Reinforcement Learning zur Ableitung autonomer Handelsstrategien in Elektrizitätsmärkten

Philipp REUBER[1], Jacob TRAN, Simon KRAHL, Albert MOSER

FGH e.V., Roermonder Straße 199 52072 Aachen, +49 241 997857-148,
philipp.reuber@fgh-ma.de, https://www.fgh-ma.de/

Kurzfassung: Zur optimalen Beschaffung und Vermarktung von Strom sind geeignete Handelsstrategien erforderlich. Hierzu müssen die komplexen stochastischen Abhängigkeiten zwischen den Einflussfaktoren geeignet berücksichtigt werden. Das Reinforcement Learning ist ein Verfahren, welches zur Ableitung von Handlungsempfehlungen angewendet werden kann. Dieses Verfahren erlernt die Wechselwirkungen zwischen den einzelnen Einflussfaktoren aus historischen Daten. In diesem Beitrag wird ein Verfahren vorgestellt, dass die Ableitungen von Handelsstrategien in Elektrizitätsmärkten ermöglicht. Hierfür wurden verschiedene Verfahren des Reinforcement Learning auf einem historischen Datensatz für den Day-Ahead Spotmarkt trainiert. Ziel des Verfahrens ist es, eine optimale Position im OTC-Markt einzugehen und diese gewinnbringend an der Strombörse zu schließen. Durch eine anschließende Optimierung der Parameter des Verfahrens konnte der Trainingsprozess beschleunigt werden. Die Validierung auf einem Testdatensatz zeigt, dass das Verfahren grundsätzlich zur Ableitung von Handelsstrategien geeignet ist, jedoch der Testgewinn eine sehr hohe Volatilität in Abhängigkeit der Trainingsepisoden besitzt.

Keywords: Stromhandel, Deep Reinforcement Learning

1 Motivation

Die zunehmende Dargebotsabhängigkeit der Stromerzeugung sowie neuartige Verbraucher sorgen für eine steigende Volatilität des Strompreises an der Strombörse. Insbesondere die mittel- und kurzfristige Unsicherheit in der Erzeugung führt zu stark schwankenden Preisen, welche vor allem untertätig und zwischen zwei Handelstagen auftreten. Als Instrument zum Ausgleich dieser Abweichungen gewinnt der kurzfristige Stromhandel an Bedeutung. Die optimale Beschaffung oder Vermarktung von Strom stellt aufgrund der vielfältigen Einflussfaktoren ein komplexes Optimierungsproblem dar und erfordert die Entwicklung geeigneter Handelsstrategien. Hierzu müssen die komplexen stochastischen Abhängigkeiten zwischen den Einflussfaktoren berücksichtigt werden, weshalb das Reinforcement Learning zur Entscheidungsfindung untersucht werden soll.

Verfahren des Machine Learning werden bereits zur Erstellung von Prognosen eingesetzt. Das Reinforcement Learning als Erweiterung dazu kann darüber hinaus zur Bestimmung von optimalen Entscheidungen in Abhängigkeit von äußeren Einflussfaktoren verwendet werden. Die Zusammenhänge der Einflussfaktoren und deren Wechselwirkungen untereinander werden dabei selbstständig erlernt. Nachfolgend soll daher ein solches Verfahren vorgestellt

[1] Jungautor

werden, welches die autonome Ableitung von Handelsentscheidungen unter Berücksichtigung der Wechselwirkungen mit den Einflussfaktoren ermöglicht. [1, 2]

2 Analyse

2.1 Elektrizitätsmärkte

Der elektrische Energiehandel erfolgt in Deutschland im Wesentlichen per „Over The Counter" (OTC)-Handel oder über die Strombörse. Beide Strommärkte können jeweils in einen Terminmarkt, bei welchem Strom bis zu mehreren Jahren im Voraus gehandelt werden kann, und einen Spotmarkt, welcher zur Deckung der restlichen Strommengen vor allem für den nächsten Tag verwendet wird, unterteilt werden. Im Folgenden wird die Preisbildung und die Abwicklung von Handelsgeschäften für den Day-Ahead Spotmarkt betrachtet. [1, 2]

Im Day-Ahead Spotmarkt an der Strombörse werden Stundenkontrakte für den folgenden Tag gehandelt. Bis 12 Uhr können alle Marktteilnehmer im Rahmen einer Auktion Verkaufs- oder Kaufgebote mit Angabe von Menge und Preislimit für jede Stunde des folgenden Tages abgeben. Der Preis einer einzelnen Stunde wird auf Basis aller eingegangenen Gebote bestimmt. Dazu werden die Angebots- und Nachfragekurven konstruiert und der Schnittpunkt beider Kurven bestimmt. Die Bildung dieser Kurven wird als Merit-Order bezeichnet. Dieser Schnittpunkt stellt das Gleichgewicht zwischen Angebot und Nachfrage dar und bestimmt den Gleichgewichtspreis, welcher auch Market-Clearing-Preis genannt wird. Alle Transaktionen über die Strombörse werden über diesen Market-Clearing-Preis abgewickelt. Ausgeführt werden alle Verkaufsgebote, die kleiner oder gleich und alle Kaufgebote, die größer oder gleich dem Market-Clearing-Preis sind. Der Strompreis ergibt sich aus den Kosten des letzten zur Deckung der Nachfrage herangezogenen Kraftwerkes. Als Index für den Strompreis wird der durchschnittliche Preis für alle Stunden des Liefertages verwendet, welcher als Base-Preis bezeichnet wird. [1, 3, 4]

Im OTC-Handel werden bilaterale Verträge zwischen Vertragspartnern außerhalb der Börse geschlossen. Die jeweiligen Vertragsbedingungen und der Preis werden üblicherweise individuell verhandelt. Für diesen Handelsplatz existieren standardisierte Produkte für die Grundlast (Base), die Spitzenlast und die Randstunden. Da bereits vor Auktionsende Handelsgeschäfte im OTC-Markt geschlossen werden, aber der Börsenpreis erst nach 12 Uhr verfügbar ist, können Preisdifferenzen zwischen beiden Marktplätzen auftreten. Die Preisbildung am OTC-Markt erfolgt demnach mittels Prognosen über den Market-Clearing-Preis, welcher sich an der Börse einstellt. Hierzu müssen verschiedenste Einflussfaktoren auf den Strompreis berücksichtigt werden. [1, 3, 4]

Einflussfaktoren auf den Strompreis

Der Strompreis in Deutschland zeichnet sich in den letzten Jahren durch eine zunehmende Volatilität aus. Wie zuvor bereits erwähnt, erfolgt die Preisbildung an der Börse über die Merit-Order. Bei der Merit-Order werden alle Kraftwerksgebote nach ihren jeweiligen Grenzkosten sortiert. Diese bestehen bei konventionellen Kraftwerken zum größten Teil aus variablen Kosten der jeweiligen Brennstoffe und den Kosten für CO2-Zertifikate. Da Anlagen auf Basis erneuerbarer Energien nur geringe Grenzkosten besitzen, verringert sich die Menge an Strom, welche über konventionelle Kraftwerke gedeckt werden muss. Dadurch werden Kraftwerke mit

hohen Grenzkosten nicht mehr zur Deckung der Nachfrage benötigt und der Strompreis sinkt. Technische Störfälle oder geplante Wartungsarbeiten verändern die Menge der verfügbaren konventionellen Kraftwerke und Anlagen auf Basis erneuerbarer Energien und stellen somit einen weiteren Einflussfaktor dar. [2, 5]

Ein weiterer Grund von Preisschwankungen liegt in der Dargebotsabhängigkeit von Anlagen auf Basis erneuerbarer Energien. In Zeiten von viel Wind und Sonne werden viele konventionelle Kraftwerke aus der Merit-Oder gedrängt und der Strompreis fällt. Falls sich die gleichen Wetterbedingungen allerdings kurzfristig und unerwartet einstellen, kann dies zu negativen Strompreisen führen, da konventionelle Kraftwerke aufgrund ihrer thermischen Trägheit ihre Erzeugung nicht schnell genug anpassen können. Die negativen Preise treten auf, wenn das Angebot an Strom die Nachfrage übersteigt. [6, 7]

Auf Seiten der Nachfrage lassen sich tägliche, wöchentliche und jährliche Strukturen erkennen. Die täglichen Trends richten sich hierbei vor allem nach den Arbeitszeiten. Diese begründen auch eine niedrigere Nachfrage an Wochenenden und Feiertagen. Darüber hinaus sind auch saisonale Trends in der Stromnachfrage erkennbar. Vor allem durch die Elektrifizierung des Wärmesektors wird sich dieser Trend weiter verstärken. [8]

Diese vielfältigen Einflussfaktoren erschweren eine Quantifizierung des Strompreises an der Börse. Zusätzlich dazu bestehen Wechselwirkungen zwischen diesen Daten. Aus diesen Gründen soll der Einsatz des Reinforcement Learning zur Ableitung von Handlungsempfehlungen untersucht werden, da solche Modelle die Zusammenhänge selbständig aus historischen Daten erlernen können.

2.2 Reinforcement Learning

Das Reinforcement Learning (RL) stellt neben dem Supervised Learning und dem Unsupervised Learning einen Teil des Maschinellen Lernens dar. RL versucht den natürlichen Lernprozess des Trial-and-Error-Lernens nachzubilden. Hierbei soll eine Maschine, bei diesem Verfahren auch Agent genannt, auf Grundlage der Eingabedaten Entscheidungen treffen. Das Training erfolgt durch Feedback, der Auswirkung der jeweiligen Handlung. Wie in der Natur soll der Agent durch Ausprobieren Aktionen mit einem positiven Feedback bevorzugen.

2.2.1 Mathematische Modellierung des Reinforcement Learning

Die Grundlage zur Modellierung des Reinforcement Learning stellen Markow-Prozesse dar. Mit diesen lassen sich Übergange zwischen verschiedenen Zuständen modellieren. Die Menge aller Zustände wird als S bezeichnet. Durch Einführung eine Reward-Funktion R lassen sich die Markow-Prozesse in Markow-Reward-Prozesse überführen. Diese Funktion weist jedem Zustand einen beliebigen Wert zu, welcher sowohl positiv als auch negativ sein kann. Die Übergänge zwischen den einzelnen Zuständen wird mit einer Menge von Übergangswahrscheinlichkeiten P dargestellt. Zusammen aus der Reward-Funktion lässt sich mit den Übergangswahrscheinlichkeiten eine Bewertung des aktuellen Zustands vornehmen. Hierbei wird die Summe der künftigen Rewards ab dem Zeitpunkt t bis zum Erreichen eines Endzustands gebildet. Diese Summe wird als Gewinn G_t bezeichnet. Neben einer einfachen Summierung kann zur Berechnung des Gewinns auch die diskontierte Summe

$$G_t = R_{t++1} + \gamma R_{t+2} + \cdots = \sum_{i=0}^{\infty} y^i R_{t+1+i} \tag{1}$$

gebildet werden. Mit dem Diskontierungsfaktor $\gamma \in [0,1]$ lässt sich der Einfluss von zukünftigen Rewards abschwächen. Im Falle von $\gamma = 0$ wird beispielsweise nur der unmittelbare Reward betrachtet, im Falle von $\gamma = 1$ erfolgt eine Summierung aller zukünftigen Rewards. Die Einführung des Diskontierungsfaktors lässt sich mathematisch, als auch wirtschaftlich herleiten. Mathematisch sorgt der Faktor dafür, dass auch bei unendlichen Zustandsfolgen der Gewinn für $\gamma < 1$ gegen einen endlichen Wert konvergiert, solange die Werte des Rewards beschränkt sind. Wirtschaftlich sorgt dieser Faktor dafür, dass spätere Rewards nur einen geringeren Wert haben als Rewards näher am aktuellen Zeitpunkt. [9, 10]

Im RL erfolgt ein Übergang in einen anderen Zustand nicht ausschließlich stochastisch, sondern ebenfalls durch eine Menge an Aktionen A. Dadurch werden die Markow-Reward-Prozesse zu Markow-Entscheidungsprozessen erweitert, welche durch das Tupel (S, A, P, R, γ) beschrieben werden können. Die Art und Weise, auf welche Aktionen in einem bestimmten Zustand ausgewählt werden, wird als Policy π bezeichnet. Eine beliebte Policy stellt dabei die greedy-Policy dar, welche die Maximierung des erwarteten Gewinns verfolgt. Der Bestandteil des RL, welcher die Aktion ausführt, wird als Agent bezeichnet. Damit dieser die auf Basis seiner verfolgten Policy bestmögliche Aktion auswählen kann, ist die Bewertung der verschiedenen Aktionen und damit auch der jeweiligen Zustände entscheidend. [9, 10]

Für die Bewertung eines einzelnen Zustandes s erfolgt mit der Wertefunktion

$$V_\pi(s) = \mathbb{E}[G_t | S_t = s] \tag{2}$$

welche den erwarteten Gewinn beschreibt, wenn ab dem aktuellen Zustand s die Policy π erfolgt. Wird in diese Funktion die Definition des Gewinns G_t eingesetzt, ergibt sich folgende rekursive Vorschrift zur Berechnung der Wertefunktion

$$\begin{aligned} V_\pi(s) &= \mathbb{E}[G_t | S_t = s] = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma V_\pi(S_{t+1}) | S_t = s] \\ &= \sum_a \pi(a | S_t = s) \sum_{s'} p(s' | S_t = s, a)(R_{t+1} + \gamma V_\pi(s') \end{aligned} \tag{3}$$

welche auch als Bellman-Gleichung bezeichnet wird und das zentrale Element des Reinforcement Learning darstellt. In dieser Gleichung werden zum einen die Übergänge in einen anderen Zustand durch eine Policy π, als auch durch einen stochastischen Übergang betrachtet. Zur Bewertung einer einzelnen Aktion a im Zustand s wird die Wertefunktion in die sogenannte Aktionswertefunktion

$$\begin{aligned} Q_\pi(s, a) &= \mathbb{E}[R_{t+1} + \gamma Q_\pi(s', a') | S_t = s, A_t = a] \\ &= \sum_{s'} p(s' | S_t = s, A_t = a)(R_{t+1} + \gamma Q_\pi(s', a')) \end{aligned} \tag{4}$$

überführt. Ziel des RL stellt die Maximierung des erwarteten Gewinns dar. Eine Policy, welche dies erreicht, wird als optimale Policy π^* bezeichnet. Analog dazu erfolgt die Benennung der dazu passenden Wertefunktion und Aktionswertefunktion, wobei dort auf π im Index zur Vereinfachung verzichtet werden kann. [9]

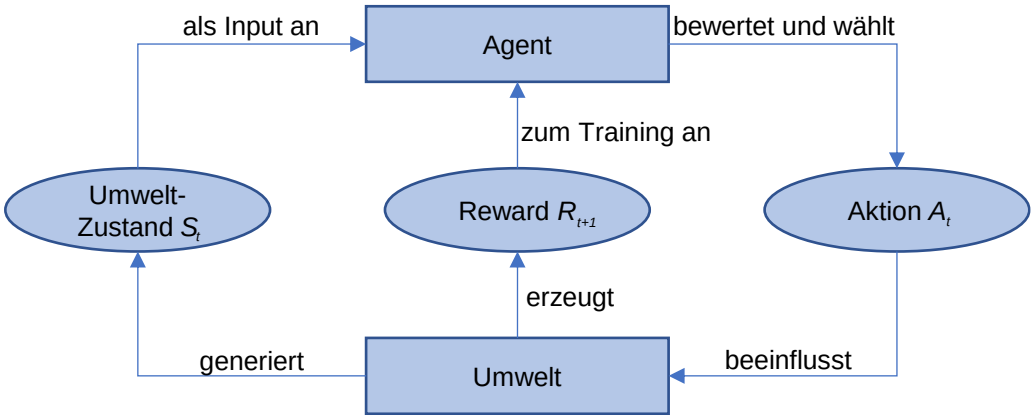

Abbildung 1: Kreislauf des RL

Neben dem Agenten stellt die Umwelt den wichtigsten Baustein des RL dar. Die Aufgabe der Umwelt ist es dabei, in Abhängigkeit der vom Agenten ausgewählten Aktion den nächsten Zustand, welcher auch Umweltzustand genannt wird, und den Reward zu ermitteln. Der Agent nimmt dann auf Grundlage des aktuellen Umweltzustands eine Bewertung der Aktionen vor. Die ausgewählte Aktion beeinflusst die Umwelt in der Berechnung des Rewards und den Übergang in einen neuen Umweltzustands. Somit ergibt sich ein Kreislauf, welcher in Abbildung 1 dargestellt ist. [9]

In den bisherigen Betrachtungen wird von einem vollständigen Wissen über die Umwelt ausgegangen. Somit sind alle Zustände, die jeweiligen Übergänge zwischen diesen und die Rewards bekannt. In solchen Fällen wird von modellbasiertem RL gesprochen. In der Realität ist dieses Modell jedoch unvollständig oder nicht vorhanden, sodass Modell-freie Ansätze verwendet werden, welche die Dynamiken, wie vorhandene Zustände, Zustandsübergänge und Rewards, durch Interaktion mit der Umwelt erlernen. Diese Ansätze basieren auf Näherungen der optimalen Policy und der Wertefunktion bzw. Aktionswertefunktion und werden im Folgenden näher vorgestellt. [9]

Monte Carlo Methoden stellen einen einfachen Modell-freien Ansatz dar. Eine Möglichkeit zur Näherung der Wertefunktion für den Zustand S_t ist es, einen Mittelwert über die Gewinne, welche von diesem Zustand bis zum Erreichen eines Endzustands gesammelt werden, zu bilden. Laut dem Gesetz der großen Zahlen konvergiert diese Näherung der Wertefunktion gegen die optimale Wertefunktion, wenn die Anzahl der Besuche gegen unendlich steigt. Die Wertefunktion allein reicht jedoch nicht aus, um die optimale Policy zu bestimmen. Hierzu wird die Aktionswertefunktion benötigt. Problematisch wird dies, wenn eine deterministische Policy zur Interaktion mit der Umwelt verwendet wird, da diese Policy in einem Zustand immer nur eine Aktion auswählt. Somit wird nicht jedes der mögliche Aktion-Zustand-Paare erreicht und es kann keine Verbesserung der Aktionswertefunktion für diese Paare erfolgen. [9]

Um zu gewährleisten, dass alle Aktion-Zustand-Paare erreicht werden, sind zwei Wege möglich, welche sich in Off-Policy und On-Policy unterteilen lassen. On-Policy Methoden bewerten und verbessern die verwendete Policy. Soll beispielsweise die greedy-Policy erlernt werden, bei welcher es sich um eine deterministische Policy handelt, liegt das oben beschriebene Problem vor. Mit den bisher vorgestellten Methoden lässt sich diese Policy nicht erlernen, da nicht alle Aktion-Zustand-Paare besucht werden könnten und somit keine Konvergenz gegen die optimale Wertefunktion gewährleistet ist. Eine Möglichkeit, dies zu

beheben, stellt die Verwendung der ε-greedy-Policy dar. Bei dieser Policy lässt sich mit dem Parameter $\varepsilon \in [0,1]$ einstellen, mit welcher Wahrscheinlichkeit nicht die Aktion mit dem höchsten erwarteten Gewinn ausgewählt wird. Für $\epsilon \neq 0$ ist gewährleistet, dass alle Aktions-Zustand-Paare bei unendlich vielen Interaktionen erreicht werden, womit sich eine Näherung der greedy-Policy erlernen lässt. Bei der Auswahl des Parameters muss ein Trade-Off zwischen Exploration, der Auswahl von Aktionen, welche nicht den höchsten erwarteten Gewinn aufweisen, und Ausbeutung, dem Verbessern der Wertefunktion für die beste Aktion, getroffen werden. Im Gegensatz dazu werden bei Off-Policy Methoden zwei Policies verwendet. Zur Erzeugung der Beobachtungen wird eine explorative Policy angewendet. Diese Beobachtungen können dann zum Erlernen einer Ziel-Policy, welche sich dann der optimalen Policy nähert, verwendet werden. Off-Policy Methoden weisen meist eine höhere Varianz und eine langsamere Konvergenz als On-Policy Methoden auf, jedoch kann mit diesen Methoden beispielsweise die greedy-Policy selbst und nicht nur eine Näherung dieser erlernt werden. [9]

Das Q-Learning ermöglichte einen der ersten Durchbrüche des RL. Hierbei handelt es sich um ein Off-Policy Verfahren, welches durch

$$Q_\pi(S_t, A_t) \leftarrow Q_\pi(S_t, A_t) + \alpha \left(R_{t+1} + \gamma \max_a Q_\pi(S_t, A_t) - Q_\pi(S_t, A_t) \right) \tag{5}$$

beschrieben wird. Durch den max-Operator wird die explorative Policy π durch die greedy-Policy zu Ende geführt. Somit wird mithilfe einer explorativen Policy die greedy-Policy erlernt. [9]

Die bisher vorgestellten Verfahren basieren auf der Wertefunktion bzw. Aktionswertefunktion und werden aus diesem Grund auch wertebasiert genannt. Im Gegensatz dazu existieren Policy-basierte Verfahren, welche eine direkte Ableitung der Policy als Ziel haben, ohne zuvor eine Wertefunktion ermitteln zu müssen. Verfahren, welche sowohl die Policy als auch gleichzeitig die Wertefunktion annähern, werden als Actor-Critic Methoden bezeichnet. Als Actor wird dabei die Policy bezeichnet, welche die Aktion auswählt. Analog dazu wird die (Aktions-)Wertefunktion als Critic bezeichnet. [9]

Policy-basierte Methoden erlernen eine parametrisierte Policy $\pi(a|s, \theta)$ mit dem Parametervektor θ. Die Veränderung der Parameter führen zu einer Veränderung der Policy. Der iterative Prozess zur Näherung der Policy erfolgt dabei basierend auf Gradienten. Hierbei wird zunächst eine Bewertungsfunktion der Policy $J(\theta)$ bestimmt. Die einzige Bedingung an die Parametrisierung der Policy ist, dass diese differenzierbar sein muss. Die Iterationsvorschrift ist durch

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)} \tag{6}$$

definiert. $\widehat{\nabla J(\theta_t)}$ bezeichnet dabei eine stochastische Näherung, deren Erwartungswert den tatsächlichen Gradienten annähert. Der Gradient dieser Bewertungsfunktion zeigt dabei in Richtung des steilsten Anstiegs. Die Parameteranpassung erfolgt in diese Richtung, sodass die Bewertungsfunktion einen höheren Wert annimmt als zuvor. Der Parameter α wird als Lernrate bezeichnet und dient zur Steuerung der Änderung der einzelnen Parameter. Die Wahl der Lernrate stellt einen wichtigen Bestandteil vieler Algorithmen dar. Dieser Parameter ist mit der Lernrate von Künstlichen Neuronalen Netzen vergleichbar, welche ebenfalls mittels

gradientenbasierten Verfahren ihre Parameteranpassungen vornehmen. Die eigentliche Policy wird durch

$$\pi(a|s,\theta) = \frac{e^{h(s,a,\theta)}}{\sum_A e^{h(s,A,\theta)}} \tag{7}$$

mit der Softmax-Funktion bestimmt. Der Term $h(s,a,\theta)$ bewertet für jede Aktion a die Auswahlwahrscheinlichkeit, welche beliebige Werte annehmen kann. Durch Anwendung der Softmax-Funktion hat die Summe über die einzelnen Aktionen den Wert eins, sodass es sich hierbei um eine stochastische Policy handelt, welche sich allerdings auch einer deterministischen Policy annähern kann. Der Vorteil von Policy-basierten Methoden liegt in der Möglichkeit zur Parametrisierung der Policy-Funktion, da durch diese Wissen über die Umwelt bereits berücksichtigt werden kann. Ebenfalls kann die Parametrisierung der Policy leichter sein als eine Näherung der Wertefunktion. [9]

Proximal Policy Optimization (PPO) gehört zu der Gruppe der On-Policy Actor-Critic Verfahren und zählt heute zu einem der Standardverfahren des RL. Die zu maximierende Zielfunktion ist dabei durch

$$L(\theta) = \widehat{\mathbb{E}_t}\left[\min\left(r_t(\theta)\hat{A}_{\pi,t}(S_t,A_t), clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\pi,t}(S_t,A_t)\right)\right] \tag{8}$$

mit $r_t(\theta) = \frac{\pi(A_t|S_t,\theta)}{\pi_{Alt}(A_t|S_t,\theta)}$ gegeben. Somit bezeichnet $r_t(\theta)$ das Verhältnis der beiden Wahrscheinlichkeitsverteilungen. Der Clip-Term in dieser Formel sorgt dafür, dass nur Änderungen der Policy in einem bestimmten Bereich möglich sind. Ist die Änderung der Policy zu groß bzw. zu klein wird der Wert auf $1+\epsilon$ bzw. $1-\epsilon$ beschränkt. $\hat{A}_\pi(s,a)$ steht für den Aktionsvorteil, welcher durch

$$\hat{A}_\pi(s,a) = Q_\pi(s,a) - V\pi(s) \tag{9}$$

gegeben ist. Der Aktionsvorteil gibt dabei für ein bestimmtes Aktion-Zustand-Paar an, ob der erwartete Gewinn dieses Paares höher ist als die Wertefunktion dieses Zustands. Der Aktionsvorteil darf dabei nicht mit der Aktion A_t zum Zeitpunkt t verwechselt werden. Die Aktion A_t wird nicht mit dem Index π versehen, sodass hier eine Verwechslung ausgeschlossen ist. Der min-Operator in Formel (8) wählt schließlich das Minimum des Produkts aus beschränktem oder unbeschränktem Verhältnis der Policy und dem Aktionsvorteil. [11]

Ein weiteres Actor-Critic Verfahren ist Advantage-Actor-Critic (A2C). Dieses On-Policy Verfahren ist durch

$$(G_t - b_t(S_t) + \beta H(\pi(A_t|S_t,\theta)))\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta)} \tag{10}$$

gegeben, wobei $b_t(S_t)$ für die erlernte Wertefunktion V_π steht, welche auch als Baseline bezeichnet wird. $H(\pi(A_t|S_t,\theta)$ ist die Entropie der Policy, welche ein Maß für den Informationsgehalt einer Wahrscheinlichkeitsverteilung darstellt. Die Entropie nimmt ihr Maximum bei einer Gleichverteilung und ihr Minimum bei einer deterministischen Policy an. Insgesamt kann die Formel (10) als eine Näherung des Aktionsvorteils gesehen werden, da die Aktionswertefunktion $Q_\pi(S_t,A_t)$ eine Näherung des erwarteten Gewinns G_t ist. Somit wird der Gradient der Policy mit der Näherung des Aktionsvorteils skaliert, was zu einem Actor-

Critic Ansatz führt. Durch den Parameter β kann der Einfluss der Entropie gesteuert werden. [12]

Deep Reinforcement Learning

In den zuvor vorgestellten Algorithmen werden parametrisierte Funktionen (Wertefunktion, Aktionswertefunktion oder Policy) betrachtet, mit denen die jeweilige optimale Funktion approximiert werden soll. Da mehrschichtige Künstliche Neuronale Netze (KNN) mit einer geeigneten Anzahl an Neuronen jede beliebige Funktion beliebig genau approximieren können, lassen sich diese auch zur Approximation der Funktionen des RL verwenden. Dies wird als Deep Reinforcement Learning bezeichnet. An dieser Stelle wird auf die Erklärung von KNN verzichtet und beispielsweise auf [13, 14] verwiesen. Der Parametervektor der jeweiligen Funktion des RL kann dabei als Gewichte des KNN gesehen werden. Der erste Algorithmus, welcher erfolgreich KNN zur Funktionsapproximation verwendet hat, stellte das Deep Q-Learning dar. Der Agent wird dabei meist als Deep-Q-Network (DQN) bezeichnet, welches auch als Synonym für das gesamte Verfahren verwendet wird. [15]

Zur Anwendung von KNN zur Approximation der Aktionswertefunktion musste der Q-Learning Algorithmus angepasst werden, da ohne diese Anpassungen KNN instabil sind oder der Trainingsprozess divergiert. Ein Grund hierfür ist die Korrelation zwischen den einzelnen Beobachtungen der Umwelt. Darüber hinaus können bereits kleine Änderungen der Aktionswertefunktion signifikante Änderungen der Policy bewirken. Zuletzt korreliert die momentane Näherung der Aktionswertefunktion Q_π mit dem Zielwert $R_{t+1} + \gamma Q_\pi(S_t, A_t)$ aus Formel (5). [15]

Eine der notwendigen Anpassungen stellt die Verwendung von zwei Parametervektoren θ und θ^- dar. Der Parametervektor θ^- wird zur Auswahl der Aktion mit dem höchsten erwarteten Gewinn durch den max-Operator in Formel (11) verwendet. Dieser wird für eine feste Anzahl an Schritten τ konstant gehalten und dann auf die Werte von θ gesetzt. Hieraus ergibt sich folgende Funktion:

$$L(\theta_t) = \mathbb{E}\left[\left(R_{t+1} + \gamma \max_a Q_\pi(a, S_t, \theta_t^-) - Q_\pi(A_t, S_t, \theta_t)\right)^2\right] \tag{11}$$

Um die Korrelation zwischen den einzelnen Beobachtungen zu verringern, wird das Experience Replay verwendet. Hierbei werden die einzelnen Schritte einer Beobachtung mit den Werten $(S_t, A_t, R_{t+1}, S_{t+1})$ gespeichert und in zufälliger Reihenfolge zum Training an das KNN übergeben. Diese beiden Anpassungen tragen maßgeblich zum Erfolg des DQN bei. [15]

Das Vorgehen zum Einsatz von KNN bei anderen Algorithmen ist analog zu dem Vorgehen beim DQN. Bei Actor-Critic Ansätzen wird sowohl ein KNN für die Wertefunktion als auch für die Policy-Funktion benötigt. Die Größe und Architektur der jeweiligen KNN ist dabei beliebig und auch die Parameter beider Netze können getrennt werden.

3 Modellbildung

Zur Modellierung des Deep Reinforcement Learning stehen mehrere vorgefertigte Frameworks zur Verfügung, welche sich im Wesentlichen in den implementierten Verfahren des RL unterscheiden. Das Framework Stable Baselines 3 [16] stellt die dritte Weiterentwicklung von OpenAI Baselines dar. OpenAI Baselines wurde als Basis und

Benchmark für die Entwicklung neuer Verfahren des RL entworfen. Die Weiterentwicklungen zu Stable Baselines bietet verbesserte Möglichkeiten zur Anwendung der Algorithmen in Machine Learning Projekten. Die dritte Version basiert dabei auf dem Deep Learning Framework PyTorch. Zu den bereits implementierten Verfahren zählen unter anderem DQN, A2C und PPO. Im Folgenden wird die Modellierung der Umwelt und des Agenten vorgestellt.

Modellierung der Umwelt

Die Aufgaben der Umwelt sind die Übergabe von Umweltzuständen, auch Beobachtungen genannt, an den Agenten, das Durchführen der vom Agenten ausgewählten Aktionen und die Berechnung des Rewards. Da die Umwelt die Durchführung der Aktionen übernimmt, können hier Nebenbedingungen für den Handel implementiert werden. Um zu verhindern, dass eine zu große Position eingegangen wird oder die Position immer weiter vergrößert wird, darf das Modell Positionen nur in einer festgelegten Höhe eingehen. Eine Position durch Kaufen wird dabei als *Long* bezeichnet, eine durch Verkaufen als *Short*. Der Wechsel von einer *Short* in eine *Long* Position (Zweifaches-Kaufen) bzw. umgekehrt ist möglich. Hieraus ergeben sich die fünf möglichen Aktionen Kaufen, Verkaufen, Halten, Zweifaches-Kaufen und Zweifaches-Verkaufen. Der Agent kann sich somit in den drei Handelszuständen *Long*, *Short* oder *keine Position* befinden. Da nicht in jedem dieser Zustände jede Aktion möglich sein soll – zum Beispiel um zu große Positionen zu verhindern – wird ein alternatives Vorgehen gewählt. Statt der zuvor genannten fünf Aktionen kann der Agent zwischen den einzustellenden Positionen *Long*, *Short* oder *keine Position* wählen. Je nachdem, in welchem der Zustände sich der Agent zuvor befunden hat, führt die Umwelt dann die Aktionen Kaufen, Verkaufen, Halten, Zweifaches-Kaufen oder Zweifaches-Verkaufen durch. Durch dieses Vorgehen ist in jedem Handelszustand jede der Aktionen möglich. Befindet sich das Modell beispielsweise im Zustand *Long* und der Agent wählt die Aktion *Long* aus, so führt die Umwelt die Aktion Halten aus.

Damit der Agent eine Aktion bewerten und auswählen kann, benötigt dieser eine Beobachtung der Umwelt. Die Beobachtung beinhaltet die jeweiligen Werte der einzelnen Merkmale des Datensatzes zum betrachteten Handelszeitpunkt. Darüber hinaus wird der aktuelle Zustand des Modells und der Preis, zu welchem eine Position eingegangen wurde an den Agenten übergeben. Für den Fall, dass das Modell *keine Position* im aktuellen Zustand hält, wird ein Preis von null verwendet. Eine der Hauptvoraussetzungen ist es, dass der Agent zum Ende eines Handelstages sich im Zustand *keine Position* befinden muss. Nachdem der Agent eine Aktion ausgewählt hat, wird diese an die Umwelt übergeben. Diese ermittelt auf Grundlage der ausgewählten Aktion den neuen Zustand des Modells und übergibt diesen zusammen mit dem Reward im nächsten Zeitschritt an den Agenten.

Der Reward stellt einen zentralen Bestandteil der Umwelt dar. Die einfachste Funktion zur Berechnung des Rewards stellt die Funktion

$$R_{t+1} = \begin{cases} p_t - p_{t-i} & \text{wenn schließen durch Kauf} \\ 0 & \text{wenn Halten oder keine Aktion} \\ p_{t-i} - p_t & \text{wenn schließen durch Verkauf} \end{cases} \tag{12}$$

dar. Hierbei steht p_t für den aktuellen Strompreis und p_{t-1} für den Preis, zu welchem die Position eingegangen wurde. Durch dieses Vorgehen ergibt sich jedoch ein Problem. Neben der Aktion zur Schließung der Position hat auch die Aktion zur Öffnung der Position einen Einfluss auf den erreichten Reward. Allerdings erhält der Agent nur für die schließende Aktion

diesen Reward und nicht für die öffnende Aktion. Zusätzlich dazu kann es notwendig sein, eine offene Position erst nach einigen Zeitschritten zu schließen, damit ein Gewinn erzielt wird bzw. der Verlust begrenzt wird. Dieses Problem der Zuteilung des Erfolges an eine Aktion wird Credit Assignment Problem genannt. Eine Möglichkeit dieses Problem zu beheben, stellt das Reward Engineering dar. Hierzu wird die Reward-Funktion aus Formel (13) zu

$$R_{t+1} = \begin{cases} p_t - \bar{p}_t & \text{wenn Kauf} \\ 0 & \text{wenn Halten} \\ \bar{p}_t - p_t & \text{wenn Verkauf} \end{cases} \tag{13}$$

modifiziert. In dieser neuen Reward-Funktion wird nun auch für das Eingehen einer Position ein Reward erzeugt. Hierzu wird ein gleitender Mittelwert über die vergangenen Strompreise eines Handelstages $\bar{p}_t = (1-\nu)\bar{p}_{t-1} + \nu\bar{p}_t)$ gebildet. Mit dem Parameter ν kann der Einfluss zwischen dem vorherigen gleitenden Mittelwert und dem aktuellen Preis eingestellt werden. Wird bei einem Handelsgeschäft eine Long Position eingegangen oder eine Short Position geschlossen wird ein positiver Reward generiert, wenn der aktuelle Strompreis niedriger als $\bar{p}_t$ ist. Durch das Reward Engineering wird nun für jede Aktion außer Halten ein Reward berechnet.

Agent

Im vorherigen Abschnitt wurden drei mögliche Aktionen vorgestellt, zwischen denen der Agent wählen kann. Dazu muss zunächst eine Bewertung dieser Aktionen vorgenommen werden. Je nach verwendetem RL-Verfahren erfolgt die Auswahl entweder über die (Aktions-)Wertefunktion, die Policy-Funktion oder durch beide Funktionen. Folglich stellt das verwendete RL-Verfahren einen Freiheitsgrad des Agenten dar.

Das hier vorgestellte Verfahren basiert auf dem Deep Reinforcement Learning, sodass zur Approximation der Bewertungsfunktion ein oder mehrere KNN verwendet werden. Der durch die Umwelt generierte Reward dient zur Verbesserung dieser Approximation. Dieser Trainingsprozess lässt sich über verschiedene Parameter beeinflussen, welche sich in Parameter des KNN und Parameter des RL unterteilen lassen. Zu ersteren gehört die Netzarchitektur und -größe, die Aktivierungsfunktion der Neuronen, die Batch-Größe und die Lernrate. Letztere sind abgesehen vom Diskontierungsfaktor des Gewinns spezifisch für das jeweilige RL-Verfahren. Für DQN lässt sich beispielsweise über den Parameter ϵ der ϵ-greedy-Policy beeinflussen, mit welcher Wahrscheinlichkeit eine explorative Aktion ausgewählt wird.

Die Architektur des KNN kann frei gewählt werden, wobei meist jedoch einfache Feed-Forward KNN verwendet werden. Die Anzahl der Neuronen in der Eingabeschicht hängt dabei von der Anzahl der Merkmale in den von der Umwelt generierten Beobachtungen ab. Die Ausgabeschicht besitzt drei Neuronen, da eine Bewertung von drei Aktionen vorgenommen werden soll. Die Anzahl der verdeckten Schichten und die Anzahl der Neuronen in diesen kann beliebig gewählt werden, wobei sowohl eine zu große als auch eine zu kleine Anzahl sich negativ auf die Ergebnisse auswirken. Zunächst wird das KNN zufällig initialisiert, weshalb die Auswahl der Aktionen zunächst auch zufällig erfolgt. Erst durch einen fortschreitenden Lernprozess können die Auswirkungen bestimmter Aktionen auf den Reward ermittelt und so adäquate Empfehlungen für den Stromhandel getroffen werden.

4 Verfahren

Transformation der Eingangsdaten

Ziel des Verfahrens ist es, autonome Handelsstrategien in Elektrizitätsmärkten mittels der Anwendung des oben beschriebenen Modells abzuleiten. Eingangsdaten im Verfahren sind Daten zu den verschiedenen Einflussfaktoren auf den Strompreis in Deutschland. Die Prognosen für EE-Anlagen sind dabei in Wind und Photovoltaik (PV) unterteilt. Konventionelle Kraftwerke werden über ihre Verfügbarkeiten getrennt nach Primärenergieträger berücksichtigt. Zusätzlich dazu werden die Kraftwerksverfügbarkeiten von Wasserkraft, unterteilt in Pumpspeicher, Fließwasser und sonstige Wasserkraft betrachtet. Die Brennstoffkosten und CO2 Zertifikate haben einen massiven Einfluss auf die Reihenfolge der Kraftwerke in der Merit-Order, weshalb diese ebenfalls berücksichtigt werden müssen. Die Temperatur besitzt ebenfalls einen Einfluss auf den Strompreis und wird in den Eingabedaten berücksichtigt.

Da der Strompreis auch durch Stromimporte bzw. -exporte beeinflusst werden kann, müssen neben den Daten für die deutsche Gebotszone Daten für die Stromerzeugung und den Stromverbrauch deutscher Nachbarländer betrachtet werden. Hierbei wird sich auf die Länder Frankreich, Belgien, Niederlande und Österreich beschränkt, da diese Teil des Flow-Based Market Coupling sind. Zusätzlich dazu werden Prognosen über die Stromerzeugung durch Kernenergie in der Schweiz betrachtet.

Weitere Einflussfaktoren stellen das Datum und der Wochentag dar. Das Datum wird dabei als Tag im Jahr und durch den jeweiligen Monat abgebildet. Dies wird jeweils durch eine eigene Sinus-Cosinus-Transformation durchgeführt. Für die Wochentage erfolgt eine Einteilung in die fünf Kategorien Montage bzw. Tage nach einem Feiertag, Werktage (Dienstag-Donnerstag), Freitage, Samstage und Sonntage bzw. Feiertage. Dazu wird eine 1-zu-N-Transformation verwendet.

Die zuvor genannten Daten liegen in unterschiedlichen Zeitschritten vor. Die Brennstoffkosten, Zertifikate und Kraftwerksverfügbarkeiten gelten jeweils für einen Tag wohingegen beispielsweise die Prognosen für EE-Anlagen eine stündliche Auflösung besitzen. Ziel dieses Verfahrens ist jedoch die Ableitung von Handelsstrategien für Base-Produkte. Somit muss eine Transformation der Daten erfolgen, damit aus den stündlichen Daten eine Ermittlung des Base-Preises erfolgen kann. Hierzu werden die Eingabedaten zunächst auf ihren Einfluss zur Bestimmung des stündlichen Strompreises verwendet. Dazu wird das Random Forest Verfahren angewendet, welches auf Entscheidungsbäumen basiert.

Entscheidungsbäume versuchen, die Grundgesamtheit der Eingabedaten durch verschiedene Kriterien in möglichst ähnliche Untergruppen aufzuteilen. Diese Aufteilung erfolgt so lange, bis eine maximale Anzahl an Entscheidungskriterien erreicht ist oder keine weitere Aufteilung mehr möglich ist. Das Random Forest Verfahren führt dieses Vorgehen auf zufällig bestimmten Teilmengen des gesamten Datensatzes durch. Aus diesen einzelnen Teilbäumen wird durch ein Mehrheitsvotum die Wichtigkeit der einzelnen Merkmale der Eingangsdaten bestimmt. Dabei ist ein Merkmal umso wichtiger, je früher es zur Aufteilung der Daten in Teilmengen verwendet wird. Zusätzlich zu dieser Wichtigkeit liefert dieses Regressionsverfahren ein Bestimmtheitsmaß, welches angibt, wie gut sich der Zielwert aus den Eingabedaten bestimmen lässt. [17]

Durch die Untersuchungen mit dem Random Forest Verfahren können verschiedene Transformationsmöglichkeiten zur Bestimmung des Base-Preises untersucht und mit den stündlichen Ergebnissen verglichen werden. Zusätzlich dazu können verschiedenen Zeiträume aus den historischen Daten bezüglich ihrer Bestimmtheitsmaße zur Bestimmung des Strompreises miteinander verglichen werden. Grundsätzlich benötigen Methoden des Maschinellen Lernens eine ausreichend große Datenmenge. Da sich das Energieversorgungssystem jedoch in einem stetigen Wandel befindet, ändert sich die Wichtigkeit der verschiedenen Einflussfaktoren auf den Strompreis und die Wechselwirkungen untereinander innerhalb der historischen Daten. Beispiele hierfür sind die Gebotszonentrennung von Deutschland und Österreich, welche zum 01.10.2018 erfolgt ist, oder der Zubau von EE-Anlagen. Das Bestimmtheitsmaß des Random Forest Verfahrens wird hier verwendet, um eine signifikante Änderung in den historischen Daten zu identifizieren und einen geeigneten Zeitraum für die weiteren Untersuchungen zu bestimmen.

Bevor mit dem Training und Test begonnen werden kann erfolgt eine Normalisierung der Eingabedaten, welche für die einzelnen Merkmale des Datensatzes getrennt durchgeführt wird. Die Temperatur wird dabei zwischen null und eins skaliert. Alle anderen Daten werden jeweils durch den größten Wert des jeweiligen Merkmals geteilt, sodass sich beispielsweise für die Kraftwerksverfügbarkeit ein Wert ergibt, welcher die prozentuale Verfügbarkeit dieses Kraftwerkstyp angibt, welche ebenfalls zwischen null und eins liegt.

Training und Test

Nach der Transformation der Eingabedaten wird der Datensatz in einen Trainings- und Testdatensatz geteilt. Hierbei werden zufällig 20% der Handelstage als Testdatensatz zur Validierung des Verfahrens gewählt. Zunächst wird je ein Modell für die RL-Verfahren PPO und DQN mit dem Trainingsdatensatz trainiert. Das Training erfolgt dabei für jeweils 20.000 Episoden. Eine Episode steht dabei für eine komplette Iteration über den Datensatz. Im Anschluss daran wird der Trainingsgewinn im Verlauf der Trainingsepochen dargestellt. Hierbei steht zum einen der maximale Gewinn der Algorithmen und zum anderen wie schnell dieses Maximum erreicht wird im Fokus. Diese Untersuchung dient dazu, ein vielversprechendes Modell zu identifizieren, welches im Anschluss parametriert wird.

Nach der Auswahl eines vielversprechenden Modells erfolgt die Parametrierung der Netzstrukturparameter des oder der KNN sowie der spezifischen Parameter des RL-Verfahrens. Hierzu wird das Framework Optuna [18] verwendet. Diesem kann für jeden Parameter ein Suchraum vorgegeben werden. Das Framework führt dann eine Suche nach einer optimalen Parameterkombination durch, welche den höchsten Gewinn liefert. Hierbei werden die einzelnen Untersuchungen gespeichert, sodass Auswertungen über den Verlauf der Parameterkombinationen möglich sind und die Wichtigkeit der einzelnen Parameter auf die Höhe des Gewinns untersucht werden kann.

Nach der Parametrierung erfolgt die Validierung des Verfahrens mit der gefundenen besten Parameterkombination. Hierbei wird sowohl der Verlauf des Trainings- als auch des Testgewinns in Abhängigkeit der Trainingsepisoden betrachtet und mit verschiedenen Referenzstrategien verglichen.

5 Exemplarische Untersuchungen

Die Validierung des Verfahrens erfolgte durch exemplarische Untersuchung anhand von historischen Daten für die deutsche Gebotszone im Umfang von zweieinhalb Jahren im Zeitraum vom 01.01.2018 bis zum 31.05.2020. Somit liegt der Beginn der Corona-Pandemie innerhalb des Untersuchungszeitraums. Von diesem Zeitraum werden 175 zufällig ausgewählte Tage zur Validierung des Verfahrens herangezogen. Zur Vereinfachung der Aufgabenstellung werden folgende Annahmen getroffen:

1. Das RL-Modell kann lediglich eine einzige Position im OTC-Handel eingehen. Gleichzeitig wird ein gegenläufiges Handelsgeschäft an der Börse platziert, sodass diese offene Position geschlossen wird.
2. Das Gebot an der Börse wird immer ausgeführt. Somit verbleibt keine offene Position, deren Ausgleich mit hohen Kosten verbunden wäre.

Aufgabe des Agenten ist es somit, eine Vorhersage des Base-Preises an der Börse vorzunehmen und aufgrund dieser im OTC-Handel eine gewinnbringende Position einzugehen.

Transformation der Eingabedaten

Alle zuvor genannten Einflussfaktoren, welche als Eingabedaten im Modell verwendet werden sollen, liegen in einem Zeitraum vom 01.05.2013 bis zum 31.05.2020 vor. Bis zum 30.09.2018 wurden die Marktgebiete Deutschland und Österreich gemeinsam an der Börse gehandelt und besaßen somit den gleichen Strompreis. Ab dem 01.10.2018 erfolgte die Trennung dieser Marktgebiete. Jedoch konnte bereits ab Ende 2017 getrennte Produkte für Deutschland und Österreich gehandelt werden. Aus diesem Grund ist eine Verwendung des gesamten historischen Datensatzes nicht sinnvoll, da es zu einer signifikanten Veränderung gekommen ist. Da bereits ab Ende 2017 getrennte Gebote durchgeführt werden konnten wurde eine Untersuchung des Einflusses der vorhandenen Daten zur Bestimmung des stündlichen Strompreises durchgeführt. Dies erfolgte durch Anwendung des Random Forest Verfahrens, welches auf Entscheidungsbäumen, einem stochastischen Verfahren, basiert.

Die Untersuchung mit dem Random Forest Verfahren für den Zeitraum vom 01.01.2018 bis zum 31.05.2020 und den Zeitraum vom 01.10.2018 bis zum 31.05.2020 ergeben sich die Bestimmtheitsmaße aus Tabelle 1.

Tabelle 1: Bestimmtheitsmaße für die verschiedenen Untersuchungszeiträume

	01.01.2018 – 31.05.2020	01.10.2018 – 31.05.2020
Ohne Residuallast	0,941	0,941
Mit Residuallast	0,946	0,950

Ebenfalls zeigt diese Tabelle den Einfluss der Residuallast auf stündliche Strompreise. Dazu wurde, anstatt den Stromverbrauch und die EE-Einspeisung getrennt als Eingabedaten zu verwenden, die Differenz von diesen gebildet, welches die Strommenge darstellt, welche noch durch konventionelle Kraftwerke an der Börse gedeckt werden muss. Es zeigt sich, dass beide Zeiträume eine ähnliche Bestimmtheit aufweisen. Da das Random Forest Verfahren ein stochastisches Verfahren darstellt, können bei mehrmaliger Ausführung dieses Verfahrens leicht veränderte Werte entstehen. Es zeigt sich jedoch eine leichter Trend, dass die Bildung der Residuallast die Bestimmtheit erhöht. Zusätzlich dazu wird nun anstatt drei getrennter

Merkmale nur ein einzelnes für jedes Land verwendet, was die Anzahl der Gesamtmerkmale reduziert. Als wichtigste Merkmale konnten in beiden Zeiträumen jeweils die EE-Einspeisungen, der Stromverbrauch in Deutschland (bzw. stattdessen die Residuallast) und der Gaspreis bestimmt werden. Da beide Zeiträume eine ähnliche Bestimmtheit aufweisen, wurde der Zeitraum vom 01.01.2018 gewählt, da dieser mehr historische Daten beinhaltet und gerade für KNN basierte Methoden die Verfügbarkeit von ausreichend Daten essenziell ist.

Als nächstes werden verschiedene Techniken untersucht, die stündlichen Eingabedaten derart umzuformen, dass eine Bestimmung des Base-Preises erfolgt. Hier werden drei verschiedene Techniken miteinander verglichen. Der einfachste Ansatz ist es, alle stündlichen Daten eines Tages als Eingabedaten zu verwenden. Dadurch steigt allerdings die Gesamtanzahl aller Merkmale von zuvor 30 (stündliche Merkmale und tageweise Konstante Merkmale) auf 234. Alternativ lassen sich die stündlichen Werte analog zu Peak (8 Uhr bis 20 Uhr), Off-Peak 1 (0 Uhr bis 8 Uhr) und Off-Peak 2 (20 Uhr bis 0 Uhr) zusammenfassen (48 Merkmale). Auch eine Zusammenfassung aller stündlichen Werte zu einem einzigen Wert pro Merkmal wurde untersucht, wodurch die Anzahl der Merkmale wieder bei 30 liegt. Die tageweisen konstanten Werte wie Kraftwerksverfügbarkeiten oder Brennstoffkosten werden unverändert den umgeformten stündlichen Daten hinzugefügt. Die Bestimmtheitswerte für diese drei Transformationsmethoden sind in Tabelle 2 gegeben.

Tabelle 2: Bestimmtheitsmaße für die drei Transformationsmethoden

Stündliche Umformung	Peak/Off-Peak	Tageweise Umformung
0,923	0,912	0,942

Es zeigt sich, dass die Umformung der stündlichen Werte zu einem einzigen Wert für einen Handelstag eine höhere Bestimmtheit liefert im Vergleich zu den anderen Transformationsmethoden. Zusätzlich dazu besitzt diese Umformung die geringste Merkmalsanzahl, was sich positiv auf die Trainingszeit auswirkt. Aus diesen Gründen wird im Folgenden die tageweise Umformung verwendet.

Trainingsergebnisse

In Tabelle 3 sind die Gewinne für den Trainings- und den Testdatensatz für verschiedene Strategien dargestellt. Für die Vergleichbarkeit der erlernten Handelsstrategien werden zwei Referenzstrategien definiert. In der Strategie „OTC-Kaufen“ bzw. „OTC-Verkaufen“ wird für jeden Handelstag ein Kaufgeschäft bzw. Verkaufsgeschäft im OTC-Markt platziert. Zur genauen Einordnung dieser Ergebnisse ist die Anzahl der Handelstage je Datensatz und der maximal erreichbare Gewinn angegeben.

Tabelle 3: Vergleich der Gewinne für die jeweiligen Datensätze für fixierte Strategien

	Trainingsdatensatz	Testdatensatz
OTC-Kaufen	-35,34€	22,86€
OTC-Verkaufen	11,34€	-28,57€
Maximal möglicher Gewinn	922,22€	248,44€
Anzahl Tage	702	175

Für die umgeformten Daten wurde jeweils ein Modell mit den RL-Verfahren PPO und DQN trainiert. In Abbildung 2 sind die Gewinne jeweils für den Trainingsdatensatz über die Anzahl

der Trainingsepochen abgebildet. Eine Trainingsepoche bezeichnet dabei eine komplette Iteration über den Trainingsdatensatz. Der maximal erreichbare Gewinn ist durch eine rote Linie dargestellt. Die Parameter dieser Modelle befinden sich im Standardzustand [16]. Es zeigt sich, dass PPO deutlich schneller einen höheren Trainingsgewinn erreicht als DQN. Ab etwa 10.000 Epochen steigt der Gewinn von PPO nicht weiter an. DQN zeigt bis zum Ende der untersuchten Epochenanzahl einen stetigen Anstieg. Somit könnte bei größerer Epochenanzahl ein vergleichbarer Trainingsgewinn wie für PPO erzielt werden. Jedoch weist DQN eine höhere Volatilität auf als PPO. Aus diesen Gründen wird nur für PPO eine Parametrierung durchgeführt.

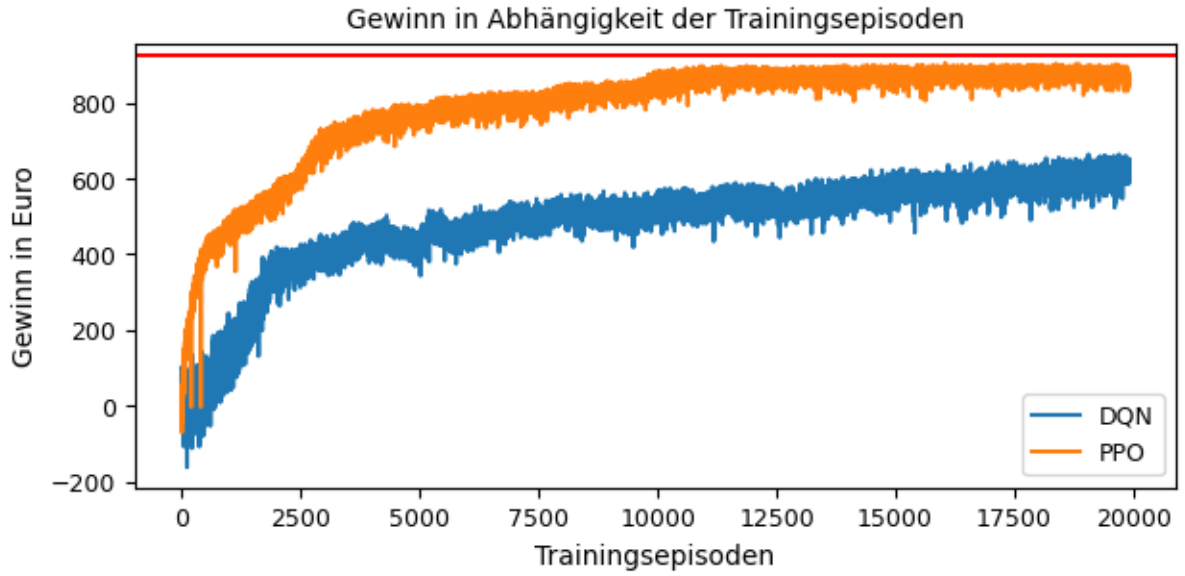

Abbildung 2: Verlauf des Trainingsgewinns in Abhängigkeit der Trainingsepisoden für die Verfahren DQN, PPO und A2C. Maximal möglicher Gewinn in rot

Parametrierung und Validierung

Auf Grundlage der bisherigen Ergebnisse erfolgte eine Parametrierung von PPO. In Abbildung 3 ist der Verlauf des Trainings- und Testgewinns in Abhängigkeit der Trainingsepisoden für eine der gefundenen Parameterkombinationen dargestellt. Es zeigt sich, dass im Vergleich zum vorherigen Modell der Trainingsgewinn deutlich schneller ansteigt. So konnte ein Trainingsgewinn von 700€ bereits nach etwa 400 Episoden erreicht werden, anstatt wie zuvor erst nach etwa 5000. Dies zeigt den Einfluss der Parametrierung auf den Trainingsprozess. Der Testgewinn weist ebenfalls positive Werte auf, besitzt aber zusätzlich eine hohe Volatilität. Für den untersuchten Testzeitraum konnte das Verfahren nach jeder Trainingsepoche einen positiven Gewinn erzielen. Im Vergleich mit der fixierten Handelsstrategie OTC-Kaufen, welche auf dem Testdatensatz 22,86€ erzielt, liefert das Verfahren in der Spitze deutlich höhere Werte. Der Gewinn über den Trainingsdatensatz bzw. den Testdatensatz bezeichnet dabei immer den kumulierten Gewinn aller Handelstage.

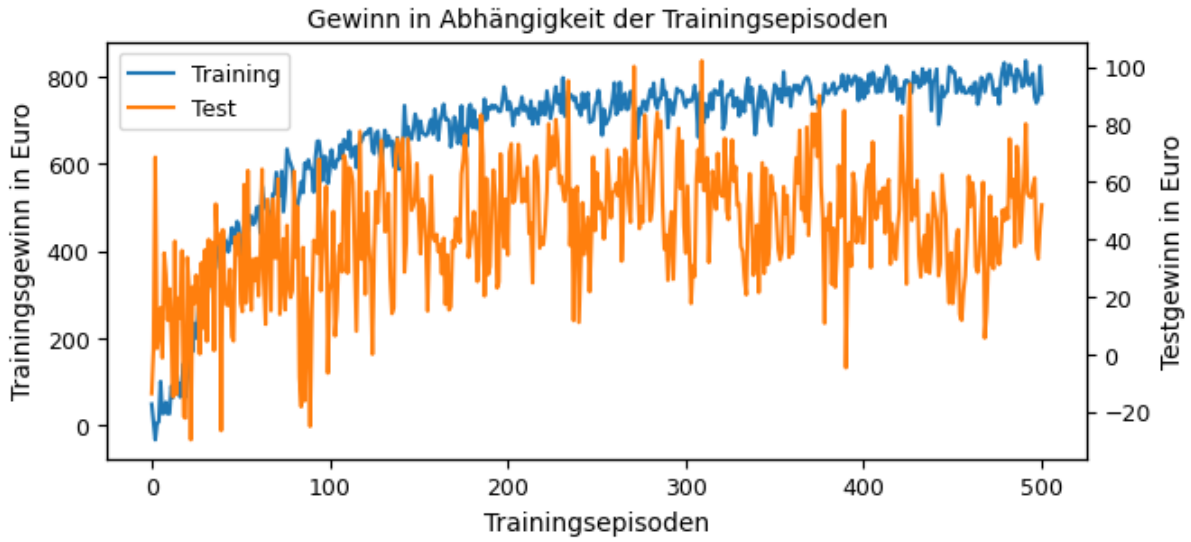

Abbildung 3: Verlauf des Gewinns für den Trainings- und Testdatensatz in Abhängigkeit der Episodenanzahl (Trainingsdurchläufe)

Im Vergleich zu den fixierten Strategien liefert dieses Verfahren bessere Ergebnisse. Als weiterer Vergleich wurde ein Random Forest Modell auf den gleichen Datensätzen trainiert. Dieses versucht, den Base-Preis zu prognostizieren. Liegt die Prognose des Base-Preises über dem OTC-Preis, kauft dieses Modell Strom im OTC-Markt ein. Analog erfolgt das Verkaufen des Stromes. Der Gewinn für einen Handelstag wird dann aus dem tatsächlichem Base-Preis und dem OTC-Preis, zu welchem Strom gekauft bzw. verkauft wurde bestimmt. Für den Testdatensatz konnte das Random Forest basierte Verfahren einen Gewinn in Höhe von 90,84€ erzielen und liegt damit im oberen Bereich der durch das DRL-Verfahren erreichten Werte. Der höchste mit dem DRL-Verfahren erreichte Gewinn beträgt 102,19€. Folglich eignet sich das Verfahren grundsätzlich zur Ableitung von Handelsstrategien in Elektrizitätsmärkten. Jedoch stellt die hohe Volatilität des Testgewinns ein Hindernis für die praktische Anwendbarkeit dar.

Das Framework zur Suche nach einer optimalen Parameterkombination speichert für jede untersuche Kombination den erreichten Gewinn ab. Somit kann der Einfluss der Parameter auf die Höhe des Rewards untersucht werden. Die untersuchten Parameter sind in Tabelle 4 gegeben. Hierbei wurde die Bezeichnung der Parameter von Stable Baselines 3 übernommen. Für eine genauere Erklärung der einzelnen Parameter wird auf die Dokumentation von Stable Baselines 3 verwiesen [16].

Als wichtigste Parameter wurde von Optuna *n_steps,* GAE λ und γ identifiziert. Von den Parametern des KNN stellt die Lernrate das wichtigste Merkmal dar. Insgesamt zeigt sich, dass die Parameter des RL-Verfahrens einen größeren Einfluss auf den Gewinn haben als die Parameter des KNN. Für letztere wurde jeweils der Suchraum auf Bereiche begrenzt, welche in verschiedenster Literatur zu KNN empfohlen werden, was einen Grund für die beobachtete Wichtigkeit der Parameter darstellen kann [16].

Tabelle 4: Aufzählung der von Optuna zu bestimmenden Parameter und deren Verwendung

	Parameter	Verwendung
Verfahrensspezifische Parameter	n_steps	Anzahl der Beobachtungen, nach denen ein Training erfolgt
	n_epochs	Anzahl der Optimierungsschritte nach Sammlung der Beobachtungen
	γ	Diskontierungsfaktor
	GAE λ	Beeinflusst Berechnung des Aktionsvorteils
	max_grad_norm	Maximaler Wert für das Gradient Clipping
	ϵ	Beeinflusst Gradient Clipping
	Koeffizient Entropie	Koeffizient zur Berechnung der Entropie
	Koeffizient Wertefunktion	Koeffizient zur Berechnung der Wertefunktion
Parameter des KNN	Netzgröße	Anzahl der Neuronen und Anzahl verdeckter Schichten
	Aktivierungsfunktion	Aktivierungsfunktion der Neuronen
	Lernrate	Beeinflusst Anpassung der Gewichte des KNN
	Batchgröße	Anzahl an Optimierungsschritten des RL-Verfahrens, bis Optimierung des KNN erfolgt

6 Zusammenfassung

Die zunehmende Dargebotsabhängigkeit der Stromerzeugung sowie neuartige Verbraucher sorgen für eine steigende Volatilität des Strompreises an der Strombörse. Als Instrument zum Ausgleich dieser Schwankungen gewinnt der kurzfristige Stromhandel an Bedeutung. Neben dem Stromhandel zur Bedarfsdeckung, lassen sich mit Handelsgeschäften die Preisdifferenzen zwischen dem OTC-Handel und der Strombörse ausnutzen. Aufgrund der vielfältigen Einflussfaktoren auf den Strompreis, stellt dies ein komplexes Optimierungsproblem dar und erfordert die Entwicklung von geeigneten Handelsstrategien. Aus diesem Grund wurde ein Verfahren basierend auf dem Deep Reinforcement Learning entwickelt, welches zur Ableitung von Handelsstrategien in Elektrizitätsmärkten dient.

Das entwickelte Verfahren liefert für den untersuchten Datensatz einen positiven kumulierten Gewinn über alle Handelstage und übertrifft die vorgestellten fixierten Strategien. Durch eine Suche nach einer optimalen Parameterkombination konnte die Anzahl der notwendigen Trainingsepisoden auf einen Bruchteil reduziert werden. Für den Testdatensatz können vereinzelt vergleichbare Gewinne wie für das Random Forest basierte Verfahren erreicht werden. Jedoch weist der Testgewinn eine hohe Sensitivität auf die Anzahl der durchgeführten Trainingsepisoden auf und bleiben im Mittel hinter dem Gewinn des Referenzmodells. Diese hohe Volatilität sollte Bestandteil weiterer Forschung sein. Der Vorteil des DRL-Verfahrens im Vergleich zum Random Forest basierten Verfahren besteht darin, dass das DRL-Verfahren derart modifiziert werden kann, dass auch mehrere Gebote im OTC-Markt möglich sind und

Preisdifferenzen an diesem Marktplatz ebenfalls zur Gewinnerzielung ausgenutzt werden könnten.

7 Literatur

[1] H.-W. Schiffer, *Energiemarkt Deutschland*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.

[2] I. Schumacher und P. Würfel, *Strategien zur Strombeschaffung in Unternehmen: Energieeinkauf optimieren, Kosten senken*. Wiesbaden: Springer Gabler, 2015.

[3] P. Konstantin, *Praxisbuch Energiewirtschaft*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[4] J. Borchert, R. Schemm und S. Korth, *Stromhandel: Institutionen, Marktmodelle, Pricing und Risikomanagement*. Stuttgart: Schäffer-Poeschel, 2006.

[5] F. Sensfuß, M. Ragwitz und M. Genoese, *The merit-order effect: A detailed analysis of the price effect of renewable electricity generation on spot market prices in Germany,* Working Paper Sustainability and Innovation. Karlsruhe: Fraunhofer ISI: Fraunhofer ISI. Verfügbar unter: http://hdl.handle.net/10419/28511.

[6] B. Aust und C. Morscher, „Negative Strompreise in Deutschland“, *Wirtschaftsdienst*, Jg. 97, Nr. 4, S. 304–306, 2017.

[7] E. Brainpool, *Zukünftige Auswirkungen der Sechs-Stunden Regelung gemäß § 24 EEG 2014: Kurzstudie im Auftrag des Bundesverbands WindEnergie e.V.* Verfügbar unter: https://www.energybrainpool.com/fileadmin/download/Studien/Studie_2014-12-11_BWE_Sechsstunden-Regelung_EnergyBrainpool.pdf. Zugriff am: 13. April 2021.

[8] J. M. Seifert, *Preismodellierung und Derivatebewertung im Strommarkt: Theorie und Empirie: Zugl.: Karlsruhe, Univ., Diss., 2009*. Karlsruhe: KIT Scientific Publ, 2010.

[9] R. S. Sutton und A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA and London: The MIT Press, 2018.

[10] S. J. Russell und P. Norvig, *Artificial intelligence: A modern approach*. Boston and Columbus and Indianapolis: Pearson, 2016.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford und O. Klimov, *Proximal Policy Optimization Algorithms.* Verfügbar unter: http://arxiv.org/pdf/1707.06347v2.

[12] V. Mnih *et al., Asynchronous Methods for Deep Reinforcement Learning.* Verfügbar unter: http://arxiv.org/pdf/1602.01783v2.

[13] R. Kruse *et al., Computational Intelligence: Eine methodische Einführung in künstliche neuronale Netze, evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze,* 2. Aufl. Wiesbaden: Springer Vieweg, 2015.

[14] U. Lämmel und J. Cleve, *Künstliche Intelligenz: Mit 51 Tabellen, 43 Beispielen, 118 Aufgaben, 89 Kontrollfragen und Referatsthemen,* 4. Aufl. München: Hanser, 2012.

[15] V. Mnih *et al.,* „Human-level control through deep reinforcement learning“, *Nature*, Jg. 518, Nr. 7540, S. 529–533, 2015.

[16] A. Raffin *et al.,* „Stable-Baselines3: Reliable Reinforcement Learning Implementations“, *Journal of Machine Learning Research*, Jg. 22, Nr. 268, S. 1–8, http://jmlr.org/papers/v22/20-1364.html, 2021.

[17] P. Stone, „Reinforcement Learning“ in *Encyclopedia of machine learning and data mining*, C. Sammut und G. I. Webb, Hg., New York, NY: Springer, 2017, S. 1088–1090.

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta und M. Koyama, „Optuna: A Next-generation Hyperparameter Optimization Framework“ in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.