19. AALE-Konferenz
Luxemburg, 08.-10. März. 2023

Analyse von Energieeffizienzpotenzialen während der Produktion mit Hilfe von Machine Learning

Tobias Schubert[1], Henrik Wilhelm Becker[2], Stefan Kapp[1]

Zusammenfassung

In Anbetracht des steigenden Umweltbewusstseins vieler Unternehmen wird in diesem Vortrag vorgestellt, wie Machine Learning (ML) eingesetzt werden kann, um Produktionsanlagen ressourcenschonender zu betreiben. Es wird ein Verfahren vorgestellt, mit dem die Energieeffizienz von Anlagen mithilfe von unüberwachten ML-Methoden analysiert werden kann. Mit den dargestellten Verfahren kann nicht nur zwischen „Produktion ist aktiv“ und „Produktionspause“ unterschieden werden, sondern vielmehr können ineffiziente Anlagenzustände identifiziert werden. Bezogen auf den Strom- und Druckluftverbräuchen, welche als Eingabedaten für die Modelle dienen, wird ein zeitunabhängiger und ein zeitabhängiger Modellansatz vorgestellt und die unterschiedlichen Fähigkeiten dargestellt. Neben der Erläuterung der algorithmischen Umsetzung sowie der Datenerhebung liegt ein Augenmerk der Arbeit darin, die Vorgehensweise so generisch zu halten, dass sich die mit Festo Didactic Lernfabriken erzielten Ergebnissen mit vertretbarem Aufwand auch auf andere Produktionsanlagen übertragen lassen.

Stichwörter

Maschinelles Lernen, Anlagenanalyse, Ressourcenschonung, Clustering, LSTM-Netze

1 Einführung

Gemäß des EU-Maßnahmenpakets „fit for 55“ [1] sollen bis zum Jahresende 2030 die Treibhausgasemissionen EU-weit im Vergleich zum Jahr 1990 um 55% reduziert werden. Deutschland – wirtschaftlich stärkstes EU-Land aber zugleich auch größter Verursacher entsprechender Emissionen – spielt in diesem Zusammenhang eine zentrale Rolle. Um die Klimaziele für 2030 zu erreichen, wird es nicht genügen, die erneuerbaren Energien massiv auszubauen, vielmehr muss die produzierte Energie auch effizienter eingesetzt werden. Letzteres ist ein Tätigkeitsbereich, in dem deutsche Unternehmen einen substanziellen Beitrag leisten können, indem sie ihre eigene Energieeffizienz verbessern oder die hierfür benötigten Technologien entwickeln. Die Wandlung in Richtung einer klimaneutralen Gesellschaft sollte dabei auch als Möglichkeit gesehen werden, neue Wirtschaftsbereiche zu erschließen. So wird laut dem Bundesministerium für Umwelt [2] im Zeitraum bis 2030 beispielsweise für die Leitmärkte in den Bereichen Umwelttechnik und Ressourceneffizienz ein jährliches Wachstum von 7% prognostiziert.

Ein Teilaspekt der Ressourceneffizienz zielt darauf ab, die Produktivität einer Anlage – bei möglichst geringem Ressourceneinsatz – zu maximieren. Ein Teil der Aufgabenstellung für diesen Vortrag ist es daher, in der Lage zu sein, energetisch ungünstige Anlagenzustände zu erkennen. Diese können z. B. entstehen, indem ein Motorantrieb ungewollt innerhalb einer Anlage noch aktiv ist oder eine Störung den Strom- und/oder Druckluftverbrauch steigert. In diesem Vortrag wird eine Methode vorgestellt, mit der eine derartige Detektion möglich ist. Dieser Ansatz ermöglicht es, die Grundzustände „Produktivbetrieb“, „Produktionspause“ und „Stand-By“ einer Anlage zu erkennen und diese von fehlerhaften Anlagenzuständen zu unterschieden. Darüber hinaus ist es auch möglich, die Prozessabläufe der Anlage zu bewerten, um ggf. Anomalien im Anlagenverhalten zu erkennen. Es werden zwei verschiedene Modellansätze vorgestellt, welche beide auf Methoden des unüberwachten Lernens basieren. Genauer werden „Clustering“-Methoden und LSTM-Netze (künstliche neuronale Netze) eingesetzt, um die Anlage und deren Verhalten zu charakterisieren.

[1] Festo Didactic SE
[2] Festo Didactic SE / Technische Universität Braunschweig

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

2 Technische Grundlagen

In diesem Kapitel werden die für diesen Vortrag benötigten technischen Grundlagen behandelt. Zunächst wird ein Überblick über das Themenfeld des maschinellen Lernens gegeben und dabei auf die grundsätzlichen Unterschiede verschiedener Methoden und Algorithmen eingegangen. Anschließend werden die in diesem Vortrag verwendeten Methoden genauer beschrieben.

2.1 Methoden des maschinellen Lernens

Maschinelles Lernen (ML) wird oft als die Adaption menschlicher Lernprozesse in einem Computersystem definiert. Durch selbstoptimierende Methoden sind diese Systeme in der Lage, anhand von Daten komplexe Sachverhalte zu modellieren und darüber hinaus neues Wissen zu erschließen [3]. Anders ist dies bei konventionellen Programmiermethoden wie bspw. von Vodencarevic et al. [4] detailliert beschreiben: Bei derartigen Herangehensweisen muss zunächst das Systemverhalten hinreichend gut beschrieben werden können. Dies setzt meistens Expertenwissen voraus, um die Bedeutung und den Einfluss der einzelnen Systemvariablen überhaupt mathematisch beschreiben zu können. Erschwerend kommt oftmals hinzu, dass viele Systemvariablen nicht direkt messbar sind und so anhand anderer Variablen ermittelt bzw. approximiert werden müssen. Des Weiteren ist das manuelle Modellieren ein sowohl zeit- als auch arbeitsintensiver Prozess, der bei einer Änderung des Zusammenspiels zwischen einzelnen Systemvariablen und/oder Umgebungsvariablen erneut durchgeführt werden muss.

Bei Methoden des maschinellen Lernens erfolgt die Modellierung unter Anwendung eines generischen Modells und eines entsprechenden Datensatzes, der das Systemverhalten beschreibt. Der Trainingsalgorithmus passt das generische Modell so an, dass es sich gemäß des Datensatzes „möglichst optimal“ verhält [5, S. 2], d.h. für gegebene Eingabedaten möglichst „korrekte“ Ausgaben liefert. Das Anpassen des Modells, auch Trainingsphase genannt, ist ein zum größten Teil automatisierter Prozess, so dass auch Anpassungen des Modells mit geringerem Arbeitsaufwand einhergehen [6, S. 395]. Gemäß Langley & Simon [7, S. 12-14] kann der Einsatz von ML-Methoden dabei prinzipiell in fünf Phasen untergliedert werden: Definition des Problems, Charakterisierung des Systems (in diesem Fall einer Produktionsanlage), Generierung von Trainingsdaten, Evaluation des Modells und zuletzt Implementierung des Modells im Produktivbetrieb.

2.2 Künstliche neuronale Netze

Künstliche neuronale Netze (KNN) sind bei allen drei Unterkategorien des maschinellen Lernens vertreten, sowohl bei den unüberwachten, den überwachten als auch den bestärkenden ML-Methoden. Gleichzeitig können sie auch für eine Vielzahl an Problemtypen angewandt werden, wie z. B. Klassifikation, Regression, Clustering oder Dimensionsreduktion [7]. Die grundlegende Modellstruktur sowie die Funktionsweise des Modells wird in diesem Unterkapitel anhand eines gewöhnlichen „Feed-Forward-Netzwerks“ erläutert. Hauptbestandteil von KNNs sind die sogenannten Neuronen. Diese werden üblicherweise in Schichten angeordnet und bilden so ein Netzwerk von Neuronen, das auch als neuronales Netz bezeichnet wird. Wie in Abb. 1 dargestellt, kann zwischen Eingabeneuronen, Ausgabeneuronen und versteckten Neuronen unterschieden werden. Während die Eingabeneuronen die Schnittstelle zu den Eingabevariablen des Modells bzw. im hier behandelten Fall zu den Daten einer Produktionsanlage bilden, stellt die Ausgabeschicht die Modellantwort zur weiteren Bearbeitung bereit. Die versteckten Neuronen bewerkstelligen den Großteil der Netzberechnung, wobei sowohl die Anzahl der Neuronen pro Schicht als auch die Anzahl der Schichten das Komplexitätsvermögen des Modells bestimmt. [8, S. 1055-1056].

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

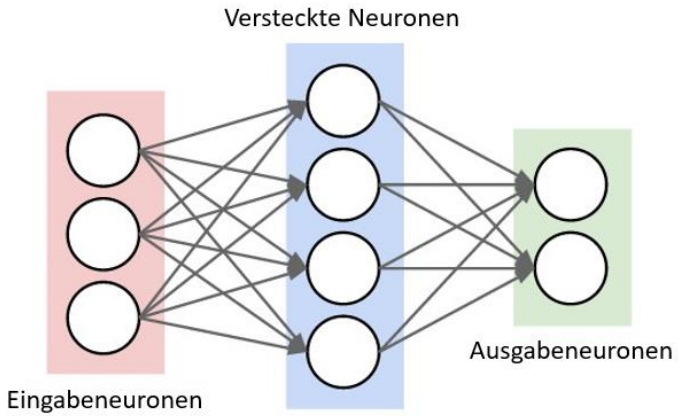

Abbildung 1: Grundstruktur eines Feed-Forward-Netzes

In Abbildung 2 ist die Funktionsweise eines einzelnen Neurons dargestellt. Jedes Neuron ist als ein Multi-Input-Single-Output System aufgebaut, indem es sämtliche Ausgabewerte aller Neuronen der vorherigen Schicht als Eingabewerte bekommt. Diese werden mit einer Gewichtungsmatrix multipliziert, aufsummiert und einer Aktivierungsfunktion (zum Beispiel Tangens-Hyperbolicus- oder Sigmoid-Funktion) übergeben. Die Ausgabe der Aktivierungsfunktion wird mit einer Konstanten addiert und an die Neuronen der nachfolgenden Schicht übergeben [9, S. 268-269]. Die Anpassung der Gewichtsmatrix und der Konstanten anhand der Trainingsdaten wird während des Trainings ausgeführt und ermöglichen so die Anpassung des Modells an die gegebene Problemstellung. Neben diesem „klassischen“ Aufbau sind in der Literatur unzählige andere Arten von Netzwerken bekannt, die wie beispielsweise Convolutional Neural Nets (CNN) besondere Stärken bei der Bildanalyse bieten oder auch die in dieser Arbeit eingesetzten LSTM-Netze (Long Short-Term Memory), die sich für den Einsatz bei Zeitreihen besonders eignen.

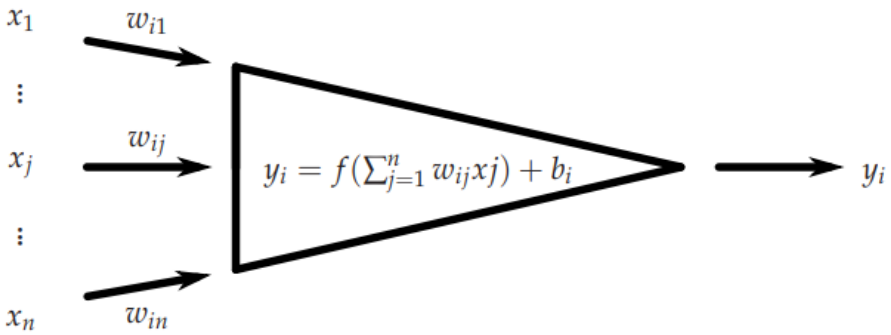

Abbildung 2: Funktionsweise eines einzelnen Neurons

2.3 Clustering-Methoden

Sogenannte Clustering-Methoden werden meistens in Verbindung mit unüberwachten Trainingsstrategien eingesetzt, wobei unüberwacht in diesem Zusammenhang bedeutet, dass die korrekte Antwort des ML-Verfahrens für gegebene Eingabewerte a-priori nicht bekannt ist [10]. Das Ziel ist es, die in den Trainingsdaten implizit gegebenen Zusammenhänge zu extrahieren und Ähnlichkeiten zu ermitteln, damit die Daten in Cluster oder auch charakteristische Gruppen unterteilt werden können. Diese während des Trainings ermittelten Cluster können dann später verwendet werden, um eine Klassifikation durchzuführen [11]. Je nach eingesetztem Verfahren unterteilen Clustering-Methoden die Daten dabei noch unterschiedlichen Kriterien, wie etwa der relativen örtlichen Lage der Daten zueinander oder anhand der Dichteverteilung. Bekannte Beispiele für Clustering-Methoden sind beispielsweise Gaussian Mixture Models, K-Means, DBSCAN, STING und Single Linkage.

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

3 Vorstellung der Modelle

In diesem Kapitel wird der gewählte Versuchsaufbau skizziert und die entwickelten Modelle vorgestellt. Zunächst soll der allgemeine Anlagenaufbau kurz erläutert und vor allem die Datengrundlage genauer beschrieben werden. Anschließend werden die in diesem Vortrag verwendeten Modellansätze detailliert diskutiert.

3.1 Allgemeiner Anlagenaufbau

Der allgemeine Anlagenaufbau basiert auf einer Festo Didactic Lernfabrik bestehend aus insgesamt fünf aneinander gereihten Modulen. Jedes Modul führt einen bestimmten Prozessschritt aus, so dass am Ende der Montage / der Produktion aus diversen Einzelteilen ein „Handy“ entsteht. Enthalten sind u.a. Module, die ein Hochregallager repräsentieren, eine robotergesteuerte Montagezelle und auch eine Presse, um Vorder- und Rückschale des Smartphones miteinander zu verpressen. Sowohl in drei Modulen als auch am Hauptanschluss der Anlage sind sogenannte Energiemessboxen integriert, welche Druckluft- und Stromverbrauch entweder auf Modulebene oder für die gesamte Anlagen messen. Insgesamt werden 26 Signale pro Energiemessbox aufgenommen und in einer Datenbank gespeichert. Die Datenakquise wird durchgeführt, indem auf der Anlage Aufträge ausgeführt und die dabei entstehenden Strom-/Druckluftverbräuche aufgenommen und gespeichert werden. Die anfallenden Daten zwischen den Aufträgen, in denen die Anlage sich in einer „Produktionspause“ befindet, werden ebenfalls erfasst und gespeichert. Diese Daten bilden die Grundlage für die nachfolgend vorgestellten algorithmischen Ansätze.

3.2 Erster Modellansatz: Zeitunabhängiges Clustering

Für den ersten Modellansatz wird angenommen, dass die Leistungsaufnahme der Anlage davon abhängt, in welchem Zustand („Stand-By“, „Produktionspause“, „Produktion“) sich diese befindet. Anhand dieser Tatsache ist es denkbar, dass die Anlage sich anhand ihrer Leistungsaufnahme in entsprechende, eindeutige Cluster bzw. Zustände eingruppieren lässt. Gleichzeitig kann ein unerwünschter Anlagenzustand dadurch definiert werden, dass er keinem der drei genannten Cluster zugewiesen werden kann. D.h. es wird ein zeitunabhängiger „Energieraum“ aufgespannt, in dem es definierte Bereiche für „Stand-By“, „Produktionspause“ sowie „Produktion“ gibt, während ein Fehlerzustand sich dadurch charakterisieren lässt, dass er sich „zwischen“ diesen Bereichen befindet.

Die Datenvorverarbeitung und die Modellierung sind in Abb. 3 dargestellt. Zunächst werden sämtliche Trainingsdaten aller vier Energiemessboxen zeitlich synchronisiert und gegebenenfalls von Fehlerwerten bereinigt. Anschließend wird jeder Datenkanal bezüglich Mittelwertes und Standardabweichung normalisiert. Um die Dimensionalität der für das Training genutzten Daten zu reduzieren, wird dann eine „Principal Component Analysis“ (PCA) durchgeführt. Ziel ist hierbei, die insgesamt 104 ursprünglichen Dimensionen (4 Energiemessboxen mit jeweils 26 Kanälen) auf einen möglichst kleinen und dennoch aussagekräftigen Vektor zu reduzieren. Im nächsten Schritt werden die Datenkanäle zunächst in Sequenzen fester Länge aufgeteilt, die wiederum anhand einer Normalverteilung mit Mittelwert μ und Standard-abweichung σ durch zeitunabhängige Werte approximiert werden. Sämtliche Mittelwerte und Standardabweichungen werden in einem Vektor geordnet, der schlussendlich als Eingabevariable der Clustering-Methode übergeben wird. Insgesamt werden drei verschiedene Clustering-Methoden eingesetzt und miteinander verglichen: K-Means, Gaussian Mixture Models und DBSCAN.

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

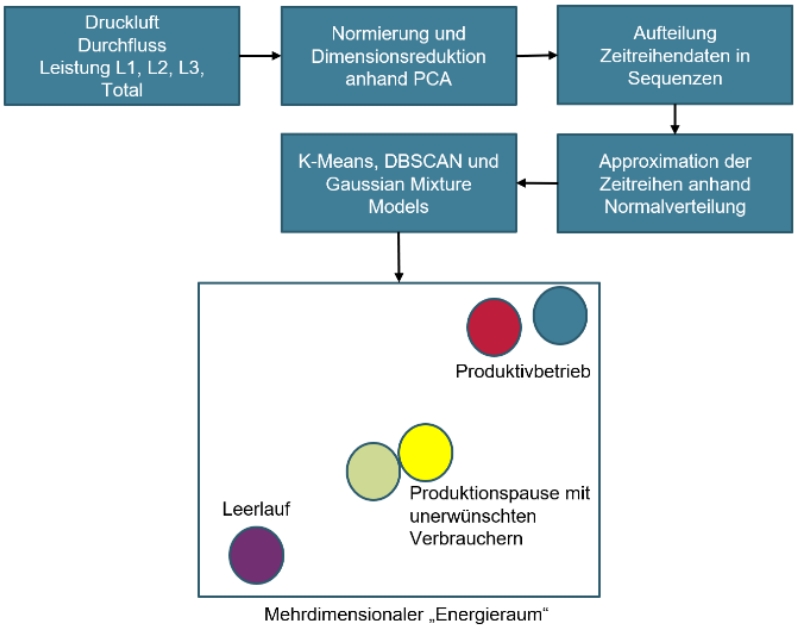

Abbildung 3: Grafische Darstellung des ersten Modellansatzes: Es wird ein zeitunabhängiger "Energieraum" aufgespannt, in dem sich die Anlagenzustände eindeutig voneinander unterscheiden lassen.

3.3 Zweiter Modellansatz: Rekonstruktiver LSTM-Autoencoder

Da die Reduzierung der Daten um die zeitliche Dimension einen erheblichen Informationsverlust darstellt, wird als zweite Variante ein zeitabhängiger Modellansatz präsentiert. Dieser basiert im Wesentlichen auf einem sogenannten LSTM-Autoencoder. LSTM steht für „Long Short-Term Memory“ und charakterisiert neuronale Netze, die auf die Verarbeitung von Zeitreihendaten optimiert sind. Der Zusatz „Autoencoder“ beschreibt in diesem Zusammenhang eine spezielle Netzarchitektur, dessen primäres Ziel es ist, die Identitätsfunktion zu berechnen, d.h. Eingabewerte an das neuronale Netz bestmöglich – mit minimaler Abweichung – an den Ausgängen des Netzes wieder zur Verfügung zu stellen. Das Besondere an dieser Art neuronaler Netze ist dabei die Tatsache, dass es zwei Teilen besteht: Einem Encoder, der die Eingabedaten auf einen möglichst kleinen, aber aussagekräftigen Merkmalsvektor reduziert, und einem Decoder, der aus diesem Vektor wieder die ursprünglichen Daten rekonstruiert. Ein erfolgreiches Training vorausgesetzt, ist ein dahingehend optimiertes Netz in der Lage, bekannte Eingabedaten mit hoher Güte zu rekonstruieren. Im Umkehrschluss kann dieses Verfahren – so wie in dieser Arbeit geschehen – aber auch herangezogen werden, um ein Fehlverhalten der Anlage (nicht in den Trainingsdaten enthalten) zu detektieren, da das neuronale Netze derartige Eingabedaten nur schlecht rekonstruieren kann. D.h. die Rekonstruktionsgüte des Netztes gibt – geeignet gewählte Trainingsdaten vorausgesetzt – Aufschluss, ob sich die Anlage in einem unbekannten, fehlerhaften Zustand befindet oder nicht. Abb. 4 stellt diese Funktionsweise grafisch dar.

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

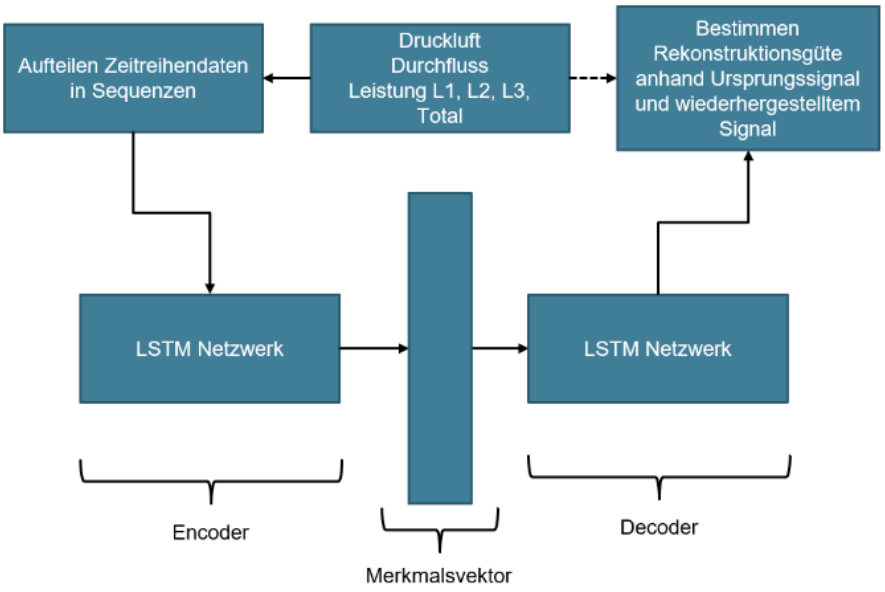

Abbildung 4: Grafische Darstellung des zweiten Modellansatzes: die Daten werden auf einen Merkmalsvektor definierter Länge komprimiert. Anhand des Merkmalsvektors wird durch den Decoder wieder das Ursprungssignal wiederhergestellt.

4 Auswertung der Modelle

In diesem Kapitel werden sowohl Training der Modelle als auch Evaluation im Realbetrieb diskutiert.

4.1 Training zeitunabhängige Clustering-Modelle

Wie bereits erwähnt, wurden drei unterschiedliche Clustering-Methoden implementiert und im Folgenden miteinander verglichen: K-Means, Gaussian Mixture Models sowie DBSCAN. Während der Modellierung wurden identische Trainingsdaten und auch die gleiche Datenvorverarbeitung verwendet. Für jede Clustering-Methode ist eine Parameterstudie durchgeführt worden, bei der Parameter wie beispielsweise Sequenzlänge der Trainingsdaten, Anzahl Cluster, PCA Dimensionen und auch weiter, methodenspezifische Parameter verändert wurden. Jedes Modell wird dabei anhand vier typischer, aus der Literatur bekannten Koeffizienten beurteilt: „Adjusted Rand Score“ (ARS), „Homogeneity Score“ (HM), „Completeness Score“ (CS) und „Silhouette Score“ (SS). Bei allen Koeffizienten ist der Bestwert 1.0. Insgesamt wurden 13 (K-Means), 5 (Gaussian Mixture) bzw. 23 (DBSCAN) Modelle erstellt, wovon das jeweils beste Modell im Realbetrieb anschließend ausgewertet wird. Die nach dem Training ermittelten Koeffizienten sind für das jeweils beste Modell in der nachfolgenden Tabelle angegeben. Wie deutlich ersichtlich ist, bewegen sich alle drei Verfahren hinsichtlich der vier gewählten Kriterien in etwa auf dem gleichen Niveau, d.h. es gibt keine Ausreißer, weder nach oben noch nach unten.

Tabelle 1: Koeffizienten des jeweils besten Modells (ARS: Adjusted Rand Score, HS: Homogeneity Score, CS: Completeness Score, SS: Silhouette Score).

Koeffizient	**K-Means**	**Gauß**	**DBSCAN**
ARS	0,8955	0,9132	0,8955
HS	0,8870	0,9178	0,8985
CS	0,8965	0,9235	0,8675
SS	0,6505	0,5745	0,6080

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

4.2 Training zeitabhängige LSTM-Netze

Auch für den zweiten Modellansatz wurde eine Parameterstudie durchgeführt, bei der sowohl die Modellarchitektur (Anzahl Neuronen und Schichten) als auch die Trainingsparameter (Lernrate, Momentum, Anzahl der Lernepochen, usw.) verändert wurden. Das beste Netz – dasjenige mit dem geringsten Loss-Wert (gemessen mit anhand der mittleren quadratischen Abweichung vom korrekten Ergebnis) – wurde ausgewählt und anschließend anhand des Realbetriebes evaluiert. Da das Training der Netze als auch die Auswahl der Parameter ein durchgängig automatisierter Prozess ist, können beliebig viele Netze ohne erheblichen Mehraufwand trainiert und evaluiert werden. Zu beachten ist jedoch, dass die Trainingsphase bei neuronalen Netzen gegenüber den zuvor betrachteten Clustering-Methoden (bei denen das Training oftmals nur Sekunden dauert) sehr zeitintensiv ist und durchaus Stunden in Anspruch nehmen kann. Dafür sind die zeitbehafteten Verfahren – wie sich im nächsten Abschnitt zeigen wird – in der Lage, Anlagenzustände detaillierter zu identifizieren.

4.3 Realbetrieb der Clustering-Modelle

Für den Realbetrieb wurde zahlreiche unterschiedliche Anlagenzustände sowohl während der Produktion von Smartphones als auch während Produktionspausen sowie Stand-By generiert und protokolliert. Die dabei gewonnenen Daten decken dabei Situationen ab, die auch in den Trainingsdaten zu finden sind, aber auch unerwünschte Zustände, die als „Fehlersituation“ vom jeweiligen ML-Verfahren erkannt werden sollen (bspw. durch Absenken der Druckluftversorgung und durch den Betrieb einzelner Transportbänder, wenn diese eigentlich gestoppt sein sollten). Insgesamt stehen Daten zur Verfügung, die etwa 1,5 Stunden Betrieb repräsentieren. Nach der Datenvorverarbeitung wurden die Eingabedaten durch die entsprechende Cluster-Methode klassifiziert. Beispielhaft sind für K-Means – das beste Verfahren in diesem Experiment – die Ergebnisse in Abb. 5 dargestellt.

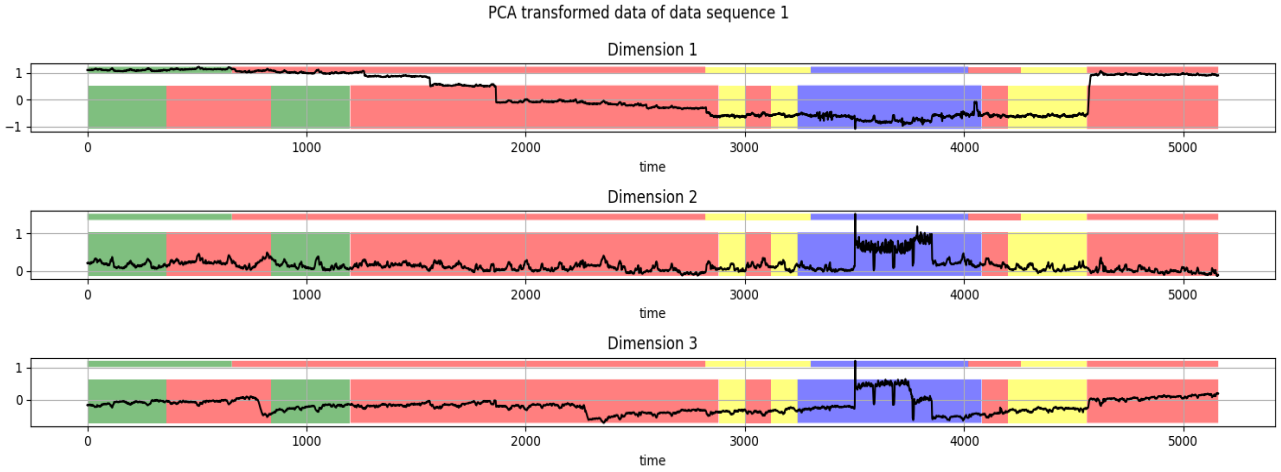

Abbildung 5: Auswertung des K-Means Modells im Realbetrieb. Die Farben geben den Zustand der Anlage an. Die obere dünne Linie ist der tatsächliche Referenzwert zum jeweiligen Zeitpunkt, während der dick hervorgehobene Farbverlauf den von K-Means ermittelten Zustand repräsentiert.

Die Daten können in die Klassen „Produktionsbetrieb“ (Blau), „Produktionspause“ (Gelb), „Stand-by“ (Grün) und „Fehlverhalten“ (Rot) untergliedert werden. Im Vergleich zur jeweils dünn markierten Referenzlinie lässt sich klar ausmachen, dass K-Means bis auf wenige Abschnitte in der Lage ist, den Zustand der Anlage zu detektieren.

Die erreichte Genauigkeit ist in Tabelle 2 für alle drei Methoden angegeben. Bei Genauigkeit ist die Übereinstimmung zwischen Target und Label gemeint, wobei der Wert „1“ eine 100% Erfolgsquote bedeutet. Alle drei Methoden erreichen ein ähnliches Ergebnis, wobei das K-Means Modell etwas besser abschneidet.

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

Tabelle 2: Genauigkeiten der Modelle im Realbetrieb. Die Genauigkeit beschreibt das Verhältnis zwischen richtig klassifizierten Sequenzen zur Gesamtanzahl Sequenzen.

Koeffizient	**K-Means**	**Gauß**	**DBSCAN**
Genauigkeit	0,8023	0,7558	0,7558

4.4 Realbetrieb der LSTM-Netze

LSTM-Netze sind wie auch die zuvor diskutierten Clustering-Methoden in der Lage, die unterschiedlichen „groben“ Anlagenzustände wie Produktion oder Stand-By zuverlässig zu erkennen. Durch den Erhalt der zeitlichen Dimension sind LSTM-Netze aber auch in der Lage, sehr viel detaillierter als Clustering-Methoden feingranulare Anlagenzustände zu ermitteln.

Dies soll im Folgenden exemplarisch für das in der Lernfabrik enthaltene Hochregallager motiviert werden. Das Lager verfügt über insgesamt 32 Plätze, die fertig produzierte Smartphones aufnehmen können, aber auch Halbzeuge wie etwa eine Frontschalte. Das Lager ist in 4 Ebenen zu je 8 Positionen organisiert. Offensichtlich ergeben sich somit geringfügig unterschiedliche Verbrauchsmuster, je nachdem welche Ebene der Lagerroboter zur Ablage/Entnahme eines Bauteils anfahren muss. Da diese Erkennung mit klassischen Clustering-Methoden nur schwer bzw. gegebenenfalls überhaupt nicht möglich ist, sind LSTM-Netze klar im Vorteil. Für das hier vorgestellte Experiment wurden während der Trainingsphase nur Aufträge bzw. die entsprechenden Energiedaten genutzt, bei denen die unterste der vier Ebenen des Lagers angesprochen wurde. Ein Beispiel für ein entsprechendes Energieverbrauchsmuster ist in Abb. 6 dargestellt. Die schwarze Linie entspricht dem tatsächlichen Verbrauch der Anlage, während die rot gestrichelte Line das rekonstruierte Verbrauchsmuster des LSTM-Netzes darstellt. Im Vergleich dazu ist in Abb. 7 der charakteristische Energieverbrauch bei (nicht in den Trainingsdaten enthaltenem) Zugriff auf eine höhere Ebene des Lagers dargestellt. Man erkennt deutlich, dass die Leistungsspitzen schlechter durch das Netz approximiert werden und ein zeitabhängiger Ansatz somit sogar in der Lage ist, feine Unterschiede am Verhalten einer Anlage herauszuarbeiten.

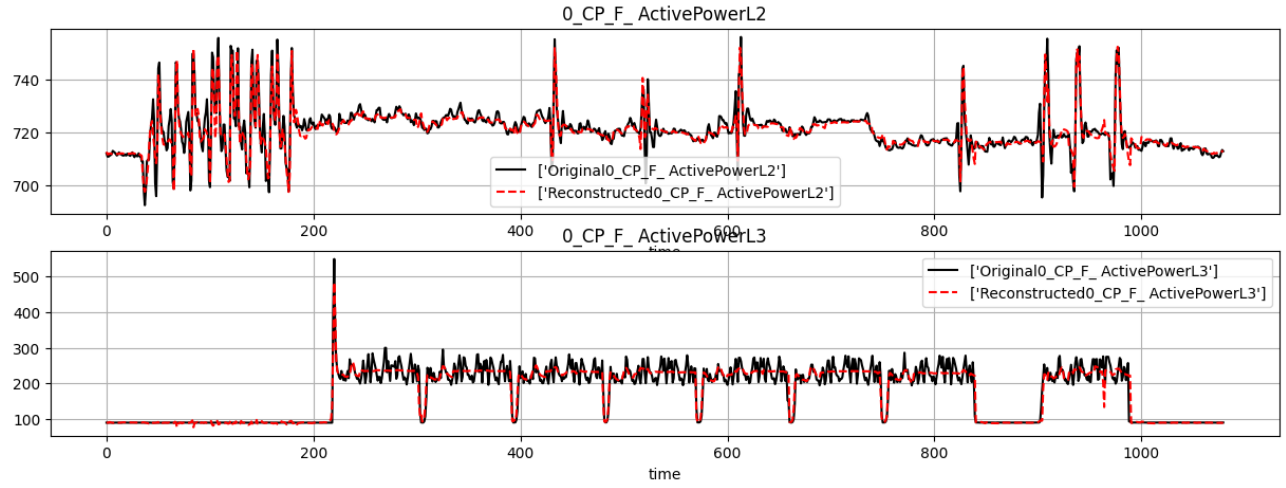

Abbildung 6: Energieverbrauchsmuster eines Auftrages gemäß der in der Trainingsmenge enthaltenen Daten.

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

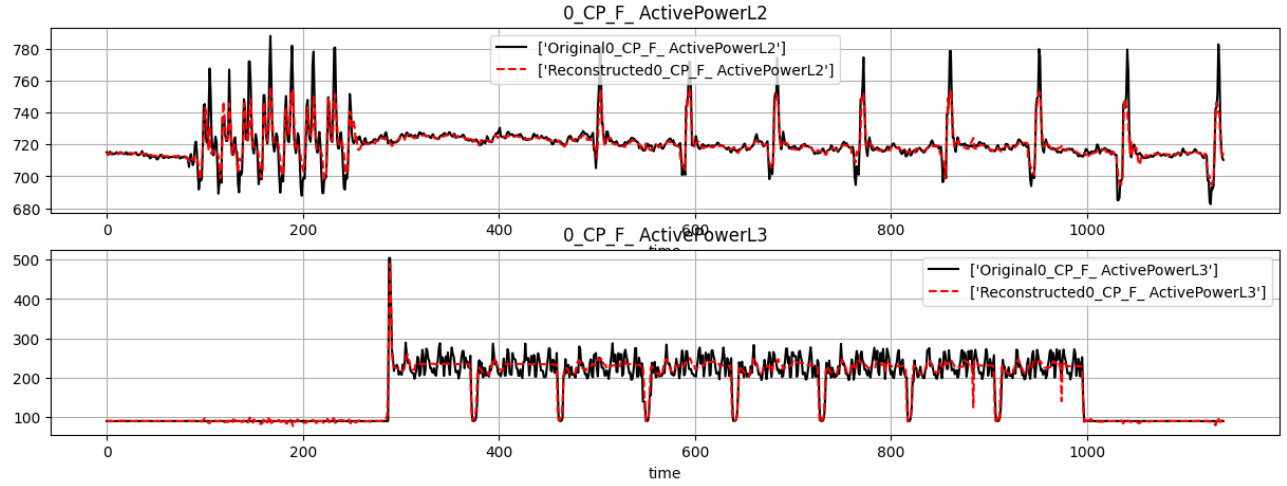

Abbildung 7: Energieverbrauchsmuster eines Auftrages, der nicht in den Trainingsdaten vorzufinden ist. Die Leistungsspitzen im oberen Diagramm werden im Vergleich zu Abb. 6 deutlich schlechter rekonstruiert.

5 Zusammenfassung

Im Rahmen dieses Vortrages wurden zwei Modellansätze zur Detektion von Energieeffizienz-potentialen präsentiert. Basierend auf Techniken des maschinellen Lernens sind die Modelle in der Lage, zwischen den Zuständen „Produktion“, „Pause“, „Stand-By“ und „Fehlverhalten“ einer Anlage zu unterscheiden. Sofern die zeitliche Dimension – wie beim Einsatz von LSTM-Netzen geschehen – erhalten bleibt, lassen sich auch feingranulare Situationen detektieren wie etwa die durch einen Roboter angefahrene Ebene eines Hochregallagers.

Angesichts der generischen Datengrundlage (Strom und Druckluft), welche anhand entsprechender Sensorik einfach erfasst werden kann, ist es möglich, diese Verfahren auch auf andere Anlagen zu übertragen. Für eine derartige Implementierung muss lediglich eine anlagenspezifische Datenakquise durchgeführt und die ML-Modelle daraufhin trainiert werden.

Weitergehende Arbeiten könnten bspw. darin bestehen, zu untersuchen, ob und wie stark zusätzliche Datenquellen wie etwa ein MES-System die Beurteilung eines Anlagezustandes verbessern. Ebenso ist denkbar, Produktionsaufträge verbrauchsoptimiert zu realisieren.

6 Literatur

[1] COMMUNICATION FROM THE COMISSION TO THE EUROPEAN PARLAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS „Fit for 55“: delivering the EU´s 2030 Climate Target on the way to climat neutrality

[2] Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit (BMU), GreenTech made in Germany 2021: Umwelttechnik-Atlas für Deutschland

[3] Hua Wang, Cuiqin Ma, and Lijuan Zhou. “A Brief Review of Machine Learning and Its Application”, pp. 1–4. DOI: 10.1109/ICIECS.2009.5362936

[4] Asmir Vodencarevic et al. “Using behavior models for anomaly detection in hybrid systems”, pp. 1–8. DOI: 10.1109/ICAT.2011.6102093

[5] Gopinath Rebala, Ajay Ravi, and Sanjay Churiwala. An Introduction to Machine Learning. Cham: Springer International Publishing, 2019. DOI: 10.1007 / 978 - 3 - 030 - 15729-6

[6] D. T. Pham and A. A. Afify. “Machine-learning techniques and their applications in manufacturing”. In: Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 219.5 (2005), pp. 395–412. ISSN: 0954-4054. DOI: 10.1243/ 095440505X32274

[7] Pat Langley and A. Herbert Simon. “Applications of Machine Learning and Rule Induction”. In: ISLE Technical Report 95-1 (1995), pp. 1–17

[8] M. Träger et al. “Künstliche neuronale Netze. Theorie und Anwendungen in der Anästhesie, Intensiv- und Notfallmedizin”. In: Der Anästhesist 52.11 (2003), pp. 1055– 1061. ISSN: 0003-2417. DOI: 10.1007/s00101-003-0576-x

[9] Wolfgang Ertel. Grundkurs Künstliche Intelligenz. Wiesbaden: Springer Fachmedien Wiesbaden, 2016. DOI: 10.1007/978-3-658-13549-2

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +

[10] Thorsten Wüst et al. “Machine learning in manufacturing: advantages, challenges, and applications”. In: Production & Manufacturing Research 4.1 (2016), pp. 23–45. DOI: 10.1080/21693277.2016.1192517
[11] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. “Machine Learning from Theory to Algorithms: An Overview”. In: Journal of Physics: Conference Series 1142 (2018), p. 012012. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1142/1/012012

Open Access. © 2023 Tobias Schubert, Henrik Wilhelm Becker, Stefan Kapp1
This work is licensed under the Creative Commons Attribution 4.0 License. +