

Simulation in Produktion und Logistik 2023
Bergmann, Feldkamp, Souren und Straßburger (Hrsg.)
Universitätsverlag Ilmenau, Ilmenau 2023
DOI (Tagungsband): 10.22032/dbt.57476

Maschinelles Lernen von Maschinenfolgen für den simulationsbasierten Test von Verfahren der Produktionsplanung und -steuerung

Machine Learning of Machine Sequences for the Simulation-Based Test of Methods for Production Planning and Control

Marvin Matthes, Oliver Guhr, Torsten Munkelt, Martin Krockert
Hochschule für Technik und Wirtschaft Dresden, Dresden (Germany),
{Marvin.Matthes, Oliver.Guhr, Torsten.Munkelt, Martin.Krockert}@htw-dresden.de
Sven Völker, Technische Hochschule Ulm,
Ulm (Germany), Sven.Voelker@thu.de

Abstract: Evaluating production planning and control (PPC) methods by means of simulation requires test data. Models of machine sequences of real productions can be created using machine learning. The paper describes how transformers and Bayesian networks are learned from real-life data of a manufacturing company. Both model types are well known for learning sequences with multiple conditional dependencies. These models are used to generate machine sequences which should be similar to the original machine sequences in terms of their statistical properties. From the generated machine sequences, work plans and production orders can be derived and used to test PPC methods by means of stochastic discrete-event simulation.

1 Einleitung

Viele Verfahren zur Produktionsplanung und -steuerung (PPS) sind Heuristiken. Ihre Effektivität und Effizienz können in der Regel nicht analytisch, sondern nur empirisch ermittelt werden, wozu gemeinhin stochastische ereignisdiskrete Simulation genutzt wird. Um Aussagen über die Eignung von PPS-Verfahren zu treffen, die sich nicht nur auf eine spezielle Produktion beziehen, sondern eine gewisse Allgemeingültigkeit besitzen, werden Testdaten benötigt, die Produktionen mit unterschiedlichen Eigenschaften realitätsnah abbilden. Testdaten aus öffentlich zugänglichen Problembibliotheken (Taillard 1993) mangelt es meist an Realitätsnähe. Derartige Daten in realen Produktionssystemen zu erheben, ist aus praktischen Gründen (Aufwand, Zugänglichkeit, Vertraulichkeit) schwierig oder unmöglich. Eine Alternative besteht darin, Testdaten zu generieren. Dieser Ansatz wurde bereits in der Vergangenheit verfolgt (Fernandes et al. 2020, Adolph et al. 2015). Ein wesentlicher

Bestandteil von PPS-Testdaten sind Maschinenfolgen, welche Arbeitsplänen und Produktionsaufträgen zugrunde liegen. In Döring et al. (1999) und Völker et al. (2001) sind Maschinenfolgen aufgrund plausibler Annahmen generiert worden. Die Analyse realer Produktionsdaten zeigt aber, dass diese Annahmen nicht immer zutreffen. Krockert et al. (2021) sind deshalb dazu übergegangen, Maschinenfolgen realer Produktionen zu analysieren und aufgrund der Analyseergebnisse Maschinenfolgen zu synthetisieren. Eine besondere Herausforderung liegt dabei in der Synthese wiederkehrender Muster aufeinanderfolgender Maschinen, sogenannter Submaschinenfolgen.

2 Maschinenübergangsmatrizen zur Generierung von Maschinenfolgen

Maschinenübergangsmatrizen bzw. -graphen sind Standardmodelle zur abstrakten Beschreibung von Maschinenfolgen (Schuh 2012). Sie werden im vorliegenden Beitrag als Referenz zum Einsatz von Bayes'schen Netzen und Transformern genutzt. Eine Maschinenübergangsmatrix ist eine quadratische Matrix, deren Elemente π_{ij} die Wahrscheinlichkeit des Übergangs von Maschine i auf Maschine j angeben. Sie können als Maschinenübergangsgraphen visualisiert werden. Beispielhaft stellt Abbildung 1 eine Maschinenübergangsmatrix in Form eines zweidimensionalen Säulendiagrammes und einen Maschinenübergangsgraphen dar. Eine bedeutende Limitation dieser Modellform besteht darin, dass wiederkehrende Submaschinenfolgen nicht adäquat abgebildet werden, weil die Wahrscheinlichkeit für den Übergang von Maschine i auf Maschine j lediglich von Maschine i abhängt. Daher bleiben potenzielle Muster in den vorausgehenden und nachfolgenden Maschinenfolgen unberücksichtigt.

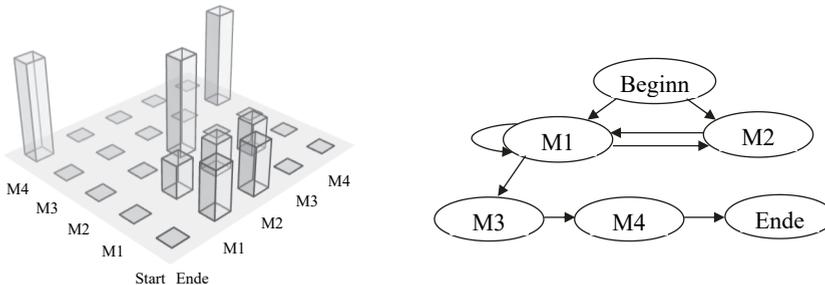


Abbildung 1: Beispiel für eine Maschinenübergangsmatrix und einen Maschinenübergangsgraphen

3 Kennzahlen zur statistischen Beschreibung von Maschinenfolgen

Die statistischen und strukturellen Eigenschaften von Maschinenfolgen können durch Kennzahlen beschrieben werden. Diese Kennzahlen können sowohl aus den Maschinenfolgen der Realdaten als auch aus den mittels maschineller Lernverfahren generierten Maschinenfolgen gewonnen werden. Aus dem Vergleich der Kennzahlen lassen sich quantitative Aussagen über die Güte der Verfahren ableiten.

Der von Döring et al. (1999) vorgeschlagene *Organisationsgrad (OG)* (siehe Formel 1) ist ein Maß für den Organisationstyp einer Produktion. Der OG erlaubt, stufenlos zwischen Organisationstypen einer Produktion unterscheiden, von der Werkstatt- bis zur Fließfertigung. Er wird gemäß folgender Formel berechnet, wobei M die Menge der Maschinen repräsentiert und π_{ij} die Wahrscheinlichkeit des Übergangs von Maschine i auf Maschine j .

$$OG = \frac{1}{|M| - 1} \sum_{i=1}^{|M|} \sum_{j=1}^{|M|} \left(\pi_{ij} - \frac{1}{|M|} \right)^2 \quad (1)$$

Abbildung 2 veranschaulicht den OG anhand dreier Maschinenübergangsmatrizen mit ihren OGen und Organisationstypen.

	M1	M2	M3	Senke
Quelle	1,0	0	0	0
M1	0	1,0	0	0
M2	0	0	1,0	0
M3	0	0	0	1,0

Fließfertigung (OG = 1)

	M1	M2	M3	Senke
Quelle	0,76	0,18	0,05	0,01
M1	0,08	0,70	0,17	0,04
M2	0,01	0,09	0,73	0,17
M3	0,00	0,01	0,11	0,88

Gemischte Fertigung (OG = 0,5)

	M1	M2	M3	Senke
Quelle	0,25	0,25	0,25	0,25
M1	0,25	0,25	0,25	0,25
M2	0,25	0,25	0,25	0,25
M3	0,25	0,25	0,25	0,25

Werkstattfertigung (OG = 0)

Abbildung 2: Maschinenübergangsmatrizen und Organisationsgrade für unterschiedliche Organisationstypen

Die *Anzahl unterschiedlicher Maschinen (Maschinenordnung MO)* (siehe Formel 2) in einer Maschinenfolge weist auf verschiedene Maschinen und eine komplexere Produktion und komplexere Materialien hin.

$$MO = |M| \quad (2)$$

M ... Menge unterschiedlicher Maschinen in allen Maschinenfolgen

Mit der *mittleren Länge der Maschinenfolge (Mittlere Länge ML)* (siehe Formel 3) wird die durchschnittliche Anzahl von Maschinen in einer Maschinenfolge definiert. Sie kann als Indikator für die Komplexität und den Umfang der betrachteten Maschinenfolgen verwendet werden.

$$ML = \frac{\sum_{i=1}^{|S|} l_{s_i}}{|S|} \quad (3)$$

S ... Menge aller Maschinenfolgen

s_i ... spezifische Maschinenfolgen, wobei $1 \leq i \leq |S|$

l ... Länge einer Maschinenfolge

Die *relative Häufigkeit des Vorkommens von Maschinen in Maschinenfolgen (Maschinenvorkommen RMV_j)* (siehe Formel 4) gibt an, wie oft eine bestimmte Maschine in allen Maschinenfolgen vorkommt.

$$RMV_j = \frac{V_j}{\sum_{i=1}^{MO} V_i} \quad (4)$$

V_m ... Gesamtzahl der Vorkommen einer Maschine m in alle Maschinenfolgen

Die *relative Häufigkeit des Vorkommens von Submaschinenfolgen* (Subfolgen-vorkommen RSV_j) (siehe Formel 5) beschreibt, wie oft bestimmte Submaschinenfolgen in den untersuchten Maschinenfolgen vorkommen. In diesem Beitrag werden Submaschinenfolgen mit einer Länge zwischen drei und sechs Maschinen ermittelt. Die Identifikation häufiger Submaschinenfolgen kann dabei helfen, Muster und in Maschinenfolgen aufzudecken. Die Häufigkeiten von Subfolgen lassen sich zudem als Histogramme darstellen und vergleichen.

$$RSV_j = \frac{\sum_{i=1}^{|S|} v_{ij}}{\sum_{i=1}^{|S|} \sum_{j=1}^{|T|} v_{ij}} \quad (5)$$

T ... Menge aller möglichen Submaschinenfolgen mit einer Länge l , wobei $3 \leq l \leq 6$

t_j ... spezifische Submaschinenfolgen, wobei $1 \leq j \leq |T|$

v_{ij} ... Anzahl der Vorkommen der Submaschinenfolgen t_j in der Maschinenfolgen s_i

4 Maschinelles Lernen von Modellen für Maschinenfolgen aus Produktionsdaten

Zunächst werden Ansätze vorgestellt, um aus vorliegenden Maschinenfolgen realer Produktionen Modelle für Maschinenfolgen maschinell zu erlernen und die Modelle anschließend zum Generieren neuer Maschinenfolgen zu verwenden. Bei den Modelltypen handelt es sich um Transformer (Vaswani et al. 2017) und Bayes'sche Netze (Neapolitan 2007, Neuberg 2003).

4.1 Transformer

Das erste für die Generierung von Maschinenfolgen eingesetzte Modell ist ein Transformer, der auf einer autoregressiven Architektur basiert. Die Transformer-Architektur wurde von Vaswani et al. (2017) vorgeschlagen, und sie hat sich für eine Vielzahl von Anwendungen in der Verarbeitung natürlicher Sprache und darüber hinaus etabliert. Der autoregressive Ansatz wurde durch die Arbeiten von Radford et al. (2018), Radford et al. (2019) und Brown et al. (2020) eingeführt und weiterentwickelt. Dieser Abschnitt beschreibt, wie der Transformer für das spezifische Problem adaptiert worden ist und welche Verbesserungen vorgenommen worden sind.

Ein autoregressiver Transformer ist ein neuronales Netz, das sequenzielle Daten verarbeitet, indem es eine Wahrscheinlichkeitsverteilung über mögliche nächste Elemente in der Sequenz erlernt, wobei die vorherigen Elemente als Kontext verwendet werden (Abbildung 3). Transformer nutzen die Self-Attention, um Beziehungen zwischen verschiedenen Positionen innerhalb der Eingabesequenz zu erfassen und zu gewichten. So können sie das nächste Element in der Sequenz auf Basis der vorhergegangenen Elemente vorhersagen, siehe Formel (6).

$$P(x_t | x_1, x_2, \dots, x_{t-1}) = \text{Transformer}(x_1, x_2, \dots, x_{t-1}; \theta) \quad (6)$$

Dabei gibt x_t das t -te Element der Sequenz an, und $[x_1, x_2, \dots, x_{t-1}]$ sind die vorangegangenen Elemente in der Sequenz. Der autoregressive Transformer

verwendet die vorangegangenen Elemente als Kontext für die Prognose. Um die Wahrscheinlichkeitsverteilung für das nächste Element x_t zu erzeugen, wird der Ausgabevektor des Transformers von einem Multilayer Perceptron (MLP) klassifiziert. Mit θ wird die Menge der Gewichte des Transformers bezeichnet. Während des Trainingsprozesses werden die Gewichte angepasst, um die Leistung des Transformers bei der Vorhersage von den Daten zu optimieren. Die angepassten Gewichte ermöglichen es dem Transformer, Muster und Zusammenhänge in den Eingabesequenzen zu erkennen und entsprechende Vorhersagen für nächste Elemente der Sequenz zu treffen.

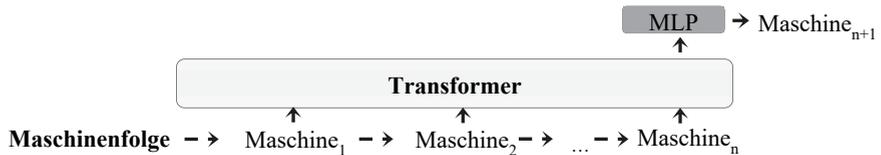


Abbildung 3: Aufbau des Transformers für die Maschinenfolge

Damit die Daten in der Sequenz $[x_1, \dots, x_t]$ vom Transformer verarbeitet werden können, müssen sie in ein geeignetes Format umgewandelt werden. Diese sogenannte Tokenisierung zerlegt die Eingabedaten in kleinere Einheiten, sogenannte Token. Im Kontext von Sprachmodellen bezieht sich die Tokenisierung auf die Zerlegung von Texten in diskrete Elemente wie Wörter, Wortgruppen oder Zeichen, abhängig von der gewählten Granularität. Die Token dienen dem Transformer als Eingabe, wodurch er in der Lage ist, die Struktur und Zusammenhänge innerhalb der Eingabesequenzen zu lernen.

Da sich die Struktur natürlicher Sprache von der von Maschinenfolgen unterscheidet, wird ein Tokenizer mit den gegebenen Produktionsdaten trainiert. Dabei kommt die „Byte-Pair Encoding (BPE)“-Tokenisierung nach Sennrich et al. (2016) zum Einsatz. Die BPE-Tokenisierung ist ein datengetriebener Ansatz, bei dem häufig vorkommende Zeichen- oder Byte-Paare in einer Eingabesequenz zu einem Token verschmolzen werden. Das Ergebnis ist ein Vokabular, das sowohl häufige als auch seltene Muster in den Daten effizient abbildet, wodurch eine flexible und anpassungsfähige Tokenisierung ermöglicht wird, die sich gut für die Abbildung von Maschinenfolgen eignet.

Die Struktur des Modells umfasst eine Token-Vektorlänge von 384, eine maximale Sequenzlänge von 150 Token, 8 Layer und 8 Attention-Heads. Um die Leistung des Modells weiter zu verbessern und es auf spezielle Anforderungen abzustimmen, sind Verbesserungen im Vergleich zum Standardmodell der Transformer-Architektur erfolgt: Erstens wird die "Swish"-Aktivierungsfunktion von Ramachandran et al. (2017) genutzt. Zweitens wird die "rotary positional embeddings" nach Sun et al. (2022) integriert, um die Skalierbarkeit und Robustheit des Modells bei der Verarbeitung langer Sequenzen zu verbessern. Drittens wird die "One Write-Head Attention"-Technik von Shazeer (2019) eingeführt, um den Berechnungsaufwand der Multi-Head Attention zu reduzieren und die Effizienz des Modells zu steigern. Schließlich wird das "Attention Head Scaling" nach Shleifer et al. (2021) angewandt, um die Leistungsfähigkeit des Modells weiter zu verbessern.

4.2 Bayes'sche Netze

Ein Bayes'sches Netz ist ein gerichteter azyklischer Graph, dessen Knoten Zufallsvariablen und dessen Kanten direkte (kausale) Abhängigkeiten zwischen den Zufallsvariablen repräsentieren. Jedem Knoten bzw. jeder Zufallsvariable sind für jede Kombination von Zuständen der Elternknoten Wahrscheinlichkeitsverteilungen zugeordnet, die aussagen, wie wahrscheinlich die Zustände der Zufallsvariablen sind, bedingt durch die Zustände der Elternknoten. Bayes'sche Netze können aus Daten gelernt und u. a. zur Prognose, Diagnose und Sensitivitätsanalyse unter Unsicherheit verwendet werden. Zudem ist es möglich, aus Bayes'schen Netzen eine Stichprobe zu ziehen, welche den Daten statistisch entspricht, aus denen die Bayes'schen Netze erstellt worden sind.

In diesem Beitrag werden Bayes'sche Netze aus realen Maschinenfolgen einer Produktion gelernt, und im Anschluss werden Stichproben neuer Maschinenfolgen aus den Bayes'schen Netzen gezogen und anhand der in Abschnitt 3 vorgestellten Kennzahlen mit den originalen Maschinenfolgen verglichen. Da Bayes'sche Netze mehrfach bedingte Wahrscheinlichkeiten kodieren können, ist zu erwarten, dass sie häufig vorkommende Submaschinenfolgen besser erzeugen als Maschinenübergangsmatrizen.

Für die Modellierung eines Bayes'schen Netzes für Maschinenfolgen bieten sich drei Ansätze an:

- *Maschinen als Knoten mit binären Zuständen*: Die Knoten des Netzes repräsentieren Maschinen, die entweder in der Maschinenfolge vorkommen (Zustand 1) oder nicht (Zustand 0). Wenn eine Stichprobe aus einem solchen Netz gezogen wird, sagt sie allerdings nur aus, dass bestimmte Maschinen vorkommen, aber nicht, wie oft und in welcher Reihenfolge.
- *Maschinen als Knoten mit Zuständen zur Anzahl des Auftretens*: Die Knoten des Netzes repräsentieren wiederum Maschinen, aber die Zustände der Knoten sagen aus, wie oft die Maschine in der Maschinenfolge vorkommt. Wenn eine Stichprobe aus einem solchen Netz gezogen wird, sagt sie nun zwar auch aus, wie oft bestimmte Maschinen in der Maschinenfolge vorkommen, aber immer noch nicht in welcher Reihenfolge.
- *Positionen der Arbeitsgänge als Knoten*: Die Knoten des Netzes repräsentieren Positionen der Arbeitsgänge, und jede Arbeitsgangposition kann als Zustände der Maschinen annehmen, die an ihr vorkommen. Mit diesem Ansatz kann zusätzlich eine Reihenfolge der Maschinen abgebildet werden.

Offenkundig ist der dritte Modellierungsansatz „Positionen der Arbeitsgänge als Knoten“ am besten geeignet. Das Lernen erfolgt mit dem Score-basierten Hill-Climbing-Ansatz (Tyugu 2008) und dem BIC-Score als Gütemaß (Koller 2009). Für die anschließende Generierung der Maschinenfolgen aus dem Bayes'schen Netz wird das Forward-Sampling verwendet (Koller 2009). Abbildung 4 zeigt einen Auszug des für die Generierung von Maschinenfolgen gelernten Bayes'schen Netzes. Der dargestellte Ausschnitt repräsentiert die bedingte Abhängigkeit der Position 6 von der Position 5 der Maschinenfolge, wobei die zugrundeliegende Wahrscheinlichkeitsverteilung für den Knoten der Position 6 dargestellt ist. Die erste Zeile der Wahrscheinlichkeitsverteilung besagt, dass, falls an Position 5 der Maschinenfolge die Maschine 10000 steht, mit einer Wahrscheinlichkeit von etwa 6,04% die Maschine 20000 auf Position 6 folgt.

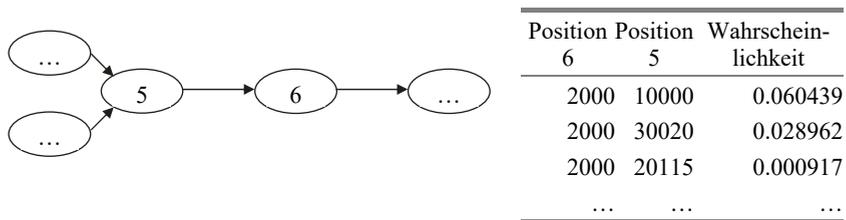


Abbildung 4: Auszug aus dem Bayes'schen Netze für Maschinenfolgen über die Positionen der Arbeitsgänge

5 Bewertung der Modelle zur Generierung von Maschinenfolgen

Das Ziel besteht darin, dass die erlernten Modelle die realen Maschinenfolgen möglichst gut abbilden. Um die Modelle zu bewerten, werden neue Maschinenfolgen generiert und mit den Realdaten anhand statistischer Kennzahlen (siehe Abschnitt 3) verglichen.

Abbildung 5 stellt den Prozess des maschinellen Lernens und den Vergleich der Modelle auf Basis der generierten Maschinenfolgen dar.

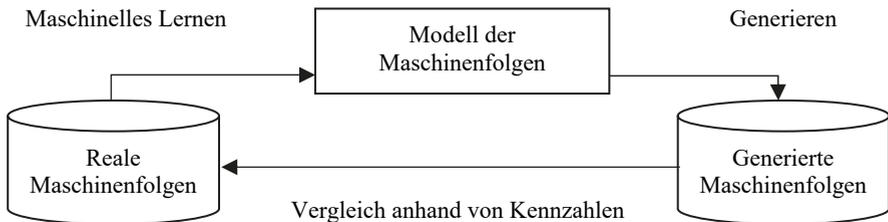


Abbildung 5: Vorgehen zum Generieren und Evaluieren von Maschinenfolgen

Eingabe für das maschinelle Lernen sind zirka 170.000 reale Maschinenfolgen aus Produktionsaufträgen eines metallverarbeitenden Betriebes mit 73 verschiedenen Maschinen. In einer Maschinenfolge können Maschinen mehrfach vorkommen, so dass Zyklen entstehen. Die relative Häufigkeit des Vorkommens einzelner Maschinen in den Maschinenfolgen unterscheidet sich stark: 15 der 73 Maschinen repräsentieren bereits 90 % der Maschinen in den Maschinenfolgen.

Das Ziel maschinellen Lernens besteht darin, die Verteilungen in den realen Maschinenfolgen möglichst gut im Modell abzubilden, d. h., generierte Maschinenfolgen und Realdaten sollten sich hinsichtlich der Kennzahlenausprägungen möglichst wenig unterscheiden. Die beiden Histogrammbasierten Kennzahlen zur relativen Häufigkeit des Vorkommens der Maschinen und der Submaschinenfolgen sind die euklidischen Distanzen zu den Realdaten. Tabelle 1 zeigt die Resultate des Vergleichs

Tabelle 1: Realitätsnähe der generierten Maschinenfolgen mit Hervorhebung der besten Werte je Kennzahl

Kennzahl	Realdaten (Zielwerte)	Trans- former	Bayes'sches Netz	Maschinen- übergangsmatrix
Organisationsgrad (<i>OG</i>)	0,51	0,54	0,28	0,43
Maschinenordnung (<i>MO</i>)	73,00	67,00	73,00	65,00
Mittlere Länge (<i>ML</i>)	5,22	5,29	5,15	7,40
Maschinenvorkommen (<i>RMV</i>)	0,00	3,64	0,65	48,78
Subfolgenvorkommen (<i>RSV</i>)	0,00	0,59	1,97	8,05

Die Auswertung verdeutlicht, dass sich *Bayes'sche Netze* sehr gut zur Abbildung der relativen Häufigkeit des Vorkommens der Maschinen und der Vollständigkeit der Maschinenordnung eignen. Allerdings zeigen sie Schwächen bei der Erkennung und Abbildung von Submaschinenfolgen. Im Gegensatz dazu erkennen *Transformer* Submaschinenfolgen sehr gut. Abbildung 6 zeigt die relativen Häufigkeiten von Submaschinenfolgen in den Realdaten sowie in den mittels der verschiedenen Modelle generierten Daten.

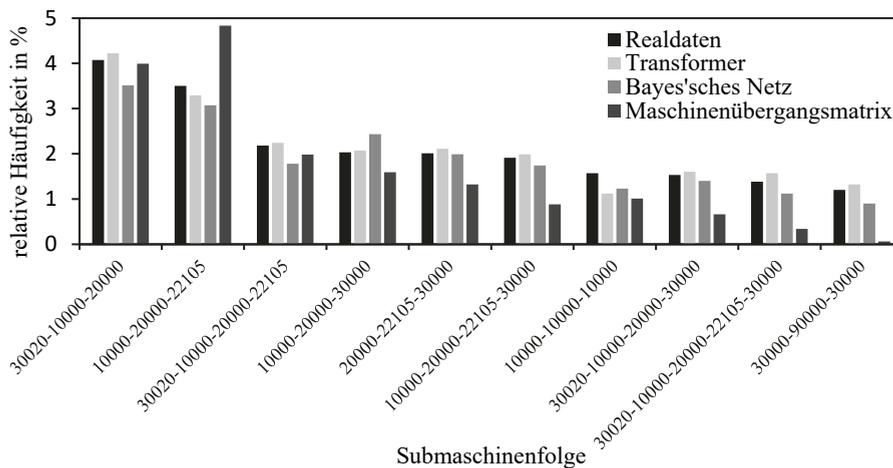


Abbildung 6: Vergleich der Modelle anhand der Histogramme der generierten Submaschinenfolgen

Transformer und Bayes'sche Netze weisen unterschiedliche Stärken und Schwächen auf. Während der Transformer in der Lage ist, Submaschinenfolgen präzise zu erkennen, kann das Bayes'sche Netz zwar Gruppen von Maschinen identifizieren, die gemeinsam auftreten, hat allerdings Schwierigkeiten, die korrekte Reihenfolge der Maschinen zu bestimmen. Die Interpretation der erstellten Modelle ist schwierig. Bei den Transformern ist die Interpretation aufgrund ihrer Komplexität anspruchsvoll, jedoch ermöglicht die Visualisierung von Attention-Mechanismen eine gewisse Einsicht in die Modellentscheidungen. Bayes'sche Netze sind Graphenmodelle und deshalb tendenziell einfacher interpretierbar. Bei der gewählten Modellierungsart „Positionen der Arbeitsgänge als Knoten“ sind die direkten Abhängigkeiten allerdings

nur schwer und die zugrundeliegenden bedingten Wahrscheinlichkeitsverteilungen aufgrund ihrer schieren Größe kaum noch zu interpretieren. Die Laufzeiten und der Speicherbedarf des maschinellen Lernens und Anwendens der Modelle variieren stark. Das Training eines Transformer-Modells über 15 Epochen benötigt beispielsweise 1:24 Minuten. Es ist dabei zu beachten, dass das resultierende Modell rund 13 Millionen Parameter aufweist und eine Größe von 52 MB hat. Der Prozess des Samplings zur Erstellung von 100.000 Maschinenfolgen nimmt auf einer GPU (Nvidia 4090) etwa 36 Minuten in Anspruch. Das Erlernen eines Bayes'sche Netzwerks dauert hingegen lediglich 5 Sekunden. Das Forward-Sampling von 100.000 Maschinenfolgen mit dem Bayes'schen Netz bedarf sogar nur einer Sekunde.

6 Zusammenfassung und Ausblick

Der Beitrag beschreibt die Erstellung von Modellen für die Generierung von Maschinenfolgen mittels maschinellen Lernens auf der Basis von Realdaten. Die generierten Maschinenfolgen können in einem nachgelagerten Schritt für die Validierung von Algorithmen zur Produktionsplanung- und -steuerung z. B. mittels stochastischer ereignisdiskreter Simulation genutzt werden. Es wurden Transformer trainiert und Bayes'sche Netze gelernt, um neue Maschinenfolgen zu generieren. Die Analyse der generierten Maschinenfolgen ergibt, dass Bayes'sche Netzwerke die relative Häufigkeit des Vorkommens von Maschinen in Maschinenfolgen sowie die Maschinenordnung gut abbilden, jedoch Schwierigkeiten bei der Erkennung und Darstellung von Submaschinenfolgen haben. Im Gegensatz dazu sind Transformer effektiv bei der Erkennung von Submaschinenfolgen. Allerdings sind Transformer und – in diesem Fall auch – Bayes'sche Netze schwer zu interpretieren. Zudem verursachen Transformer in Größenordnungen mehr Rechenaufwand als Bayes'sche Netze.

Um die Vorteile beider Ansätze zu kombinieren, könnten zukünftige Arbeiten das sukzessive Anwenden von Bayes'schen Netzen und von Transformern untersuchen: Die Netze bestimmen die Maschinen, die in einer generierten Maschinenfolge enthalten sein sollen, und die Transformer bringen diese Maschinen in eine Reihenfolge. Zudem wäre eine weitere Anwendung der durch maschinelles Lernen erstellten Modelle zu Generierung von Maschinenfolgen denkbar: Sie könnten z. B. die Erstellung von Arbeitsplänen unterstützen. Es ist auch denkbar, Modelle für günstige Abarbeitungsreihenfolgen von Arbeitsgängen auf Maschinen zu lernen, günstige Abarbeitungsfolgen zu generieren und somit einen Beitrag zur Effizienzsteigerung in der Produktion leisten.

Danksagung

Dieser Beitrag wurde gefördert durch das Bundesministerium für Wirtschaft und Klimaschutz aufgrund eines Beschlusses des Deutschen Bundestages. Finanziert von der europäischen Union.

Literatur

Adolphy, S.; Grosser, H.; Kirsch, L.; Stark, R.: Method for Automated Structuring of Product Data and its Applications. In: Procedia CIRP 38 2015, S. 153–158.

- Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P. et al.: Language Models are Few-Shot Learners 2020.
- Döring, T. Völker, S.; Munkelt, T.: Generierung komplexer Testdaten zur statistischen Analyse von Verfahren der Produktionsplanung und -steuerung. In: Bösel, M. (Hrsg.): Amtliche und Nichtamtliche Statistiken - 12. Ilmenauer Wirtschaftsforum, Tagungsband: Technische Universität Ilmenau, Fakultät für Wirtschaftswissenschaften, Fachgebiet Wirtschaftsstatistik und Operations Research 1999, S. 34–46.
- Fernandes, E. C.; Santos, L. I. dos; Camatti, J. A.; Brown, L.; Borsato, M.: Flexible Production Data Generator for Manufacturing Companies. In: *Procedia Manufacturing* 51 2020, S. 1478–1484.
- Krockert, M.; Matthes, M.; Munkelt, T.; Völker, S.: Generierung realitätsnaher Testdaten für die Simulation von Produktionen. In: Franke, J. und Schuderer P. (Hrsg.): *Simulation in Produktion und Logistik 2021*. Göttingen: Cuvillier Verlag, 2021, S. 565–574.
- Koller, D., Friedman, N.: *Probabilistic graphical models: principles and techniques*: MIT press 2009.
- Neapolitan, R. E.: *Learning Bayesian networks*. In: Berkhin, P.; Caruana, R. und Wu X. (Hrsg.): *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD07: The 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Jose California New York (USA) 2007, S. 1.
- Neuberg, L. G.: *Causality: Models Reasoning and Inference by Judea Pearl* Cambridge University Press 2000. In: *Econ. Theory* 19 (2003), S. 675–685.
- Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.: *Improving language understanding by generative pre-training* 2018.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; others: *Language models are unsupervised multitask learners*. In: *OpenAI blog* 1 (2019), S. 9.
- Schuh, G.; Stich, V.: *Produktionsplanung und -steuerung. 4., überarbeitete Auflage* (Hrsg.): Berlin, Heidelberg Springer Vieweg (VDI-Buch) 2012.
- Taillard E., "Benchmarks for basic scheduling problems", *EJOR* 64(2):278-285, 1993. <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>, letzter Zugriff: 10.07.2023.
- Tyugu, E., & Tyugu, E. K.: *Algorithms and architectures of artificial intelligence* (Vol. 159) (Hrsg.): IOS Press 2007.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N. et al.: *Attention Is All You Need*, 2017. <http://arxiv.org/pdf/1706.03762v5>, letzter Zugriff: 27.06.2023.
- Völker, S.; Döring, T.; Munkelt, T.: *The Generation of Large Test Data for the Empirical Analysis of Heuristic Procedures for Production Planning and Control*. In: Fleischmann, B.; Lasch, R.; Derigs, U.; Domschke, W. und Rieder, U. (Hrsg.): *Operations Research Proceedings 2000. Selected Papers of the Symposium on Operations Research (OR 2000)*, Dresden, 9.-12. September, 2000. Berlin: Springer, S. 266–270.