Julien Pierre Georg Pédurand

Deep Reinforcement Learning für Portfolio Optimierung auf Krypto Märkten

Masterarbeit

am Institut für Ökonometrie und Wirtschaftsstatistik
(Westfälische Wilhelms-Universität, Münster)

Themensteller:	Prof. Dr. Mark Trede
Betreuer:	Dr. Mawuli Segnon
vorgelegt von:	Julien Pierre Georg Pédurand Horstmarer Landweg 84 48149 Münster +49 176 92692569 julien.pedurand@uni-muenster.de
Abgabetermin:	03.06.2022

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

A2C ..... Advantage-Actor Critic
BAH ..... Buy-and-Hold
BCM ..... Verhaltensklonmodul
CAAN ..... Cross-Asset-Attention-Network
CNN ..... Convolutional-Neural-Network
CRIX ..... Cryptocurrency Index
CRP ..... Constant-Rebalance-Portfolio
CVaR ..... Conditional-Value-at-Risk
DAM ..... adversarischen Datenerweiterungsmodul
DDPG ..... Deep Deterministic-Policy-Gradient
DJIA ..... Dow Jones Industrial Average Index
DL ..... Deep Learning
DQN ..... Deep-Q-Networks
DRL ..... Deep Reinforcement Learning
EAM ..... Evolving-Agent-Module
EG ..... Exponentiated-Gradients
EI3 ..... Ensemble of Identical Independent Inception
EIIE ..... Ensemble of Identical Independent Evaluators
fPW ..... finale Portfoliowert
GAE ..... Generalized-Advantage-Estimator
IPM ..... infundierten Vorhersagemodul
LSTM ..... Long-Short-Term-Memory
MAPS ..... Multi-Agenten-Portfoliomanagementsystem
MD ..... maximale Diversifikation
MDD ..... Maximum-Drawdown
MDP ..... Markov-Entscheidungsprozesses
MLP ..... Multi-Layer-Perceptrons
MVO ..... Mean-Variance-Optimierung
NN ..... neuronaler Netze
OHLC ..... Open High Low Close
PAMR ..... Passive Aggressive Mean Reversion
PG ..... Policy Gradient
PM ..... Portfoliomanagement
PPO ..... Proximal Policy Optimization
RAT ..... Relation-Aware-Transformer
ReLu ..... Rectified-Linear-Unit
RL ..... Reinforcement Learning
RNN ..... Recurrent-Neural-Network
SAM ..... Strategic Agent Module
SARL ..... State-Augmented-RL
t-LSTM ..... zeitlich bewusstes LSTM
TRPO ..... Trust-Region-Policy-Optimization
Value-at-Risk ..... Value-at-Risk

1 Einleitung

Portfoliomanagement (PM) ist ein kontinuierlicher Prozess der Umschichtung von Kapital in mehrere Vermögenswerte (Markowitz 1952) und zielt darauf ab, die kumulierten Gewinne zu maximieren – mit der Option, die Gesamtrisiken des Portfolios zu minimieren (Huang und Tanaka 2022). Betancourt und Chen (2021) argumentieren zudem, dass die Erzielung finanzieller Gewinne durch den Handel mit Kryptowährungen – aufgrund ihrer unberechenbaren Preisschwankungen – eine Herausforderung darstellt. Zudem argumentieren Corbet et al. (2017), dass die Preise von Kryptowährungen eine höhere Volatilität aufweisen, als die traditioneller Vermögenswerte. Fang et al. (2022) argumentieren das Kryptowährungen eine breite Marktakzeptanz und eine schnelle Entwicklung erfahren haben, entgegen ihrer kürzlichen Konzeption. Zudem legitimieren sie die Relevanz der Kryptowährungen dadurch, dass Hedgefonds und Vermögensverwalter mit Kryptowährungen verbundene Vermögenswerte in ihre Portfolios und Handelsstrategien aufgenommen haben. Petukhina et al. (2021) merken an, dass Kryptowährungen seit der Erfindung der Blockchain durch Nakamoto (2008) eine bemerkenswerte Wertentwicklung aufgewiesen haben. Zudem argumentieren sie, dass infolge der enormen Kapitalzuflüsse in den Markt und der starken Preisschwankungen Kryptowährungen stark an Marktwert gewonnen haben. Jiang und Liang (2016) argumentieren innerhalb ihrer Untersuchung, dass Kryptowährungen sich anhand von zwei Eigenschaften gegenüber traditionellen Finanzanlagen unterscheiden: Dezentralisierung und Offenheit. Demnach erläutern sie, dass ohne eine zentrale Regulierungsinstanz jeder mit geringen Zugangsvoraussetzungen am Kryptowährungshandel teilnehmen kann. Zudem kommen sie aufgrund der Datenverfügbarkeit der Handelsplattformen und deren Eigenschaft, unbegrenztes Handeln zu ermöglichen, zu der Schlussfolgerung, dass der Kryptomarkt ideal für Experimente mit maschinellem Lernen im Portfoliomanagement ist.

In den letzten Jahren hat sich das Deep Reinforcement Learning (DRL) als Kombination von Deep Learning (DL) und Reinforcement Learning (RL) etabliert. Durch die Verwendung neuronaler Netze (NN) ist ein DRL-basierter Agent besonders fähig, nützliche Informationen aus hochdimensionalen Daten zu extrahieren und sequenzielle Aktionen auf der Grundlage von Belohnungen durchzuführen. DRL-Methoden haben zu vielen Durchbrüchen in verschiedenen Bereichen geführt (Huang und Tanaka 2022). Kürzlich erreichte Meilensteine von RL-Algorithmen sind insbesondere: DeepMinds AlphaZero, welcher in der Lage ist, den Weltmeister im Spiel GO zu besiegen und die bestehenden Schach-Engines bei weitem zu übertreffen (Silver et al. 2018). Einen Entwicklungsschritt weiter ging OpenAI mit der Einführung eines Multiplayer-Agenten

namens FIVE, der darauf trainiert wurde, gegen menschliche Teams im Multiplayer-Spiel DOTA2 zu spielen (OpenAI et al. 2019).

In den vergangenen Jahren hat das Interesse an der Anwendung von Deep Reinforcement Learning (DRL) zur Lösung komplexer dynamischer Entscheidungsprobleme zugenommen. Eine repräsentative Klasse von Problemen ist das Portfoliomanagement, dessen Formulierung typischerweise eine große Menge an kontinuierlichen Zustands- und Aktionsvariablen und eine besondere Form der Risikofunktion erfordert, um die inhärente Komplexität von Finanzmärkten, Handelsumgebungen und Anlegerpräferenzen zu erfassen (Marzban et al. 2021).

Innerhalb dieser Arbeit wird ermittelt, ob Deep Reinforcement Learning für das Portfolio Management auf dem Kryptowährungsmarkt erfolgreich verwendet werden kann. Demnach wird ein Actor-Critik-Netzwerk unter Verwendung des On-Policy-Reinforcement-Learning-Algorithmus Proximal Policy Optimization (PPO) trainiert. Um die Leistungsfähigkeit des Agenten im Rahmen des Portfoliomanagements zu veranschaulichen, werden drei unterschiedliche Rewardfunktionen genutzt. Zum einen wird die Portfoliorendite als Zielfunktion für den Agenten verwendet, zum anderen werden die Sharpe Ratio sowie die Sortino Ratio in die Untersuchung einbezogen.

Die Untersuchung gründet dabei auf den täglichen Preisdaten von 50 Kryptowährungen, welche auf der Webseite Coinmarketcap veröffentlicht und anhand eines Webscraping Prozesses akkumuliert wurden. Der dadurch erstellte Datensatz umfasst einen Zeitraum von über 3 Jahren, welcher am 01.01.2019 beginnt und am 31.03.2022 endet. Dieser erstellte Datensatz wird anschließend in ein Trainingsdatensatz vom 01.01.2019 bis zum 31.12.2020 und in einen Testdatensatz vom 01.01.2021 bis zum 31.02.2022 aufgeteilt.

Die in dieser Arbeit konzipierten Agenten werden in einem ersten Schritt anhand des Trainingsdatensatzes trainiert und anschließend wird die Leistungsfähigkeit der Agenten anhand des Testdatensatzes unter Verwendung einer Auswahl an unterschiedlichen Leistungsmetriken evaluiert. Zusätzlich werden verschiedene Portfolio Strategien in die Untersuchung mitaufgenommen, um ebenfalls die die Performance der drei entwickelten Agenten zu vergleichen und zu beurteilen.

2 Literaturrückblick

Im Folgenden Abschnitt dieser Arbeit werden eine Auswahl an Untersuchungen, die sich mit dem Thema des DRL für die Portfoliooptimierung auseinandersetzen, sowie die daraus resultierenden Ergebnisse vorgestellt. Eine Arbeit die sich mit Kryptomärkten auseinandersetz, ist die von Jiang und Liang (2016) vorgestellte Untersuchung. Diese wenden in ihrer Arbeit ein Convolutional-Neural-Network (CNN) zur Portfoliooptimierung auf Kryptomärkten an. Zudem merken sie an, dass sie einen vollständigen Ansatz des maschinellen Lernens auf das allgemeine Problem des Portfoliomanagements anwenden und somit keine Vorkenntnisse über die Finanzmärkte voraussetzen. Demnach erfassen die von ihnen konzipierten Algorithmen ausschließlich den Marktverlauf und lernen aus diesen. Für ihre Untersuchung betrachten sie eine Auswahl von 12 Kryptowährungen in einem Zeitraum von einem Jahr mit einer Frequenz von 30 Minuten. Das Training des Netzwerkes erfolgt anhand eines Deterministic Policy Gradientens und maximiert den Gesamtertrag der getätigten Transaktionen, welcher als die Belohnungsfunktion des neuronalen Netzes definiert wird. Jiang und Liang merken zudem an, dass sich ihre Verwendung von CNNs und Policy-Gradient-Netze zur vorherrschenden Verwendungsweise unterscheidet. Demnach argumentieren sie, dass CNNs im Finanzbereich primär zur Vorhersage von Preisänderungen eingesetzt werden, so dass das Netzwerk prognostizierter Preise ausgibt. In Bezug auf Policy-Gradient-Netze erläutern Sie, dass diese die Wahrscheinlichkeit jeder Aktion ausgeben und sich die Aktionen auf diskrete Fälle beschränken. Die von ihnen gewählte Verwendungsweise unterscheidet sich demnach zu den beiden konventionellen Ansätzen dadurch, dass ihr Netzwerk direkt die Portfoliogewichte ausgibt.

Die Performance der CNN-Strategie wird mit drei Benchmarks und drei anderen Portfoliomanagement-Algorithmen in einem Zeitraum vom 14.05.2016 bis zum 07.03.2016 verglichen. Die Ergebnisse zeigen, dass die Leistung des von Jiang und Liang konzipierten Agenten die meisten der ausgewählten Algorithmen sowie die Benchmarks übertrifft. Lediglich die Passive Aggressive Mean Reversion (PAMR) konnte eine höhere Rendite im betrachteten Zeitraum erzielen. Allerdings ist hierbei zu berücksichtigen, dass der CNN-Händler ein deutlich geringeres Risiko und damit eine höhere Sharpe Ratio als PAMR aufweist.

In einer weiteren Arbeit konzipieren Jiang et al. (2017) ein Reinforcement Learning System, welches speziell für die Aufgabe des Portfoliomanagements konzipiert ist. Eine zentrale Rolle innerhalb dieses Systems nimmt das von ihnen entwickelte Ensemble of Identical Independent Evaluators (EIIE) ein. Demnach ist EIIE ein neuronales Netzwerk, dessen Funktion es ist, das potenzielle Wachstum eines Vermögenswertes in der

unmittelbaren Zukunft anhand dessen bisheriger Wertentwicklung zu untersuchen und zu bewerten. Um die Auswirkungen der getätigten Portfolioallokationen – und die damit verursachten Transaktionskosten auf das Gesamtvermögen – zu berücksichtigen, werden die Portfoliogewichte der vorherigen Handelsperiode in das EIIE einbezogen. Jiang et al.

argumentieren, dass dies dazu führt, dass der RL-Agent die Transaktionskosten minimieren kann, indem zu große Änderungen von aufeinanderfolgenden Portfoliogewichten vermieden werden.

Die von ihnen gewählten technischen Implementierungen werden anhand von drei unterschiedlichen Netzwerkarchitekturen – CNN, Recurrent-Neural-Network (RNN) und Long-Short-Term-Memory (LSTM) – getestet welche alle anhand eines Deterministic Policy Gradienten trainiert werden. Zusätzlich werden die Ergebnisse mit verschiedenen, bereits etablierten Portfoliomanagementstrategien verglichen. Als Datengrundlage für ihre Experimente selektierten sie die 11 Kryptowährungen mit der größten Marktkapitalisierung auf der Handelsplattform Poloniex aus den letzten 30 Tagen in einer Frequenz von 30 Minuten.

Aufgrund ihrer Ergebnisse, welche auf drei unterschiedlichen Zeiträumen beruhen, kommen Jiang et al. zu der Schlussfolgerung, dass die von ihnen konzipierten RL-Agenten die traditionellen Portfoliomanagement-Methoden übertreffen. Des Weiteren merken sie an, dass die drei verschiedenen Netzwerkarchitekturen in allen drei Zeiträumen die traditionellen Portfoliomanagement-Methoden, gemessen am endgültigen Portfoliowert und an den risikobereinigten Portfolio-Metriken, übersteigen.

Shi et al. (2019) greifen die von Jiang et al. (2017) vorgestellte EIIE-Netzwerkstruktur auf und erweitern sie um eine mehrskalige Merkmalsextraktion, welche sie als Ensemble of Identical Independent Inception (EI3) bezeichnen. Das vorgeschlagene EI3-Netzwerk extrahiert zunächst über mehrere Verzweigungen mehrskalige zeitliche Merkmale, welche anschließend mit den vorherigen Portfoliogewichten aggregiert und normalisiert werden, um die neu ausbalancierten Portfoliogewichte zu erhalten. Sie argumentieren, dass das von ihnen vorgestellte Netzwerk in der Lage ist, zeitliche Muster in der Preisbewegung auf verschiedenen Ebenen zu erfassen, während die Vermögenswerte unabhängig bleiben.

Um die Leistungsfähigkeit der von ihnen vorgestellten Architektur zu verifizieren und zu vergleichen, werden die unterschiedlichen Netzwerkimplementierungen der EIIE-Architektur sowie verschiedene traditionelle Portfoliomanagement-Methoden von Jiang et al. (2017) in die Untersuchung einbezogen. Demnach orientieren sich Shi et al. weitgehend an dem von Jiang et al. etablierten Versuchsaufbau und betrachten in ihrer

Untersuchung die Preise von 11 Kryptowährungen, mit einer Frequenz von 30 Minuten, mit der höchsten Marktkapitalisierung der letzten 30 Tage, relativ zum ersten Tag des Evaluierungszeitraumes. Dabei wird das EI3-Netzwerk anhand eines Recurrent Reinforcement Learning Algorithmus trainiert. Als Zielfunktion fungiert dabei die logarithmierte Rendite des Portfolios der jeweiligen Handelsperiode. Die Ergebnisse ihrer Untersuchung zeigen, dass das vorgestellte EI3-Netzwerk die unterschiedlichen EIIE-Netzwerke und die klassischen Portfoliomanagementmethoden übertreffen. Aufgrund ihrer Ergebnisse kommen Shi et al. zu der Schlussfolgerung, dass es keine dominante Strategie gibt, die sich in allen Marktsituationen durchsetzen kann. Weiter merken sie an, dass EI3 in Aufschwung- und regulären Marktphasen den höchsten Gewinn erzielt, während es in einem sinkenden Markt Verluste reduzieren kann. Dies führen sie darauf zurück, dass die in EI3 extrahierten mehrstufigen zeitlichen Informationen die Erfassung von Kauf- und Verkaufsmöglichkeiten verbessern.

Ye et al. (2020) stellen innerhalb ihrer Arbeit ein State-Augmented-RL (SARL) -Modell für das Portfoliomanagement vor. Es ermöglicht die Nutzung zusätzlicher Informationen aus anderen Quellen, um das Prognostizieren von Markttrends zu verbessern. Dies ermöglicht das Einbeziehen von Finanznachrichten, die sich auf die ausgewählte Vermögenswerte beziehen, da diese neuen Informationen für die Vorhersage von Vermögensbewegungen liefern. Unter Verwendung verschiedener Methoden der natürlichen Sprachverarbeitung werden die Nachrichten in ein hierarchisches Aufmerksamkeitsnetz eingefügt, um einen binären Klassifikator zu trainieren, der die Preisbewegungen prognostiziert. Um das Netzwerk zu trainieren, wird ein Deterministic-Policy-Gradient angewandt, welcher die Zielfunktion der logarithmierten Portfoliorenditen maximiert. Zur Ermittlung der Leistung ihres Modells betrachten Ye et al. den Kryptomarkt sowie den amerikanischen Aktienmarkt. Für einen Zeitraum vom 30.06.2015 bis zum 30.06.2017 werden analog zu Jiang et al. (2017) 10 Kryptowährungen, in einer Preisfrequenz von 30 Minuten, in die Untersuchung einbezogen. Auf dem Aktienmarkt wird eine Auswahl von 9 Technologieaktien in einem Zeitraum vom 20.10.2006 bis zum 20.11.2013 betrachtet. Dabei werden sowohl tägliche Schlusskurse der betrachteten Vermögenswerte als auch Finanznachrichten, die diese betreffen, als Datengrundlage verwendet.

Die Ergebnisse ihrer Untersuchung zeigen, dass SARL einen deutlich bessere Portfoliowerte und Sharpe-Ratios erzielt als die zum Vergleich einbezogenen klassischen Portfoliomanagementmethoden. Aufgrund der von ihnen durchgeführten Experimente schlussfolgern Ye et al., dass die Nutzung diverser Informationen – wie Finanznachrichten – die Auswirkungen der Umweltunsicherheit reduzieren. Zudem

kommen sie zu der Erkenntnis, dass die Berücksichtigung von Finanznachrichten die Leistung des Portfoliomanagement-Agenten verbessert wird.

Um sowohl sequenzielle Muster als auch die inneren Korrelationen zwischen finanziellen Vermögenswerten zu erfassen, entwickeln Xu et al. (2020) im Rahmen des klassischen Recurrent-RL-Verfahrens einen Relation-Aware-Transformer (RAT) – eine Weiterentwicklung des klassischen Transformers. Konkret folgt RAT einer Encoder-Decoder-Struktur, wobei der Encoder für die sequenzielle Merkmalsextraktion und der Decoder für die Entscheidungsfindung zuständig ist. Experimente mit zwei Kryptowährungs- und einem Aktiendatensatz zeigen, dass die von Xu et al. konkretisierte Erweiterung zum einen die verbesserte Leistung von RAT gegenüber betrachteten Portfoliomethoden zeigen. Zum anderen, dass RAT effektiv eine bessere Repräsentation erlernt. Basierend auf ihren Ergebnissen kommen Xu et al. zu der Schlussfolgerung, dass der Aufmerksamkeitsmechanismus der Transformer-Architektur das vorherrschende Verfahren im Bereich des Portfoliomanagements darstellt.

Alessandretti et al. (2018) untersuchten die Leistungsfähigkeit dreier Prognosemodelle für die täglichen Preise von Kryptowährungen aus dem Zeitraum zwischen dem 11.11.2015 und dem 24.04.2018. Die Daten wurden der Website Coin Market Cap entnommen, die tägliche Daten von 300 Börsenplattformen sammelt. Aus den insgesamt 1681 Kryptowährungen wurden jeweils nur Währungen in die Analyse aufgenommen, die ein tägliches Handelsvolumen von mehr als 100.000 USD aufweisen können.

Daraufhin erstellten sie Anlageportfolios auf der Grundlage der Vorhersagen der drei verschiedenen Methoden und verglichen ihre Leistung mit der einer Basislinie, die durch die Strategie des einfachen gleitenden Durchschnitts repräsentiert wird. Zwei der Modelle basieren auf Gradient-Boosting-Entscheidungsbäumen und eines auf rekurrenten neuronalen Netzen mit LSTM. Ihre Ergebnisse zeigen, dass alle Strategien über den gesamten betrachteten Zeitraum – und für eine große Anzahl kürzerer Handelszeiträume, die sich aus verschiedenen Kombinationen von Start- und Enddaten für die Handelsaktivität zusammensetzen – Gewinne erzielen konnten. Zudem zeigen ihre Ergebnisse, dass über den gesamten betrachteten Zeitraum die drei Methoden eine bessere Performance als die zum Vergleich einbezogene Basisstrategie aufweisen konnten. Eine weitere Erkenntnis, die sie aus ihren Ergebnissen ziehen, ist, dass die auf Gradient-Boosting-Entscheidungsbäumen basierenden Methoden eine erhöhte Leistungsfähigkeit aufweisen, wenn die Vorhersagen auf kurzfristigen Fenstern von 5–10 Tagen beruhen. Daraus schlussfolgern sie, dass diese Methoden vor allem kurzfristige Abhängigkeiten analysieren und ausnutzen können. In Bezug auf die Leistungsfähigkeit von rekurrenten neuronalen LSTM-Netzen argumentieren sie, dass diese eine erhöhte Performance

aufweisen, wenn die Vorhersagen auf Daten von 50 Tagen beruhen. Dies führen sie auf die Eigenschaft der LSTM-Netzwerke zurück, langfristige Abhängigkeiten erfassen zu können, was dazu führt, sehr stabil gegenüber Preisschwankungen zu agieren. Zudem konnten sie beobachten, dass ausschließlich das eingesetzte LSTM-Netzwerk bei Transaktionsgebühren von bis zu 1 % einen Gewinn erzielen konnte. Demnach schlussfolgern sie, dass die auf rekurrenten neuronalen LSTM-Netzen basierende Methode systematisch die beste Rendite erzielt.

Lucarelli und Borrotti (2020) erweitern einen von ihnen vorgestellten Ansatz eines Handelssystems, welches auf DRL beruht, zum Handeln von Bitcoin. Dabei orientieren sie sich an der von Patel (2018) vorgestellten Methode des Einsatzes eines Multiagenten, der eine dynamische Verwaltung des Kryptowährungsportfolios durchführt. Mit der dynamischen Verwaltung beabsichtigen sie eine Verwaltung, die in der Lage ist, die beste Reihe von Aktionen zu jedem Handelszeitpunkt mit dem Endziel der Maximierung der Portfoliorendite zu definieren. Überdies weicht der von ihnen verfolgte Ansatz von dem von Patel gewählten Ansatz ab. In dem von ihnen ausgearbeiteten Ansatz wurde eine Gruppe lokaler Agenten gebildet, jeweils ein Agent für jeden Vermögenswert des Portfolios. In jedem lokalen Agenten konkurriert eine spezialisierte Deep-Q-Learning-Technik bei der Definition einer globalen Belohnungsfunktion, die zur dynamischen Aktualisierung der Informationen jedes lokalen Agenten verwendet wird. Die globale Belohnungsfunktion wird von einem globalen Agenten verwaltet. Daher werden die Aktionen gewichtet und auf jeden Vermögenswert im Portfolio angewendet. Demgegenüber betrachtet Patel zwei RL-Agenten, einen für Tickdaten und einen für Orderbuchdaten, für die Verwaltung von nur einem Vermögenswert.

Für ihre Untersuchung wählen Lucarelli und Borrotti drei verschiedene Deep RL-Ansätze für die lokalen Agenten aus: Deep-Q-Networks (DQN), Double-Deep-Q-Networks und Dueling-Double-Deep-Q-Networks. Jeder lokale Agent wurde dann in Kombination mit zwei globalen Belohnungsfunktionen bewertet: der Summe der nominalen Nettorenditen sowie einer linearen Kombination aus Sharpe-Ratio und Portfolio-Nettorendite. Anschließend wurde das von ihnen vorgeschlagene Q-Learning-Portfoliomanagementframework an einem Kryptowährungsportfolio getestet, das aus vier Kryptowährungen besteht, Bitcoin, Litecoin , Ethereum und Riple. Für die Experimente wurde ein Zeitraum vom 01.07.2017 bis zum 25.12.2018 betrachtet.

Ihre Ergebnisse zeigen, dass alle Modelle positive durchschnittliche Tagesrenditen für eine Reihe kürzerer Handelszeiträume – bestehend aus verschiedenen Kombinationen von Start- und Enddaten für die Trading-Aktivitäten – aufwiesen. Das am besten performende Modell in ihrer Untersuchung – in Bezug auf die durchschnittlichen

täglichen Renditen – ist eine Kombination aus einem DQN mit einer Belohnungsfunktion, die auf der Sharpe-Ratio beruht. In einem weiteren Schritt wurde das beste Modell mit zwei alternativen Ansätzen verglichen. Zum einen der gleichgewichteten Portfoliotechnik und zum anderen mit einer auf der genetischen Algorithmus-Optimierung basierenden Portfolioverwaltungstechnik. Die Ergebnisse zeigen, dass das von Lucarelli und Borrotti konzipierte Modell in Bezug auf die Sharpe-Ratio die beiden anderen Modelle übertrifft. Dabei merken sie jedoch an, dass keines der drei betrachteten Modelle bemerkenswerte Ergebnisse erzielt.

Betancourt und Chen (2021) stellen in ihrer Arbeit eine neuartige Portfoliomanagement-Methode vor, welche DRL auf Märkten mit einer dynamischen Anzahl von Vermögenswerten verwendet. Sie betonen dabei, dass dieses Problem besonders wichtig bei Kryptowährungsmärkten ist, da diese bereits den Handel mit Hunderten von Vermögenswerten unterstützen – wobei sich Anzahl der Währungen monatlich erhöhen. Demnach wird von ihnen eine neuartige neuronale Netzwerkarchitektur vorgeschlagen, die mithilfe des PPO-Algorithmus trainiert wird. Diese verfügt über eine Zielfunktion, die entweder die Renditen oder die Sharp-Ratio maximiert. Diese Architektur berücksichtigt alle Vermögenswerte auf dem Markt und ist während der Anwendung in der Lage, neu eingeführte Vermögenswerte in den Prozess zu integrieren. Durch das Einführen dieser Fähigkeit argumentieren Betancourt und Chen, dass ihre Methode allgemeiner und stichprobenartig effizienter ist als frühere Methoden.

Die in ihrer Arbeit verwendeten Daten entsprechen der Marktgeschichte vom 17.08.2017 bis zum 01.11.2019 auf der Kryptohandelsbörse Binance. Während des ausgewählten Zeitraums stieg die Anzahl der aktiven Vermögenswerte, die für die Untersuchung in Betracht gezogen werden, von 3 auf 85. Für das Experiment wurden drei Erfassungszeiträume der Daten ausgewählt: 30 Minuten, sechs Stunden und ein Tag. Für jeden Vermögenswert stehen in jedem Stichprobenzeitraum sechs Merkmale zur Verfügung, darunter die vier Preismerkmale Eröffnungs-, Höchst-, Tiefst- und Schlusskurs sowie die beiden Volumenmerkmale Standard- und Quotierungsvolumen. Der Testdatensatz entspricht einer 11-monatigen Historie von Vermögenswerten, was etwa 40 % des Gesamtdatensatzes entspricht.

Um die Ergebnisse ihrer Experimente zu validieren, wurde die von ihnen vorgeschlagene Methode anhand von drei RL-Methoden verglichen, die von Bu und Cho (2018) , Jiang et al. (2017) und Pendharkar und Cusatis (2018) konzipiert wurden. Die betrachteten Methoden werden in drei verschiedenen Setups getestet, die Episoden mit einer Länge von 1 Tag, 30 Tagen und 16 Wochen umfassen und mit Halteperioden von 30 Minuten, 1 Tag und 1 Woche versehen sind. Die Leistung der Methoden wurde anhand von zwei

Standardkennzahlen für Investitionen und Handel bewertet: der Gesamtrendite und der Sharpe-Ratio.

Aufgrund der von ihnen ermittelten Ergebnisse kommen Betancourt und Chen zu der Schlussfolgerung, dass sich die von ihnen konzipierte Methode allgemein besser an Veränderungen der Haltedauer und der Länge der Handelssitzungen anpasste als die zum Vergleich einbezogenen Baselines.

Neben Arbeiten, die sich auf den Kryptowährungsmarkt fokussieren, existiert ein weiterer Zweig in der Literatur des DRL für das Portfoliomanagement, der sich auf Aktienmärkte als Untersuchungsgrundlage fokussiert. Eine Untersuchung die sich auf den Aktienmarkt fokussiert, ist die von Wang et al. (2019) ihr Modell anhand von monatlichen Vermögenspreisen auf dem amerikanischen Aktienmarkt trainieren. Demnach entwerfen sie eine auf RL basierende Anlagestrategie, die um interpretierbare tiefe Aufmerksamkeitsnetze erweitert wird, um eine Interpretation der Entscheidungsabläufe zu ermöglichen. AlphaStock beruht im Wesentlichen auf einer Strategie für Aktienwerte zum Kauf von Gewinnern und zum Verkauf von Verlierern (Buying Winners selling Loosers - BWSL), welche aus drei Komponenten zusammengesetzt ist. Die erste dieser drei Komponenten ist ein LSTM mit Historischem Zustands Attention-Netzwerk, das zur Extraktion von Vermögenswertdarstellungen aus mehreren Zeitreihen verwendet wird. Die zweite Komponente ist ein Cross-Asset-Attention-Network (CAAN), welches eingesetzt wird um die Wechselbeziehungen zwischen den Vermögenswerten sowie deren Anstieg im Vorfeld umfassend zu modellieren. Die dritte Komponente ist ein Portfoliogenerator, der den Investitionsanteil jedes Vermögenswerts anhand der durch das CAAN generierten Werte bestimmt. Anschließend wird der Agent mittels eines Policy-Gradienten Verfahrens optimiert, um eine Maximierung der Sharpe-Ratio zu gewährleisten.

Wang et al. trainieren ihr Modell anhand von monatlichen Vermögenspreisen auf dem amerikanischen Aktienmarkt in einem Zeitraum von Anfang 1970 bis Anfang 1990. Zur Evaluation des Modls wählen sie den Zeitraum von 1970 bis 2016. Um die Robustheit des Modells zu überprüfen, wird das Modell in einem weiteren Schritt für den gleichen Testzeitraum auf dem chinesischen Aktienmarkt evaluiert. Aufgrund ihrer Ergebnisse auf den US-Aktienmärkten kommen sie zu der Schlussfolgerung, dass das von ihnen konzipierte AlphaStock-Modell – gemessen an einer Vielzahl von Bewertungsmaßstäben – besser abschneidet als einige der zum Vergleich einbezogenen Methoden. Durch die ebenfalls positiven Ergebnisse auf dem chinesischen Aktienmarkt leiten sie ab, dass ihr Modell robuste Leistungsfähigkeit aufweist. Abschließend merken sie an, dass

AlphaStock den Kauf von unterbewerteten Aktien mit hohem langfristigen Wachstum, geringer Volatilität sowie einem hohen intrinsischen Wert präferiert.

Ding et al. (2018) stellen innerhalb ihrer Arbeit das Investor-Imitator-Modell vor, welches anhand historischer Handelsaufzeichnungen das Verhalten verschiedener Arten von Investoren mit Hilfe einer Reihe von logischen Beschreibungskomponenten imitieren. Das Investor-Imitator-Modell verfolgt dabei das Ziel, drei Arten von Investoren (Orakel-Investor, Kollaborations-Investor, öffentlicher Investor) zu reproduzieren, indem jeweils eine investorenspezifische Belohnungsfunktion entworfen wird. Um die Leistungsfähigkeit ihres Modells zu ermitteln, selektieren Ding et al. eine Auswahl von 167 Aktien, welche aus dem chinesischen Aktienindex HS300 stammen. Dabei verwenden sie einen Zeitraum von Anfang 2005 bis Ende. Um die Leistungsfähigkeit des Investor-Imitators zu erfassen, werden die darauffolgenden drei Jahre 2014–2016 separat analysiert. In den Experimenten auf dem chinesischen Aktienmarkt extrahiert Investor-Imitator erfolgreich interpretierbares Wissen über Portfoliomanagement, welches menschlichen Händlern helfen kann, den Finanzmarkt besser zu verstehen.

Um die Lücke zwischen dem traditionellen Markowitz-Portfolio und RL-gestützten Methoden zu schließen und dadurch eine Absicherungsstrategie für einen riskanten Vermögenswert zu entwickeln, wenden Benhamou et al. (2020a) einen Adversial Policy Gradient mit einer verzögerten Belohnungsfunktion an. Neben den täglichen Renditen werden ebenfalls die Standardabweichungen der vergangenen Renditen in das Modell einbezogen. Dies begründen sie dadurch, dass Standardabweichungen der Renditen als nützliche Merkmale zur Erkennung von Krisen fungieren. Zusätzlich werden Informationen in das Modell eingespeist, die in keinem direkten Zusammenhang zum Portfolio stehen. Dabei treffen Benhamou et al. die Annahme, dass diese eine Vorhersagekraft für die Vermögenswerte des Portfolios besitzen und demnach einen positiven Einfluss auf die Leistungsfähigkeit des Agenten besiaufweisentzen. Das von ihnen vorgestellte Model wurde mit den täglichen Renditen des MSCI-World in einem Zeitraum von Anfang Mai 2000 bis Mitte 2020 trainiert und evaluiert. Zudem werden die Renditen von 4 weiteren Hedging-Strategien einbezogen, die tendenziell eine stärkere Performance in fallenden Aktienmärkten aufweisen.

Insgesamt können sie anhand ihrer Ergebnisse feststellen, dass das von ihnen konzipierte Modell – verglichen mit den traditionellen Methoden – eine verbesserte und robustere Leistungsfähigkeit besitzt.

Wang et al. (2021) schlagen für die Portfolioverwaltung ein DRL-basiertes Modell namens DeepTrader vor. DeepTrader baut dabei hauptsächlich auf zwei Einheiten auf, um vermögensübergreifende Beziehungen bzw. den Risiko-Ertrags-Ausgleichs anhand

eines PG zu erlernen. Dabei ist die Asset-Scoring-Einheit so aufgebaut, dass sie anhand der eingespeisten Daten einen ‚Gewinner-Score' für einen individuellen Vermögenswert ausgibt. Dieser soll die Einschätzung des Modells gegenüber der zukünftigen Preisentwicklung eines Vermögenswertes darstellen. Anschließend werden in einem weiteren Schritt die Gewinner-Scores in einem Graphen zusammengefasst, um die kurzen und langfristigen Zusammenhänge zwischen allen Aktien besser zu verarbeiten und hierarchisch zu erfassen. Die zweite Einheit, die sogenannte Market-Scoring-Unit, nutzt die Marktstimmungsindikatoren als Input und bettet dann die finanzielle Situation als Indikator ein, um die gewählte Position in jeder Handelsperiode anzupassen. Diese dynamische Anpassung ermöglicht es dem Modell, das durch komplizierte Marktschwankungen verursachte Risiko sowie Abwärtsbewegungen des Marktes rechtzeitig und effektiv zu reduzieren. Während die Kurssteigerungsrate als Belohnungsfunktion in der Asset-Scoring-Einheit verwendet wird, wird der Negativ-Maximum-Drawdown-Indikator als Belohnungsfunktion in der Market-Scoring-Unit angewandt. Wang et al. führten Experimente mit drei bekannten Aktienindizes aus den amerikanischen und chinesischen Märkten durch. Die Ergebnisse zeigen, dass die von DeepTrader optimierte Anlagepolitik sich dem Markttrend gut anpasst und ein optimales Gleichgewicht zwischen Risiko und Ertrag ermittelt. DeepTrader übertrifft sowohl traditionelle Handelsstrategien als auch andere DRL-basierte Strategien in Bezug auf Risiko-Gewinn-Kriterien, einschließlich der annualisierten Sharpe-Ratio, Calmar-Ratio und Sortino-Ratio.

Sawhney et al. (2021) entwerfen ein neues Model, welches unter Verwendung des Deep Deterministic Policy Gradientens trainiert wird. Es verarbeitet Finanznachrichten und Tweets, um vermögenswertbeeinflussende Signale zu modellieren und Handelsentscheidungen zu optimieren. Dafür wird ein zeitlich bewusstes LSTM (t-LSTM) zur Erfassung von Unregelmäßigkeiten bei der Veröffentlichung von Nachrichten eingesetzt. Ein Intraday-Attention-Mechanismus ermöglicht einem Handelsagenten, Texte zu priorisieren, die sich voraussichtlich stärker auf den Preis auswirken werden. Zudem lernt der Aufmerksamkeitsmechanismus, die variable Anzahl versteckter Zustände des t-LSTM adaptiv zu einem tagesinternen Textinformationsvektor zu aggregieren. Anschließend wird dieser Vektor über den zeitlichen Verlauf hinweg auf hierarchische Weise mit einem LSTM kombiniert.

Zur Evaluierung ihres Modells betrachteten sie dabei amerikanische sowie chinesische Aktienmärkte. Dabei wurden für den amerikanischen Aktienmarkt der S&P500 und für den chinesischen Aktienmarkt der A-Shares-Index ausgewählt. Der Einfluss von Finanznachrichten auf die Vermögenspreisentwicklung wird von ihnen auf den amerikanischen Aktienmärkten anhand von englischsprachigen Tweets abgebildet. Für

die chinesischen Aktienmärkte werden die Nachrichtenschlagzeilen mehrerer chinesischer Finanzwebseiten betrachtet. Die Daten des amerikanischen Marktes werden anschließend vom 01.01.2014 bis zum 31.07.2015 für das Training, vom 01.08.2015 bis zum 30.09.2015 für die Validierung und vom 01.10.2015 bis zum 01.01.2016 für das Testen der Modelle aufgeteilt. Demgegenüber wurden für den chinesischen Markt die Zeiträume vom 01.01.2015 bis zum 31.08.2015 für das Training, der Zeitraum 01.09.2015 bis zum 30.09.2015 und für die Validierung der 01.10.2015 bis zum 01.01.2016 für das Testen der Modelle ausgewählt.

In Anbetracht ihrer Ergebnisse argumentieren Sawhney et al., dass ihr Modell höhere risikobereinigte Renditen – gemessen an der Sharpe-Ratio – sowie geringere Verluste – gemessen am Maximum-Drawdown (MDD)– generiert als alle anderen betrachteten Methoden. Zusätzlich stellen sie fest, dass Methoden, die Informationen aus Textquellen über mögliche Auswirkungen auf Aktien einbeziehen, höhere oder vergleichbare Gewinne erzielen wie Methoden, die diese zusätzlichen Informationen nicht einbeziehen. Die Leistungsfähigkeit ihres Modells führen sie dabei darauf zurück, dass dieses die hierarchischen Abhängigkeiten aus den Nachrichtendaten erfasst und mittels eines Attention-Mechanismus lernt, wichtige Handelsindikatoren in volatilen Märkten zu erkennen.

Yu et al. (2019) trainieren einen Agenten anhand einer von ihnen konzipierten RL-Architektur, die aus einem infundierten Vorhersagemodul (IPM), einem generativen adversarischen Datenerweiterungsmodul (DAM) und einem Verhaltensklonmodul (BCM) besteht. Die Funktion eines IPMs ist es, die Vorhersage erwarteter zukünftiger Beobachtungen in moderne RL-Algorithmen einzubeziehen. Die Integration eines DAM begründen sie damit, dass Finanzmärkte nur über begrenzte Daten verfügen. Sie setzen ein DAM ein, das anhand eines generativen adversarischen Netzwerkes synthetische Marktdaten zu erzeugen. Das Integration des BCMs ermöglicht dem RL-Agenten, eine einstufige Expertendemonstration zu beobachten. Dabei erhält der Agent Verhaltensbeispiele von einem Experten und versucht, eine Aufgabe zu lösen, indem er das Verhalten des Experten imitiert.

Der Zeitraum vom 01.01.2005 bis zum 31.12. 2016 wurde zum Trainieren des Agenten verwendet, während der Zeitraum vom 01.01.2017 bis zum 04.12.2018 zum Testen des Agenten benutzt wurde. Als Datengrundlage dienen die Vermögenswertpreise von 8 ausgewählten amerikanischen Aktien, welche laut Yu et al. eine ausgewogenen Mischung aus volatilen und weniger volatilen Aktien darstellten.

Aus ihren Ergebnissen schlussfolgern sie, dass IPM die Portfoliomanagementperformance in Bezug auf Sharpe- und Sortino-Ratios deutlich

verbessert. Dies sehen sie als relevante Erkenntnis für Anleger, die ihre Rendite pro Risikoeinheit maximieren wollen. Darüber hinaus merken sie an, dass DAM dazu beiträgt, eine Überanpassung – insbesondere bei größeren und komplexeren Netzwerkarchitekturen – zu verhindern. Dabei ist jedoch zu beachten, dass dies das Risiko-Ertrags-Verhältnis beeinträchtigen kann. Durch den Einsatz des BCM konnten sie über alle Experimente feststellen, dass dies zur Verringerung des Portfoliorisikos – gemessen an der Volatilität oder dem MDD – beiträgt. Abschließend kommen sie zu der Erkenntnis, dass durch die Aktivierung aller drei Module des von ihnen vorgeschlagenen Ansatzes eine erhebliche Leistungsverbesserung im Vergleich zu den einbezogenen Benchmark- und Baseline-Methoden erzielt werden kann.

Wang et al. (2020) fokussieren sich auf ein PM-Szenario, in dem Portfoliomanager regelmäßig ein neues Portfolio erstellen, um einen langfristigen Gewinn zu erzielen. Händler hingegen befassen sich zur Minimierung der Handelskosten um die beste Ausführung zum günstigsten Preis. In Anlehnung an dieses Rangordnungsszenario schlagen sie ein hierarchisches RL-System vor. Dabei besteht das von ihnen konzipierte System aus einer Hierarchie zweier Entscheidungsprozesse. Die High-Level-Politik ist dem Portfoliomanager nachempfunden und ändert die Portfoliogewichtung mit geringerer Frequenz. Die Low-Level-Politik ist dem Händler nachempfunden und entscheidet, zu welchem Preis und in welcher Menge die Kauf- oder Verkaufsaufträge innerhalb eines kurzen Zeitfensters zu platzieren sind, um die Ziele der High-Level-Politik zu erfüllen. Die High-Level-Politik wurde anhand des REINFORCE-Algorithmus mit einem Entropie-Bonus-Term trainiert, um die Portfolio-Diversifizierung zu fördern. Der Low-Level-Rahmen nutzt das Branching-Dueling-Q-Network, um Agenten mit einem 2-dimensionalen Aktionsraum (Preis und Menge) zu trainieren. Die Experimente wurden anhand von Aktiendaten vom US-Aktienmarkt und vom chinesischen Aktienmarkt durchgeführt. Dafür wählten sie 23 Aktien aus dem Dow Jones Industrial Average Index (DJIA) und 23 Aktien aus dem SSE 50-Index für den US-Aktienmarkt bzw. den chinesischen Aktienmarkt aus. Diesbezüglich wurden für die High-Level-Politik tägliche Daten und für die Low-Level-Politik Daten in einem Frequenzbereich von 30 Sekunden verwendet. Die Daten des amerikanischen Marktes wurden vom 01.03.2000 bis 01.03.2014 für das Training, vom 01.06. 2014 bis Ende 2015 für die Validierung und vom 01.04.2016 bis 22.05.2018 für das Testen der Modelle aufgeteilt. Demgegenüber werden für den chinesischen Markt die Zeiträume vom 08.02.2007 bis 03.07.2015 für das Training, vom 06.07.2015 bis 03.07.2017 für die Validierung und vom 04.07.2017 bis 04.07.2018 für das Testen der Modelle ausgewählt.

Die Ergebnisse des Experimentes auf den chinesischen Aktienmärkten zeigen, dass das von ihnen entworfene Modell alle anderen einbezogenen Methoden – gemessen an der

Sharpe-Ratio, der MDD und der Downside-Deviation-Ratio – übertrifft. Dieses Ergebnis konnte überwiegend auf den amerikanischen Aktienmärkten bestätigt werden. Demnach konnte das von Jiang et al. (2017) konzipierte Modell, welches auf der EIIE-Topologie aufgebaut ist, auf den amerikanischen Märkten eine bessere annualisierte Sharpe-Ratio erzielen.

Lee et al. (2020) konzentrieren sich auf die Tatsache, dass Investmentfirmen nicht nur die Vermögenswerte diversifizieren, aus denen sich die Portfolios zusammensetzen, sondern auch die Portfolios selbst. Daher schlagen sie ein kooperatives, auf RL basierendes Multi-Agenten-Portfoliomanagementsystem (MAPS) vor, das von diesen Diversifizierungsstrategien inspiriert ist, die in großen Investmentgesellschaften verwendet werden. Anstatt eine einzige optimale Strategie zu entwickeln, erstellt MAPS – durch die Verteilung von Vermögenswerten an jeden Agenten – diversifizierte Portfolios. Demnach erstellt jeder Agent sein eigenes Portfolio, basierend auf der aktuellen Marktlage. Die Verlustfunktion von MAPS wurde dabei so gestaltet, dass die Agenten so diversifiziert wie möglich handeln und gleichzeitig ihre eigenen Erträge maximieren. Die Agenten können als eine Gruppe unabhängiger einzelner Investoren betrachtet werden, die zusammenarbeiten, um ein einzelnes diversifiziertes Portfolio zu erstellen. Dabei werden alle Agenten anhand des Deep-Q-Learnings trainiert. Zudem besteht jeder Agent aus einem MLP-Encoder und einem Q-Netz, die innerhalb der Agenten variieren.

Als Datengrundlage für die Experimente werden tägliche Renditen aus 18 Jahren von 3000 amerikanischen Unternehmen aus dem Russel 3000-Index verwendet. Demnach werden die ersten 4 Jahre von 2000–2004 als Trainingsdatensatz benutz und die darauffolgenden 2 Jahre von 2004–2006 zur Validierung des Modells verwendet. Abschließend wird der Zeitraum von 2006–2018 zum umfangreichen Testen des Modells verwendet. Die Ergebnisse zeigen, dass MAPS die risikobereinigten Renditen und die Diversifizierung der Portfolios effektiv verbessert. Zudem merken sie an, dass MAPS in Bezug auf Rendite und Sharpe-Ratio alle anderen berücksichtigten Methoden übertrifft. Darüber hinaus stellen die Autoren fest, dass das Hinzufügen weiterer Agenten zu dem von ihnen konzipierten MAPS System zu einem stärker diversifizierten Portfolio mit höherer Sharpe Ratio führen kann.

Huang und Tanaka (2022) entwickeln basierend auf RL in ihrer Arbeit ein Multi-Agenten-System mit einer modularisierten und skalierbaren Architektur für PM (MSPM). MSPM besteht dabei aus zwei wesentlichen Bestandteilen, zum einem dem Evolving-Agent-Module (EAM), welches anhand von heterogenen Daten mittels eines DQN-basierten Agenten signalbezogene Informationen für einen Vermögenswert erzeugt.

Demnach entscheidet ein EAM – auf Grundlage der neuesten Preis- und Finanznachrichtendaten – ob eine Position gekauft, verkauft oder gehalten wird, um seine Gesamtbelohnung zu maximieren. Der zweite Bestandteil des MSPM ist das Strategic Agent Module (SAM), das – auf der Grundlage des Outputs des EAM – profitable Portfolios anhand eines PPO-Agenten erstellt.

Um die Leistungsfähigkeit ihres erstellten Modells zu untersuchen, wurden von Huang und Tanaka zwei Portfolios anhand des MSPM erstellt. Dabei werden für jedes der beiden Portfolios die Preise von drei amerikanischen Aktien sowie Finanznachrichtendaten einbezogen. In einem ersten Schritt wurden die EAMs mit Daten aus einem Zeitraum von Januar 2009 bis Dezember 2015 trainiert. Daraufhin wurden die Prognosen der trainierten EAMs für den Zeitraum von Januar 2016 bis Dezember 2020 als Datengrundlage für die SAMs verwendet. Anschließend wurden die SAMs anhand eines Datensatzes von Januar 2016 bis Dezember 2018 trainiert und die daraus resultierenden Modelle mit Daten aus dem Zeitraum Januar 2019 bis Dezember 2019 validiert. Schlussendlich wird das MSPM mit Daten aus dem Jahr 2020 evaluiert und mit anderen Modellen verglichen. Aufgrund der von ihnen präsentierten Ergebnisse können Huang und Tanaka bestätigen, dass das von ihnen konzipierte Modell, in Bezug auf die kumulierte Rendite, die tägliche Rendite und die Sortino-Ratio besser abschneidet als die einbezogenen verschiedenen Baselines. Zudem schlussfolgern sie, dass aufgrund der erhöhten Sortino-Ratio, MSPM die abfallende Volatilität besser berücksichtigt und demzufolge höhere risikobereinigte Renditen erzielt.

Für die höheren Anforderungen an die Robustheit und Risikosensitivität von Deep-Reinforcement-Learning im Portfoliomanagement schlagen Liang et al. (2018) ein sogenanntes adversariales Training vor. Dies fügt während der Trainingsphase des Modells den Marktpreisen ein zufälliges Rauschen hinzu. Für ihre Untersuchung wählen sie drei RL-Algorithmen aus – PPO, Deep Deterministic-Policy-Gradient (DDPG). und Policy Gradient (PG). Das adversiale Training wird dabei lediglich auf den PG-Algorithmus angewandt. Die Experimente wurden dabei ausschließlich auf den chinesischen Aktienmärkten durchgeführt. Demzufolge wurde aus einem Aktienpool eine zufällige Auswahl von 5 Aktien ausgewählt, die über eine Handelshistorie von mehr als 1200 Tagen verfügen. Die Experimente zeigen, dass die durch den PG-Algorithmus ermittelte Strategie bei der Vermögensallokation die einheitliche Constant-Rebalance-Portfolio (CRP)-Strategie übertreffen kann. Zudem schlussfolgern sie, dass Deep-Reinforcement-Learning in der Lage ist, Muster von Marktbewegungen zu erfassen, selbst wenn nur eine begrenzte Menge an Daten und Merkmalen zu Verfügung steht, und seine Leistung selbst zu verbessern.

Yang et al. (2020) verwenden für ihre Untersuchung drei auf Actor-Critic basierenden Algorithmen: PPO, Advantage-Actor Critic (A2C) und DDPG. In einem weiteren Schritt wurden diese drei trainierten Modelle in einzelnes Modell komprimiert, der sogenannten Ensemblestrategie. Der Vorteil dieser Ensemblestrategie ist, dass diese die besten Eigenschaften der drei Algorithmen übernimmt und integriert. Diese Eigenschaft führt dazu, dass die Ensemblestrategie sich an unterschiedliche Marktsituationen anpassen kann und somit robust ist. Die von ihnen ausgewählten Algorithmen wurden an 30 Dow-Jones-Aktien getestet, die eine ausreichende Liquidität aufweisen. Dabei verwendeten sie zur Performancebewertung historische Tagesdaten vom 01.01.2009 bis zum 05.08.2020. Zusätzlich wurden neben den täglichen Preisdaten vier weitere Indikatoren wie der Relative-Strength-Index für jede einzelne Aktie miteinbezogen.

Die Agenten wurden in einem Zeitfenster von drei Monaten neu trainiert. Die Validierung der Agenten fand anhand der darauffolgenden 3 Monate statt. Daraufhin wurde der Agent mit der besten Performance und der höchsten Sharpe Ratio ausgewählt. Dieser wurde anschließend für die Vorhersagen und den Wertpapierhandel für das nächste Quartal verwendet. Yang et al. begründen das von ihnen ausgewählte Vorgehen zur Erstellung der Ensemblestrategie dadurch, dass jeder Handelsagent auf unterschiedliche Trends anspricht. Demnach argumentieren sie, dass ein Agent, welcher in einem Aufwärtstrend positive Renditen erzielt, in einem Abwärtstrend negative Ergebnisse erzielt. Zusätzlich argumentieren sie, dass sich mit zunehmendem Sharpe-Ratio eines Agenten seine Erträge im Verhältnis zum eingegangenen Anlagerisiko verbessern. Demnach wird der Handelsagent ausgewählt, der die Rendite unter Berücksichtigung des steigenden Risikos maximieren kann.

Für das Trainieren der Modelle wurde der Datensatz des Zeitraums vom 01.01.2009 bis zum 30.09.2015 verwendet, die Daten vom 01.10.2015 bis zum 31.12.2015 wurden für die Validierung und die Abstimmung der Parameter genutzt. Abschließend wurde die Leistung der Agenten anhand von Handelsdaten vom 01.01.2016 bis zum 08.05.2020 evaluiert, die nicht in der Stichprobe enthalten waren. Um diese besser nutzen zu können, wurde das Training der drei Agenten während der Handelsphase fortgeführt. Dies soll dazu beitragen, dass sich die Modelle besser an die Marktdynamik anpassen können, um die Leistungsfähigkeit zu erhöhen. Um die Performance der drei Agenten und der Ensemblestrategie adäquat zu vergleichen, werden der DJIA und die traditionelle Minimum-Varianz-Portfolio-Allokationsstrategie in die Untersuchung einbezogen.

Die Ergebnisse zeigen, dass die von ihnen konzipierte Ensemblestrategie sowie die drei ausgewählten Algorithmen den Dow Jones Industrial Average und die Min-Varianz-Portfolio-Allokationsmethode in Bezug auf die Sharpe-Ratio übertreffen. Daraus

schlussfolgern die Autoren, dass die von ihnen ausgewählten und erstellten Algorithmen die einbezogenen Methoden übertreffen, indem sie Risiko und Rendite unter Berücksichtigung der Transaktionskosten miteinander in Einklang bringen.

Pretorius und van Zyl (2022) vergleichen die Portfoliomanagement-Performance von RL-Methoden mit traditionellen Mean-Variance-Methoden. Zusätzlich erstellten sie ein Modell, das unterschiedliche Anlegerpräferenzen von Investoren berücksichtigt und dazu in der Lage ist, Ergebnisses in eine optimale Pareto-Grenze im Risiko-Rendite-Raum zu erschaffen. Dies wurde durch das Einbeziehen verschiedener Parameter für die Anlegerpräferenzen in die Belohnungsfunktionen der RL-Modelle erreicht. Dieses Vorgehen ermöglicht zudem einen umfassenden Vergleich zwischen RL-Modellen und traditionellen Mean-Variance-Optimierungsmodellen im Risiko-Ertrags-Raum. Die Integration von RL für die Portfoliooptimierung begründen sie dadurch, dass diese die Fähigkeit besitzen, erwartete Erträge über längere Zeiträume zu optimieren. Demnach argumentieren sie, dass diese langfristige Optimierung bei der Betrachtung von Portfoliomanagementproblemen wichtig ist, da unmittelbare Handlungen aufgrund von Transaktionskosten die Fähigkeit eines Agenten beeinträchtigen können, in der Zukunft optimale Belohnungen zu erzielen. In der Untersuchung wurden drei bestehende Actor-Kritik-RL-Modelle als Vergleichsgrundlage für das von ihnen konzipierte RL-Modell verwendet. Die drei verwendeten Methoden waren: A2C, PPO und DDPG, die alle drei bereits in der Studie von Yang et al. (2020) verwendet wurden.

Um die von ihnen ausgewählten Modelle unter verschiedenen Marktbedingungen zu bewerten, wurden Daten für drei unterschiedliche Märkte erhoben, die jeweils unterschiedliche Volkswirtschaften und allgemeine Markttrends repräsentierten. Für die Auswahl der Märkte orientierten sich Pretorius und van Zyl an zwei Zielen: Das Hauptziel besteht darin, drei Märkte auszuwählen – einen mit steigenden Markttrend; einen mit sinkendem allgemeinen Trend und einen, bei dem der allgemeine Trend stabil oder seitwärtsgerichtet ist. Ein zweites Ziel besteht darin, Märkte auszuwählen, auf denen diese Trendbedingungen über ausreichend lange Zeiträume vorhanden sind, so dass sich diese Charakteristika sowohl auf den Trainings- als auch auf den Testzeitraum erstrecken.

Für einen Markt in einem Aufwärtstrend entschieden sie sich für den amerikanischen DJIA mit 30 gelisteten Aktien. Des Weiteren wurden der Nikkei 225 und der Latin America 400-Index mit jeweils 24 Aktienwerten in die Untersuchung miteinbezogen, um einen seitwärts gerichteten und einen abwärts gerichteten Markt zu repräsentieren. Die Ergebnisse der Untersuchung zeigen, dass die von ihnen konzipierten RL-Modelle die traditionellen Mean-Variance-Optimierungsmethoden in aufwärts tendierenden Märkten bis zu einer gewissen Überschussrisikogrenze deutlich übertreffen. Zudem deuten die

Ergebnisse darauf hin, dass in seitwärts tendierenden Märkten die Leistung der ausgewählten RL-Modelle die der traditionellen Methoden für den Großteil des getesteten Überschussrisikobereichs nahezu erreicht werden kann. Abschließend schlussfolgern Pretorius und van Zyl, dass ihre Ergebnisse darauf hindeuten, dass die Verwendung von RL-Methoden im Vergleich zu traditionellen Mean-Variance-Optimierungsmethoden für das Portfoliomanagement von Vorteil sein kann. Diese Eigenschaft sehen sie dadurch begründet, dass RL-Methoden die erwarteten zukünftigen Erträge unter bestimmten Marktbedingungen über längere Zeiträume optimieren.

Zhang et al. (2020) verwendeten innerhalb ihrer Untersuchung Deep-Learning-Modelle zur direkten Optimierung der Sharpe-Ratio eines Portfolios. Dabei nutzen sie eine einzelne Schicht von LSTM-Konnektivität mit 64 Einheiten, um die Portfoliogewichte zu modellieren und dann das ökonometrische Verständnis der Sharpe-Ratio zu optimieren. Um ihr Modell zu untersuchen, verwendeten sie tägliche Renditen von vier amerikanischen Exchange-Traded-Funds. Dies begründen sie dadurch, dass der Handel mit Indizes Vorteile gegenüber dem Handel mit Einzelwerten bietet, da diese Indizes in der Regel unkorreliert sind, was zu einer Diversifizierung des Portfolios führt. Ein solches diversifiziertes Portfolio liefert wiederum eine höhere Rendite pro Risiko. Die Daten für ihre Untersuchung umfassen die Jahre von 2006 bis 2020, während sich der Testzeitraum insgesamt von 2011 bis Ende April 2020 erstreckt. Zusätzlich verglichen sie ihre Methode mit einer Gruppe von Grundmodellen. Bei der ersten Gruppe von Basismodellen handelt es sich um Reallokationsstrategien, die den relevanten Vermögenswerten ein festes Allokationsverhältnis zuweisen. Diese wurden anschließend jährlich neu ausgeglichen, um das gewählte Verhältnis beizubehalten. Die zweite Gruppe von Vergleichsmodellen sind die Mean-Variance-Optimierung (MVO) und die maximale Diversifikation (MD). Für beide wurden gleitende Durchschnitte mit einem rollierenden Fenster von 50 Tagen verwendet, um die erwarteten Renditen und die Kovarianzmatrix zu schätzen. Die Portfoliogewichte werden täglich aktualisiert und es werden Gewichte ausgewählt, die die Sharpe Ratio für MVO maximieren. Der letzte Basisalgorithmus ist der des diversitätsgewichteten Portfolios, der die Portfoliogewichte mit der Marktkapitalisierung der Vermögenswerte in Relation setzt.

Ihre Ergebnisse zeigen, dass das von ihnen erstellte Modell die beste Performance bei allen Bewertungsmaßstäben – mit Ausnahme eines etwas größeren Drawdowns – liefert. Diese Ergebnisse können sie ebenso in einem Szenario mit höheren Kosten bestätigen, da ihr Modell ebenfalls die beste erwartete Rendite und die höchsten Sharpe- und Sortino-Ratio erzielt.

Soleymani und Paquet (2020) schlagen in ihrer Arbeit ein Verfahren zur Portfolioverwaltung und -optimierung auf der Grundlage eines Deep-Reinforcement-Learning Modells mit dem Namen DeepBreath vor. Die DeepBreath-Methode kombiniert einen Restricted-Stacked-Autoencoder und ein CNN, die anhand eines SARSA-Algorithmus trainiert wurden. Um die mit der Größe des Trainingsdatensatzes verbundene Rechenkomplexität zu verringern, verwendeten sie einen eingeschränkten gestapelten Autoencoder, um nicht korrelierte und hochinformative Merkmale zu erhalten. Der Algorithmus wurde zunächst offline trainiert, was zu einem vortrainierten Modell anhand historischer Daten führt. Anschließend wurden die Gewichte nach einem Online-Lernschema aktualisiert, um der Entwicklung des Marktes zu folgen. Schließlich wurde die Leistung des DeepBreath-Modells anhand von vier Testreihen über drei verschiedene Investitionszeiträume von 30, 60 und 90 Tagen getestet.

Um die Leistungsfähigkeit ihres Modells zu demonstrieren, wählten sie eine Auswahl an amerikanischen Aktien aus, die aus verschiedenen Industrie Sektoren wie dem Technologie-, Finanzen- und Gesundheitswesen stammen. Um die Leistung ihres Systems weiter zu vergleichen, wird die erwartete Kapitalrendite des Modells mit der erwarteten Kapitalrendite des DJIA verglichen. Der Zeitraum des Datensatzes beginnt am 02.01.2002 und endet am 31.07.2018 und beinhaltete Eröffnungs-, Tiefst-, Höchst- und Schlusskurse aller betrachteten Vermögenswerte. Zusätzlich werden acht unterschiedliche Finanzindikatoren für jede Aktie in die Untersuchung einbezogen. Jeder Trainingssatz deckt einen anderen Zeitraum oder ein anderes Fenster ab: Alle Zeiträume beginnen am 02.01.2002 und enden jeweils am 15.08.2008, am 06.12.2011, am 06.04.2015 und am 19.05.017. Diesen vier Trainingsdatensätzen entsprechen vier Testdatensätze von jeweils 90 Tagen, die am Tag nach dem Ende des jeweiligen Trainingssets beginnen. Die Ergebnisse zeigen, dass die von DeepBreath erzielte Rendite aktuelle Experten-Anlagestrategien übertrifft und gleichzeitig das Marktrisiko minimiert.

In einer darauffolgenden Arbeit erweiterten Soleymani und Paquet (2021) das von ihnen vorgestellte DeepBreath-Modell und erstellten ein Graph-Convolutional-Reinforcement-Learning-Verfahren namens DeepPocket. Dabei argumentieren sie, dass Vermögenswerte nicht voneinander unabhängig, sondern innerhalb eines kurzen Zeitraums korreliert sind. Demnach ist das Ziel des konzipierten Modells, die zeitlich variierenden Wechselbeziehungen zwischen Finanzinstrumenten auszunutzen. Um dieses Ziel zu erreichen, wurden diese Wechselbeziehungen durch einen Graphen dargestellt. Innerhalb dieses Graphen werden Vermögenswerte als Knoten dargestellt, während die Kanten einer paarweisen Korrelationsfunktion zwischen den Vermögenswerten entsprechen. Eine weitere Veränderung des DeepPocket-Modells ist die Verwendung eines Akteur-Kritiker-Netzwerkes, das anhand des entsprechenden Algorithmus trainiert

wurde. Die Akteur-Kritiker-Struktur enthält zwei Faltungsnetzwerke, in denen der Akteur eine Investitionspolitik erlernt und durchsetzt, die wiederum vom Kritiker bewertet wird, um die beste Vorgehensweise zu bestimmen. Die Leistung des DeepPocket-Modells wurde anhand von fünf anstatt vier Testreihen über drei verschiedene Investitionszeiträume von 30, 60 und 90 Tagen getestet. Dabei betrachteten sie ebenfalls eine Auswahl an amerikanischen Aktien aus unterschiedlichen Industriesektoren, wie dem Gesundheits-, Finanz- und Gesundheitswesen. Zusätzlich weiteten sie für ihre Untersuchung den Zeitraum ab dem 02.01.2002 bis zum 31.07.2018 auf den 24.03.2022 aus. In einem weiteren Schritt wurden vier Benchmark-Portfolios, die jeweils aus dem DJIA, dem EURO STOXX 50 ETF, Nasdaq Composite Index, und dem S&P500 bestehen, mit DeepPocket verglichen.

Aufgrund der von ihnen erzielten Ergebnisse kommen Soleymani und Paquet zu der Erkenntnis, dass das von ihnen konzipierte Modell über alle 5 Testzeiträume und alle drei Anlagenzeiträume hinweg eine bessere Performance gegenüber den verglichenen Portfolios aufweist. Zusätzlich heben sie die während der Covid-19-Krise erzielten Ergebnisse hervor, da es DeepPocket gelungen ist, durch die Nutzung profitabler Vermögenswerte wie Amazon und Facebook sowie durch die Nutzung von Offline-Wissen, z. B. aus der globalen Finanzrezession 2008, einen Gewinn zu erzielen. Diese Leistung steht im Gegensatz zu den Marktindizes, die im gleichen Zeitraum erheblich an Wert verloren haben. Diese festgestellte Leistungsfähigkeit sehen sie darin begründet, dass das von ihnen konzipierte neuronale Netzwerk die zugrundeliegenden Regeln, Verbindungen und vorhersehbaren Muster direkt aus den Daten erlernt. Demnach passt sich das Netzwerk an, wenn neue Informationen verfügbar werden, wodurch das System bessere Vorhersagen auf der Grundlage von Erkenntnissen aus zuvor analysierten Informationen treffen kann. Abschließend kommen Soleymani und Paquet zu der Schlussfolgerung, dass das von ihnen erstellte Modell in der Lage ist, Anomalien wie unerwartete Ausschläge, Einbrüche, Niveauverschiebungen und Trendwechsel zu erkennen. Zusätzlich merken sie an, dass die Menge an komplexen Informationen, die DeepPocket verarbeiten kann, bei weitem die menschlichen Fähigkeiten übersteigt.

Der von Lim et al. (2021) vorgestellte Algorithmus ist in der Lage, das Portfoliomanagement zu verbessern, indem er das dynamische Rebalancing von Portfolios mit starkem Risiko durch einen auf einem Q-Netz basierenden RL-Agenten ermöglicht. In ihrer Arbeit betrachteten sie vier verschiedene Methoden, die – mit und ohne Verwendung des LSTM-Modells – zur Vorhersage von Aktienkursen für die Anpassung der technischen Indikatorzentrierung kombiniert werden sowie die vollständige und die schrittweise Umschichtung des Portfolios – ohne und mit Preisvorhersagemodellen – auf der Grundlage technischer Indikatoren. Die Leistung der

vier Methoden wurde anschließend anhand von drei konstruierten Finanzportfolios bewertet und verglichen. Diese drei Portfolios bestehen zum einem aus einem Portfolio mit globalen Marktindexwerten unterschiedlicher Risikoniveaus aus einem Zeitraum von 2014 bis 2018 sowie aus zwei Portfolios mit unkorrelierten Aktienwerten aus unterschiedlichen Sektoren und Risikoniveaus aus dem Nasday Composite Index und dem S&P 500 für einen Zeitraum von Januar 2016 bis Dezember 2018.

Aus den von Lim et al. dargestellten Versuchsergebnissen geht hervor, dass der RL-Agent mit schrittweisem Portfolio-Rebalancing und LSTM-Vorhersagemodell besser abschneidet, da er bei der schrittweisen Anpassung der Portfoliozusammensetzung das entsprechende Handelsverhalten nutzt, anstatt die Portfoliozusammensetzung an einem einzigen Handelstag zu ändern. Die Experimente zeigen, dass ein korrekt abgestimmter RL-Agent mit und ohne LSTM-Kursvorhersagemodell – das technische Indikatoren in den Mittelpunkt stellt – ein dynamisches Rebalancing mit Risikoanpassung zur Verbesserung der Portfoliorenditen nutzen kann. Der vorgeschlagene Algorithmus zeigt zudem, dass die Strategie des dynamischen Portfolio-Rebalancings mit starken Risiken in Verbindung mit den Konzepten der SAA- und TAA-Strategien positive Ergebnisse erzielt.

Pendharkar und Cusatis (2018) betrachten in ihrer Forschungsarbeit ein persönliches Ruhestandsportfolio mit zwei Vermögenswerten und schlagen mehrere RL-Agenten für den Handel mit Portfoliovermögenswerten vor. Dabei verwenden sie On- und Off-Policy Algorithmen sowie diskrete und kontinuierliche Agenten in ihren Experimenten. Für ihre Untersuchungen gingen sie von einem minimalistischen Ruhestandsportfolio aus, das auf zwei Anlageklassen beruhte. Diese bestanden zum einen aus dem S&P 500 Indexfonds/ETF und zum anderen entweder aus dem AGG Bond Index oder der 10-jährigen US- Staatsanleihe. Das Anlageziel der unterschiedlichen Agenten bestand darin, eine Handelsstrategie zu ermitteln, die entweder die Portfoliorenditen oder die unterschiedlichen Sharpe-Ratios über lange Anlagezeiträume von 10 Jahren oder länger maximiert. Die Daten des S&P 500 und des AGG Bond Index wurden aus den Jahren 1976–2016 verwendet, für Experimente mit T-Note-Daten betrachteten sie einen Zeitraum von 46 Jahren, welcher 1970 beginnt und 2016 endet. Zusätzlich wurden drei verschiedene Handelszeiträume betrachtet: vierteljährlich, halbjährlich und jährlich.

Die Ergebnisse der von ihnen durchgeführten Experimente zeigen, dass ein adaptiver Agent mit kontinuierlichen Aktionen bei der Vorhersage von Portfolioallokationen für die nächste Periode durchweg am besten abschneidet. Demzufolge kommen sie zu der Schlussfolgerung, dass RL-Agenten erfolgreich von Einzelpersonen zur Verwaltung ihrer Ruhestandsportfolios eingesetzt werden können.

3 Methode

Reinforcement-Learning ist eine selbstlernende Methode, bei der der Agent ohne definiertes Modell und mit wenigen Vorinformationen mit der Umgebung interagiert und durch Erkundung der Umgebung lernt, während er gleichzeitig seine Strategie optimal aktualisiert, um eine Leistungsmetrik zu maximieren.

RL besteht aus einem Agenten, der den kontrollierten Zustand des Systems und eine mit dem letzten Zustandsübergang verbundene Belohnung erhält. Anschließend wählt dieser eine Aktion a_t aus, die an das System zurückgesendet wird. Als Reaktion darauf geht das System in einen neuen Zustand S_{t+1} über, woraufhin der Agent eine Belohnung R_{t+1} erhält.

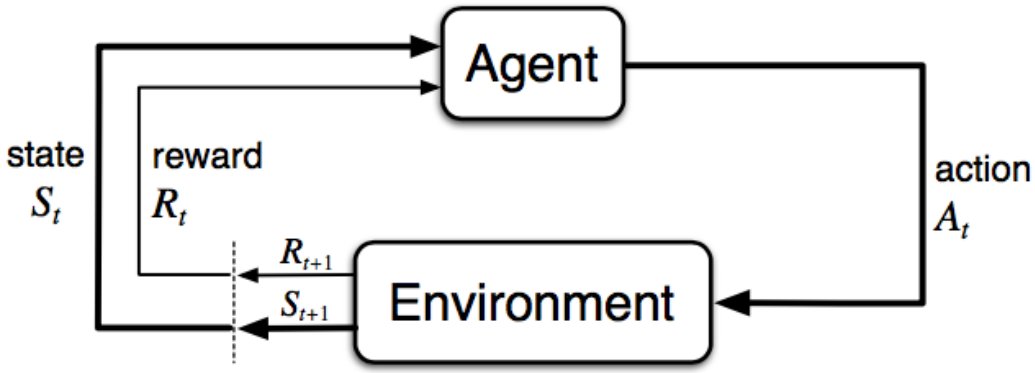

Abbildung 1: Interagieren des Agenten mit der Umgebung Quelle (Sutton und Barto 2018)

Ziel ist es, eine Reihe von Aktionen für jeden Zustand (Politik) zu erlernen, um die kumulative Belohnung

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \qquad (1)$$

in einer dynamischen Umgebung zu maximieren. Dabei ist $\gamma \in (0,1]$ ein Diskontierungsfaktor, wobei niedrigere Werte den unmittelbaren Belohnungen mehr Bedeutung beimessen. Für die Parameterwahl $\gamma = 1$ gilt, dass eine Belohnung ihren vollen Wert zu jedem zukünftigen Zeitindex behält, unabhängig von der relativen Zeit, ab dem aktuellen Zeitschritt. Wenn γ abnimmt, nimmt die Wirkung der Belohnung in der Zukunft exponentiell um γ ab.

Die Umgebung für RL wird häufig als ein Markov-Entscheidungsprozesses (MDP) mit einer Menge von Zuständen $S \in S$ und einer Menge von Aktionen $A \in A$ beschrieben.

In einem MDP interagiert der Entscheidungsträger – der Agent – kontinuierlich mit der Umwelt, d. h. mit allem, was sich außerhalb des Agenten befindet. In jeder Periode $t, t = 1, 2, \ldots, T$, beobachtet der Agent den Zustand der Umwelt, $\mathrm{S_t} \in S$. Anschließend wählt er eine Aktion $A_t \in A$ aufgrund des beobachteten Zustandes aus. Als Folge der Aktion erhält der Agent eine numerische Belohnung, $R_{t+1} \in R \subset R$, und das System wechselt in den nächsten Zustand – $\mathrm{S_{t+1}}$ – gemäß den zeitinvarianten Zustandsübergangswahrscheinlichkeiten F. Diese beschreiben eine Wahrscheinlichkeitsverteilung für jede Kombination von $A \in A$ und $S \in S$:

$$F(s', r|s, a) = Pr\{\mathrm{S_{t+1}} = S', R_{t+1} = R \mid S_t = s, \mathrm{A_t} = A\}. \quad (2)$$

Die zukünftigen Belohnungen des Agenten hängen von seiner Handlungsweise ab, die als Policy bezeichnet wird:

$$\pi(\mathrm{a}|\mathrm{s}) = \mathbb{P}_\pi[A = \mathrm{A}|S = \mathrm{S}]. \quad (3)$$

Die Policy beschreibt, mit welcher Wahrscheinlichkeit ein Agent $A_t = \mathrm{A}$ wählen wird, während er sich in einem Zustand $S_t = \mathrm{S}$ befindet. Somit definiert π eine Wahrscheinlichkeitsverteilung über jedes $\mathrm{A} \in A$ für jedes $\mathrm{S} \in S$ (Sutton und Barto 2018).

Um eine optimale Strategie π^* zu finden, die den maximalen erwarteten Ertrag aus allen Zuständen erzielt:

$$\pi^* = argmax\, \pi\, E[G|\pi] \quad (4)$$

werden zwei Hauptansätze für die Lösung des Problems verwendet. Demnach basieren eine Methode auf der Grundlage von Wertfunktionen und eine Methode auf der Grundlage der Politiksuche. Darüber hinaus existiert ein hybrider, Akteurs-kritiker Ansatz, der sowohl Wertfunktionen als auch Policy-Suche einsetzt (Arulkumaran et al. 2017).

Die Wertfunktion eines Zustands s unter einer Strategie π, bezeichnet als $V_\pi(\mathrm{S})$, ist der erwartete Ertrag, wenn man in einem Zustand S beginnt und anschließend einer Policy π folgt. Demnach können MDPs $V_\pi(\mathrm{S})$ formal definieren werden durch

$$V_\pi(\mathrm{S}) = \mathbb{E}_\pi[G_t|S_t = \mathrm{S}] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|S_t = \mathrm{S}\right], für\ alle\ \mathrm{S} \in S, \quad (5)$$

wobei $\mathbb{E}_\pi$[-] den Erwartungswert einer Zufallsvariable unter der Voraussetzung bezeichnet, dass der Agent die Politik π verfolgt.

In ähnlicher Art und Weise wird ein Wert der Durchführung einer Aktion a im Zustand s unter einer Strategie π definiert. Dieser wird als Aktionswertfunktion für die Politik π oder als Q-Wert bezeichnet und wie folgt definiert:

$$Q_\pi(\mathrm{S},\mathrm{A}) = \mathbb{E}_\pi[G_t|S_t = \mathrm{S}, A_t = \mathrm{A}] = \mathbb{E}_\pi[\textstyle\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\,|S_t = \mathrm{S}, A_t = \mathrm{A}]. \qquad (6)$$

Die optimale Wertfunktion wird entsprechend in folgender Weise dargestellt (Li 2018):

$$V_*(\mathrm{S}) = \max_{a\in\mathcal{A}(s)} Q_{\pi_*}(\mathrm{S},\mathrm{A}). \qquad (7)$$

Demgegenüber müssen Policy-Suchmethoden kein Wertfunktionsmodell verwenden, sondern suchen direkt nach einer optimalen Policy π^*. In der Regel wird eine parametrisierte Politik π_θ gewählt, deren Parameter θ aktualisiert werden, um den erwarteten Ertrag $E_\pi[G|\theta]$ entweder durch gradientenbasierte oder gradientenfreie Optimierung zu maximieren (Deisenroth 2011).

3.1 Proximal Policy Optimization

Der von Schulman et al. (2017) entwickelte PPO-Algorithmus ist ein On-Policy modellfreier Akteur-Kritiker Algorithmus, d.h. ermittelt dieser nicht ausschließlich eine Politik π (Akteur), sondern auch eine Wertfunktion $v_\pi(\mathrm{S})$ (Kritiker), die den Wert der abgezinsten Belohnung schätzt, die der Agent am Ende des Prozesses nach der Politik π ausgehend von einem beliebigen Zustand S erhalten wird (Charpentier et al. 2020). On-Policy-Methoden versuchen, die Politik zu bewerten oder verbessern, die für die Entscheidungsfindung verwendet wird, während Off-Policy-Methoden eine unterschiedliche Politik als die, die zur Erzeugung der Daten verwendet wird, bewerten oder verbessern (Sutton und Barto 2018). Im Gegensatz zur modellbasierten Methode lernen modellfreie Algorithmen direkt eine Wert- (oder Zustandswert-) Funktion oder eine optimale Strategie, ohne auf das Modell zu schließen (Hambly et al. 2021). Die Akteurskritik zielt darauf ab, die Vorteile der beiden Value- und Policy- Ansätze zu nutzen. Durch die Verschmelzung beider Ansätze können kontinuierliche und stochastische Umgebungen, die schnellere Konvergenz des Policy-Lernens sowie die Stichprobeneffizienz und die Stetigkeit des Value-Ansatzes genutzt werden. Im Actor-Critic-Ansatz interagieren zwei Modelle, um die beste kumulative Belohnung zu erhalten. Durch gleichzeitige Verwendung eines Akteurs, der die Policy-Parameter aktualisiert, und eines Kritikers, der die Wertfunktion oder Aktions-Wert-Funktion aktualisiert, ist dieses Modell in der Lage, sowohl komplexe Umgebungen als auch komplexe Wertfunktionen zu erlernen (Charpentier et al. 2020). Zudem ist PPO eine Verbesserung

des Trust-Region-Policy-Optimization (TRPO)-Algorithmus. Dieser zielt darauf ab, ein Stellvertreter-Lernziel zu optimieren, das vordefinierten Verhaltensrestriktionen unterliegt. Dabei ist das Hauptzieles des TRPO, große Schritte im Parameterraum zu machen und dabei die Richtlinien nahe genug zu halten, um den Zusammenbruch des Modells zu vermeiden Um dieses Lernproblem anzugehen, werden sowohl eine lineare Annäherung des Lernziels als auch eine quadratische Annäherung der Beschränkung verwendet, um die Aktualisierung der Strategie gemeinsam zu steuern, was zu einer hohen Berechnungskomplexität führt. TRPO ist jedoch nicht mit NN-Architekturen für die gemeinsame Nutzung von Parametern (zwischen der Strategie und der Wertfunktion) kompatibel (Schulman et al. 2017). Zudem eignet sich TRPO auch nicht für die Durchführung mehrerer Epochen der Minibatch-Politikaktualisierung, welches für eine hohe Stichprobeneffizienz unerlässlich ist. Um diese Schwierigkeiten zu beheben, wurde von Schulman et al. der PPO-Algorithmus entwickelt. Dieser interagiert asynchron mit der Umwelt und optimiert dabei indirekt eine Policy $\pi_\theta(a_t|s_t)$ durch eine geklippte Zielfunktion, die die Differenz zwischen der neuen Policy nach der letzten Aktualisierung $\pi_\theta(a|s)$ und der alten Policy vor der letzten Aktualisierung $\pi_{\theta_{alt}}(a|s)$ darstellt. Um einen Zusammenbruch des Modells zu verhindern und die Lernstabilität zu erhöhen, wird die Zielfunktion sorgfältig beschnitten, um Parameteraktualisierungen zu bestrafen, die die neue Politik zu weit von der alten entfernen.

Chen et al. (2018) argumentieren zudem, dass aufgrund der verwenden Optimierungsmethoden erster Ordnung, PPO einfacher zu implementieren und effizienter zu betreiben ist als TRPO. Darüber hinaus ermöglicht PPO die wiederholte Optimierung von Richtlinien auf der Grundlage zuvor gesampelter Daten, um die Komplexität von Stichproben zu reduzieren. Für das beschnittene Ersatzziel des PPO Algorithmus wird die Verlustfunktion als:

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min\left(r_t(\theta)\hat{A}_t\,, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right) - c_1 L_t^{VF}\right]$$

$$mit\ r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{alt}}(a_t \mid s_t)}$$

$$und\ clip(r_t(\theta), 1-\epsilon, 1+\epsilon) = \begin{cases} r_t(\theta),\ für\ 1-\epsilon \le r_t(\theta) \le 1+\epsilon \\ 1-\epsilon,\ für\ r_t(\theta) < 1-\epsilon \\ 1+\epsilon,\ für\ 1+\epsilon < r_t(\theta) \end{cases} \quad (8)$$

$$und\ L_t^{VF} = \hat{E}_t[(V_\pi(S_t) - G_t)^2]$$

definiert. Dabei ist $r_t(\theta)$ das Wahrscheinlichkeitsverhältnis zwischen der alten und der neun Policy. Für den Fall, dass $r_t(\theta)$ einen Wert größer als 1 annimmt, bedeutet dies, dass die Aktion a_t in der aktuellen Policy mit höherer Wahrscheinlichkeit durchgeführt

wird als in der alten. Demgegenüber bedeutet ein Wert von $r_t(\theta)$ zwischen 0 und 1, dass die Aktion a_t in der aktuellen Policy mit einer geringeren Wahrscheinlichkeit durchgeführt wird als in der alten Policy. L_t^{VF} bildet den durchschnittlichen quadrierten Fehlerterm der Wertfunktion $V_\pi(S_t)$ mit den kumulierten Belohnungen G_t ab und wird als Verlustfunktion für den Kritiker verwendet, dessen Einfluss auf die gesamte Verlustfunktion mit dem Parameter c_1 bestimmt wird. Die Funktion $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)$ sorgt dafür, dass das Wahrscheinlichkeitsverhältnis $r_t(\theta)$ auf eine Reichweite von $[1-\epsilon, 1+\epsilon]$ beschränkt wird.

Um den Advantage Wert $\hat{A}_t$ zu berechnen wird der Generalized-Advantage-Estimator (GAE) (Schulman et al. 2015):

$$\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V$$
$$\text{mit } \delta_t^V = R_t + V(s_{t+1}) - V(s_t) \tag{9}$$

verwendet. Der Parameter λ kann hierbei einen Wert von 0 bis 1 annehmen. Schulman et al. argumentieren zudem, dass $GAE(\gamma, 1)$ eine hohe Varianz aufweist, während $GAE(\gamma, 0)$ eine hohe Verzerrung und eine geringe Varianz aufweist. Demnach stellt der generalisierte Vorteilsschätzer für eine Parameterwahl von $0 < \lambda < 1$ einen Kompromiss zwischen Verzerrung und Varianz dar, der durch den Parameter λ gesteuert wird.

3.2 Agent

Analog zu Jiang et al. (2017) werden folgende Definitionen verwendet. Demzufolge besteht die Aktion des Agenten darin, den optimalen Gewichtungsvektor für das Portfolio zu bestimmen, um eine Belohnung R_t zu maximieren und kann daher durch:

$$a_t = w_t \tag{10}$$

abgebildet werden. Aufgrund der Beziehung zwischen der Rendite und dem Transaktionsfaktor wird die aktuelle Aktion teilweise durch die vorhergehende Aktion bestimmt (Soleymani und Paquet 2020). Demnach besteht der aktuelle Zustand s_t aus zwei Teilen: dem externen Zustand und dem internen Zustand. Der externe Zustand ist eine dreidimensionale Preismatrix X_t und der interne Zustand wird durch den vorherigen gewählten Portfolioaktionsgewichtsvektor w_{t-1} dargestellt. Somit wird der Zustand des Agenten als:

$$s_t = (X_t, w_{t-1}) \tag{11}$$

dargestellt. Die dreidimensionale Preismatrix X_t aus der Stapelung der Matrizen des Schlusskurses, des höchsten sowie des niedrigsten Preises ergibt:

$$X_t = [\boldsymbol{V}_t, \boldsymbol{V}_t^{(\mathrm{hi})}, \boldsymbol{V}_t^{(\mathrm{lo})}]. \tag{12}$$

Zudem wurden alle drei Preise jeweils anhand einer elementweisen Division, dargestellt als $\oslash$, durch den letzten Schlusskurs des Datensatzes normalisiert:

$$\begin{aligned}
\boldsymbol{V}_t &= [\boldsymbol{v}_{t-n+1} \oslash \boldsymbol{v}_t | \boldsymbol{v}_{t-n+2} \oslash \boldsymbol{v}_t | \cdots | \boldsymbol{v}_{t-1} \oslash \boldsymbol{v}_t | \mathbf{1}], \\
\boldsymbol{V}_t^{(\mathrm{hi})} &= \left[\boldsymbol{v}_{t-n+1}^{(\mathrm{hi})} \oslash \boldsymbol{v}_t \middle| \boldsymbol{v}_{t-n+2}^{(\mathrm{hi})} \oslash \boldsymbol{v}_t \middle| \cdots \middle| \boldsymbol{v}_{t-1}^{(\mathrm{hi})} \oslash \boldsymbol{v}_t \middle| \boldsymbol{v}_t^{(\mathrm{hi})} \oslash \boldsymbol{v}_t\right], \\
\boldsymbol{V}_t^{(\mathrm{lo})} &= \left[\boldsymbol{v}_{t-n+1}^{(\mathrm{lo})} \oslash \boldsymbol{v}_t \middle| \boldsymbol{v}_{t-n+2}^{(\mathrm{lo})} \oslash \boldsymbol{v}_t \middle| \cdots \middle| \boldsymbol{v}_{t-1}^{(\mathrm{lo})} \oslash \boldsymbol{v}_t \middle| \boldsymbol{v}_t^{(\mathrm{lo})} \oslash \boldsymbol{v}_t\right], \\
&mit\ \mathbf{1} = [\mathbf{1}, \mathbf{1}, \ldots, \mathbf{1}]^{\boldsymbol{T}}.
\end{aligned} \tag{13}$$

Als Belohnungsfunktion für den Agenten werden drei unterschiedliche Kennzahlen angewandt. Die erste Belohnungsfunktion ist die der nominalen Rendite des Portfolios:

$$r_t = \frac{p_t}{p_{t-1}} - 1 = \frac{p_t - p_{t-1}}{p_{t-1}} \tag{14}$$

welche den analog zu Betancourt und Chen (2021) verwendet wird.

Um sicherzustellen, dass die durch den Agenten gewählte Strategie sowohl die Rendite als auch das Risiko einer Investition berücksichtigt, wird in Übereinstimmung zu Wang et al. (2019) und Leem und Kim (2020) die Sharpe-Ratio (Sharpe 1994):

$$SR_t = \frac{E(r_t - r_f)}{\sqrt{Var(r_t - r_f)}} \tag{15}$$

verwendet. Dabei ist r_t die tägliche Rendite des Portfolios und r_f bildet die risikofreie Rendite ab. Die Sharpe-Ratio wird berechnet in dem der Erwartungswert der Differenz der Renditen durch die Standardabweichung der Differenz der Renditen geteilt wird. Um die Sharpe-Ratio für die Periode t zu bestimmen, werden demnach die Portfoliorenditen der gesamten vorherigen Perioden, einschließlich bis zur Periode t, berücksichtigt. Srivastava und Mazhar (2018) argumentieren, dass die Sharpe Ratio die Risikoprämie pro Risikoeinheit darstellt, welche durch die Standardabweichung des Portfolios quantifiziert wird. Demnach misst die Sharpe-Ratio die Performance des Portfolios im Vergleich zum

eingegangenen Risiko: Je besser die Performance ist, desto größer ist der Gewinn durch die Inkaufnahme zusätzlicher Risiken.

Es ist generell anerkannt, dass eine Sharpe-Ratio unter eins als nicht optimal gilt, eine Ratio zwischen eins und zwei wird als akzeptabel betrachtet, während eine Ratio zwischen zwei und drei als sehr gut gilt. Ein Verhältnis von mehr als drei wird als ausgezeichnet angesehen (Maverick 2015).

Eine weitere Belohnungsfunktion die in die Untersuchung mitaufgenommen wird, ist die von Sortino und van der Meer (1991) konzipierte Sortino-Ratio, die ebenfalls von Leem und Kim (2020) und Benhamou et al. (2020b) verwendet wird. Die Sortino-Ratio ist eine Abwandlung der Sharpe-Ratio, verwendet jedoch die Abwärtsabweichung und nicht die Standardabweichung als Risikomaß. Diese Vorgehensweise ist damit begründet, dass die Sichtweise der Sortino-Variante der Auffassung ist, dass die Aufwärtsvolatilität ein Vorteil für die Wertanlage darstellt und daher nicht in die Risikoberechnung einbezogen werden sollte. Infolgedessen gelten ausschließlich Renditen, die unter ein benutzerspezifisches Ziel oder eine geforderte Rendite fallen, als riskant (Srivastava und Mazhar 2018). Infolgedessen wird die Sortino-Ratio als:

$$SoR_t = \frac{E(r_t - r_{Ziel})}{TDD_t}$$

$$mit\ TDD = \sqrt{\frac{1}{t}\sum_{i=0}^{t}(min(r_{Ziel}, r_i - r_{Ziel}))^2} \tag{16}$$

definiert. Die Sortino-Ratio wird analog zur Sharpe-Ratio berechnet, in dem der Erwartungswert der Differenz der Renditen durch die Standardabweichung der Differenz der Renditen geteilt wird. Die Variable r_{Ziel} bildet dabei eine Zielrendite ab und zusätzlich werden durch die Minimumfunktion lediglich die Standardabweichungen der Renditen betrachtet, welche unter die Zielrendite fallen.

3.3 Portfolio Rahmenbedingungen

Für die hier durchgeführte Untersuchung werden m Vermögenswerte betrachtet, die zusammen ein Portfolio bilden. Die Schlusskurse aller Vermögenswerte zu einem Zeitpunkt t bilden den Preisvektor v_t. Demnach wird der Schlusskurs des Vermögenswertes i zu einem Zeitpunkt t als $v_{i,t}$ dargestellt. Analog dazu bezeichnen v_t^{hi} und v_t^{lo} die höchsten und niedrigsten Preise der Vermögenswerte für einen Zeitraum t. Zudem ist anzumerken, dass durch das Miteinbeziehen von Bargeld als Vermögenswert, der erste Eintrag der drei Preisvektoren zu jedem Zeitpunkt den Wert 1 annimmt. Der

relative Preisvektor für die Handelsperiode t, y_t, ist definiert als die elementweise Division von v_t durch v_{t-1}:

$$y_t = v_t \oslash v_{t-1} = \left(1, \frac{v_{1,t}}{v_{1,t-1}}, \frac{v_{2,t}}{v_{2,t-1}}, \dots, \frac{v_{m,t}}{v_{m,t-1}}\right)^T. \tag{17}$$

Der preisrelative Vektor kann zur Berechnung der Veränderung des gesamten Portfoliowertes – ohne Transaktionskosten – in einer Periode verwendet werden:

$$p_t = p_{t-1} y_t \cdot w_{t-1}. \tag{18}$$

Dabei bildet w_{t-1} den Portfolio-Gewichtsvektor zu Beginn der Periode t ab und dessen Element $w_{t-1,i}$ beschreibt den Anteil des Vermögenswertes i am Portfolio p_{t-1}. Eine weitere Eigenschaft des Portfolio-Gewichtsvektors ist, dass sich die Elemente des Vektors für alle Zeitpunkte auf 1 aufsummieren.

Die Rendite des Portfolios für die Periode t wird durch:

$$r_t = \frac{p_t}{p_{t-1}} - 1 = \frac{p_t - p_{t-1}}{p_{t-1}} \tag{19}$$

berechnet. Der endgültige Portfoliowert in Abwesenheit von Transaktionskosten mit einem Anfangswert von p_0, wird anhand von:

$$p_f = p_0 \prod_{t=1}^{t_f+1} y_t \cdot w_{t-1} \tag{20}$$

berechnet. Aufgrund von Preisbewegungen auf dem Markt entwickeln sich die Gewichte am Ende desselben Zeitraums gemäß:

$$w'_t = \frac{y_t \odot w_{t-1}}{y_t \cdot w_{t-1}}. \tag{21}$$

Wobei $\odot$ die elemetweise Multiplikation darstellt. Bei Zahlung aller Provisionen, die durch die Entscheidungen des Portfoliomanagers entstehen, relevante Vermögenswerte zu kaufen und zu verkaufen, schrumpft der Portfoliowert durch diese Umschichtungsaktion um den Faktor μ_t:

$$p_t = \mu_t p'_t \quad \text{mit } \mu_t \in (0, 1]. \tag{22}$$

Um den Parameter μ_t zu ermitteln, wird der rekursive Lösungsansatz von Jiang et al. (2017) verwendet, welcher eine Erweiterung der von Ormos und Urbán (2013) vorgestellten Arbeit darstellt. Dieser ermöglicht, den Transaktionsrestfaktor μ_t mit beliebiger Genauigkeit zu approximieren. Demzufolge wird der Parameter μ_t durch:

$$\mu_t = \frac{1}{1 - c_t w_{t,0}} \left[1 - c_t w'_{t,0} - (c_s + c_p - c_s c_p) \sum_{i=1}^{m} (w'_{t,i} - \mu_t w_{t,i})^+ \right] \tag{23}$$

$$mit\ (x)^+ = ReLu(x) = \begin{cases} x, & f\ddot{u}r\ x > 0 \\ 0, & f\ddot{u}r\ x < 0 \end{cases}$$

ermittelt. Dabei bilden c_s und c_p die Transaktionskosten für den Verkauf und den Erwerb von Vermögenswerten ab. Um die Konvergenz des approximierten Parameters für den Fall, dass die Transaktionskosten für den Verkauf und den Erwerb übereinstimmen, zu erhöhen wird der von Moody und Saffell (1998) vorgestellte Lösungsansatz verwendet und somit der erste Schätzwert für μ_t als:

$$\mu_t = c \sum_{i=1}^{m} |w'_{t,i} - w_{t,i}| \tag{24}$$

berechnet. Demnach wird die rekursive Annäherung des Schätzwertes unterbrochen, sobald die Differenz des vorherigen Schätzwertes zu dem jetzigen Schätzwert unter einen vorher definierten Wert fällt.

Der Handlungsverlauf wird in Abbildung 2 für zwei Perioden veranschaulicht. Diese beschreibt, wie die Preisveränderungen der Vermögenswerte in Periode t, die durch den Vektor $\overrightarrow{y_t}$ dargestellt werden, den Portfoliowert p_{t-1} und die Portfoliogewichte w_{t-1} verändern und in einem neuen Portfoliowert p'_t und den veränderten Portfoliogewichten w'_t resultieren.

Am Ende der Periode t werden die Entscheidungen des Portfoliomanagers umgesetzt, welche zu einem Kauf und Verkauf von Vermögenswerten führen. Diese Umverteilung der Anlagewerte wird in der Veränderung von w'_t zu w_t dargestellt und beschreibt die Portfoliogewichte zum Anfang der darauffolgenden Periode $\mathrm{t} + 1$. Zusätzlich führt diese Umverteilung in Gegenwart von Transaktionskosten zu einer Verringerung des Portfolios in Höhe von μ_t. Demzufolge wird der Portfoliowert zu Beginn der Periode $\mathrm{t} + 1$ durch p_t abgebildet. Anschließend werden durch weitere Preisveränderungen auf den Märkten in der neuen Periode erneut die Portfoliogewichte w_t und der Portfoliowert p_t verändert.

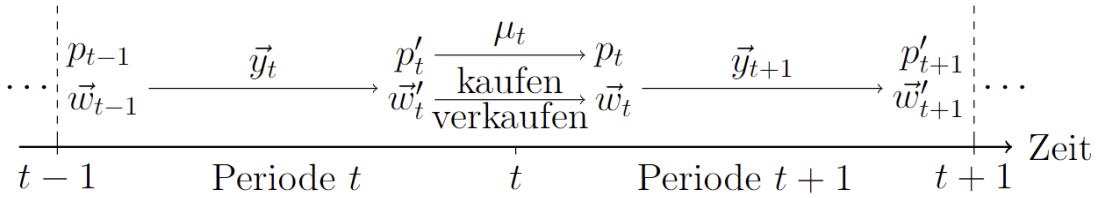

Abbildung 2: Handelsverlauf in Anlehnung an Jiang et al. (2017)

3.4 Markthypothesen

In Übereinstimmung mit Jiang et al. (2017) und Yashaswi (2021) werden zwei Annahmen bzgl. des betrachteten Markt getroffen:

1. Zero-Slippage: Aufgrund der hohen Liquidität aller betrachteten Marktwerte kann jeder Handelsauftrag sofort zum letzten Preis ausgeführt werden, wenn ein Auftrag erteilt wird.

2. Zero-Market-Impact: Das von dem eingesetzten Agenten investierte Vermögen ist zu gering, um einen Einfluss auf den Markt zu besitzen.

Diese getroffenen Annahmen entsprechen der Realität einer realen Handelsumgebung, solange das Handelsvolumen für die betrachteten Märkte angemessene Maße erreicht.

3.5 Evaluierung der Leistung

Um die Leistungsfähigkeit der jeweiligen Modelle in Bezug auf die Testdaten zu bewerten und zu beurteilen, wird eine Auswahl an unterschiedlichen Metriken verwendet. In Anlehnung an Shi et al. (2019) wird der finale Portfoliowert (fPW) verwendet. Shi et al. begründen dabei die Verwendung des fPWs damit, dass das Übergeordnete Ziel des Portfoliomanagements die Erzielung von Gewinnen ist. Demnach spiegelt der endgültige Portfoliowert die Leistung der gewählten Portfoliomanagementstrategie ausreichend ab. So wird der fPW folgendermaßen berechnet:

$$fPW = \frac{p_f}{p_0}. \tag{25}$$

Der fPW ergibt sich demnach aus der Division des Endwertes des Portfolios p_f durch den initialen Portfoliowert p_0. Ebenfalls in Anlehnung an Shi et al. wird die Sharpe-Ratio als eine weitere Leistungsmetrik in die Bewertung mitaufgenommen. Die Verwendung der Sortino-Ratio zur Leistungsevaluierung der Modelle ist durch die Untersuchung von Yu

et al. (2019) begründet. Diese wenden ebenfalls die Sortino-Ratio zur Bewertung der von ihnen betrachteten Portfolios an.

Weitere Metriken, die in die Bewertung einbezogen werden, ist die Calmar-Ratio und der MDD (Magdon-Ismail et al. 2004), welche ebenfalls in der Analyse der Portfolioperformance von Wang et al. (2021) verwendet werden. MDD beschreibt dabei den größten Verlust – von einem Höhepunkt zu einem Tiefstand – über den gesamten betrachtet Zeitraum. Die MDD wird als:

$$MDD = \max_{\tau}(\max_{t<\tau}\frac{p_t - p_\tau}{p_t}) \tag{26}$$

definiert. Die Calmar-Ratio wird durch:

$$Calmar = \frac{E(r_t - r_{Ziel})}{MDD} \tag{27}$$

berechnet und gleicht in ihrer Berechnung der Sharpe- und Sortino-Ratio. Ein Unterschied ergibt sich jedoch dadurch, dass die MDD-Metrik die verschiedenen Varianzberechnungen ersetzt.

Um die mit der Portfoliooptimierung verbundenen Risiken ausführlicher zu untersuchen, werden analog zu Yu et al. (2019) zwei weitere Risikobewertungskriterien – Value-at-Risk (VaR) und Conditional-Value-at-Risk (CVaR) – verwendet. Ersteres misst und kontrolliert das Risiko in Bezug auf die Perzentile der Verlustverteilung, während letzteres das Tail-Risiko eines Portfolios schätzt. Linsmeier und Pearson (2000) beschreiben VaR als eine einzelne, zusammenfassende statistisches Kennzahl, um mögliche Portfolioverluste darzustellen. Bei einer Wahrscheinlichkeit von p Prozent und einer Haltedauer von t Tagen ist der VaR eines Vermögenswertes der Verlust, der mit einer Wahrscheinlichkeit von p Prozent während der nächsten Haltedauer von t Tagen überschritten wird. Vereinfacht ausgedrückt ist dies der Verlust, der während p Prozent der betrachteten Haltedauer voraussichtlich überschritten wird. Demnach wird die VaR (Hakwa et al. 2015) durch:

$$\mathrm{VaR}_\alpha(X) = inf\{l \in \mathbb{R}: \Pr(L \leq l) \geq \alpha\} \tag{28}$$

berechnet. Die VaR eines Portfolios mit einem Konfidenzniveau $\alpha \in (0,1)$ ist durch die kleinste Zahl l gegeben, so dass die Wahrscheinlichkeit, dass der Verlust L kleiner gleich l ist, größer gleich α ist.

Ein alternatives Risikomaß zum VaR ist die CVaR (Alexander et al. 2006). Die CVaR für ein gewähltes Niveau α ist die erwartete Rendite eines Vermögenswertes X im schlechtesten (1 - α) Prozent der Fälle und wird durch:

$$CVaR_{\alpha}(X) = E[X|X \geq VaR_{\alpha}(X)] \tag{29}$$

berechnet.

3.6 Benchmark Portfolios

Um die Leistungsfähigkeit der drei Reinforcement-Learning-Portfolios zu vergleichen, wird eine Auswahl unterschiedlicher Portfolios in die Analyse der Ergebnisse aufgenommen. Demnach wird analog zu Wang et al. (2019) das einheitliche Buy-and-Hold (BAH)-Portfolio als ein Referenzwert für die in dieser Arbeit konzipierten Portfolios verwendet. Für dieses Portfolio wird zu Beginn des Investitionszeitraumes das gesamte Vermögen gleichmäßig auf alle Anlagen verteilt (Borodin et al. 2004).

Die Integration des konstanten einheitlichen CRP-Portfolios (Cover 1991) ist durch Weng et al. (2020) motiviert, die es ebenfalls zur Evaluierung der Leistung des von ihnen konzipierten Portfolios in ihre Analyse aufnehmen. Dabei unterscheidet sich das konstante einheitliche BAH-Portfolio vom CRP-Portfolio dadurch, dass es die einheitliche Gewichtung der betrachteten Vermögenswerte im Portfolio über den gesamten Zeitraum konstant hält (Borodin et al. 2004).

Die Aufnahme des Cryptocurrency Index (CRIX) in die Untersuchung geschieht in Anlehnung an Brauneis und Mestel (2019), die diesen ebenfalls (Trimborn und Härdle 2018) für die Analyse der betrachteten Portfolios in ihrer Untersuchung verwendeten. Der CRIX ist ein Index, der auf der Indexierung der Marktkapitalisierung basiert und demnach größere Gewichtung auf Kryptowährungen in Bezug auf ihre Marktkapitalisierung legt. Er erfasst die breite Kryptowährungs-Marktbewegung mit einer statistisch optimierten, variierenden Anzahl von Komponenten (Petukhina et al. 2021).

Angelehnt an Shi et al. (2019) wird zum Vergleich der betrachteten Portfolios die PAMR-Strategie (Li et al. 2012) in die Untersuchung einbezogen. Die Hauptidee von PAMR besteht darin, eine Verlustfunktion zu entwerfen, die die Mean-Reversion-Eigenschaft widerspiegelt. Die Verlustfunktion wird somit berechnet durch:

$$\ell_{\epsilon}(\mathrm{w};\mathrm{x}_t) = \begin{cases} 0 & \mathrm{w}\cdot\mathrm{x}_t \leq \epsilon \\ \mathrm{w}\cdot\mathrm{x}_t - \epsilon & \text{sonst} \end{cases}$$
$$mit\ 0 \leq \epsilon \leq 1. \tag{30}$$

Demnach steigt der Wert der Verlustfunktion linear an, wenn die erwartete Rendite auf Grundlage des letzten relativen Kurses größer als ein vorher bestimmter Schwellenwert ϵ ist. Ist dies nicht gegeben, nimmt die Verlustfunktion den Wert null an. Dabei bildet $\mathbf{x}_t$ die relativen Preisänderungen der Schlusspreise ab, die durch Division der Schlusspreise der jetzigen Periode durch die Schlusspreise der vorherigen Periode berechnet werden. Für den Fall, dass die Verlustfunktion den Wert null annimmt, wird die vorherige Portfolioallokation beibehalten. Für den Fall, dass die Verlustfunktion nicht einen Wert von null annimmt, werden die neuen Portfoliogewichte durch:

$$\mathrm{w}_{t+1} = \arg\min_{\mathbf{w}\in\Delta_m} \frac{1}{2}\|\mathrm{w} - \mathrm{w}_t\|^2 \text{ s.t. } \ell_\epsilon(\mathrm{w}; \mathrm{x}_t) = 0 \tag{31}$$

bestimmt. Dieses Optimierungsproblem wird anschließend durch:

$$\begin{gathered} \mathrm{w}_{t+1} = \mathrm{w}_t - \tau_t(\mathrm{x}_t - \bar{x}_t 1) \\ mit\ \tau_t = max\left\{0, \frac{\mathrm{w}_t \cdot \mathrm{x}_t - \epsilon}{\|\mathrm{x}_t - \bar{x}_t \mathbf{1}\|^2}\right\} \end{gathered} \tag{32}$$

gelöst. In einem finalen Schritt werden die Gewichte anhand eines Simplex-Projektionsschrittes (Duchi et al. 2008) transformiert, um eine Nichtnegativitätsbeschränkung des Portfolioallokationsvektors zu erfüllen.

Eine weitere Methode, die analog zu Shi et al. (2019) zum Vergleich der Ergebnisse in die Untersuchung aufgenommen wird, ist die des Exponentiated-Gradients (EG) (Helmbold et al. 1998). Die EG-Methode wählt einen Portfolio-Gewichtungsvektor, der darauf abzielt, die Gesamtbelohnung in der nächsten Periode zu maximieren, während große Veränderungen des Portfoliovektors, gemessen an der relativen Entropie zwischen den gewählten Gewichten für die nächster Periode und den Gewichten der vorherigen Periode, penalisiert werden:

$$\begin{gathered} \underset{\boldsymbol{w}_{t+1}}{\operatorname{argmax}}\, \eta \log\left(w_{t+1} \cdot x_t\right) - D_{RE}(w_{t+1} \,||\, w_t) \\ mit\ D_{RE}(w_{t+1} \,||\, w_t) \;=\; \sum_{i=1}^{N} w_{t+1}\, log \frac{w_{t+1}}{w_t}. \end{gathered} \tag{33}$$

Dabei stellt der Parameter η ($\eta > 0$) welcher als Lernrate bezeichnet wird, die relative Bedeutung der beiden Terme zueinander dar. Eine Näherungslösung für die obige Optimierung führt zu der folgenden Formel für die Aktualisierung der Gewichte, welche als:

$$w_{t+1}^{i} = \frac{w_t^i \exp\left(\eta x_t^i / w_t \cdot x_t\right)}{\sum_{j=1}^{N} w_t^j \exp\left(\eta x_t^j / w_t \cdot x_t\right)} \tag{34}$$

dargestellt wird. Eine einfache Erläuterung der Optimierungsmethode besteht darin, die Aktie mit der besten Performance in der letzten Periode zu erwerben, jedoch das neue Portfolio in der identischen Gewichtung des vorherigen Portfolios zu halten (Li und Hoi 2014).

3.7 Deep Learning

Ein tiefes neuronales Netz ist durch eine Abfolge mehrerer Verarbeitungsschichten gekennzeichnet. Jede Schicht besteht aus einer nichtlinearen Transformation und die Abfolge dieser Transformationen führt zum Erlernen verschiedener Abstraktionsebenen (Olah et al. 2017; D. Erhan et al. 2009). Zudem argumentieren Goodfellow et al. (2016), dass sich Deep-Learning-Techniken sich bestens für die Extraktion komplexer Muster aus großen Datensätzen eigenen. Des Weiteren bekräftigen sie, dass tiefe Feedforward-Netzwerke, auch Feedforward-Neural-Networks oder Multi-Layer-Perceptrons (MLP) genannt, die Quintessenz der Deep-Learning-Modelle darstellen. Feedforward Neural Networks sind vollständig verbunden, d.h. stellen sie eine Netzwerkklasse dar, in der jedes Neuron aus einer Schicht mit allen Neuronen der folgenden Schicht verbunden ist. Ein Neuron ist dabei am besten als eine Funktion zu verstehen (Goldberg 2015).

Ein solches Netzwerk ist in Abbildung 3 abgebildet. Die einzelnen Neuronen der verborgenen Schicht basieren auf drei aufeinanderfolgenden mathematischen Operationen. Zunächst wird eine gewichtete Summe aller Ausgänge (Aktivierungen) aus der vorherigen Schicht berechnet, wobei das Gewicht, das einer dieser Aktivierungen entspricht, durch seine Verbindung zum betreffenden Neuron repräsentiert wird. Dabei bezeichnet (L) die Schicht und m die Position des Neurons in der jeweiligen Schicht. Nach Hinzufügen einer – als Bias bezeichneten – Konstante $b_k^{(L)}$ zur gewichteten Summe wird der resultierende Ausdruck $z_k^{(L)}$ einer nichtlinearen Funktion $f(.)$ zugeführt. So werden die M Aktivierungen für die Neuronen der Schicht (L) $a_0^{(L)}, \dots, a_M^{(L)}$

aus den J Aktivierungen der vorherigen Schicht $a_0^{(L)}, \dots, a_J^{(L)}$ durch:

$$\begin{bmatrix} a_0^{(L)} \\ \vdots \\ a_k^{(L)} \\ \vdots \\ a_K^{(L)} \end{bmatrix} = f\left(\begin{bmatrix} w_{0,0} & \cdots & w_{0,j} & \cdots & w_{0,J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{k,0} & \cdots & w_{k,j} & \cdots & w_{k,J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{K,0} & \cdots & w_{K,j} & \cdots & w_{K,J} \end{bmatrix} \begin{bmatrix} a_0^{(L-1)} \\ \vdots \\ a_j^{(L-1)} \\ \vdots \\ a_J^{(L-1)} \end{bmatrix} + \begin{bmatrix} b_0^{(L)} \\ \vdots \\ b_k^{(L)} \\ \vdots \\ b_K^{(L)} \end{bmatrix} \right) \tag{35}$$

berechnet. Dies kann vereinfachter dargestellt werden durch:

$$\mathbf{a}^{(L)} = f\left(\mathbf{z}^{(L)}\right)$$
$$mit\ \mathbf{z}^{(L)} = \mathbf{W}^{(L)}\mathbf{a}^{(L-1)} + \mathbf{b}^{(L)}. \tag{36}$$

Um eine Prognoseaufgabe zu erfüllen, wird das Netz mit Beobachtungen – zusammen mit ihren entsprechenden Antworten – aus einem Trainingsdatensatz versorgt. Eine Zielfunktion wird verwendet, um die Differenz zwischen den Vorhersagen des Modells und den tatsächlichen Ergebnissen zu messen. Hauptziel ist, diese Differenz oder den Verlust zu minimieren (LeCun et al. 2015). Die Modellparameter, die geändert werden, um diese Aufgabe zu erfüllen, sind die Gewichte und Verzerrungen des Netzwerks (Goldberg 2015). Die dabei angewandte numerische Methode wird als Gradientenabstieg bezeichnet. Dieser läuft auf die Berechnung des negativen Gradienten der Kostenfunktion hinaus, der die Richtung des steilsten Abstiegs im mehrdimensionalen Vektorraum, welcher durch die Gewichte und Bias aufgespannt wird. Jeder Term im Gradientenvektor enthält eine Information darüber, wie die entsprechende Gewichtung oder Verzerrung zu ändern ist, um den Verlust so direkt wie möglich zu verringern: Der Absolutwert eines Terms gibt Aufschluss über die Größe, das Vorzeichen über die Richtung der Änderung (LeCun et al. 2015).

Der Gradientenvektor, anhand dessen ein neuronales Netzwerk ‚lernt', eine gegebene Eingabe mit der entsprechenden Ausgabe in Einklang zu bringen, wird anhand des Backpropagation-Algorithmus (Rumelhart et al. 1986) berechnet. Ausgangspunkt der Backpropagation sind die Vorhersagen des Modells und eine Aufzeichnung darüber, wie diese verändert werden sollen, um den Verlust gemäß der Kostenfunktion zu verringern (Goldberg 2015).

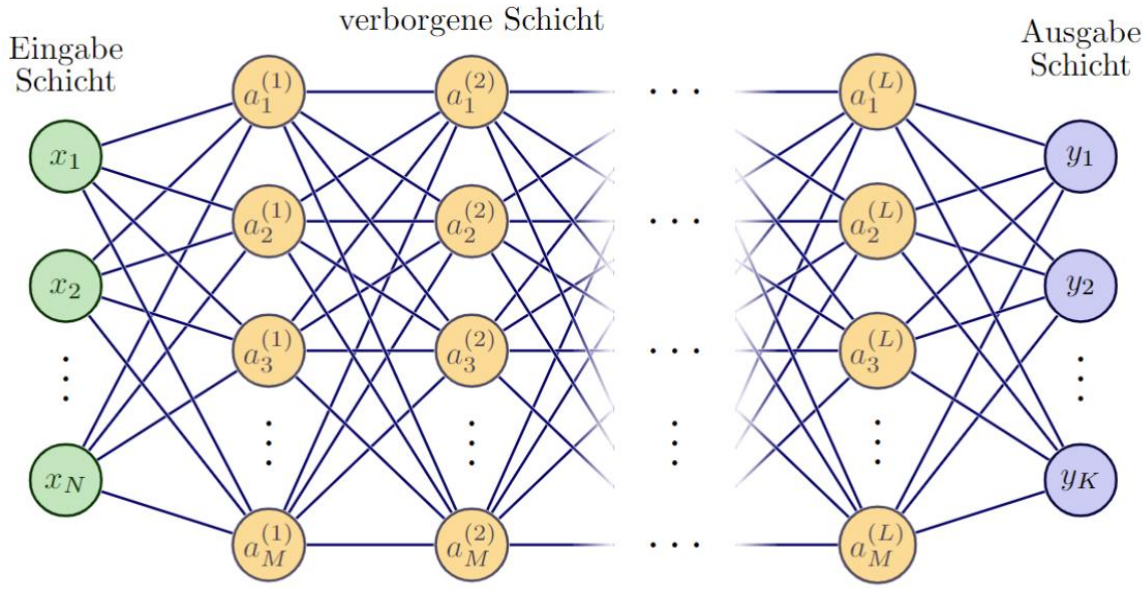

Abbildung 3: Neuronales Netzwerk eigene Darstellung

3.8 Netzwerkarchitektur

Das in dieser Arbeit verwendete Netzwerk besteht aus einem MLP-Netzwerk, welches den Vektor s_t als Input verwendet, der eine Dimension von 201 besitzt. Der Agent wählt seine Aktion über eine stochastische Politik $\pi\theta$, wobei θ die Parameter eines tiefen neuronalen Netzwerkes darstellen. Dieses Netzwerk besteht aus zwei Teilen, einem Policynetz, dass eine Aktionsverteilung erzeugt, und ein Kritiker-Netz, das diskontierte zukünftige Erträge errechnet. Demnach wird analog zu Liang et al. (2018) ein Netzwerk gebildet, welches das Policy- und Kritiker-Netzwerk vereint. So besteht das hier verwendete Netzwerk aus einer geteilten Netzwerkkomponente, die sich anschließend in zwei unterschiedliche Bestandteile des Netzwerkes aufspaltet. Diese beiden Bestandteile repräsentieren wiederum jeweils das Policy- oder Akteur-Netzwerk und Kritiker-Netzwerk. Eine Repräsentation des Netzwerkes ist in Abbildung 4 dargestellt.

Das hier verwendete Netzwerk besteht zudem aus einem wiederverwendeten Netzwerkblock. Dieser besteht aus 5 Schichten mit jeweils 201 Neuronen, welche mit einer Softplus-Aktivierungsfunktion (Hao Zheng et al. 2015) ausgestattet sind. Dieser Netzwerkblock wird zum einen als Hauptbestandteil im geteilten Netzwerk verwendet, das sich anschließend in zwei weitere Komponente aufteilt. Zum anderen wird der Netzwerkblock ebenfalls im Policy- und Kritikernetzwerk verwendet und nimmt dort eine zentrale Rolle ein.

Demnach wird der Inputvektor s_t von einer Netzwerkschicht mit übereinstimmender Dimension aufgegriffen und ohne Aktivierungsfunktion an den ersten Netzwerkblock, der gleichermaßen durch den Kritiker und den Akteur verwendet wird, weitergeleitet. In einem weiteren Schritt wird anschließend der Output des ersten Blocks and den Akteur und den Kritiker weitergeleitet, welche ebenfalls jeweils über einen identischen Netzwerkblock verfügen. Der Kritiker wird anschließend mit zwei weiteren Netzwerkschichten ausgestattet, die mit einer Rectified-Linear-Unit (ReLu)-Aktivierungsfunktion verbunden sind (Agarap 2018):

$$ReLu(x) = x^+ = max(0, x). \tag{37}$$

Die ReLu-Aktivierungsfunktion ist linear für Eingabewerte größer null.

Angelehnt an Holubar und Wiering (2020) erhält das Policynetz zwei Ausgangssignale, die den Parametern μ und σ einer Normalverteilung entsprechen. Demnach werden die Handlungen, die die Portfoliogewichte widerspiegeln, aus dieser Verteilung stichprobenartig ausgewählt. Um die beiden Parameter zu generieren, wird der Output des Kritikernetzwerkblocks aufgeteilt und für den Parameter μ von einer weiteren

neuronalen Schicht aufgegriffen, die anschließend einen Outputvektor mit 51 Werten generiert. Demgegenüber wird für den Parameter σ der Output des Kritikernetzwerkblocks durch eine neuronale Schicht aufgegriffen, welche ebenfalls einen Outputvektor mit 51 Werten generiert. In einem zusätzlichen Schritt wird jedoch der Outputvektor durch eine Leaky-ReLu-Aktivierungsfunktion transformiert (Maas et al. 2013) :

$$LeakyReLu(x) = \begin{cases} x & \text{für } x \geq 0 \\ \alpha x & \text{sonst} \end{cases} \tag{38}$$

Dabei ist α ein Parameter, der die Steigung der Geraden im negativen Bereich darstellt und innerhalb dieser Untersuchung – analog zu Maas et al. – einen Wert von 0.01 annimmt. In einem letzten Schritt werden diese Werte exponenziert, um ausschließlich positive Werte für die Standardabweichung der Normalverteilung zu erhalten.

Die kontinuierlichen Aktionen, die der Agent wählt, werden infolgedessen aus der – durch das Netzwerk parametrisierten – Normalverteilung bestimmt. In einem weiteren Schritt werden die Stichproben der Normalverteilung durch eine Softmax-Funktion transformiert (Bridle 1989):

$$Softmax(x) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{39}$$

Demzufolge wird jedes exponenzierte Element des Vektors x durch die Summe der gesamten exponenzierten Elemente des Vektors dividiert. Dieser finale Schritt wird getätigt, um die Elemente des ausgewählten Portfolioallokationsvektors auf eine Reichweite von 0 bis 1 zu beschränken sowie sicherzustellen, dass die Elemente des Portfolioallokationsvektors sich auf 1 aufsummieren. Demnach werden für diese Untersuchung – in Anlehnung an Jiang et al. (2017), Zhang et al. (2020) und Petukhina et al. (2021) – ausschließlich positive Portfoliogewichte gebildet und somit keine Short-Positionen zugelassen.

Ein gesamter Ablauf des Algorithmus während des Trainings des Netzwerkes wird in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt. Dabei werden – a nalog zu Meng et al. (2020) – Mini Batches während der Trainingsphase des Netzwerks verwendet. In Anlehnung an Andrychowicz et al. (2020) werden die Advantage-Werte auf der Mini-Batch Ebene normalisiert. Die Parameter des Netzwerkes werden anhand des Adam-Optimierers (Kingma und Ba 2014) optimiert.

Motiviert durch die Ergebnisse der Untersuchungen von Zhang et al. (2020) und Jiang und Liang (2016) – wird ebenfalls die Regularisierungsmethode Dropout verwendet, um ein Overfitten des Netzwerks auf die Trainingsdaten zu verhindern. Dropout ist eine

Regularisierungsstrategie zum Trainieren eines Netzwerks von Teilnetzwerken durch zufälliges temporäres Entfernen von Einheiten ohne Output – mit einer vorher definierten Wahrscheinlichkeit – aus dem ursprünglichen Netzwerk.

Eine Zusammenfassung der in der Untersuchung verwendeten Parameter ist in Abbildung 6 abgebildet. Die drei Agenten werden demnach für 50 000 Epochen trainiert und anschließend aufgrund ihrer Performance evaluiert. Der PPO-Clip Parameter ϵ wird auf einen Wert von 0.4 festgelegt. Der GAE Koeffizient λ, der einen Kompromiss zwischen Verzerrung und Varianz der Advantage-Werte darstellt, wird in Einklang mit den Ergebnissen aus den umfangreichen Experimenten und den daraus gezogenen Schlussfolgerungen von Andrychowicz et al. (2020) auf einen Wert von 0.95 gesetzt. Der Value-Koeffizient, der den Einfluss des Kritikers auf die Verlustfunktion beeinflusst, wird auf einen Wert von 0.9 festgelegt. Die Parameter r_{Ziel} und r_f, die in der Berechnung für das Sharpe- und das Sortino-Ratio und somit ebenfalls für die Rewardfunktion für zwei der Agenten sowie für die Evaluierung der Portfolios verwendet werden, nehmen innerhalb dieser Arbeit – in Einklang mit Leem und Kim (2020) – jeweils einen Wert von 0 an. Die Lernrate des Adam-Optimierers, der die Parameter des Netzwerkes optimiert, wird für das geteilte Netzwerk sowie für den Kritiker auf einen Wert von 0.003 festgelegt, demgegenüber nimmt die Lernrate des Akteurs einen Wert von 0.001 an. Die Transaktionskosten, die durch das Handeln von Vermögenswerten auf dem Kryptomarkt entstehen, werden innerhalb dieser Untersuchung – in Anlehnung an Li et al. (2018) – auf 0.5 % festgelegt. Das Konfidenzinterwall für die VaR und die cVaR wird analog zu Yu et al. (2019) jeweils auf einen Wert von 0.95 festgelegt. Für die Trainings- und Testläufe der betrachteten Portfolios, wird das Startkapital auf einen Wert von Tausend USD festgelegt.

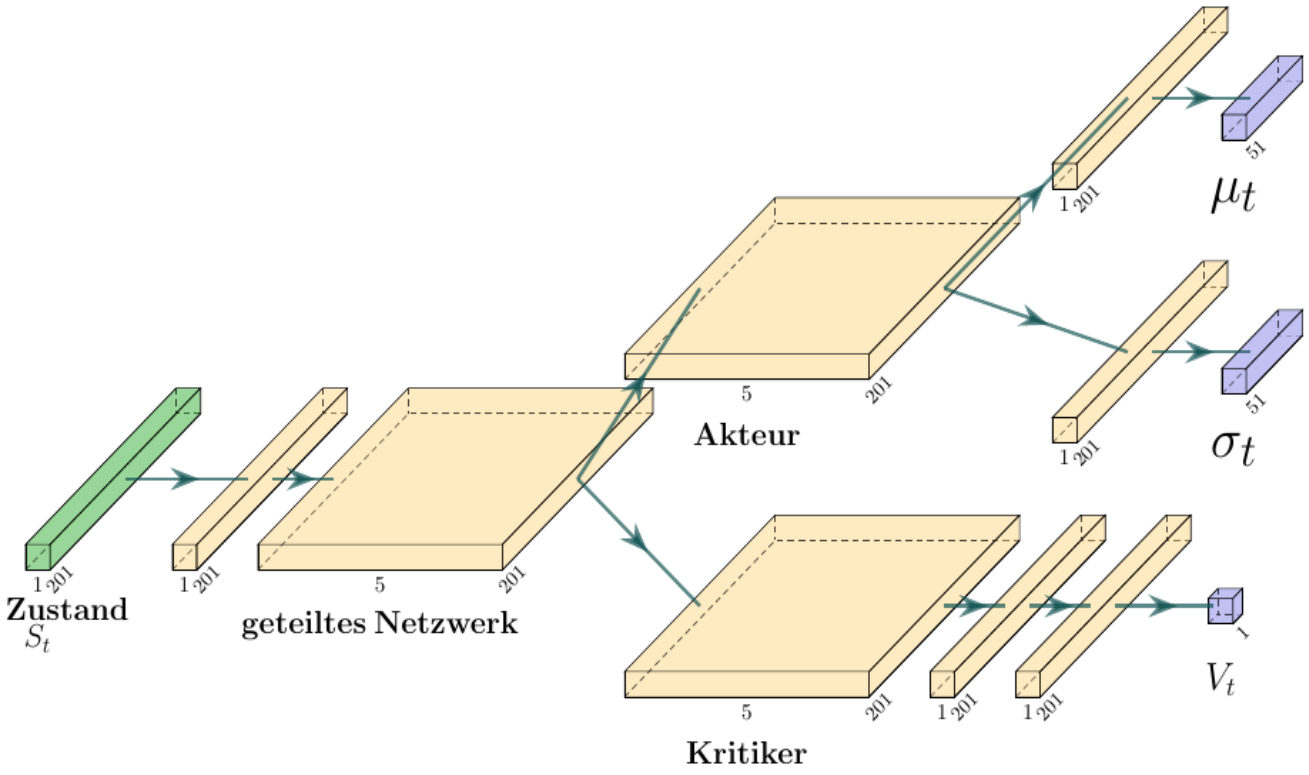

Abbildung 4: Netzwerkarchitektur eigene Darstellung

Algorithm 1 PPO-Clip

for episode $= 1, 2, \ldots, M$ **do**
  for step $= 1, 2, \ldots, S$ **do**
    Wähle Aktion a_t aus welche durch die Policy $\pi(A_t \mid S_t; \theta_{alt})$ bestimmt wird.
    mit $\pi_{\theta alt} \sim N(\mu(S_t), \sigma(S_t))$ und $\sigma : diag(\sigma_1, \sigma_2, \cdots, \sigma_{51})$.
    Aufzeichnung der Übergänge $(S_t, A_t, \log \pi(A_t \mid S_t; \theta_{alt}), R_t, V_t)$ in $\mathcal{B}$
  end for
  anhand des Speichers $\mathcal{B}$,berechnen von
  $G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{S} \gamma^k R_{t+k+1}$
  und $\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V$
  mit $\delta_t^V = R_t + V(S_{t+1}) - V(S_t)$
  for step $= 1, 2, \ldots, MB$ in Mini-Batch **do**
    Normalisierung der Advantage Werte $\hat{A}_t^{Norm} = \frac{\hat{A}_t - E_t(\hat{A})}{\sigma(\hat{A})}$
    Berechnen der log Wahrscheinlichkeiten $\log \pi(a_t \mid s_t; \theta)$ und State Values $V(S_t, \theta)$
    Berechnung von $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{alt}}}(a_t|s_t)} = \exp(\log \pi(a_t \mid s_t; \theta) - \log \pi(a_t \mid s_t; \theta_{alt}))$
    Berechnung von $surr_1 = r_t(\theta) \times \hat{A}_t^{Norm}$
    Berechnung von $surr_2 = clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \times \hat{A}_t^{Norm}$
    Berechnung von $L_{Actor} = E_t[\min(surr_1, surr_2)]$
    Berechnung von $L_{Kritiker} = E_t[(V(S_t, \theta) - G_t)^2]$
    Berechnung des gesamten Verlustes $L = L_{Actor} + c_1 L_{Kritiker}$
    Aktualisierung des Netzwerkparameters θ anhand des Adam Optimierungs-verfahrens
  end for
  Aktualisierung von θ zu θ_{alt}
  Zurücksetzen des Speichers $\mathcal{B}$
end for

Abbildung 5: PPO-Clip Algorithmus eigene Darstellung

Parameter	
Anzahl Epochen (M)	50.000
Gamma (γ)	0.99
PPO Clip Parameter (ϵ)	0.4
GAE-Lambda (λ)	0.95
Value Koeffizient (c_1)	0.9
Mini-Batch Anzahl (MB)	4
Ziel Rendite (r_{Ziel})	0
Risikofreie Rendite (r_f)	0
Lernrate für Akteur	0.0001
Lernrate für Kritiker	0.0003
Transaktionskosten für den Verkauf (c_s)	0.5%
Transaktionskosten für den Erwerb (c_p)	0.5%
EG (η)	0.05
PAMR (ϵ)	0.5
Value-at-Risk (α)	0.95
cVaR (α)	0.95

Abbildung 6: Verwendete Parameter eigene Darstellung

4 Daten

Für die hier vorgestellte Untersuchung wird der Kryptowährungsmarkt betrachtet. Kryptowährungen sind dezentralisierte elektronische Finanzanlagen, die als Alternative zu Fiat-Währungen entstanden sind (Nakamoto 2008). Kryptowährungen nutzen die Blockchain-Technologie, um Dezentralität, Transparenz und Unveränderlichkeit zu erreichen (Meunier 2018). Eines der wichtigsten Merkmale von Kryptowährungen ist der Ausschluss von Finanzinstituten als Vermittler (Harwick 2014) sowie der Tatsache, dass diese von keiner zentralen Behörde kontrolliert werden (Rose 2015): Die dezentralisierte Natur der Blockchain stellt sicher, dass Kryptowährungen theoretisch immun gegen staatliche Kontrolle und Einmischung sind. Die erste Kryptowährung, Bitcoin, entstand 2008 und wurde bald darauf von einer großen Anzahl an unterschiedlichen Coins gefolgt, die heterogene Merkmale aufweisen und einen Teil der Funktionalitäten von Bitcoin verbessern. Wichtige Beispiele sind die Ethereum-Plattform (Buterin 2014), auf der Smart-Contracts-Funktionen eingeführt wurden, sowie Monero (van Saberhagen 2013) und ZCash (Ben Sasson et al. 2014), die verbesserte Anonymisierungsverfahren beinhalten.

In Anlehnung an Brauneis und Mestel (2019), die in ihrer Untersuchung ebenfalls den Kryptowährungsmarkt analysiert haben, werden die täglichen Open High Low Close (OHLC)-Preisdaten der Webseite Coinmarketcap[1] verwendet. Diese Website sammelt Kryptowährungsdaten von mehreren Börsen weltweit und veröffentlicht volumengewichtete Preise (Brauneis und Mestel 2019). Die Preise der jeweiligen Vermögenswerte werden in einem Öffnungs-, Höhst-, Tiefst- und Schlusspreis in täglicher Frequenz dargestellt und in USD angegeben. Die Daten werden anhand eines Webscrapers aus der API[2] für historische Preisdaten generiert.

Um ein Survival-Bias der selektionierten Vermögenswerte zu vermeiden, wird der Lösungsansatz von Jiang et al. (2017) verwendet. Dabei werden zu Beginn des Backtesting-Zeitraumes Coins mit der höchsten Marktkapitalisierung ausgewählt. Um dies zu erreichen, wird auf eine historische Zeitaufnahme der Webseite Coinmarketcap[3] zurückgegriffen, um ein historisches Ranking der Anlagewerte nach Marktkapitalisierung zu erhalten. Anschließend wurden die 100 höchstplatzierten Vermögenswerte aus der historischen Rangliste ausgewählt und die täglichen OHLC-Preisdaten seit Beginn der

[1] https://coinmarketcap.com/
[2] https://api.coinmarketcap.com/data-api/v3/cryptocurrency/historical
[3] https://coinmarketcap.com/historical

Auflistung des jeweiligen Vermögenswertes innerhalb der Webseite Coinmarketcap bis einschließlich zum 31.03.2022 gesammelt.

In einem weiteren Schritt werden alle Stablecoins für die weitere Verwendung ausgeschlossen. Dies sind Krypto-Vermögenswerte, deren Werte an Körbe von Fiat-Währungen oder Bargeldäquivalenten, bestehenden Finanzinstrumenten, physischen Vermögenswerten wie Rohstoffen sowie an Körbe anderer Krypto-Vermögenswerte gekoppelt sind (Bullmann et al. 2019). Die Entscheidung, Stablecoins auszuschließen, ist darin begründet, dass sie geringe bis keine Preisveränderungen aufweisen. Anschließend werden – übereinstimmend mit Petukhina et al. (2021) – ausschließlich Vermögenswerte für die Untersuchung betrachtet, die für den gesamten relevanten Zeitraum einen Wert aufweisen können. Somit werden nur Coins herangezogen, die bereits zum Start und bis zum Ende der Zeitspanne gehandelt wurden. Dies führt dazu, dass alle Coins, die nach dem Start der betrachteten Periode den Handel begonnen haben, und alle, die vor Ende der betrachteten Periode den Handel gestoppt haben, nicht in die Portfolioentscheidung einbezogen werden. Petukhina et al. argumentieren, dass durch dieses zusätzliche Kriterium zu Auswahl von Kryptowährungen ein Schwerpunkt auf robuste Coins gelegt wird, welche vor allem für Anleger von Interesse sind, die Investitionen in diese neue Anlageklasse in Betracht ziehen.

Abschließend werden die 50 Kryptowährungen ausgewählt, die zu Beginn des Testdatenzeitraumes die höchste Marktkapitalisierung aufweisen. Für die Untersuchung wird ein Zeitraum vom 01.01.2019 bis zum 31.03.2022 betrachtet. Der gesamte Datensatz beinhaltet 1186 tägliche Preisdaten für 50 Kryptowährungen. Die Modelle werden dabei anhand von Trainingsdaten, die die zwei Jahre vom 01.01.2019 bis zum 31.12.2020 umfassen, trainiert. Zur Evaluierung der Leistungsfähigkeit und Analyse der Modelle wird der Testdatensatz vom 01.01.2021 bis zum 31.03.2022 verwendet, welcher 15 Monate beinhaltet. Demnach entspricht die Aufteilung des Datensatzes auf die Test- und Trainingsdaten ungefähr einem Verhältnis von 40 % zu 60 %.

Ein weiterer Datensatz, der für diese Arbeit verwendet wird, ist der CRIX-Index, der von der Webseite des S&P Dow Jones Indices (2022) herangezogen wurden. Dieser beinhaltet den täglichen Indexwert für einen Zeitraum vom 16.03.2018 bis zum 21.04.2022 und besitzt zu Beginn des Datensatzes einen Wert von 1000. Im Gegensatz zu den täglichen OHLC-Preisdaten besitzt der CRIX-Datensatz keine Beobachtungen für Wochentage wie Samstage und Sonntage. Da der Index innerhalb dieser Untersuchung als Benchmark für die drei konzipierten Portfolio fungiert, werden lediglich die Daten des Testzeitraumes vom 01.01.2021 bis zum 31.03.2022 verwendet, die in Abbildung 7 dargestellt sind. Demnach besitzt der Index zu Beginn des betrachteten Zeitraumes einen

Wert von ungefähr 2000, der zum Ende des Zeitraumes einen Wert von 4333 erreicht und sich somit mehr als verdoppelt. Ein Höchstwert von ungefähr 6430 wird zwischenzeitlich im September 2021 erreicht.

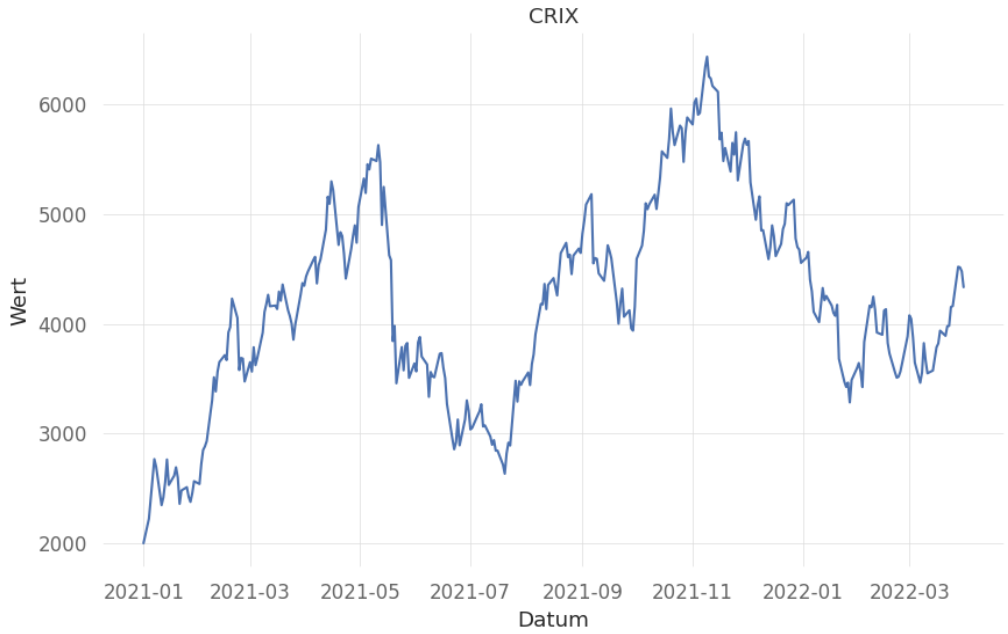

Abbildung 7: CRIX-Index von 2021 bis Anfang 2020 eigene Darstellung

5 Ergebnisse

Die Ergebnisse der unterschiedlichen Portfolios sowie der innerhalb dieser Arbeit konzipierten Portfolios sind in Abbildung 8 dargestellt. Die Leistungsfähigkeit der betrachteten Portfolios wurde auf Basis des Testdatensatzes für einen Zeitraum vom 01.01.2021 bis zum 31.03.2022 evaluiert.

In Bezug auf die Leistungsevaluierung der Portfolios durch die fPW-Metrik ist anzumerken, dass der Agent PPO-Profit von allen betrachteten Portfolios den höchsten Wert erreicht. Den zweithöchsten fPW erreicht das Portfolio, welches durch die PAMR Strategie erstellt wurde. Diese Ergebnisse können durch die Ergebnisse der Untersuchung von Weng et al. (2020) bestätigt werden. Demnach konnten Weng et al. innerhalb eines Backtestes feststellen, dass der von ihnen konzipierte Deep-Reinforcement-Learning Agent alle anderen miteinbezogenen Portfolios in Bezug auf den fPW überbieten konnte. Ebenso erreichte der PAMR-Portfolio-Ansatz den zweit-höchsten fPW-Wert. Zusätzlich ist anzumerken, dass das Portfolio, das durch den Agenten mit der Zielfunktion des Sortino-Wertes konzipiert wurde, den schlechtesten fPW-Wert der drei in dieser Untersuchung konzipierten Portfolios aufweist. Diese Ergebnisse stehen demnach in einem Konflikt zu den Ergebnissen von Leem und Kim (2020), da diese innerhalb ihrer Untersuchung feststellen konnten, dass der Agent mit einer Sortino-Rewardfunktion die Performance der anderen Agenten überbieten konnte.

Eine weitere Leistungsmetrik, in der das PPO-Profit-Portfolio einen hohen Wert erreicht, ist die der Sharpe-Ratio. Hierbei erreicht lediglich das EG-Portfolio einen höheren Wert. Der drittgrößte Sharpe-Wert der untersuchten Portfolios konnte für das PPO-Sharpe-Portfolio festgestellt werden. Die geringste Sharpe-Ratio der gesamten betrachteten Portfolios konnte für den CRIX-Index festgestellt werden, der damit eine geringere Sharpe Ratio aufweist als das CRP. Brauneis und Mestel (2019) konnten in ihrer Untersuchung auf dem Kryptowährungsmarkt ebenfalls feststellen, dass die von ihnen gebildeten CRPs einen höheren Sharpe-Wert aufweisen konnten als der CRIX.

In der von Jiang et al. (2017) durchgeführten Untersuchung konnten diese ebenfalls feststellen, dass die von ihnen entwickelten Deep-Reinforcement-Learning-Agenten für das Portfoliomanagement auf Kryptomärkten ebenfalls eine höhere Sharpe Ratio erreichen konnten als die zum Vergleich einbezogenen Portfolios. Dabei wurde von ihnen ebenfalls ein EG- und ein PAMR-Portfolio bei der Untersuchung genutzt. Aus ihren Ergebnissen geht ebenfalls hervor, dass das EG-Portfolio über die drei Backtesting-Zeiträume höhere Sharpe-Werte erreicht als das PAMR-Portfolio. Eine weitere Beobachtung, die mit den Ergebnissen von Jiang et al. übereinstimmt, ist der Umstand, dass das CRP ein höheres Sharpe-Ratio erreicht als das BAH-Portfolio. Anhand der

Leistungsevaluierung der Portfolios durch die Sortino-Ratio ist zu beobachten, dass auch der PPO-Profit-Agent einen Höchstwert erreicht. Zudem ist anzumerken, dass das PAMR-Portfolio einen besseren Sortino-Wert aufweist als das EG-Portfolio.

In Bezug auf die Calmar-Ratio kann ebenfalls festgestellt werden, dass das PPO-Profit-Portfolio den höchsten Wert erreicht. Die zweithöchste Calmar-Ratio kann für das PAMR-Portfolio gemessen werden. Analog zu diesen Ergebnissen wurde von Imajo et al. (2020) ebenfalls festgestellt, dass das von ihnen konzipierte Portfolio die höchste Calmar-Ratio erreicht. In der von Xu et al. (2020) vorgestellten Untersuchung stellen sie fest, dass das von ihnen konzipierte DRL-Portfolio die Leistungsfähigkeit – gemessen an der Calmar Ratio – der zum Vergleich herangezogenen Portfolios übersteigt. Dieses Ergebnis haben sie anhand von zwei Untersuchungen auf Kryptowährungsmärkten und einem Experiment auf dem amerikanischen Aktienmarkt ermittelt.

Die beste Performance der betrachteten Portfolios – gemessen an dem MDD für den Trainingszeitraum – wurde durch den EG-Algorithmus erreicht. So weist das EG-Portfolio einen MDD von 45 % auf. Das PPO-Profit-Portfolio belegt mit einem MDD-Wert von 48 % den zweiten Platz. Demgegenüber besitzen PAMR- und BAH-Strategien die höchsten MDD–Werte – mit jeweils 88 % und 67 %. In den von Jiang et al. (2017) durchgeführten Backtest-Experimenten kommen sie auf Ergebnisse, die sich von den hier vorgestellten Ergebnissen unterscheiden. Ihre Untersuchungen zeigen, dass in zwei der drei durchgeführten Testzeiträumen das CRP den geringsten MDD aufweist. Des Weiteren weist die PAMR-Strategie in ihrer Untersuchung hohe MDD Werte auf.

Eine weitere Kennzahl, an der die Leistung der betrachteten Portfolios evaluiert und verglichen werden kann, ist der VaR. Aus den Ergebnissen der Experimente geht hervor, dass das PAMR-Portfolio für den betrachteten Zeitraum den höchsten VaR-Wert aufweist. Der zweitgrößte VaR-Wert wurde für das PPO-Profit-Portfolio festgestellt. Des Weiteren konnte der drittgrößte VaR-Wert der acht betrachteten Portfolios für das EG-Portfolio gemessen werden.

In dem betrachteten Zeitraum konnte ebenfalls beobachtet werden, dass das CRP-, BAH- und das PPO-Sortino-Portfolio den geringsten VaR-Wert mit 8 % aufweisen. Demnach besitzen das CRP-, das BAH- und das PPO-Sortino-Portfolio die gleiche VaR. Der zweit geringste VaR-Wert von 9 % konnte für das PPO-Sharpe-Portfolio gemessen werden. Für das PPO-Sharpe-Portfolio wurde ein VaR-Wert von 9 % ermittelt, welches den zweitgeringsten VaR der betrachteten Portfolios darstellt.

In der von Yu et al. (2019) anhand des amerikanischen Aktienmarktes durchgeführten Untersuchung konnten diese feststellen, das für den betrachteten Zeitraum vom

01.01.2017 bis zum 04.12.2018 das CRP den geringsten VaR aufweist. Demnach konnte in dieser Untersuchung ebenfalls festgestellt werden, dass das CRP einen der geringsten VaR-Werte aufweist.

Die Leistungsevaluierung der Portfolios durch die CVaR zeigt, dass die PAMR-Strategie den höchsten Wert erreicht. Demnach konnte innerhalb dieser Untersuchung ein CVaR für das PAMR-Portfolio von 63 % ermittelt werden. Der zweitgrößte CVaR-Wert wurde für das PPO-Profit-Portfolio festgestellt, welches für den betrachteten Zeitraum ein CVaR von 37 % aufweist. Demgegenüber konnte der geringste CVaR von 12 % für das CRP und das PPO-Sortino-Portfolio festgestellt werden. Der zweitgeringste CVaR-Wert konnte zudem ebenfalls für zwei Portfolios gemessen werden. Demnach besitzen das BAH und das PPO-Sharpe Portfolio einen CVaR von 13 % für den Trainingsdatenzeitraum.

Somit konnten die Ergebnisse, die von Yu et al. (2019) im Laufe ihrer Untersuchung präsentiert wurden, reproduziert werden, da die hier ermittelten Eine zusammengefasste Ergebnisdarstellung der verschiedenen Portfoliowerte wird in Abbildung 9 vorgenommen. Aus der Grafik geht hervor, dass alle Portfolios – mit Ausnahme des BAH-Portfolios – einen individuellen Höchstwert zu Beginn September 2021 erreichen. Eine weitere Beobachtung, die aus den Abbildungen gezogen werden kann, ist die des stark volatilen Verlaufes des PAMR-Portfolios. Demnach konnte dies einen viel stärkeren Wertzuwachs zu Beginn des Oktobers 2021 vorweisen als die anderen betrachteten Portfoliostrategien. Aus der Abbildung geht ebenfalls hervor, dass die einbezogenen Portfolios den allgemeinen Marktverlauf, gemessen am CRP- und BAH- Portfolio verfolgen.

Abschließend kann aus den hier vorgestellten Ergebnissen geschlussfolgert werden, dass einer der in dieser Arbeit konzipierten Deep-Reinforcement-Learning-Agenten für das Portfolio Management auf dem Kryptowährungsmarkt erfolgreich eingesetzt werden konnte.

	EG	PAMR	BAH	CRP	CRIX	PPO-Profit	PPO-Sharpe	PPO-Sortino
Start Period	2021-01-01	2021-01-01	2021-01-01	2021-01-01	2021-01-01	2021-01-01	2021-01-01	2021-01-01
End Period	2022-03-31	2022-03-31	2022-03-31	2022-03-31	2022-03-31	2022-03-31	2022-03-31	2022-03-31
fPW	1765.32	4212.52	6.41	22.56	2.33	207695.52	46.37	4.92
Sharpe	3.11	1.89	1.63	2.46	1.41	2.89	2.71	1.49
Sortino	8.47	11.17	2.4	3.86	2.14	12.52	4.79	2.2
Max Drawdown	0.45	0.88	0.67	0.51	0.65	0.48	0.5	0.63
Calmar	898.08	933.04	5.18	22.06	2.52	39370.76	41.71	4.14
Value-at-Risk	0.16	0.63	0.08	0.08	0.1	0.37	0.09	0.08
cVaR	0.18	0.63	0.13	0.12	0.14	0.37	0.13	0.12

Abbildung 8: Leistung der Portfolios eigene Darstellung

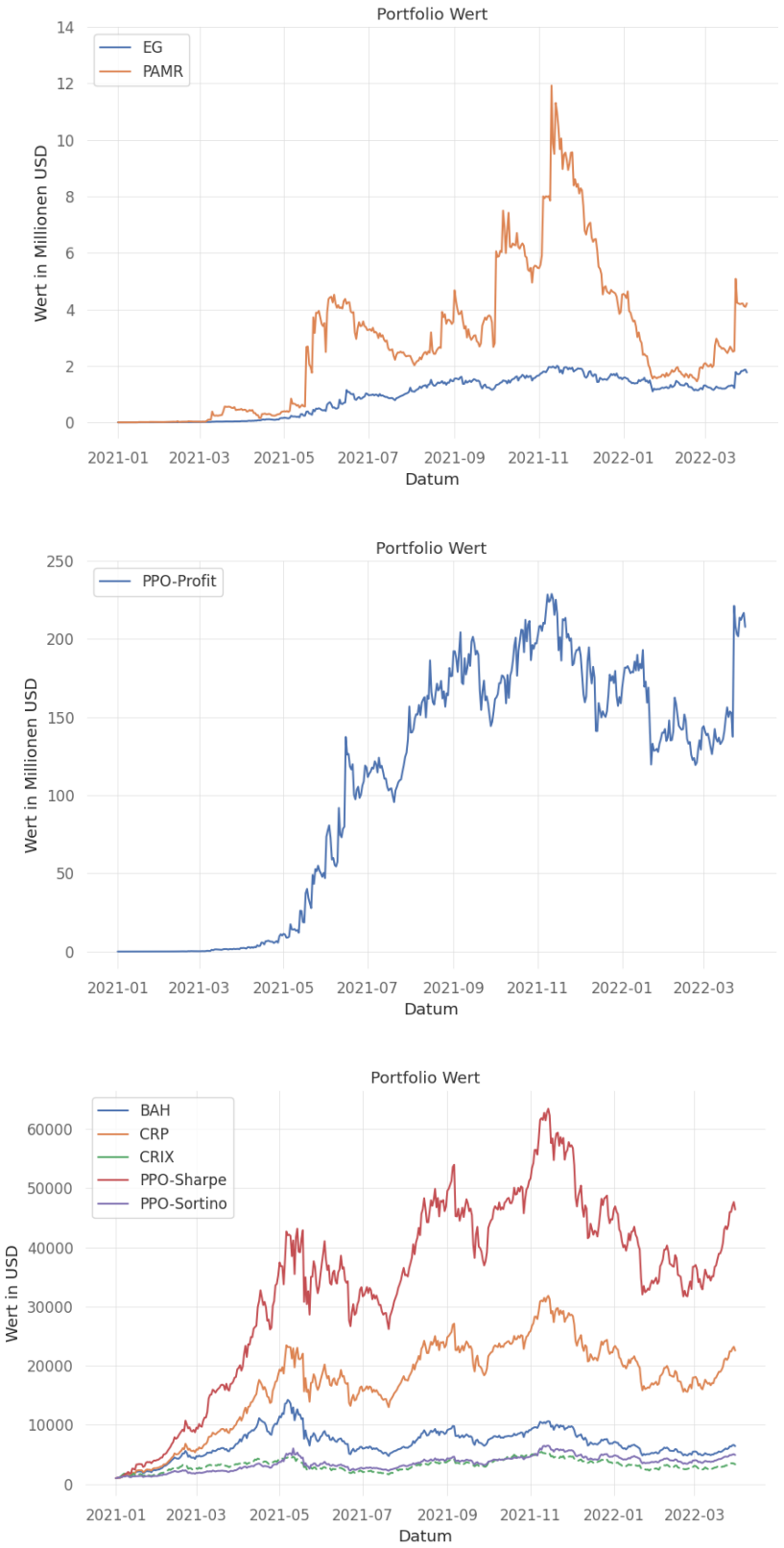

Abbildung 9: Wertentwicklung der Portfolios eigene Darstellung

6 Fazit

In dieser Arbeit wurde untersucht, ob Deep-Reinforcement-Learning für das Portfoliomanagement auf dem Kryptowährungsmarkt erfolgreich verwendet werden kann. Für diese Untersuchung wurde eine Auswahl von 50 Kryptowährungen für einen Zeitraum vom 01.01.2019 bis zum 31.03.2022 betrachtet und in einen Trainings- und Testdatensatz aufgeteilt. In einem weiteren Schritt wurde in drei Trainingsläufen ein neuronales Netzwerk unter Verwendung des Akteur-Kritiker-Reinforcement-Learning-Algorithmus PPO-Clip trainiert. Dabei wurden die folgenden drei Portfolioevaluierungsmetriken als Rewardfunktionen verwendet: Profit-, Sortino- und Sharpe-Ratio.

Um die Leistungsfähigkeit der drei konzipierten Deep-Reinforcement-Learning-Portfolios zu evaluieren, wurden eine Auswahl an unterschiedlichen Portfolio Strategien in die Untersuchung aufgenommen. Anschließend wurden diese anhand verschiedener Portfolio-Performance-Metriken für den Zeitraum vom 01.01.2021 bis zum 31.03.2022 beurteilt und verglichen.

Die Ergebnisse der Untersuchung zeigen, dass der PPO-Profit Agent alle anderen betrachteten Portfolios – gemessen am finalen Portfoliowert – übersteigt. Eine weitere Erkenntnis, die aus den Ergebnissen gezogen werden konnte, ist, dass der PPO-Sharpe-Agent den Markt – gemessen an den BAH, CRP und CRIX Strategien – übertreffen konnte, jedoch nicht der Lage ist, die Leistungsfähigkeit der EG- und PAMR-Strategien zu überbieten. Aufgrund der in dieser Untersuchung vorgestellten Ergebnisse wird die Schlussfolgerung gezogen, dass Deep-Reinforcement-Learning für das Portfoliomanagement auf dem Kryptowährungsmarkt erfolgreich verwendet werden kann.

Diese hier präsentierten Ergebnisse sollten jedoch in weiteren Analysen – basierend auf unterschiedlichen Rahmenbedingungen – untersucht werden. Weitere Untersuchungen sollten demnach eine größere Auswahl an Anlagemöglichkeiten für einen größeren Zeitraum und einer detaillierteren Datenfrequenz wie stündliche Daten verwenden. Zusätzlich sollten weitere unterschiedliche Reinforcement-Learning-Algorithmen in Kombination mit einer Auswahl an variierenden Netzwerkarchitekturen eingesetzt werden. Darüber hinaus wäre es ebenfalls möglich, eine alternative Verteilungsfunktion einzusetzen, die für den PPO-Algorithmus verwendet wird.

Literaturverzeichnis

Agarap, Abien Fred (2018): Deep Learning using Rectified Linear Units (ReLU). Online verfügbar unter http://arxiv.org/pdf/1803.08375v2.

Alessandretti, Laura; ElBahrawy, Abeer; Aiello, Luca Maria; Baronchelli, Andrea (2018): Anticipating Cryptocurrency Prices Using Machine Learning. In: *Complexity* 2018, S. 1–16. DOI: 10.1155/2018/8983590.

Alexander, S.; Coleman, T. F.; Li, Y. (2006): Minimizing CVaR and VaR for a portfolio of derivatives. In: *Journal of Banking & Finance* 30 (2), S. 583–605. DOI: 10.1016/j.jbankfin.2005.04.012.

Andrychowicz, Marcin; Raichuk, Anton; Stańczyk, Piotr; Orsini, Manu; Girgin, Sertan; Marinier, Raphael et al. (2020): What Matters In On-Policy Reinforcement Learning? A Large-Scale Empirical Study. Online verfügbar unter https://arxiv.org/pdf/2006.05990.

Arulkumaran, Kai; Deisenroth, Marc Peter; Brundage, Miles; Bharath, Anil Anthony (2017): A Brief Survey of Deep Reinforcement Learning. In: *IEEE Signal Process. Mag.* 34 (6), S. 26–38. DOI: 10.1109/MSP.2017.2743240.

Ben Sasson, Eli; Chiesa, Alessandro; Garman, Christina; Green, Matthew; Miers, Ian; Tromer, Eran; Virza, Madars (2014): Zerocash: Decentralized Anonymous Payments from Bitcoin. In: 2014 IEEE Symposium on Security and Privacy. 2014 IEEE Symposium on Security and Privacy (SP). San Jose, CA, 18.05.2014 - 21.05.2014: IEEE, S. 459–474.

Benhamou, Eric; Saltiel, David; Ungari, Sandrine; Mukhopadhyay, Abhishek (2020a): Bridging the gap between Markowitz planning and deep reinforcement learning. Online verfügbar unter http://arxiv.org/pdf/2010.09108v1.

Benhamou, Eric; Saltiel, David; Ungari, Sandrine; Mukhopadhyay, Abhishek; Atif, Jamal (2020b): AAMDRL: Augmented Asset Management with Deep Reinforcement Learning. Online verfügbar unter https://arxiv.org/pdf/2010.08497.

Betancourt, Carlos; Chen, Wen-Hui (2021): Deep reinforcement learning for portfolio management of markets with a dynamic number of assets. In: *Expert Systems with Applications* 164, S. 114002. DOI: 10.1016/j.eswa.2020.114002.

Borodin, A.; El-Yaniv, R.; Gogan, V. (2004): Can We Learn to Beat the Best Stock. In:

jair 21, S. 579–594. DOI: 10.1613/jair.1336.

Brauneis, Alexander; Mestel, Roland (2019): Cryptocurrency-portfolios in a mean-variance framework. In: *Finance Research Letters* 28, S. 259–264. DOI: 10.1016/j.frl.2018.05.008.

Bridle, John (1989): Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters. In: D. Touretzky (Hg.): Advances in Neural Information Processing Systems, Bd. 2: Morgan-Kaufmann. Online verfügbar unter https://proceedings.neurips.cc/paper/1989/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf.

Bu, Seok-Jun; Cho, Sung-Bae (2018): Learning Optimal Q-Function Using Deep Boltzmann Machine for Reliable Trading of Cryptocurrency. In: Hujun Yin, David Camacho, Paulo Novais und Antonio J. Tallón-Ballesteros (Hg.): Intelligent Data Engineering and Automated Learning - IDEAL 2018. Cham: Springer International Publishing, S. 468–480.

Bullmann, Dirk; Klemm, Jonas; Pinna, Andrea (2019): In Search for Stability in Crypto-Assets: Are Stablecoins the Solution? In: *SSRN Journal*. DOI: 10.2139/ssrn.3444847.

Buterin, Vitalik (2014): A next-generation smart contract and decentralized application platform.

Charpentier, Arthur; Elie, Romuald; Remlinger, Carl (2020): Reinforcement Learning in Economics and Finance. Online verfügbar unter https://arxiv.org/pdf/2003.10014.

Chen, Gang; Peng, Yiming; Zhang, Mengjie (2018): An Adaptive Clipping Approach for Proximal Policy Optimization. Online verfügbar unter https://arxiv.org/pdf/1804.06461.

Corbet, Shaen; McHugh, Grace; Meegan, Andrew (2017): The influence of central bank monetary policy announcements on cryptocurrency return volatility. In: *Investment Management and Financial Innovations* 14 (4), S. 60–72. DOI: 10.21511/imfi.14(4).2017.07.

Cover, Thomas M. (1991): Universal Portfolios. In: *Mathematical Finance* 1 (1), S. 1–29. DOI: 10.1111/j.1467-9965.1991.tb00002.x.

D. Erhan; Yoshua Bengio; Aaron C. Courville; Pascal Vincent (2009): Visualizing Higher-Layer Features of a Deep Network. In:

Deisenroth, Marc Peter (2011): A Survey on Policy Search for Robotics. In: *FNT in Robotics* 2 (1-2), S. 1–142. DOI: 10.1561/2300000021.

Ding, Yi; Liu, Weiqing; Bian, Jiang; Zhang, Daoqiang; Liu, Tie-Yan (2018): Investor-Imitator. In: Yike Guo und Faisal Farooq (Hg.): Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. London United Kingdom, 19 08 2018 23 08 2018. New York, NY, USA: ACM, S. 1310–1319.

Duchi, John; Shalev-Shwartz, Shai; Singer, Yoram; Chandra, Tushar (2008): Efficient projections onto the l1 -ball for learning in high dimensions. In: William Cohen (Hg.): Proceedings of the 25th international conference on Machine learning. the 25th international conference. Helsinki, Finland, 7/5/2008 - 7/9/2008. Association for Computing Machinery. New York, NY: ACM (ACM Other conferences), S. 272–279.

Fang, Fan; Ventre, Carmine; Basios, Michail; Kanthan, Leslie; Martinez-Rego, David; Wu, Fan; Li, Lingbo (2022): Cryptocurrency trading: a comprehensive survey. In: *Financ Innov* 8 (1). DOI: 10.1186/s40854-021-00321-6.

Goldberg, Yoav (2015): A Primer on Neural Network Models for Natural Language Processing. Online verfügbar unter https://arxiv.org/pdf/1510.00726.

Goodfellow, Ian; Courville, Aaron; Bengio, Yoshua (2016): Deep learning. Cambridge, Massachusetts: The MIT Press (Adaptive computation and machine learning).

Hakwa, Brice; Jäger-Ambrożewicz, Manfred; Rüdiger, Barbara (2015): Analysing systemic risk contribution using a closed formula for conditional value at risk through copula. In: *COSA* 9 (1). DOI: 10.31390/cosa.9.1.08.

Hambly, Ben; Xu, Renyuan; Yang, Huining (2021): Recent Advances in Reinforcement Learning in Finance, 08.12.2021. Online verfügbar unter http://arxiv.org/pdf/2112.04553v2.

Hao Zheng; Zhanlei Yang; Wenju Liu; Jizhong Liang; Yanpeng Li (2015): Improving deep neural networks using softplus units. In: 2015 International Joint Conference on Neural Networks (IJCNN), S. 1–4.

Harwick, Cameron (2014): Crypto-Currency and the Problem of Intermediation. In: *SSRN Journal*. DOI: 10.2139/ssrn.2523771.

Helmbold, David P.; Schapire, Robert E.; Singer, Yoram; Warmuth, Manfred K. (1998): On-Line Portfolio Selection Using Multiplicative Updates. In: *Mathematical Finance* 8 (4), S. 325–347. DOI: 10.1111/1467-9965.00058.

Holubar, Mario S.; Wiering, Marco A. (2020): Continuous-action Reinforcement Learning for Playing Racing Games: Comparing SPG to PPO. Online verfügbar unter http://arxiv.org/pdf/2001.05270v1.

Huang, Zhenhan; Tanaka, Fumihide (2022): MSPM: A modularized and scalable multi-agent reinforcement learning-based system for financial portfolio management. In: *PloS one* 17 (2), e0263689. DOI: 10.1371/journal.pone.0263689.

Imajo, Kentaro; Minami, Kentaro; Ito, Katsuya; Nakagawa, Kei (2020): Deep Portfolio Optimization via Distributional Prediction of Residual Factors. Online verfügbar unter http://arxiv.org/pdf/2012.07245v1.

Jiang, Zhengyao; Liang, Jinjun (2016): Cryptocurrency Portfolio Management with Deep Reinforcement Learning. Online verfügbar unter https://arxiv.org/pdf/1612.01277.

Jiang, Zhengyao; Xu, Dixing; Liang, Jinjun (2017): A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. Online verfügbar unter http://arxiv.org/pdf/1706.10059v2.

Kingma, Diederik P.; Ba, Jimmy (2014): Adam: A Method for Stochastic Optimization. Online verfügbar unter https://arxiv.org/pdf/1412.6980.

LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (2015): Deep learning. In: *Nature* 521 (7553), S. 436–444. DOI: 10.1038/nature14539.

Lee, Jinho; Kim, Raehyun; Yi, Seok-Won; Kang, Jaewoo (2020): MAPS: Multi-Agent reinforcement learning-based Portfolio management System. Online verfügbar unter https://arxiv.org/pdf/2007.05402.

Leem, JoonBum; Kim, Ha Young (2020): Action-specialized expert ensemble trading system with extended discrete action space using deep reinforcement learning. In: *PloS one* 15 (7), e0236178. DOI: 10.1371/journal.pone.0236178.

Li, Bin; Hoi, Steven C. H. (2014): Online portfolio selection. In: *ACM Comput. Surv.* 46

(3), S. 1–36. DOI: 10.1145/2512962.

Li, Bin; Wang, Jialei; Huang, Dingjiang; Hoi, Steven C. H. (2018): Transaction cost optimization for online portfolio selection. In: *Quantitative Finance* 18 (8), S. 1411–1424. DOI: 10.1080/14697688.2017.1357831.

Li, Bin; Zhao, Peilin; Hoi, Steven C. H.; Gopalkrishnan, Vivekanand (2012): PAMR: Passive aggressive mean reversion strategy for portfolio selection. In: *Mach Learn* 87 (2), S. 221–258. DOI: 10.1007/s10994-012-5281-z.

Li, Yuxi (2018): Deep Reinforcement Learning. Online verfügbar unter http://arxiv.org/pdf/1810.06339v1.

Liang, Zhipeng; Chen, Hao; Zhu, Junhao; Jiang, Kangkang; Li, Yanran (2018): Adversarial Deep Reinforcement Learning in Portfolio Management. Online verfügbar unter https://arxiv.org/pdf/1808.09940.

Lim, Qing Yang Eddy; Cao, Qi; Quek, Chai (2021): Dynamic portfolio rebalancing through reinforcement learning. In: *Neural Comput & Applic*. DOI: 10.1007/s00521-021-06853-3.

Linsmeier, Thomas J.; Pearson, Neil D. (2000): Value at Risk. In: *Financial Analysts Journal* 56 (2), S. 47–67. DOI: 10.2469/faj.v56.n2.2343.

Lucarelli, Giorgio; Borrotti, Matteo (2020): A deep Q-learning portfolio management framework for the cryptocurrency market. In: *Neural Comput & Applic* 32 (23), S. 17229–17244. DOI: 10.1007/s00521-020-05359-8.

Maas, Andrew L.; Hannun, Awni Y.; Ng, Andrew Y. (2013): Rectifier nonlinearities improve neural network acoustic models. In: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing.

Magdon-Ismail, Malik; Atiya, Amir F.; Pratap, Amrit; Abu-Mostafa, Yaser S. (2004): On the maximum drawdown of a Brownian motion. In: *Journal of Applied Probability* 41 (1), S. 147–161. DOI: 10.1239/jap/1077134674.

Markowitz, Harry (1952): Portfolio Selection. In: *The Journal of Finance* 7 (1), S. 77. DOI: 10.2307/2975974.

Marzban, Saeed; Delage, Erick; Li, Jonathan Yumeng; Desgagne-Bouchard, Jeremie; Dussault, Carl (2021): WaveCorr: Correlation-savvy Deep Reinforcement Learning for

Portfolio Management. Online verfügbar unter http://arxiv.org/pdf/2109.07005v2.

Maverick, J.b. (2015): What Is a Good Sharpe Ratio? In: *Investopedia*, 08.01.2015. Online verfügbar unter https://www.investopedia.com/ask/answers/010815/what-good-sharpe-ratio.asp, zuletzt geprüft am 16.04.2022.

Meng, Yuan; Kuppannagari, Sanmukh; Prasanna, Viktor (2020): Accelerating Proximal Policy Optimization on CPU-FPGA Heterogeneous Platforms. In: 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). Fayetteville, AR, USA, 03.05.2020 - 06.05.2020: IEEE, S. 19–27.

Meunier, Sebastien (2018): Blockchain 101. In: Alastair Marke, Bianca Sylvester, Justin D. Macinante und Stefan Klauser (Hg.): Transforming climate finance and green investment with blockchains. London, San Diego, CA, Cambridge, MA, Oxford: Academic Press an imprint of Elsevier, S. 23–34.

Moody, John; Saffell, Matthew (1998): Reinforcement Learning for Trading. In: M. Kearns, S. Solla und D. Cohn (Hg.): Advances in Neural Information Processing Systems, Bd. 11: MIT Press. Online verfügbar unter https://proceedings.neurips.cc/paper/1998/file/4e6cd95227cb0c280e99a195be5f6615-Paper.pdf.

Nakamoto, Satoshi (2008): Bitcoin: A Peer-to-Peer Electronic Cash System. In: *SSRN Journal*. DOI: 10.2139/ssrn.3977007.

Olah, Chris; Mordvintsev, Alexander; Schubert, Ludwig (2017): Feature Visualization. In: *Distill* 2 (11). DOI: 10.23915/distill.00007.

OpenAI; Berner, Christopher; Brockman, Greg; Chan, Brooke; Cheung, Vicki; Dębiak, Przemysław et al. (2019): Dota 2 with Large Scale Deep Reinforcement Learning. Online verfügbar unter https://arxiv.org/pdf/1912.06680.

Ormos, Mihály; Urbán, András (2013): Performance analysis of log-optimal portfolio strategies with transaction costs. In: *Quantitative Finance* 13 (10), S. 1587–1597. DOI: 10.1080/14697688.2011.570368.

Patel, Yagna (2018): Optimizing Market Making using Multi-Agent Reinforcement Learning. Online verfügbar unter https://arxiv.org/pdf/1812.10252.

Pendharkar, Parag C.; Cusatis, Patrick (2018): Trading financial indices with reinforcement learning agents. In: *Expert Systems with Applications* 103, S. 1–13. DOI: 10.1016/j.eswa.2018.02.032.

Petukhina, Alla; Trimborn, Simon; Härdle, Wolfgang Karl; Elendner, Hermann (2021): Investing with cryptocurrencies – evaluating their potential for portfolio allocation strategies. In: *Quantitative Finance*, S. 1–29. DOI: 10.1080/14697688.2021.1880023.

Pretorius, Ruan; van Zyl, Terence (2022): Deep Reinforcement Learning and Convex Mean-Variance Optimisation for Portfolio Management. DOI: 10.36227/techrxiv.19165745.v1.

Rose, Chris (2015): The Evolution Of Digital Currencies: Bitcoin, A Cryptocurrency Causing A Monetary Revolution. In: *IBER* 14 (4), S. 617. DOI: 10.19030/iber.v14i4.9353.

Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (1986): Learning representations by back-propagating errors. In: *Nature* 323 (6088), S. 533–536. DOI: 10.1038/323533a0.

S&P Dow Jones Indices (2022): Custom Indices - Royalton CRIX Crypto Index | S&P Dow Jones Indices. Online verfügbar unter https://www.spglobal.com/spdji/en/custom-indices/royalton-partners-ag-rpag/royalton-crix-crypto-index/#overview, zuletzt aktualisiert am 02.06.2022, zuletzt geprüft am 02.06.2022.

Sawhney, Ramit; Wadhwa, Arnav; Agarwal, Shivam; Shah, Rajiv Ratn (2021): Quantitative Day Trading from Natural Language using Reinforcement Learning, S. 4018–4030. DOI: 10.18653/v1/2021.naacl-main.316.

Schulman, John; Moritz, Philipp; Levine, Sergey; Jordan, Michael; Abbeel, Pieter (2015): High-Dimensional Continuous Control Using Generalized Advantage Estimation. Online verfügbar unter https://arxiv.org/pdf/1506.02438.

Schulman, John; Wolski, Filip; Dhariwal, Prafulla; Radford, Alec; Klimov, Oleg (2017): Proximal Policy Optimization Algorithms. Online verfügbar unter https://arxiv.org/pdf/1707.06347.

Sharpe, William F. (1994): The Sharpe Ratio. In: *JPM* 21 (1), S. 49–58. DOI: 10.3905/jpm.1994.409501.

Shi, Si; Li, Jianjun; Li, Guohui; Pan, Peng (2019): A Multi-Scale Temporal Feature

Aggregation Convolutional Neural Network for Portfolio Management. In: Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke Rundensteiner, David Carmel et al. (Hg.): Proceedings of the 28th ACM International Conference on Information and Knowledge Management. CIKM '19: The 28th ACM International Conference on Information and Knowledge Management. Beijing China, 03 11 2019 07 11 2019. New York, NY, USA: ACM, S. 1613–1622.

Silver, David; Hubert, Thomas; Schrittwieser, Julian; Antonoglou, Ioannis; Lai, Matthew; Guez, Arthur et al. (2018): A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. In: *Science (New York, N.Y.)* 362 (6419), S. 1140–1144. DOI: 10.1126/science.aar6404.

Soleymani, Farzan; Paquet, Eric (2020): Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. In: *Expert Systems with Applications* 156, S. 113456. DOI: 10.1016/j.eswa.2020.113456.

Soleymani, Farzan; Paquet, Eric (2021): Deep Graph Convolutional Reinforcement Learning for Financial Portfolio Management -- DeepPocket. Online verfügbar unter https://arxiv.org/pdf/2105.08664.

Sortino, Frank A.; van der Meer, Robert (1991): Downside risk. In: *JPM* 17 (4), S. 27–31. DOI: 10.3905/jpm.1991.409343.

Srivastava, Pooja; Mazhar, Syed Shahid (2018): Comparative Analysis of Sharpe and Sortino Ratio with reference to Top Ten Banking and Finance Sector Mutual Funds. In: *International Journal of Management Studies* V (4(2)), S. 93. DOI: 10.18843/ijms/v5i4(2)/10.

Sutton, Richard S.; Barto, Andrew (2018): Reinforcement learning, second edition. An introduction. 2nd ed. Cambridge: MIT Press (Adaptive computation and machine learning).

Trimborn, Simon; Härdle, Wolfgang Karl (2018): CRIX an Index for cryptocurrencies. In: *Journal of Empirical Finance* 49, S. 107–122. DOI: 10.1016/j.jempfin.2018.08.004.

van Saberhagen, Nicolas (2013): CryptoNote v 2.0, 2013.

Wang, Jingyuan; Zhang, Yang; Tang, Ke; Wu, Junjie; Xiong, Zhang (2019): AlphaStock: A Buying-Winners-and-Selling-Losers Investment Strategy using Interpretable Deep Reinforcement Attention Networks, S. 1900–1908. DOI:

10.48550/arXiv.1908.02646.

Wang, Rundong; Wei, Hongxin; an Bo; Feng, Zhouyan; Yao, Jun (2020): Deep Stock Trading: A Hierarchical Reinforcement Learning Framework for Portfolio Optimization and Order Execution. Online verfügbar unter https://arxiv.org/pdf/2012.12620.

Wang, Zhicheng; Huang, Biwei; Tu, Shikui; Zhang, Kun; Xu, Lei (2021): DeepTrader: A Deep Reinforcement Learning Approach for Risk-Return Balanced Portfolio Management with Market Conditions Embedding. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (1), S. 643–650. Online verfügbar unter https://ojs.aaai.org/index.php/AAAI/article/view/16144.

Weng, Liguo; Sun, Xudong; Xia, Min; Liu, Jia; Xu, Yiqing (2020): Portfolio trading system of digital currencies: A deep reinforcement learning with multidimensional attention gating mechanism. In: *Neurocomputing* 402, S. 171–182. DOI: 10.1016/j.neucom.2020.04.004.

Xu, Ke; Zhang, Yifan; Ye, Deheng; Zhao, Peilin; Tan, Mingkui (2020): Relation-Aware Transformer for Portfolio Policy Learning. In: Christian Bessiere (Hg.): Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20: International Joint Conferences on Artificial Intelligence Organization, S. 4647–4653.

Yang, Hongyang; Liu, Xiao-Yang; Zhong, Shan; Walid, Anwar (2020): Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. In: *SSRN Journal*. DOI: 10.2139/ssrn.3690996.

Yashaswi, Kumar (2021): Deep Reinforcement Learning for Portfolio Optimization using Latent Feature State Space (LFSS) Module. Online verfügbar unter https://arxiv.org/pdf/2102.06233.

Ye, Yunan; Pei, Hengzhi; Wang, Boxin; Chen, Pin-Yu; Zhu, Yada; Xiao, Jun; Li, Bo (2020): Reinforcement-Learning based Portfolio Management with Augmented Asset Movement Prediction States. Online verfügbar unter https://arxiv.org/pdf/2002.05780.

Yu, Pengqian; Lee, Joon Sern; Kulyatin, Ilya; Shi, Zekun; Dasgupta, Sakyasingha (2019): Model-based Deep Reinforcement Learning for Dynamic Portfolio Optimization. Online verfügbar unter http://arxiv.org/pdf/1901.08740v1.

Zhang, Zihao; Zohren, Stefan; Roberts, Stephen (2020): Deep Learning for Portfolio Optimization (4). Online verfügbar unter https://arxiv.org/pdf/2005.13665.

Abschließende Erklärung

Ich versichere hiermit, dass ich meine Masterarbeit „Deep Reinforcement Learning für Portfolio Optimierung auf Krypto Märkten“ selbstständig und ohne fremde Hilfe angefertigt habe, und dass ich alle von anderen Autoren wörtlich übernommenen Stellen wie auch die sich an die Gedankengänge anderer Autoren eng anlehnenden Ausführungen meiner Arbeit besonders gekennzeichnet und die Quellen zitiert habe.

Münster, den 2. Juni 2022

Julien Pierre Georg Pédurand

Einverständniserklärung

zur Prüfung meiner Arbeit mit einer Software zur Erkennung von Plagiate

Name: Pédurand **Vorname:** Julien Pierre Georg

Matrikelnummer: 432619 **Studiengang:** VWL

Adresse: Horstmarer Landweg 84, 48149 Münster

Titel der Arbeit: Deep Reinforcement Learning für Portfolio Optimierung auf Krypto Märkten

Was ist ein Plagiat? Als ein Plagiat wird eine Übernahme fremden Gedankengutes in die eigene Arbeit angesehen, bei der die Quelle, aus der die Übernahme erfolgt, nicht kenntlich gemacht wird. Es ist dabei unerheblich, ob z.B. fremde Texte wörtlich übernommen werden, nur Strukturen (z.B. argumentative Figuren oder Gliederungen) aus fremden Quellen entlehnt oder Texte aus einer Fremdsprache übersetzt werden.

Softwarebasierte Überprüfung. Alle Bachelor- und Masterarbeiten werden vom Prüfungsamt mit Hilfe einer entsprechenden Software auf Plagiate geprüft. Die Arbeit wird zum Zweck der Plagiatsüberprüfung an einen Software-Dienstleister übermittelt und dort auf Übereinstimmung mit anderen Quellen geprüft. Zum Zweck eines zukünftigen Abgleichs mit anderen Arbeiten wird die Arbeit dauerhaft in einer Datenbank gespeichert. Ein Abruf der Arbeit ist ausschließlich durch die Wirtschaftswissenschaftliche Fakultät der Westfälischen Wilhelms-Universität Münster möglich. Der Studierende erklärt sich damit einverstanden, dass allein zum beschriebenen Zweck der Plagiatsprüfung die Arbeit dauerhaft gespeichert und vervielfältigt werden darf. Das Ergebnis der elektronischen Plagiatsprüfung wird dem Erstgutachter mitgeteilt.

Sanktionen. Liegt ein Plagiat vor, ist dies ein Täuschungsversuch i.S. der Prüfungsordnung, durch den die Prüfungsleistung als „nicht bestanden“ gewertet wird. Es erfolgt eine Mitteilung an das Prüfungsamt und die dortige Dokumentation. In schwerwiegenden Täuschungsfällen kann der Prüfling von der Prüfung insgesamt ausgeschlossen werden. Dies kann unter Umständen die Exmatrikulation bedeuten. Plagiate können auch nach Abschluss des Prüfungsverfahrens und Verleihung des Hochschulgrades zum Entzug des erworbenen Grades führen.

Hiermit erkläre ich, dass ich die obigen Ausführungen gelesen habe und mit dem Verfahren zur Aufdeckung und Sanktionierung von Plagiaten einverstanden bin.

Münster, den 2. Juni 2022

Julien Pierre Georg Pédurand