefangen. Der Angreifer navigiert zur Bewertungsfunktion, wählt eine Bewertung (z. B. 1 Stern) aus und schickt diese ab. Burp Suite fängt die Anfrage clientseitig ab, bevor sie den Server erreicht, sodass der Angreifer sie einsehen und bearbeiten kann.

Beispiel des relevanten Teils einer abgefangenen Anfrage:

```
{
  "captchaID": 0,
  "captcha": "10",
  "comment": "Das ist ein törichter Saftladen!",
  "rating": 1
}
```

In der abgefangenen Anfrage wird der Parameter „rating“ identifiziert, der die Anzahl der Sterne (hier 1) überträgt. Statt des ursprünglich gewählten Wertes wird ein ungültiger Wert, wie z. B. 0, eingetragen.

Die manipulierte Abfrage wird anschließend an den Server weitergeleitet. Dieser Angriff nutzt wiederum vor allem eine fehlende Validierung auf Serverseite aus. Wenn die Webseite nur auf Client-Seite im Browser überprüft, ob die Bewertung innerhalb der erlaubten Werte liegt, kann dies leicht umgangen werden. Wie bei den vorangegangenen Beispielen beschrieben, sollten Eingabedaten stets auf Serverseite validiert werden.

Fazit: Burp Suite ermöglicht neben dem Abfangen von Anfragen auch viele weitere Möglichkeiten für Sicherheitsfachleute, wie etwa automatisierte Schwachstellenanalyse, Manipulation von Daten, Durchführung von Angriffssimulationen wie Brute-Force- und Wörterbuchan-

griffe sowie die umfassende Sicherheitsprüfung von Webanwendungen.

Fazit

Webanwendungen sind ein zentraler Bestandteil moderner IT-Infrastrukturen, jedoch auch ein häufiges Ziel für Angriffe. Strukturierte Sicherheitsmaßnahmen sind zur Abwehr erforderlich. Durch eine Kombination aus manuellen Tests und automatisierten Werkzeugen (Burp Suite) lassen sich Sicherheitslücken präzise identifizieren und beheben, was die Stabilität und Verlässlichkeit von Anwendungen entscheidend erhöht. Durch eine Simulation und Thematisierung der Angriffsszenarien im Informatikunterricht wird Sicherheit als integraler Bestandteil der Softwareentwicklung offenbart und damit ein Problembewusstsein für Cyber-Sicherheit geschaffen. Darauf aufbauend lässt sich eine Verhaltensänderung hin zu einem sicheren Umgang entwickeln. Dabei sollten neben den beschriebenen Angriffen auf die Technik auch der Faktor Mensch als Schwachstelle bewusstgemacht werden, der durch Social Engineering (Phishing etc.) Angriffsvektoren bietet, aber auch durch unbeabsichtigte Fehler eine Gefahr darstellt.

Der OWASP Juice Shop bietet neben den skizzierten Fallbeispielen eine große Anzahl weiterer lehrreicher Angriffsvektoren für gezielte Angriffe auf ein Softwaresystem. In diesem Sinne ist er ein interessanter und herausfordernder Spiel- und Übungsplatz zum Thema Informationssicherheit und – für diesen Zweck – überhaupt kein Saftladen!

Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 21.01.2025.

Brichzin, Peter u. a. (2024): Informatik 6 erhöhtes Niveau. Cornelsen, Berlin.

Bundesamt für Sicherheit in der Informationstechnik (2024): Die Lage der IT-Sicherheit in Deutschland https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2024.pdf?__blob=publicationFile&v=5

Bundesamt für Sicherheit in der Informationstechnik (2025): Awareness https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Angriffszielen/Faktor-Mensch/Awareness/awareness_node.html

² <https://portswigger.net/burp>

Koch, A. (2024). Symmetrische Kryptologie und ihre Veranschaulichung. Informatische Bildung in Schulen 2(2). <https://doi.org/10.18420/ibis-02-02-07>

Ministerium für Bildung, Wissenschaft und Kultur des Landes Mecklenburg-Vorpommern, Institut für Qualitätsentwicklung (Institut für Qualitätsentwicklung) (2019). Rahmenplan Informatik für die Qualifikationsphase der gymnasialen Oberstufe. https://www.bildung-mv.de/export/sites/bildungserver/downloads/unterricht/rahmenplaene_allgemeinbildende_schulen/Informatik/RP_INFO_SEK2.pdf

Qualitäts- und UnterstützungsAgentur - Landesinstitut für Schule (QUA-LiS) (2024). Kernlehrplan Informatik für die Gymnasiale Oberstufe. <https://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/informatik/informatik-klp/index.html>

Staatsministerium für Schulqualität und Bildungsforschung (ISB), München (2025). LehrplanPLUS Gymnasium. Fachlehrplan Informatik 12 (erhöhtes Anforderungsniveau). <https://www.lehrplanplus.bayern.de/fachlehrplan/gymnasium/12/informatik/erhoeht>

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Reinhild Kokula
Gymnasium Münchberg

Han-Min Kufner
Carl-Friedrich-Gauß-Gymnasium Schwandorf

Benedikt Leesch
Deutschherren-Gymnasium Aichach

Klaus Reinold
Studienseminar am Rupprecht-Gymnasium München
reinold@rupprecht-gymnasium.de

Julian Scholz
Gymnasium Tutzing

Sebastian Wich
Gymnasium LSH Marquartstein
sebastian@wichematik.education

Philipp Winter
Carl-Spitzweg-Gymnasium Germering
philipp.winter@tum.de

Zum Stellenwert von Verfahren des maschinellen Lernens im allgemeinbildenden Informatikunterricht

Eickhoff-Schachtebeck, A., Strecker, K.

DOI: 10.18420/ibis-03-01-08

Zusammenfassung

Wir wollen mit diesem Artikel einen Beitrag liefern, der exemplarisch an einigen Beispielf Verfahren des maschinellen Lernens aufzeigt, welche allgemeinbildenden Lernziele damit im Unterricht verfolgt werden können. Dabei steht weniger das wiederholte Anwenden der Verfahren in verschiedenen Kontexten im Mittelpunkt des Unterrichts als vielmehr die daran herauszuarbeitenden Reflexionsaspekte.

Einleitung

Sicher ist mittlerweile unbestritten, dass das Themengebiet „künstliche Intelligenz“ Einzug in die Lehrpläne und Curricula des Informatikunterrichts entweder schon bereits erhalten hat oder zukünftig haben wird. Auch Unterrichtsmaterialien zum Themengebiet KI und Maschinelles Lernen (ML) sind schon vielerorts entstanden und ihre Menge nimmt kontinuierlich zu.

Oft werden bei bereits vorhandenen Unterrichtsmaterialien einzelne Verfahren des ML (z.B. k-nächste Nachbarn, Q-Lernen-Algorithmus oder k-Means-Clustering) in den Mittelpunkt gestellt und anhand didaktisierter Materialien von den Schülerinnen und Schülern erkundet oder angewendet. Das ist auch richtig, denn ohne konkretes, anwendbares Verfahren bleibt das Themengebiet zu abstrakt und für die Schülerinnen und Schüler nicht greifbar. Um allgemeine Konzepte erarbeiten zu können, ist sogar die Kenntnis von mindestens zwei Verfahren der gleichen Kategorie notwendig, damit die Schülerinnen und Schüler überhaupt zwischen allgemeinem Konzept und verfahrensspezifischer Umsetzung unterscheiden können.

Trotzdem muss bei jedem neu zu integrierenden Thema im Informatikunterricht zusätzlich die Frage gestellt werden, welche allgemeinbildenden Lernziele sich daraus ableiten lassen, die nicht bereits mit den bisherigen inhalts- und prozessbezogenen Kompetenzen abgedeckt sind. Die Entmystifizierung des vermeintlich „schlau“ Computers kann als einziges Argument nicht gelten, zumal dies auch im bisherigen Informatikunterricht durch geeignete Unterrichtsszenarien im Bereich der Algorithmik

oder des physical computings geschehen kann. Die Integration des Themenbereichs Informatik und Gesellschaft findet meist bereits ebenfalls im herkömmlichen Informatikunterricht in Verknüpfung mit den Themengebieten Datenbanken oder Algorithmik statt. Wir sollten festmachen, wodurch genau die Entmystifizierung des „schlau“ Computers im Bereich des Maschinellen Lernens gelingen kann oder wie man im Unterricht vermitteln kann, wie die Begriffe „Lernen“ oder „Künstliche Intelligenz“ informatisch interpretiert werden.

ML als Themengebiet im Informatikunterricht sollte auch dadurch legitimiert werden, dass neben dem bekannten algorithmischen Problemlösen jetzt zusätzlich das datenbasierte Problemlösen tritt, wobei nicht der Schüler oder die Schülerin in seiner/ihrer Rolle als Programmierer oder Programmiererin explizit die Regeln zum Erzeugen von Ausgaben vorgibt, sondern eine Datenanalyse gegebener Daten implizit zu Ausgaben für neue, ähnliche Daten führt. Damit erweitern sich die Strategien der Problemlösung für die Schülerinnen und Schüler um den datenbasierten Ansatz.

Ist das Themengebiet ML durch eine alternative Form des Problemlösens begründet, so stellt sich immer noch die Frage, welche konkreten Verfahren in welcher Tiefe Bestandteil des Unterrichts sein sollten. Auch dies muss u.a. aus dem Blickwinkel der Allgemeinbildung begründet werden. Eine reine rezeptartige, wiederholte Anwendung (didaktisierter) Verfahren und Algorithmen des maschinellen Lernens kann nicht das alleinige Lernziel einer allgemeinbildenden Schule sein. Vielmehr muss aus der Kenntnis und Anwendung der informatischen Verfahren auf der Reflexionsebene im Unterricht auch herausgearbeitet werden, welche Potentiale und Nutzen, aber auch Schwächen und Grenzen die dahinterstehenden Konzepte eint, um die Schülerinnen und Schüler zu befähigen, auch zukünftig dem gesellschaftlichen Diskurs in diesem Themenbereich folgen zu können.

Daher wollen wir in diesem Artikel für die folgende Auswahl konkreter Verfahren des maschinellen Lernens die Frage nach den dahinterstehenden allgemeinbildenden Lernzielen beantworten. Implizit findet sich dazu auch eine Didaktisierung des jeweiligen Verfahrens und eine methodische Vorgehensweise, wobei

auf verschiedene Zugänge geachtet wird und neben enaktiven Herangehensweisen und Visualisierungen auch programmiersprachliche Zugänge in vereinfachter Form verwendet werden.

Beispiele konkreter Verfahren

k- nächste Nachbarn: didaktisiertes Verfahren und Lernziele

Wir beginnen mit dem Verfahren k-nächste Nachbarn. In dem Lernszenario aus (Brandt, Eickhoff-Schachtebeck & Strecker 2022a) geht es um die Suche der passenden Mantelgröße für den eigenen Hund (Datenpunkt B in Abbildung 1). Die Käufe zufriedener Kunden (Trainingsdaten) sind in einer Grafik abgebildet. Ohne Kenntnis des Verfahrens k-nächste Nachbarn sollen die Lernenden begründen, für welche Mantelgröße sie sich entscheiden würden (bzw. die Besitzer der Hunde F(ifi), R(ocky) und N(ala)). Ein Hund wird über einen Datenpunkt mit den Merkmalen Rückenlänge (x-Achse) und Schulterhöhe (y-Achse) abgebildet. Die zugehörige Mantelgröße wird über die Farbe des Datenpunktes bestimmt (Abbildung 1). Die Beschränkung auf drei Merkmale ermöglicht die Darstellung der Daten in einem zweidimensionalen Koordinatensystem und so einen enaktiven Zugang.

Es wird gehofft, dass sich eine Strategie unter den Ideen der Lernenden findet, die dem Verfahren k-nächste Nachbarn entspricht, nämlich sich für die Mantelgröße zu entscheiden, welche der Mehrheit der z.B. fünf nächsten Datenpunkte entspricht. In diesem Fall würde die Wahl für den eigenen Hund Blacky auf S fallen (Fifi: S, Rocky: M und Nala: L).

So entdecken die Lernenden selbst das Grundprinzip des Verfahrens k-nächste Nachbarn. Gleichzeitig kann man aber auch über das Verfahren an sich reflektieren und dabei folgende Punkte festhalten:

- Vielleicht ist Größe M für Blacky besser geeignet. Das Gewicht des Hundes wurde ja gar nicht berücksichtigt und könnte bei der Wahl der Mantelgröße ebenfalls eine Rolle spielen. Dahinter steht die Erkenntnis, dass die Daten im Beispiel auf drei Merkmale reduziert wurden (Rückenlänge des Hundes, Schulterhöhe des Hundes und die gewählte Mantelgröße). Die Frage ist, ob durch die Reduktion nicht relevante Merkmale unberücksichtigt geblieben sind, bzw. wie man überhaupt relevante Merkmale identifiziert.
- Durch die Reduktion auf zwei Merkmale des Hundes sind die Trainingsdaten vielleicht ungeeignet oder ungenügend.
- Bereits klassifizierte Trainingsdaten müssen in ausreichender Zahl vorliegen.
- Um den nächsten Nachbarn zu bestimmen, wurde der euklidische Abstand gewählt. Das ist aber nur möglich, wenn auf den Achsen Zahlenwerte vorliegen. Mindestens muss es eine Ordnung auf den Datenwerten geben. Auch müssen die Einheiten an den Achsen stimmig zueinander sein, bzw. die Daten normiert vorliegen. Was wäre beim Merkmal „Beruf“ der Abstand zwischen einem Feuerwehrmann und einer Polizistin?
- Welche Zahl wählt man für k? Würden wir k=3 wählen, würden wir für Nala im obigen Beispiel die Mantelgröße M vorhersagen, im Fall k=5 die Größe L. Dahinter steht eigentlich die Frage: Wann kann man von Mustern in Daten sprechen und was sind Ausreißer?

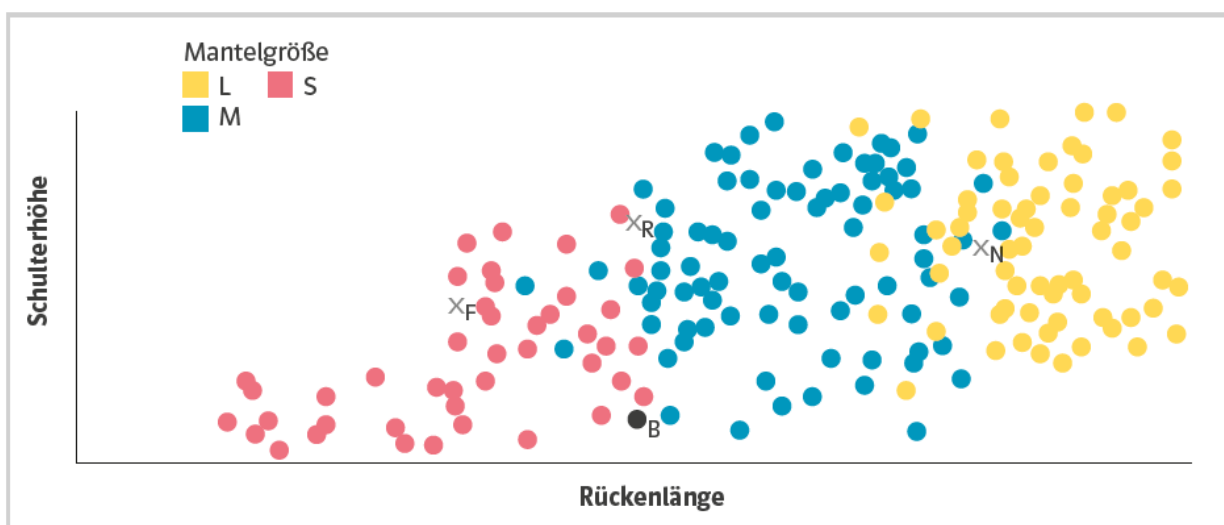


Abbildung 1: Mantelgröße für Blacky gesucht

- Basierend auf ausreichend vielen Trainingsdaten, die klassifiziert sind, treffen wir mit der Methode k-nächste Nachbarn eine Entscheidung für neue, ähnliche Daten. Dabei handelt es sich um eine datenbasierte Problemlösestrategie. Was sind die Unterschiede, was die Gemeinsamkeiten zu einer regelbasierten Problemlösestrategie?
- Probleme der Überanpassung oder Unteranpassung können ebenfalls reflektiert werden. Für $k=1$ beispielsweise würden Ausreißer in Daten als Muster interpretiert. Die Wahl für den Datenpunkt N im Fall $k=1$ wäre Mantelgröße M. Man spricht von Überanpassung. Für $k=\text{Anzahl aller Datenpunkte}$ würde selbst für den Datenpunkt F die Mantelgröße M gewählt, weil die Anzahl blauer Datenpunkte insgesamt am größten ist. Man spricht auch von Unteranpassung.

Entscheidungsbäume: didaktisiertes Verfahren und Lernziele

In dem Lernszenario zu den Abbildungen 2-5 aus (Brandt, Eickhoff-Schachtebeck & Strecker 2022b) geht es darum, dass eine Software zukünftig automatisiert entscheiden soll, ob geerntet oder noch gewartet werden soll. Dazu sind die Entscheidungen eines erfahrenen Landwirts in der Vergangenheit protokolliert worden. Aus diesen Trainingsdaten erstellt eine Software automatisch einen Entscheidungsbaum, der dann für zukünftige Entscheidungen genutzt wird. Die Daten beinhalten drei Merkmale: die aktuelle Regenwahrscheinlichkeit, die aktuelle Feuchte des Getreides und die Entscheidung des Landwirts zur Ernte. Weizen ist unter einer Feuchte von 14,5% trocken und erntereif. Droht Regen, ist es aber manchmal besser, nicht ganz trockenes Getreide einzufahren (auch wenn dies beim Verkauf Abzüge bedeuten könnte), als noch größere Einbußen durch den eventuellen Regen in Kauf zu nehmen. Die Trainingsda-

ten sind in Abbildung 2 gegeben. Abbildungen 3 bis 5 (nächste Seite) zeigen die Erstellung eines Entscheidungsbaums per Hand, obwohl die Konstruktion des Baums eigentlich von einer Software automatisiert durchgeführt wird. Die Beschränkung auf drei Merkmale ermöglicht die Darstellung der Daten in einem zweidimensionalen Koordinatensystem und so einen enaktiven Zugang, bei dem das Verfahren der rekursiven Partitionierung angewendet werden kann. Diese Didaktisierung ergibt sich also aufgrund der gewählten Methodik.

Basierend auf den Trainingsdaten, die das Erfahrungswissen des Landwirts abbilden, werden nach und nach vertikale oder horizontale Bereiche ausgewählt, die Datenpunkte gleicher Farbe vereinen. Mithilfe des Entscheidungsbaums würde für die, nicht in den Trainingsdaten abgebildete, Situation einer Feuchte des Weizens von 15% und einer Regenwahrscheinlichkeit von 40% die Entscheidung „noch warten“ getroffen.

Im Vergleich zum Verfahren k-nächste Nachbarn ergibt sich Folgendes, wodurch teilweise auch ein allgemeines Konzept unabhängig vom konkreten Verfahren verdeutlicht werden kann:

- Die Bedeutung der Trainingsdaten kann hier ebenso gut gezeigt werden, wie beim Verfahren k-nächste Nachbarn
- Auch die Beschränkung auf wenige Merkmale und sich daraus ergebende Probleme kann hier gut thematisiert werden, ebenso aber auch beim Verfahren k-nächste Nachbarn.
- Entscheidungsbäume benötigen weniger eine Vereinheitlichung an den Achsen, da die Merkmale nicht in dem Sinne miteinander kombiniert werden wie beim euklidischen Abstand.

Weitere Lernziele, die anhand von Entscheidungsbäumen angestrebt werden können, wer-

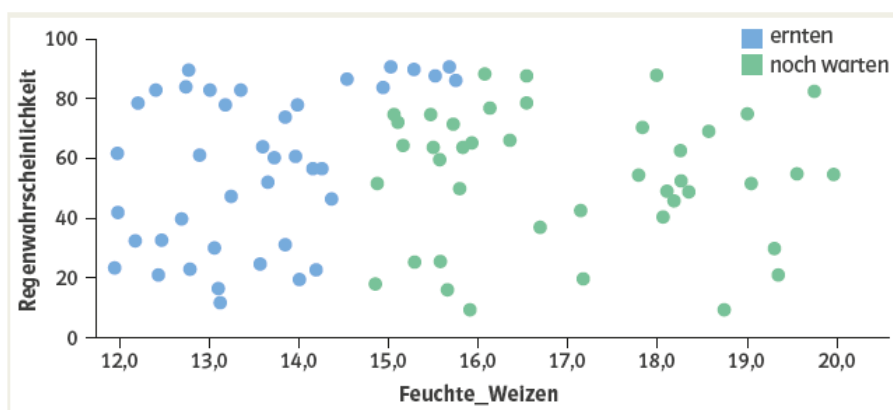


Abbildung 2: Trainingsdaten Ernteentscheidung

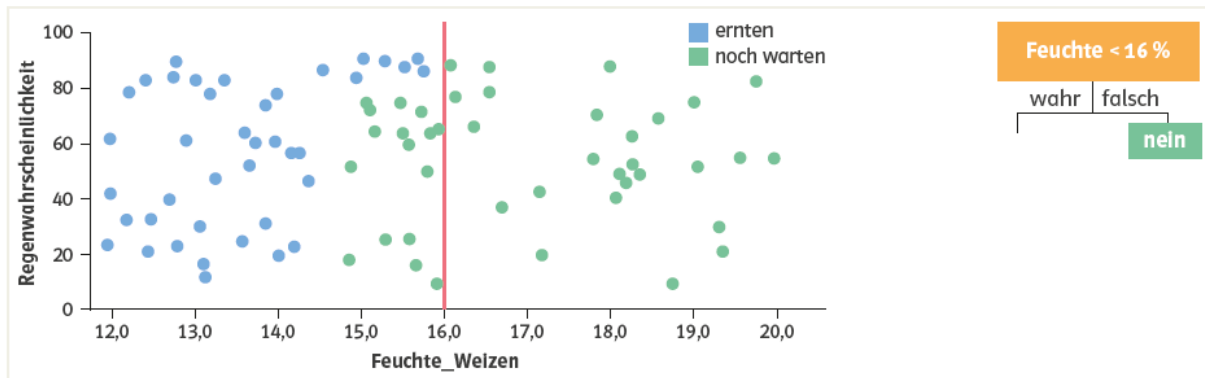


Abbildung 3: Erstellung eines Entscheidungsbaums

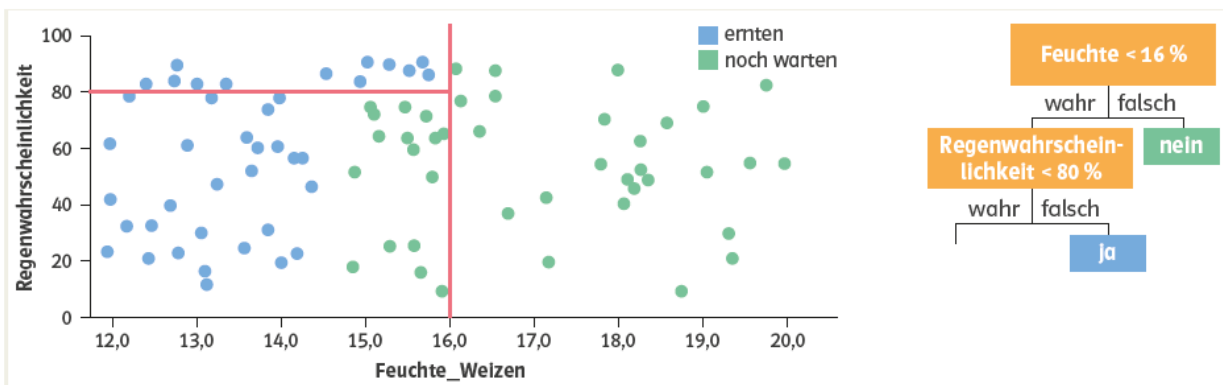


Abbildung 4: Erstellung eines Entscheidungsbaums

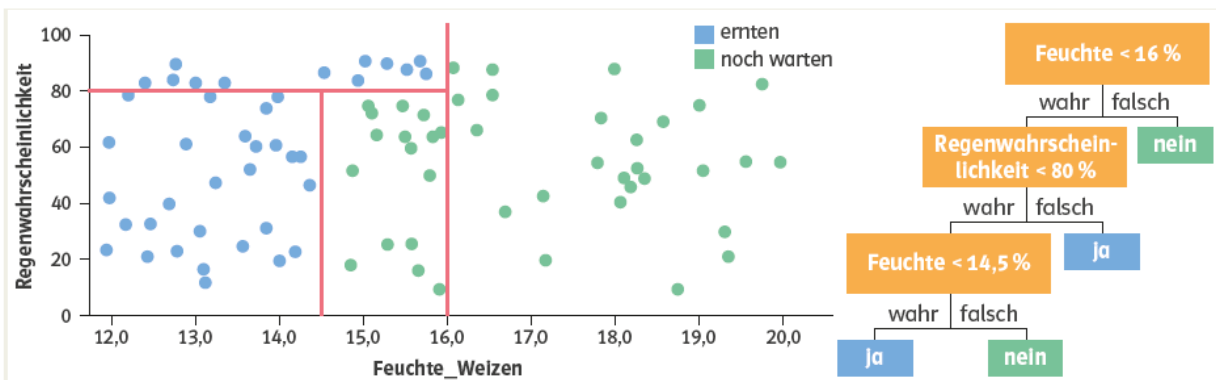


Abbildung 5: Erstellung eines Entscheidungsbaums

den deutlich, wenn man sich überlegt, was passieren würde, wenn es einen Ausreißer in den Daten gibt, bzw. (um bei unserer Geschichte zu bleiben) wenn der Mähdrescher an einem Tag nicht funktioniert hat und diese Entscheidung ebenfalls in die Trainingsdaten mit aufgenommen wurde (Abbildung 6).

- Der Entscheidungsbaum würde in dem Szenario in Abbildung 6 unnötig tief werden. Wir sprechen von Überanpassung. Lernende könnten den Algorithmus daraufhin dahingehend anpassen, dass eine Toleranz eingebaut wird, z.B. es werden horizontale oder vertikale Linien eingezeichnet, die Da-

ten gleicher Farbe vereinen, wobei z.B. 95% der Daten dieselbe Farbe haben müssen.

- Das Beispiel verdeutlicht, dass die Vorhersage eines Verfahrens des ML im individuellen Einzelfall falsch sein kann
- Vielleicht vermischt sich an dieser Stelle das Verfahren mit Vorkenntnissen, die die Lernenden im Themenbereich Algorithmik mit der Datenstruktur „Baum“ gesammelt haben. Deshalb ist es hier von Bedeutung, dass den Lernenden transparent wird, dass die Software den Entscheidungsbaum automatisch generiert, und zwar auf Basis von

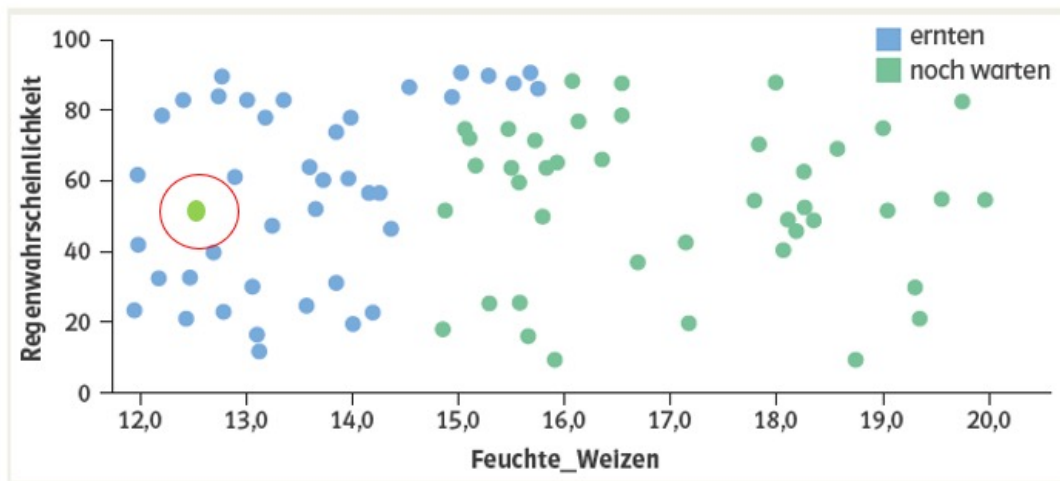


Abbildung 6: Ausreißer in den Daten

Trainingsdaten, und nicht der Programmierer oder die Programmiererin. Thematisiert man dies, wird der Unterschied zwischen einer regelbasierten und einer datenbasierten Strategie greifbar.

- Weiterhin wird bei diesem Verfahren sehr deutlich, dass die Ergebnisse und Vorhersagen des Verfahrens aus Daten der Vergangenheit berechnet werden.

Lineare Regression: Lernziele

Stellt sich bei den verwendeten Beispielen zu k-nächste Nachbarn und Entscheidungsbäumen das Muster in den Daten durch die Farbe der Datenpunkte dar und hilft damit bei der Kategorisierung ähnlicher, neuer Daten, so kann sich das Muster in Daten auch anders zeigen. Gibt es beispielsweise eine lineare Abhängigkeit der Ausgabedaten von den Eingabedaten, dann kann man mithilfe einer Regressionsgeraden Aussagen für neue, ähnliche Daten treffen. An dieser Stelle verweisen wir auf Beispiele aus dem Mathematikunterricht. Die Thematisierung der linearen Regression als Beispiel für ein Verfahren des überwachten Lernens kann aber sehr gut genutzt werden, um den Unterschied zwischen Korrelation und Kausalität zu verdeutlichen. Wird z.B. das prominente Beispiel der Korrelation zwischen der Anzahl der Geburten und der Anzahl der Storchenpaare (siehe z.B. wikipedia 2025) verwendet, das auch eine lineare Regressionsgerade zeigt, kann der Unterschied zwischen Korrelation und Kausalität thematisiert werden, da Verfahren des ML nur Korrelationen in Daten identifizieren können. Für Unterrichtsideen verweisen wir auf (Eickhoff-Schachtebeck & Strecker 2025).

Neuronale Netze: didaktisiertes Verfahren und Lernziele

Das Themengebiet maschinelles Lernen ohne neuronale Netze zu thematisieren, erscheint in der Übermächtigkeit des Verfahrens in der Realität schwierig, ist aber didaktisch eine Herausforderung, da das Prinzip der Backpropagation mathematisch komplex ist. Da ein einzelnes Neuron nicht so mächtig ist wie ein neuronales Netz, sollte auch die Funktionalität eines kleinen Netzes thematisiert und verstanden werden. Wir skizzieren an dieser Stelle das virtuell-entdeckte der Funktionsweise neuronaler Netze aus (Brandt, Eickhoff-Schachtebeck & Strecker 2022b) und verweisen zur näheren Erläuterung der hier verwendeten didaktischen Reduktion neuronaler Netze auf (Eickhoff-Schachtebeck & Strecker 2025).

In all unseren folgenden Beispielen verwenden wir die Methode des virtuell-entdeckten Erkundens. Wir präsentieren jeweils eine interaktive Simulation einzelner Neuronen oder kleiner neuronaler Netze, in der die Lernenden aktiv Eingaben verändern können. Durch Beobachtung der Ergebnisse und Reaktionen kann sich so zumindest in Teilen die Funktionsweise erschlossen werden. Die interaktiven Simulationen sind in Scratch implementiert, wodurch der Programmcode von den Lernenden nachvollzogen werden kann. Durch die Bereitstellung einer Implementierung wird gehofft, dass bei den Lernenden zumindest das Gefühl entsteht, einer Implementierung grundsätzlich „gewachsen“ zu sein. Mindestens wird aber deutlich, dass zur Implementierung neuronaler Netze nur algorithmische Grundbausteine von Scratch verwendet werden und keine „Magie“.

An einem einzelnen Neuron kann zunächst gut thematisiert werden, wie sich die Ausgabe aus

Eingaben, Gewichten und Schwellenwert berechnet, wenn wir folgende Didaktisierung zugrunde legen: Das Neuron feuert (Ausgabe dabei wird von dem Programmierer oder der Programmiererin festgelegt), wenn die Summe der Produkte aus Eingabe und dem zugehörigen Kantengewicht einen Schwellenwert erreicht bzw. überschreitet. Sonst feuert das Neuron nicht (Ausgabe dabei wird von dem Programmierer oder der Programmiererin festgelegt.).

Das vorrangige und erste Lernziel sollte sein, dass sich die Funktionalität eines Neurons auch durch die Wahl der Gewichte ergibt. Ändert man die Gewichte eines Neurons oder neuronalen Netzes, ändert sich u. U. auch dessen Funktionalität, was im ersten Schritt auch händisch an einem konkreten Beispiel überlegt werden kann (siehe Eickhoff-Schachtebeck & Strecker 2025), das wir im Folgenden kurz vorstellen. Es geht um den Kontext, dass eine Gärtnerei ihre Freilandpflanzen automatisch bewässern möchte. Es gibt drei verschiedene Gärtnerarten:

Die erste Gärtnerin hat folgende Anforderung: Es soll immer dann bewässert werden, wenn es sehr unwahrscheinlich ist, dass es regnen wird. Egal wie trocken der Boden ist. Die neu gepflanzten Pflanzen brauchen viel Wasser.

Der zweite Gärtner wünscht sich von der automatischen Bewässerung folgendes: Da Wasser gespart werden soll, soll immer nur dann bewässert werden, wenn es sehr unwahrscheinlich ist, dass es regnen wird und gleichzeitig der Boden sehr trocken ist. Und der dritte Gärtner hat diese Wünsche: Es soll immer dann bewässert werden, wenn der Boden sehr trocken ist. Auf die Wettervorhersage verlässt er sich nicht.

Es wird deutlich, dass die Funktionsweise des Neurons von der Wahl der Gewichte abhängt, denn bei unterschiedlicher Wahl der Gewichte reagiert die Bewässerungsanlage auf dieselben Eingaben jeweils verschieden. In den Abbildungen 7 und 8 sind deshalb dieselben Eingabewerte verwendet worden. Die Wahl der Gewichte passt in Abbildung 7 zu den Wünschen der ersten Gärtnerin und in Abbildung 8 zu den Wünschen des dritten Gärtners. Die Wahl der Gewichte für das zweite Szenario kann von den Schülerinnen und Schülern überlegt werden.

In einem zweiten Schritt muss transparent gemacht werden, dass das Einstellen der Gewichte eines Neurons nicht händisch passiert, sondern die Gewichte automatisch in einer Trainingsphase angepasst werden. Dafür ist z.B. ein Fehlersignal notwendig, wenn die Gewichte noch nicht optimal eingestellt sind.

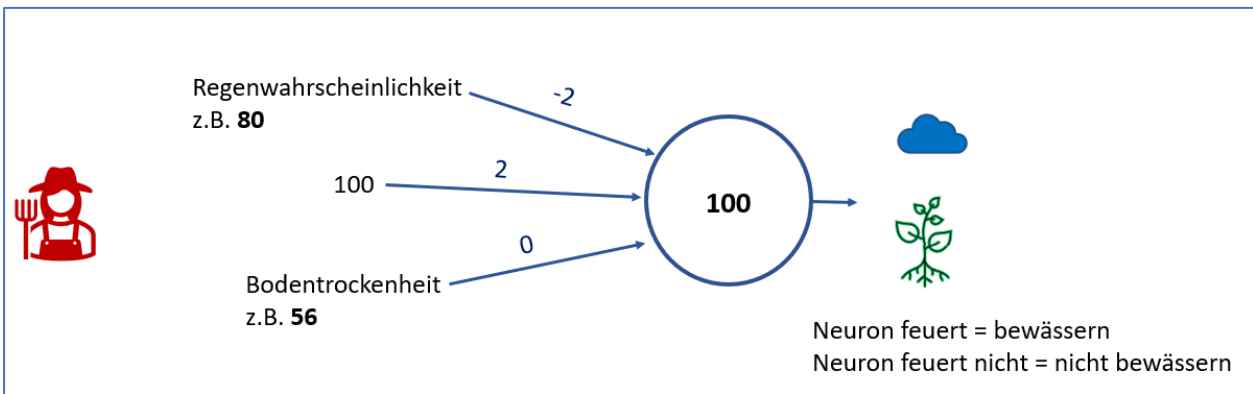


Abbildung 7: Erkunden der Funktionsweise eines Neurons

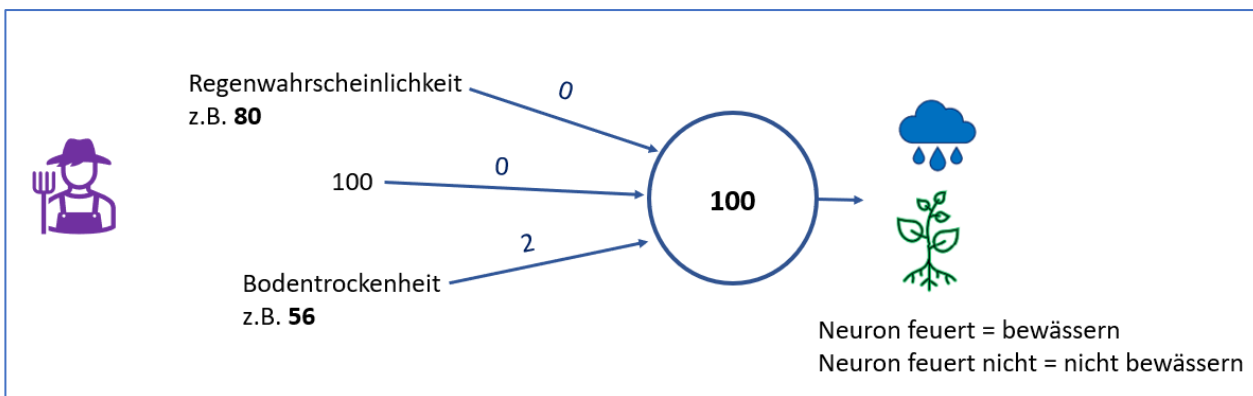


Abbildung 8: Erkunden der Funktionsweise eines Neurons

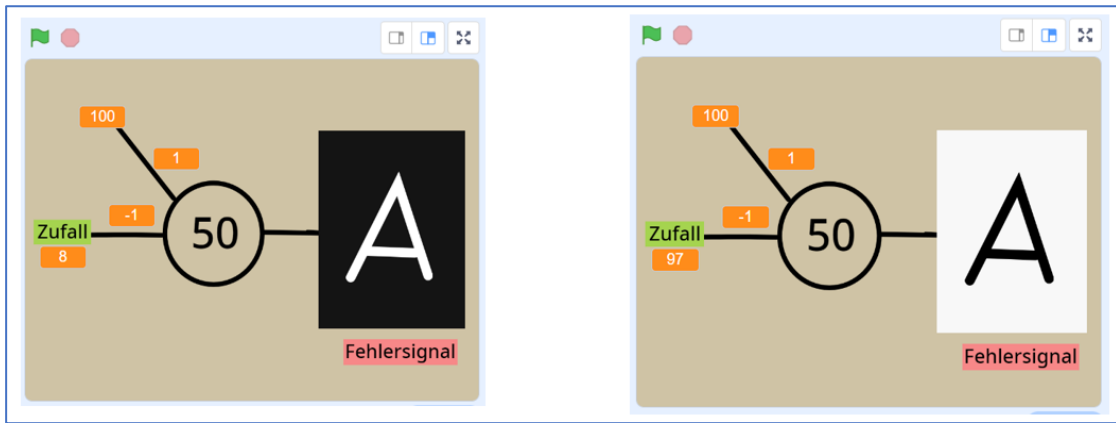


Abbildung 9: Wahl einer Schriftfarbe bei dunklem und hellem Hintergrund

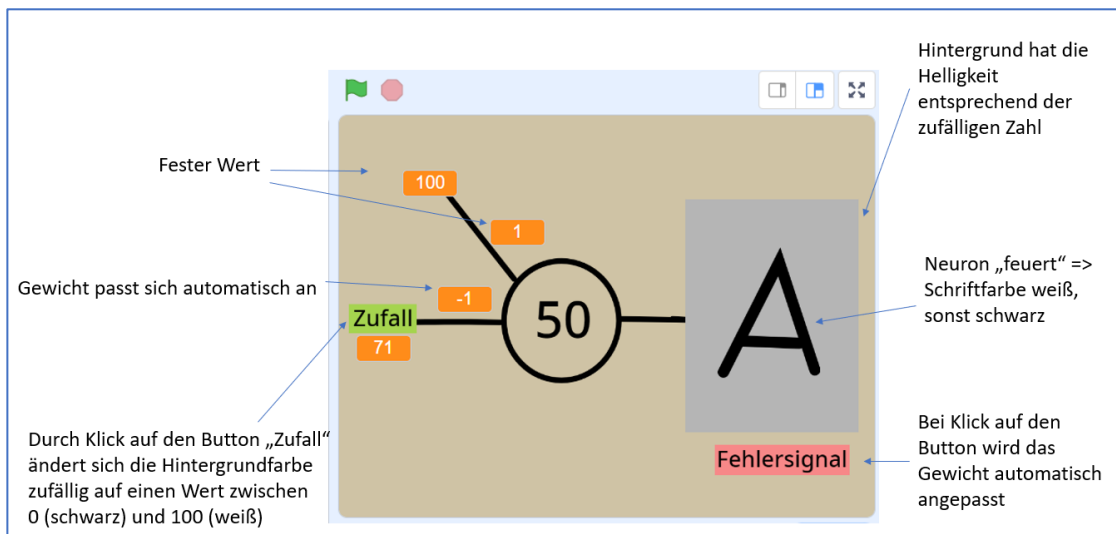


Abbildung 10: Scratch-Simulation zur automatischen Wahl einer Schriftfarbe

Auf diesen Aspekt geht das folgende Beispiel u.a. ein: Für dunklen Hintergrund eignet sich helle Schriftfarbe, für hellen Hintergrund dunkle Schriftfarbe. Bei Grautönen dazwischen können die Empfindungen individuell unterschiedlich sein.

Die Änderung des Gewichts beläuft sich dabei auf den Lernalgorithmus in Abbildung 11 und findet sich im Skript zum Button „Fehlersignal“:



Abbildung 11: Skript zum Ändern des Gewichts

Für unsere nächsten beiden Beispiele verwenden wir folgenden Kontext: Es geht um eine automatische Fenstersteuerung im Smart-Home. Es gibt zwei Sensoren: ein Sensor, der angibt, ob es draußen laut (Lautstärke=1) oder leise (Lautstärke=-1) ist und ein Sensor, der angibt, ob es draußen hell (Lichtstärke=1) oder dunkel (Lichtstärke=-1) ist. Das neuronale Netz, bzw. ein Neuron liefert den Wert 1, wenn es feuert und sonst den Wert -1. Die Ausgabe des Netzes bewirkt ein automatisches Öffnen des Fensters (wenn das Neuron der letzten Schicht feuert) oder ein Schließen des Fensters (wenn das Neuron der letzten Schicht nicht feuert). Die Begründung für die Implementierung mit einem Verfahren des maschinellen Lernens, entgegen der mit individuellen regelbasierten Algorithmen für jeden der Bewohnerinnen und Bewohner des Hauses wollen die automatische Fensteröffnung in ihrem Schlafzimmer installieren, aber jede der Personen hat andere Wünsche, unter welchen Bedingungen sich das Fenster öffnen oder schließen soll. Implementen-

tiert man das System mit einem neuronalen Netz, kann jeder der Bewohnerinnen und Bewohner es den eigenen Wünschen entsprechend trainieren und so liefert derselbe Algorithmus (durch jeweils unterschiedliche Gewichte nach der Trainingsphase) bei jeder Person andere Ergebnisse. Funktioniert die automatische Fensteröffnung noch nicht wie gewünscht, wird das Fenster „manuell“ mit Klick auf das Fenster in der Simulation geöffnet oder geschlossen. Diese „manuelle“ Betätigung des Fensters ist das Fehlersignal, woraus geschlossen werden kann, wie die Gewichte verringert oder erhöht werden müssen.

Für ein einzelnes Neuron verwenden wir zur Anpassung der Gewichte die folgenden Lernalgorithmen mit einer Lernrate von 0,5. (Eine Erläuterung findet sich in (Eickhoff-Schachtebeck & Strecker 2025)).

alle folgenden Wünsche der Bewohnerinnen und Bewohner realisieren zu können:

Onkel Andreas: „Ich arbeite Nachtschicht. Mein Fenster soll nachts zu sein, damit keine Einbrecher einsteigen. Tagsüber schlafe ich. Da soll das Fenster nur offen sein, wenn es draußen leise ist.“ (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Oma Emmi: „Mein Fenster soll nachts offen sein, wenn es leise draußen ist, damit ich gut schlafen kann. Bei Lautstärke kann ich nicht schlafen. Tagsüber soll das Fenster offen sein, wenn die Enkel im Garten spielen und es laut ist, damit ich sie hören kann. Ist es tagsüber leise, soll das Fenster zu bleiben.“ (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Lin: „Ich kann nur schlafen, wenn das Fenster nachts offen und es leise ist. Bei Lärm soll das Fenster nachts zu bleiben. Tagsüber soll das Fenster zum Lüften aber immer offen stehen,



Abbildung 12: Lernalgorithmen

Die Simulation stellt sich den Lernenden wie

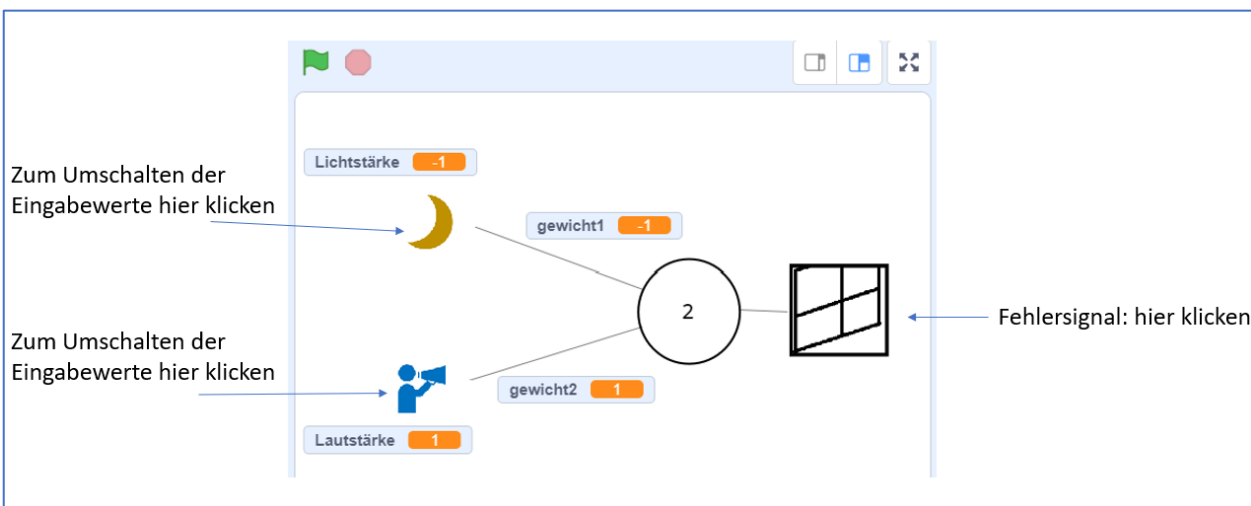


Abbildung 13: Scratch-Simulation zur automatischen Fensteröffnung mit einem Neuron

folgt dar:

Allerdings lassen sich mit einem Neuron nicht alle möglichen Wünsche und Einstellungen realisieren. Ein neuronales Netz ist notwendig, um

egal wie laut oder leise es ist. (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Papa Erik: „Ich bin tagsüber immer bei der Arbeit und daher soll das Fenster tagsüber zu sein.

Nachts soll es aber zum Schlafen offen sein, egal wie laut oder leise es ist.“ (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Wir verweisen an dieser Stelle darauf, dass die Anforderungen von Oma Emmi einem NOT-XOR entsprechen und damit nur mit einem neuronalen Netz implementiert werden können, nicht mit einem einzelnen Neuron.

Wir verwenden ein didaktisiertes neuronales Netz mit festen Kantengewichten in der letzten Schicht. Die Simulation stellt sich den Schülerinnen und Schülern wie folgt dar:

Durch die festen Gewichte in der letzten Schicht kann mit ähnlichen Lernalgorithmen wie in Abbildung 12 nachvollzogen werden, welche Gewichte bei einem Fehlersignal wie verändert werden müssen. Gleichzeitig können für alle denkbaren Szenarien und Wünsche der Bewohner:innen pas-

angepasst werden und die Wahl der Gewichte die Funktionalität des Netzes bestimmt, bedeutet das in der Konsequenz folgendes: wenn in der Trainingsphase ungenügende oder ungeeignete Daten verwendet werden, ist die Vorhersage des Netzes für ähnliche Daten schlecht.

- Es wird deutlich, dass das „Lernen“ eines Neurons dem automatischen Anpassen der Gewichte entspricht.
- Es wird ebenfalls deutlich, dass neuronale Netze nur diejenigen der Aufgaben sehr gut lösen können, für die sie trainiert wurden.
- Ein virtuell-enaktives Erkunden des Verfahrens neuronale Netze zeigt deutlich, dass das Verfahren algorithmisch beschreibbar ist, also nur algorithmische Grundbausteine verwendet werden und keine „Magie“.
- Skaliert man das neuronale Netz und spricht z.B. von 100 Milliarden Neuronen, dann wird

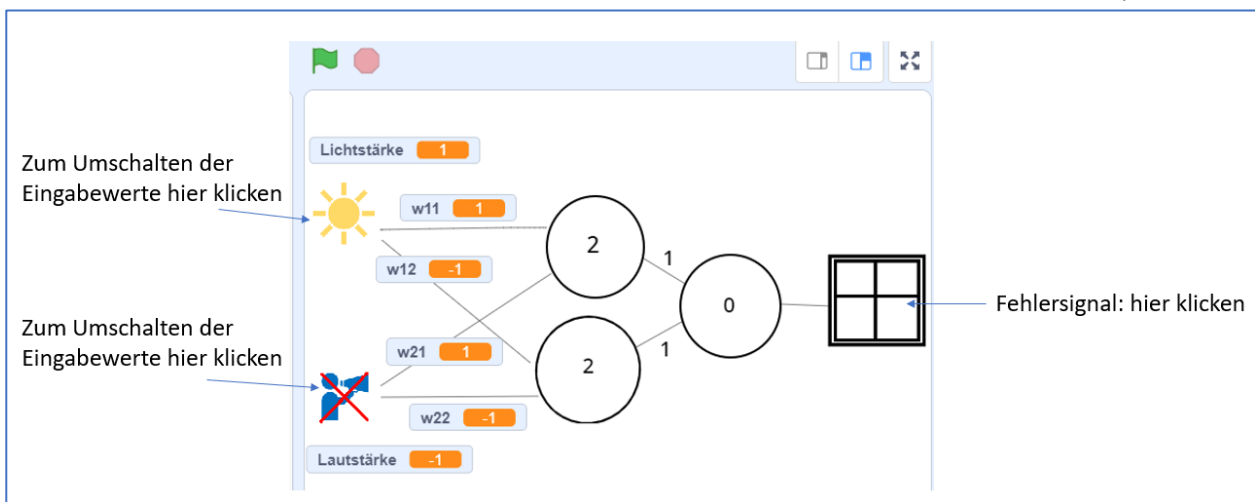


Abbildung 14: Scratch-Simulation zur automatischen Fensteröffnung mit einem neuronalen Netz

sende Gewichte automatisch „gelernt“ werden. Die theoretischen Hintergründe werden in (Eickhoff-Schachtebeck & Strecker 2025) beschrieben.

Das Erkunden der Funktionsweise eines didaktisch reduzierten neuronalen Netzes ermöglicht folgende Reflexion und hat dementsprechend folgende Lernziele:

- Die Trainingsphase wird hier sehr gut greifbar. Während bei den anderen vorgestellten Verfahren die Trainingsphase und die Anwendungsphase deutlich getrennt waren, sieht man hier, dass Trainings- und Anwendungsphase theoretisch auch miteinander verzahnt ablaufen können.
- Man sieht weiterhin sehr gut, dass die Trainingsdaten die Funktionalität bestimmen. Da die Gewichte aufgrund der Trainingsdaten

deutlich, dass das Ergebnis eines Netzes für den Menschen nicht nachzurechnen ist. Es ist also nicht nachvollziehbar, wie das Netz zu seinem Ergebnis kommt.

Zusammenfassung der Lernziele konkreter Verfahren und Fazit

In der Einleitung haben wir gefordert, dass sich die Auswahl und Tiefe konkreter Verfahren auch daraus bestimmen sollte, welche dahinterstehenden allgemeinbildenden Lernziele abgedeckt werden. Wir fassen solche Lernziele unserer Beispiele zusammen und nennen dahinter die Verfahren, mit denen die Lernziele unserer Meinung nach besonders gut erreicht werden können:

- Entmystifizierung des Begriffs der künstlichen Intelligenz (alle)
- „Was heißt Lernen?“ (alle)
- Entscheidung beruht auf Daten der Vergangenheit (Entscheidungsbäume, k-nächste Nachbarn)
- Unterschied Ausreißer und Muster (Entscheidungsbäume, k-nächste Nachbarn)
- Ergebnisse können im individuellen Einzelfall falsch sein (k-nächste Nachbarn, Entscheidungsbäume)
- Unterscheidung regelbasierte und datenbasierte Systeme (k-nächste Nachbarn, neuronale Netze)
- Überanpassung (Entscheidungsbäume, k-nächste Nachbarn)
- Unteranpassung (k-nächste Nachbarn)
- Notwendigkeit von Trainingsdaten (alle)
- Trainingsdaten bestimmen die Ergebnisse (alle)
- Notwendigkeit der Datenaufbereitung (k-nächste Nachbarn)
- Geeignete Merkmale der Daten identifizieren (alle)
- Unterschied zwischen Korrelation und Kausalität (lineare Regression)

Wir haben drei Verfahren des überwachten Lernens im Informatikunterricht etwas genauer vorgestellt und hoffen, die dahinterstehenden Lernziele transparent gemacht zu haben. Dabei betonen wir ausdrücklich, dass diese Liste nicht vollständig ist. Sie lässt sich insbesondere dann erweitern, wenn weitere Vorgehensweisen des ML hinzugenommen werden. Betrachten wir stellvertretend das verstärkende Lernen, z.B. im Unterricht konkretisiert mit dem Q-Lernen-Verfahren (Eickhoff-Schachtebeck & Strecker 2025). Beim verstärkenden Lernen wird nicht vorgegeben, wie ein Problem gelöst werden soll, sondern es werden nur erwünschte Ergebnisse belohnt. Die Schülerinnen und Schüler machen im Informatikunterricht vor allem beim algorithmischen Problemlösen die Erfahrung, dass es unterschiedliche Lösungen für eine Problemstellung gibt, also Algorithmen nicht alternativlos sind. Wenn beim verstärkenden Lernen nur das Ergebnis belohnt wird, kann reflektiert werden, dass der „gelernte“ Weg dorthin auch ungewöhnlich sein kann oder vielleicht sogar unerwünschte Nebeneffekte hat, da nur festgelegt wird, welches Ergebnis belohnt und nicht wie es erreicht wird.

Auch das unüberwachte Lernen (z.B. konkretisiert mit dem k-Means-Clustering-Algorithmus) (siehe z.B. Eickhoff-Schachtebeck & Strecker 2025) fordert eine Reflexion über Ähnlichkeit in Daten und den Zusammenhang zwischen Ähnlichkeit und Abstand. Das wiederum kann aufgegriffen werden, wenn über eine Modellierung von Semantik

der Daten reflektiert wird, z.B. bei der Thematisierung von Grundprinzipien großer Sprachmodelle (siehe Eickhoff-Schachtebeck & Strecker 2025).

Auch wenn die vorgestellte Liste nicht vollständig ist, soll sie aber einen Anlass geben, bei der Integration des Themenbereichs maschinelles Lernen in den Informatikunterricht neben den konkreten Verfahren auch die allgemeinbildenden Aufgaben von Schule im Blick zu behalten. Für die einzelnen Verfahren sollte neben der didaktischen und methodischen Aufbereitung also stets mitgedacht werden, welche Lernziele auf der Reflexionsebene langfristig für die späteren mündigen Bürgerinnen und Bürger relevant und gewinnbringend sind und zwar unabhängig von verfahrensspezifischen Besonderheiten.

Material

Fenstersteuerung mit Neuron in Scratch:
<https://scratch.mit.edu/projects/1128527593>

Fenstersteuerung mit Neuronalem Netz in Scratch:
<https://scratch.mit.edu/projects/1128528506>

Links zuletzt geprüft: 19.02.2025

Quellen

Eickhoff-Schachtebeck, A. und Strecker, K. (2025, im Erscheinen): „Themenheft starkeSeiten Künstliche Intelligenz und Maschinelles Lernen“, Klett-Verlag, ISBN: 978-3-12-007626-1

Brandt, Y., Eickhoff-Schachtebeck, A. und Strecker, K. (2022a): „Schulbuch starkeSeiten Informatik Jahrgang 9/10 Differenzierende Ausgabe Niedersachsen“, Klett-Verlag, ISBN: 9783120075707

Brandt, Y., Eickhoff-Schachtebeck, A. und Strecker, K. (2022b): „Schulbuch starkeSeiten Informatik Jahrgang 9/10 Gymnasium Niedersachsen“, Klett-Verlag, ISBN: 9783120075721

wikipedia (2025): URL: <https://de.wikipedia.org/wiki/Scheinkorrelation> (Zugriff: 23.1.25)

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Annika Eickhoff-Schachtebeck
Regionales Landesamt für Schule und Bildung
Braunschweig
annika.eickhoff-schachtebeck@rlsb.de

Kerstin Strecker
Georg-August Universität Göttingen
kerstin.strecker@informatik.uni-goettingen.de