

Algorithmen sind nicht schuld, aber wer oder was ist es dann?

1. Einleitung

Algorithmen sind ein zunehmend in den Medien, auch den seriösen und wissenschaftlichen Medien, aufgeworfener Hype. Man fühlt sich von ihnen versorgt, umhegt, gesteuert, überwacht, betrogen, bedrängt usw. Es wird ihnen Handlungs- oder Entscheidungsmacht zugebilligt, sie werden für organisatorische, institutionelle und politische Steuerungen, ja auch für Wahlausgänge verantwortlich gemacht. Doch werden dabei die Begriffe Computerprogramme/Software, Maschinen, Künstliche Intelligenz (KI), Big Data etc., auch wenn dort überall Algorithmen benutzt werden, mit Algorithmen unzulässig identifiziert. Es ist, als wollte man von der Addition Moral verlangen. Algorithmen sind Rechenvorschriften, also formale Anweisungen zur Ausführung mathematischer Funktionen, denen all diese sozialen Funktionen und Zuschreibungen fremd sind. Sie können korrekt, effizient, sparsam, schnell, auch adaptiv bzw. „lernfähig“ sein, aber sie sind weder objektiv noch intelligent, weder gut noch böse. Warum werden ihnen fälschlich soziale oder ethische Qualitäten zugebilligt? Weil sie sozusagen den Kern der sehr wohl werthaltigen Automatisierung ausmachen? Weil „Algorithmus“ bedeutsamer oder auch sexier klingt als Software?

Computer haben den Weg für den massenhaften Einsatz von Algorithmen eröffnet. Damit werden sie, in Datenstrukturen und Software gerahmt, mit Datenmassen gefüttert, „unsichtbare“ Technologie, ja mental, methodisch und physisch unsichtbare Technologie, die wir benutzen, fast ohne sie in Frage zu stellen.¹ Es sind immer menschliche Entscheidungen, wie Software spezifiziert wird, welche Algorithmen dafür ausgewählt und kombiniert werden, welche Datenangebote ihnen zukommen, und die so die Technologie zu etwas sozial, politisch und ethisch Relevantem machen. Leider wird in der Regel nicht demokratisch entschieden, ob und wie welche Technologie eingesetzt wird. Die meisten Menschen können auch nicht selbst bestimmen, welche Technik sie benutzen, um mit ihr etwa Software-Produkte herzustellen. Der soziale Druck zwingt ja vor allem viele junge Menschen, Nutzende zu werden, um sich nicht selbst ins Abseits zu stellen.

Die rasante Entwicklung der Automatisierung wird zu Recht als Bedrohung empfunden. Doch Schaden, Diskriminierungen und Katastrophen entstehen durch die Rahmung der Algorithmen in Software, mit Design, Interfaces und ihrer Amalgamierung mit kontingenten Daten anstelle der Algorithmen eigenen Univer-

salität, mit den dabei vielfältig verfolgten Absichten. Dieser Text versucht, die Begriffe Algorithmus, Programm, Software und Daten klarer auseinander zu halten, und dabei die Orte aufzuzeigen, wo und wie mittels Computer-Software Wertsetzungen, Priorisierungen, Ausschlüsse und Diskriminierungen für Menschen und soziale Systeme in die Welt gesetzt werden (können). Dabei erscheint die Forderung nach ethischen und sozialen Algorithmen als grundlegendes Missverständnis und problematisch, weil so die Verantwortung auf formal Mathematisches verlagert wird, als unveränderlich erscheint und die menschliche Beteiligung verschleiert.

2. Algorithmen

Algorithmen sind dem Menschen „zuhandenes“ Werkzeug. Softwareprozesse organisieren eine Kombination von Algorithmen. Dabei wird bestimmt, welche Algorithmen in welcher Reihenfolge genutzt und mit welchen Argumenten aufgerufen werden sollen. Das legen Menschen fest, die ein Ziel haben. Dabei muss hinterfragt werden, wer mit der zu erstellenden Software welchen Zweck verfolgt, ob, wo, wie und in welchem Kontext diese Algorithmen als Werkzeug eingesetzt werden und ob das gebilligt werden kann. Algorithmen sind nicht fähig moralische oder politische Ziele zu erwägen oder zu beachten. Sehr wohl kann dies alles aber unsere Instandsetzung von Softwaresystemen und unser Umgang mit Automatisierung. Softwareprogramme mitsamt den in ihnen organisierten Algorithmen sind, wie die Informatikerin und Genderforscherin Cecile Crutzen schreibt, Werkzeuge, die zum Handeln bereitliegen, unsere Hand führen, (Arten der) Nutzung nahelegen, Handlungen beeinflussen oder insbesondere als *Gadgets* sogar zum Handeln auffordern. Diese zweckbestimmte Organisation in Software kann natürlich moralische Fragen aufwerfen, ja sie tut dies mit der für die informationstechnische Modellierung notwendigen Abstraktion sogar notwendigerweise. Abstraktion impliziert immer auch Weglassen, und dieses Ausschließen ist subjektiv. Ebenso wenig lassen sich die Ergebnisse des Ablaufs von Software und durch sie bedingte Veränderungen der Umgebungen eindeutig vorhersehen. Dies auch deshalb, weil sich die Anforderungen und Einsatzumgebungen ändern oder Software umgenutzt wird für Verwendungszwecke, für die sie ursprünglich nicht gedacht war.

Ein Beispiel: Angenommen, man kann an jedem Tag zwei Leute retten, wie viele sind das in drei Wochen? Antwort: 42. Ein Al-



Britta Schinzel

Britta Schinzel promovierte in Mathematik, arbeitete in der Computerindustrie und habilitierte sich in der Informatik. Im Rahmen ihrer Professur für Theoretische Informatik an der RWTH Aachen arbeitete sie zunehmend interdisziplinär. Sie war von 1991 bis 2008 Professorin für *Informatik und Gesellschaft und Gender Studies in Informatik und Naturwissenschaft* an der Universität Freiburg.

gorithmus wird die Eingaben zwei und drei variabel halten, er ist so universell auf allen natürlichen Zahlen. Um einen Algorithmus anwenden zu können, muss man erklären, dass eine Woche 7 Tage hat, aber nicht, was *retten* heißt. Wenn etwa *retten* durch *ertrinken lassen* ersetzt wird, führt das auf den gleichen Algorithmus. Algorithmen können also für ganz verschiedene Zwecke eingesetzt werden. Wenn man die Namensgebung in einem Softwareprogramm ändert, wie im Beispiel, so erscheint das Programm für die Nutzer in einem ganz anderen Licht. Das Programm selbst und der Algorithmus aber haben auf diese Sichten keinen Einfluss.

Algorithmen operierten historisch gesehen zuerst auf Zahlen; sie wurden benannt nach dem persischen Mathematiker Al Chwarizmi aus dem 9. Jhd. nach Chr., der auch die 0 erfunden hat. Später wurden sie auf viele andere Gebiete erweitert, die einer mathematischen Manipulation zugänglich waren, u. a. auf Worte, auf Sätze einer Sprache, auf Musiknotate und Musik, auf Graphiken etc. Präzisierungen des Algorithmusbegriffs wurden in Anschluss an Hilberts Versuch der Formalisierung der Mathematik von Kurt Gödel mit der Formalisierung des Berechenbarkeitsbegriffs getroffen. Das hat sehr umfangreiche mathematische Folgerungen und Theorien zur Folge gehabt, die sich u. a. mit den Fragen, was man überhaupt, und dann mit welchem Aufwand an Zeit und Speicherplatz, berechnen kann.

Für schnelle Abläufe im Inneren von Rechnern, wie etwa für Compiler, Betriebssystem und BIOS, bedarf es dabei spezieller Algorithmen, die rasche Antwortzeiten auf Eingaben garantieren. Ähnliches gilt für Sortier- und Suchalgorithmen, wie sie etwa von Suchmaschinen verwendet werden, und für manche Arten von Datenanalyse-Algorithmen.

Algorithmen finden sich mit den verschiedensten Eigenschaften in allen Computerprogrammen und Software-Anwendungen. Wichtige und effiziente Standardalgorithmen werden aus Algorithmenbanken gezogen. Betriebssysteme benötigen Warteschlangen-Algorithmen für die Abwicklung von Daten-Paketen. Compiler und – besonders wichtig für Software-Anwendungen – Sortier- und Such-Algorithmen operieren auf Zeichenketten, und letztere über hierarchisierte Anordnungen in Bäumen oder Graphen. Kürzeste Wege in Graphen sind grundlegend für unsere Navigationssysteme. Geometrische Algorithmen operieren auf diskretisierten Kurven, Flächen oder 3D-Körpern. Approximationsalgorithmen, wie Archimedes' Algorithmus zur Berechnung beliebig vieler Stellen von π , nähern sich in der Genauigkeit *irrationalen* Zahlen. Kompressionsalgorithmen wie MP3 verkleinern Bilder- oder Musikdatenmengen. Kryptografische Algorithmen verschlüsseln Daten, heute meist auf der Basis der komplex zu berechnenden Primzahlfaktorisation zweier zuvor miteinander multiplizierter Primzahlen. Es gibt biometrische Algorithmen zur Analyse von DNA- oder Protein-Sequenzen, Clusteralgorithmen klassifizieren Mengen oder Muster. Es gibt genetische und evolutionäre Algorithmen, die nicht in gleicher Weise determiniert sind wie die zuvor erwähnten geschlossenen Algorithmen. Künstliche Neuronale Netze klassifizieren oder simulieren natürliche Systeme und werden für Lernalgorithmen, und in geschichteten Kaskaden für „Deep Learning“ verwendet. Zufallsalgorithmen arbeiten mit Zufallszahlen aus Zufallszahlengeneratoren zur Bearbeitung unterschiedlichster Probleme, am bekanntesten die Las-Vegas- und Monte-Carlo-Algorithmen.

Data-Mining- und *Data-Analytics*-Algorithmen durchforsten einen nicht endenden Strom von Big Data und ziehen aus diesen Schlüsse, bereiten Entscheidungen vor, Entscheidungshilfen treffen solche sogar. Und natürlich kann man darin indirekt Meinungen unterbringen und bestimmte Absichten verfolgen. Bei heutigen Suchmaschinen, Big Data, Data Mining, Machine Learning, Social Bots etc. erscheinen oft Daten mit Algorithmen untrennbar verschmolzen, sodass leicht begriffliche Verwechslungen entstehender Softwaresysteme mit Algorithmen entstehen können. Doch die Verschmelzungsprodukte sind nicht mehr universell, keine Algorithmen mehr, sondern Software.

3. Wie entstehen Bias, Ungleichgewichte, Wertsetzungen, diskriminierende Hierarchien oder Auslassungen in Software?

3.1 Das Entwerfen von Software

Wenn man einen diskriminierenden Text hat, so schreibt man die Ursache dafür nicht der Verwendung von Buchstaben oder Zahlen zu. Nicht anders verhält es sich mit der Verwendung von Algorithmen. Sie definieren mathematische Funktionen, die weder der Moral, der Diskriminierung noch der Klugheit oder analoger ethischer, emphatischer oder auch intelligenter Eigenschaften fähig sind. Jedoch werden sie über die Software-Anwendungen zur Ausführung an Computern in unsere Lebenswelt eingebettet. Menschen mit ihren Vorverständnissen, blinden Flecken und den beschränkten Möglichkeiten, das Feld zu verstehen, in dem ihre Software-Programme leben und einwirken werden, müssen den zu bearbeitenden Problembereich spezifizieren, „rational rekonstruieren“. Sodann entwickeln sie solche, das rekonstruierte Feld verändernde Problemlösungen, mit Design, Modellierung, Implementierung, Benutzungsschnittstelle, Testen und Dokumentation. Für „geschlossene“ Programmlösungen wurden solche Mechanismen bei der Software-Entwicklung während der 1980er- und 1990er-Jahre in den Bereichen von Informatik und Gesellschaft, den Arbeits- und Organisationswissenschaften ausgiebigst diskutiert und erforscht. Nebst dem Zeit- und Kostendruck, unter dem die Spezifikation des zu erstellenden Programms in Zusammenarbeit zwischen Auftraggebern, prospektiven Nutzenden und Entwickelnden definiert wird, ist oft der von Gerhard Wohland u. a. so genannte „ego-approach“² von Programm-Entwicklern ein Fehler und Bias erzeugendes Problem.

Die Spezifikation oder das Pflichtenheft für eine solche Software in einer Software-Umgebung wird von Menschen über das Requirements Engineering hergestellt. Damit wird die Aufgabe oder die Problemstellung eruiert, d. h. der Weltausschnitt, in dem das prospektive Programm operieren soll, abgegrenzt und, soweit für die Lösung nötig, definiert. Dabei muss die Komplexität unserer Lebenswelt reduziert werden, durch Dekontextualisierung, Vereinfachung und Abstraktion. Auch ist es nicht gleichgültig, mit welchen Zielen Software spezifiziert und modelliert wird. Wissen und Software-Handeln sind immer situiert.³ So sind bei klassischer Softwareentwicklung für geschlossene Lösungen vorwiegend die Auftraggebenden, die beschränkten Ressourcen und die meist unter Zeitdruck stehenden Entwickelnden für zuweilen problematische Problemlösungen verantwortlich. Ein-

fluss haben aber auch die Software-Engineering-Modelle, wie das klassische Wasserfallmodell oder die inkrementelle Entwicklung, oder die agile Softwareentwicklung, die nicht mehr von geschlossenen Lösungen ausgeht, sondern auf veränderliche Anforderungen durch adaptive Methoden reagiert.

3.2 Programmiersprachen

Nicht ganz so „unschuldig“ wie Algorithmen sind auch die Programmiersprachen, Programmierumgebungen und integrierten Rechner-Software-Systeme, die nicht mehr (oder nur mehr mühsam) frei programmierbar sind. Sie können unsere Vorstellungen vom Aufbau von Algorithmen, über den intendierten Ablauf im Rechner, oder sogar die Möglichkeiten universeller Programmierung einschränken. In Programmiersprachen sind Strukturmodelle eingebaut, bei der Objektorientierung beispielsweise *hierarchische Objektklassen*, dieses Modell wird dann auch in der Analyse verwendet. So werden die Analyse und die Repräsentation der Welt beeinflusst durch die Programmiersprache, die man wählt. Weglassen (abstrahieren) und Sichtbarmachen stehen in direkter Relation mit der Programmierumgebung (-sprache). Sie alle lenken die (menschliche) Vorstellung vom Ablauf eines Algorithmus in unterschiedlicher Weise, sind für das eine oder andere mehr oder weniger geeignet, und „erziehen“ das lösungsorientierte Denken der Programmierenden. Mehr noch schränken die Programmierumgebungen und viele Programmierhilfen die Möglichkeiten der Benutzenden ein, indem sie Hilfestellungen geben, die allerdings nicht mehr alle Möglichkeiten offenlassen. So wirken etwa die *Application Programming Interfaces* (APIs), die von Entwicklern als Aufbaublöcke für Anwendungssoftware, Web-basierte Systeme, Betriebssysteme, Datenbanksysteme, Plattformen für soziale Medien oder Software-Bibliotheken benutzt werden, als *Black Boxes*, weil man nur ihre Schnittstellen sehen kann.

Sukzessive wurde die Software-Entwicklung, aber mit der Interaktivität auch die Anwendung, offener gestaltet. Dabei wurde die Kontrollierbarkeit entsprechend reduziert.

Software, die statistische Methoden und stochastische Optimierungsverfahren verwendet, ist teilweise nur mehr empirisch zu beurteilen. Dies trifft auch für evolutionäre Ansätze und genetische Algorithmen zu, die gleichermaßen die Evolution natürlicher Lebewesen als Metapher verwenden.

3.3 Verteiltes Rechnen auf Netzen

Für die Kooperation und das *Resource Sharing* in Netzwerken von Computern im Grid- und Cloud-Computing, Fog- und Dew-Computing werden verteilte Algorithmen benötigt. Mit ihren spezifischen Paradigmen stellen sie für Verifikation und Sicherheit oft unlösbare Aufgaben dar, und sie bieten der kommerziellen Datengewinnung ein nahezu unerschöpfliches Feld problematischer Analysen.

Eine spezielle Form des Distributed Computing stellen *Künstliche Neuronale Netze* (KNNs) dar. Sie orientieren sich auf grob vereinfachte Weise an den Organisations- und Verarbeitungsprinzipien lebendiger neuronaler Systeme. Mit ihnen kann man

u. a. Bilder klassifizieren, Muster erkennen oder motorische Aufgaben lösen, und sie werden zum Machine Learning verwendet. Sie können sehr unterschiedliche Netzwerktopologien haben. Sie bestehen aus statischen Elementen, den Knoten, und Kanten zwischen ihnen, die je mit Eingabewerten initialisiert werden, sowie Regeln über die Verarbeitung all dieser Werte im Netz, die die Informationsverarbeitung in den KNNs beschreiben. Diese stellen den Netzwerkalgorithmus dar. Oft wird die Dynamik des Netzes in eine Lernphase und eine Einsatzphase unterschieden, wobei das Netz in der Lernphase Beispiele präsentiert bekommt, die es nachher zu erkennen oder als Aufgaben zu lösen gilt. Dabei ist der Trainings-Datensatz in der Lernphase entscheidend. So zeigen KNNs, die auf die Erkennung von Hunden oder auch von Türmen und Autos trainiert sind, augenfällig, dass sie am Ende alle Bilder in Hunde oder in Türme und Autos transformieren.⁴

Natürlich ist auch die Auswahl der Netzwerktopologie und des Propagierungsalgorithmus für ein gegebenes Problem ebenso entscheidend für das Ergebnis⁵ wie die (nicht zum Algorithmus selbst gehörenden) Anfangsgewichte der Netzvariablen vor dem Training. Von menschlicher Bewertung ebenso abhängig ist der Zeitpunkt des Auslesens der Ergebnisse, denn das Netz pulsiert immer weiter, es sei denn, es erreicht einen stabilen Zustand.

3.4 Big Data, Data Mining und Machine Learning

Datafication soll alles mittels IT Vorfindliche, Welt und Leben, in Daten transformieren. Big Data aggregiert, ordnet und verdaut in von Umfang, Heterogenität und Geschwindigkeit nie zuvor erreichten Dimensionen die von Digitalen Medien, wie Smartphones, Internet, dem Internet der Dinge und sozialen Medien gelieferten ungeheuren Datenmassen zu sozial, ökonomisch und kulturell relevantem Wissen. *Data Mining* und Datenanalyse benutzen u. a. statistische Modelle, probabilistische Methoden, Lernverfahren und Entscheidungsmodelle, um aus Datenmengen unterschiedlichster Art Korrelationen und Schlüsse zu ziehen. Die Auswahl der Daten und der Kriterien, nach denen de- und induziert wird, sind immer diskutierbar. Dieser in multiplen Bereichen angewandte „digitale empirische Turn“ erhebt den Anspruch von Rationalität und Faktizität, obgleich nicht nur die Daten oft defizient, einseitig vorfindlich oder erhoben und inkompatibel sind, sondern auch methodisch Unsicherheit einzieht. findet. Der empirische Turn nimmt jedoch normativen Charakter an, die Behauptung von Objektivität und Zurechenbarkeit durch die Automatisierung lässt sich konkret nur schwer widerlegen. Doch treffen die folgenden Profilanalysen Menschen u. U. gravierend, von ihrer Kreditwürdigkeit über die Jobchancen. Viel schlimmer noch als Opfer tödlicher Drohnenangriffe.

Machine Learning versucht, aus vorhandenen Beispielen ein allgemeines Gesetz (welches man aber nicht hat) zu erhalten, welches die Beispiele erklärt. Dazu gibt es sehr unterschiedliche Methoden, die sich im Wesentlichen an der Struktur der Domäne orientieren, aus der die Beispiele kommen. Diese Methoden und Algorithmen werden durch die Inhalte nicht berührt. Man kann sie nach Belieben einsetzen, wie sonstige mathematische Methoden auch.

Beim Machine Learning bleiben die Lernalgorithmen zwar statisch, doch die entstehende Software erweitert und spezifiziert sich mit den Trainingsdaten. Die Selektivität solcher Daten wird insbesondere in der *Künstlichen Intelligenz* problematisch, wenn es um die Modellierung menschlicher Eigenschaften und Fähigkeiten durch Training von Daten einer eingeschränkten Bevölkerungsgruppe geht.

Folgen der Verbreitung all dieser Methoden sind statistische *Bias* (Ungleichgewichte) und Diskriminierungen.⁶ So zeigen sich Rassen- und Geschlechter-Diskriminierungen beispielsweise beim Anzeigenangebot.⁷

3.5 Suchmaschinen

Suchmaschinen suchen nach Schlüsselwörtern durch die Webseiten im Internet und sortieren die Ergebnisse nach unterschiedlichen Rangfolgen. Die Organisation der Rangfolgen ist dabei zufällig und priorisiert somit Einträge unterschiedlich. Eine Suchmaschine erstellt ein Netz von Verlinkungen, das die *Beliebtheit* einer Seite beurteilt. Google zeichnet dafür im *PageRank* die gewichteten Verweise auf eine Webseite aus, die Linkpopularität, und erstellt eine Ergebnisreihenfolge. Bei einer Suchabfrage werden die Links je nach Gewicht sortiert. Ein anderes Kriterium ist *Hubs und Authorities* von Kleinberg, wo jede Seite zwei Bewertungen erhält. Beim *TrustRank* hingegen wird eine kleine Anzahl von vertrauenswürdigen Seiten manuell ausgewählt. Diese erwerben dann Vertrauen auf verlinkte Webseiten weiter.

Bei Google werden Suchergebnisse anhand der gespeicherten individuellen Suchhistorie, dem lokalen Aufenthaltsort und weiteren Faktoren personalisiert ausgegeben. Weitere Adaptionen werden wegen der schnellen Veränderungen im Netz vorgenommen, um Manipulationen der Suchergebnisse auszuschließen und um Werbung geeignet zu platzieren. Seit einiger Zeit sind auch semantische Suchverfahren im Einsatz, Google setzte 2013 das neue Verfahren *Hummingbird* auf, um die Absicht der Suchanfrage besser zu verstehen. Werden auch Gruppenprofile und Bildanalysen für Antworten auf Suchanfragen verwendet, kommt es zu Ungleichheiten und Benachteiligungen für unterschiedliche Nutzende, etwa für Frauen bei der Jobsuche: wenn vorwiegend Männer nach hoch bezahlten Jobs suchen und dies bildlich unterstützt wird, werden Männer eher lukrative Jobs angeboten bekommen, während eine Frau, die das Gleiche sucht, ihrer Gruppe und den verzerrenden Bildangeboten entsprechend weniger gute Angebote erhält.⁸ Hier wirkt die Automatisierung als Verstärkung bestehender Ungleichheiten.

3.6 Soziale Netzwerke

Soziale Netzwerke sind Online-Dienste, die auf Plattformen eingeehtete Gemeinschaften mit von ihnen selbst erzeugten Inhalten beherbergen, und die ihren Gewinn i. d. R.⁹ ebenso wie Suchmaschinen aus der Werbung ziehen. Nutzende können hier mit diversen Einstellungen und Kontaktlisten in von ihnen definierten Gruppen kommunizieren, Daten austauschen und ihre eigenen Profile erstellen. Natürlich sind weder die verwendeten Algorithmen noch die dafür erstellte Software für die in diesen Netzen kursierenden Inhalte verantwortlich. Aber die

Software liefert Möglichkeiten, Inhalte zu verstärken, Gruppenprofile zu erstellen und Nutzer zu beeinflussen. So entstehen Echokammern von Nutzenden, die sich wechselseitig bestätigen. *Bots* (autonome mobile Softwareprozesse), wie z. B. *Webcrawler*, arbeiten automatisch Aufgaben ab. *Social Bots* operieren in sozialen Medien, geben dort aus bisherigen Äußerungen generierte Antworten oder setzen mit realistisch wirkenden Fake-Accounts und Profilen vorbereitete Informationen ab. Sie werden nicht nur zur Werbung, sondern auch zum *Nudging* und zu Propaganda und Wahlbeeinflussung eingesetzt und unterhöhlen so die Demokratie. Solches ist sowohl beim *Brexit* wie auch im letzten Wahlkampf in den USA bekannt geworden.¹⁰ Chatbots werden in Messengerdiensten eingesetzt, um die Nutzungsfrequenz zu erhöhen und Nutzende in einem sozialen Netz festzuhalten. Leider lassen sich reale aber kaum von *Fake News* unterscheiden. Mit Bots sind haarsträubende Bias bekannt geworden, etwa Googles Bilderkennungs-Anwendung, die schwarze Menschen als Gorillas identifizierte.¹¹ Dies, weil ihr als Trainingsdaten vorwiegend weiße Menschen angeboten worden waren. Die Wahl eines unvollständigen Datensatzes ist bei solchen Effekten für die Diskriminierung verantwortlich. Microsofts Chatbot *Tay*, der nach 24 Stunden Aufnahme von Konversationsdaten aus Twitter eine unglaublich rassistische, sexistische, homophobe Persönlichkeit annahm, wiederholte und verstärkte leider entsprechend gepostete Tweets.¹²

4. Resümee

Die Frage, ob Algorithmen verantwortlich gemacht werden können, wird durch Darstellung ihrer Beschaffenheit obsolet. Hingegen ermöglichen Softwaresysteme und Automatisierung, die Algorithmen verwenden, Diskriminierungen oder Manipulationen etc. Wo ist die Grenze zu ziehen, ab wann können aus Algorithmen bestehende Programme soziale, moralische, diskriminierende Eigenschaften aufnehmen? Dies ist dann der Fall, wenn dafür mehr oder weniger passende oder gute Algorithmen ausgewählt werden, wenn, wie für Software üblich, Gestaltungsaspekte mit *Likeability* und *Seducability* dazu kommen, Anforderungsermittlung getätigt wird, Datenstrukturen, Schnittstellen, der Zugang und Interfaces definiert werden, die die Möglichkeiten zur Interaktion bieten, sei es für Mensch oder Sensoren, und auch wenn Sicherheitsaspekten Genüge geleistet wird; und weiter wenn kontingente Daten(-mengen) in die Software integriert werden. Beim Entwerfen von Software, für die Entwicklung und Benutzung von Programmiersprachen, für verteiltes Rechnen auf Netzen zeichnen vor allem Auftraggebende und Entwickelnde verantwortlich, für Big Data, Data Mining und Machine Learning, Suchmaschinen und Soziale Netzwerke überdies, und sehr viel mehr, die großen Internet-Firmen, die Nutzenden und Textgebenden. Immer sind es letztlich Menschen, die entscheiden, welche Algorithmen wo und in welchen Zusammenhängen wie kombiniert und verwendet werden. Es sind Menschen, die etwa unvollständige Datensätze in Programme füttern oder dies an Automaten delegieren; die trotz aller bekannt gewordenen politischen Wirkungen von Hassbotschaften, Echokammern und Fake News *Bots* damit füttern. Für die Wirkung sind weder die Algorithmen noch die benutzten Buchstaben verantwortlich.

Software wirkt immer als Verstärker, von Besonderheiten in Organisationen, hinsichtlich eventueller Bias, Einbindungen oder Auslassungen. Sie vergrößert alle Effekte, die aus den sozialen Zusammenhängen gezogen werden, sie bestätigt und zementiert nicht nur Verhältnisse, sondern reifiziert und vertieft gesellschaftliche Ungleichgewichte. Dies noch unvergleichlich mehr, wenn die Datenprodukte in iterierte und rekursive Prozeduren gefüttert werden, wie dies beispielsweise für Big Data der Fall ist. Zudem werden sie durch die Virtualisierung oft unsichtbar und verfestigen sich. Dringender denn je ist deshalb zu fragen, wieweit es zuträglich ist, unsere Arbeits- und Lebenswelt weiter zu automatisieren. Es ist ein Fehler zu glauben, mit KI, Big Data, Handlungsplänen und Optimierungsverfahren die Welt steuern zu können. Der Menschen zugängliche Umgang mit dem Unerwarteten wird mit der Automatisierung darauf beschränkt, das Mögliche als aufzählbar zu betrachten. Die Zukunft wird dabei auf die De- oder Induktion aus Vergangenen reduziert.

Wir sind es, die wir Daten an soziale Netzwerke liefern und dort Profile erstellen, bestimmte Ansichten durch *liken*, *sharen* etc. verstärken, während wir gleichzeitig Internetdienste umsonst nutzen (wollen); wir, die wir als Informatik-Profession eine ahistorische und angeblich neutrale, vermeintlich objektive Haltung einnehmen. Die Verantwortung bleibt jedoch immer bei uns, auch wenn wir gern die Schuld an andere oder etwas anderes abschieben. Erweitert man aber den Algorithmenbegriff auf jegliche Automatisierung, so weist man individuelle und kollektive Verantwortung als Objektives und Unveränderliches ab.

Ich danke Cecile Crutzen für Anregungen und Verbesserungen an diesem Text, und Jörg Pflüger und Wolfgang Coy für klärende Gespräche darüber.

Anmerkungen

- 1 Crutzen CKM (2013) *Masks between the Visible and the Invisible*. In: Ernst W, Horwath I (eds) *Gender in Science and Technology. Interdisciplinary Approaches*. Transcript, Bielefeld, pp 79–110
- 2 Im Bereich der Gender Studies wurde später der „ego-approach“ von Madeline Akrich in *The De-Description of Technical Objects* (1992) Bijker W, Law J (eds) *Shaping Technology / Building Society*. MIT Press, Cambridge, pp 205–224 als sogenannte „I-methodology“ neu entdeckt und wird heute so bezeichnet. Dort betrifft sie allerdings vorzugsweise die Benutzung, die sich Entwickelnde gemäß ihren eigenen Bedürfnissen vorstellen.
- 3 Suchman, L (1987) *Plans and Situated Action. The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge
- 4 <http://www.zeit.de/digital/internet/2015-07/neuronale-netzwerke-google-inception>
- 5 Ein anderes KNN hätte u. U. die Differenzierung nicht nach der Hautfarbe treffen können, sondern nach der Kopfform, oder aber auch gar keine für Menschen erkennbare Unterscheidungen.
- 6 Sweeney, L (2013) *Discrimination in Online Ad Delivery*. CACM 56(5): 44–54
- 7 Custers B, et al. (2013) *Discrimination and Privacy in the Information Society*. Springer, Berlin
- 8 <http://www.washington.edu/news/2015/04/09/whos-a-ceo-google-image-results-can-shift-gender-biases/>
- 9 Freie dezentrale verteilte soziale Netzwerke wie Diaspora oder firendica finanzieren sich durch Spenden, ebenso freie Suchmaschinen, wie DuckDuckGo, Wegtam, DeuSu, MetaGer, Unbubble oder ixquick.eu, die jeweils auch Datenschutzaspekte berücksichtigen
- 10 <http://www.spektrum.de/kolumne/die-macht-der-algorithmen/1429137>, <https://www.theguardian.com/politics/2017/feb/26/robert-mercer-breitbart-war-on-media-steve-bannon-donald-trump-nigel Farage>
- 11 <http://www.theverge.com/2015/7/1/8880363/google-apologizes-photos-app-tags-two-black-people-gorillas>
- 12 <https://qz.com/653084/microsofts-disastrous-tay-experiment-shows-the-hidden-dangers-of-ai/>



Rainer Rehak und Jens Wernicke

Die Manipulation von Denken und Handeln ist zur treibenden Kraft der IT-Entwicklung geworden

Jens Wernicke interviewt Rainer Rehak,
erstveröffentlicht auf dem Nachrichtenportal nachdenkseiten.de

In unserer technologisierten Gesellschaft ist die Manipulation von Denken und Handeln zur treibenden Kraft der IT-Entwicklung geworden. Das ist kein Zufall, sondern die Folge der Entwicklung der Technik, die uns das Leben immer mehr steuert. Die Informatikerinnen und Informatiker für Frieden und gesellschaftliche Verantwortung nicht hinnehmen wollen. „Versteckte Informationstechnik ist nicht diskutierbar“, kritisiert Rainer Rehak, einer der Organisatoren der diesbezüglichen Jahreskonferenz¹, im Interview mit Jens Wernicke.

erschienen in der Fiff-Kommunikation,
herausgegeben von Fiff e.V. - ISSN 0938-3476
www.fiff.de

Individuelle Selbst- und demokratische Verantwortung der Kunden, der Staat sich seine Verantwortung nicht überlassen darf. Die Entwicklung, die uns das Leben immer mehr steuert, ist die Folge der Entwicklung der Technik, die uns das Leben immer mehr steuert.

Jens Wernicke (JW): Herr Rehak, Sie sind im Vorstand des Fiff, dem Forum InformatikerInnen für Frieden und gesellschaftliche Verantwortung, und dieses Jahr Mit-Organisator der 32. Fiff-Konferenz, die vom 25. bis 27.11. in Berlin stattfindet und die Gefahren sogenannter „unsichtbarer Systeme“ behandeln wird. Als Informatiker und Informatikerinnen, die sich für Frieden engagieren, postulieren Sie im Fiff: „Die Manipulation von Denken

und Handeln ist zur treibenden Kraft der IT-Entwicklung geworden.“ Was bitte sind „unsichtbare Systeme“? Und wer bemüht sich um Kontrolle unseres Denkens und Handelns?

Rainer Rehak (RR): Das lässt sich am besten anhand eines technischen Beispiels erklären: Früher bestand ein Auto aus Motor, Fahrgestell, Getriebe, Reifen usw. Die komplexesten Dinge waren viel-