



hochschule mannheim

Künstliche Intelligenz und Finanzrisikomanagement

Madrid Dajsinani

Bachelorarbeit

zur Erlangung des akademischen Grades Bachelor of Science (B.Sc.)

Studiengang Informatik

Fakultät für Informatik

Hochschule Mannheim

06.02.2020

Betreuer

Prof. Dr. Elena Fimmel, Hochschule Mannheim

Prof. Dr. Ivo Wolf, Hochschule Mannheim

Dajsinani, Madrid:

Künstliche Intelligenz und Finanzrisikomanagement / Madrid Dajsinani. –
Bachelorarbeit, Mannheim: Hochschule Mannheim, 2020. 66 Seiten.

Dajsinani, Madrid:

Artificial Intelligence and Financial Risk Management / Madrid Dajsinani. –
Bachelor Thesis, Mannheim: University of Applied Sciences Mannheim, 2020. 66 pages.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich bin damit einverstanden, dass meine Arbeit veröffentlicht wird, d. h. dass die Arbeit elektronisch gespeichert, in andere Formate konvertiert, auf den Servern der Hochschule Mannheim öffentlich zugänglich gemacht und über das Internet verbreitet werden darf.

Mannheim, 06.02.2020

Madrid Dajsinani

Abstrakt

Künstliche Intelligenz und Finanzrisikomanagement

Künstliche Intelligenz (KI) und maschinelles Lernen (ML) ändern auch die Finanzbranche. Die enorme Menge an Transaktionen, Börsenbewegungen und spezifischen Techniken der Finanzkriminalität bedeuten eine enorme Menge an Daten, die ein unterschiedliches finanzielles und nichtfinanzielles Risiko darstellen.

Ziel dieser Bachelorarbeit ist es, einen Überblick über KI und Techniken des MLs und deren Auswirkungen auf das finanzielle Risikomanagement zu geben. Die häufigsten ML-Algorithmen wie Support Vector Machine, Random Forest, Künstliches Neuronales Netz usw. werden beschrieben und ihre Anwendung auf finanzielle Risiken wie Kredit-, Börsen-, Devisen-, Betriebs- und Liquiditätsrisiken. Letztendlich wird ein Long-Short-Term-Memory (LSTM) Ansatz implementiert, um den Marktpreis von Google und Microsoft vorherzusagen. Die Performance hängt von vielen Faktoren ab, verspricht jedoch eine Verbesserung des Marktrisikos. In den letzten beiden Kapiteln werden die Herausforderungen beim Einsatz von Technologien für ML und KI im Finanzrisikomanagement beschrieben und die innovativsten KI-Startups nach Forbes und Bloomberg aufgelistet.

Artificial Intelligence and Financial Risk Management

Artificial Intelligence (AI) and Machine Learning (ML) are also disrupting the financial industry. The enormous amount of transactions, stock exchange movements and specific techniques of financial crime mean an enormous amount of data that represent different types of financial and non-financial risk.

The aim of this bachelor thesis is to provide an overview of AI and ML techniques and their effects on financial risk management. The most common ML algorithms such as Support Vector Machine, Random Forest, Artificial Neural Network (ANN) etc. are described and their application to financial risks such as credit, stock market, foreign exchange, operational and liquidity risks. Ultimately, an LSTM approach is implemented to predict the stock market price of Google and Microsoft. Performance depends on many factors, but it promises to improve market risk. The last two chapters describe the challenges of using ML and AI technologies in financial

risk management and list the most innovative AI-startups according to Forbes and Bloomberg.

Inhaltsverzeichnis

1	Einleitung	1
2	Maschinelles Lernen	3
2.1	Was ist Maschinelles Lernen?	3
2.2	Analysetechniken	5
2.2.1	Descriptive Analytik	5
2.2.2	Predictive Analytik	6
2.2.3	Preskriptive Analytik	6
2.3	Anwendungen von maschinellen Lernen	7
2.3.1	Natural Language Processing - Computerlinguistik	7
2.3.2	Automatisierte Planung	7
2.4	Lernarten	8
2.4.1	Supervised Learning - (Überwachtes Lernen)	8
2.4.2	Unsupervised Learning - (Unüberwachtes Lernen)	9
2.5	Reinforcement Learning - (Bestärktes Lernen)	10
2.6	Neuronale Netze und Deep Learning	11
2.7	Datenanalyse beim maschinellen Lernen	12
2.7.1	Klassifikation	12
2.7.2	Regression	13
2.7.3	Clustering	13
3	Algorithmen	15
3.1	Entscheidungsbaum	15
3.1.1	Entscheidungsbaumsarten	16
3.1.2	ID3-Algorithmus	16
3.2	Support Vector Machine	17
3.2.1	Berechnung der optimalen Hyperebene	19
3.3	k-Nächste-Nachbarn	19
3.4	Bayesian networks	21
3.5	Random Forest	22
3.6	Künstliches neuronales Netz	24
3.7	Convolutional Neural Network	25
3.7.1	Convolutional Layer	26
3.7.2	Pooling Layer	27

3.7.3	Fully-Connected Layer	27
3.8	Recurrent Neural Network	27
3.9	Long-Short-Term-Memory	30
4	KI-Anwendung im Finanzrisikomanagement	33
4.1	Arten von Finanzrisiken	33
4.1.1	Kreditrisiko	33
4.1.2	Marktrisiko	34
4.1.3	Liquiditätsrisiko - Refinanzierungsrisiko	34
4.1.4	Operationales Risiko	35
4.1.5	Wechselkursrisiko	35
4.2	Anwendung auf das Kreditrisiko	36
4.3	Anwendung auf das Marktrisiko	37
4.4	Anwendung auf das Liquiditätsrisiko	39
4.5	Anwendung auf das Operationales Risiko	40
4.6	Anwendung auf das Forex Risiko	41
5	RNN-Ansatz bei der Vorhersage von Aktienkursen	43
5.1	Das Ziel dieses Ansatzes	43
5.2	Technologien	44
5.3	Implementierung	45
5.4	Optimierung von Hyperparametern	49
5.5	Ergebnisse	51
6	KI-Startups in der Finanzindustrie	55
7	Herausforderungen des MLs und der KI	61
7.1	Datenverfügbarkeit	61
7.2	Rechenleistung	62
7.3	Qualifiziertes Personal	63
7.4	Black-Box-Technologie	63
7.5	Algorithmus Bias	63
7.6	Datenschutz	64
8	Diskussion	65
	Abkürzungsverzeichnis	ix
	Tabellenverzeichnis	xi
	Abbildungsverzeichnis	xiii
	Quellcodeverzeichnis	xv
	Literatur	xvii

Index	xxv
9 Erster Anhang	xxv
9.1 Importieren der erforderlichen Bibliotheken.	xxv
9.2 Daten laden und aufteilen	xxv
9.3 Datenvorbereitung	xxvi
9.4 Model erstellen	xxvi
9.5 Vorhersage auf Validierungsdaten	xxvii
9.6 Berechnung der Vorhersagefehlerrate	xxvii

Kapitel 1

Einleitung

KI ist in letzter Zeit eine der am meisten erforschten Technologien von Forschern und großen Technologieunternehmen. Ihre Anwendungen sind Teil des täglichen Lebens und jeder interagiert täglich mit ihr. Es handelt sich um eine Technologie, die das Umfeld jeder Industrie und jedes Menschenlebens sowohl positiv als auch negativ verändert.

Aufgrund der umwälzenden Technologien stehen Industrie vor einem grundlegenden Wandel. Auch die Finanzindustrie ist Teil dieses Wandels. Online- und nicht Online-Zahlungen, die Instabilität der Preise, das Volumen der Aktien- und Devisenmärkte nimmt zu, usw., das bedeutet eine gigantische Menge an Daten. Informationen stellen auch eine Sicherheitslücke dar, wie z.B. Betrug, Finanzkriminalität usw. Es gibt also viele finanzielle und nicht-finanzielle Risiken, die erkannt und gesteuert werden müssen. Die Analyse und Verwaltung aller Informationen auf traditionelle Weise ist eine Herausforderung für Finanzanalysten. Die KI verspricht eine neue Art des Verständnisses, der Analyse der Finanzdaten und des Lernens aus ihnen, um die zukünftigen Ereignisse vorherzusagen.

Banken und Finanzinstitute investieren viel Geld in KI, um ihre Leistungen beim Management finanzieller und nicht-finanzieller Risiken zu verbessern.

Kapitel 2

Maschinelles Lernen

KI und ML ist eine Technologie, die für Unternehmen immer interessanter wird. Aufgrund der großen Datenmengen und der steigenden Rechenleistung versuchen die Unternehmen, die Vorteile der Daten zu nutzen, sie zu verstehen, um eine bessere Performance in der jeweiligen Branche zu erreichen. Beide Technologien sind Teil unseres täglichen Lebens, erleichtern uns tägliche Arbeit und bieten den Unternehmen die Möglichkeit, ihre Leistung zu verbessern. Dieses Kapitel gibt einen Überblick über die Techniken des MLs und der KI und deren Anwendungen.

2.1 Was ist Maschinelles Lernen?

ML ist der am häufigsten verwendete Begriff in den letzten Jahrzehnten. Viele Unternehmen sind auf der Suche nach neuen Wegen der Innovation, um einen Mehrwert in ihren Unternehmen zu schaffen. Aufgrund der großen Datenmengen versuchen sie, eine neue Ebene des Datenverständnisses zu erreichen, indem sie die Technologie und den Algorithmus des MLs nutzen. Das ML gibt den Unternehmen ein neues Wissen über die Daten, so dass sie die Veränderungen in ihrem Unternehmen kontinuierlich vorhersagen und analysieren können. Das Verständnis der Änderungen im Geschäftsbetrieb gibt ihnen die Möglichkeit, den nächsten Entscheidung innerhalb des Unternehmens vorherzusagen und zu planen. Es bedeutet, dass ML die Fähigkeit gibt, die Zukunft vorherzusagen und in den frühen Phasen des Auftretens von Problemen zu reagieren.

Ein Computer braucht einen Algorithmus, um ein Problem zu lösen. Es ist eine Sequenz von Anweisungen, die ausgeführt werden müssen, um die Eingabedaten

in Ausgabedaten umzuwandeln. [6]. Ein Beispiel ist die Entwicklung eines Algorithmus, um eine ungerade Zahl in einer Liste mit Zufallszahlen zu finden. Dieser Algorithmus prüft daher auf die eine oder andere Weise, ob die Zahl ungerade oder gerade ist. Algorithmen werden angewendet, um eine große Anzahl von Aufgaben zu lösen, aber es gibt auch viele Algorithmen, um die gleichen Aufgaben mit unterschiedlicher Leistung zu lösen. Es gibt viele Fälle, in welchen der verwendete Algorithmus viele Daten zur Lösung der Problem benötigt. Der Grundstein des MLs sind Algorithmen. Die Begriffe "ML" und "KI" werden oft missverstanden. Maschinelles Lernen ist ein Teil der Künstlichen Intelligenz. KI hat auch anderen Namen wie artifizielle Intelligenz, maschinelle Intelligenz, es ist also die Intelligenz von Maschinen. Es bedeutet, dass eine Maschine von der Umgebung lernen kann und die besten Entscheidungen trifft, um die Annäherung an die erfolgreiche Lösung eines Problems zu maximieren [8].

ML ist ein Mix aus einer Vielzahl von Algorithmen und das Antreiben des MLs sind Daten. Der Algorithmus kann kontinuierlich aus Daten lernen, so dass er sich selbst verbessert, Muster zur Beschreibung von Daten finden, um eine bessere Annäherung zu machen, das Ergebnis aus Daten vorhersagen. Also, ML ist die Programmierung von Computern, um erfolgreich die Problemlösung mit Hilfe von Daten aus der Vergangenheit zu approximieren [8].

Das Ziel des Algorithmus für ML ist es, ein Modell zu erstellen, das auf diesen Daten trainiert wird. Die Daten dienen als Eingabe für den Algorithmus und das Ergebnis ist ein Modell, das zur Lösung des Problems auf der Grundlage der trainierten Daten verwendet werden kann [8]. Es gibt drei Arten von Modellen, die durch den Algorithmus des MLs erzeugt werden können. Prädiktives Modell, um die Zukunft vorherzusagen, was als nächstes kommt, descriptives Modell, um wichtige, einflussreiche Informationen aus den Daten zu gewinnen oder beides, das präskriptives Modell genannt wird [8]. ML ist eine Technik oder Technologie zur Erstellung/Aufbau/Entwicklung eines Analysemodells.

Wir interagieren täglich mit ML Anwendungen. Der größte Teil der Weltbevölkerung nutzt soziale Medien wie Facebook, Instagram etc. für verschiedene Angebote. Diese Kommunikationsplattformen nutzen maschinelles Lernen in vielen Fälle wie z.B. Vorschläge von Freunden. Basierend auf der Profilhistorie und Hintergrundinformationen, macht maschinelles Lernen einen Vorschlag, was für Leute Sie kennen könnten, so dass es eine neue Freundschaftsverbindung basierend auf Benutzeraktivität, Profil etc. vorhersagt.

2.2 Analysetechniken

Die Analytik hat sich stark verändert und einen Fortschritt von der deskriptiven und prädiktiven Analytik zum ML gemacht. Die Analytik gibt den Unternehmen die Möglichkeit, die Daten zu verstehen, analysieren und davon lernen um die zukünftige Entscheidungen besser einzutreffen. Die beschreibt auch die bekannte und unbekannte Faktoren und die Zusammenhänge zwischen Einflussfaktoren. Aufgrund der prediktive Fähigkeit der Analytik sind die Unternehmen gezwungen damit zu beschäftigen, wenn die eine neue erfolgreiche Perspektive im Geschäft haben wollen. Durch die Datenpräsentation, die Zusammenfassung der Daten, die Datenabstraktion und die Übersichtlichkeit der Daten wird die Datenanalyse zur Grundlage des MLs.

2.2.1 Descriptive Analytik

Einblicke und Verständnis der Vergangenheit

Die deskriptive Analytik ist eine Methode und Technik der Datenanalyse. Sie hat die Fähigkeit, Daten aussagekräftig zu beschreiben, zu präsentieren und zusammenzufassen, sodass sie Daten in wertvolle Einsichten verwandelt [21]. Basierend auf diesen Erkenntnissen gibt sie den Datenwissenschaftlern die Möglichkeit, Schlussfolgerungen über die Daten hinaus zu ziehen oder Hypothesen aufzustellen. Für ein Unternehmen ermöglicht die Beschreibung, Darstellung und das Verständnis der Daten, ihre Tätigkeit im Geschäft besser zu verstehen und ihre zukünftigen Geschäftsaktivitäten und Entscheidungen zu optimieren [21]. Ein Beispiel ist die Anwendung der deskriptiven Analytik auf das Produktmarketing in Social Media. Basierend auf Benutzerprofil und Aktivität interpretiert sie die Daten, um zu verstehen, für welche Art von Produkt sich der Benutzer interessiert. In diesem Fall erhöht das Unternehmen das Ziel, Produkte zu verkaufen, dies wird als Zielmarketing bezeichnet.

Es handelt sich also um eine Methode, um möglichst viele einflussreiche, relevante und wichtige Informationen aus großen Datenmengen zu gewinnen und zu abstrahieren [21].

2.2.2 Predictive Analytik

Zukunftsverständnis

Die prädiktive Analytik ist Teil der fortgeschrittenen Analytikmethode zur Datenanalyse. Sie ist eine interdisziplinäre Methode der Statistik, der Modellierung, des MLs und der KI. Das Ziel der prädiktiven Analytik ist die Vorhersage zukünftiger Ereignisse oder Ergebnisse auf der Basis vergangener, historischer und aktueller Daten [21]. Sie verwendet eine große Variationsbreite verschiedener Algorithmen, um Muster und Beziehungen zwischen den Variablen im Datensatz zu finden, so dass sie die zukünftigen potentiellen Risiken, Ereignisse und Chancen identifiziert. Es gibt viele Algorithmen, die die Zukunft auf der Grundlage von Vergangenheits- und Aktuelldaten vorhersagen, aber es hängt von der Art des zu lösenden Problems ab. Diese Technik gibt dem Unternehmen die Möglichkeit, bessere Leistungen in ihrem Geschäft zu erbringen, indem es bessere zukünftige Entscheidungen trifft [21]. Sie verwendet strukturierte und unstrukturierte Daten, um Muster und Beziehungen zwischen den Daten zu finden, um Vorhersagen zu treffen [21]. Wenn Daten also Verzerrungen oder falsche Informationen enthalten, kann die Vorhersage aufgrund von Anomalien falsch sein [21]. Es handelt sich also um eine Technik zur Vorhersage zukünftiger Ereignisse durch Analytik und Lernen aus vergangenen und aktuellen Datensätzen.

2.2.3 Preskriptive Analytik

Beratung über mögliche zukünftige Ergebnisse

Die präskriptive Analytik ist ein wichtiger Teil des MLs. Sie ist eine Kombination aus deskriptiver und prädiktiver Analyse. Die präskriptive Analytik ist ein neuer Bereich der deskriptiven und prädiktiven Analytik, der versucht, den wahrscheinlichen, möglichen Ausgang eines Ereignisses vorherzusagen und den Faktor zu verstehen, der den Eintritt in diese Vorhersage beeinflusst [21]. Diese Methode kann sowohl für kurzfristige als auch für langfristige Vorhersagen verwendet werden. Sie gibt einem Unternehmen die Möglichkeit, den möglichen Ausgang zukünftiger Ereignisse vorherzusagen und zu verstehen, warum ein Ereignis eingetreten ist. Das Unternehmen kennt also auch den Faktor, der den Eintritt in diese Vorhersage beeinflusst. Der Motor der Analytik sind die Daten, so dass die präskriptive Analytik sowohl strukturierte als auch unstrukturierte Datentypen behandelt. Es handelt

sich also um eine Kombination aus fortschrittlichen verschiedenen Analysetechniken zur Vorhersage, Verschreibung und Anpassung, um das beste Ergebnis eines Ereignisses aus möglichen Ergebnissen in den verfügbaren Daten zu finden [21].

2.3 Anwendungen von maschinellen Lernen

KI und ML ist Teil unseres Lebens und ihre Anwendung ist überall, von unserem täglichen Leben bis hin zur Großindustrie. Zwei von vielen Anwendungen sind Computerlinguistik und Automatisierte Planung.

2.3.1 Natural Language Processing - Computerlinguistik

Natural Language Processing ist ein Bereich des MLs und der KI, welche der versucht den Computer und die Menschen durch natürliche Sprache zu verbinden. Diese Technologie gibt dem Computer die Fähigkeit, die Sprache sowohl in geschriebenem Text als auch in Sprache zu verstehen. Für einen Computer ist es ein schwieriger Prozess, die menschliche Sprache zu verstehen, da es viele Regeln zu lernen und zu befolgen gibt. Diese Technik gibt jedem die Möglichkeit, mit den fortschrittlichsten Technologien zu interagieren.

2.3.2 Automatisierte Planung

Automatisierte Planung ist ein Prozess von Entscheidungssequenzen, die ausgeführt werden müssen, um ein Ziel zu erreichen. Mit Hilfe von KI und ML ist ein Unternehmen in der Lage, seinen Planungsprozess zu automatisieren. Der traditionelle Weg der automatisierten Planung ist die Programmierung aller möglichen Entscheidungen, die in einer Entscheidungssequenz geschehen können. Die automatisierte Planung verarbeitet die unstrukturierten Daten und trifft die beste Entscheidung, um das Endziel zu erreichen, basierend auf den Informationen, die die Umgebung umgeben.

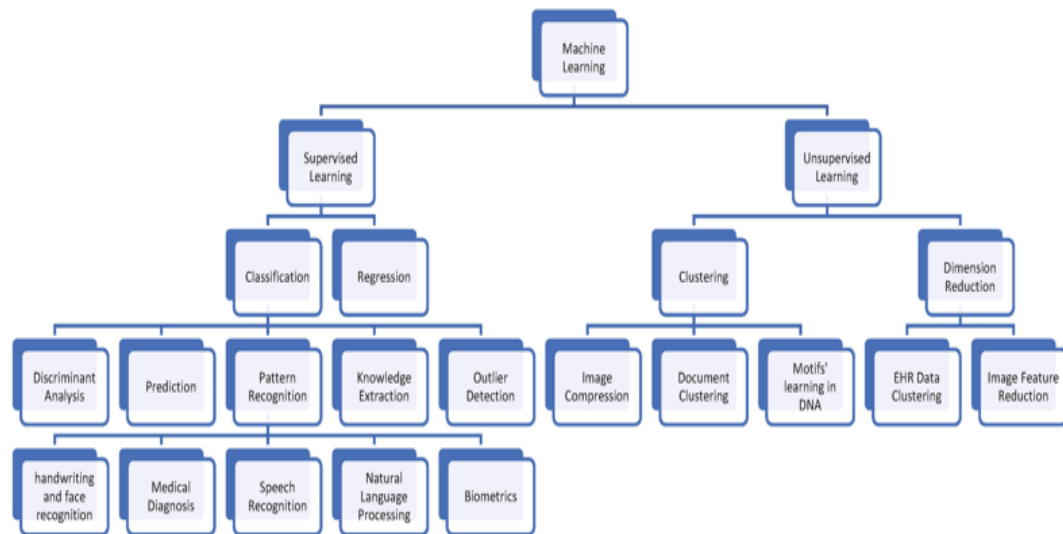


Abbildung 2.1: Methoden des maschinellen Lernens und ihre Anwendungen [21].

2.4 Lernarten

Die obige Abbildung 2.1 gibt einen großen Überblick über alle Methoden des MLs und einige Beispiele für deren Anwendung. Das Lernen aus Daten ist ein wichtiger Schritt beim ML, der sich in zwei Hauptgruppen unterteilt, nämlich in überwachtes und unbewachtes Lernen.

2.4.1 Supervised Learning - (Überwachtes Lernen)

Die am häufigsten angewandte Methode ist das überwachte Lernen. Es funktioniert, indem ein fest vorgegebener Datenwert als Input und ein Zielwert als Output definiert wird, so dass eine Abbildungsfunktion gefunden werden muss [29]. Die Daten werden mit aussagekräftigen Merkmalen versehen, so dass die Bedeutung der Daten definiert wird. So lernt und trainiert das überwachte Lernen selbst, um die beste Abbildungsfunktion zwischen den definierten Daten und den Zieldaten zu finden, wobei diese auch erklärbar sein sollte. Die Zieldaten sind die richtige Antwort der Eingabedaten. Der Trainingsprozess wird beendet, wenn eine gute Approximation des Mappings von Eingangs- zu Ausgangsdaten erreicht ist. Nach dem Training sollte das trainierte Modell in der Lage sein, die korrekte Ausgabe aus unbekannten Daten vorherzusagen. Es gibt zwei Arten von überwachtem Lernen, nämlich Klassifikation und Regression [29]. Die Klassifikation ist eine Technik der Klassenzuordnung. Sie kann Daten auf der Basis von trainierten Daten vorhersagen und

sie in Klassen klassifizieren. Ein einfaches Beispiel ist die Klassifikation, ob eine empfangene E-Mail ein Spam ist oder nicht. Regression ist eine weitere Technik der kontinuierlichen Datenvorhersage. Sie kann also eine Vorhersage auf der Basis von trainierten Daten treffen. Ein Beispiel für die Anwendung der Regression ist die Vorhersage des Aktienwertes auf der Basis von Vergangenheitsdaten. Die Basis für überwachtes Lernen sind Daten. Um die bestmögliche Abbildungsfunktion zu finden, werden die Daten in drei Gruppen aufgeteilt: Trainingsdaten, Validierungsdaten und Testdaten. Die Anwendung jeder dieser Gruppen hat einen Zweck im überwachten Lernen. Die Trainingsdaten werden zum Training des Modells verwendet. Der Aufbau eines guten Vorhersagemodells mit der geringsten Fehlerrate der Abbildungsfunktion erfordert die Optimierung einiger Parameter. Diese Parameter können mithilfe der Validierungsdaten spezifiziert werden.

2.4.2 Unsupervised Learning - (Unüberwachtes Lernen)

Neben überwachtem Lernen wird auch unüberwachtes Lernen angewendet. Diese Methode versucht, die Grenzen des überwachten Lernens zu überwinden. Die Grenze des überwachten Lernens ist das Training, so dass es nur vorhersagen kann, wofür es trainiert wurde [29]. Ein weiteres Problem von überwachtem Lernen ist, dass es mit etikettierten Daten arbeitet, was eine Herausforderung für das unüberwachte Lernen darstellt. Im Vergleich zum überwachten Lernen haben die Daten des unüberwachten Lernens nicht die richtige Antwort auf die Frage, was die bestmögliche Abbildungsfunktion vom Input zum Output ist, aber sie versuchen, die tatsächlichen Daten zu verstehen, um den Output vorherzusagen, dass er nicht vordefiniert ist. Da es keine vordefinierte Ausgabe, keine richtige Antwort in Daten gibt, kann auf Daten nicht eine Technik angewendet werden, um eine gut approximierte Abbildungsfunktion wie durch überwachtes Lernen [29].

Beim unüberwachten Lernen werden die Daten also in Gruppen aufgeteilt, die die Ähnlichkeit der Merkmale teilen. Dieser Prozess wird als Clustering bezeichnet, um den Abstand der Daten in einer Gruppe zu minimieren und den Abstand zwischen den Gruppen zu maximieren [29]. Im Allgemeinen hilft das unüberwachte Lernen den Datenwissenschaftlern, unentdeckte Muster oder Cluster in den Daten zu finden, die Dimensionalität der Daten zu reduzieren und die Anomalien in den Daten zu entfernen. Es werden verschiedene Techniken des unüberwachten Lernens wie das Clustering angewendet, die im Folgenden beschrieben werden.

2.5 Reinforcement Learning - (Bestärktes Lernen)

Reinforcement Learning (RL) ist eine maschinelle Lerntechnik, die sich auf das Training und das Lernen aus dem Misserfolg und den erfolgreichen Handlungen konzentriert. Es ist keine ganze Kombination aus überwachtem und unüberwachtem Lernen. Es beruht nicht auf etikettierten Trainingsdaten und es ist auch kein unüberwachtes Lernen, da der Algorithmus ein Feedback erhält, das für jede Entscheidung Belohnung oder Bestrafung sein kann [29]. So lernt das Modell durch Ausprobieren und Fehler.

Ein Beispiel für RL ist der Devisenhandel. Der Aktienkurs bewegt sich nach oben und unten und das Modell macht kontinuierlich Vorhersagen, um eine Entscheidung zu treffen. In jeder Situation des Forex-Marktes muss das Modell zwischen drei Alternativen entscheiden: Verkaufen, Kaufen oder Halten eines Währungspaares. Für jede falsche Entscheidung wird der Agent bestraft, so dass er lernt, dass es eine schlechte Entscheidung war, sonst wird er belohnt. Das Modell lernt aus seinen Versuchen und dem Misserfolg, die beste Entscheidung zu finden. Dieser Prozess oder dieses Feedback wird als "Reinforcement" bezeichnet [66].

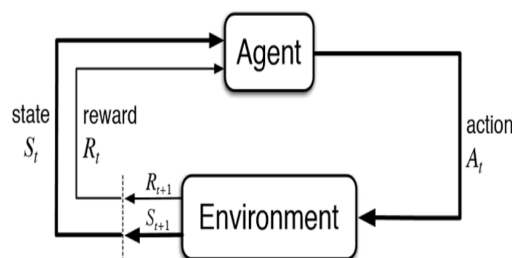


Abbildung 2.2: Bestandteil des Reinforcement Learning und deren Interaktionen[66]

RL basiert auf drei Komponenten:

1. Zustand (state): den aktuellen Status des Modells, der alle relevanten Informationen aus der Vergangenheit und der Gegenwart enthält.
2. Aktion (action): Entscheidungen, die vom Agent getroffen werden.
3. Belohnung (reward): basierend auf dem Erfolg der Aktion oder Entscheidung erhält der Agent ein Feedback oder eine Belohnung.

In der Abbildung 2.2 wird gezeigt, wie die Komponenten miteinander interagieren. Für jede Aktion wertet der Agent den aktuellen Zustand aus, trifft eine Aktion und erhält nach der Aktion ein Feedback, das Belohnung oder Bestrafung sein kann.

Im obigen Beispiel wird der Agent basierend auf vorherigen Entscheidungen und gesammelten Daten in Aktion gesetzt und erhält dann basierend auf dem Erfolg der Aktion ein Feedback. Auf der Grundlage von Verlust oder Gewinn erhält der Agent ein Feedback, das die nächste Aktion beeinflusst [29].

2.6 Neuronale Netze und Deep Learning

Aufgrund der großen Datenmengen und der zunehmenden Rechenleistung ist das Wort Deep Learning (DL) für Unternehmen bekannter geworden. Dieser Begriff und diese Technologie gab es schon früher, aber es braucht eine Menge an Daten, die trainiert werden müssen und auch Rechenleistung. DL ist ein spezieller Bereich des MLs, der sich auf die Entwicklung von Algorithmen spezialisiert hat, die versuchen, wie das menschliche Gehirn zu arbeiten, zu verstehen und zu analysieren [29]. Es ist eine Kombination aus überwachten und unüberwachten Lernen und nutzt deren Vorteile. Das Training dieser Algorithmen erfordert eine große Menge an Daten und Rechenleistung [29]. Es ist auch nützlich, Muster aus unstrukturierten Daten zu finden und daraus zu lernen.

Tiefe neuronale Netzwerke "Deep Neural Network (DNN)" sind dem menschlichen Gehirn ähnlich und enthalten eine große Anzahl von Neuronen, die miteinander verbunden sind. Das Gebiet des MLs wird auch als Neuronale Netze (NN) bezeichnet. NN bestehen aus verborgenen Schichten zwischen Eingangs- und Ausgangsdaten, haben also eine Eingangsschicht, eine oder mehrere verborgene Schichten und eine Ausgangsschicht [29]. Die Daten werden in die Eingabeschicht aufgenommen und das Ergebnis, die Vorhersage basierend auf der erzeugten Information aus der verdeckten Schicht, wird auf der Ausgabeschicht angezeigt. Die Berechnung wird von der verdeckten Schicht ausgeführt. Ein NN besteht je nach Komplexität des Problems aus mindestens drei bis vier verborgenen Schichten. Jede verborgene Schicht nimmt die Informationen der vorherigen Schicht auf und gibt sie an die nächste weiter. Der wichtigste Teil eines NNs ist das Lernen aus Daten. Jede Neuronenverbindung wird gewichtet und beginnt mit einem Zufallswert. Basierend auf dem Fehlerergebnis der vorhergehenden Gewichtung passt das NN die Gewichtung an [29]. Ist ein neuronales Netz nach dem Training trainiert, kann das Modell zur Vorhersage verwendet werden. Das Konzept des Fehlers ist sehr wichtig, da er während der

Trainingsphase minimiert werden muss. So wird die Fehlerrate durch jede Schicht reduziert [29].

DL ist eine Kombination aus überwachten und unüberwachten Algorithmen und nutzt deren Vorteile. Das Training dieser Algorithmen erfordert eine große Menge an Daten und Rechenleistung. Es ist auch nützlich, Muster aus unstrukturierten Daten zu finden und daraus zu lernen [29].

2.7 Datenanalyse beim maschinellen Lernen

Das Modell ist das Ergebnis eines trainierten Algorithmus. Es gibt drei Arten von Modellen für verschiedene Arten von Problemen. Die Klassifizierung der Daten in Klassen wird als "Klassifikation" und die Gruppierung von Daten in Gruppen als "Clustering" bezeichnet. Die "Regression" ist eine weitere Technik, um Eingabedaten auf kontinuierliche Ausgabewerte abzubilden, indem Beziehungen zwischen den Werten gefunden werden.

2.7.1 Klassifikation

Klassifizierung ist ein Problem beim ML und in der Statistik, das das Identifizieren von Klassen gegebener Daten beinhaltet [63]. Es ist also der Prozess der Vorhersage der Klasse, der als Ziele oder Kategorien bezeichnet werden kann. Diese Technik zum Erzeugen eines Vorhersagemodells ist die Aufgabe des Approximierens einer Abbildungsfunktion [63]. Durch die Abbildungsfunktion kann die diskrete Ausgabe basierend auf Eingabevariablen vorhergesagt werden. Die Klassifizierung wird in vielen Bereichen verwendet, und eine davon ist, ob es sich bei einer E-Mail um Spam oder Nicht-Spam handelt. Es handelt sich also um eine binäre Klassifizierung, da es nur zwei Klassen als Spam und Nicht-Spam gibt [63]. Um das Muster in Daten zu verstehen und zu identifizieren, muss der Klassifikator mit Trainingsdaten trainiert werden. Nach dem Training kann der Klassifikator jede E-Mail erkennen. Beim ML gehört die Klassifikation zum überwachten Lernen [63].

2.7.2 Regression

Regression ist auch eine weitere Technik des MLs und der Statistik zur Datenanalyse. Aufgrund der großen Menge an unstrukturierten und strukturierten Daten gibt es viele Daten, die ignoriert werden können, aber auch Daten, die wichtig und einflussreich für andere Daten sind [56]. Die Regression wird verwendet, um den Einfluss der abhängigen Variablen auf eine oder mehrere unabhängige Variablen zu messen [56]. Es gibt einen großen Unterschied zwischen Regression und Klassifikation, wobei die Regression eine Funktion zur Unterscheidung von Daten in kontinuierliche Daten statt in Klassen oder diskrete Werte findet [56].

Diese beiden Variablen sind die wichtigsten:

- Abhängige Variable: Dies ist die Variable, die vorhergesagt werden soll.
- Unabhängige Variable: Dies ist die Variable, die einen Einfluss auf die abhängige Variable haben soll.

Die Funktion, mit der das Modell in kontinuierlichen Daten gefunden wird, ist eine Regressionslinie, die als folgende Gleichung dargestellt wird [56]:

$$Y_i = a_0 + a_1 * X_i$$

2.7.3 Clustering

Clustering ist eine weit verbreitete Technik des unüberwachten Lernens. Das Ziel des Clustering ist es, Gruppen von ähnlichen Datenpunkten zu finden. Die Gruppierung von Daten erfolgt durch das Auffinden von Daten mit ähnlichen Eigenschaften und Merkmalen, die sich von den Daten anderer Gruppen unterscheiden [29]. Die Suche nach dem nächstgelegenen Datenpunkt mit der ähnlichen Eigenschaft wird durch einige Funktionen wie Cosinus-Distanz, Manhattan-Distanz oder Maximal-Distanz definiert [29]. Mit Hilfe dieser Funktionen kann der Algorithmus den nächstgelegenen Nachbarn finden. Im Allgemeinen sucht der Clustering-Algorithmus nach den nächstgelegenen Datenpunkten mit gleichen Merkmalen und Eigenschaften und gruppiert die Datenpunkte, die nahe beieinander liegen. Es gibt viele Fälle, in denen die Gruppen sich gegenseitig überlappen. In diesen Fall bringt Clustering nicht die beste Performance [29].

Kapitel 3

Algorithmen

3.1 Entscheidungsbaum

Der Entscheidungsbaum ist ein Algorithmus für überwachtes Lernen, der sowohl für die Lösung von Klassifikations- als auch von Regressionsproblemen eingesetzt wird [53]. Die haben eine sehr leistungsfähige Anwendung im ML bei der Klarstellung und Informationsgewinnung aus Daten. Das Ziel des Entscheidungsbaums ist die Vorhersage des Ausgangswertes auf der Basis von Eingangsdaten. Entscheidungsbaum basiertes Modell für Klassifikations- sowie Regressionsprobleme werden in Form einer Baumstruktur aufgebaut [53]. Die Lerninformationen des Entscheidungsbaums sind in einer hierarchischen Struktur strukturiert. Der besteht aus eine Reihe von if-then-else-Entscheidungsregeln, welche versuchen die bestmögliche Vorhersage zu finden.

Entscheidungsbäume werden in vielen Bereichen zur Lösung verschiedener Probleme eingesetzt, auch im Finanzbereich. Wie in der Abbildung 3.1 dargestellt, wird



Abbildung 3.1: Aktienkursprognose mit Hilfe von Entscheidungsbaum [71].

ein Entscheidungsbaum angewendet, um vorherzusagen, ob der Aktienkurs nach oben oder unten geht. Der Entscheidungsbaum unterteilt die Daten in Teilsätze, kleinere Teilsätze und das Ergebnis ist ein entwickelter endgültiger Entscheidungsbaum. Ein Entscheidungsbaum basiert auf zwei Elementen, Entscheidungsknoten und Blattknoten. Abhängig von der Komplexität des Problems kann ein Knoten einen oder mehrere Zweige haben, wobei die Hauptknoten eine Klassifizierung oder Entscheidung darstellen. Um eine klare Darstellung der Daten in einem Entscheidungsbaum zu erhalten und um eine bessere Entscheidung zu ermöglichen, wird der Baum reduziert. Durch den Reduzierungsprozess werden die Zweige des Baums verkürzt, indem einige Zweige in Führungsknoten umgewandelt werden. Dadurch wird die Größe des Baums reduziert [53].

3.1.1 Entscheidungsbaumsarten

Je nach Art der Zielvariablen werden die Entscheidungsbäume in zwei Gruppen unterteilt:

- **Kategorialer Variablenentscheidungsbaum:** Entscheidungsbaum, der eine kategoriale Zielvariable hat und dann als Entscheidungsbaum für kategoriale Variablen bezeichnet wird.
- **Kontinuierlicher Variablenentscheidungsbaum:** Der Entscheidungsbaum hat eine kontinuierliche Zielvariable und wird dann als kontinuierlicher Variablenentscheidungsbaum bezeichnet.

Es gibt mehrere Algorithmen zur Generierung eines Entscheidungsbaums und einer davon ist der weit verbreitete ID3-Algorithmus.

3.1.2 ID3-Algorithmus

Der am häufigsten verwendete Algorithmus zur Durchführung dieser Transformation von Rohdaten in die regelbasierte Entscheidungsbäume ist der Iterative Dichotomiser 3 (ID3). Es ist ein einfacher Lernalgorithmus für Entscheidungsbäume, der von Ross Quinlan (1986) entwickelt wurde [53]. Dichotomisierung bedeutet, Daten in zwei völlig entgegengesetzte Teile aufzuteilen [53]. Dies ist ein iterativer Algorithmus, der denselben Schritt ausführt, bis ein Entscheidungsbaum erstellt wurde. Um die dominantesten Attribute für die Erstellung eines Baums zu finden, werden

die Attribute in zwei Gruppen unterteilt [53]. ID3 ist ein Top-Down-Algorithmus. Es wird in jeder Menge ausgeführt, um jedes Attribut des Baumknotes zu testen. Um so viele Informationen wie möglich über ein ausgewähltes Attribut zu erhalten, wird eine Informationsverstärkungsfunktion verwendet. Diese Funktion wird verwendet, um das ausgewählte Attribut einem optimalen Lernsatz zuzuordnen. Dieser Algorithmus basiert auf zwei Funktionen:

1. **Entropie:** Die Informationstheorie ist ein grundlegender Satz, der häufig verwendet wird, um die Bedeutung von Informationen im Verhältnis zu ihrer Größe zu messen [53]. Ein Datensatz X enthält sowohl positive als auch negative Beispiele und die Entropie von X relativ zu dieser Klassifizierung ist.

$$H_{Entropie}(X) = \sum_{x \in X} -p(x) \log_2 p(x) \quad (3.1)$$

2. **Informationsgewinn:** In der Informationstheorie wird ein anderes ähnliches Konzept verwendet, das den Informationsgewinn jedes Attributs misst. Es misst den Unterschied in der Entropie vor und nach dem Trainingssatz X , der auf ein Attribut Y aufgeteilt ist [53]. Die Formel des Informationsgewinns lautet.

$$I_{Informationsgewinn}(X, Y) = H(X) - H(X|Y) \quad (3.2)$$

Entropie und Informationsgewinn sind Funktionen, die auf jedes Attribut angewendet werden, sodass das dominanteste Attribut bestimmt werden kann.

3.2 Support Vector Machine

Die Support Vector Machine (SVM) ist ein Algorithmus zum überwachten Lernen, um Modelle für Klassifikations- und Regressionsprobleme zu erstellen. Die SVM wird eher zur Lösung von Klassifikationsproblemen eingesetzt [32]. Das Ziel der SVM ist es, zwei Klassen durch eine Linie, die Hyperebene genannt wird, zu trennen.

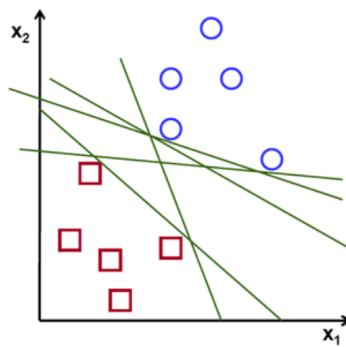


Abbildung 3.2: Alle möglichen Hyperebenen [45].

Wie in der Abbildung 3.2 gezeigt, gibt es viele Möglichkeiten, die beiden Klassen von Datenpunkten zu trennen. Das Ziel von SVM ist es, die beste Hyperebene zu finden, so dass die Trennung durch die Maximierung des Randes zwischen den Punkten jeder Seite erfolgt. Das bedeutet, dass der Abstand der trennenden Hyperebene zu den Entscheidungslinien maximiert wird [32]. Die Abbildung 3.3 zeigt, wie die optimale Hyperebene durch Maximierung des Randes gewählt wird.

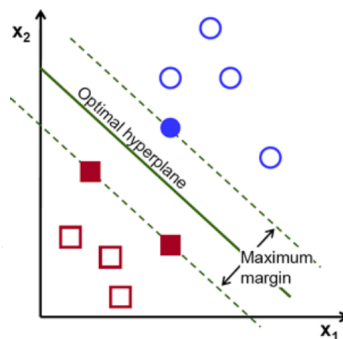


Abbildung 3.3: Maximum-Margin-Hyperebene, die aus zwei Klassen trainiert wurde [45].

Es gibt viele Unklarheiten, dass die Unterstützung der SVM nur im zweidimensionalen Raum funktioniert, aber sie zeigt auch gute Leistung in mehrdimensionalen Räumen, indem sie eine Hyperebene erzeugt. Die Hyperebene ist $p - 1$ dimensional als Datenpunkte, während die Datenpunkte die p Dimension haben [32]. SVM Modelle sind Teil des binären Klassifizierungsalgorithmus und das Ziel ist es, die Hyperebene mit maximaler Bandbreite zu finden [32].

3.2.1 Berechnung der optimalen Hyperebene

Das Ziel dieser Methode ist es, zwei Klassen durch eine optimale Hyperebene zu trennen, die eine $p - 1$ Dimension unter der p Dimension der Datenpunkte liegt. In einem 2D-Raum ist eine Hyperebene eine Linie und in einem 3D-Raum eine Ebene. Die Gleichung der optimalen trennenden Hyperebene lautet wie folgt:

$$f(x) = a_0 + a_1 * x_1 + a_2 * x_2 + \dots + a_p * x_p + b = 0$$

und eine allgemeine Funktion zur Berechnung der Hyperebene:

$$f(x) = \beta^T * x + \beta_0 = 0$$

wobei β einen Normalvektor der Hyperebene und β_0 die Bias darstellt [45]. Durch Skalierung von β und β_0 gibt es unendlich viele Möglichkeiten, eine Hyperebene darzustellen. Der y -Wert der Funktion kann sowohl einen negativen als auch einen positiven Wert zwischen -1 und 1 haben. Die Funktion der optimalen Maximum-Margin-Hyperebene ist:

$$|\beta^T * x + \beta_0| = 1$$

wobei x für die Punkte steht, die der Hyperebene am nächsten liegen, dem so genannten Support Vector [45].

In der SVM muss eine optimale Hyperebene durch Maximierung des Randes dargestellt werden. Die Gleichung zur Maximierung des Randes lautet also [45]:

$$Abstand_{supportvectors} = \frac{|\beta_0 + \beta^T x|}{||\beta||} = \frac{1}{||\beta||}$$

3.3 k-Nächste-Nachbarn

Die k-nächste-Nachbarn (kNN) ist ein Algorithmus des überwachten Lernens, der bei Klassifikationsproblemen weit verbreitet ist [77]. Er wird auch für Regressionsvorhersageprobleme verwendet. Dieser Algorithmus macht keine Vermutungen

über die Trainingsdaten, sondern versucht, die Eigenschaft der Trainingsdaten, die bei den nächsten Nachbarn liegen, zu vereinfachen [77].

In kNN ist k die Anzahl der Nachbarn, um die Beobachtungen zu klassifizieren. In diesem Algorithmus spielen einige Parameter eine wichtige Rolle und einer davon ist der k Faktor. Wie in der Abbildung 3.4 gezeigt wird, entscheidet k , wie groß der Klassifikatorradius sein soll. Ein groß gewähltes k kann den Zufallsfehler reduzieren, aber auf der anderen Seite die kleinen Merkmale der Daten ignorieren, die für die Daten vielleicht wichtig sind. Der Wert von k beeinflusst das Ergebnis und versucht, das Modell zwischen Unter- und Überanpassung auszugleichen [18].

Wie in der Abbildung 3.4 für $k = 1$ zeigt, sind die ausgewählten Punkte nur ein einziger roter Punkt, was die Vorhersage weniger stabil macht. Aber wenn $k = 4$ ist, sind vier Datenpunkte ausgewählt und einer davon ist grün. Das bedeutet nicht, dass 4 ein optimaler Wert für k ist, aber es ist viel besser, die gewünschte Vorhersage zu erhalten.

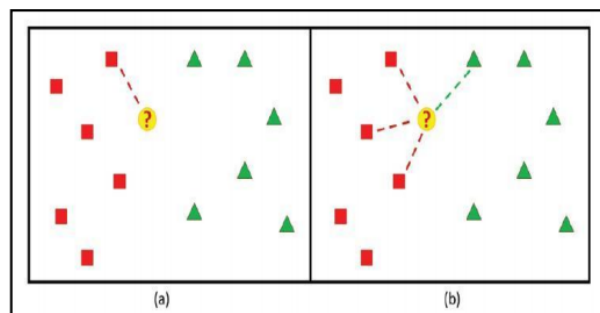


Abbildung 3.4: a) 1NN, b) 4NN mit dem Entscheidungspunkt [31].

Der Algorithmus sucht nach den nächsten Nachbarn vom Beobachtungspunkt aus. Die Suche nach dem nächsten Nachbarn bedeutet, dass der Abstand zwischen den Datenpunkten und der Beobachtung minimiert wird. Es gibt eine Vielzahl von Funktionen, die den Abstand zwischen zwei Punkten berechnen, und die Standardfunktion ist der euklidische Abstand. Die folgende Gleichung zeigt, wie der Abstand zwischen zwei Vektoren mit Koordinaten $p = (p_1, \dots, p_n)$ und $q = (q_1, \dots, q_n)$ gemessen wird [18]:

$$\text{Abstand}(p, q) = \|q - p\|_2 = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$

Andere Methoden, die zur Berechnung des Abstands zwischen zwei Datenpunkten im kNN angewendet werden können, sind Manhattan, Hamming usw. [18].

3.4 Bayesian networks

Ein weiterer Algorithmus, der für eine Vielzahl von Aufgaben einschließlich der Vorhersage verwendet wird, Bayesian Network (BN). Es ist ein wichtiger Algorithmus im Bereich der KI und wird verwendet, um Modelle aus Daten und/ oder Berichten zu erstellen. Diese können für eine Vielzahl von Aufgaben verwendet werden, einschließlich Vorhersage, Erkennung von Anomalien, Diagnose, automatisierte Inspektion, Argumentation, Zeitreihenvorhersage und Entscheidungsfindung unter Unsicherheit. Es gehört zur Gruppe der probabilistischen grafischen Modelle, die auf einem direkten azyklischen Graph basieren, das die Variable und ihre bedingten Abhängigkeiten darstellt [25]. Direkte azyklische Graphen sind in Statistiken, ML und KI sehr beliebt, und das BN basiert auf dieser Art von Graphen. Direkte azyklische Graphen bestehen aus einer Menge von Knoten und der Menge von Kanten, wobei Knoten Zufallsvariablen und Kanten direkte Abhängigkeiten zwischen verbundenen Variablen darstellen [25]. Jeder Knoten im BN repräsentiert eine Zufallsvariable und die Kanten repräsentieren Abhängigkeiten zwischen Knoten. Diese Abhängigkeit zwischen Knoten repräsentiert die abhängige Wahrscheinlichkeit zwischen Knoten. Insbesondere stellt eine Kante von Knoten X_i zu Knoten X_j eine abhängige Wahrscheinlichkeit zwischen den entsprechenden Variablen dar, so dass die Variable X_i beeinflusst die Variable X_j .

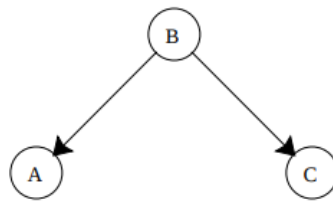


Abbildung 3.5: Ein Bayes'sches Netzwerk [64]

Die Figur 3.5 zeigt ein einfaches Bayes'sches Netzwerk mit einer Menge von Knoten $V = A, B, C$ und die Kanten $E = (B, A), (B, C)$. In diesem Netzwerk sind Knoten A und C von B abhängig, aber voneinander unabhängig. In diesem Beispiel ist die Wahrscheinlichkeit des Bayes'schen Netzwerks:

$$P(A, B, C) = P(A|B) * P(B) * P(C|B) \quad (3.3)$$

Im Allgemeinen zeigt diese Formel, wie die Wahrscheinlichkeit des BNs berechnet wird, wobei P die Wahrscheinlichkeit aller Knoten X_i und $Eltern(X_i)$ die Wahrscheinlichkeit der vorherigen Knoten ist [64].

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Eltern(X_i)) \quad (3.4)$$

3.5 Random Forest

Der Random Forest (RF) Algorithmus ist einer der leistungsfähigsten und beliebtesten Algorithmen für überwachtes ML. Er wird häufig verwendet, um Klassifikations- und Regressionsaufgaben zur Lösung von Problemen durchzuführen. Wie vom Namen her verständlich ist, konstruiert der Algorithmus einen Baumwald. Dieser Algorithmus ist eine Erweiterung der Entscheidungsbaummethode, die im vorherigen Abschnitt des Kapitels 3.1 erläutert wurde. RF ist eine Verbesserung des Entscheidungsbaums. Der Algorithmus enthält also mehrere Entscheidungsbäume, und jeder von ihnen macht seine eigene Vorhersage. Diese Methode ist eine Ensemble ML Methode. Das Konzept ist der Bildung eines starken Klassifikators, der auf einer Gruppe von schwachen Klassifikatoren basiert. Auf dem RF machen also die mehr Entscheidungsbäume das Vorhersagemodell robuster und stabiler [17].

Der RF ist gut bei Klassifizierungsproblemen, aber nicht so gut wie Regressionsprobleme. Daher gibt es keine präzise kontinuierliche Natur der Vorhersage. RF model wird in einer Vielzahl von Branchen wie der Finanzbranche angewendet. Er wird für die Vorhersage von Aktienkursen und zur Unterstützung von Banken oder Finanzinstituten bei der Kreditvergabe eingesetzt.

Die RF wählt zufällig eine Reihe von Entscheidungsbäumen aus dem ursprünglichen Trainingsdatensatz aus, wobei die Regressions- und Klassifizierungsbäume nicht beschnitten werden. RF basiert auf dem Ensemble-Lernen. Die zwei bekannten Methoden des Ensemble Lernens sind das "Bagging" und das "Boosting" 3.6. "Boosting" ist eine Methode, bei der jeder Klassifikator von der Vorhersage des vorherigen Baumes abhängt [17]. Ansonsten ist Bagging eine Technik, bei der das Ergebnis jedes Baumes nicht von der Vorhersage des vorhergehenden Baumes abhängt, sondern ein paralleler Prozess der Vorhersage ist [17]. Am Ende werden alle Klassifikatoren aggregiert, um eine endgültige Vorhersage zu treffen.

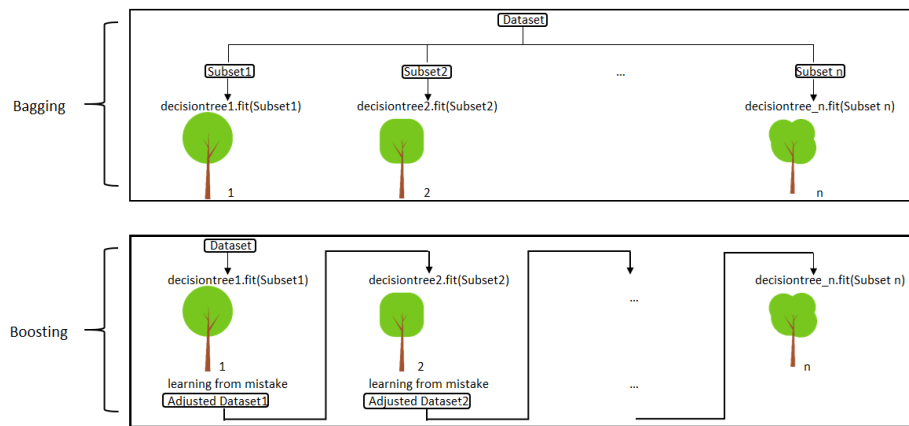


Abbildung 3.6: Bagging und Boosting Struktur [13]

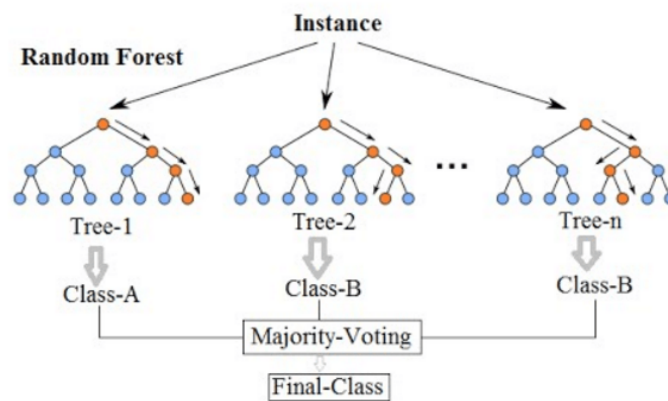


Abbildung 3.7: Struktur der Random Forest Klassifikation [55]

Wie in der Abbildung 3.7 gezeigt wird, kann ein RF viele Bäume haben und jeder von ihnen kann eine Klasse vorhersagen. Im Klassifizierungsmodell erfolgt die Vorhersage für eine Klasse als Vorhersage durch Stimmenmehrheit. Das Konzept dahinter ist, dass die Klasse mit der höchsten Stimme als Vorhersage gewählt wird. Das Endergebnis ist eine Aggregation der Stimmen aus jedem Baum [17]. Bei einem Regressionsmodell ist die Endvorhersage jedoch die Summe aller Baumvorhersagen geteilt durch die Anzahl der Bäume, also der Durchschnitt [17].

Der RF ist eine Kombination aus vielen Entscheidungsbäumen, die jeweils auf eine andere Beobachtung trainiert werden. Die endgültige Vorhersage der RF wird durch Mittelung der Vorhersagen jedes einzelnen Baumes getroffen [17].

3.6 Künstliches neuronales Netz

NN sind ein weit verbreiteter Algorithmus im Bereich des MLs. Die Inspiration des NNs kommt vom menschlichen Gehirn. Das Ziel des NNs ist es, dem Computer die Fähigkeit zu geben, Daten zu verstehen, zu analysieren, mit Daten zu arbeiten und von denen zu lernen, wie es das menschliche Gehirn tut. NN werden auch ANN genannt.

Das NN hat die Fähigkeit, Muster in Daten zu finden und die Ausgabe von Daten vorherzusagen. Sie werden in verschiedenen Bereichen der Industrie, wie z.B. im Finanzwesen, beim Risikomanagement im Finanzbereich, eingesetzt.

In der obigen Abbildung 3.8 ist die Architektur eines NNs und seine Funktionsweise dargestellt. Wie im menschlichen Gehirn ist das fundamentale und grundlegende Element von NN das Neuron, das als Knoten bezeichnet wird. Die Rolle eines Neurons ist es, Inputs von anderen Neuronen oder externen Quellen zu erhalten und den generierten Wert an ein anderes Neuron weiterzugeben. Das Neuron multipliziert die Eingabewerte mit dem entsprechenden Gewicht der Eingaben. Das Neuron wendet eine Funktion auf die Summe der einzelnen Multiplikationsergebnisse an. Ein NN besteht aus vielen Neuronen, die in Schichten organisiert sind. Es enthält eine Eingabeschicht, verdeckte Schichten und eine Ausgabeschicht. Je nach dem zu lösenden Problem ist auch die Anzahl der verborgenen Schichten, aber normalerweise enthält es drei bis vier verborgene Schichten.

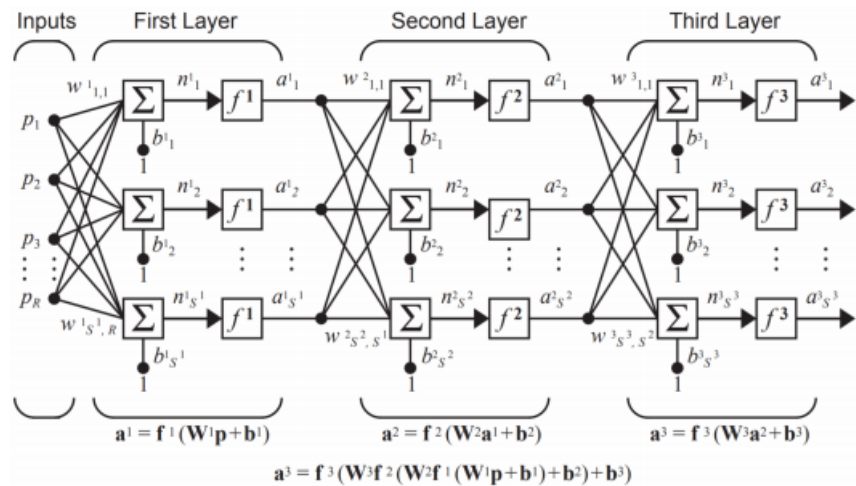


Abbildung 3.8: Mehrschichtiges neuronales Netzwerk [28].

Aufgrund des Informationsflusses wird der Vorgang im NN als "Feed Forward" bezeichnet. Das bedeutet, dass das Neuron die produzierte Information nicht an die vorhergehende Schicht sendet.

Wie die meisten Netze hat auch das NN eine Trainingsphase, in der aus den Daten gelernt wird. Nach der Trainingsphase muss das Netzwerk mit Testdaten getestet werden. Auch diese Phase ist wichtig, denn es gilt, die bestmöglichen Neuronengewichte zu finden. Das Neuronengewicht spielt eine große Rolle bei der Ausgabe von Informationen, so dass es bei unbekannten Daten das beste Ergebnis liefern kann [28]. Das Ziel des NNs ist es, durch das Verstehen und Analysieren der Eingangsinformationen die Ausgangsinformationen zu produzieren [28].

3.7 Convolutional Neural Network

Ein Convolutional Neural Network (CNN), das auch als CNN oder ConNet bezeichnet wird, ist eine Spezialisierung des NN, das bisher am häufigsten für die Analyse von Bildern verwendet wurde. Obwohl die Bildanalyse die am weitesten verbreitete Verwendung von CNNs ist, können sie auch für andere Datenanalyse- oder Klassifizierungsprobleme verwendet werden [5]. CNN ist ein NN, das eine Spezialisierung hat, um Muster zu erkennen und zu verstehen. Im Gegensatz zu NN, die Mustererkennung macht CNN so nützlich für die Bildanalyse. Das Problem bei den klassischen NN ist die Anzahl der Parameter, die die Modellierung zur Lösung komplexer

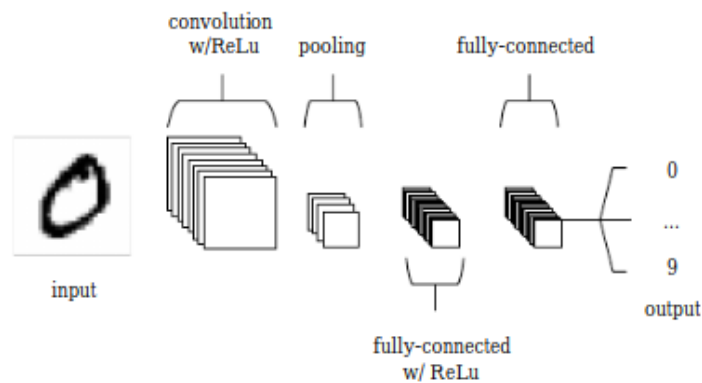


Abbildung 3.9: CNN-Architektur [47]

Aufgaben zu schwierig machen [5]. CNN reduziert die Anzahl der Parameter, die sie für die Annäherung an große Modelle zur Lösung komplexer Aufgaben besser geeignet machen. Das Merkmal hat die Forscher dazu veranlasst, größere Modelle zu entwerfen. CNN enthält mehrere Schichten, einschließlich convolutional Layer, non-linearity Layer, pooling Layer und fully-connected Layer. In der letzten Zeiten können und haben CNN auch andere nicht-convolutional Layer, aber die Basis von CNN sind die convolutional Layer [5]. Das Ziel von CNN ist die Bilder in einer Form zu bringen, die einfacher zu verarbeiten ist, welche dabei für eine gute Vorhersage hilft [5].

3.7.1 Convolutional Layer

Genauso wie jede andere convolutional Layer eine Eingabe empfängt, transformiert sie die Eingabe auf irgendeine Weise und gibt die Transformationseingabe an die nächste Ebene aus. Mit der convolutional Layer ist diese Transformation eine Convolution Funktion [5]. Wie bereits erwähnt, können die CNN Muster und Bilder erfassen, genauer gesagt, können convolutional Layer Muster erfassen.

Jede convolutional Layer hat eine Reihe von Filtern, die die Muster tatsächlich erkennen. Das Erkennen von Mustern in einem Bild bedeutet das Erkennen von Merkmalen wie Kanten, Formen, Objekten usw. Eine Art von Muster, die ein Filter erkennen kann, können Kanten und Bilder sein. Daher wird dieser Filter als Kantendetektor bezeichnet. Einige Filter erkennen möglicherweise Ecken, andere Kreise usw. Dies sind also einige einfache Filter, die einige übergeordnete Features erkennen sollen, sodass sie sich am Anfang des Netzwerks befinden. Je tiefer das

Netzwerk ist, desto ausgefeilter werden diese Filter. In weiteren Schichten können diese Filter also einfache Objekte wie Augen, Ohren usw. erkennen. Mit der zusätzlichen Ebene des Netzwerks sind die Filter in der Lage, übergeordnete Merkmale und Objekte wie Hunde, Katzen usw. zu erkennen.

3.7.2 Pooling Layer

Eine weitere Schicht, die nach der convolutional Layer hinzugefügt wird, ist die pooling Layer.

Die pooling Layer ist ähnlich wie die convolutional Layer. Die ist für die Reduzierung der räumlichen Größe der Matrix und der Anzahl der Eingangsparameter verantwortlich [5]. Dieser Prozess reduziert die Rechenleistung, die für die Verarbeitung der Daten durch dimensionale Reduktion erforderlich ist [5].

Es gibt zwei Arten von Pooling:

- *Max Pooling* gibt den Maximalwert aus dem Teil des Bildes zurück, der vom Kernel abgedeckt wird.
- *Average Pooling* gibt den Durchschnitt aller Werte aus dem Teil des Bildes zurück, der vom convolutional Layer abgedeckt wird.

3.7.3 Fully-Connected Layer

Eine andere Schicht, die auch eine wichtige Rolle in CNN hat, ist Fully-Connected Layer. Das Ziel einer Fully-Connected Layer besteht darin, die Ergebnisse des Convolutional-/ Pooling-Prozesses zu erfassen und sie zur Klassifizierung des Bildes in ein Etikett zu verwenden [47].

3.8 Recurrent Neural Network

Eine andere Technik für NN, die in verschiedenen Anwendungen weit verbreitet ist, das Recurrent Neural Network (RNN). In der realen Anwendung gibt es einige Szenarien, in denen das traditionale NN aufgrund seiner Einschränkung nicht angewendet werden kann. Die Einschränkung besteht darin, dass die Eingabedaten und

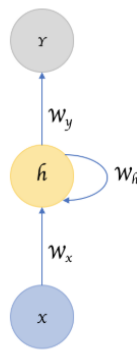


Abbildung 3.10: RNN-Architektur [27]

die Ausgabedaten des Netzwerks nicht untereinander verbunden sind [27]. Ein weiteres Problem besteht darin, dass dieses Netzwerk keinen Speicher modelliert [27]. Dies bedeutet, dass ein Netzwerk während der Berechnung keine Informationen speichern kann. Es gibt zu viele Anwendungen von NN, bei denen der Eingabedaten und der Ausgabedaten wie Sätze, Videoströme, Aktienkurse usw. miteinander verbunden werden sollten. Ein traditionelle NN, ein feedforward NN, kann das Problem der Vorhersagezeitreihen nicht lösen, die Vorhersage basiert auf der vorherigen Vorhersage. Ein reales Szenario ist die Vorhersage des Aktienkurses eines Unternehmens, wobei der nächste Kurs auf der Grundlage des vorherigen vorhergesagt wird und das Netzwerk einige Informationen über vorherige Schritte speichern muss. Das bedeutet, dass die Ausgabe des vorherigen Schritts die Eingabe des aktuellen Schritts und aller Berechnungen bis zum aktuellen Schritt ist. Daher wird dieser Prozess als rekursive Funktion bezeichnet. Das RNN löst dieses Problem mit Hilfe der verborgenen Schicht, und das wichtigste Merkmal von RNN ist der verborgene Zustand. Um Informationen darüber zu speichern, was bis zum aktuellen Schritt berechnet wurde, verwendet RNN einen Speicher. Dieses Netzwerk ist eine Art des NNs, das zur Erfassung von Informationen aus Sequenzen/ Zeitreihendaten ausgelegt ist. Im Vergleich zum feedforward NN teilt RNN über alle Zeitschritte den gleichen Parameter, so dass es nicht erforderlich ist, die Regeln für alle Eingabedaten zu lernen [27].

Die obige Abbildung 3.10 zeigt die allgemeine Komponente, aus der ein RNN besteht:

1. x sind die Eingabedaten als Sequenz.
2. h ist die rekursive Funktion, der Zustand des Netzwerks.

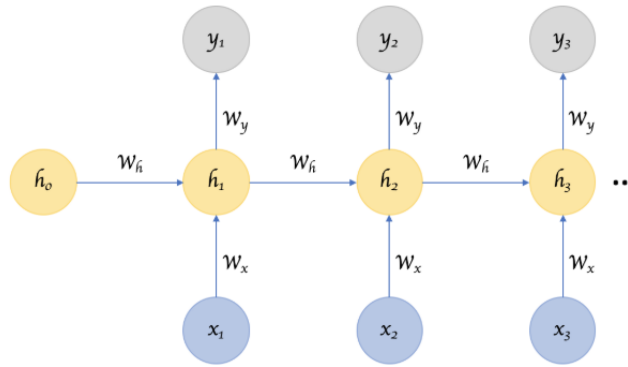


Abbildung 3.11: Die innere RNN-Architektur [27]

3. y ist die Vorhersage, die auf die Eingabedaten und die gespeicherte Daten zwischen Schritten basiert.
4. w_x und w_y sind die Gewichte von jeden Schritt.

Eine RNN ist wie eine rekursive Funktion, bei der sie sich wiederholt, bis das Ergebnis vorliegt. Die RNNs können wie mehrere Kopien vom desselben Netzwerks berücksichtigt werden. Die Abbildung 3.11 zeigt die Zwischenschritten eines RNNs. Die Menge der Eingabedaten x_1, x_2, x_3 bis x_t mit dem Zeitstempel 1, 2, 3 bis t und dem versteckten Zustand des RNNs h_1, h_2, h_3 bis h_t für jeden Zeitstempel.

Die Hauptfunktion von RNN ist es, den aktuellen Zustand des Netzwerks zu berechnen, also alle Informationen über die gesamten vergangenen Schritte. Die folgende Formel zeigt die Funktion, die den aktuellen Zustand des Netzwerks mit t Schritte und mit der Funktion $g^{(t)}$ berechnet:

$$h^{(t)} = g^{(t)}(x^{(t)}, x^{(t-1)}, x^{(t-2)}, \dots, x^{(2)}, x^{(1)}) \quad (3.5)$$

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta) \quad (3.6)$$

wobei die Variable h den Zustand und die θ die Parameter repräsentiert [27]. Um den aktuellen Netzwerkzustand zu berechnen, ruft die Funktion $g^{(t)}$ alle vergangene Daten (Sequenzen) $(x^{(t)}, x^{(t-1)}, x^{(t-2)}, \dots, x^{(2)}, x^{(1)})$ als Eingabeparameter ab.

3.9 Long-Short-Term-Memory

LSTM werden Teil der RNN-Familie bezeichnet. Das Problem bei RNN ist die Fähigkeit, mit langen Abhängigkeiten umzugehen. Das bedeutet, dass die Vorhersage von Daten abhängt, die weiter zurückverarbeitet wurden. Dies ist eine große Entfernung zwischen relevanten Daten und aktuellen Daten. Diese riesige Lücke kann dazu führen, dass der Gradient explodiert oder verschwindet, was dazu führt, dass das Modell falsche Vorhersagen liefert. Das Ziel von LSTM ist es, das Problem der Langzeitabhängigkeit zu lösen, das ihnen die Fähigkeit verleiht, sich Informationen über einen langen Zeitraum zu merken und dennoch zu lernen [30]. Die LSTMs sind speziell dafür ausgelegt, das Problem der Langzeitabhängigkeit zu lösen, d.h. zu lernen und sich für lange Zeiträume zu erinnern. LSTM haben auch die gleiche Architektur wie RNN 3.12, aber der Hauptunterschied ist die Architektur der LSTM-Einheit. Die LSTM-Einheit besteht aus Gates, die für die Regulierung des Informationsflusses in den und aus dem Speicher und die Zelle für den LSTM-Speicher verantwortlich sind [30]. Die Abbildung 3.12 zeigt die vollständige Architektur einer LSTM-Einheit.

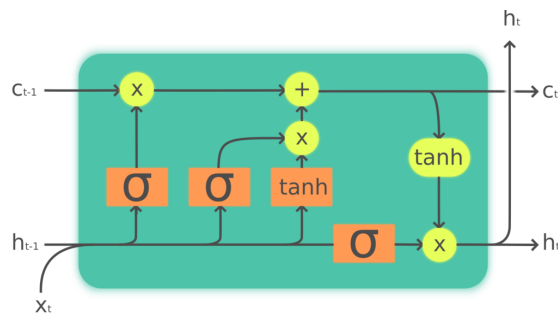


Abbildung 3.12: Die innere LSTM-Einheit Architektur [40]

1. Forget Gate (Vergesstor): Das erste Gatter, das die Ausgabe des vorherigen Zustands erhält, ist das Forget Gate. Die Zellzustandsdaten enthalten relevante und nicht relevante Daten. Um nur relevante Daten im Speicher zu haben, lässt Forget Gate nicht relevante Daten aus dem Speicher herausfallen. [39].

$$f_t = \text{sig}(W_f * [h_{t-1}, x_t] + b_f) \quad (3.7)$$

Die Datenbereinigung erfolgt durch eine Sigmoidschicht 3.7, deren Ausgabewerte zwischen $[0, 1]$ liegen. Ein Nullwert in der Sigmoidschicht steht für "alle Daten nicht relevant" sind, während eins für "alle Daten relevant" sind.

2. Input gate (Eingangstor): Die Rolle des Eingangsgatters besteht darin, die neuen Eingangsinformationen zu steuern, die in den Zellzustand eingefügt werden müssen [39]. Zuerst muss bestimmt werden, welche Werte aktualisiert werden sollen. Während eine Sigmoidschicht i_t die relevanten Werte des Zellzustands bestimmt, erzeugt eine Tanhschicht einen Vektor C'_t mit allen möglichen Werten, die dem Zellzustand hinzugefügt werden könnten 3.9.

$$i_t = sig(W_i * [+W_i * h_{t-1}] + b_i) \quad (3.8)$$

$$C'_t = tanh(W_c * [h_{t-1}, x_t] + b_c) \quad (3.9)$$

Nach der Entscheidung, welche neuen Informationen im Zellzustand gespeichert werden, ist der nächste Schritt die Aktualisierung des tatsächlichen Zellzustands gegenüber dem alten Zellzustand. Die Aktualisierung hängt von drei Komponenten ab: Zustand der alten Zelle C_{t-1} , mögliche Informationen, die aus dem alten Zellzustand entfernt werden könnten f_t , und die neuen möglichen Werte, die aktualisiert werden könnten i_t [39]. Die folgende Gleichung zeigt die Aktualisierungsgleichung.

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (3.10)$$

3. Output gate (Ausgangstor): Nach der Berechnung geht der letzte Schritt in die Ausgabeschicht. Vor dem Senden der Daten müssen einige Operationen durchgeführt werden. Zuerst muss die Ausgabe gefiltert werden. Die Sigmoidschicht wird angewendet, um die nicht benötigten Daten aus den berechneten Daten zu entfernen.

$$o_t = sig(W_o * [W_o * h_{t-1}] + b_o) \quad (3.11)$$

Die Werte im Zellzustand liegen in einem weiten Bereich, so dass eine Normalisierungsfunktion erforderlich ist, um sie in den gleichen Bereich zu bringen. Eine Tanhschicht $tanh(C_t)$ wird auf die Zellzustandswerte angewendet, so dass sie zwischen $[-1, 1]$ liegen. Beide Ergebnisse werden multipliziert, und dies ist die Ausgabe h_t der LSTM-Einheit, also die Eingabe der nächsten Einheit 3.12.

$$h_t = o_t * tanh(C_t) \quad (3.12)$$

Kapitel 4

KI-Anwendung im Finanzrisikomanagement

Heutzutage schreitet die Technologie sehr schnell voran und sie hat einen großen Einfluss auf alle Branchen. Big Data und ML, der Kern der KI, spielt eine große Rolle bei der Veränderung der Funktionsweise der Banken. Während des letzten Jahrzehnts setzen viele amerikanische Banken KI ein, um Risiken in einer Vielzahl von Filialen zu managen. Dieses Kapitel beschreibt die Auswirkung und Anwendung von Algorithmen für ML und KI im Risikomanagement, insbesondere bei finanziellen Risiken wie Kreditrisiko, Marktrisiko, Liquiditätsrisiko, operationellem Risiko und Wechselkursrisiko.

4.1 Arten von Finanzrisiken

4.1.1 Kreditrisiko

Das Kreditrisiko ist ein finanzielles Risiko, das hauptsächlich von Finanzinstituten wie Banken ausgeht. Banken leihen einem Kunden einen unterschiedlichen Geldbetrag aus, bei dem es sich um einen Privatkunden, ein Unternehmen oder eine Organisation handeln kann. Durch die Vergabe eines Darlehens muss die Bank ein Risiko eingehen, falls der Schuldner ein Darlehen nicht zurückzahlt oder die vertraglichen Vereinbarungen nicht einhält. Wenn der geschuldete Betrag und die Zinsen nicht eingehen, kann dies zu einer Unterbrechung der Zahlungsströme durch die Bank führen [34]. Um auf dieses Risiko zuzugreifen, führt der Kreditgeber eine Operation aus, die auf der Fähigkeit des Kreditnehmers basiert, dieses Darlehen

zurückzuzahlen [34]. Es gibt fünf wichtige Aspekten, die die Banken prüfen, bevor sie einen Kredit vergeben:

1. Bonitätshistorie des Kreditnehmer
2. Fähigkeit zur Rückzahlung
3. Kapital
4. die Darlehensbedingungen
5. zugehörige Sicherheiten

Das Kreditrisiko beschreibt auch das Risiko, dass ein Kreditnehmer auf Aufforderung keine Zahlung leistet oder eine Versicherungsgesellschaft nicht in der Lage ist, einen Anspruch zu begleichen [34]. Einige Unternehmen haben Abteilungen eingerichtet, die ausschließlich für die Beurteilung der Kreditrisiken ihrer derzeitigen und potenziellen Kunden zuständig sind [34]. Dank der Technologie können Unternehmen Daten zur Bewertung des Risikos eines Kunden schnell analysieren und bewerten.

4.1.2 Marktrisiko

Das Marktrisiko ist eine Art von Risiko, das durch Veränderungen bei Finanzinstituten entsteht. Es besteht die Möglichkeit von Verlusten aufgrund der Faktoren, die die Performance des Finanzmarktes beeinflussen. Um eine effektive Analyse des Marktrisikos zu erhalten, wird es in ein direktionales Risiko und ein nicht-direktionales Risiko unterteilt. Richtungsrisiken entstehen durch Bewegungen des Aktienkurses, der Zinssätze und mehr [34]. Im Gegensatz dazu können nicht-direktionale Risiken Volatilitätsrisiken sein [34]. Es gibt verschiedene Risikofaktoren, die das Marktrisiko ausmachen und die häufigsten sind die Zinssätze, Wechsel- und Börsenkurse, Commoditypreise, Volatilitäten [34]:

4.1.3 Liquiditätsrisiko - Refinanzierungsrisiko

Liquidität ist eine Art von Finanzrisiko, welches durch die Unfähigkeit der Durchführung der Transaktionen entsteht [34]. Liquidität ist die Fähigkeit eines Unternehmens oder einer Einzelperson, ihre Schulden zu begleichen, ohne katastrophale Verluste zu erleiden [34]. Umgekehrt ergibt sich das Liquiditätsrisiko aus der man-

gelnden Marktfähigkeit einer Anlage, die nicht schnell genug gekauft oder verkauft werden kann, um einen Verlust zu verhindern oder zu minimieren. Dies spiegelt sich normalerweise in ungewöhnlich großen Geld-Brief-Spreads oder großen Kursbewegungen wider. Das Liquiditätsrisiko kann in Asset Liquidity Risk und Funding Liquidity Risk unterteilt werden [34]. Das Asset-Liquiditätsrisiko entsteht entweder durch unzureichende Käufer oder unzureichende Verkäufer gegen Verkaufs- bzw. Kaufaufträge [34].

4.1.4 Operationales Risiko

Ein weiteres finanzielles Risiko ist ein operationales Risiko, das im Wesentlichen mit dem Ausfall des Geschäftsbetriebs aufgrund menschlicher Fehler verbunden ist [34]. Während des Tagesgeschäfts ist jedes Unternehmen Risiken und Gefahren ausgesetzt, und das Betriebsrisiko fasst diese Risiken und Probleme zusammen. Das Risiko ergibt sich aus Betriebsstörungen wie Missmanagement oder technischen Ausfällen [34]. Das Betriebsrisiko konzentriert sich darauf, wie ein Unternehmen arbeitet und wie es die Dinge im Tagesgeschäft verwaltet [34]. Dieses Risiko ist mit menschlichem Versagen verbunden, was bedeutet, dass weniger menschliche Interaktion ein geringeres operationales Risiko bedeutet.

4.1.5 Wechselkursrisiko

Das letzte finanzielle Risiko ist das Wechselkursrisiko. Jedes Unternehmen und jeder Investor sind an internationalen Transaktionen beteiligt, was jedoch zu Verlusten für das Unternehmen führen kann. Der Finanzmarkt bewegt sich jede Sekunde, daher bewegt sich auch der Wechselkurs auf und ab. Aufgrund finanzieller Schwankungen kann das Unternehmen Verluste verursachen [34]. Die Wertminderung der Anlagen aufgrund von Marktschwankungen der beteiligten Währungen wird durch das Wechselkursrisiko beschrieben [34]. Um dieses Risiko zu verringern, müssen die Risikoanalysten kontinuierlich große Datenmengen analysieren, was die Handhabung dieses Prozesses erschwert. Es gibt hochentwickelte Tools, die diese Minimierung des Risikos ermöglichen.

4.2 Anwendung auf das Kreditrisiko

Das Kreditrisiko ist ein wirtschaftlicher Verlust eines Finanzinstituts, der eintritt, wenn der Darlehnehmer seinen vertraglichen Verpflichtungen nicht nachkommt. Die vertragliche Verpflichtung des Darlehens für Darlehensteilnehmer besteht in der rechtzeitigen und regelmäßigen Zahlung des Kapitals oder der Darlehenszinsen. Der Prozess der Kreditvergabe nimmt im Laufe der Jahre aufgrund der unterschiedlichen Möglichkeiten, Marktanforderungen und Umstände der Kunden an Komplexität zu. Dies macht die Bank zu einem stark regulierten Unternehmen, von dem erwartet wird, dass es bei der Gewährung eines Kredits reagiert. Zusammen mit der ständig wachsenden Nachfrage der Benutzer nach Geschwindigkeit und Personalisierung greifen Banken und alle Kreditaussteller auf die Leistungsfähigkeit verschiedener Algorithmen zurück. In den letzten Jahren werden viele Algorithmen entwickelt, um die Analyse des Kreditrisikos zu verbessern. Altman veröffentlicht 1968 einen Aufsatz, der die Verwendung klassischer linearer, logit- und probit-Regressionen zur Modellierung des Kreditrisikos beschreibt [9]. Heutzutage verändert Technologie jeden Sektor der Branche aufgrund der Vorteile und der Unvollständigkeit der bisher angewandten Techniken. Auch Finanzinstitute spüren die Dynamik der KI und viele Start-ups oder Finance Technology (FinTech) erforschen die Anwendung der KI und des MLs im Kreditrisikomanagement. Die Unvollständigkeit und der Nachweis, dass KI das Kreditrisikomanagement verbessert führt dazu, dass FinTechs diese neuen Techniken untersuchen. Der Hauptgrund, warum KI und ML das Kreditrisikomanagement verbessern können, ist die Fähigkeit, unstrukturierte Daten semantisch zu verstehen und daraus ein Modell zu entwickeln. Die Nutzung von KI und ML zur Modellierung des Kreditrisikos ist nichts Neues. Bereits 1994 führen Altman und seine Kollegen eine Analyse durch, in der traditionelle statistische Methoden zur Vorhersage von Notfällen und Insolvenzen mit alternativen Algorithmen für NN verglichen wurden, und kommen zu dem Schluss, dass eine Kombination der beiden Methoden die Genauigkeit signifikant verbessert [9]. Die Komplexität der Beurteilung des Kreditrisikos eröffnet die Möglichkeit, KI und ML einzusetzen. Seit vielen Jahren werden Algorithmen für ML im Kreditrisikomanagement eingesetzt. Im Jahr 2018 veröffentlichtes Papier von Peter Addo beschreibt und vergleicht die Leistung mehrerer Algorithmen für ML, um festzustellen, ob ein Unternehmen einen Bankkredit nicht in Anspruch nimmt [1]. Die drei Algorithmen für ML sind "elastic net" (Erweiterung der logistischen Regression), RF und "gradient boosting". Es wird beobachtet, dass die gradient boosting alle

Modelle übertrifft, einschließlich der auf tiefem Lernen basierenden. Dies ist eine Demonstration dafür, dass bei Verwendung eines korrekten Feature-Engineerings Standardtechniken des MLs für die Beurteilung der Kreditwürdigkeit weiterhin relevant sind.

Das Potenzial von DL kann jedoch nicht ignoriert werden, und es ist nur eine Frage der Zeit, wann DL Modelle die Standardmodelle für ML für Aufgaben wie die Kreditrisikobewertung übertreffen. In einem weiteren Artikel aus dem Jahr 2018 wird die Möglichkeit untersucht, ein tiefes NN zur Vorhersage von Kreditkartenausfällen einzusetzen [65]. Das 5-stufige tiefe NN wird auf der Grundlage von Daten einer großen Bank in Brasilien ausgewertet, die 711.397 Datensätze von Kreditkarteninhabern mit 0,92 Kriminalfällen enthält. Das Modell übertrifft bestehende Lösungen wie Entscheidungsbäume, BN und logistische Regression. Bei der Ermittlung des Kreditrisikos komplexer, großvolumiger und unstrukturierter Daten wird intensiv nachgeforscht, wie tiefe NN eingesetzt werden können. Aufgrund der geringen Rechenleistung des tiefen NNs und der höheren Verfügbarkeit von Daten setzen viele Finanzinstitute DL-Lösungen ein, um das Kreditrisiko vorherzusagen [2].

4.3 Anwendung auf das Marktrisiko

Die Wahrscheinlichkeit, dass ein Anleger aufgrund von Finanzmarktfaktoren verlieren kann, wird als Marktrisiko bezeichnet. Es gibt verschiedene Faktoren, die sich auf die Wertentwicklung des Finanzmarkts auswirken und sich auf die Anlage und den Handel auswirken. Die Vorhersage des Börsenkurses und der Börsenwerte ist an sich schon eine Herausforderung und wird auch weiterhin eine Herausforderung für alle Finanzanalysten und Anleger sein. Mit der Entwicklung der Informatik und insbesondere der KI verketteten sich die Techniken des MLs, auch der Aktienmarkt. Die Vorteile des MLs haben sich auch auf die Art der Investition und des Handels ausgewirkt. Die Vorhersage der Börsenkurse und die Steuerung des Marktrisikos ist ein klassisches Problem. Es gibt viele Faktoren, die sich auf den Aktienmarkt auswirken, und es gibt viele verschiedene Techniken, um eine gute Vorhersage des Aktienkurses zu treffen. Die Vorhersage der Aktienkursen mit Techniken des MLs ist ein Bereich, der untersucht wurde. Es gibt viele Algorithmen und Lösungen für ML, die die Richtung des Aktienkurses vorhersagen. In diesem Artikel wird eine Lösung für ML vorgeschlagen, die auf der Vorhersage von Aktienkursen

basiert. Demnach sind NN und Support Vector Regression (SVR) der am häufigsten verwendete Algorithmus zum ML für die Vorgersage [48]. Zhang schlägt ein Aktienkurs-Vorhersagesystem vor [75]. Es ist in der Lage, Börsenkursbewegungen vorherzusagen und innerhalb eines vorgegebenen Zeitraums die Wachstums- oder Abnahmerate vorherzusagen [75]. Das vorgeschlagene Vorhersagemodell basiert auf einem RF-Model, das auf historischen Daten des Shenzhen Growth Enterprise Market (China) basiert. Dieses unüberwachte Lernmodell wird verwendet, um mehrere Clips von Aktien in vier Klassen zu klassifizieren (Hoch, Runter, Flach und Unbekannt). Ihre Bewertung in diesem Papier zeigt, dass das vorgeschlagene System der Volatilität des Marktes standhält und einige bestehende Methoden in Bezug auf Genauigkeit und Rendite pro Trade übertrifft [75]. ML wie das NN, RF und SVR wird in der Börsenvorhersage häufig eingesetzt. In diesem Artikel wird vorgeschlagen, die Anwendung des MLs bei der Preisvorhersage auf der Grundlage historischer Daten von zwei Unternehmen des indischen Aktienmarktes zu evaluieren: CNX Nifty und S&P BSE Sensex [48]. Für eine gute Leistung und Vorhersage ist dieser Ansatz ein zweistufiger Ansatz. Die erste Phase ist die Datenvorbereitung mithilfe von SVR. Der zweite Ansatz wendet NN, RF und SVR an, was zu SVR-NN, SVR-RF und SVR-SVR Vorhersagemodellen führt [48]. Die Ergebnisse dieser Arbeit zeigen, dass das zweistufige Vorhersagemodell eine viel bessere Leistung aufweist als das einstufige Vorhersagemodell. Im Falle von NN und RF zeigen sich deutliche Verbesserungen bei der Kombination von SVR [48].

Da alle Indikatoren eine große Datenmenge analysieren, sind langfristige Investitionen für Anleger und Finanzanalysten eine Herausforderung. Ein Ansatz zur langfristigen Vorhersage des Aktienkurses kann auch als Klassifizierungsaufgabe behandelt und gelöst werden [42]. Basierend auf der Börsenperformance von 1739 Aktien in einem Jahreszeitraum werden Aktien in "gute" und "schlechte" Aktien unterteilt. Wenn der Aktienkurs über 10% lag, wird er als "gut" bezeichnet, andernfalls als "schlecht". Um eine gute Vorhersage zu erhalten, werden 11 relevante Merkmale ausgewählt und mehrere Algorithmen zum ML für die Aktienvorhersage angewendet. Die Ergebnisse dieses Tests zeigen, dass Random Forest mit einem Ergebnis von 0,751 gegen SVM und BN die beste Vorhersageleistung erzielte [42]. RNN Architecture wird auch angewendet, um den Börsenkurs vorherzusagen. Diese Studie vergleicht die Leistung verschiedener RNN-Architekturen bei der Vorhersage von Google-Aktienkursbewegungen. Der Vergleich erfolgt zwischen dem mehrschichtigen RNN, dem LSTM und der Gated Recurrent Unit (GRU) [49]. Die

Modelle werden anhand der historischen Daten der letzten fünf Jahre aus den Aktienkursen von Google trainiert. Diese Studie zeigt, dass LSTM bei einer Vorhersage von 5 Tagen eine Genauigkeit von bis zu 75% erreicht und in der Praxis erfolgreich angewendet werden kann [49]. Die Automatisierung des Aktienhandels mit Techniken des MLs umfasst das RL, um aus den Tricks des Marktes zu lernen und eine bessere Vorhersage der Aktienkursbewegung zu treffen. Diese Studie implementiert ein System zur Vorhersage des Aktienkurses basierend auf Finanznachrichten und dem Aktienwert [12]. Für die Sentimentanalyse wird ein R-CNN Model verwendet, das auf der Grundlage von Finanznachrichten eine Prognose der Aktienentwicklung erstellt. RL wird mit gesammelten Daten trainiert, um Aktien zu handeln. Das vorgeschlagene System ist geschult und arbeitet mit den Beständen eines Unternehmens. Dieses Papier zeigt, dass das RL genutzt werden kann, um Aktien zu handeln und eine bessere Entscheidung zu treffen. [12] Es wird vorgeschlagen, dass diese Architektur verwendet werden könnte, um mehrere Aktien zu handeln [12].

4.4 Anwendung auf das Liquiditätsrisiko

Das Liquiditätsrisiko ist ein finanzielles Risiko, das auftritt, wenn die Nachfrage nach Vermögenswerten sinkt. Die Unfähigkeit eines Unternehmens, einen wertvollen Vermögenswert ohne Kapital- oder Einkommensverlust in Bargeld umzuwandeln/ zu verkaufen. Die Unfähigkeit einer Bank oder eines Unternehmens, einen Vermögenswert zu verkaufen, zwingt sie dazu, die kurzfristigen finanziellen Anforderungen nicht zu erfüllen. Während des letzten Jahrzehnts wenden viele Forscher KI-Techniken an, um Liquiditätsrisikomanagement zu modellieren. ML kann verwendet werden, um das Liquiditätsrisiko, die verschiedenen Faktoren, die das Risiko verursachen, und die Verbindung zwischen ihnen zu messen. Diese Studie versucht, das Problem der Definition einer effizienten Lösung zu lösen, die das Liquiditätsrisiko managen und bewerten kann [67]. Dies ist eine Lösung, um das Liquiditätsrisiko anzugehen, und es ist in Teile zu unterteilen. In jedem Teil werden verschiedene Methoden des MLs verwendet, wobei ein NN und ein BN angewendet werden. NN Ansatz wird verwendet, um den allgemeinen Trend des Risikos zu finden und die beiden einflussreichsten Faktoren zu finden [67]. Die Wahrscheinlichkeit des Eintritts eines Liquiditätsrisikos, auch wenn nicht alle Indikatoren messbar sind, kann durch Anwendung von BN geschätzt werden, die auch die einflussreichsten Faktoren eines Risikos bestimmen. Diese Studie zeigt, dass die Implementierung der NN-

und BN-Technik in der Lage ist, die einflussreichsten oder kritischsten Indikatoren zu finden/ zu bestimmen und das Risiko durch eine funktionale Approximation bzw. eine Verteilungsschätzung zu messen [67].

4.5 Anwendung auf das Operationales Risiko

Das operationelle Risiko wird durch unzureichende oder fehlgeschlagene Prozesse, Personen und Systeme oder durch externe Ereignisse verursacht, worauf der Verlust von Ergebnissen folgt. In der Finanzbranche umfasst es andere Arten von Risiken wie Betrug, Sicherheit und Datenschutz. Während die Technologie im Fortschritt begriffen ist, digitalisieren sich die Finanzinstitute selbst. Das Betrugsrisiko ist ein häufiges Risiko für Banken. Aufgrund riesiger, massiver Finanzdatenbanken ist es schwierig, Betrug aufzudecken, und es ist eine Herausforderung für Banken [16]. Aufgrund der großen Datenmenge, ihrer Komplexität und der unterschiedlichen Arten von operationellen Risiken erforschen und entwickeln Finanzinstitute KI-Lösungen [16]. ML kann angewendet werden, um die Erkennung von Kreditkartenbetrug zu verbessern. In diesem Artikel wird ein Meta-Klassifikator vorgeschlagen, der die Leistung der bestehenden auf NN basierenden Betrugserkennung der kanadischen Bank verbessert [51]. Um den Meta-Klassifikator zu implementieren, verwendeten sie drei verschiedene ML-Techniken. Dies sind Entscheidungsbaum-, BN und kNN-Algorithmen. Sie wendet ihren vorgeschlagenen Klassifikator auf alle Transaktionen an, nachdem er vom bestehenden Betrugserkennungssystem der Bank analysiert wurde. Das Ergebnis dieser Arbeit zeigt, dass die Bereitstellung von Meta-Klassifizierern mit einem auf NN basierenden System zur Betrugserkennung bei Banken deren Leistung um bis zu 28% verbessert [51].

Geldwäsche ist auch eine Art operationelles Risiko, das sich negativ auf den Finanzsektor und die wirtschaftliche Entwicklung auswirkt [20]. Mehrere ML-Algorithmen bauen ein System zur Bekämpfung von Geldwäsche auf, um verdächtige und schlechte Transaktionen in riesigen Finanzdatenbanken zu erkennen. Die Ressourcen zur Anwendung von ML gegen Geldwäsche sind begrenzt. In diesem Artikel wird jedoch die Anwendung mehrerer Algorithmen für ML zur Erkennung von Geldwäsche untersucht. Unter Verwendung der Transaktionsdaten der US-Finanzinstitute wurden die fünf wichtigsten Algorithmen für ML, Bayes Logistic Regression, Entscheidungsbaum, RF, SVM und NN untersucht [76]. Die Leistung von NN macht es

zu einem starken Konkurrenten bei der Verwendung von Klassifizierungsalgorithmen, aber auch SVM und RF lieferten im Vergleich zu herkömmlichen logistischen Regressionsmodellen eine bessere Leistung. Ein wichtiger Schlüssel für Trainingsdaten ist es, Bias und Fehler zu vermeiden, damit das Modell eine bessere Leistung erzielt. Die Entwicklung eines effektiven Anti-Geldwäschesystems ist schwierig, da es eine Zusammenwirkung vieler Komponenten wie unabhängige Überprüfung, Audit und Mitarbeiterschulung ist [76].

Kreditkartenbetrug ist auch ein häufiges Problem bei Offline- und Online-Zahlungen. Die Zahlung mit Kreditkarte nimmt von Tag zu Tag zu, sodass die Gefahr von Betrug zunimmt. Aufgrund der großen Menge an Transaktionsdaten ist es schwierig, Betrug zu analysieren und aufzudecken. Banken setzen ML-Algorithmen ein, um ein effizienteres und zuverlässigeres System zur Erkennung von Kreditkartenbetrug zu entwickeln. Die Verwendung der NN-Anwendung weist ein großes Potenzial für Klassifizierungsprobleme auf, um Betrug zu erkennen und zu verhindern. In diesem Artikel wird die Anwendung für NN gegen Kreditkartenbetrug vorgestellt, die auf historischen Transaktionsdaten basiert [26]. Es werden NN verwendet, da es das Verhalten und die Verwendungsmuster von Karten erkennen kann. Diese Anwendung erreicht eine hohe Genauigkeit von 94,2% [26]. DL zeigt und richtet die Aufmerksamkeit von Aktienhandelspraktikern und -forschern auf sein hochentwickeltes, effizientes und robustes Finanzzeitreihen-Vorhersagemodell. In diesem Beitrag wird eine Deep Learning-basierte Trader-Agent-Anwendung zur Unterstützung von Handelsentscheidungen im Forex-Handel vorgeschlagen. Das Ergebnis dieser Arbeit ist, dass die Leistung des DL-Modells eine bessere Leistung als CNN, RNN bei der Zeitreihenvorhersage für den Devisenhandel erzielt [37].

4.6 Anwendung auf das Forex Risiko

Der Forex-Markt hat ein großes Gewinnpotential. Mit einem durchschnittlichen täglichen Handel von 5 Billionen US-Dollar und einem 10-fachen des globalen Börsenvolumens ist dieser Markt ein riesiges Potenzial für Händler auf der ganzen Welt [54]. Es gibt ein enormes Gewinnpotential, aber es ist auch ein enormes Risiko einzugehen. Aufgrund der starken Schwankungen der Trends ist dieser Markt eine große Herausforderung für Händler und Unternehmen. Um eine bessere Leistung bei Vorhersage zu erzielen, werden viele verschiedene ML-Techniken angewendet,

um den Forex-Handel zu automatisieren und eine bessere Vorhersage durchzuführen.

Diese Studie untersucht und vergleicht die am häufigsten verwendeten Techniken des MLs für die Forex-Vorhersage. Das vorgeschlagene System wird aus historischen Daten eines Währungspaares trainiert, um die Vorhersage zu treffen. Es werden drei Arten von Technologien für ML, um anhand historischer Daten des Währungspaares EURO gegenüber US-Dollar Vorhersagen zu treffen: SVM, LSTM und GRU getestet [50]. Im Vergleich zu LSTM ist GRU recheneffizienter und weniger komplex, aber die Vorhersageleistung ist paarweise [50].

RL ist auch eine andere weit verbreitete Technik zur Implementierung von Anwendungen für automatisierte Handelssysteme. In diesem Artikel wird ein System, das auf adaptivem RL basiert implementiert [19]. Hierbei handelt es sich um eine Technik des RLs, die mit dynamischer Optimierung kombiniert wird. Die dynamische Optimierung wird verwendet, um die Risikomanagementebene dynamisch zu optimieren, um den Nutzen eines Händlers zu maximieren. In Bezug auf den historischen Forex-Hochfrequenzdatensatz schnitt das vorgeschlagene System sehr gut ab [19].

Kapitel 5

RNN-Ansatz bei der Vorhersage von Aktienkursen

Der RNN-Ansatz zeigt die Anwendung von LSTM zur Vorhersage des Aktienkurses. Eigenschaften von LSTM wie der Zustand des Netzs, wiederkehrendes Lernen und der Umgang mit Zeitreihendaten machen es für die Aktienpreisvorhersage anwendbar. Um die nächste Kursbewegung vorherzusagen, werden alle vorherigen Informationen benötigt, so dass die nächste Bewegung von den Daten der Vergangenheit abhängt. Wie die LSTM funktioniert, wird im vorigen Kapitel 3.9 erläutert.

5.1 Das Ziel dieses Ansatzes

Das Ziel dieser Arbeit ist die Entwicklung eines LSTMs zur Vorhersage des Aktienkurses eines Unternehmens. Dieses Netz wird durch historische Daten der beiden Unternehmen Google und Microsoft trainiert und vorhersagt den jeweils nächsten Kurs. Am Ende werden alle Parameter der LSTM beschrieben und ihre Auswirkungen auf das Ergebnis. Um ein gut trainiertes Modell zu erreichen, wird der Datensatz ebenfalls in drei Teile aufgeteilt, wobei 60% des Datensatzes Trainingsdaten, 20% des Datensatzes Validierungsdaten und der Rest 20% für das Testen der LSTM-Modell sind.

Der Aufbau eines Vorhersagemodells erfordert drei Phasen:

1. Training des Modells auf der Basis historischer Daten

2. Validierung des Modells, Optimierung der Modellparameter, um eine gute Leistung und Vorhersage zu erreichen
3. Testen des trainierten Modells

5.2 Technologien

Tensorflow

Tensorflow ist eine Bibliothek für numerische Berechnungen, um ML schneller und einfacher zu gestalten [68]. Es ist eine von Google entwickelte Bibliothek. Sie wird zum Training und zur Entwicklung von NNen und tiefen NNen für verschiedene Zwecke verwendet. Sie ermöglicht Central Processing Unit (CPU)- und Graphics Processing Unit (GPU)-Berechnungen [68].

Matplotlib

Matplotlib ist eine Python-Bibliothek, die zur Erstellung von Grafiken, Diagrammen und Zahlen verwendet wird [41].

Numpy

Numpy ist eine Python-Bibliothek, die es Datenwissenschaftlern ermöglicht, mit mehrdimensionalen Array-Objekten zu arbeiten. Mit Numpy können sowohl numerische als auch logische Operationen durchgeführt werden.

Pandas

Pandas ist eine leichte Python-Bibliothek für den Umgang mit Datenstruktur- und Datenanalyse-Tools [52]. Sie ist ein Manipulationstool und bietet zwei grundlegende Elemente: DataFrame und Series. Sie ermöglichen die Manipulation und das Speichern von Daten in Tabellen [52].

Scikit-learn

Scikit-learn ist ebenfalls eine Bibliothek für ML. Sie bietet eine breite Palette von Algorithmen für ML, wie z.B. überwachtes und unüberwachtes Lernen. Sie bietet Funktionen wie Klassifikation, Regression, Clustering, Datenvorverarbeitung, Datennormalisierung und Konstruktionsmodelle [59].

5.3 Implementierung

Ziel der Arbeit ist die Implementierung eines Vorhersagemodells auf der Basis von LSTM. Für das Training des Modells werden die vergangenen Kursdaten von Google verwendet. Der Datensatz wird von Yahoo Finance heruntergeladen, die Daten sind vom 2017 – 01 – 23 bis zum 2020 – 01 – 23. Im folgenden Abschnitt wird die Implementierung des LSTM für Aktienkursvorhersagen Schritt für Schritt erklärt. Der Code für jeden Abschnitt der Implementierung ist im Anhang zu finden.

Daten laden

Der erste Schritt zur Umsetzung dieses Ansatzes ist für einige Bibliotheken erforderlich. Die oben genannten Bibliotheken müssen importiert werden. Der entsprechende Code ist im Anhang zu finden. Die Daten werden in einer csv-Datei gespeichert. Die Daten werden geladen und in einen Dataframe transformiert. Alle Daten im Dataframe werden angezeigt und nach Datum aufsteigend sortiert.

	date	open	high	low	close	adj_close	volume	month
0	2017-01-23	807.250000	820.869995	803.739990	819.309998	819.309998	1963600	1
1	2017-01-24	822.299988	825.900024	817.820984	823.869995	823.869995	1474000	1
2	2017-01-25	829.619995	835.770020	825.059998	835.669983	835.669983	1494500	1
3	2017-01-26	837.809998	838.000000	827.010010	832.150024	832.150024	2973900	1
4	2017-01-27	834.710022	841.950012	820.440002	823.309998	823.309998	2965800	1

Abbildung 5.1: Tabellarische Darstellung der Google-Aktienkursdaten

Zum Aufbau dieses Modells wird der Datensatz in drei Teile aufgeteilt, wobei 60% des Datensatzes Trainingsdaten, 20% Validierungsdaten und der Rest 20% zum Tes-

ten des Modells sind. Um eine bessere Vorstellung von der Tendenz der Aktien zu bekommen, werden die Daten in einem Graphen visualisiert, wobei das Attribut "date" als Index verwendet wird. Die Abbildung 5.2 zeigt die Visualisierung von Daten aus Google-Aktien. Dieser Graph zeigt einen allgemeinen Überblick über die Tendenz der Aktienpreise im Laufe der Zeit. Der Graph hat Höhen und Tiefen, und die Analyse der vergangenen Daten durch das Erkennen von Mustern ist der Kern der Vorhersage.



Abbildung 5.2: Graphische Darstellung der Google-Aktienkursdaten

Code 9.2

Datenvorbereitung

Ein wichtiger Schritt beim Aufbau eines Vorhersagemodells ist die Datennormalisierung. Das bedeutet, die Daten so zu transformieren, dass es einfach ist, mit ihnen zu arbeiten. Wie aus der Tabelle hervorgeht, haben die Daten "adj_close" einen großen Bereich von 819,3 bis 1484,4. Die Datennormalisierung hat einen großen Einfluss auf die Leistung, die Ergebnisse des Netzes und die Berechnungen [61]. Es gibt viele verschiedene Algorithmen zur Datennormalisierung, aber die am häufigsten verwendeten sind MinMaxScaler und StandardScaler. Bei diesem Ansatz wird als Normierungsmethode MinMaxScaler verwendet.

1. MinMaxScaler skaliert alle Merkmalswerte des Datensatzes in einem Bereich von $[0, 1]$. Der kleinste Merkmalswert ist 0 und der größte ist 1 [61].

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (5.1)$$

Code 9.3

Erstellung des Modells

LSTM besteht aus vielen Schichten, daher wird *Sequential()* zum Aufbau verwendet. Das Netz besteht aus zwei LSTM-Schichten. Ein großes Problem im NN ist die Überanpassung des Modells. Das hat einige Konsequenzen wie langsame Leistung und falsche Ergebnisse bei neuen unbekannten Daten. Auf jede LSTM-Schicht folgt eine Dropout-Schicht. Die Dropout-Schicht ist eine einfache Möglichkeit, das Modell vor einer Überanpassung zu schützen [62]. Die Grundidee der Dropout-Schicht besteht darin, Einheiten und Ein- und Ausgangsverbindungen während der Trainingsphase aus dem Netz zu entfernen. Nach Anwendung der Dropout-Schicht besteht das Netz aus Einheiten, die nach der Dropout-Schicht verblieben sind. Es nimmt einen Wert an, der zwischen $[0, 1]$ liegt. Die letzte Schicht des Netzes ist eine dichte Schicht. Sie ist mit jedem Neuron der vorhergehenden Schicht verbunden. Die Rolle der dichten Schicht besteht darin, einen mehrdimensionalen Eingang in einen eindimensionalen Ausgang zu transformieren.

LSTM benötigt einige Parameter und jeder von ihnen hat Auswirkungen auf das Modell und das Ergebnis.

1. *lstm_units*: Anzahl von Einheiten pro Schicht
2. *input_shape*: Eingabedaten
3. *loss*: Die Verlustfunktion Root Mean Square Error (RMSE)
4. *optimizer*: Adaptive Moment Estimation (adam)
5. *epoch*: Anzahl der Iterationen, wie oft das Netz trainieren wird
6. *batch_size*: Anzahl der Blöcke der Eingabedaten
7. *dropout_prob*: Dropout Wahrscheinlichkeit

Code 9.4

Vorhersage auf Validierungsdaten

Nachdem das Modell auf Trainingsdaten trainiert wurde, ist der nächste Schritt die Vorhersage *est*. Der folgende Code zeigt, wie das Modell eine Vorhersage auf Bewertungsdaten macht. Nach der Vorhersage werden die Werte investiert, um sie in unskalierte Werte *est_inv* zu transformieren. Die Variable *est_inv* ist der vorausgesagte Wert und die andere der tatsächliche Wert *y_val_inv* des Aktienkurses auf dem Validierungsdatensatz. Danach können die vorausgesagten Werte in einem Graphen dargestellt oder in der Tabelle aufgelistet werden.

Code 9.5

Berechnung der Vorhersagefehlerrate

RMSE ist eine gute Funktion zur Berechnung der Fehlerrate. Sie wird verwendet, um zu messen, wie gut eine Funktion zu den Daten passt. Für die Vorhersage zeigt sie, wie gut das Modell bei der Vorhersage auf der Grundlage historischer Daten funktioniert. Sie berechnet den Abstand zwischen den vorhergesagten Datenpunktwerten und den tatsächlichen Datenpunktwerten. Es handelt sich um die Summe der Entfernung zwischen den Punkten geteilt durch die Anzahl der Punkte. Am Ende ist die Summe geteilt durch die Anzahl der Datenpunkte und das Quadrat der Wurzel aus dem Ergebnis.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y'_n - y_n)^2} \quad (5.2)$$

Je kleiner die Vorhersagefehlerrate, der Wert von RMSE, desto besser ist die Leistung des Modells bei der Vorhersage, d.h. die Vorhersage ist besser. Ein Nachteil davon ist, dass es nicht auf Daten mit unterschiedlicher Skalierung funktioniert. In diesem Fall liefert die Methode ein falsches Ergebnis [60].

Mean Absolute Percentage Error (MAPE) ist eine weitere Methode zur Messung der Vorhersagegenauigkeit einer Vorhersage. Die basiert auf dem prozentualen Fehler. Sie berechnet die Summe des Absolutwerts des Abstands zwischen den Punkten geteilt durch die Anzahl der Punkte. Am Ende wird der Prozentsatz davon ermittelt.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y'_n - y_n}{y_n} \right| \quad (5.3)$$

Code 9.6

5.4 Optimierung von Hyperparametern

Dieser LSTM-Ansatz basiert auf einigen Ausgangswerten. Dieser Ansatz liefert nicht das beste Ergebnis bei der Vorhersage des Aktienkurses. Die Leistung der LSTM basiert auf den Parametern wie epoch Größe, Batchgröße, dropout Wahrscheinlichkeit, Anzahl der LSTM-Einheiten und den Optimierungsmethoden. Um den besten Parameter zu finden, muss man RNN mit verschiedenen Parametern ausprobieren. Im folgenden Teil wird die Rolle der einzelnen Parameter erläutert.

Anzahl von LSTM-Schichten

Anzahl der LSTM-Schichten ist wichtig, um Vorhersagen auf der Grundlage von Trainingsdaten treffen zu können. Die Anzahl der Schichten hängt von Trainingsdaten ab. Es geht also darum, die optimale Anzahl von Schichten zu finden. Die am häufigsten angewandte LSTM-Modelle haben 1, 2, 3 und 4 Schichten. Der eigentliche Ansatz hat zwei Schichten. Die Anzahl der Schichten beeinflusst die Leistung des Netzs.

Anzahl der recurrent Einheiten

Die Anzahl der wiederkehrenden Einheiten im RNN hat einen großen Einfluss auf das Modell. Die große Anzahl von Einheiten kann zu einer Überanpassung des Netzs an die Trainingsdaten führen, aber auf der anderen Seite kann eine geringe Anzahl zu einer Unteranpassung des Netzs führen. Der beste Weg ist also, eine optimale Anzahl von Einheiten zu finden. Der Vorschlag besteht darin, die beste Leistung aus dem Netz mit 10, 50, 64 oder 128 Einheiten zu finden.

Batchgröße

Die Batchgröße ist die Anzahl der Trainingsdaten, mit der das Netz vor der Aktualisierung des internen Zustands trainiert wird. Die Batchgröße hängt vom Datensatz

und der Rechenleistung ab, welche auf das Endergebnisse auswirkt. Eine zu kleine Batchgröße bedeutet Überanpassung und eine zu große Unteranpassung, da die Gewichte zu hoch werden können. Im eigentlichen Ansatz wird Batchgröße 1 verwendet. Die 1 ist möglicherweise nicht die optimale Größe. Um die optimale Größe zu finden, sollten die Größe mit folgenden Werten 1, 2, 4, 8, 16, 32, 64 und 128 ausprobiert.

Epoch

Epoch besteht aus einer oder mehreren Batches. Es bedeutet, wie viele Male das Netz die Trainingsdaten durchläuft, es handelt sich also um einen Vorwärts- und einen Rückwärtsdurchlauf aller Trainingsdaten. Dies wirkt sich auf den Gewichtswert jedes Neurons aus, da in jeder Epoch-Iteration der interne Modellzustand aktualisiert wird. Eine große Anzahl von Epochen kann zu einer Überanpassung führen, da das Netz sehr gut zu den Trainingsdaten passt. Eine optimale Anzahl von Epoch bedeutet ein besseres Ergebnis. Das Ergebnis von RSME in den Validierungs- und Testdaten für die verschiedenen Epoch zeigt, welche Epochzahl für das Modell optimal ist. Die zu bewertenden Epochzahlen sind standardmäßig 1, 10, 20, 30, 40, 50, 100.

Dropout

Dropout ist ein weiterer wichtiger Parameter im NN. Er verhindert eine Überanpassung des Modells, indem er Einheiten und Verbindungen entfernt. Er nimmt die Werte von $[0, 1]$, wobei der Wert 0 für "kein Dropout" und 1 für "keine Verbindung" steht. Die häufigsten Dropout-Werte sind 0.5, 0.6, 0.7, 0.8, 0.9, 1.

Optimierer

Es gibt verschiedene Arten von Optimierern für neuronale Netze. Die Verlustfunktion ist die Differenz zwischen dem vorhergesagten Wert und dem tatsächlichen Wert. Die Verlustfunktion wird durch den Gewichtswert der Neuronen beeinflusst. Die Rolle des Optimierers besteht also darin, den besten Wert der Gewichtung zu finden, um die Verlustfunktion zu minimieren. Es gibt eine Reihe von Optimierern, die angewendet werden können. Die am häufigsten verwendeten Optimierer in der

LSTM sind: adam [36], adamax [58], nadam [58] usw. In diesem Modell wird der adam-Optimierer verwendet, um die Verlustfunktion zu minimieren. Indem man also jede von ihnen auf den Datensatz anwendet, um den am besten funktionierenden Optimierer zu finden.

5.5 Ergebnisse

Das Ziel dieser Implementierung ist es, ein Modell für die Vorhersage von Aktienkursen zu entwickeln. Es gibt viele Möglichkeiten der Analyse, um den Aktienkurs vorherzusagen, und eine davon ist die technische Analyse. Sie bedeutet, dass der Aktienkurs auf der Grundlage historischer Daten vorhergesagt werden kann. Die Vorhersage des Aktienkurses eines Unternehmens ist aufgrund von Zeitreihen eine Herausforderung für viele Methoden des MLs. In dieser Arbeit wird ein LSTM-Netz zur Vorhersage verwendet, das eine Art RNN ist (siehe Kapitel 3.8 und 3.9). Die Aktienvorhersage ist abhängig von früheren Daten und Trends in den Daten. Daher ist LSTM ein Netz, das für dieses Problem adaptiv ist. LSTM ist auch das am häufigsten verwendete Netz im Finanzsektor, da es die beste Leistung bei Zeitreihendaten aufweist. Es versucht, Muster und Trends in den Daten zu finden und dann eine Vorhersage zu treffen, wie sich der Aktienmarkt bewegen wird. In dieser Arbeit werden die historischen Daten der beiden Unternehmen Google und Microsoft verwendet. Das Modell wird auf der Basis dieser Daten trainiert. Basierend auf dem Lernen in der Trainingsphase wird der nächste Preis vorausgesagt.

Die Leistung des Netzes ist bei verschiedenen Daten unterschiedlich. Aber es gibt viele andere Faktoren, die sich auf die Leistung und die Vorhersagegenauigkeit auswirken. Im obigen Abschnitt 5.4 wird einige Parameter besprochen, die optimiert werden müssen, um den optimalen Wert für bestimmte Aktiendaten zu finden. In der Tabelle 5.1 sind einige Experimente mit verschiedenen Epochenzahlen und Batchgrößen aufgeführt. Die Ergebnisse der Tabelle 5.1 zeigen, dass das Modell bei verschiedenen Daten und verschiedenen Parametern unterschiedliche Leistungen erbringt. Die optimalen Parameter für das Modell sind die Parameter, bei denen das Modell die geringste Vorhersagegenauigkeit aufweist. In diesem Experiment schneidet das Modell besser ab, wenn die Epochenzahl 20 und die Batchgröße 1 beträgt. Um jedoch das optimale und gute Vorhersagemodell zu haben, müssen al-

LSTM-Hyperparameter				
Unternehmen	Epoch	Batchgröße	RMSE	MAPE
Google	20	1	19.732	1.249%
Google	20	2	24.858	1.813%
Google	20	4	20.952	1.397%
Google	20	8	33.045	2.481%
Google	30	1	19.853	1.267%
Google	30	2	20.186	1.332%
Google	30	4	22.238	1.557%
Google	30	8	23.246	1.604%
Google	50	1	19.891	1.282%
Microsoft	20	1	1.916	1.287%
Microsoft	20	2	2.391	1.731%
Microsoft	20	4	2.828	1.858%
Microsoft	20	8	4.830	3.575%
Microsoft	30	1	2.226	1.596%
Microsoft	30	2	2.294	1.604%
Microsoft	30	4	2.299	1.535%
Microsoft	30	8	2.966	1.969%
Microsoft	50	1	2.560	1.811%

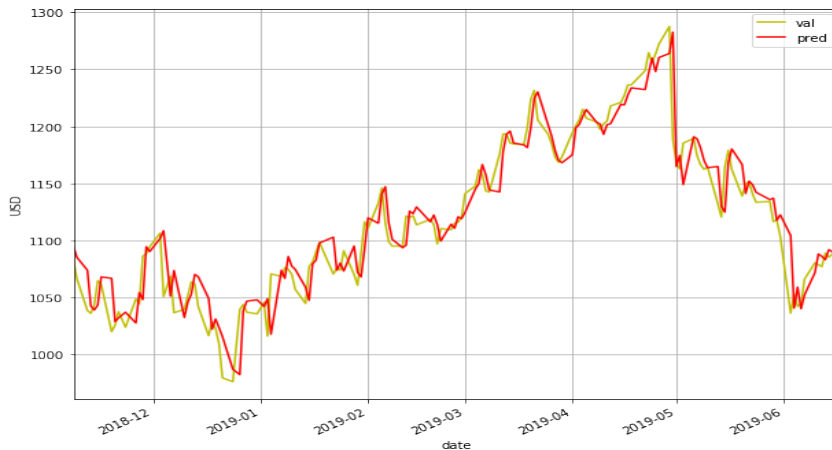
Tabelle 5.1: Statistiken für die Vorhersage von Validierungsdaten.

le Kombinationen der möglichen Parameter ausprobiert und die optimalen Werte gefunden werden, bei denen der RMSE am kleinsten ist.

Um einen besseren Überblick über die Vorhersage zu erhalten, wie das Modell Leistung erbringt, visualisieren die folgenden Charts 5.5 und 5.8 die Vorhersage des Modells mit den gleichen Parametern auf den Validierungs 5.5- und Testdaten 5.8.

Aus den Charts ist leicht zu erkennen, dass das Modell die Fähigkeit besitzt, Muster und Trends zwischen den Daten zu erkennen. Die Vorhersagegenauigkeit ist nicht die beste, die erreicht werden könnte, aber das LSTM-Modell zeigt das Potenzial der Identifizierung von Höhen und Tiefen des Aktienmarktes. Sowohl bei Google als auch bei Microsoft zeigt das Modell in beiden Fällen unterschiedliche Leistungen. Es basiert auf der Art der Daten, der Komplexität der Erkennung von Trends usw.

Ein weiterer Faktor, der sich auf die Vorhersage auswirkt, ist die Rechenleistung. Um ein solches Modell zu trainieren, das eine riesige Datenmenge analysieren, verstehen und lernen muss, benötigte es genügend Rechenleistung. Die Rechenleistung beeinflusst die Parameter wie LSTM-Einheiten, Epocheanzahl und Batchgröße.



0.5

Abbildung 5.3: Google



0.5

Abbildung 5.4: Microsoft

Abbildung 5.5: Vorhersage des Aktienpreises auf Validierungsdaten

ße. Wenn es sich um einen normalen Computer handelt und eine große Datenmenge, die das Netz mit allen Daten auf einmal versorgt, wird das Ergebnis nicht wie erwartet sein.

Die Über- und Unteranpassung des Modells ist ebenfalls ein weiteres Problem in ML. Die Werte aus den Tabellen zeigen, dass das Modell mit hohen RMSE-Werten unterangepasst ist, so dass es falsche Aktienkurswerte vorhersagt. Auf der anderen Seite führt die Überanpassung des Modells zu einer falschen Vorhersage, da das Modell zu sehr an die Trainingsdaten angepasst ist. Das optimale Modell ist das Modell mit dem optimalen RMSE-Wert. Das bedeutet, dass sein Wert nicht zu hoch, aber auch nicht zu niedrig ist. In diesem Fall könnte eine Über- und Unteranpassung des Modells vermieden werden.



Abbildung 5.6: Google

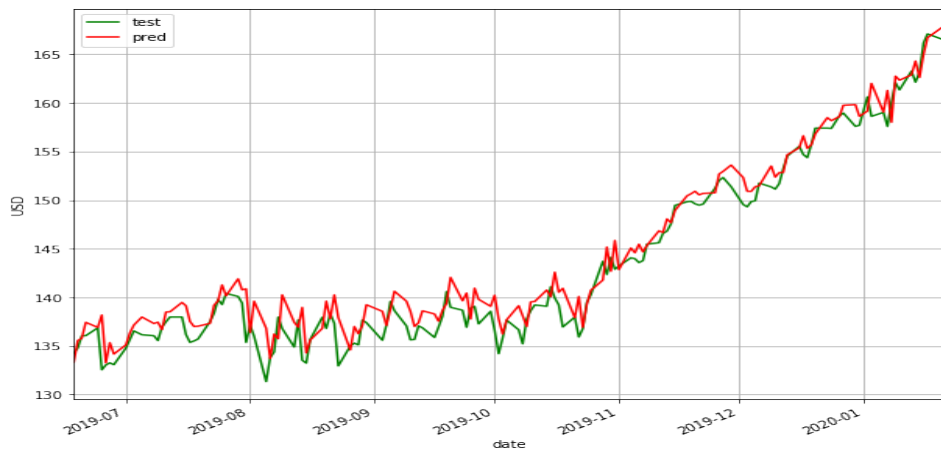


Abbildung 5.7: Microsoft

Abbildung 5.8: Vorhersage des Microsoft-Aktienkurses auf Testdaten

Kapitel 6

KI-Startups in der Finanzindustrie

Dieses Kapitel listet die innovativsten KI-Startups laut Forbes und Bloomberg auf, die in den Bereich Finanzdienstleistungen und Risikomanagement beschäftigen.

ZestFinance

ZestFinance ist eines der innovativsten Finanzunternehmen, das den eigentlichen Finanzdienstleister durch KI-Anwendungen transformiert. Es bringt die Kraft des MLs für die Kredit- und Risikomodellierung für Kreditgeber [70]. Das 2009 gegründete Unternehmen setzt auf die Transformation des Kredit-Underwritings für Kreditgeber. Sie entwickeln eine KI-basierte Lösung Zest Automated Machine Learning (ZAML). Entsprechend ihrer Ressourcen ist ZAML eine KI-basierte Lösung für die Kredit- und Risikomodellierung mit umfassender Erklärbarkeit. 2017 kündigten sie ihre ZAML-Plattform an, die prediktive Analytik und Funktionen zur Verarbeitung natürlicher Sprache kombiniert [70].

DataRobot

DataRobot ist ein weiteres KI-Startup, das Datenwissenschaftlern eine KI-basierte prediktive Analytik-Plattform bietet [22]. Ihre Plattform bietet hervorragende Leistung beim Erstellen und Bereitstellen genauer Vorhersagemodelle. Verschiedene Unternehmen wie Finanzinstitute verwenden ihre bereitgestellte Plattform, um ein genaues Vorhersagemodell zu erstellen, das sich mit Themen wie Kreditbetrugserkennung, Vermögensverwaltung, Kreditvergabe und mehr befasst [22].

LendBuzz

LendBuzz, ein 2015 gegründetes Bostoner FinTech-Unternehmen, verändert die Art und Weise der Kreditvergabe. Trotz fehlenden Kreditgeschichten innerhalb der USA bieten sie Autokredite für Immigranten und internationale Studenten an [38]. Im Vergleich zu Banken oder anderen Finanzinstituten, die sich auf Kreditbewertungen verlassen, um Kredite zu genehmigen oder zu verweigern, nutzen sie die Vorteile KI, um Kredite mit niedrigen Kreditprofilen zu vergeben [38]. Um die Kreditwürdigkeit zu bestimmen, nutzen sie die Hintergrundinformationen der Antragsteller wie Bildungshintergrund, Beschäftigungshintergrund, etc.

Numerai

Numerai ist ein Crowdsourcing-Hedgefonds. Es ist das erste Startup, das KI an der Börse einsetzt. Die zur Verfügung stehende Plattform und die Daten sind leicht zugänglich und jeder kann die Beispieldaten herunterladen und das Modell trainieren [43]. Die Numerai-Plattform bietet Datenwissenschaftlern mit Finanzhintergrund die Möglichkeit, ihre Vorhersagemodelle zu erstellen [43]. Jeder Mitwirkende wird mit dem Plattform-Kryptografietoken Numeraire bewertet und belohnt.

Epoque

Epoque-plus ist ein im Handel tätiges Unternehmen. Es ist ein vollständig KI-Handelssystem, das alles in der Handelsbranche verändert. Um Bestellungen zu kontrollieren, zu identifizieren, Risiken zu überwachen, die Leistung zu steigern, Trades zu erstellen und zu schließen, den Markt zu analysieren und die aktiven Bestellungen zu verwalten, wendet das System in Echtzeit maschinelle Lernalgorithmen auf eine große Datenmenge an [23]. Laut ihrer Website verwenden sie KI, um alles in ihrem Handelssystem zu verwalten [23].

Trading technologies

Tading Technologies entwirft und entwickelt professionelle Handelssoftware, Infrastruktur und Datenlösung für verschiedene Arten von Benutzern. Sie stellen auch ihre eigene KI-basierte Handelsplattform zur Verfügung, die das Muster von Big Data

in Echtzeit analysiert und findet [69]. Sie kombinieren verschiedene fortschrittliche Techniken des MLs und der Hochgeschwindigkeitsdatenverarbeitung, um den Kunden ein besseres Handelserlebnis zu bieten [69].

Auquan

Auquan ist ein 2017 gegründetes KI-Startup mit Sitz in London, das in der Handelsbranche tätig ist. Es bietet eine Wettbewerbsplattform für alle Datenwissenschaftler, um erfolgreich algorithmische Handelsstrategien zu entwerfen [11]. Es wandelt die Finanzdaten in Daten-, Mathematik- und ML Probleme um, indem es Finanzinformationen extrahiert und auf die Daten verschiedene ML-Techniken anwendet, um Muster zu identifizieren und Modelle zu erstellen [11]. Forbes hat es in die "Top 15 der zu beobachtenden Unternehmen für maschinelles Lernen in Europa" eingestuft.

Causalens

CausaLens ist ein KI-Startup mit Sitz in London und wurde 2016 gegründet. Es entwickelt und implementiert verschiedene Technologien für ML, um Vorhersagen für Echtzeit-Seriendaten zu ermöglichen [15]. Sie bauen komplexe und dynamische Vorhersagemodelle mit KI-Technologie. Laut ihrer Homepage bieten sie automatisierten ML-Dienst, Echtzeit-Marktvorhersagen in einem komplexen und dynamischen System [15]. Dieses Unternehmen wird auch von Forbes als "die 15 besten Unternehmen für maschinelles Lernen in Europa" eingestuft.

Aladdin (BlackRock)

Aladdin ist ein Betriebssystem für Investmentprofis, das von Blackrock, der größten Vermögensverwaltungsgesellschaft der Welt, unterstützt wird. Die Aladdin-Plattform bietet alle für Investoren erforderlichen Services/ Tools auf einer einzigen Plattform [4]. Es kombiniert verschiedene Risikoanalysetechniken mit Portfolio-Management-, Handels-, Betriebs- und Buchhaltungs-Tools. Die Aladdin-Engine wird von KI und ML-Technologie angetrieben [4]. Die Aladdin-Plattform verwaltet ein Vermögen von 18 Billionen US-Dollar und ist damit die riesige Vermögensverwaltungsplattform der Welt [4].

Alphasense

AlphaSense ist ein KI-Startup, das im Jahr 2016 gegründet wurde. Sie bieten eine Suchmaschine für die Finanzbranche, indem sie die KI-Technologie nutzen [10]. KI- und Natural Language Processing (NLP)-Technologie werden verwendet, um das Keyword innerhalb der Inhaltsquelle zu analysieren und das Ergebnis dem Kunden bereitzustellen [10]. Diese Plattform bietet ihren Kunden private und öffentliche Informationen über Unternehmen, so dass eine schnelle finanzielle Entscheidung getroffen werden kann [10].

Kavout

Kavout, gegründet im Jahr 2015, ist ein KI-Startup, das einen KI-angetriebenen Aktienranker bereitstellt. Um das Muster auf dem Finanzmarkt in einem riesigen unstrukturierten Datensätze zu identifizieren, verwendet die Plattform maschinelles Lernen und quantitative Analysen [33]. Die Lösung heißt *K* Score und liefert Finanzmarktergebnisse in Echtzeit [33].

Alpaca

Alpaca wurde 2015 gegründet und ist Asiens führendes KI Fintech-Startup. Sie kombinieren DL und Big Data-Technologie, um kurz- und langfristige Vorhersageanwendungen zu entwerfen, zu erstellen und bereitzustellen [7]. Es bietet seinen Kunden eine Software-as-a-Service Lösung. AlpacaForecast ist eine KI-basierte Plattform, die die Börsenkursbewegung für kurz- und langfristige Investitionen in Echtzeit prognostiziert [7]. AlpacaRadar verwendet die historischen Börsendaten, um das Finanzmuster zu identifizieren und die Kursbewegungen an den Börsen in Echtzeit vorherzusagen. Es bietet Bloomberg auch eine Anwendung, um Benutzern den AlpacaForecast KI Prediction Market zur Verfügung zu stellen, der kurzfristige Vorhersage in Echtzeit vorhersagt [7].

ORX

ORX ist ein KI-Startup, das sich auf das operative Risikomanagement in den Finanzdienstleistungen konzentriert. Es bietet eine KI-basierte Plattform, die eine

große Menge von Echtzeitdaten analysiert, indem die Technologie des maschinellen Lernens verwendet wird, um das operationelle Risiko und das nicht finanzielle Risiko zu messen und zu verwalten [46].

Feedzai

Feedzai ist ein 2015 gegründetes KI-Startup, das KI-basierte Betrugsbekämpfungslösungen anbietet. Seine Aufgabe ist es, das Bankgeschäft und den Handel sicherer zu machen. Daher werden große Datenmengen und Technologien für ML kombiniert, um betrügerische Zahlungsvorgänge zu identifizieren. Es bietet Banken und anderen Finanzunternehmen eine KI-basierte Lösung, um sie vor Betrug und Geldwäsche zu schützen [24].

Vectra AI

Vectra AI ist ein 2010 gegründetes KI-Startup, das KI einsetzt, um Cyberangriffe in Echtzeit zu erkennen und darauf zu reagieren. Es bietet eine KI-basierte Lösung, um Clouds, Rechenzentren und Unternehmensnetzwerke gegen Cyberangriffe zu unterstützen. Laut seiner Homepage ist es eine Partnerschaft mit vielen Banken und Finanzinstituten, um Cyberangriffe in ihrem System aufzudecken [72].

Bleckwen

Bleckwen, ein französisches Startup, wurde 2016 gegründet und bietet KI-basierte Lösungen zur Bekämpfung von Geldwäsche und Finanzkriminalität. Es dient vielen Banken und Finanzinstituten zum Schutz vor Finanzverbrechen. Es kombiniert KI, ML und Big-Data-Technologien, um ein schnelleres und effektiveres Betrugserkennungssystem aufzubauen [14].

Kensho

Kensho wurde 2012 gegründet und ist ein Startups im Bereich KI. Es bietet eine KI-basierte Lösung, die Banken und Kunden bei Finanzverbrechen wie der Bekämpfung von Geldwäsche, Kreditbetrug, Steuerbetrugsermittlung usw. unterstützt

[35]. Forbes hat sie als eines der "Top 50 Fintech-Unternehmen weltweit" bezeichnet [35].

Kapitel 7

Herausforderungen des MLs und der KI

Heutzutage investieren viele Unternehmen das Geld in die Erforschung der Vor- und Nachteile der Integration von ML und KI in ihr Unternehmen. Diese Technologie bringt jedoch einige wichtige Probleme mit sich, die behoben werden müssen, bevor sie in eine konkrete Lösung umgesetzt werden kann.

7.1 Datenverfügbarkeit

KI ist eine Technologie, die auf großen und massiven Datenmengen basiert. Daten sind der Schlüsselfaktor in der KI, da diese Technologie von ihnen lernt und auf der Grundlage von erlerntem Wissen intelligente Entscheidungen trifft. Das bedeutet, dass die Verfügbarkeit von Daten bei dieser Technologie eine wichtige Rolle spielt. Die beiden am häufigsten verwendeten Programmiersprachen bei der Entwicklung und Implementierung von Anwendungen für KI und maschinelles Lernen sind: Python und R. Diese Programmiersprachen werden häufig verwendet, da sie einen breiten Lesebereich für alle Arten von Daten von Excel bis SQL bieten. Tatsache ist, dass Unternehmen und Organisationen in der heutigen Zeit mehr Zugang zu Daten haben als jemals zuvor, aber es gibt eine Datenknappheit. Die leistungstärksten KI-Maschinen basieren auf überwachtem Lernen, für das etikettierte Daten erforderlich sind. Etikettierte Daten sind so organisiert, dass sie für Maschinen verständlich sind, aber sie haben Grenzen. Mit der Zeit investieren Unternehmen viel in das Entwerfen von Techniken und konzentrieren sich darauf, wie KI-Modelle erstellt werden können, die trotz des Mangels an etikettierten Daten gelernt werden können.

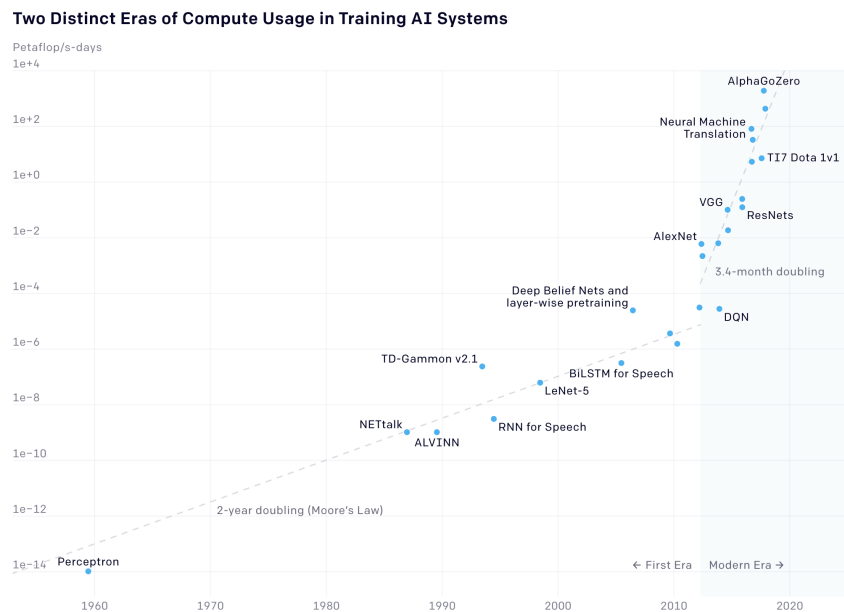


Abbildung 7.1: Die Rechenleistung, die zum Trainieren der KI verwendet wird, nimmt exponentiell zu [3].

7.2 Rechenleistung

Um zu verstehen und Entscheidungen auf der Grundlage eines riesigen Datenvolumens zu treffen, müssen eine Reihe von Berechnungen sehr schnell durchgeführt werden. Aufgrund einer Vielzahl von Berechnungen, die in kürzester Zeit durchgeführt werden müssen, wird viel Rechenleistung verbraucht. Rechenleistung bedeutet Rechenkapazität. OpenAI ist eine gemeinnützige Organisation, die die Leistungsfähigkeit von KI untersucht [44] und 2018 einen Artikel über die zunehmende Rechenleistung von KI-Berechnungen veröffentlicht hat.

Die obige Abbildung 7.1 beschreibt die Entwicklung der KI im Hinblick auf die Rechenleistung. Diese Abbildung basiert auf den Daten von 1959 bis heute. In der Abbildung sind die Änderungen zu sehen, die die KI-Evolution in zwei Epochen unterteilt. Der erste Zeitraum von 1959 bis 2012 ist durch Ergebnisse definiert, die in etwa dem Moore-Gesetz entsprechen, und der zweite Zeitraum von 2012 bis heute durch Ergebnisse, bei denen Rechenleistung verwendet wird, die die Makrotrends erheblich übertrifft [3]. Diese Analyse von OpenAI zeigt, dass sich die Zukunft der Rechenleistung in der KI-Anwendung und der neue Trend zur Erhöhung der Rechenleistung rasch fortsetzen werden [3].

7.3 Qualifiziertes Personal

Ein weiteres Problem, das angegangen werden muss und für jedes Unternehmen eine Herausforderung darstellt, ist die Verfügbarkeit von qualifiziertem Personal. Während des Jahres 2017 hat das MIT eine Umfrage zur KI-Implementierung in den 1000 größten Unternehmen der USA durchgeführt. Diese Umfrage zeigt, dass ihre größte Sorge bei der Implementierung von KI 1) die Bereitschaft und 2) die Fähigkeit der Mitarbeiter war, diese neuen Lösungen zu verstehen und mit ihnen zu arbeiten [57].

7.4 Black-Box-Technologie

Das Problem mit KI-Technologien ist, dass es für Menschen wie eine Black Box ist. Forscher und Ingenieure verstehen immer noch nicht, wie die Entscheidungen von dieser Technologie getroffen werden. Der Teil, der diese Unverständlichkeit verursacht, sind versteckte Ebenen zwischen Eingabe und Ausgabe. Dies ist eine Herausforderung für alle Forscher, die versuchen, KI in verschiedenen Branchen anzuwenden und ihnen das Gefühl zu geben, sich unwohl zu fühlen. Diese Unverständlichkeit der Daten führt zu Unübersichtlichkeit. Zum Beispiel ist die Transparenz des Kreditrisikos für die staatliche Institutionen sehr wichtig, weil sie wissen müssen, wie die Entscheidung basierend auf was getroffen wird.

7.5 Algorithmus Bias

Ein weiteres Problem, das im KI-System angegangen werden muss, ist die algorithmische Verzerrung. Die KI-Anwendung wird anhand verschiedener Datentypen trainiert, sodass Entscheidungen basierend auf diesen Daten getroffen werden können. Dies bedeutet, dass ihr Grad an Güte oder Schlechtigkeit von den Daten abhängt, auf denen sie trainiert werden. Schlechte Daten werden mit ethnischen, kommunalen, rassistischen und ideologischen Vorurteilen in Verbindung gebracht. Schlechte Daten beeinflussen die Entscheidungen der KI, was dies zu einem andauernden Problem macht. Bias in KI-Anwendungen ist ein Phänomen, das in den Daten des algorithmischen Modells auftritt. Um dieses Problem zu vermeiden, ver-

suchen die Forscher, diese Anwendungen mit unverfälschten Daten zu entwickeln und zu trainieren und Algorithmen zu entwickeln, die sich leicht erklären lassen.

7.6 Datenschutz

Datenschutz ist weltweit ein heißes Thema. Die Top-Tech-Unternehmen, die den größten Teil der Daten beherrschen, werden beschuldigt, die Benutzerinformationen missbraucht zu haben. KI-Anwendungen werden geschult und lernen aus Daten, die sensibler und persönlicher Natur sein können. Aufgrund des semantischen Lernens können KI-Anwendungen anfällig für Datenverletzungen und Identitätsdiebstahl sein. Um den Schutz personenbezogener Daten zu vervollständigen, hat die Europäische Union die Datenschutzgrundverordnung umgesetzt [73]. Dieser Schritt wird unternommen, nachdem das Bewusstsein der Kunden für eine zunehmende Anzahl von maschinellen Entscheidungen geschärft wurde. Es gibt auch eine andere einzigartige Methode, die als Föderiertes Lernen bekannt ist und darauf abzielt, das KI-Paradigma zu stören. Das Konzept des föderierten Lernens besteht darin, Datenwissenschaftler dazu zu ermutigen [74], KI-Anwendungen zu entwickeln und zu implementieren, ohne die Datensicherheit und Vertraulichkeit der Benutzer zu beeinträchtigen [74].

Kapitel 8

Diskussion

Diese Arbeit beschreibt und analysiert die weit verbreiteten Techniken des ML im finanziellen Risikomanagement. Das Management von Finanzrisiken ist eine Herausforderung für Banken und andere Finanzinstitutionen. Aufgrund der großen Datenmenge ist die Vorhersage des nächsten Ergebnisses eine Herausforderung für sich. Das Aktienmarktrisiko ist ein Aspekt des finanziellen Risikos, der das Risiko beinhaltet, durch den Kauf und Verkauf von Aktienpositionen Geld zu verlieren. Viele Faktoren beeinflussen den Aktienpreis, und es gibt viele Techniken, um ihn vorherzusehen. Eine davon ist die technische Analyse, bei der die Vorhersage auf historischen Aktiendaten basiert, indem Muster und Trends in diesen Daten erkannt werden. In dieser Arbeit wird ein ML-Ansatz zur Vorhersage der Aktienkursbewegung auf der Grundlage historischer Aktiendaten implementiert. Sie verwendet eine Spezialisierung von RNN, die im Finanzbereich weit verbreitet ist, nämlich LSTM. LSTM ist ein Algorithmus, der in der Zeitreihenvorhersage und einem iterativen Lernprozess spezifiziert ist, was sehr gut auf den Aktienmarkt anwendbar ist. LSTM Die Herangehensweise wird anhand historischer Bestandsdaten von Google und Microsoft trainiert. Auf der Grundlage von Mustern und Trends auf historischen Daten konnte das Modell den nächsten Aktienkurs des Unternehmens vorhersagen. Das Ergebnis des Modells zeigt, dass dies eine mögliche Lösung zur Minimierung des Aktienmarktrisikos ist. Es ist zu beobachten, dass das Modell den tatsächlichen Aktienpreis übertrifft, aber es ist nicht das beste Ergebnis, das durch die Anwendung von ML-Techniken erreicht werden kann. Die Leistung des Modells hängt von vielen Faktoren wie Hyperparametern ab. Das Modell mit den gleichen Hyperparametern schneidet bei Google und Microsoft unterschiedlich ab. Das bedeutet, dass das Modell für verschiedene Arten von Aktien trainiert werden sollte. Opti-

mierung ist der Prozess, den optimalen Hyperparameter zu finden, um eine niedrige Fehlerquote zu erreichen. Die Suche nach den optimalen Hyperparametern erfordert viel Rechenleistung und Zeit, um alle möglichen Hyperparameterkombinationen zu berechnen. Die Kombination von LSTM mit Ensemble-Lerntechniken könnte die Leistung des Modells verbessern. Bagging und Boosting sind zwei Techniken des Ensemble-Lernens. Die Kombination von Bagging und Boosting mit LSTM könnte die Fehlerrate (RMSE) senken, da die Vorhersage das Ergebnis jeder Trainingsphase ist. Eine Einschränkung von LSTM ist, dass sie nicht aus früheren Fehlern in der Vorhersage lernt. Dies führt dazu, dass sich Fehler ständig neu auftreten. Die Einschränkungen von LSTM könnten durch RL aufgehoben werden. Die Kernidee ist der Agent, der sich an die Umgebung anpasst und die Aktionen durch Feedback verbessert. Feedback ist sehr wichtig, weil es dem Agenten mitteilt, ob die durchgeführte Aktion erfolgreich war oder nicht. Dadurch wird der Agent trainiert, die Fehlerrate bei der Preisvorhersage zu reduzieren. Aufgrund der Fähigkeit des Agenten zur Selbstanpassung und Selbstoptimierung in der Umgebung könnte er im Aktien- und Devisenhandel besser als LSTM performen.

Abkürzungsverzeichnis

ANN	Artificial Neural Network
RNN	Recurrent Neural Network
SVM	Support Vector Machine
KI	Künstliche Intelligenz
AI	Artificial Intelligence
ML	maschinelles Lernen
RF	Random Forest
kNN	k-nächste-Nachbarn
BN	Bayesian Network
CNN	Convolutional Neural Network
LSTM	Long-Short-Term-Memory
RL	Reinforcement Learning
DL	Deep Learning
NLP	Natural Language Processing
ID3	Iterative Dichotomiser 3
NN	Neuronale Netze
adam	Adaptive Moment Estimation
SVR	Support Vector Regression
GRU	Gated Recurrent Unit
DNN	Deep Neural Network
FinTech	Finance Technology
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
CPU	Central Processing Unit
GPU	Graphics Processing Unit

ZAML Zest Automated Machine Learning

Tabellenverzeichnis

5.1	Statistiken für die Vorhersage von Validierungsdaten.	52
-----	---	----

Abbildungsverzeichnis

2.1	Methoden des maschinellen Lernens und ihre Anwendungen [21].	8
2.2	Bestandteil des Reinforcement Learning und deren Interaktionen[66]	10
3.1	Aktienkursprognose mit Hilfe von Entscheidungsbaum [71].	15
3.2	Alle möglichen Hyperebenen [45].	18
3.3	Maximum-Margin-Hyperebene, die aus zwei Klassen trainiert wurde [45].	18
3.4	a) 1NN, b) 4NN mit dem Entscheidungspunkt [31].	20
3.5	Ein Bayes'sches Netzwerk [64]	21
3.6	Bagging und Boosting Struktur [13]	23
3.7	Struktur der Random Forest Klassifikation [55]	23
3.8	Mehrschichtiges neuronales Netzwerk [28].	25
3.9	CNN-Architektur [47]	26
3.10	RNN-Architektur [27]	28
3.11	Die innere RNN-Architektur [27]	29
3.12	Die innere LSTM-Einheit Architektur [40]	30
5.1	Tabellarische Darstellung der Google-Aktienkursdaten	45
5.2	Graphische Darstellung der Google-Aktienkursdaten	46
5.3	Google	53
5.4	Microsoft	53
5.5	Vorhersage des Aktienpreises auf Validierungsdaten	53
5.6	Google	54
5.7	Microsoft	54
5.8	Vorhersage des Microsoft-Aktienkurses auf Testdaten	54
7.1	Die Rechenleistung, die zum Trainieren der KI verwendet wird, nimmt exponentiell zu [3].	62

Listings

9.1	Notwendige Bibliotheken importieren	xxv
9.2	Laden und Aufteilen der Daten.	xxv
9.3	Vorbereitung der Daten.	xxvi
9.4	Trainieren des Modells	xxvi
9.5	Vorhersage des Preises auf Validierungsdaten	xxvii
9.6	RMSE und MAPE berechnen	xxvii

Literatur

- [1] Peter Addo, Dominique Guegan und Bertrand Hassani. „Credit Risk Analysis Using Machine and Deep Learning Models“. In: *Risks* 6 (Apr. 2018), S. 38. DOI: 10.3390/risks6020038.
- [2] Peter Addo, Dominique Guegan und Bertrand Hassani. „Credit Risk Analysis Using Machine and Deep Learning Models“. In: *Risks* 6 (Apr. 2018), S. 38. DOI: 10.3390/risks6020038.
- [3] *AI and Compute*. <https://openai.com/blog/ai-and-compute/>. (Accessed on 11/24/2019).
- [4] *Aladdin* | *BlackRock*. <https://www.blackrock.com/aladdin>. (Accessed on 01/15/2020).
- [5] Saad Albawi, Tareq Abed Mohammed und Saad ALZAWI. „Understanding of a Convolutional Neural Network“. In: Aug. 2017. DOI: 10.1109/ICEngTechnol.2017.8308186.
- [6] *Algorithm* | *Definition of Algorithm by Merriam-Webster*. <https://www.merriam-webster.com/dictionary/algorithm>. (Accessed on 01/14/2020).
- [7] *Alpaca*. <https://www.alpaca.ai/>. (Accessed on 01/15/2020).
- [8] Ethem Alpaydin. *Introduction to Machine Learning*. 2nd. The MIT Press, 2010.
- [9] Edward I. Altman, Giancarlo Marco und Franco Varetto. „Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience)“. In: *Journal of Banking & Finance* 18.3 (Mai 1994), S. 505–529. URL: <https://ideas.repec.org/a/eee/jbfina/v18y1994i3p505-529.html>.
- [10] *Amplifying Intelligence* | *AlphaSense*. <https://www.alpha-sense.com/>. (Accessed on 01/15/2020).

- [11] *Auquan*. <https://www.auquan.com/>. (Accessed on 01/15/2020).
- [12] Akhil Azhikodan, Anvitha Bhat und Mamatha Jadhav. „Stock Trading Bot Using Deep Reinforcement Learning“. In: Mai 2019, S. 41–49. DOI: 10.1007/978-981-10-8201-6_5.
- [13] *Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained*. <https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725>. (Accessed on 01/10/2020).
- [14] *Bleckwen Homepage - Bleckwen - Fraud Detection*. <https://bleckwen.ai/>. (Accessed on 01/15/2020).
- [15] *causaLens - A Machine that Predicts the Global Economy in Real-Time*. <https://www.causalens.com/>. (Accessed on 01/15/2020).
- [16] T. Choi, H. K. Chan und X. Yue. „Recent Development in Big Data Analytics for Business Operations and Risk Management“. In: *IEEE Transactions on Cybernetics* 47.1 (Jan. 2017), S. 81–92. DOI: 10.1109/TCYB.2015.2507599.
- [17] *Classification-and-Regression-by-RandomForest.pdf*. https://www.researchgate.net/profile/Andy_Liaw/publication/228451484_Classification_and_Regression_by_RandomForest/links/53fb24cc0cf20a45497047ab/Classification-and-Regression-by-RandomForest.pdf. (Accessed on 10/16/2019).
- [18] T. Cover und P. Hart. „Nearest Neighbor Pattern Classification“. In: *IEEE Trans. Inf. Theor.* 13.1 (Sep. 2006), S. 21–27. DOI: 10.1109/TIT.1967.1053964. URL: <https://doi.org/10.1109/TIT.1967.1053964>.
- [19] Michael Dempster und V. Leemans. „An automated FX trading system using adaptive reinforcement learning“. In: *Expert Systems with Applications* 30 (Apr. 2006), S. 543–552. DOI: 10.1016/j.eswa.2005.10.012.
- [20] Amir Razi Dr. Iqra Ihsan. „Money Laundering-A Negative Impact on Economy“. In: *Global Journal of Management And Business Research* 12.17 (2012). URL: <https://journalofbusiness.org/index.php/GJMBR/article/view/814>.
- [21] Christo El Morr und Hossam Ali-Hassan. „Descriptive, Predictive, and Prescriptive Analytics“. In: *Analytics in Healthcare: A Practical Introduction*. Cham: Springer International Publishing, 2019, S. 31–55. DOI:

- 10.1007/978-3-030-04506-7_3. URL:
https://doi.org/10.1007/978-3-030-04506-7_3.
- [22] *Enterprise AI | DataRobot*. <https://www.datarobot.com/>. (Accessed on 01/15/2020).
- [23] *Époque Plus - époque plus*. <http://www.epoque-plus.ch/>. (Accessed on 01/15/2020).
- [24] *Fraud Prevention with Machine Learning - Feedzai*. <https://feedzai.com/>. (Accessed on 01/15/2020).
- [25] Nir Friedman, Dan Geiger und Moises Goldszmidt. „Bayesian Network Classifiers“. In: *Machine Learning* 29.2 (Nov. 1997), S. 131–163. DOI: 10.1023/A:1007465528199. URL: <https://doi.org/10.1023/A:1007465528199>.
- [26] Sevdalina Georgieva, Maya Markova und Velisar Pavlov. „Using neural network for credit card fraud detection“. In: Bd. 2159. Okt. 2019, S. 030013. DOI: 10.1063/1.5127478.
- [27] Ian Goodfellow, Yoshua Bengio und Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [28] Martin T Hagan u. a. *Neural network design*. Bd. 20. Pws Pub. Boston, 1996.
- [29] 1964- [Herausgeber/in] Haneke Uwe u. a., Hrsg. *Data Science: Grundlagen, Architekturen und Anwendungen*. Deutsch. 1. Auflage. tdwi Wissen. Hochschule Mannheim. Heidelberg: dpunkt.verlag, 2019, xvi, 320 Seiten.
- [30] Sepp Hochreiter und Jürgen Schmidhuber. „Long short-term memory“. In: *Neural computation* 9.8 (1997), S. 1735–1780.
- [31] S.B. Imandoust und Mohammad Bolandraftar. „Application of K-nearest neighbor (KNN) approach for predicting economic events theoretical background“. In: *Int J Eng Res Appl* 3 (Jan. 2013), S. 605–610.
- [32] Vikramaditya R. Jakkula. „Tutorial on Support Vector Machine (SVM)“. In: 2011.
- [33] *Kavout - find alpha from factors, anomalies and signals*. <https://www.kavout.com/>. (Accessed on 01/15/2020).
- [34] 1952- [Verfasser/in] Keitsch Detlef, Hrsg. *Risikomanagement: [Finanzrisiken, Betriebsrisiken, Interne Revision, KonTraG, Frühwarn- und Überwachungssysteme, Corporate Governance]*. Deutsch. Bd. 3.

- Handelsblatt Mittelstands-Bibliothek ; 3. Hochschule Mannheim. Stuttgart: Schäffer-Poeschel, 2007, XII, 327 Seiten.
- [35] *Kensho*. <https://www.kensho.com/>. (Accessed on 01/15/2020).
- [36] Diederik Kingma und Jimmy Ba. „Adam: A Method for Stochastic Optimization“. In: *International Conference on Learning Representations* (Dez. 2014).
- [37] J. Korczak und M. Hemes. „Deep learning for financial time series forecasting in A-Trader system“. In: *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. Sep. 2017, S. 905–912. DOI: 10.15439/2017F449.
- [38] *Lendbuzz - Car loan for expats and international students*. <https://lendbuzz.com/>. (Accessed on 01/15/2020).
- [39] *Long Short-Term Memory Networks.pdf*. <http://pages.cs.wisc.edu/~shavlik/cs638/lectureNotes/Long%20Short-Term%20Memory%20Networks.pdf>. (Accessed on 01/24/2020).
- [40] *Long short-term memory - Wikipedia*. https://en.wikipedia.org/wiki/Long_short-term_memory. (Accessed on 01/25/2020).
- [41] *Matplotlib: Python plotting — Matplotlib 3.1.2 documentation*. <https://matplotlib.org/>. (Accessed on 01/23/2020).
- [42] Nikola Milosevic. „Equity forecast: Predicting long term stock price movement using machine learning“. In: (März 2016).
- [43] *Numerai*. <https://numer.ai/>. (Accessed on 01/15/2020).
- [44] *OpenAI*. <https://openai.com/>. (Accessed on 11/24/2019).
- [45] *OpenCV: Introduction to Support Vector Machines*. https://docs.opencv.org/3.4/d1/d73/tutorial_introduction_to_svm.html. (Accessed on 01/10/2020).
- [46] *Operational risk management in financial services | ORX*. <https://managingrisktogether.orx.org/>. (Accessed on 01/15/2020).
- [47] Keiron O’Shea und Ryan Nash. „An Introduction to Convolutional Neural Networks“. In: *ArXiv e-prints* (Nov. 2015).

-
- [48] Jigar Patel u. a. „Predicting stock market index using fusion of machine learning techniques“. In: *Expert Systems with Applications* 42 (Okt. 2014). DOI: 10.1016/j.eswa.2014.10.031.
- [49] Luca Di Persio und Oleksandr Honchar. „Recurrent Neural Networks Approach to the Financial Forecast of Google Assets“. In: 2017.
- [50] Mrs. Vidya Pujari u. a. „Forex Trading System“. In: 2018.
- [51] Joseph Pun und Yuri Lawryshyn. „Article: Improving Credit Card Fraud Detection using a Meta-Classification Strategy“. In: *International Journal of Computer Applications* 56.10 (Okt. 2012). Full text available, S. 41–46.
- [52] *Python Data Analysis Library — pandas: Python Data Analysis Library*. <https://pandas.pydata.org/>. (Accessed on 01/23/2020).
- [53] J. R. Quinlan. „Induction of Decision Trees“. In: *Mach. Learn.* 1.1 (März 1986), S. 81–106. DOI: 10.1023/A:1022643204877. URL: <http://dx.doi.org/10.1023/A:1022643204877>.
- [54] Angelo Ranaldo und Paolo Magistris. „Trading Volume, Illiquidity and Commonalities in FX Markets“. In: *SSRN Electronic Journal* (Jan. 2018). DOI: 10.2139/ssrn.3289026.
- [55] *Random Forest Classification - Towards Data Science*. <https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0>. (Accessed on 01/10/2020).
- [56] *Regression analysis - Wikipedia*. https://en.wikipedia.org/wiki/Regression_analysis. (Accessed on 09/28/2019).
- [57] M. S. M. Review und P. Michelman. „18 The Jobs That Artificial Intelligence Will Create“. In: *What the Digital Future Holds: 20 Groundbreaking Essays on How Technology Is Reshaping the Practice of Management*. MITP, 2017, S. 97–103. URL: <https://ieeexplore.ieee.org/document/8333218>.
- [58] Sebastian Ruder. „An overview of gradient descent optimization algorithms“. In: (Sep. 2016).
- [59] *scikit-learn: machine learning in Python — scikit-learn 0.22.1 documentation*. <https://scikit-learn.org/stable/>. (Accessed on 01/23/2020).

- [60] Maxim Shcherbakov u. a. „A survey of forecast error measures“. In: *World Applied Sciences Journal* 24 (Jan. 2013), S. 171–176. DOI: 10.5829/idosi.wasj.2013.24.itmies.80032.
- [61] J. Sola und J. Sevilla. „Importance of input data normalization for the application of neural networks to complex industrial problems“. In: *IEEE Transactions on Nuclear Science* 44.3 (Juni 1997), S. 1464–1468. DOI: 10.1109/23.589532.
- [62] Nitish Srivastava u. a. „Dropout: A Simple Way to Prevent Neural Networks from Overfitting“. In: *Journal of Machine Learning Research* 15 (2014), S. 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [63] *Statistical classification - Wikipedia*.
https://en.wikipedia.org/wiki/Statistical_classification. (Accessed on 11/04/2019).
- [64] Todd Andrew Stephenson. *An Introduction to Bayesian Network Theory and Usage*. Idiap-RR Idiap-RR-03-2000. IDIAP, 2000.
- [65] Ting Sun und Miklos A. Vasarhelyi. „Predicting credit card delinquencies: An application of deep neural networks“. In: *Intelligent Systems in Accounting, Finance and Management* 25.4 (2018), S. 174–189. DOI: 10.1002/isaf.1437. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/isaf.1437>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/isaf.1437>.
- [66] Richard S. Sutton und Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: <http://incompleteideas.net/book/the-book-2nd.html>.
- [67] Madjid Tavana u. a. „An Artificial Neural Network and Bayesian Network Model for Liquidity Risk Assessment in Banking“. In: *Neurocomputing* 275 (Nov. 2018), S. 2525–2554. DOI: 10.1016/j.neucom.2017.11.034.
- [68] *TensorFlow*. <https://www.tensorflow.org/>. (Accessed on 01/23/2020).
- [69] *Trading Technologies | Futures Trading Platform*.
<https://www.tradingtechnologies.com/>. (Accessed on 01/15/2020).
- [70] *Transform Your Underwriting with AI & Machine Learning | Zest AI*.
<https://www.zest.ai/>. (Accessed on 01/15/2020).
- [71] *Using Decision Trees in Finance*.
<https://www.investopedia.com/articles/financial-theory/11/decisions-trees-finance.asp>. (Accessed on 10/01/2019).

- [72] Vectra - AI-driven threat detection and response platform.
<https://www.vectra.ai/>. (Accessed on 01/15/2020).
- [73] Paul Voigt und Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. 1st. Springer Publishing Company, Incorporated, 2017.
- [74] Qiang Yang u. a. „Federated Machine Learning: Concept and Applications“. In: *ACM Trans. Intell. Syst. Technol.* 10.2 (Jan. 2019), 12:1–12:19. DOI: 10.1145/3298981. URL: <http://doi.acm.org/10.1145/3298981>.
- [75] Jing Zhang u. a. „A Novel Data-Driven Stock Price Trend Prediction System“. In: *Expert Systems with Applications* 97 (Dez. 2017). DOI: 10.1016/j.eswa.2017.12.026.
- [76] Yan Zhang und Peter Trubey. „Machine Learning and Sampling Scheme: An Empirical Study of Money Laundering Detection“. In: *SSRN Electronic Journal* (Jan. 2018). DOI: 10.2139/ssrn.3161436.
- [77] Zhongheng Zhang. „Introduction to machine learning: k-nearest neighbors“. In: *Annals of Translational Medicine* 4.11 (2016). URL: <http://atm.amegroups.com/article/view/10170>.

Kapitel 9

Erster Anhang

9.1 Importieren der erforderlichen Bibliotheken.

```
from pandas_datareader import data
import matplotlib.pyplot as plt
import pandas as pd
import datetime as dt
import numpy as np
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler
```

Listing 9.1: Notwendige Bibliotheken importieren

9.2 Daten laden und aufteilen

```
df = pd.read_csv(file_path, sep = ",")

# change date into datetime
df.loc[:, 'Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')

# lower case column
df.columns = [str(x).lower().replace(' ', '_') for x in df.columns]
df['month'] = df['date'].dt.month

# Sort dataframe by datetime
df.sort_values(by='date', inplace=True, ascending=True)

df.head()

# define size of each dataset
num_val = int(val_size*len(df))
num_test = int(test_size*len(df))
num_train = len(df) - num_val - num_test

# split dataset into train, val and test dataset
train = df[:num_train][['date', 'adj_close']]
```

```
val = df[num_train:num_train+num_val][['date', 'adj_close']]
train_val = df[:num_train+num_val][['date', 'adj_close']]
test = df[num_train+num_val:][['date', 'adj_close']]

# print all define datasets sizes and shapes
print("num_train = " + str(num_train))
print("num_val = " + str(num_val))
print("num_test = " + str(num_test))

print("train.shape = " + str(train.shape))
print("val.shape = " + str(val.shape))
print("train_val.shape = " + str(train_val.shape))
print("test.shape = " + str(test.shape))

# define plot figure parameters
rcParams['figure.figsize'] = 10, 8

# define x and y values
ax = df.plot(x='date', y='adj_close', style='b-', grid=True)

# define labels
ax.set_xlabel("date")
ax.set_ylabel("USD")
```

Listing 9.2: Laden und Aufteilen der Daten.

9.3 Datenvorbereitung

```
# Converting dataset into x_train and y_train
# scale traing dataset
scaler = MinMaxScaler(feature_range=(0, 1))
train_scaled = scaler.fit_transform(np.array(train['adj_close']).reshape(-1,1))

# Split into x and y
x_train, y_train = get_x_y_from_dataset(train_scaled, N, N)

# scale train_val dataset for model
scaler_final_model = MinMaxScaler(feature_range=(0, 1))

# final scaled train_val dataset
train_val_scaled_final = scaler_final_model.fit_transform(np.array(train_val['adj_close']).reshape(-1,1))

# Scale the test dataset according the min and max obtained from train_cv set
test_scaled = scaler_final_model.transform(np.array(df['adj_close']).reshape(-1,1))
```

Listing 9.3: Vorbereitung der Daten.

9.4 Model erstellen

```
# build LSTM network
model = Sequential()
```

```

# first LSTM Layer
model.add(LSTM(units=lstm_units, return_sequences=True, input_shape=(x_train.shape
    [1],1)))
# avoid model overfitting
model.add(Dropout(dropout_prob)) # dropout probability of 1

# second LSTM layer
model.add(LSTM(units=lstm_units))
# avoid model overfitting
model.add(Dropout(dropout_prob)) # dropout probability of 1

# output layer
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer=optimizer, metrics=["accuracy"])
model.fit(x_train, y_train, epochs=epochs, batch_size=batch_size, verbose=2)

```

Listing 9.4: Trainieren des Modells

9.5 Vorhersage auf Validierungsdaten

```

# Do prediction
est = model.predict(x_val)
est_inv = scaler.inverse_transform(est)

# Get correct scale of y_cv
y_val_inv = scaler.inverse_transform(y_val)

# Calculate RMSE
rmse_pred_val = get_rmse(y_val_inv, est_inv)
print("RMSE = %0.3f" % rmse_pred_val)

# Calculate MAPE
mape_pred_val = get_mape(y_val_inv, est_inv)
print("MAPE = %0.3f%" % mape_pred_val)

```

Listing 9.5: Vorhersage des Preises auf Validierungsdaten

9.6 Berechnung der Vorhersagefehlerrate

```

# calculate MAPE
def get_mape(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

#calculate RMSE
def get_rmse(y_true, y_pred):
    return math.sqrt(mean_squared_error(y_true, y_pred))

# Calculate RMSE

```

```
rmse_pred_val = get_rmse(y_val_inv, est_inv)
print("RMSE = %0.3f" % rmse_pred_val)

# Calculate MAPE
mape_pred_val = get_mape(y_val_inv, est_inv)
print("MAPE = %0.3f%" % mape_pred_val)
```

Listing 9.6: RMSE und MAPE berechnen