

Vorhersage von Aktiendaten

Arne Wiese
stu227813@mail.uni-kiel.de

ZUSAMMENFASSUNG

Das Hauptziel dieser Arbeit war die Generierung und Evaluierung von Vorhersagen für zukünftige Kurse der Apple-Aktie auf Basis historischer Daten. Im Fokus stand die Frage, ob sich mit den Prognosen über einen längeren Zeitraum hinweg ein profitabler Gewinn realisieren lässt. Für die Vorhersagen wurde das Long-Short-Term-Memory (LSTM) Netzwerk ausgewählt, das sich unter vier verschiedenen Architekturen als die effizienteste herausstellte. Das LSTM-Modell wurde speziell mit aufbereiteten Daten zum Apple-Aktienkurs trainiert.

Die Ergebnisse der Studie offenbaren eine Vorhersagegenauigkeit von 52,3% hinsichtlich des korrekten Trends des Aktienkurses. Diese Genauigkeit impliziert, dass die Prognosen in ihrer jetzigen Form nicht als ausreichend zuverlässig für eine langfristige Gewinnerzielung betrachtet werden können. Diese Erkenntnis wirft weiterführende Fragen auf bezüglich der Potenziale und Grenzen der Anwendung von LSTM-Netzwerken für die Prognose von Aktienkursen und der Möglichkeit, auf Basis der aktuellen Ergebnisse langfristig Gewinne zu erzielen.

SCHLAGWÖRTER

Deep Learning, Long-Short-Term-Memory, Aktien, Apple, Vorhersagen, Zeitreihen

1 EINFÜHRUNG

Maschinelles Lernen kann definiert werden als die Daten, die durch Wissensextraktion gewonnen werden. Maschinen müssen nicht explizit programmiert werden, sondern werden darauf trainiert, datengesteuerte Entscheidungen zu treffen. Anstatt für jedes spezifische Problem einen Code zu schreiben, werden den generischen Algorithmen Daten zur Verfügung gestellt und auf der Grundlage dieser Daten eine Logik entwickelt [4]. Im dynamischen Feld des Deep Learning, einem spezialisierten Bereich des maschinellen Lernens, unterscheiden wir primär zwei Hauptkategorien: Unsupervised Learning und Supervised Learning. Unsupervised Learning bezieht sich auf den Umgang mit Daten ohne vordefinierte Labels. Das heißt, der zugrundeliegende Datensatz ist nicht gut definiert und schwer zu interpretieren [4]. Die Herausforderung hier liegt darin, verborgene Strukturen und Muster in den Daten zu identifizieren und dadurch Datenpunkte in sinnvolle Cluster zu gruppieren. Ein typisches Beispiel für diese Art von Lernprozess ist der Einsatz von Autoencodern.

Im Gegensatz dazu basiert Supervised Learning auf Daten, die bereits mit Labels versehen sind. Die Kernaufgabe besteht darin, aus diesen gelabelten Daten Regeln für die Klassifikation abzuleiten. Diese erlernten Regeln ermöglichen es, ungelabelte Daten zu klassifizieren oder präzise Vorhersagen zu treffen. In diesem Bereich finden auch künstliche neuronale Netze wie das Multi-Layer Perceptron, Convolutional Networks, Long-Short-Term Memory (LSTM) und Transformer breite Anwendung.

Diese Arbeit fokussiert sich speziell auf den Einsatz des Long-Short-Term-Memory Netzwerks, um Vorhersagen für den Kurs der Apple-Aktie zu treffen. Dabei gilt der Aktienmarkt als chaotisch, komplex, volatil und dynamisch. Zweifellos ist seine Vorhersage eine der schwierigsten Aufgaben bei der Zeitreihenprognose [8]. Die leitende Forschungsfrage dieser Studie ist: Kann man trotz dieser Bedingungen auf der Grundlage historischer Daten den Kurs der Apple-Aktie für einen Tag in der Zukunft zuverlässig vorhersagen?

Um diese Frage zu beantworten, nutzen wir eine von zwei gängigen Methoden zur Analyse von Aktienpreisen: die technische Analyse [7]. Die technische Analyse ist ein Ansatz, bei dem historische Marktdaten, vor allem Preise und Handelsvolumina, untersucht werden, um zukünftige Preisbewegungen auf den Finanzmärkten vorherzusagen. Sie stützt sich auf die Identifizierung von Mustern und Trends in Preischarts, mit der Annahme, dass sich historische Verhaltensweisen und Preisbewegungen zukünftig wiederholen können. Dafür wird zunächst in Kapitel 2 und 3 die angewandte Methodik und der verwendete Datensatz vorgestellt. Es folgt in Kapitel 4 eine ausführliche Darstellung des Trainingsprozesses. In Kapitel 5 werden die erzielten Ergebnisse präsentiert, und Kapitel 6 liefert ein abschließendes Fazit über die Effektivität der Prognose des Aktienkurses.

2 METHODEN

2.1 Netzwerk Architekturen

Die effektive Vorhersage von Zeitreihen, insbesondere im Kontext von Aktienmärkten wie der Apple-Aktie, erfordert eine sorgfältige Auswahl der Netzwerkarchitektur. Unterschiedliche Architekturen haben ihre eigenen Stärken, die je nach Art der Daten und der gewünschten Analyse variieren. Wir betrachten vier zentrale Architekturen, die sich in zwei Hauptkategorien einteilen lassen: Feedforward-Neuronale Netzwerke (FNNs) und Rekurrente Neuronale Netzwerke (RNNs). FNNs sind ideal für Situationen, in denen einfache Mustererkennung erforderlich ist, während RNNs sich durch ihre Fähigkeit auszeichnen, komplexe sequenzielle Daten zu verarbeiten - ein Schlüsselement bei der Analyse von Finanzzeitreihen.

2.1.1 Multi-Layer Perceptron (MLP). Das Multi-Layer Perceptron, eine klassische Feedforward-Neuralnetzwerkarchitektur, besteht aus mehreren vollständig verbundenen Neuronenschichten. Diese Architektur ist für nicht-sequenzielle Daten geeignet und hat sich bei der Erkennung von Mustern in Finanzindikatoren als nützlich erwiesen. MLPs sind effektiv in Szenarien, wo die Beziehungen zwischen verschiedenen Finanzparametern erfasst werden sollen, ohne dass eine zeitliche Komponente eine Rolle spielt. Trotzdem ist diese Architektur durchaus für einfache, weniger komplexe Zeitreihen in betracht zu ziehen.

2.1.2 Convolutional Neural Network (CNN). Das Convolutional Neural Network ist hauptsächlich für die Verarbeitung von räumlichen Mustern konzipiert, wie sie in Bildverarbeitungsaufgaben

vorkommen. Durch die Anwendung von Convolutional Layers und Pooling-Layers zur Musterextraktion und Dimensionsreduktion können CNNs jedoch, insbesondere in Kombination mit RNNs, auch effektiv zeitliche Abhängigkeiten in Finanzdaten modellieren. Dies macht sie für Aufgaben wie die Analyse von Zeitreihenmustern in Aktienkursen relevant.

2.1.3 Long-Short-Term-Memory (LSTM). LSTM, eine spezielle Form von RNN [3], wurde entwickelt, um das Problem des verschwindenden Gradienten [2] zu überwinden. Mit seiner Fähigkeit, Informationen über lange Zeiträume hinweg zu speichern, eignet sich LSTM hervorragend für die Modellierung von Zeitreihen. In der Finanzwelt ist dies besonders nützlich, da es ermöglicht, langfristige Trends und zyklische Muster in Aktienkursen zu erfassen und zu analysieren.

2.1.4 Transformer. Die Transformer-Architektur, basierend auf dem Prinzip der Selbst-Aufmerksamkeit, ist eine aufmerksamkeitsbasierte Netzwerkstruktur ohne rekurrente Verbindungen. Diese Eigenschaft ermöglicht eine effiziente Parallelisierung, was die Trainingsgeschwindigkeit erheblich verbessert. Obwohl ursprünglich für die Verarbeitung von Sprachdaten entwickelt, hat sich der Transformer auch in der Zeitreihenprognose bewährt. Seine Fähigkeit, komplexe Muster und Abhängigkeiten in großen Datensätzen zu erkennen, macht ihn ideal für die Vorhersage von Aktienkursbewegungen.

2.2 Netzwerk Performance

Die Performance eines neuronalen Netzwerks in der Vorhersage von Finanzzeitreihen hängt stark von der Vermeidung von Underfitting und Overfitting ab.

2.2.1 Underfitting. Underfitting tritt auf, wenn das Modell nicht in der Lage ist, die zugrunde liegenden Muster der Trainingsdaten zu erfassen, was sich in einer schlechten Leistung sowohl auf Trainings- als auch auf Testdaten zeigt. Im Kontext der Finanzzeitreihenanalyse bedeutet dies, dass das Modell nicht in der Lage ist, wichtige Markttrends und -signale zu identifizieren. Zu einfache Modelle oder unzureichende Trainingszeiten sind häufige Ursachen.

2.2.2 Overfitting. Overfitting ist das Gegenteil: Das Modell passt sich den Trainingsdaten zu genau an und verliert dabei seine Generalisierbarkeit. In der Praxis bedeutet dies, dass das Modell zwar historische Daten genau vorhersagen kann, aber bei neuen, unbekannten Daten versagt. Um Overfitting zu vermeiden, werden Techniken wie L1- und L2-Regularisierung, Dropout, Early Stopping und das Sammeln von mehr Daten eingesetzt. Diese Techniken helfen dabei, die Robustheit des Modells zu erhöhen und seine Fähigkeit zu verbessern, auf neue Daten angemessen zu reagieren.

3 DATENSATZ

3.1 Apple Datensatz

Der verwendete Datensatz dieser Arbeit wurde mithilfe der Yahoo Finance API [9] erhoben und umfasst folgende Attribute: Der Index ist ein Datum, dann folgen die Spalten 'Open', 'Low', 'High', 'Close', 'Adj Close' und 'Volume'. Open gibt den Preis der Apple Aktie an, der bei Eröffnung der Börse festgelegt wurde. Close den

Count	3025.00
Mean	66.29
Std	56.63
Min	11.99
25%	22.45
50%	39.47
75%	122.22
Max	198.11

Abbildung 1: Deskriptive Statistiken über die letzten 12 Jahre der Apple Aktie.

Preis bei Schließung der Börse. Low und High geben jeweils den Niedrigsten und Höchsten Preis des Tages an. Adjusted Close ist der Close Preis, der jedoch auch nach einem Aktiensplit konstant bleibt. Abschließend gibt die Spalte Volume an, wie viele Aktien pro Tag durch eine Transaktion ihren Besitzer gewechselt haben.

3.2 Explorative Datenanalyse

Eine explorative Datenanalyse wurde durchgeführt, um tiefer in den Datensatz einzutauchen und ihn anhand der folgenden Forschungsfragen zu untersuchen:

- (1) Wie ist die Verteilung der Daten über die letzten 12 Jahre?
- (2) Welche Trends zeigt der Moving Average in diesem Zeitraum?
- (3) Wie verhält sich die tägliche Rendite (Daily Return) im letzten Jahr?
- (4) Wie hoch ist das Investitionsrisiko in Apple-Aktien?

3.2.1 Deskriptive Statistiken. Die deskriptiven Statistiken (Abbildung 1) zeigen, dass die Verteilung der Daten eine signifikante Streuung aufweist, mit 50% der Werte unter 39.47 und über 11.99, bei einem Maximum von 198.11. Dies deutet auf eine breite Varianz in der Wertentwicklung der Aktien hin. Die Datenmenge enthält 3025 Beobachtungen. Das ist eine recht kleine Datenmenge und kann das Trainieren von Neuronalen Netzen erschweren.

3.2.2 Moving Average. Die Analyse des Moving Averages über einen Zeitraum von 12 Jahren (Abbildung 2) zeigt einen generell positiven Trend, mit einigen negativen Ausreißern in den Jahren 2020 und 2023, was auf kurzfristige Marktschwankungen hinweist.

3.2.3 Tägliche Rendite (Daily Return). Die Betrachtung der täglichen Rendite (Daily Return) über das letzte Jahr zeigt eine durchschnittliche Tendenz um 0% (Abbildung 4), mit einzelnen positiven und negativen Ausreißern (Abbildung 3). Dies deutet auf eine stabile, aber volatile Marktlage hin.

3.2.4 Risiko. Die Risikobewertung ist ein entscheidender Aspekt bei der Investition in Aktien. In dieser Arbeit wurde das Risiko der Investition in Apple-Aktien durch den Standardfehler des Daily Returns quantifiziert. Der Standardfehler gibt Aufschluss über die Variabilität der täglichen Rendite und somit über das Schwankungsrisiko der Aktie.

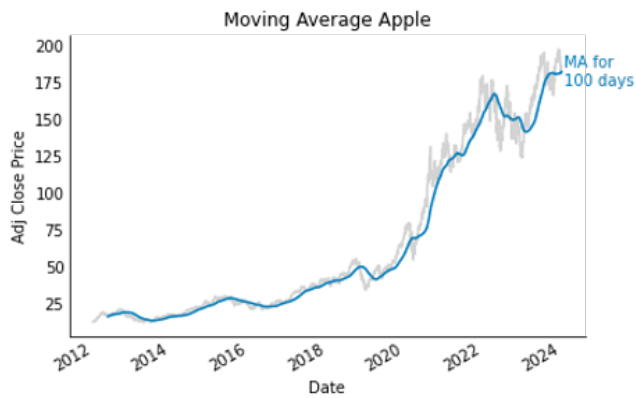


Abbildung 2: Der Moving Average über 100 Tage über die letzten 12 Jahre hinweg.

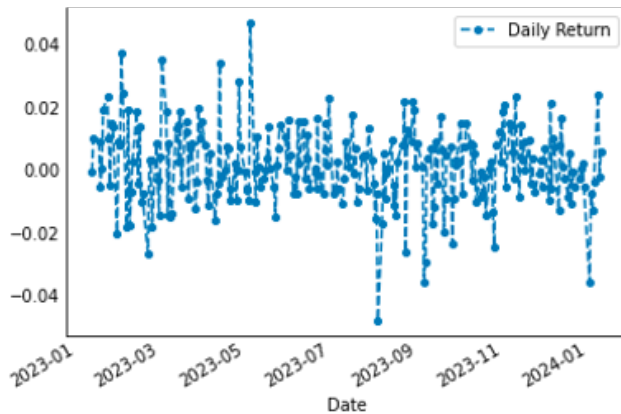


Abbildung 3: Die tägliche Rendite über das letzte Jahr hinweg.

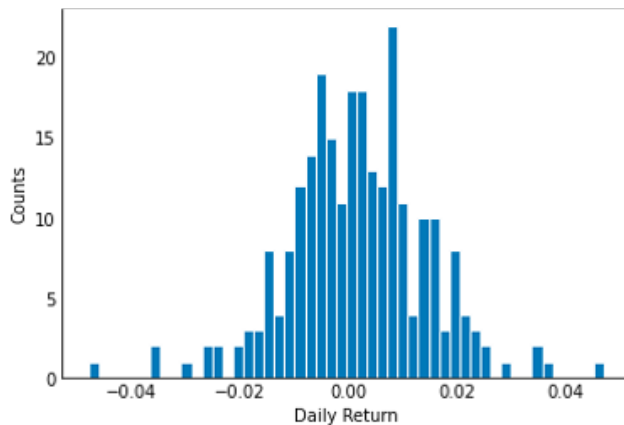


Abbildung 4: Histogramm der täglichen Rendite über das letzte Jahr hinweg.

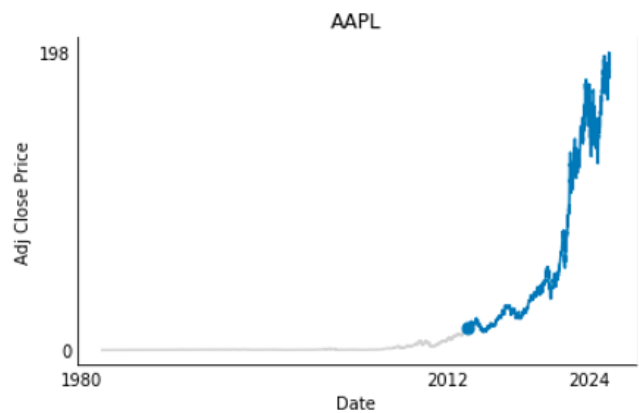


Abbildung 5: In blau ist der Anteil der relevanten Daten, der letzten 35 Jahre markiert.

Durch die Analyse des Daily Returns wurde festgestellt, dass das durchschnittliche Risiko bei einer Investition in Apple-Aktien bei 1.28% liegt, mit einem durchschnittlichen Daily Return von 0.17%. Dies bedeutet, dass unter normalen Marktbedingungen 68% der täglichen Renditen innerhalb eines Bereichs von $[-1.11\%, 1.45\%]$ liegen. Ein solcher Bereich ist für eine Aktie wie Apple relativ moderat, was auf eine gewisse Stabilität des Wertpapiers hindeutet.

3.3 Datenaufbereitung

Für das Training des neuronalen Netzwerks wurden die Daten des Apple-Datensatzes speziell aufbereitet. Die Analyse beschränkte sich auf Daten seit dem Jahr 2012 (Abbildung 5) (blau markierter Bereich), da frühere Daten als nicht relevant für den Trainingsprozess identifiziert wurden (grau markierter Bereich). Die Daten wurden in zwei unterschiedliche Arten unterteilt: Preis-Daten und Trend-Daten.

3.3.1 Preis-Daten. Die Preis-Daten bestehen aus dem normalisierten 'Adjusted Close' Preis über die Zeit. Diese Normalisierung ist wichtig, um Instabilitäten im Trainingsprozess zu vermeiden, die durch wenige, aber sehr hohe Werte (Abbildung 1) entstehen könnten.

3.3.2 Trend-Daten. Die Trend-Daten zeigen den prozentualen Trend zwischen aufeinanderfolgenden Tagen, basierend auf dem 'Adjusted Close' Preis. Diese Daten wurden nicht zusätzlich normalisiert, da sie bereits in einem niedrigen Wertebereich liegen. Außerdem ist es wichtig, dass negative Werte erhalten bleiben, um die Richtung des Trends genau abbilden zu können.

4 TRAININGSPROZESS

Dieser Abschnitt konzentriert sich auf den Trainingsprozess der neuronalen Netze für die Aktienvorhersage, der ausschließlich auf den Trend-Daten basiert. Die Wahl fiel auf diese Daten, da sie in vorangegangenen Tests eine höhere Genauigkeit bei der Vorhersage des richtigen Trends zeigten. Die Preis-Daten sind in dieser Dokumentation nicht weiter beachtet, denn wie in Abbildung (Abbildung 6) zu erkennen, wurde hier das Modell darauf optimiert,

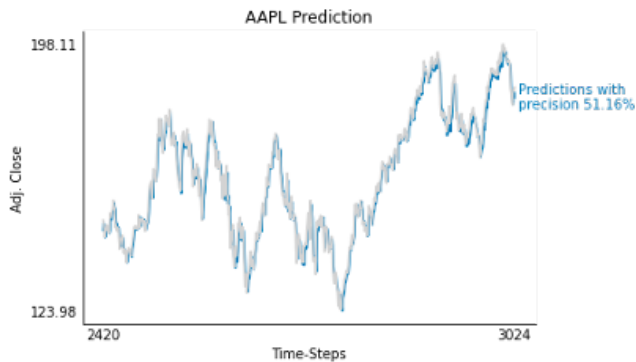


Abbildung 6: Obwohl der Preis sehr präzise geschätzt wird, ist die Genauigkeit zur Vorhersage des richtigen Trends zufällig.

einen möglichst exakten Preis vorherzusagen mit Vernachlässigung der Genauigkeit des Trends. Das Resultiert zwar in einer genauen Schätzung des Preises, jedoch in einer zufälligen Genauigkeit des Trends.

4.1 Vorbereitung der Trend-Daten

Für das Training des Modells wurden die Trend-Daten nochmals speziell vorbereitet. Die Daten wurden in zusammengehörige Input-Variablen und Zielvariablen aufgeteilt. Die Inputvariablen bestanden aus 62 (2 Monate) aufeinanderfolgenden Trend-Datenpunkten. Die Zielvariable gab den Trend am 63. Tag an. Der Trend wurde als binärer Wert kodiert, wobei 0 einen negativen und 1 einen positiven Trend des Aktienkurses anzeigt.

Im Rahmen dieser Dokumentation ist nur ein Inputsequenz von 62 Tagen berücksichtigt worden, weil das die besten Ergebnisse gebracht hat. Es ist aber zu beachten, dass die Sequenzlänge einen großen Einfluss auf die Performance haben kann!

4.2 Netzwerkarchitektur und Aktivierungsfunktion

Jedes neuronale Netzwerk endete mit einer Sigmoid Aktivierungsfunktion in der letzten Schicht. Diese Funktion transformiert die Ausgabe des Netzwerks in einen Wahrscheinlichkeitswert zwischen 0 und 1, welcher die Chance repräsentiert, dass der Aktienkurstrend am 63. Tag positiv ist.

4.3 Loss-Funktion und Modellinterpretation

Als Loss-Funktion wurde die binäre Kreuzentropie (Binary Crossentropy) verwendet. Diese Funktion misst die Leistung eines Klassifizierungsmodells, dessen Ausgabe ein Wahrscheinlichkeitswert zwischen 0 und 1 ist. Die binäre Kreuzentropie (Gleichung 1) eignet sich besonders gut für binäre Klassifizierungsprobleme, wie in

diesem Fall die Vorhersage eines positiven oder negativen Trends.

$$L(y, p) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (1)$$

Für jede Beobachtung berechnet die Funktion den Verlust basierend auf dem Vergleich zwischen dem tatsächlichen Wert und der vom Modell vorhergesagten Wahrscheinlichkeit. Ist der tatsächliche Wert 1, so ist der erste Teil der Summe relevant (Gleichung 1). Je näher die Wahrscheinlichkeit von 100% vorhergesagt wird, desto kleiner wird der Verlustwert. Ist der tatsächliche Wert 0, so ist der zweite Teil der Summe relevant. Hier gilt, je höher die Wahrscheinlichkeit vorhergesagt wird, dass der tatsächliche Wert 1 ist, umso größer fällt der Verlustterm aus. Die logarithmusfunktion sorgt dafür, dass das Modell versucht, möglichst genaue Wahrscheinlichkeiten vorherzusagen, denn die Abweichung zwischen dem tatsächlichen Wert und der Wahrscheinlichkeit wird damit exponentiell bestraft.

Das Modell lernt somit, basierend auf den Trend-Daten der vorherigen 62 Tage, eine möglichst gute Vorhersage für den Trend am 63. Tag zu treffen. Die Ausgabe des Modells – ein Wahrscheinlichkeitswert – lässt sich wie folgt interpretieren: Wie hoch ist die Wahrscheinlichkeit, dass am 63. Tag ein positiver Trend vorliegt?

4.4 Grid-Search Vergleich

Mit den fertig formatierten Trend-Daten und Modellspezifikationen konnte dann mit dem Training der einzelnen Modelle begonnen werden. Dabei war ein wesentlicher Bestandteil dieser Arbeit die Implementierung einer Grid-Search-Optimierung [6] über verschiedene Netzwerkarchitekturen, dem MLP (Abbildung 7, 10), dem CNN (Abbildung 8, 11) und dem LSTM (Abbildung 9, 12). Damit sollten die optimalen Hyperparameter, insbesondere Batch-Größe, Anzahl der Epochen, Lernrate, Dropoutrate für jede Architektur ermittelt werden, damit man einen guten Vergleich der Performance zwischen den Architekturen erhält. Tabelle 1 zeigt die jeweiligen Parameter und Metriken der jeweils besten Modelle jeder Architektur. Die Transformer Architektur wurde hierbei, aufgrund zu hoher Trainingszeiten, vernachlässigt.

4.5 Modell Auswahl und Fine Tuning

Basierend auf den umfangreichen Ergebnissen der Grid-Search-Optimierung und der vergleichenden Analyse der Leistungsmetriken, wie in Tabelle 1 dargestellt, fiel die Wahl der Architektur für das finale Modell für die Aktienvorhersage auf das Long Short-Term Memory (LSTM). Das LSTM-Modell übertraf die anderen Architekturen in mehreren wichtigen Leistungsmetriken, einschließlich der Trainings- und Validierungsgenauigkeit, Präzision, F1-Score und AUC-ROC. Diese umfassende Leistungsstärke deutet darauf hin, dass das LSTM-Modell besser in der Lage ist, die zugrunde liegenden Muster in den Zeitreihendaten zu erfassen und effektiv auf neue Daten zu generalisieren.

Das Fine-Tuning beschränkte sich dann darauf, die Dropoutrate in Kombination mit der Daten Augmentation zu optimieren. Dabei gab es aber hinsichtlich der Ergebnisse keine Verbesserungen mehr. Das finale Modell ist demnach in Tabelle 1 zu sehen.

Parameter / Metrik	MLP	CNN	LSTM
Batch-Größe	64	128	128
Epochen	50	30	50
Lernrate	0.001	0.001	0.005
Schichtenkonfiguration	[512, 512, 512, 64, 64]	[64, 64, 64, Pooling, 128]	[8, 16]
Dropout-Rate	0.4	0.4	0.1
Trainingsverlust	0.9437	0.6916	0.6830
Trainingsgenauigkeit	0.5088	0.5214	0.5597
Validierungsverlust	0.7241	0.6858	0.7113
Validierungsgenauigkeit	0.4927	0.5527	0.5190
Genauigkeit (Testdatensatz)	0.4967	0.5164	0.5230
Präzision (Testdatensatz)	0.4858	0	0.5100
Recall (Testdatensatz)	0.6373	0	0.4339
F1-Score (Testdatensatz)	0.5513	0	0.4689
AUC-ROC (Testdatensatz)	0.4966	0.5207	0.5108

Tabelle 1: Vergleich der optimalen Parameter und Leistungsmetriken für MLP, CNN und LSTM, mit Hervorhebung der besten Werte

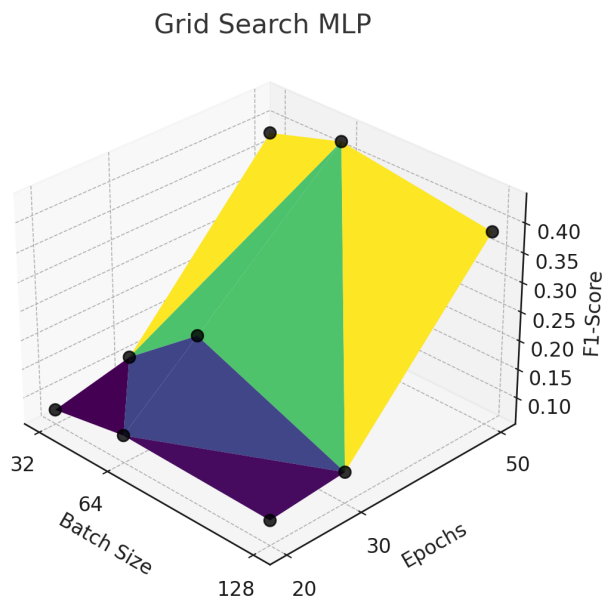


Abbildung 7: Grid-Search über Batchsize und Epochen auf MLP

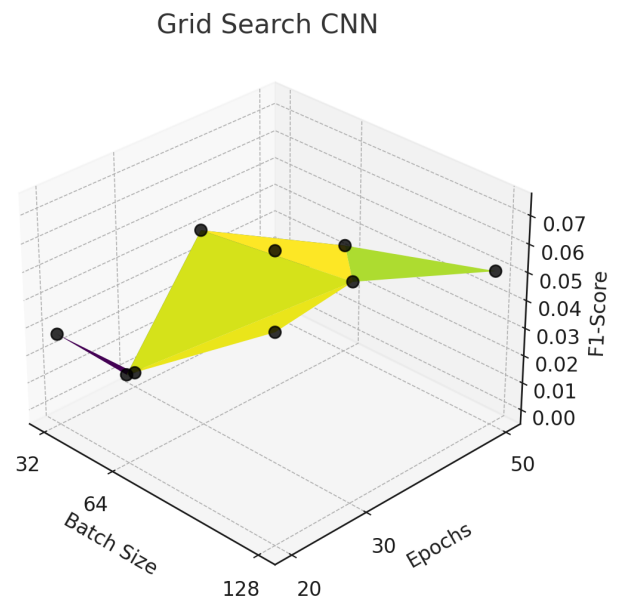


Abbildung 8: Grid-Search über Batchsize und Epochen auf CNN

5 ERGEBNISSE

Die Ergebnisse des LSTM-Modells, insbesondere die Metriken Präzision (Precision), Recall (Sensitivität) und F1-Score, bieten wertvolle Einblicke in die Leistung des Modells bei der Vorhersage von Aktienkurstrends. Im Folgenden werden diese Ergebnisse detailliert interpretiert:

Präzision (0.5100): Die Präzision gibt an, welcher Anteil der als positiv klassifizierten Vorhersagen tatsächlich korrekt war. Eine Präzision von 51.00% bedeutet, dass etwas mehr als die Hälfte der vom Modell vorhergesagten positiven Trends tatsächlich positiv

waren. Dieser Wert ist ein Indikator dafür, wie zuverlässig das Modell bei der Identifizierung positiver Trends ist. Obwohl eine Präzision von über 50% besser als zufälliges Raten ist, zeigt sie auch, dass fast die Hälfte der positiven Vorhersagen falsch war, was auf einen Verbesserungsbedarf hinweist.

Recall (0.4339): Der Recall misst, wie gut das Modell alle tatsächlichen positiven Fälle erfasst hat. Ein Recall von 43.39% bedeutet, dass das LSTM-Modell weniger als die Hälfte der tatsächlichen positiven Trends korrekt identifiziert hat. Dies deutet darauf hin, dass

Grid Search LSTM

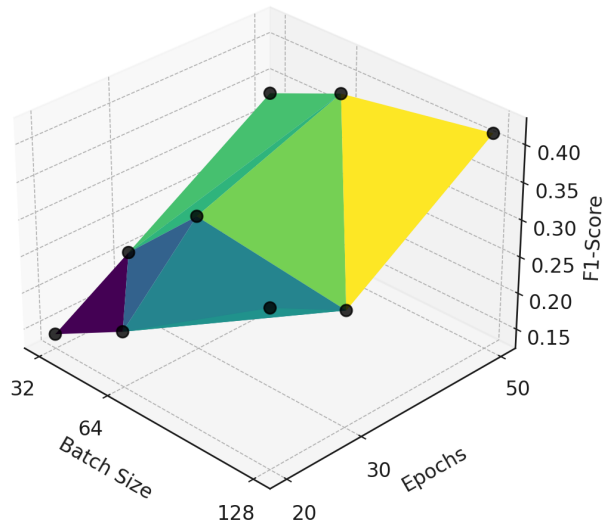


Abbildung 9: Grid-Search über Batchsize und Epochen auf LSTM

Grid Search CNN

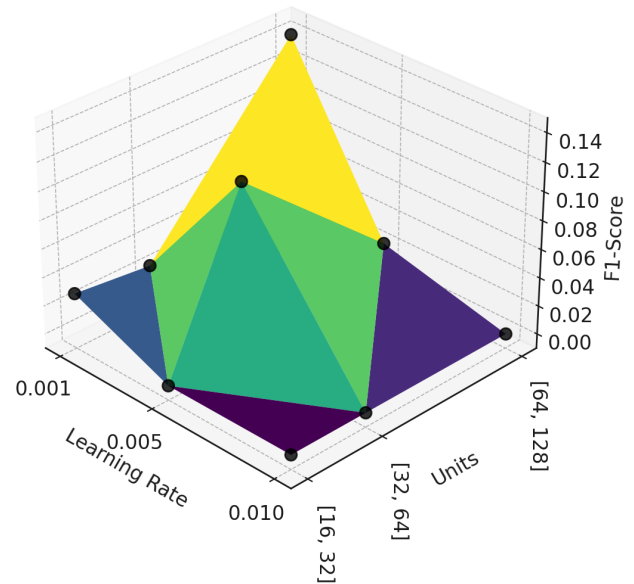


Abbildung 11: Grid-Search über Lernrate and Units auf CNN

Grid Search MLP

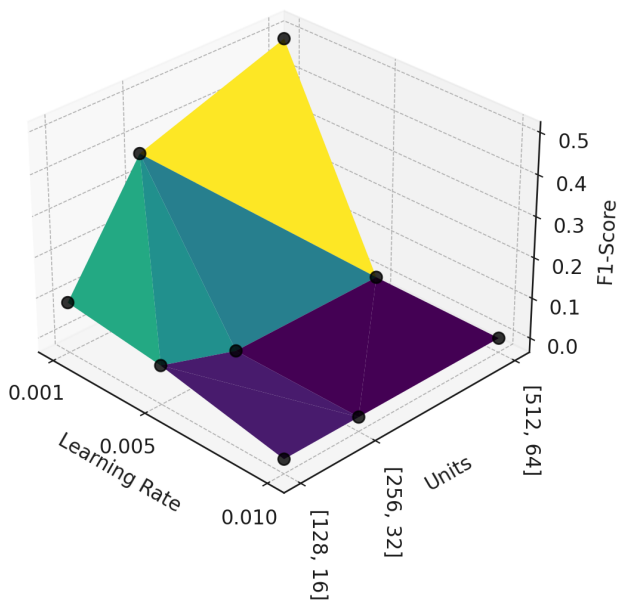


Abbildung 10: Grid-Search über Lernrate und Units auf MLP

das Modell viele positive Trends übersieht, was besonders in Anwendungsfällen problematisch sein kann, bei denen das Erkennen jedes positiven Trends von entscheidender Bedeutung ist.

F1-Score (0.4689): Der F1-Score ist das harmonische Mittel aus Präzision und Recall und bietet ein ausgewogenes Maß für die

Grid Search LSTM

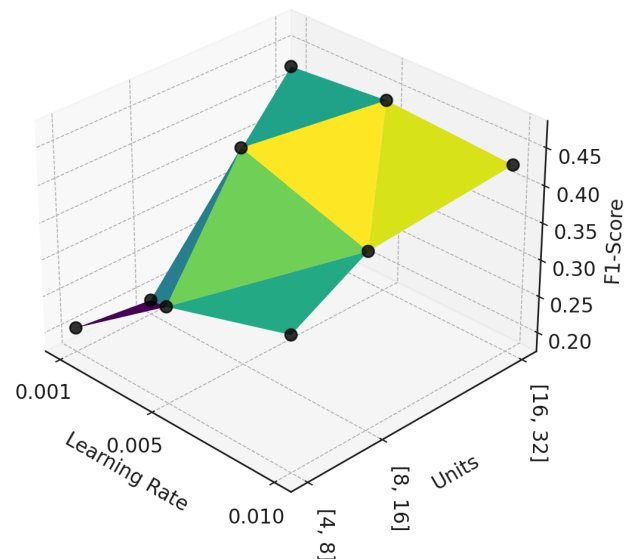


Abbildung 12: Grid-Search über Lernrate und Units auf LSTM

Gesamtleistung des Modells. Ein F1-Score von 46.89% zeigt, dass das Modell in Bezug auf die Balance zwischen Präzision und Recall mäßig abscheidet. Der Wert unterstreicht die Notwendigkeit, das Modell weiter zu optimieren, um sowohl die Präzision als auch

den Recall zu verbessern, da beide Metriken direkt den F1-Score beeinflussen.

Die Interpretation dieser Ergebnisse legt nahe, dass das LSTM-Modell in seiner aktuellen Form zwar einige Fähigkeiten besitzt, positive Trends in Aktienkursdaten zu identifizieren, jedoch noch weit davon entfernt ist, eine hohe Genauigkeit in seinen Vorhersagen zu erreichen. Die suboptimale Balance zwischen Präzision und Recall, reflektiert durch den F1-Score, deutet auf ein zentrales Dilemma hin: Das Modell ist derzeit nicht in der Lage, eine große Anzahl wahrer positiver Trends zu identifizieren, ohne dabei eine signifikante Anzahl falscher Positiver zu generieren. Die Stärke des Modells liegt eher darin, dass, wenn ein positiver Trend vorhergesagt wird, dieser auch zu 51% dem tatsächlichen Trend entspricht.

Die Analyse der Leistung des LSTM-Modells anhand der Metriken Präzision, Recall und F1-Score deutet darauf hin, dass Optimierungspotenzial besteht. Ein wesentlicher Schritt zur Steigerung der Modellleistung könnte die Akquise und Integration zusätzlicher Daten sein. Derzeit basiert das Modell auf einem Datensatz mit rund 3000 Beobachtungen, eine Menge, die möglicherweise nicht ausreicht, um die komplexen Muster und Dynamiken des Aktienmarktes vollständig zu erfassen. Die Erweiterung des Datensatzes könnte auf verschiedene Weise erfolgen:

- Erhöhung der Datenmenge: Durch die Beschaffung zusätzlicher historischer Daten für die Apple-Aktie (mehr Werte pro Tag) oder durch die Einbeziehung weiterer Finanzindikatoren und externer Variablen, wie zum Beispiel Markt-sentiment, könnte das Modell von einem umfassenderen Kontext profitieren. Mehr Daten könnten dem LSTM helfen, relevante Muster präziser zu identifizieren und seine Vorhersagen zu verbessern.
- Anwendung von Bootstrap-Verfahren: Um die vorhandenen Daten effektiver zu nutzen, könnten Bootstrap-Methoden eingesetzt werden. Diese statistischen Techniken ermöglichen die 'künstliche' Generierung neuer Datensätze durch wiederholtes Ziehen mit Zurücklegen aus dem bestehenden Datensatz [1]. Dadurch kann die Variabilität der Daten erhöht und die Robustheit des Modells gegenüber Überanpassung verbessert werden. Die Einhaltung einer korrekten Sequenz stellt aber eine Schwäche dieser Methode dar.

Außerdem ist es zu beachten, dass die Generierung künstlicher Daten mit Vorsicht behandelt werden sollte, um die realen Eigenschaften des Marktes nicht zu verzerren. Die Qualität und Repräsentativität der Daten sind entscheidend für die Vorhersagegenauigkeit. Daher sollten solche Methoden ergänzend und nicht als Ersatz für die Akquise realer, zusätzlicher Daten verwendet werden.

6 FAZIT

Die Untersuchung, ob der Kurs der Apple-Aktie für einen Tag in der Zukunft basierend auf historischen Daten zuverlässig vorhergesagt werden kann, ergab gemischte Resultate. Mittels LSTM-Netzwerken konnten gewisse Muster in den Daten identifiziert werden, doch die erzielte Vorhersagegenauigkeit erreichte nicht das Niveau für eine zuverlässige Anwendung. Die Arbeit offenbart die Herausforderungen der Aktienmarktprognose, einschließlich der Marktvolatilität mit seiner Vielzahl an externen Faktoren [5] und der Limitation des

gewählten Datensatzes. Obwohl die Ergebnisse Potenzial für die Nutzung von LSTM-Modellen in der Vorhersage zeigen, unterstreichen sie auch die Notwendigkeit für weiterführende Forschung, um die Genauigkeit der Modelle für Aktienvorhersagen zu verbessern und sie praxistauglich zu machen.

7 ANMERKUNGEN

Die Abschnitte 2, 5 und 6 wurden mit Unterstützung von ChatGPT erstellt, während die Formulierungen in allen anderen Abschnitten durch ChatGPT verbessert wurden (Verbessere die Formulierung dieses Textes: ...).

LITERATUR

- [1] Peter Bruce, Andrew Bruce und Peter Gedeck. 2017. *Praktische Statistik für Data Scientists*. O'Reilly, Heidelberg. ISBN: 978-3-96009-153-0.
- [2] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6, 02, 107–116.
- [3] Sepp Hochreiter und Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9, 8, 1735–1780. doi: 10.1162/neco.1997.9.8.1735.
- [4] N Pahwa, Neeha Khalfay, Vidhi Soni und Deepali Vora. 2017. Stock prediction using machine learning a review paper. *International Journal of Computer Applications*, 163, 5, 36–43.
- [5] Nirali Patel, Keta Patel und Mehul Patel. 2021. 2021-03-05 - review on stock market prediction using machine learning techniques full paper, (März 2021).
- [6] F. Pedregosa u. a. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [7] V Kranthi Sai Reddy. 2018. Stock market prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)*, 5, 10, 1033–1035.
- [8] Ritika Singh und Shashi Srivastava. 2017. Stock prediction using deep learning. *Multimedia Tools and Applications*, 76, 18, 18569–18584. doi: 10.1007/s11042-016-4159-7.
- [9] Yahoo Finance. 2024. Yahoo finance api. <https://pypi.org/project/yfinance/>. Accessed: 2024-02-19. (2024).