FERDINAND PORSCHE
FERNFH

Stock Market Forecasting DJIA
Prognose anhand neuronaler Netze

Masterarbeit

Eingereicht von: **Ing. Johannes Hatter, BA**

Matrikelnummer: 51905460

im Fachhochschul-Masterstudiengang Wirtschaftsinformatik

der Ferdinand Porsche FernFH GmbH

zur Erlangung des akademischen Grades

Master of Arts in Business

Betreuung und Beurteilung: Prof. Dr. Joachim Steinwendner

Zweitgutachten: Priv.-Doz. DI Dr. Zsolt Saffer

Wiener Neustadt, 05 2024

Ehrenwörtliche Erklärung

Ich versichere hiermit,

1. dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.
2. dass ich diese Masterarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.
3. dass die vorliegende Fassung der Arbeit mit der eingereichten elektronischen Version in allen Teilen übereinstimmt.

Wampersdorf, 20.05.2024 _____

Unterschrift

Kurzzusammenfassung:

Stock Market Forecasting DJIA - Prognose anhand neuronaler Netze

Adäquate Finanzmarktprognosen sind in kapitalistisch geprägten Marktwirtschaften sowohl für Kleinanleger als auch Investitionsexperten begehrte Werkzeuge. Dabei werden für die Vorhersage unterschiedliche Methoden und Informationen eingesetzt. Die gegenständliche Untersuchung widmet sich der Prognose des Dow Jones Industrial Average Aktienindex, mit dem Ziel, den Indexkursanstieg oder -abfall am Folgetag korrekt vorherzusagen. Neuartig an der hier gewählten Herangehensweise ist die automatisierte Verschränkung von Kleinanlegerstimmungen mit Einschätzungen von Finanzexperten, ergänzt um historische Aktienindex Kursinformationen. Die Indexvorhersage erfolgt anhand der mittels VADER-Sentimentanalyse gewonnenen Sentimentinformationen, extrahiert aus Postings der Social Media Plattform www.reddit.com, in Kombination mit qualitativer Inhaltsanalyse von Nachrichtenartikeln der New York Times und des Wall Street Journals. Zur Analyse der Finanzmarktnachrichten wurde das Latent Dirichlet Allocation-Verfahren eingesetzt. Ergänzend wurden die verfügbaren Handelsdaten zum Aktienindex selbst in die Vorhersage miteinbezogen. Als Prognosemodell wurde mittels Supervised-Learning-Verfahren ein Künstliches-Neuronales-Netz des Typs Long Short Term Memory entwickelt. Im Untersuchungszeitraum 2023 zeigte sich, dass anhand der gewählten Inputparameter und Prognosemodellarchitektur eine Vorhersagegenauigkeit des korrekten Kursanstiegs und -abfalls am Folgetag von 71,93%, sowie 95,71% für die Folgewoche erreicht werden kann.

Schlagwörter:

Aktienkursprognose; Dow Jones; Sentimentanalyse; Reddit; LDA; LSTM; NYT; WSJ;

Abstract: Stock market forecasting DJIA – prediction using neural networks

Adequate financial market forecasts are coveted tools for investors in capitalistic organized markets. Surveys examine different methods for stock market prediction. This study's goal is the forecast of the Dow Jones Industrial Average stock index, with the aim of correctly predicting the rise or fall of the index price on the following day. New about the chosen approach is the automated combination of small investor sentiments with assessments from financial experts, supplemented by historical stock index price information. The index prediction is based on sentiment-information obtained using VADER-sentiment-analysis, extracted from postings on the social media platform www.reddit.com, in combination with qualitative content analysis of news articles from The New York Times and The Wall Street Journal. Latent Dirichlet Allocation was used to analyse financial news. In addition, available trading data of the stock index itself was included into the forecast. A neural network of the LSTM type was developed as a forecast model using supervised learning methods. In the 2023 study period, it was shown that, based on the selected input parameters and model architecture, a prediction accuracy of 71.93% for the correct price rise or fall on the following day and 95.71% for the following week can be achieved.

Keywords:

Stock forecasting; Dow Jones; Sentiment analysis; Reddit; LDA; LSTM; NYT; WSJ;

Danksagung

An dieser Stelle möchte ich meiner Frau Daniela Danke sagen, die mich während meiner gesamten Studienzeit unterstützt und selbst zurückgesteckt hat, um mir mein Studium zu ermöglichen. Danke für deinen andauernden Support und dein Verständnis in den letzten fünf Jahren.

Inhaltsverzeichnis

1 Einleitung

Vorhersagen über zukünftige Entwicklungen am Finanzmarkt treffen zu können, ist eine der gefragtesten, wenn auch komplexesten Themenstellungen des Kapitalmarkts. Zuverlässige Prognosen sollen Unternehmen und Investoren dabei unterstützen, die korrekten Entscheidungen zu treffen, Gewinne zu erwirtschaften und Verluste zu vermeiden. Für viele Akteure am Finanzmarkt ist es daher von großem Interesse über ein möglichst genaues Prognosemodell zu verfügen. Die mittlerweile breit zur Verfügung stehenden Technologien rund um künstliche Intelligenz wie Deep Learning Verfahren werden im gegenständlichen Sektor bereits intensiv genutzt, unter anderem da dadurch bestehende statistische Modelle zur Korrelation- oder Kausalitätsermittlung ergänzt oder abgelöst werden können. Ein bekanntes System, welches bereits auf Machine Learning Algorithmen zurückgreift, ist etwa das Bloomberg Terminal (Bloomberg, 2023).

Untersuchungen historischer Ereignisse am Kapitalmarkt haben gezeigt, dass neben Finanzexperten und großen Hedgefonds auch Kleinanleger erheblichen Einfluss auf den Verlauf von Aktienkursen haben können (Bollen et al., 2011). Ein plakatives Beispiel der jüngeren Vergangenheit stellt etwa der GameStop Short Squeeze aus dem Jahr 2021 dar (Lyocsa et al., 2021). Aber auch zuvor kam es bereits zu ähnlichen Marktsituationen wie etwa dem Harlem Corner 1861 (Allen et al., 2017). Bollen et. al. zeigten 2011 in einer Studie, dass das Verhalten der Kleinanlegerschaft als gesammelt auftretender Akteur am Finanzmarkt nicht zu unterschätzen ist und weiterführend, unter bestimmten Umständen, auch Aufschluss über mögliche Kursentwicklungen bieten kann (Bollen et al., 2011). Im Jahr 2016 wurde veröffentlicht, dass der Einfluss von Social Media auf den Aktienmarkt „signifikant größer“ als jener der herkömmlicher Medienkanäle ist (Jiao & Walther, 2016). So bestehen etwa laut Brown et al. Korrelationen zwischen zeitgleichen Social-Media-Stimmungen und Aktienrenditen (Brown et al., 2004).

Aufgrund der Bedeutung, zukünftige Entwicklungen am Finanzmarkt vorhersagen zu können, gibt es eine schiere Menge an wissenschaftlichen Arbeiten, die sich der Konzeption, Entwicklung oder Validierung von Prognosemodellen widmen. So gibt es bereits Untersuchungen zur Entwicklung von Prognosemodellen, etwa für die Aktienkurse der DAX-Unternehmen anhand von Twitter-Sentimentdaten (Schröter, 2019) oder wie von Shilpa 2023 entwickelt ein Prognoseframework basierend auf Sentiment Analyse von Social Media Postings und vorhandenen Vorlaufindikatoren des Kapitalmarktes (Shilpa, 2023). Chakraborty et al wiederum erstellten im Jahr 2017 eine

Kursvorhersage von Apple und dem Dow Jones Index anhand einer Support Vector Machine und anderen Machine Learning Verfahren (Chakraborty et al., 2017). Im Rahmen dieser Studie wurden ebenfalls Sentimentdaten zur Aktienprognose miteinbezogen. Swathi et al erstellten 2022 auf Basis von Twitter Sentimentanalysedaten und historischen Aktienkursinformationen ein LSTM-Modell, welches Prognoseergebnisse für die täglichen Aktienkurse mit einer Vorhersagegenauigkeit von über 90% lieferte (Swathi et al., 2022). Die unterschiedlichen Prognosemethoden unterscheiden sich dabei hinsichtlich der Vorgehensweise, die vorliegenden Aktienkurse zu analysieren und zu prognostizieren. Nach Erkenntnissen von Ranco et al existiert zudem ein Zusammenhang zwischen der Social Media Stimmung und außergewöhnlichen Renditen (Ranco et al., 2015).

In den letzten Jahren erfuhren die Techniken rund um künstliche Intelligenz und neuronale Netze neuerliche Beliebtheit in diesem Sektor. Bisherige Untersuchungen zur Entwicklung von Prognosemodellen anhand neuronaler Netze basieren entweder rein auf vergangenen Aktienkursen, auf Social-Media-Vorhersagen (beispielhaft Schröter, 2019) oder auf Aussagen von Experten in Kombination mit Aktienkursentwicklungen (García-Méndez et al., 2023). Im Rahmen dieser Untersuchung wurde an die bereits existierenden Erkenntnisse zur Entwicklung von Prognosemodellen auf Basis der Informationsgewinnung von neuen Medien (Social Media, Online-Nachrichten etc.) angeknüpft. Es wurde untersucht wie durch Kombination von Sentimentanalysedaten, gewonnen aus Postings der Reddit-Community (Subreddit Wallstreetbets), sowie Artikeln aus renommierten Fachzeitschriften - der New York Times (NYT) und dem Wall Street Journal (WSJ), unter Hinzunahme der historischen Aktienindexkursdaten des Dow Jones Industrial Average- (DJIA) ein geeignetes Prognosemodell entwickelt werden kann, welches für ebendiesen Aktienindexschlusskurs anwendbar ist. Neuartig an der gegenständlichen Herangehensweise ist es, anhand eines neuronalen Netzes, im konkreten Fall eines Long Short Term Memory-Modells (LSTM), ein Prognosemodell mittels Prototyping aus Verschränkung von Kleinanlegermeinungen – Reddit-Kommentaren, Expertenmeinungen – Zeitungsartikel von WSJ und NYT, sowie historischen Aktienindexinformationen zu implementieren. Nähere Details zur Konzeption und Umsetzung finden sich in den Abschnitten 4 und 5.

1.1 Ziel der Arbeit

Als Hauptziel wird die Entwicklung eines Prognosemodellprototypen, der auf die nachfolgend beschriebenen Datenquellen zugreift und dabei eine Vorhersagegenauigkeit

von mindestens 60%, bezogen auf den täglichen Aktienindexschlusskurs des Dow Jones Industrial Average aufweist, festgelegt. Die Vorhersage gestaltet sich hierbei so, dass durch das Prognosemodell ein Schlusskursanstieg oder -abfall am Folgetag korrekt prognostiziert, werden können soll. Eine Abschätzung von Intra-Day Kursentwicklungen wurde nicht durchgeführt. Dieses zu entwickelnde Prognosemodell wurde auf Basis eines Künstlichen-Neuronalen-Netzes mit den untenstehenden Informationen konzipiert und angelernt. Zur Erreichung der oberhalb angeführten Vorhersagegenauigkeit von mindestens 60% greift der gegenständlich entwickelte Prognoseprototyp auf die folgenden Informationsquellen zurück:

- Historische Aktienindexinformationen des Dow Jones Industrial Average
- Kleinanlegermeinungen und -stimmungen zur Aktienkursentwicklung des betroffenen Aktienindex
- Expertenmeinungen zur zukünftigen und vergangenen Aktienindexentwicklung

Konkret lautet die untersuchte Forschungsfrage:

„Wie kann unter Einsatz Künstlicher-Neuronaler-Netze ein Prognosemodellprototyp für den Aktienindexschlusskurs des Dow Jones Industrial Average entwickelt werden, welcher auf digital abrufbare Kleinanleger- und Expertenmeinungen im Prognoseprozess Rücksicht nimmt?“

Die zugehörige aufgestellte Hypothese besagt, dass es durch Verknüpfung historischer Aktienindexinformationen des Dow Jones Industrial Average, ergänzt um Sentimentinformationen aus Redditkommentaren zu ebendiesem Index, sowie aus Ergebnissen, gewonnen durch qualitative Inhaltsanalyse mittels Latent Dirichlet Allocation, aus Artikeln der Fachmagazine New York Times und Wall Street Journal möglich ist (Blei et al., 2003), ein Prognosemodell auf Basis neuronaler Netze zu entwickeln, welches die Meinungen von Kleinanlegern und Investitionsexperten, sowie historischen Aktienindexkursen verbindet und zumindest temporär eine Vorhersagegenauigkeit von mindestens 60% erreicht. Jene Vorhersagegenauigkeit bezieht sich dabei auf den möglichen Schlusskursanstieg oder -abfall des Aktienindex am Folgetag.

Wenn auch nur als Kleinst-Player am Finanzmarkt ist es für mich und andere Kleinanleger von Interesse über ein adäquates Prognosemodell für Investitionsentscheidungen zu verfügen, weshalb die gegenständliche Untersuchung angestellt wurde.

1.2 Vorgehensweise

Zur Prüfung der aufgestellten Hypothese wurde eine Korrelationsstudie angefertigt, welche anhand eines Prototyping Verfahrens einen Prognosemodellprototypen auf Basis eines neuronalen Netzes des Typs LSTM hervorgebracht hat. Um dies zu erreichen, war es erforderlich einen breiten Baukasten an wissenschaftlichen Methoden anzuwenden. Initial wurden die benötigten Inputfaktoren definiert, eine Auswahl der Programmiersprache für die Implementierung getroffen und es erfolgte das Erheben des benötigten Datenmaterials der einzelnen Eingabevariablen. Aufgrund der breiten bereits existierenden APIs fiel die Wahl zur Implementierung des Prototyps in Python. Die benötigten Aktienindexinformationen als erster Inputstream wurden über die deutsche Finanzmarktplattform www.ariva.de eingeholt (ariva, 2023).

Einen weiteren Inputfaktor für das Prognosemodell stellen die aggregierten Kleinanlegermeinungen und -stimmungen zum untersuchten Aktienindex dar. Dafür wurden sämtliche Kommentare zu Submissions mit Dow Jones Industrial Average Bezug von der Plattform www.reddit.com herangezogen. Dabei wurde auf den Subreddit /r/Wallstreetbets zurückgegriffen. Eine von Bollen, Mao und Zeng 2010 durchgeführte Studie hat gezeigt, dass insbesondere Social Media Postings und deren Sentimentinformationen den Aktienkurs bestimmter Unternehmen beeinflussen können (Bollen et al., 2011). Auch andere Untersuchungen wie etwa von Dounis 2020 am Beispiel von Tesla belegen, dass Korrelationen zwischen Social Media Aktivitäten von Anlegern und der NYSE bestehen können (Dounis, 2020). Jene auf diesem Weg generierten Daten wurden weiterführend Textmining-Verfahren und einer Sentimentanalyse unterzogen, welche Aufschluss zum Stimmungsbild der Kleinanlegerschaft zum gegenständlichen Aktienindex lieferten. Für die Sentimentbeurteilung selbst, wurde der lexikalische Ansatz Valence Aware Dictionary and sEntiment Reasoner (VADER) von Hutto & Gilbert verwendet. Hutto & Gilbert entwickelten mit VADER ein Stimmungslexikon, das dezidiert für die Verwendung von Beiträgen aus sozialen Medien konzipiert wurde (Hutto & Gilbert, 2014).

Der dritte Eingabeblock wird übergreifend als „Expertenmeinungen zur zukünftigen und vergangenen Aktienindexentwicklung“ definiert. Dahinter verbirgt sich die Hinzunahme von mittels qualitativer wissenschaftlicher Methoden, im konkreten der Systematik der Inhaltsanalyse nach Mayring folgend (Mayring, 2010), gewonnene tagesaktuelle Informationen zum Dow Jones Industrial Average aus den Fachzeitschriften Wall Street Journal und New York Times. Um die Schritte maschinell durchzuführen, kam das von

David Blei entwickelte LDA „Latent Dirichlet Allocation"-Verfahren zum Einsatz (Blei et al., 2003). Das Miteinbeziehen der online abrufbaren Inhalte aus den Zeitschriften New York Times und Wall Street Journal diente als Gegengewicht zur Markteinschätzung und Meinungsbildung der Kleinanleger. Dabei wurden vorhandene APIs und Webscraping Verfahren eingesetzt, um von New York Times und Wall Street Journal aktuelle und historische Online-Artikel zu den DJIA-Unternehmen im Untersuchungszeitraum abzufragen (Pal et al., 2021).

Die so gewonnenen Eingabedaten wurden für die Nutzung im LSTM bereinigt und vorbereitet. Dazu war es erforderlich, die nicht benötigten Informationen zu entfernen und Data Clearing Systematiken zu definieren. Details dazu finden sich in Abschnitt 4.2.

Nach der Datenaufbereitung erfolgte die Konzeption und Entwicklung des neuronalen Netzes als Grundlage des Prognosemodellprototypen. Studien zeigten, dass sich für die gegenständliche Aufgabenstellung rekurrente neuronale Netze eignen (Wiesinger, 2021), weshalb auch in dieser Untersuchung darauf zurückgegriffen wurde. Im konkreten wurde ein neuronales Netz vom Typ Long Short Term Memory (LSTM) eingesetzt(Hochreiter & Schmidhuber, 1997). Jenes hat gegenüber herkömmlichen rekurrenten neuronalen Netzen der Vorteil, dass innerhalb eines Neurons zusätzlich zum Output des vorhergegangenen Verarbeitungsschrittes, ein interner Status existiert, der ebenfalls in die Verarbeitung innerhalb des Neurons einfließt. Zum Anlernen des neuronalen Netzes wurde ein Supervised Learning Verfahren mit Gradientenabstieg eingesetzt (Kubiak, 1991) (Swathi et al., 2022). Das anhand der einleitend beschriebenen Inputfaktoren angelernte Künstliche-Neuronale-Netz wurde als Prototyp des Prognosemodells herangezogen, um die Forschungsfrage zu beantworten.

Im Rahmen des Prototyping-Verfahrens stellte sich unter Modifizierung der LSTM-Modellparameter heraus, dass es anhand der durch die vorherigen Schritte generierten 372 Eingabevariablen möglich ist, eine Prognosequote von 71,93% für die korrekte Vorhersage eines Dow Jones Industrial Average Schlusskursanstiegs oder -abfalls für den Folgetag zu erzielen. Zudem wurde erkannt, dass eine Prognosequote von bis zu 95,71% für die Folgewoche, bezogen auf Schlusskursanstieg oder -abfall erreicht werden kann. Es wurde weiterführend festgestellt, dass anhand der gewählten Inputparameter je nach gewünschtem Prognosehorizont, die Modellparameter des Prognose-LSTMs individuell ausgelegt werden müssen, und nicht das eine Modell für alle Anwendungsfälle existiert. Nähere Details zu den Erkenntnissen können dem Kapiteln 5 und 6 entnommen werden.

1.3 Gliederung der Arbeit

Im anschließenden Abschnitt wird der aktuelle Stand von Wissenschaft und Technik beschrieben. Dabei erfolgt eine Fokussierung auf ausgewählte Kapitalmarktgrundlagen, um ein besseres Bild über die zugrundeliegenden volkswirtschaftlichen Einflussfaktoren darzustellen. Daran reiht sich ein Überblick zu Möglichkeiten der Textanalyse und welche Verfahren für die Bearbeitung des gegenständlichen Untersuchungsinhalts zweckdienlich sind. Die Aufarbeitung der derzeitigen wissenschaftlichen Grundlagen schließt mit einer Beschreibung zur Konzeption und Validierung Künstlicher-Neuronaler-Netze (KNN). Dabei findet eine Fokussierung auf rekurrente neuronale Netze statt. Anschließend wird theoretisch erläutert, welcher Lösungsweg zur weiteren Beantwortung der Forschungsfrage gewählt wurde. Im Detail wird dabei auf bereits angestellte Untersuchungen und Prognosemodelle, sowie die durchzuführenden Arbeitsschritte und Prognoseverfahren eingegangen. Im darauffolgenden Kapitel der Arbeit werden die Ergebnisse des zuvor entwickelten Lösungswegs für die Prognose des Dow Jones Industrial Average Aktienindex im Betrachtungszeitraum 2023 beschrieben. Die Untersuchungsergebnisse werden analysiert, gegen die Hypothese geprüft und zusammengefasst. Erkenntnisse und mögliche Folgemaßnahmen werden gelistet. Abschließend folgt die Conclusio, sowie eine Handlungsempfehlung.

2 Stand der Wissenschaft und Technik

Um ein besseres Verständnis für den gegenständlich entwickelten Prognoseprototypen zu schaffen, wird in den folgenden Kapiteln auszugsweise auf bestimmte Themengebiete von volkswirtschaftlicher, mathematischer und informationstechnischer Relevanz eingegangen. Dies dient dazu, die Grundlagen und den aufbauenden Stand aus Wissenschaft und Technik, zum gewählten Lösungsweg, angeführt in Kapitel 4, darzulegen. Dabei werden relevante Kapitalmarktgrundlagen, Verfahren zu Text Mining und Sentimentanalyse, sowie der qualitativen Inhaltsanalyse und der Latent Dirichlet Allocation erörtert. Weiters wird ein kompakter Überblick zu maschinellen Lernmethoden mit Fokussierung auf Künstliche-Neuronale-Netze gegeben. Daran anknüpfend wird eine Kurzfassung zu bisher auf Basis neuer Medien angestellten Prognosen des Dow Jones Industrial Average oder anderer Kapitalmarktwerte präsentiert. Das Kapitel schließt mit einer Übersicht über bisher stattgefundene Untersuchungen zur Prognose von Aktienkursen anhand neuronaler Netze. Begonnen wird einleitend mit einem Einblick in die benötigten Kapitalmarktgrundlagen.

2.1 Ausgewählte Kapitalmarktgrundlagen

Im Rahmen der weiteren Untersuchung wird angenommen, dass grundlegende Kapitalmarktkenntnisse vorliegen, weshalb darauf nur kurz eingegangen wird und nachfolgend ausschließlich der grundlegende Verlauf eines Aktienkurses beschrieben wird, als auch eine detailliertere Darlegung des Dow Jones Industrial Average erfolgt.

2.1.1 Verlauf eines Aktienkurses

Fama stellte 1970 mehrere Hypothesen zum Aktienmarkt auf, die bis heute die vorherrschende Meinung zu deren Entwicklung und Verlauf prägen. Dabei wird angenommen, dass der Aktienkurs einem Zufallsprozess folgt (Fama, 1970). Aufgrund dieser Annahme war es bis heute nicht möglich ein Verfahren zu entwickeln, welches eine vollständig adäquate Vorhersage eines Aktienkurses ermöglicht. In einer Vielzahl von Untersuchungen wird dies auf die Markteffizienzhypothese von Fama zurückgeführt. Jene besagt, dass zu jeder Zeit alle für den Markt relevanten und aktuellen Informationen bereits im Aktienkurs eingepreist sind (Fama, 1970). Anders ausgedrückt bedeutet dies, dass sich relevante Kursveränderungen nur durch neue Informationen ergeben, welche von der Anlegerschaft aber nicht vorhersagbar oder antizipierbar sind (Bollen et al., 2011). Unter Berücksichtigung der eingangs erwähnten Hypothese von Fama entstehen diese

neuen Informationen zufällig und führen zu einer Zufallsbewegung der Aktienkurse (Qian & Rasheed, 2007).

Die gegenständliche Untersuchung folgt weiterführend dem Ansatz von Schröter welcher besagt, dass Sentimentinformationen, extrahiert von der Plattform Reddit neue Informationen darstellen, welche in dieser Form noch nicht in den Aktienkursen berücksichtigt wurden. Zusammenhängende Kursbewegungen erfolgen daraus schließend nicht gänzlich zufällig und es lassen sich dadurch neue Informationen für Prognoseszenarien extrahieren (Schröter, 2019).

2.1.2 Dow Jones Industrial Average Aktienindex

Im Jahr 1884 von den Gründern des Wall Street Journals Charles Dow und Edward Jones geschaffen, stellt der Dow Jones Industrial Average (DJIA) einen der ältesten noch am Kapitalmarkt gehandelten Indizes dar. Seine ursprüngliche Entwicklung geht auf das Bestreben von Charles Dow zurück, welcher mit dem „Dow Jones Railroad Average" im Jahr 1884 den Vorgänger des heutigen Dow Jones Industrial Average schuf, um damit die Entwicklung des US-amerikanischen Aktienmarkts zu messen. Die erste Veröffentlichung des „Dow Jones Railroad Average" fand am 03. Juli 1884 im „Customers' Afternoon Letter", dem ersten Börsenbrief der heutigen Art, statt. Im Gegensatz zur heutigen Zusammensetzung des DJIA bestand der damalige Index aus neun Eisenbahngesellschaften, einer Geldtransfergesellschaft und einer Dampfschifffahrtsgesellschaft. Um die Performance des Aktienmarkts zu messen, bezog er die Aktienkurse ebendieser Unternehmen, addierte sie und dividierte die erhaltene Summe durch die Anzahl der im Index enthaltenen Index (Heinemann, 2004). Am Berechnungsprinzip hat sich bis heute grundlegend nichts geändert. Beim heute veröffentlichten Dow Jones Industrial Average handelt es sich weiterhin um einen rein preisgewichteten Index, dessen Stand sich ausschließlich aus den enthaltenen Aktienkursen ermittelt wird. Dabei werden Größen wie Dividenden, Bezugsrechte, Sonderzahlungen, Marktkapitalisierung oder Anzahl der Aktien im Streubesitz nicht berücksichtigt. Aktien mit stärkerem Kurs wirken somit stärker auf den Index als jene mit niedrigem Kurs (Barrons, 2023).

Der Down Jones wird nach folgender Formel berechnet:

Formel 1: Dow Jones Industrial Average Ermittlung (vgl. Barrons, 2023)

$$DJIA = \frac{\sum p}{d}$$

p – Kurs der Einzelwerte
d – Dow-Divisor

Anders als am Berechnungsprinzip hat sich jedoch der Divisor und dessen Ermittlung im Laufe der Jahre verändert. Stattfindende Änderungen der Aktienkurse von Einzelunternehmen des Index, welche aufgrund von Veränderungen in der Index-Zusammensetzung, wie Auswechslungen von Index-Teilnehmern oder durch Kapitalmaßnahmen der Unternehmen, wie Aktiensplits, -zusammenlegungen, Übernahmen oder Ausgliederungen, auftreten, dürfen zwangsläufig allein nicht zu einer Veränderung des Indexwertes führen. Daher wird der Dow-Jones-Divisor als Folgeerscheinung auf Ereignisse dieser Art so angepasst, dass der Index unmittelbar durch diese Maßnahmen idealerweise keine Veränderungen erfährt. Der aktuelle Divisor wird unter anderem von Barron's, einem Magazin des Wall Street Journals, veröffentlicht und berechnet sich wie folgt (Barrons, 2023*)*:

Formel 2: Berechnung Dow-Jones-Divisor (vgl. Barrons, 2023)

$$d_{neu} = d_{alt} * \frac{\sum C_{neu}}{\sum C_{alt}}$$

d_{neu} *– Divisor, der nach Umsetzung von Maßnahmen gilt*
d_{alt} *– alter Divisor vor Maßnahmen*
C_{neu} *– Schlusskurs der Komponenten für den gleichen Handelstag, mit Maßnahmenumsetzung*
C_{alt} *– Schlusskurse der Komponenten für einen Handelstag vor Maßnahmenumsetzung*

Zum Ende des 19. Jahrhunderts prosperierte die US-amerikanische Wirtschaft. Durch zahlreiche Unternehmensübernahmen gingen große Industrieunternehmen hervor. Aus dieser Marktentwicklung heraus entstand der bis heute veröffentlichte Index des Dow Jones Industrial Average, auch als Dow-Jones-Index bezeichnet. Der heutige Index setzt sich aus 30 der größten US-Unternehmen zusammen, welche nach Ermessen des Herausgebers S&P Dow Jones Indices ausgewählt werden und weiterführend keinen festen Regeln in der Zusammenstellung folgen (Heinemann, 2004*)*. Mit Stand März 2023 enthält der Index die folgenden Unternehmen.

Tabelle 1: Zusammensetzung Dow Jones Industrial Average (Barrons, 2023)

Unternehmensname	**Indexgewicht in Prozent**
UnitedHealth	9,61
Microsoft	6,56
Goldman Sachs	6,51
Home Depot	5,77
McDonald's	5,74
Visa	4,49
Amgen	4,37
Caterpillar	4,22
Salesforce	4,22
Boeing	4,04
Honeywell International	3,87
Travelers	3,49
Apple	3,48
Chevron	3,11
Johnson & Johnson	3,11
American Express	3,04
Walmart	2,94
Procter & Gamble	2,92
JP Morgan Chase	2,73
IBM	2,55
Merck & Co.	2,26
Nike	2,16
3M	1,95
Disney	1,77
Coca-Cola	1,22
Dow	1,01
Cisco	0,99
Verizon Communications	0,70
Walgreens Boots Alliance	0,60
Intel	0,55

2.2 Text Mining

Im nachfolgenden Kapitel werden die beiden Bereiche des Textminings erläutert, auf welche im Rahmen dieser Untersuchung zurückgegriffen wird. Es handelt sich hierbei um die Sentimentanalyse von Social Media Daten (siehe Kapitel 2.2.1) und die qualitative Inhaltsanalyse nach Mayring für die Kategorisierung der Artikel von New York Times und Wall Street Journal (siehe Kapitel 2.2.2 und 2.2.3). Es werden dabei jeweils die Grundlagen beschrieben und weshalb diese Systematik im Rahmen der gegenständlichen Arbeit angewandt wird.

Text Mining, stellt eine Subkategorie des Data Minings dar, welches sich mit der maschinenbasierten Analyse und Informationsgewinnung aus großen Textmengen befasst (Kolb, 2011). Dabei greifen Text Mining Verfahren auf sprachstatistische Methoden, sowie Verfahren des Natural Language Processings (NLP) zur Textanalyse zurück. In einem breiter gefassten Kontext sucht Data Mining relevante Informationen in Daten, Text Mining in Sprachdaten. Im deutschsprachigen Raum werden Text Mining Verfahren daher auch zu den Anwendungsfällen der Computerlinguistik (CL) oder linguistischer Datenverarbeitung (LDV) gezählt (Kolb, 2011).

2.2.1 Sentimentanalyse

Als Bestandteil von Text Mining ist es der Zweck der Sentimentanalyse oder Opinion Mining, Meinungsäußerungen in neuen Medien, Foren oder anderen Plattformen automatisch zu erkennen, zu klassifizieren und damit letztlich neues Wissen über Meinungen zu extrahieren (Schröter, 2019).

Ein beliebter Anwendungsfall für Sentimentanalysen ist der Social Media Bereich. Es können durch diese Methodik etwa Kommentare und deren Zusammenhänge untersucht werden. Weitere Anwendungsfälle sind etwa die Gewinnung von neuen Informationen aus umfangreichen Online-Bibliotheken von Schriftstücken. Das Ziel der Sentimentanalyse ist es, die Polarität einer Aussage (z.B.: positiv, neutral, negativ) oder deren Subjektivität maschinenbasiert aus Texten zu generieren (Kolb, 2011).

Stimmungen können eindimensional, unter anderem mithilfe mathematischer Methoden wie dem arithmetischen Mittel, Summen der Maxima oder Medianen gebildet werden (Basiri et al., 2014). Andere Methoden führen zu einer Stimmungsinterpretation in mehr Dimensionen. Arvidsson verwendet dazu etwa die Parameter der Stärke einer Nachricht, die Häufigkeit der Erscheinung und der Netzwerkzentralität (Arvidsson, 2011).

Sentimentanalyseverfahren werden auch in der gegenständlichen Arbeit eingesetzt, um die Polaritäten der Kommentare auf der Social Media Plattform Reddit als Inputstream für den Prognosemodellprototypen zu extrahieren und deren Auswirkung auf den DJIA-Kurs korrekt zu kategorisieren. Um natürliche Sprache in Form von Text- und Sprachdaten mit Hilfe eines Computers algorithmisch verarbeiten zu können, bedarf es einer Reihe von Arbeitsschritten, die sequenziell durchlaufen werden müssen. Erst dieses geordnete Vorgehen ermöglicht es, korrekte Rückschlüsse aus dem untersuchten Datenmaterial zu ziehen. Eine hierfür geeignete Vorgehensweise ist das Saarbrücker Pipelinemodell (Kolb, 2011).

Saarbrücker Pipelinemodell

Das Vorgehen des Saarbrücker Pipelinemodells gliedert sich in mehrere Schritte. Initial erfolgt die Umwandlung von Schallinformation mittels Spracherkennung in Text, sofern der Text nicht bereits als solcher vorliegt. Nachfolgend wird der Untersuchungstext in Wörter, Sätze etc. segmentiert. Im Falle von Zeichenfolgen aus neuen Medien ist hierbei insbesondere auf das Entfernen von etwaigen Hyperlinks, Sonderzeichen oder Emoticons zu achten. Gerade Emoticons werden hierbei für das Training von Algorithmen maschinellen Lernens besonders berücksichtigt (Go et al., 2009). Ein Emoticon wird dabei in die zugrundeliegende Darstellung eines meist lachenden oder traurigen Gesichts, generiert aus Standardzeichen zerlegt. Dieser Prozessschritt wird als Tokenisierung bezeichnet (García-Méndez et al., 2023). Anschließend erfolgt eine morphologische Analyse. Dabei werden Personalformen und Fallmarkierungen analysiert und Wörter in Ihre Stammform zurückgeführt. Beispielhaft wird aus dem Begriff „lief“, wie in „Er lief zum Fußballtor“, durch morphologische Analyse „laufen“ gebildet. Nachfolgend wird eine syntaktische und semantische Untersuchung der Textblöcke durchgeführt. Die zuvor beschriebenen Schritte können in beliebiger Iterationsanzahl durchgeführt werden. Bei Bedarf kann darauf aufbauend eine Dialog- und Diskursanalyse implementiert werden, welche die Beziehungen zwischen aufeinanderfolgenden Sätzen untersucht (Xing et al., 2018). Das Saarbrücker Pipelinemodell ist so konzipiert, dass nicht immer sämtliche Verfahren der Computerlinguistik die gesamte Kette durchlaufen müssen. Die zunehmende Verwendung von maschinellen Lernverfahren hat zu der Einsicht geführt, dass auf jeder Analyseebene statistische Regelmäßigkeiten existieren, welche zur Sprachmodellierung eingesetzt werden können (Kolb, 2011) (Piryani et al., 2017).

Der Aufbau der Datenstruktur neuer Medien erschwert die Realisierung von darauf basierenden Prognosemodellen. Soziale Medien enthalten unstrukturierte Daten

(Gundecha & Liu, 2012). Für eine computerbasierte Verarbeitung erschwert werden die Umstände zudem durch eventuell enthaltene Werbung oder mehrdeutige Ausdrücke (Han & Baldwin, 2011). Da im gegenständlichen Fall auf die Plattform Reddit, konkret den Subreddit /r/Wallstreetbets zurückgegriffen wird, sind hierbei die Besonderheiten dieser Plattform zu berücksichtigen. Aufgrund des Einsatzes von Kurznachrichten verfügen Social Media Plattformen über eine eigene Syntax (Zhang et al., 2011.). Durch die Kürze der Nachricht, Grammatikfehler, Jargon und Mehrsprachigkeit, ergeben sich zusätzliche Herausforderungen in der computerbasierten Analyse. Eine Studie von Brody et al. aus dem Jahr 2011 hat sich mit diesem Themenbereich befasst und etwa herausgefunden, dass ein Wort mit subjektiver Bedeutung, wenn es in die Länge gezogen wird, stimmungsverstärkend wirkt (z.B.: „Woooow“) (Brody & Diakopoulos, 2011).

Ein auf das Saarbrücker Pipelinemodell aufbauendes regelbasiertes Modell zur allgemeinen Stimmungsanalyse wurde von Hutto und Gilbert mit dem Namen VADER – „Valance Aware Dictionary for sEntiment Reasoning (VADER)“ entwickelt. Speziell konzipiert für die Analyse von Texten im Stil von sozialen Medien lässt es sich auf Plattformen wie Reddit, Nachrichtendiensten wie Twitter oder Bewertungsplattformen anwenden (Hutto & Gilbert, 2014). Kern des Modells ist ein generalisierbares, auf Wertigkeiten basiertes und von Menschen organisiertes Lexikon für die Polaritätsbeurteilung. Hutto und Gilbert bewiesen dabei in ihrer Untersuchung aus dem Jahr 2014, dass mit ihrem Modell eine ähnliche Präzision bei der Stimmungsbeurteilung von Beiträgen aus sozialen Medien erreicht werden kann, wie durch menschliche Klassifizierung (Hutto & Gilbert, 2014).

2.2.2 Qualitative Inhaltsanalyse

Unter qualitative Inhaltsanalyse werden Vorgehensmodelle zusammengefasst, die eine systematische Bearbeitung von Texten beschreiben, um in wissenschaftlichem Kontext ein Forschungsziel zu erreichen. Als Methode der empirischen Forschung wird es vor allem in der Sozialforschung dazu eingesetzt, neue Erkenntnisse aus Textmaterial zu extrahieren (Mayring, 2014).

Eine renommierte Methodik zur Durchführung einer qualitativen Inhaltsanalyse wurde vom deutschen Psychologen Philipp Mayring entwickelt. Sein Modell der Mayring-Inhaltsanalyse beinhaltet fünf Schritte zur Durchführung einer solchen Untersuchung (Mayring, 2010). Im ersten Schritt ist das zugrunde zulegende Material auszuwählen. Im Falle der gegenständlichen Untersuchung stellt dies die Onlineartikel der beiden

Zeitungen Wall Street Journal (WSJ) und New York Times (NYT) dar. Anschließend sind die daraus abzuleitenden Untersuchungsziele zu definieren. Anhand der festgelegten Analyseziele erfolgt die Wahl der Grundform. Es wird hierbei zwischen zusammenfassender, explizierender und strukturierender Inhaltsanalyse unterschieden. Für die aktuelle Untersuchung ist die Grundform der strukturierenden Inhaltsanalyse entscheidend, da aus den Informationen der News-Plattformen die Artikel kategorisiert werden sollen, um einen eigenen Eingabestrom für das neuronale Netz zu bilden. Das Ziel ist es, Kategorien aus den Texten abzuleiten, die es ermöglichen sollen, in Kombinationen mit den weiteren Eingabevariablen, einen Kursanstieg oder -abfall vorherzusagen. Die Kategorisierung der untersuchten Texte erfolgt anhand eines Kodierleitfadens der induktiv oder deduktiv implementiert werden kann. Am Ende der Analyse gilt es die Gütekriterien der qualitativen Forschung – Transparenz, Reichweite und Intersubjektivität sicherzustellen (Mayring, 2010).

Zur Analyse großer unbekannter Textmengen eignet sich ein induktives Vorgehen mit einer Kategorienbildung während der Analyse (Mayring, 2014).

2.2.3 Latent Dirichlet Allocation

Eine angelehnte Vorgehensmethode um dies automatisiert computerbasiert durchzuführen stellt die von David Blei et al entwickelte Methode im Bereich des Topic Modeling (der Kategorienbildung) dar. Das von ihnen entwickelte Verfahren wird „Latent Dirichlet Allocation (LDA)“ bezeichnet und zeigte in bereits durchgeführten Untersuchungen adäquate Ergebnisse in der Analyse großer Mengen von Nachrichtenartikeln (García-Méndez et al., 2023) (Gupta et al., 2022). LDA stellt ein dreistufiges, hierarchisches Baiyessches Wahrscheinlichkeitsmodell dar, welches es ermöglicht effizient aus großen Datenmengen eine Kategorisierung abzuleiten (Blei et al., 2003). Es handelt sich dabei um ein Unsupervised-Learning Verfahren, welches auf dem Grundsatz basiert, dass Texte aus wenig unterschiedlichen Themen bestehen und zu jedem Thema des Texts bestimmte Wörter gehören. Ein Thema stellt dabei nach der Latent Dirichlet Allocation eine Wahrscheinlichkeitsverteilung für eine Gruppe von Wörtern („bag of words“) dar (Blei et al., 2003). Die so entstehende Themenmodellierung ist vergleichbar mit dem Clustern nummerischer Daten. Durch dieses Clustering lassen sich weiterführend latente Themen eines Dokuments identifizieren (Blei et al., 2003). Gerade im Finanzsektor wurde diese Systematik bereits in einer Vielzahl von Untersuchungen zielführend angewandt, um Marktprognosen auf Basis unterschiedlichster Informationsquellen anzustellen. Eine Studie von Garcia-Mendez et

al 2023 untersuchte etwa anhand von über 2000 Finanzmarktnachrichten unter Einsatz von LDA, ob es möglich ist, relevante darin angeführte Marktereignisse zu identifizieren und diesen Ereignissen zugehörige Eintrittswahrscheinlichkeiten zuzuordnen. Besonders an dieser Untersuchung war, dass die automatisiert als relevant ermittelten Informationen, menschlichen, mittels qualitativer Inhaltsanalyse untersuchten Ergebnissen, gegenübergestellt wurden. Dabei zeigte sich, dass unter Anwendung der Data Clearing Schritte (siehe auch Saarbrücker Pipelinemodell) und automatisierter Kategorienbildung nach LDA eine vergleichbare Aussagekraft erreicht werden kann (García-Méndez et al., 2023).

Eine weitere Untersuchung, welche auf Text Mining Basis mittels LDA durchgeführt wurde, war etwa die Intra-Day-Aktienprognose unter Hinzunahme von Finanznachrichten (Reuters US) nach Atkins et al 2018 (Atkins et al., 2018). Differenzierte Studien betrachteten Börsenstandorte wie New York oder London und untersuchten dabei den Einfluss, den soziale Medien und Nachrichten auf die Börsen selbst und die darauf gehandelten Einzelaktien nehmen können. Dabei zeigte sich, dass die New Yorker Börse stärker mit Aktivitäten aus sozialen Medienkanäle korreliert, während die Londoner Börse stärkere Zusammenhänge mit Nachrichtenberichten aufweist (Khan et al., 2022). Exakt an diese Verschränkung von mehreren Informationsquellen aus dem Umfeld der neuen Medien schließt auch diese Untersuchung an.

In der gegenständlichen Arbeit wurde die Vorgehensweise der qualitativen Inhaltsanalyse mit induktiver Kategorienbildung eingesetzt. Dabei wurde nicht auf klassischem Weg menschlich die Mayring-Inhaltsanalyse durchgeführt, sondern es wurden computerbasierte Algorithmen zur automatisierten Kategorienbildung (Topic Mining) eingesetzt. Nach den Data Clearing Schritten erfolgte als zentrales Element das LDA-Verfahren nach Blei et al (Blei et al., 2003). Hierbei wurde insbesondere auf die Erkenntnisse der bisherigen Forschung aufgebaut, um eine effiziente Kategorienbildung zu erreichen (García-Méndez et al., 2023).

2.3 Neuronale Netze

Im nachfolgenden Abschnitt wird einleitend auf die Entstehung neuronaler Netze als Teilsegment der künstlichen Intelligenz eingegangen. Anschließend folgt ein Überblick zu aktueller Forschung im Zusammenhang von Aktienkursprognosen mithilfe neuronaler

Netze. Das Kapitel führt zudem auf den im Kapitel 4.3 gewählten Lösungsweg ein, und beschreibt, weshalb genau diese Vorgehensweise als probat erachtet wurde.

Unter dem Terminus künstliche Intelligenz (KI oder engl. AI für artifical intelligence) wird ein Teilgebiet der Informatik verstanden, welches sich mit der Simulation von Verhaltensmustern beschäftigt, die von Menschen als intelligent wahrgenommen werden (Steinwendner & Schwaiger, 2020). Im weiteren Sinne geht es darum Maschinen „Denken“ zu lassen (Chollet, 2021). Daraus abgeleitet kann künstliche Intelligenz als Bestreben, intellektuelle Aufgaben, die normalerweise durch Menschen durchgeführt werden, zu automatisieren, definiert werden. Um in diesem Zusammenhang Intelligenz zu definieren kann auf den Turing Test, entwickelt von Alan Turing verwiesen werden. Jener beschreibt Intelligenz wie folgt: „Als intelligent gilt ein Algorithmus, wenn ein Mensch der schriftlich Fragen gestellt hat, nicht erkennen kann, ob die gegebene Antwort auf seine Frage von einem Mensch oder Maschine kommt“ (Turing, 1950). Im Zeitraum von 1950 – 1980, mit Höhepunkt im Rahmen des „Expert systems booms“ Ende der 1980er Jahre, fokussierte sich die Forschung rund um künstliche Intelligenz auf den Bereich der symbolischen KI. Darunter wird verstanden, dass Systeme durch explizit vorhandenes Wissen, welche nach mathematischer Logik verarbeitet wurden, generiert werden, um Daten zu manipulieren. Symbolische KI zeigte sich bereits damals als adäquat zur Lösung klar definierter, logischer Problemstellungen (z.B. eine Runde Schach). Zur Lösung komplexerer, undurchsichtiger Fragestellungen ist es jedoch nicht geeignet (Chollet, 2021). Neben der symbolischen KI hat sich zudem das Themengebiet der subsymbolischen künstlichen Intelligenz hervorgebildet, bei welchem das Wissen implizit repräsentiert wird (Steinwendner & Schwaiger, 2020). Dem Menschen sind hierbei keine expliziten Regeln zu den stattfindenden Wissensrepräsentationen und daraus erzielten Ergebnissen bekannt. Zu diesem Gebiet zählen auch die nachfolgend detailliert erläuterten Künstlichen-Neuronalen-Netze.

Machine Learning

Als Teil dieser subsymbolischen KI wird auch Machine Learning (ML) verstanden (Chollet, 2021). Im Gegensatz zu bisherigen Programmierparadigmen, einem Computerprogramm über mitgeteilte Regeln dazu zu verhelfen, die Inputdaten in die gewünschten Outputdaten zu verwandeln, änderte sich diese Herangehensweise bei Machine Learning Systemen grundlegend. Jene werden trainiert, anstatt explizit programmiert zu werden. Dabei werden der Maschine Inputdaten und die korrespondierenden Outputdaten präsentiert. Die Maschine bildet anhand der

bereitgestellten Daten die benötigten statistischen Strukturen und Rechenschritte, die es ermöglichen vom gegebenen Input zum gegebenen Output zu gelangen. Beispielhaft kann anhand dieser Vorgehensweise das Klassifizieren von Urlaubsbildern automatisiert werden. Durch Präsentation von bereits von Menschenhand klassifizierten Bildern kann das Machine Learning System über statistische Regeln automatisiert die Bilder zu gewünschten Klassen zuordnen (Chollet, 2021).

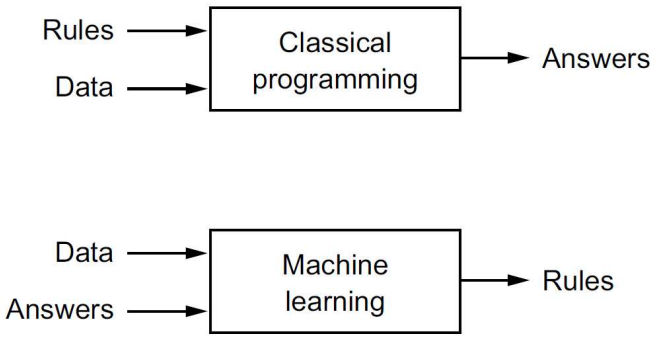


Abbildung 1: Machine Learning Programmierung (Chollet, 2021)

Ein Machine Learning System transformiert somit Inputinformationen in Ouputinformationen. Dieser Prozess findet anhand des Lernens aus vorhandenen Beispielen statt. Die Kernherausforderung hierbei ist die sinnvolle Transformation von Daten – anders formuliert, das sinnvolle Extrahieren von Repräsentationen der Inputdaten, die das Modell näher zum gewünschten Output bringen. Es handelt sich dabei bei einigen Machine Learning Verfahren, wie den gegenständlich eingesetzten neuronalen Netzen, um ein iteratives Vorgehen, um mittels der extrahierten Repräsentationen immer näher an, dem gewünschten Output ähnelnde, Repräsentationen der Daten zu gelangen. Dies erfolgt so lange, bis der gewünschte Output erreicht wurde. Dabei wird ergänzend ein Feedbackmechanismus eingesetzt, um aus den präsentierten Daten und den entstehenden Repräsentationen zu „lernen“. Machine Learning Algorithmen sind dabei nicht kreativ, sondern nutzen für die Transformation dieser Repräsentationen einen vordefinierten Werkzeugkasten von Operationen, der auch als „hypothesis space“ bezeichnet wird (Tetzner et al., 2021).

In der Praxis wird einem Machine Learning (ML) Algorithmus daher ein Trainingsdatensatz präsentiert, welcher durch den Algorithmus nach Mustern und Zusammenhängen durchsucht wird (Sarker, 2021). Nach Abarbeitung des Trainingsdatensatzes wird das trainierte ML-Modell dazu genutzt, unbekannte Daten zu bewerten. Zum sogenannten Anlernen von Machine Learning Modellen gibt es drei

Methoden. Diese Methoden unterscheiden sich darin, wie das entsprechende Modell angelernt wird. Man unterscheidet zwischen überwachtem (supervised), unüberwachtem (unsupervised) und verstärkendem Lernen (reinforced learning) (Steinwendner & Schwaiger, 2020).

Unter überwachtem Lernen wird das Lernen anhand bekannter Daten verstanden, um daraus Muster und Zusammenhänge zu erkennen. Einem Datensatz wird neben dem Algorithmus auch der gewünschte Output bereitgestellt. Es wird immer in Zusammenhang mit einer Zielvariable gelernt, welche der ML-Algorithmus versucht, korrekt vorherzusagen (Sarker, 2021). Bei dem Lernverfahren der unüberwachten Methoden wird dem ML-Algorithmus nur ein Input-Datensatz zur Verfügung gestellt. Der Output ist dem Modell dabei nicht bekannt. Ziel dahinter ist es, aus dem vorhandenen Datensatz neue, bisher unbekannte Erkenntnisse zu Tage zu bringen. Der Unterschied zum überwachten Lernen besteht somit darin, dass die Zielvariable in diesem Setting nicht bekannt ist. Ein häufiges Anwendungsfeld stellen zum Beispiel Clusteranalysen dar (Sarker, 2021).

Verstärkendes Lernen verfolgt eine abweichende Lernhypothese. Bei dieser Lernmethode interagiert der Algorithmus mit der Umgebung und wird durch eine Kostenfunktion oder ein Belohnungssystem bewertet. Dies erfolgt mit dem Hintergedanken, dass der Algorithmus eine Strategie entwickeln soll, die Belohnung zu maximieren. Dem Algorithmus wird somit nicht gezeigt, welche Operation in der aktuellen Situation die richtige ist, sondern er erhält anhand einer Kostenfunktion eine entsprechende Rückmeldung. Es führt somit zu einem „bestärkendem“ Lernen durch „Lob“ oder „Bestrafung“ über die Belohnungs- bzw. Kostenfunktion. Weit verbreitete Machine Learning Methodiken sind die Support Vector Machine, die zur Klassifikation oder Regression eingesetzt werden kann. Andere Beispiele sind Entscheidungsbäume, Random Forrests, Gradient Boosting-Verfahren oder auch Künstliche-Neuronale-Netze (Chollet, 2021).

Deep Learning

Deep Learning stellt ein spezifisches Teilgebiet des maschinellen Lernens dar. Hinter „Deep“ versteckt sich dabei die Idee einer großen Anzahl von Repräsentationsebenen. Die Anzahl der existierenden Ebenen wird als Tiefe bezeichnet. Moderne Deep Learning Anwendungen verfügen dabei oftmals über mehrere hundert Ebenen von Repräsentation zur Analyse von Datensätzen (Chollet, 2021). Die Anlernmethodik bei Deep Learning

Modellen erfolgt über Künstliche-Neuronale-Netze. Der Begriff neuronale Netze kommt dabei aus der Neurobiologie. Zum besseren Verständnis der Funktionsweise wird daher einleitend auf das biologische Vorbild, die neuronalen Netze im menschlichen Organismus, eingegangen. Als Grundbaustein des Nervensystems fungieren die Nervenzellen, auch Neuronen genannt (Acig, 2001).

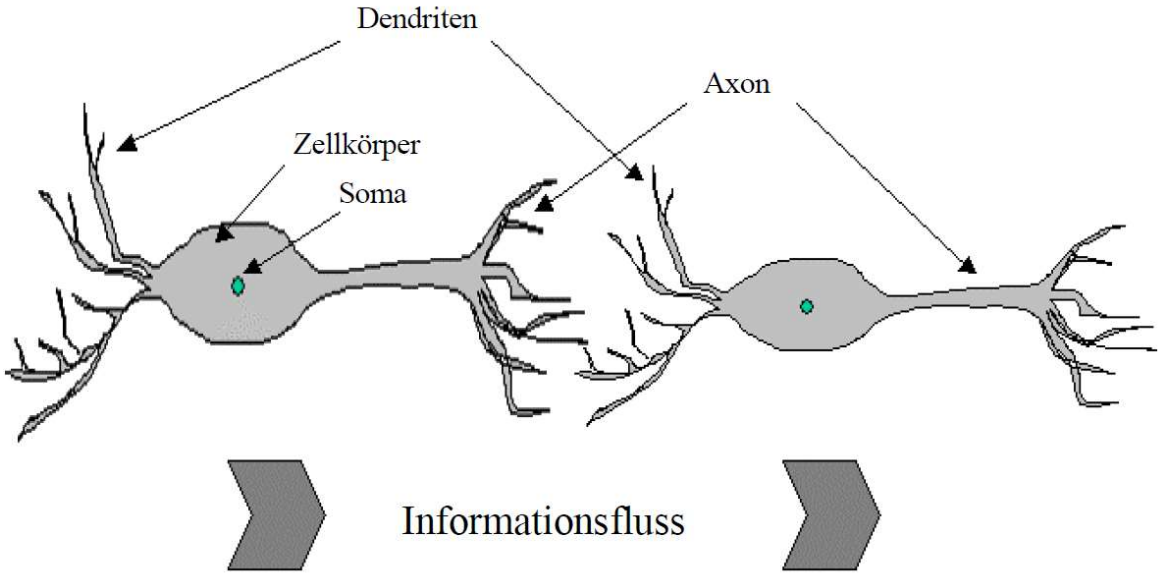


Abbildung 2: Darstellung Neuron (Acig, 2001)

Ein Neuron selbst, besteht aus drei Bestandteilen – dem Zellkörper, mehreren Dendriten und einem Axon. Im Zellkörper befindet sich der Zellkern (Soma), der das genetische Material der Zelle enthält. Die Dendriten sind feinverzweigte Ausläufer des Zellkörpers, die der Signalaufnahme dienen. Das Axon wiederum stellt die Leitungsbahn dar, die für die Informationsweitergabe verantwortlich ist. Als Synapse bezeichnet man die Stelle an der das Axon, über die Axonterminale, eines Zellkörpers mit den Dendriten eines anderen Zellkörpers in Kontakt tritt. (Acig, 2001). Diese Synapsen nehmen daher eine wesentliche Funktion des neuronalen Netzes ein. Über sie werden Signale in Form von elektrischen Ladungen in eine chemische Substanz umgewandelt (Grotian & Beelich, 1999). Diese chemische Substanz, als Transmitter bezeichnet, wird durch elektrisches Potenzial, dass an der Synapse gemessen werden kann an das nachfolgende Membran transportiert. Die auf diesem Weg ankommenden Signale werden summiert und in Form von elektrischer Ladung an den Zellkern weitergeleitet. Ist im Kern ein bestimmter Schwellwert erreicht, wird das empfangende Neuron selbst aktiv und gibt Signale über sein Axon weiter. Durch das elektrische Potenzial der Synapse erfolgt somit eine Gewichtung und damit

weiterführend eine Bewertung der Eingangsinformation (Füser, 1995). Diese können im biologischen Kontext erregend oder hemmend wirken. Synapsen sind damit Informationsträger und -bewerter gleichzeitig, die ihre Methodik zeitlich verändern können und somit eine Form des „Lernens“ repräsentieren (Alex, 1998).

Angelehnt am biologischen Vorbild bestehen Künstliche-Neuronale-Netze aus mehreren Neuronen. Das künstliche Neuron ist dabei dem biologischen Vorbild nachempfunden. Als Dendriten können beim künstlichen Neuron die Kanäle zur Aufnahme der Inputwerte angesehen werden. Die Dendriten geben die erhaltenen Informationen über eine Schnittstelle – die Synapse – weiter. Die Synapse nimmt dabei wie beim biologischen Vorbild die gewichtende Rolle ein. Die Inputparameter werden daher um Gewichte ergänzt. Über eine Rechenoperation werden die Inputinformationen mit den Gewichten verbunden. Anschließend erfolgt über eine Aktivierungsfunktion im Zellkern die Entscheidung, ob die Information (im Vergleich zur Biologie das Signal) über das Axon als Output weitergegeben wird oder nicht (Füser, 1995). Dabei spielt sie eine maßgebliche Rolle in der Funktionsweise des Neurons, da anhand ihrer festgelegt wird, ob und welche Informationsweitergabe an die weiteren Neuronen des Netzes erfolgt. In der Abbildung 3 unterhalb sind vier häufige Typen dargestellt (Tetzner et al., 2021).

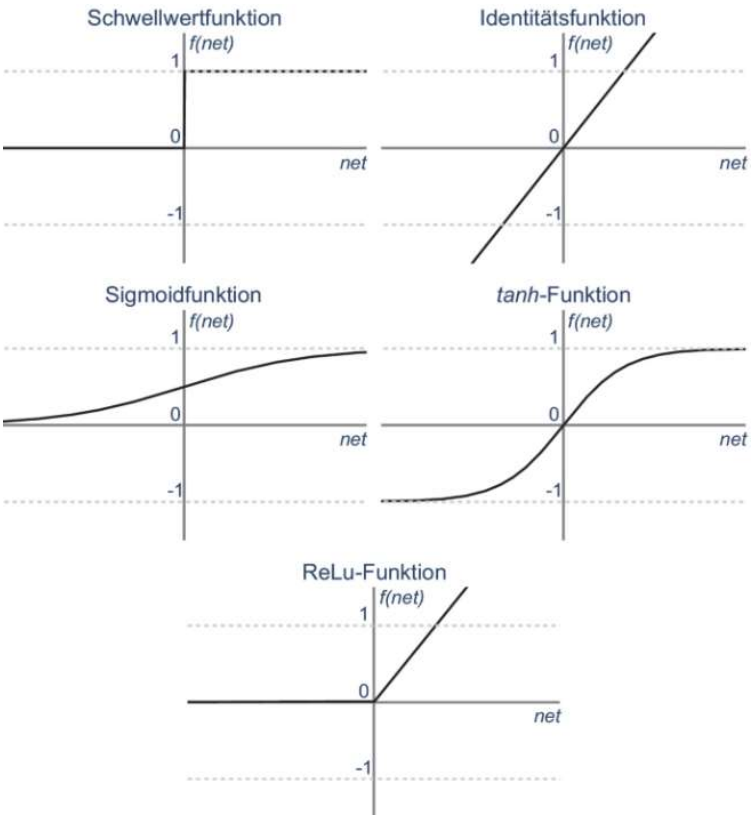


Abbildung 3: Häufig genutzte Aktivierungsfunktionen (Tetzner et al., 2021)

Ein Künstliches-Neuronales-Netz (KNN) verfügt darauf aufbauend über unterschiedliche Anzahlen von Neuronen und damit verbunden Repräsentationsebenen (engl. Layer). Jedes neuronale Netz verfügt hierbei allerdings zumindest über die drei unterhalb beschriebenen Ebenen (Chollet, 2021).

Über das Input-Layer (oder Eingabeschicht genannt) gelangen die Informationen in das Künstliche-Neuronale-Netz. Von dort aus gelangen sie in eine weitere Ebene, das erste Hidden-Layer. In diesem Hidden-Layer findet eine Form der Verarbeitung statt. Das Verarbeitungsergebnis wird an die nächste Ebene, die Output-Ebene, bestehend aus den Output Neuronen, weitergegeben (Berning, 2016). Systemtheoretisch werden neuronale Netze daher als offene, dynamische Systeme bezeichnet. Dies kommt daher, dass die Verbindungen und deren Gewichte, somit die Topologien innerhalb des Netzes nicht starr sind, sondern Veränderungen unterliegen und jedes Neuron in seine Umwelt integriert ist. Im Falle von „Deep-Neural-Networks" existieren dieser Systematik folgend eine hohe Anzahl von Hidden-Layers (versteckten Ebenen) die für die Verarbeitung der Eingabeinformationen zuständig sind. Aufgrund dieser Struktur werden Künstliche-Neuronale-Netze oftmals auch als informationsverarbeitende Black-Boxen bezeichnet (Füser, 1995).

Topologien neuronaler Netze

Grundsätzlich wird zwischen zwei Topologien Künstlicher-Neuronaler-Netze entschieden, die wiederum über eine Reihe von Subtypen verfügen. Als Differenzierungsmerkmale dieser Subtypen können die Lernart, Architektur, Informationsfluss, Berechnungsart und der Datenverarbeitungsmodus genannt werden. In der Tabelle unterhalb wird eine grobe Übersicht über bekannte Typen Künstlicher-Neuronaler-Netze dargestellt (Tetzner et al., 2021).

Tabelle 2: Überblicksauszug bekannte Typen Künstlicher-Neuronaler Netze

Typen Künstlicher-Neuronaler-Netze	
Feed-Forward-Netze	Feedback-Netze
Perzeptron	Boltzmann-Maschine
Multi-Layer Perzepton	Hopfield-Netze
Adaline	Cognition / Neocognition
Madaline	Recurrent Perceptron
Convolutional Neural Networks	Long Short Term Memory Networks
Counterpropagation Netze	

Im folgenden Unterkapitel wird nur auf die grundsätzlichen topologischen Unterschiede von KNNs eingegangen.

Vorwärtsgerichtete, rückkopplungsfreie Netze (Feed-Forward-Neural-Networks)

Unter Feed-Forward-Netzen werden Künstliche-Neuronale-Netze verstanden, in denen die die Informationen von einer Ebene in die nächste Ebene des Netzes gelangen (von Layer zu Layer in eine Richtung). Mit jedem Weitergabeschritt erfolgt eine neue Repräsentation der Daten, die näher an den gewünschten Output führen soll. Im Falle von hierarchischen vorwärtsgerichteten Netzen wird dabei jede einzelne Ebene von Input- über Hidden- bis zur Outputebene durchlaufen. Allgemeine Feed-Forward-Netze verfügen über die Funktion einzelne Ebenen zu überspringen. Dies wird im englischen auch als „shortcut connection" bezeichnet (Sarker, 2021).

Eine weit verbreitete Form eines mehrschichten künstlichen Feed-Forward-Netzes ist das Convolutional Neuroal Network (CNN). Jenes wird vor allem in der Verarbeitung von Bild- und Audiodateien eingesetzt. Sie zählen dabei zur Klasse der Deep-Neural-Networks. CNNs setzen sich aus zwei Abschnitten zusammen – dem Kodierungsblock und dem Prädiktionsblock. Der Kodierungsblock besteht selbst nochmals aus drei Schichten – der Faltungsschicht (Convolutional-Layer), der Aktivierungsschicht (Activation-Layer) und der Poolingschicht (Pooling-Layer). Anhand dieser Schichten erfolgt die Extraktion der Bildeigenschaften. Als Verarbeitungsergebnis dieses Kodierungsblocks steht ein vollständig kodiertes Bild zur Verfügung, welches darauf aufbauend im Prädiktionsblock anhand der Kodierung aufgelöst und die Elemente des Bildes klassifiziert werden. Dafür werden in der Regel Fully Connected Networks eingesetzt. (Steinwendner & Schwaiger, 2023). Bei sogenannten Fully Connected Networks sind alle Neuronen einer Ebene mit allen Neuronen der nächsten Ebene verknüpft. Diese Form der Netze werden auch als Hopfield-Netze bezeichnet (Acig, 2001)

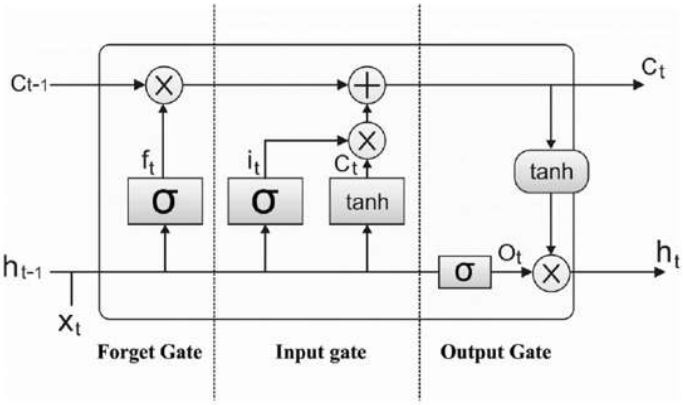
Anwendungsfälle vorwärts-gerichteter neuronaler Netze liegen neben der Bildverarbeitung bei CNNs vor allem in Themenfeldern der Klassifikation und Regression. So ist es beispielhaft anhand neuronaler Netze möglich Risikopatienten in unterschiedliche Klassen einzuteilen, um so abzuwägen, welche priorisiert auf die Intensivstation verlegt werden sollten. Aus dem Feld der Regression könnte ein Beispiel etwa die Ermittlung der Verspätung eines bestimmten Fluges anhand der historischen Flugdaten sein (Pan, 2024). Auf wissenschaftliche Anwendungsfälle im Finanzbereich wird im Kapitel 3.2 näher eingegangen.

Netze mit Rückkopplung (Recurrant neural Networks (RNNs))

Als Gegenstück zu Feed Forward Netzen, die nur in eine Richtung durchlaufen werden, verfügen Recurrent-Neural-Networks, auch als Feedback-Netze bezeichnet, über Mechanismen der Informationsrückkopplung (Sarker, 2021). Deren Topologieeinteilung kann in drei Übergruppen erfolgen. Netze mit direkter Rückkopplung verfügen über Neuronen, die direkt mit sich selbst verbunden sind und ihr Signal damit selbst aktivieren oder hemmen können (Direct-Feedback-Netze). Netze mit indirekter Rückkopplung verfügen über Rückkopplungsmechanismen zwischen einzelnen Ebenen. (Indirect-Feedback-Netze). Netze mit möglichen Rückkopplungen innerhalb einer Ebene werden als Lateral-Feedback-Netze bezeichnet. Diese Netze werden dann eingesetzt, wenn die Aufgabenstellung es erfordert, dass nur das stärkste Neuron aktiv werden soll (Acig, 2001).

Eine Spezialform des RNNs stellt das 1998 entwickelte Modell der Long Short Term Memory (LSTM)-Netze dar. Das Problem das Feedback-Netze aufweisen ist, dass sie zwar aufgrund der Rückkopplungslogik über eine Form des „Kurzzeitgedächtnisses“ verfügen, jenes aber bei der Abarbeitung längerer Sequenzen stark abnimmt. Um diesem Umstand entgegenzuwirken wurden LSTM-Netze entwickelt, um vergangene Informationen zu bewerten und dadurch relevante Informationen länger vorhalten zu können. Jenes hat gegenüber herkömmlichen rekurrenten neuronalen Netzen der Vorteil, dass innerhalb eines Neurons zusätzlich zum Output des vorhergegangenen Verarbeitungsschrittes, ein interner Status existiert, der ebenfalls in die Verarbeitung innerhalb des Neurons einfließt (Hochreiter & Schmidhuber, 1997). Dieser interne Status besteht dabei aus drei Gates – dem Forget-, Input- und Output-Gate. Der Forget-Gate entscheidet dabei, welche Information aus dem internen Status nicht mehr benötigt wird und „vergessen“ werden kann. Darin enthalten sind der Hidden Status aus dem vorherigen Durchlauf, sowie der aktuelle Input. Über eine Aktivierungsfunktion wird entschieden, ob die vorherige Information vergessen werden kann oder nicht. Die daraus entstandenen Ergebnisse werden mit dem aktuellen Zellenstatus multipliziert. Der Input-Gate entscheidet weiterführend, welche neue Information im internen Status aufgenommen werden soll, um die Aufgabe zu lösen. Dazu wird der aktuelle Input mit dem Hidden State und der Weight Matrix des letzten Durchlaufs multipliziert. Alle Informationen, die im Input Gate als wichtig erscheinen, werden dann mit dem Zellenstatus addiert und bilden dadurch den neuen Zellenstatus. Der Output Gate legt abschließend fest, welche konkrete Information im gegenständlichen Verarbeitungsschritt ausgegeben werden soll. Dabei entscheidet eine

Aktivierungsfunktion, welche Informationen durch das Output Gate gelangen (Swathi et al., 2022).



$$i_t = \sigma(W_{ix}x_t + W_{\mathfrak{I}}m_{t-1} + b_i)$$
$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + b_o)$$
$$f_t = 1 - i_t$$
$$tc_t = g(W_{cx}x_t + W_{cm}m_{t-1} + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot tc_t$$
$$m_t = o_t \odot hc_t$$
$$y_t = \phi(W_{ym}m_t + b_y)$$

Abbildung 4: LSTM-Struktur & Berechnung (Swathi et al., 2022)

i,o und *f* stellen dabei die Input-, Output- und Forget-Gates dar. t_c ist der Dateninput des Neurons, *c* repräsentiert die Neuronenaktivierungsvektoren und *m* die Daten im Memory-Output des Neurons. *W* stellt weiterführend die Weight-Matrizen dar (beispielhaft W_{ix} die Gewichtsmatrix für Input *x* zum Inputgate *i*). *b* definiert das Bias (b_i repräsentiert den Input-Gate Bias-Vektor), welches bei starkem Gewicht sicherstellt, dass die aktuelle Einheit aktiv bleibt und umgekehrt. *g* und *h* stellen die Aktivierungsfunktionen von Neuroneninput und -output dar (Swathi et al., 2022).

Anwendungsfälle von LSTMs sind etwa die Tastaturvervollständig von Apple oder die AlphaGo-Software von Google, die darauf trainiert ist, Menschen im Go-Spiel zu schlagen. Generell werden sie eingesetzt, um Muster in Datensequenzen zu erkennen wie beispielsweise Sensordaten oder Aktienkursen (Nelson et al., 2017).

Trainieren von neuronalen Netzen

Im vorangegangenen Kapitel wurden bereits die unterschiedlichen Lernformen neuronaler Netze beschrieben. Die Aussagekraft eines Vorhersagemodells auf Basis von Künstlichen-Neuronalen-Netzen (KNN) hängt maßgebend von der Qualität der zugrunde liegenden Daten und dem Training des KNNs ab. Dabei werden die zur Verfügung stehenden Daten in zwei Blöcke geteilt – in Trainings- und Testdaten (Nelson et al., 2017). Anhand der Trainingsdaten wird das neuronale Netz angelernt. Ein Teil der Trainingsdaten wird für das tatsächliche Lernen des Modells eingesetzt, ein weiterer Teil -der Validierungsdatensatz- wird zur Validierung des Modells genutzt. Mithilfe des

anhand der Trainingsdaten trainierten Modells, werden darauf aufbauend für die Testdaten die Outputwerte prognostiziert (Berning, 2016).

Ein weit verbreiteter und auch hier zur Anwendung gelangender Algorithmus zum Training maschineller Lernformen wie neuronalen Netzen ist der Gradientenabstieg (Lanham, 2018). Es handelt sich dabei um ein Optimierungsverfahren, bei dem Eingangssignale das Künstliche-Neuronale-Netz vollständig durchlaufen, um einen Fehlerwert zu berechnen. Die Anpassung der Netzparameter (Gewicht und Bias) findet dabei proportional zur partiellen Ableitung der Fehlerfunktion statt. Das beschriebene Vorgehen wird dabei in dem Ausmaß wiederholt, bis eine fest definierte Anzahl an Epochen erreicht ist, oder die Funktion einen Threshold erreicht, um den kleinstmöglichen Fehler zu erzielen (Vogt, 2021). Eine Problematik die beim Einsatz des Gradientabstiegs auftreten kann, ist das sogenannte „Vanishing Gradient Problem" (Burkov, 2019). Dabei kommt es in mehrschichtigen neuronalen Netzen nur mehr zu minimalen Anpassungen der Parameter in den zuvor liegenden Layern. Besondere Vorsicht ist zudem in der Konstruktion des neuronalen Netzes zu nehmen, um Overfitting zu vermeiden. Unter Overfitting wird im gegenständlichen Kontext die Spezialisierung des Prognoseprototypen auf die Trainingsdaten verstanden. Dies hat zur Folge, dass für den Trainingsdatensatz eine hohe Modellgüte erreicht wird, die sich bei Anwendung des Prototypenmodells auf den Prognosedatensatz jedoch nicht bestätigt. (Lechelt, 1998).

3 Prognosemodelle mittels neuer Medien und Künstlicher-Neuronaler-Netze

Versuche Aktienkurse zu prognostizieren sind nicht neu. Daher ist es nicht verwunderlich, dass bereits eine Vielzahl von Studien zum Einsatz Künstlicher-Neuronaler-Netze für Prognosezwecke von Aktienkursen vorliegen. Insbesondere deren Anwendbarkeit zur Identifizierung von nicht-linearen Zusammenhängen stellt dabei einen Mehrwert dar (Kubiak, 1991). In den nachfolgenden Abschnitten wird eine Beleuchtung bisheriger Erkenntnisse und Studien zu angestellten Prognosemodellen auf Basis neuer Medien durchgeführt. Dabei erfolgt im ersten Unterkapitel ein Überblick über die Bedeutung neuer Medien für den Kapitalmarkt, sowie deren Wechselwirkung. Anschließend wird die vorteilsbehaftete Hinzunahme ebendieser Informationen, insbesondere aus Online-Fachzeitschriften wie der New York Times oder Sentimentinformationen, extrahiert von Social Media Plattformen, für die Finanzmarktprognose erörtert. Im zweiten Unterkapitel wird auf bisherige Kapitalmarktprognosen anhand Künstlicher-Neuronaler-Netze eingegangen. Es wird dargelegt, weshalb der Einsatz eines Long Short Term Memory-Neuronalen-Netzes zur Prognose von Aktienschlusskursen oder Aktienindizes als adäquat zu erachten ist.

Dieses Kapitel bietet einen Überblick über einen ausgewählten Stand der Wissenschaft und eine Erläuterung des Einstiegspunkt für die gegenständliche Untersuchung.

3.1 Bedeutung neuer Medien für Kapitalmarktprognosen

Die Korrelation des Zeitverlaufs von Handelsvolumen am Kapitalmarkt, sowie der zeitliche Verlauf der Nachrichtenmenge gilt als nachgewiesen (Engelberg & Parsons, 2011) (Da et al., 2011). Abbildung 5 unterhalb zeigt dies für das Handelsvolumen (ausgedrückt in der Anzahl der gehandelten Aktien pro Tag) der Toyota Aktie und dem Verlauf des Nachrichtenvolumens, gemessen in Wörtern pro Tag in Zeitungsartikeln, die den Namen „Toyota“ beinhalten. Eine Studie, die anstatt der Wortanzahl in Zeitungsartikeln nur die Anzahl der Artikel nutzte, kam zum selben Ergebnis (Hisano et al., 2013).

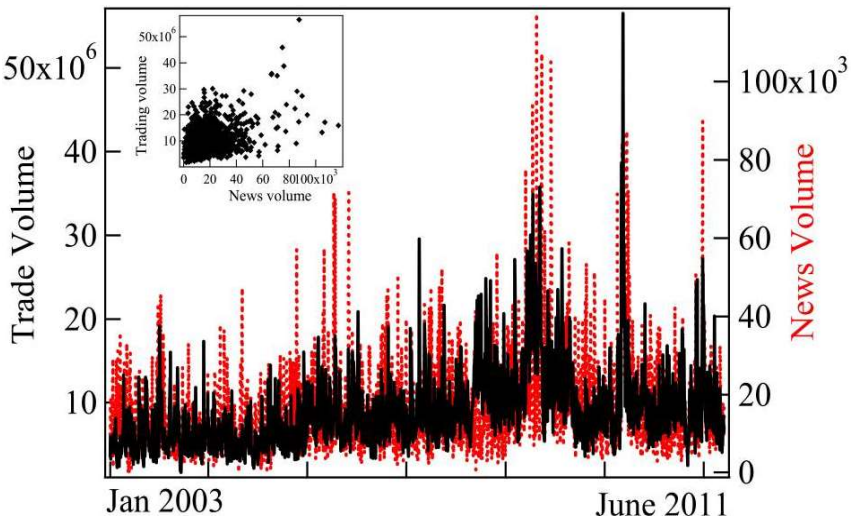


Abbildung 5: Vergleich des täglichen Handelsvolumens und dem Nachrichtenaufkommen des Unternehmens Toyota (Hisano et al., 2013)

Eine andere Untersuchung aus dem Jahr 2001 zeigte, dass es im Durchschnitt zu 20 Minuten Zeitverzögerung kommt, bis eine neu veröffentlichte Nachricht sich quantifizierbar auf den Kapitalmarkt auswirkt (Gidofalvi, 2001).

Im Jahr 2016 wurde veröffentlicht, dass der Einfluss von Social Media auf den Aktienmarkt „signifikant größer" als jener der herkömmlicher Medienkanäle ist (Jiao & Walther, 2016). So bestehen laut Brown et al. Korrelationen zwischen zeitgleichen Stimmungen und Aktienrenditen (Brown et al., 2004).

Neue Medien, allen voran das Internet und insbesondere Soziale Medien, sind herkömmlichen Medien in Hinblick auf Aktualität und Informationsdichte überlegen. Wie etwa im Kapitel zum Dow Jones ausgeführt erhielten Investoren ihre Informationen lange Zeit aus klassischen Medien. Durch die DotCom-Revolution wurde es den Investoren ermöglicht, als Nutzer im Internet selbst Inhalte zu erfassen und Nachrichten auszutauschen. Insbesondere Foren zum Austausch über Themen der Börse erfreuen sich besonderer Beliebtheit (Wu, D et al., 2014) (Piryani et al., 2017). Diese Entwicklungen machen es auch für die Schaffung neuer Prognoseansätze interessant, auf neue Medien als Datenbasis zurückzugreifen. Bereits 2003 wurde von Watkins nachgewiesen, dass Korrelationen zwischen den Foreninhalten und ausgewählten Aktienmärkten existieren (Watkins, 2003). Betrachtet man die Informationseinholung der breiten Anlegerschaft gilt dieses Narrativ bis heute. Einen elementaren Teil bei der Entscheidungsbildung nehmen in Zeiten der „App Culture" dabei die sozialen Medien und unweigerlich damit verknüpft Meinungen einzelner Individuen als Teil einer Masse ein. Die Literatur zeigt, dass insbesondere bei der Investitionsentscheidung der Emotionalität eine große Rolle

zugeordnet werden muss (Dowling & Lucey, 2003). Im Finanzbereich ist es allen voran die kombinierte Userschaft von Reddit und Twitter, die es Kleinanlegern, bei Ressourcenbündelung erlaubt, einen entsprechenden Leverage Effekt zu erzielen. Ein Beispiel stellt hierbei der GME-Short Squeeze aus dem Jahr 2021 dar (Anand & Pathak, 2021). Eine Studie hat gezeigt, dass „Twitter-Daten den Dow Jones Industrial Average mit einer Genauigkeit von 87,6% vorhersagen" (Burnette, 2021). Diese Vorhersagegenauigkeit demonstriert die Marktmacht, welche Privatanlegern über Social-Media-Kanäle zu Teil wird. Im Jahr 2016 wurde veröffentlicht, dass der Einfluss von Social Media auf den Aktienmarkt „signifikant größer" als jener der herkömmlicher Medienkanäle ist (Jiao & Walther, 2016). Dabei ist das Herdenverhalten der Anleger im Hintergrund entscheidend. Der Begriff des Herdenverhaltens stammt aus der Tierwelt. Im Falle von Unsicherheit oder etwa bei der Nahrungssuche, wird durch Bildung eines Kollektivs aus einzelnen Individuen, die Sicherheit für die gesamte Gruppe erhöht. In der Menschenwelt tritt dieses Verhalten etwa im Rahmen von Modeerscheinungen, Demonstrationen oder vor allem auch am Kapitalmarkt auf. Menschen neigen insbesondere in für sie unsicher erscheinenden Entscheidungssituationen dazu, nicht rational auf eigene Erfahrungsschätze zurückzugreifen, sondern sich an der Masse zu orientieren und „mitzuschwimmen" (Fabio, 2015). Ein weiterer fördernder Faktor des Herdenverhaltens ist der ‚sharing-the-blame-effect', der besagt, dass bei der Annahme von Entscheidungen anderer, die Verantwortung einer Fehlentscheidung mit allen Entscheidungsträgern geteilt wird. Die Fehlentscheidung der Masse stellt dabei eine emotional akzeptablere Situation als das alleinige Verfehlen dar (Fabio, 2015). Dieses Verhaltensmuster wurde von John Maynard Keynes treffend festgehalten: *„Worldly wisdom teaches that it is better for reputation to fail conventionally than to succeed unconventionally".*

In Zusammenhang mit der COVID-19-Pandemie und dem Wegfall des persönlichen Face-to-Face-Austauschs durch Ausgangsbeschränkungen, fand ein noch stärkerer Drift der Informationsverteilung und -einholung, sowie eine Verstärkung dieses Phänomens über Social Media Kanäle statt (Yarovaya, 2021). Aber nicht nur soziale Medien auch das Internet als allgemeine Informationsquelle und Recherchetool kann erheblichen Einfluss auf die Aktienkursentwicklungen herbeiführen. Eine Studie aus dem Jahr 2021 hat gezeigt, dass ein statistischer Zusammenhang zwischen Google-Suchanfragen nach bestimmten Aktien und deren Kurs nachweisbar ist (Vasileiou et al., 2021). Wissenschaftliche Untersuchungen belegen weiterführend auch die Berechtigung des auf diesem Weg ausgeführten Investitions- und Informationseinholungsansatzes. Nofer et al.

fanden im Jahr 2014 heraus, dass private Investoren in Internetforen gewinnmaximierendere Informationen erhalten, als durch ihren persönlichen Bankberater (Nofer & Hinz, 2014). Diese Erkenntnis geht auf das Phänomen der kollektiven Intelligenz zurück, welches im gegenständlichen Kontext besagt, dass die Kaufempfehlung einer Gruppe gewinnbringender ist als die Kaufempfehlung eines einzelnen Experten. Jenes Narrativ gilt auch dann, wenn die Gruppe nur aus Amateuren besteht (Schröter, 2019.). Eine im Journal of Computational Sciences im Jahr 2010 veröffentlichte Studie von Johan Bollen, Mao Huina und Zeng Xiao-Jun unter dem Titel „Twitter mood predicts the stock market" hat gezeigt, dass insbesondere Tweets den Aktienkurs bestimmter Unternehmen beeinflussen können. Ausgehend von der Erkenntnis, dass Emotionen, das Entscheidungsbewusstsein beeinflussen, wurde anhand der Tools Opinion Finder und Google-Profile of Mood States (GPOMS), welches Stimmungen in sechs Kategorien untergliedert untersucht, wie Tweets positiver und negativer Polarität mit dem Dow Jones Industrial Average Index korrelieren. Dabei wurde die faszinierende Erkenntnis erzielt, dass es anhand dem von Ihnen entwickelten Modell möglich war, den Anstieg oder Abfall des DJIA-Schlusskurses mit einer Genauigkeit von 87,6% vorherzusagen (Bollen et al., 2011). Oliviera et al. zeigten anhand einer Analyse im Jahr 2013, dass eine erhöhte Anzahl von Tweets gleichzeitig mit einem erhöhten Handelsvolumen und einer einhergehenden höheren Volatilität der Börsenkurse auftritt (Oliveira et al., 2013). Weiterführend weisen soziale Medien eine effizientere Vorhersagekraft als Onlineausgaben von klassischen Medien auf. Dies geht auf die stärkere Korrelation dieser Medienform mit den Aktienkurses zurück (Y. Yu et al., 2013). Studien belegen zudem den Zusammenhang zwischen der Social Media-Strategie und der Ertragslage von ausgewählten Unternehmen (Schniederjans et al., 2013). Dass Finanzentscheidungen anhand von Stimmungen getroffen werden, wurde bereits von Verhaltensökonomen bewiesen (Nofsinger, 2005). Gilbert und Karahalios bestätigten dies auch für den Bereich neuer Medien. Dabei wurde anhand der Granger Kausalität und einer Monte Carlo-Simulation nachgewiesen, dass Stimmungen einzelner Post auf der Website LiveJournal mit den Aktienkursen nicht nur korrelieren, sondern sich darauf auswirken können. Ihre Untersuchung anhand von 20 Millionen Posts hat gezeigt, dass sich insbesondere negative Stimmung in Blogposts, auf den Kapitalmarkt, in der betroffenen Studie auf den Aktienindex S&P500, auswirken kann (Gilbert & Karahalios, 2010). Nach Erkenntnissen von Ranco et al existiert zudem ein Zusammenhang zwischen der Social Media Stimmung und außergewöhnlichen Renditen. In deren Untersuchung wurde ebenfalls auf Basis der Granger-Kausalität herausgefunden, dass eine hohe Anzahl von Postings mit starken Schwankungen der Renditen korreliert (Ranco et al., 2015).

Weitere Studien extrahierten Frühindikatoren aus sozialen Netzwerken, um Aktienkurse vorherzusagen. Die Stimmung in Tweets stellte dabei einen Indikator dieses Typs dar, welcher zu einer Kauf- oder Verkaufsentscheidung rät (Bollen et al., 2011). Ein anderes Einsatzbeispiel von Sentimentinformationen, gewonnen aus Social Media Postings, ist der Hochfrequenzhandel. Dabei wurden Texte auf einen eingeschränkten Kontext unter Zuhilfenahme von genetischen Algorithmen untersucht und zur Prognose eingesetzt (Vincent & Armstrong, 2010). Andere Untersuchungen beschäftigen sich wiederum mit der Entwicklung von Prognosemodellen anhand der historischen Aktienkurse und Tweets für die FENG-Unternehmen (Vogt, 2021). Shilpa entwickelte ein Prognoseframework basierend auf Sentiment Analyse und vorhandenen Vorlaufindikatoren des Kapitalmarktes (Shilpa, 2023). Weiterführend wurde bereits untersucht, wie sich der Aktienmarkt bei plötzlich auftretenden Stimmungsausbrüchen verhält. Zhang et al konnte einen signifikanten Zusammenhang zwischen Tweets die Hoffnung, Angst oder Sorge verbreitet hatten und dem Kursrückgang des Dow Jones nachweisen (Zhang et al., 2011). Die oberhalb angeführten Beispiele unterstreichen die Relevanz und Möglichkeiten, welche neuen Medien, insbesondere Social Media Plattformen für die Prognose von Aktienkursentwicklungen zu Teil werden. Aufgrund dessen wird das Zurückgreifen auf neue Medien zur Aktienindexprognose als probat erachtet.

Im nachfolgenden Kapitel wird auf die Prognosemöglichkeiten anhand neuronaler Netze näher eingegangen.

3.2 Prognosen mittels neuronaler Netze

Im Rahmen einer angestellten Sekundäranalyse bietet die Arbeit von Torsten Lechelt an der Wirtschaftswissenschaftlichen Fakultät Martin-Luther-Universität Halle-Wittenberg aus dem Jahr 1998 einen Überblick über das breite Themenfeld der „Aktienkursprognosen auf Basis neuronale Netzwerke“ im vergangenen Jahrhundert. Dabei wird nicht nur auf die Grundlagen, Konzeption, Schwierigkeiten und Umsetzungsmöglichkeiten neuronaler Netze zur Aktienkursprognose eingegangen, es wird auch ein klarer Überblick zu historischen Untersuchungen von Aktienkursprognosen anhand neuronaler Netze gegeben (Lechelt, 1998).

Die Eignung rekurrenter neuronaler Netze, insbesondere des auch hier eingesetzten LSTM-Typs, für die Prognose von Aktienkursbewegungen wurde von Nelson et al. Im Jahr 2017 untersucht (Nelson et al., 2017). Dabei wurde die Performance des LSTM-Modells in Bezug auf deren Genauigkeit mit anderen maschinellen Lernverfahren verglichen. Der

Vergleich fand mit Multilayer-Perceptron, Pseudo-Zufall- und Random-Forest-Modellen statt. Dabei konnte beobachtet werden, dass das LSTM-Modell die anderen Modelle in Bezug auf die Ausübung gängiger Handelsstrategien wie „Buy and Hold“ übertrifft, obwohl 180 Inputparameter ohne Dimensionsreduktion eingesetzt wurden. Eine weitere Studie, welche die Eignung von LSTM-Modellen für die Aktienkursprognose untermauert ist jene von Achkar et al aus dem Jahr 2018 (Achkar et al., 2018). Im Rahmen dieser Arbeit wurde versucht, den Schlusskurs der Aktien von Google, Facebook und Bitcoin anhand des Einsatzes eines Multilayer-Perceptrons (MLP) und eines LSTM vorherzusagen. Das Autorenteam stellte dabei zudem die Vorhersage der nächsten Kurstage aus beiden Modellen und die tatsächliche Entwicklung für die Facebook Aktie gegenüber. Diese Gegenüberstellung zeigte, dass die Abweichungen der Schlusskursprognose bei dem MLP-Modell zwischen 3 und 16%, bei dem LSTM-Modell zwischen 0,5 und 12% lag, was auf die Eignung des LSTM als Prognosewerkzeug von Aktienschlusskursen schließen lässt. Im Jahr 2020 wurde in einer Studie die Effektivität der Nutzung von Tweets im Rahmen der Aktienkursprognose untersucht. Dabei wurde die Genauigkeit der Prognose unter der Anwendung von Künstlichen-Neuronalen-Netzen gegenüber traditionelleren Machine Learning Verfahren geprüft. Diese Untersuchung von Kolasani und Assaf baut dabei auf die Studie von Chakraborty et al aus dem Jahr 2017 auf, die eine Kursvorhersage von Apple und dem Dow Jones Index inspizierte (Chakraborty et al., 2017). Im Rahmen beider Studien wurden unterschiedliche Modelle wie die Support Vector Machine, Decision Tree, Random Forrests und KNNs eingesetzt, um Sentimentdaten in Machine Learning Modellen zur Aktienprognose miteinzubeziehen. Als Ergebnis weist die Studie von Chakraborty et al die Support Vector Machine mit einer Genauigkeit von 83% als effizienteste Variante zur Prognose aus. Kolasani und Assaf fanden darauf aufbauend heraus, dass das KNN (MLP) durchschnittlich noch bessere Ergebnisse bei der Preisdifferenz der Aktien erzielt als die Support Vector Machine. Die Autoren bemerkten zudem, dass im gewählten Untersuchungsmodus, das KNN eher pessimistischere Vorhersagen der Preisdifferenz traf, als andere maschinelle Lernverfahren (Kolasani & Assaf, 2020). Ein andere von Lin et al. im Jahr 2020 durchgeführte Studie zeigte anhand des Einsatzes von rekurrenten neuronalen Netzen, dass es möglich ist, den Öffnungskurs, den Schlusskurs und die Differenz zwischen beiden Tageskursen für den S&P500 und den Dow Jones Aktienindex mit neuronalen Netzen effizienter vorherzusagen als mit herkömmlichen Machine-learning oder Korrelations- bzw. Kausalitätsansätzen (Shen & Shafiq, 2020). Auch ältere Untersuchungen wie etwa von Lee und Park zeigten bereits in den 1990er Jahren, dass mit Ansätzen rekurrenter neuronaler Netze, trainiert anhand von Daten von elf Finanzmarktindikatoren, gute Prognoseergebnisse erzielt werden

können (Lee & Park, 1992). Shah et al untersuchten auf Basis der Erfolge von neuronalen Netzen im Bereich der Finanzmarktprognosen, welche Arten von Künstlichen-Neuronalen-Netzen am besten geeignet sein könnten. In Ihrem Ansatz wurde der indische BSE Sensex Index anhand eines LSTM und einem DNN (Deep-Neural-Network) untersucht. Ziel war es die tägliche und wöchentliche Index Veränderungen vorherzusagen. Bei beiden Netzen wurden Maßnahmen zur Reduktion von Overfitting getroffen (Jabbar & Khan, 2014). Die anhand der Untersuchung entwickelten Prognoseergebnisse wurden danach an der Aktie des Unternehmens Tech Mahindra auf Generalisierbarkeit getestet. Beide Modelle und Formen von KNNs erwiesen sich als adäquat in Bezug auf die tägliche Prognose der Aktienschlusskurse. Das LSTM jedoch performte in Bezug auf die wöchentlichen Kursentwicklungen besser als das DNN. (Shah et al., 2018). Eine ähnlich aufgebaute, von Althelaya et al., 2018 durchgeführte, Untersuchung, kommt zu vergleichbaren Ergebnissen. Im Rahmen dieser Studie zeigte sich, dass ein LSTM im Vergleich mit anderen Formen von neuronalen Netzen in der Kurz- und Langristprognose des S&P500 Aktienindizes bessere Vorhersageergebnisse liefert (Althelaya et al., 2018).

Eine weitere Studie, durchgeführt von Rokhsatyazdi et al., 2020, widmete sich der Frage ob der Einsatz von Long Short Term Memory-Netzen in der heutigen Zeit, die eine Verarbeitung von unzähligen Einflussfaktoren weltweit benötigt, besser geeignet sind, als State-of-the-Art-Verfahren wie NAIVE, ETS oder SARIMA. Dabei wurde deren Performance mit einem auf Basis des Differential Evolution (DE) optimierten LSTM mit dem Ziel der Vorhersage des Aktienkurses am Folgetag verglichen. Es zeigte sich, dass das von ihnen entwickelte LSTM erzielte einen geringeren RMSE als die herkömmlichen statischen Verfahren, was deren Eignung für die Aktienkursprognose unterstreicht (Rokhsatyazdi et al., 2020). Im Jahr 2019 wurden von Torres et al die Börsendaten von Apple Inc. unter Hinzunahme von Machine Learning Algorithmen untersucht, um Supervised Learning Verfahren im Finanzbereich zu bewerten. Dabei nutzten die beiden Autoren ein Künstliches-Neuronales-Netz und einen Decision Tree. Sie beobachteten die historischen Aktiendaten wie Eröffnungskurs, Schlusskurs, Tageshöchst- und -tiefststände, sowie das Handelsvolumen der Aktie. Auf Basis dieser Informationen wurde versucht, die jeweiligen Aktienschlusskurse zu prognostizieren. Als Ergebnis der Untersuchung kommen die Autoren zu dem Schluss, dass sich beide Methoden zur Vorhersage von historischen Finanzmarktdaten eignen, eine Sentimentanalyse die Effektivität der Vorhersagen womöglich jedoch noch weiter verbessern könnte (Torres et al., 2019). Swathi et al erstellten 2022 auf Basis von Twitter Sentimentanalysedaten und

historischen Aktienkursinformationen ein LSTM-Modell basierend auf dem „Teaching and Learning“ Optimierungsansatz, welches Prognoseergebnisse für die täglichen Aktienkurse mit einer Vorhersagegenauigkeit von über 90% lieferte (Swathi et al., 2022). Die am nächsten mit der gegenständlichen Arbeit verwandte Untersuchung ist eine Arbeit der TU Wien aus dem Jahr 2021, welche analysiert, ob anhand von historischen Aktienkursen, kombiniert mit Social-Media-Sentimentdaten und Sentimentanalyse von Zeitungsartikeln der New York Times der Aktienkurs der Unternehmen Apple, Tesla und Biontech SE vorhergesagt werden kann (Wiesinger, 2021). Während der darin gewählte Ansatz zur Sentimentanalyse der Twitterdaten zielführend erscheint, wird die gewählte Systematik zur Einbindung der Nachrichteninformationen der New York Times nur anhand einer Sentimentanalyse als nicht adäquat angesehen, da hierbei davon auszugehen ist, dass Fachartikel objektiv und neutral formuliert sind, weshalb die daraus extrahierten Polarität der Texte keinen bedeutsamen Mehrwert für das Prognosemodell darstellen (Wiesinger, 2021). Hier wäre es zielführender, auf eine qualitative Inhaltsanalyse zur Extraktion geeigneter Informationen aus der Fachzeitschrift New York Times oder anderen Publikationen zurückzugreifen. An diesem Punkt steigt die gegenständliche Arbeit zur Aktienkursprognose des Dow Jones Aktienindex ein.

4 Konzeptionelle Entwicklung des Prognosemodells

Anhand der zuvor exemplarisch angeführten Studien wurde dargelegt, dass statistische Zusammenhänge zwischen neuen Medien und Aktienkursentwicklungen ausgewählter Unternehmen existieren. Diese Korrelationen können in gewissen Ausprägungen kausale Zusammenhänge darstellen, weshalb sie für die Nutzung zur Prognose von Finanzmarktassets im Sinne der angenommenen Hypothese geeignet erscheinen. Die Hypothese lautet dabei, dass es durch Verknüpfung historischer Aktienkursinformationen des Dow Jones Industrial Average, ergänzt um Sentimentinformationen aus Submissions und Kommentaren von Reddit, zu ebendiesem Aktienindex, sowie aus Ergebnissen, gewonnen durch qualitative Inhaltsanalyse, aus Artikeln der Fachmagazine New York Times und Wall Street Journal möglich ist, ein Prognosemodell auf Basis neuronaler Netze zu entwickeln, welches die Meinungen von Kleinanlegern und Investitionsexperten, sowie historischen Aktienkursinformationen verbindet, um zumindest temporär eine Vorhersagegenauigkeit von 60% zu erreichen. Jene Vorhersagegenauigkeit bezieht sich dabei auf den möglichen Schlusskursanstieg oder -abfall am nächsten Markttag.

Im folgenden Kapitel wird dazu beginnend geschildert, wie die Datenerhebung und -aufbereitung für die Inputfaktoren des Künstlichen-Neuronalen-Netzes erfolgte. Dabei wird der Export der Submissions und Kommentare der Internetplattform Reddit aus dem Subreddit /r/Wallstreetbets, die Einholung der facheinschlägigen Zeitungsartikel der Onlineauftritte von Wall Street Journal und New York Times, sowie deren Datenanalyse und -aufbereitung für das KNN beschrieben. Ebenfalls ausgeführt wird der Download und die Aufbereitung der Finanzmarktdaten. Anschließend wird im Detail auf die Konzeption und das Training des LSTM zur Dow Jones Industrial Average Index-Prognose eingegangen.

4.1 Datenerhebung

Im nachfolgenden Abschnitt werden die Schritte der Datenerhebung, aufgeteilt in die drei Blöcke Finanzmarktdaten, Kleinanlegerdaten (Reddit) und Finanzmarktnachrichten, beschrieben. Es werden jeweils die Herangehensweise, sowie die eingesetzten Mittel angeführt.

4.1.1 Finanzmarktdaten

Die in der gegenständlichen Untersuchung verwendeten Finanzmarktdaten stammten vom deutschen Online Finanzportal www.ariva.de. Ein Export der historischen Kursdaten ist hier nach kostenloser Registrierung möglich. Für den Dow Jones Industrial Average Index (weiterführend auch als DJIA bezeichnet) wurden für den Betrachtungszeitraum von 01.01.2023 bis 31.12.2023 die folgenden Informationen tagesfein in tabellarischer (als .csv) und grafischer Form extrahiert (Ariva, 2024.)

Tabelle 3: www.ariva.de - Export Finanzmarktdaten Inhalte

Datum	Startkurs	Höchstkurs	Tiefstkurs	Schlusskurs	Volumen

An Tagen an jenen die Börsen geschlossen waren und daher kein Handel stattfand (Wochenenden und Feiertagen) wurden für den gegenständlichen Prototypen zur Aktienindexvorhersage folgende Parameter festgelegt:

Aktienindexschlusskurs: An handelsfreien Tagen wird als Schlusskurs der letztgültige Kurs des Vortags herangezogen. Sollte am Vortag die Börse ebenfalls geschlossen gewesen sein, so wird der letzte Tag mit gültigem Schlusskurs angenommen.

Gewinn/Verlust/Volumen: An handelsfreien Tagen wird der eventuelle Gewinn respektive der Verlust und das Handelsvolumen einer Unternehmensaktie mit 0 festgelegt.

Diese getroffenen Modellparameter sind für ein adäquates Training des LSTM-Modells erforderlich, falls an handelsfreien Tagen Finanzmarktnachrichten oder Tweets zum DJIA veröffentlicht wurden.

4.1.2 Kleinanlegermeinungen – Plattform Reddit

Nachfolgend werden die einzelnen Arbeitsschritte zum automatisierten Download ausgewählter Inhalte von der Internetplattform www.reddit.com beschrieben.

Automatisierter Zugriff auf www.reddit.com

Die Internetplattform www.reddit.com stellt eine eigene Schnittstelle für den automatisierten Datenabgriff bereit. Um jene nutzen zu können, ist es erforderlich ein Benutzerkonto anzulegen, welches es einem erlaubt, auf die Inhalte der Webseite automatisiert zuzugreifen. Jene können zwar auch ohne Account abgerufen werden, um erweiterte Funktionen wie Schnittstellen zu eigenen IT-Anwendungen freizuschalten, ist ein Benutzerkonto jedoch unumgänglich. Nach erfolgreicher Benutzeranlage kann unter

dem Teilbereich */Preferences/Apps* der Plattform eine eigene App erzeugt werden. Diese App muss angelegt werden um später automatisiert, über das zu erstellende Python-Skript, auf die Inhalte der Plattform zugreifen zu können. Im Hintergrund wird auf der Plattform eine neue Client-Instanz mit eigenem Zugangsschlüssel (Secret) generiert, über die dann später die Master-Datenbank (die gesammelten Webseiteninhalte) eingesehen werden kann[1].

Export der Kommentare

Nach erfolgreicher Verbindung mit der Plattform Reddit, können im weiteren Schritt nun die eigentlichen Aktionen zum Kommentarexport gesetzt werden. Nachfolgend wird der grundlegende Aufbau der Online-Plattform überblicksmäßig skizziert. www.reddit.com (weiterführend auch als Reddit bezeichnet) wird dabei in unterschiedliche Subreddits gegliedert. Jene Subreddits widmen sich einem thematisch abgegrenzten Gebiet. Im aktuellen Fall beschäftigt sich der ausgewählte Subreddit „Wallstreetbets“ mit allen erdenklichen Themen zur Finanzwirtschaft. Diese Subreddits werden weiter in einzelne Threads (auch Submissions genannt) eingeteilt. Submissions können wiederum ebenfalls in kleinere Einheiten, sogenannte Subthreads, unterteilt sein. Die Zuvor genannten Einheiten, beginnend ab einer Submission, können von Usern kommentiert werden. Um alle zugehörigen Kommentare zu gewählten Schlagworten exportieren zu können, muss anhand iterativer Vorgehensweisen, die gesamte Kommentarbaumstruktur, beginnend ab der Submission durchlaufen und analysiert werden. Je Subreddit werden die einzelnen Submissions von der Plattform automatisiert diversen Kategorien zugeordnet. Dabei wird unterschieden zwischen „hot“ – aktuell „heiße“ Nachrichten und Informationen, „new“ – neueste, „rising“- stark und schnell an Interaktion zunehmende Posts (Threads), „top“ – die meistgelikten Threads, oder „controversial“- Posts mit vielen Up- aber auch vielen Downvotes zugleich[2]. Long et al kamen in einer im Jahr 2021 durchgeführten Studie zu der Erkenntnis, dass im gegenständlichen Subreddit „wallstreetbets“, lange Threads mit einer hohen Kommentaranzahl und breiter Reichweite das generelle Userverhalten beeinflussen können, während bei kürzeren Threads keine Auswirkungen detektiert werden konnten (Long et al., 2021). Um in der gegenständlichen Studie alle relevanten Informationen zu extrahieren, werden neben dem Bereich „hot“ jedoch auch die weiteren obengenannten Bereiche des Subreddits ausgewertet. Es ist nach den Erkenntnissen von Yu et al.

[1] Reddit. Reddit Einstellungen. Abgerufen 30. März 2024, von https://www.reddit.com/prefs/apps
[2] Wallstreetbets. Abgerufen 30. März 2024, von https://www.reddit.com/r/wallstreetbets/

allerdings davon auszugehen, dass aus der „hot"-Kategorie die meisten relevanten Kommentare abgegriffen werden, da es sich bei diesem Bereich um die „Frontpage", die am meisten interagierte Seite innerhalb des Subreddits handelt. (Yu et al., 2013).
Bis Mitte 2023 war es möglich über die plattformeigene Reddit API und einer Vielzahl von darauf aufbauender Dritt-APIs wie etwa PRAW, die in Einklang mit den Coding-Richtlinien der Plattform Reddit entwickelt wurden, kostenfrei beliebig viele Submissions zu extrahieren[3] (Selvi & Arulchelvan, 2024). Dabei existiert eine Limitierung von 1000 Submissions pro Request innerhalb eines kurzen Zeitraums. Mittels rekursiven Vorgehensmodellen war es darauf aufbauend möglich, beliebig viele Submissions zu extrahieren. 1000 Submissions entsprechen im gegenständlichen Subreddit rund drei Kalendertagen an Inhalten (Long et al., 2021). Ähnlich wie Twitter hat aber auch die Plattform Reddit im vergangenen Jahr eine neue Pricing Policy mit einer Bepreisung von Requests an das eigene API eingeführt. So müssen, Stand Juni 2023 0,24 US-Dollar pro 1000 bzw. 12.000 US-Doller für 50 Mio. Requests bezahlt werden. Aus diesem Grund wurde im Rahmen der gegenständlichen Untersuchung eine alternative Extraktionsquelle herangezogen – Pushshift. Die Pushshift Library stellt eine unabhängige Datenbank dar, welche in Ihren Speicherbeständen sämtliche Informationen von www.reddit.com darstellt. Jene werden auch weiterhin laufend mit der Plattform synchronisiert[4]. Neben dem Datenbestand selbst, stellt Pushshift eine eigene API – pushshift.io Reddit API – zur Verfügung, über welche die darin enthaltenen Daten abgefragt werden können (Baumgartner et al., 2020). Aufgrund der Änderungen der Bezahlungsmodalitäten von Reddit selbst steht diese jedoch ebenfalls nur mehr ausgewählten Benutzern zur Verfügung. Für die breite Masse werden die Daten jedoch als hochkomprimierte zstandard-compressed-json Archive bereitgestellt, die vor allem für akademische Zwecke über torrents kostenfrei bezogen werden können (Baumgartner, 2024). Im gegenständlichen Fall wurden die archivierten Submissions des Subreddits /r/Wallstreetbets auf diesem Weg bezogen. Neben den Submissions wurden zudem alle Kommentare als eigenes Archiv bezogen und für die weitere Untersuchung genutzt. Ein Vorteil der Pushshift-Library besteht darin, dass die extrahierten Daten anhand der eindeutigen Submission-IDs auch über die benutzerfreundliche PRAW-Library oder über JSON-Handling weiterverarbeitet werden können, worüber sämtliche Funktionen der „hauseigenen" Reddit-API zur Verfügung stehen[5]. Für den Umgang mit zstandard komprimierten Daten existiert ebenfalls eine

[3] PRAW 7.7.1 documentation. Abgerufen 30. März 2024, von https://praw.readthedocs.io/en/stable/
[4,3] Pushshift API. GitHub—Pushshift/api: Abgerufen 30. März 2024, von https://github.com/pushshift/api

nutzbare Python library (zstandard), die im gegenständlichen Fall eingesetzt wurde[6]. Um nicht alle Submissions und Kommentare zu jedem diskutierten Thema im Subreddit „Wallstreetbets“ aus dem Archiv zu extrahieren, wurden bestimmte Suchbegriffe definiert und nur Threads und Kommentare mit Bezug zu diesen Begriffen für die weitere Untersuchung herangezogen. Dabei wurde sobald über die Submission oder ein Top-Level-Kommentar die Zugehörigkeit zu einem der Suchbegriffe gegeben war, nicht mehr untersucht, ob in einem niedriger geordneten Kommentar selbst, einer der gewünschten Suchbegriffe enthalten war, da in menschlicher Kommunikation zu einem Thema, das Thema selbst nicht laufend fällt (Röhner & Schütz, 2020). Sämtliche Textinformationen wurden zudem auf lower-case-character formatiert. Als Suchbegriffe wurden „dow jones“ und „djia“ eingesetzt. Im ersten Schritt wurden daher im Rahmen der Arbeit aus dem Submissions-Archiv die Submissions des Untersuchungszeitraum von 01.01.2023 00:00 bis zum 31.12.2023 23:59 extrahiert, die die entsprechenden Suchbegriffe enthielten. Je ermittelter Submission wurden die in Tabelle 4 dargestellten Informationen in ein .csv-File gespeichert.

Tabelle 4: Aufbau Submission .csv

Submission ID	**Datum**	**Link**	**User**	**Inhalt**
64129 3356	2023-04-27	https://www.reddit.com/r/wallstreetbets/comments/130g2hh/daily_discussion_thread_for_april_27_2023/jhy7plv/	27,u/Ready2gambleboomer	DJIA currently up 420 even my weed bags are getting lighter. Truly a monster day.

Über die Submission-IDs war es im darauffolgenden Schritt möglich aus dem Comments-Archiv die einzelnen, den Submissions zugehörigen Kommentare zu extrahieren. Kommentare selbst sind in eine Baumstruktur untergliedert und müssen nach Zuordnung zur korrekten Submission-ID iterativ vom First-Level-Kommentar über Second- und Third-Level bis zu einer beliebig tief werdenden Struktur durchlaufen werden, um daraus alle zugehörigen Kommentare zu exportieren (siehe beispielhaft Abbildung 5).

[6] Zstandard: bindings for Python (0.22.0). [C, Python]. Abgerufen 30. März 2024, von https://github.com/indygreg/python-zstandard

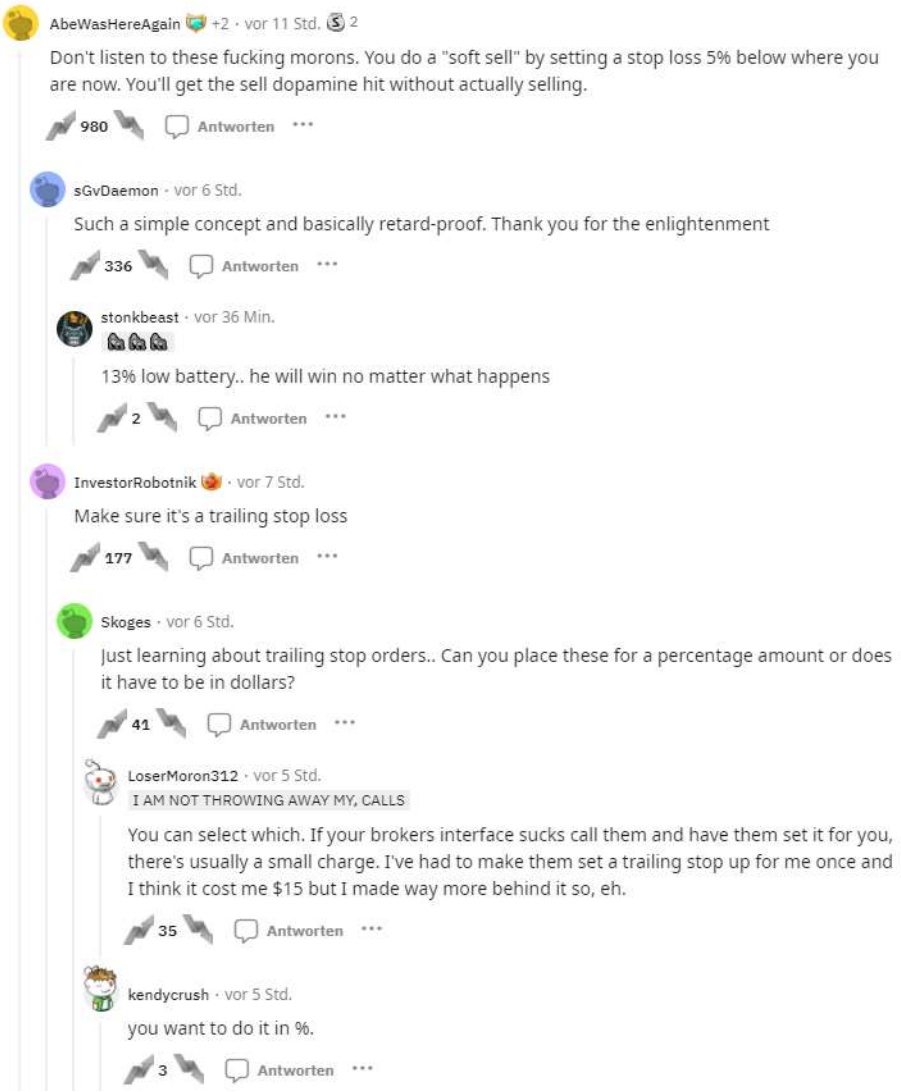


Abbildung 5: Reddit Kommentarstruktur[7]

Je Kommentar wurden die identen Informationen wie für die Submissions extrahiert (Submission-ID, Datum, Link, User, Inhalt). Im letzten Schritt wurden die Submissions mit den Kommentaren verbunden und daraus drei .csv-Dateien generiert, die jeweils rund ein Jahresdrittel an mit den Suchbegriffen korrelierenden Kommentaren enthielten. Auf diesem Weg wurden für die gegenständliche Studie in Summe 130.697 Submissions und 7.900.595 Kommentare herangezogen. Die weitere Analyse und Aufbereitung dieser Daten werden in Kapitel 4.2.2 näher beschrieben.

[7] Reddit, 2023.“ I was about to check on my stocks I bought in margin with my life savings and see this shit in the mail”. In */r/Wallstreetbets*. Abgerufen am 11. Dezember 2023 https://www.reddit.com/r/wallstreetbets/comments/rdsulq/i_was_about_to_check_on_my_stocks_i_bought_in/

4.1.3 Extraktion Finanzmarktnachrichten

Im nachfolgenden Abschnitt wird die Extraktion der online abrufbaren Finanzmarktnachrichten, der beiden Fachmagazine New York Times (weiterführend auch als NYT bezeichnet) und Wall Street Journal (WSJ) jeweils in einem eigenen Abschnitt, beginnend mit dem WSJ erläutert.

Wall Street Journal

Da das Wall Street Journal keine eigene API für den Datenabgriff bereitstellt, wurde für den Export der Finanzmarktnachrichten in Relation zum Dow Jones Industrial Average auf Webscraping Verfahren zurückgegriffen. Im Rahmen dieser Untersuchung wird unter Webscraping die automatisierte Extraktion von Informationen zu Zeitungsartikeln aus dem Webauftritt des Wall Street Journal verstanden (Pal et al., 2021). Dabei wurde die Python Library Selenium genutzt, die eine umfangreiche Methodenpalette für Webscraping-Aktivitäten bereitstellt. Selenium ist eine Python Library, die Funktionen für Web Browser Automation beinhaltet, weshalb sie in der Python Community für Webscraping Verfahren eingesetzt wird[8] (García et al., 2020). Der Nachrichtenexport erfolgte dabei in drei Phasen. Im ersten Schritt wurden mittels Selenium und der Wall-Street-Journal-eigenen Suchmaske inklusive Filtermöglichkeit alle Artikel aufgelistet, welche in Überschrift oder Textkörper die Suchbegriffe „Dow Jones“, „Dow Industrials“ oder „DJIA“ enthalten. Die WSJ-Suche ist dabei case-insensitive. Über die so generierte Such-URL wurden weiterführend die Links aller Artikel inklusive Headline mittels Selenium und der pandas-Library automatisiert in eine .csv-Datei geschrieben. Pandas wurde im Zuge dessen für das Handling großer Datenmengen mittels DataFrames eingesetzt[9] (McKinney, 2011).

Tabelle 5: Aufbau WSJ-Link-CSV

Index	Headline	Link

Im zweiten Schritt wurde mittels Browser Automation über Aufruf der Artikel-Links der jeweilige Nachrichteninhalt abgerufen. Dabei wurde unter Zuhilfenahme der Libraries Selenium und librecaptacha ein automatisiertes Cookie- und Captcha-Handling

[8] WebDriver. Selenium. Abgerufen 8. März 2024, von https://www.selenium.dev/documentation/webdriver/

[9] Pandas. User Guide—Pandas 2.2.1 documentation. Abgerufen 8. März 2024, von https://pandas.pydata.org/docs/user_guide/index.html

implementiert (Motoyama et al., 2023)[10]. Die für den Anwendungsfall relevanten Datenfelder wurden dabei über die XPATH-Referenzen aus dem Quelltext der Webseite ausgelesen. Auf diesem Weg wurden die Inhalte von 1295 Artikeln im Kalenderjahr 2023 exportiert und in einer .csv-Datei nach Datum und Nachrichteninhalt getrennt gespeichert. Im Nachrichteninhalt wurden Headline, Abstrakt und Volltext des jeweiligen Artikels zusammen gesichert. Problematisch war hierbei, dass über die Suche nicht nur nach dem gesamten Suchbegriff „dow jones" sondern nach auch nach Teilbegriffen, im aktuellen Beispiel auch nach „dow" und „jones" gesucht wurde, weshalb im Rahmen der Kontrolle der Nachrichteninhalte eine zusätzliche Bereinigungsschleife zur Datenaufbereitung erforderlich wurde. Im dritten Schritt wurden daher teilautomatisiert mittels Python-Skript und händischer Kontrolle all jene Artikel ausgefiltert, die zwar textuell Teile der Suchparameter enthielten jedoch inhaltlich nicht dem gegenständlichen Gebiet zugehörig waren. Beispielhaft das Gerichtsverfahren von Alex Jones in den USA 2023. Durch diese Kontrollschleife wurde die Gesamtzahl der später weiterverarbeiteten Nachrichtenartikel des Wall Street Journal auf 475 Stück reduziert. Auffallend waren hierbei, dass insbesondere Nachrichten zur Inflationsbekämpfung in den Vereinigten Staaten von Amerika über die FED zumindest wöchentlich erschienen und darin immer Auswirkungen und Prognosen zum Dow Jones Industrial Average enthalten waren. Auch das tägliche Börsenupdate des Wall Street Journals enthielt verlässlich Informationen zum Dow Jones. Um die oberhalb gelisteten Schritte durchführen zu können war ein Abonnement des Wall Street Journals erforderlich, da ohne Abo nicht auf die Inhalte der Online Plattform zugegriffen werden kann[11].

New York Times

Im Gegensatz zum Wall Street Journal stellt die New York Times eine Reihe von Python-APIs kostenfrei zur Verfügung, sofern jene nicht für kommerzielle Zwecke eingesetzt werden. Die bereitgestellten APIs sind jedoch mit 500 Requests pro Tag bzw. 5 Requests pro Minute limitiert[12]. Aufgrund des Betrachtungszeitraums von einem Jahr wurde daher nicht auf den Einsatz dieser APIs zurückgegriffen, sondern ein täglich aktualisierter Datensatz zu den New York Times Artikeln ab dem Jahr 2000 von der Data Science Plattform Kaggle.com genutzt[13] ((Tauchert et al., 2020). Dieser Datensatz enthielt mit

[10] librecaptcha: A free/libre interface for solving reCAPTCHA challenges. (0.7.3). [Python; OS Independent]. Abgerufen 8. März 2024, von https://github.com/taylordotfish/librecaptcha
[11] The Wall Street Journal. Abgerufen 30. März 2024, von https://store.wsj.com/shop/emea/wsjaoemea24/
[12] Dev Portal New York Times. FAQ Dev Portal. Abgerufen 8. März 2024, von https://developer.nytimes.com/faq#a11
[13] Kaggle NYT Articles Dataset. Abgerufen 8. März 2024, von https://www.kaggle.com/datasets/aryansingh0909/nyt-articles-21m-2000-present

Stand Februar 2024 rund 2.1 Mio Artikel der New York Times. Je Artikel sind im Datensatz Informationen zu Abstract, URL, Druckbereich, Druckseite (bei Printausgabe), Zusammenfassung des Artikels, Schlagworte, Kategorie, Datum, Wortanzahl und weitere Details enthalten (weitere Inhalte können den Quellen im Literaturverzeichnis entnommen werden). Neben diversen Zusammenfassungen in unterschiedlichen Längen enthält dieser Datensatz größenbedingt jedoch nicht die gesamten ursprünglichen Nachrichtenartikel. Da der Datensatz im .csv-Format mit einer Größe von rund 4.5GB nur als Gesamtes vorliegt, wurde jener im ersten Schritt der Datenverarbeitung auf das Kalenderjahr 2023 reduziert. Dies erfolgte durch Einsatz eines CSV-Editors. Weiterführend wurde er mittels Python-Skript nach den identischen Suchbegriffen wie das Wall Street Journal durchsucht („dow jones“, „dow industrials“ und „djia“). Im Rahmen dieser Analyse zeigte sich, dass nur ein geringer Anteil der Nachrichtenartikel aufgrund der bereits erfolgten Zusammenfassung der Artikel tatsächlich einen der gesuchten Begriffe enthielt, weshalb weiterführend die Suche auf Artikel ausgeweitet wurde, die der Kategorie „Stocks and Bonds“ zugeordnet wurden. Für alle Nachrichten, die diesen Gesichtspunkten zugeordnet werden konnten, wurde ein .csv-File angefertigt, in welchem Datum, URL und alle Formen der Zusammenfassungen zu jedem Artikel gespeichert wurden. Es erfolgte abschließend eine manuelle Prüfung der Qualität der Zusammenfassungen anhand der Originalartikel. Auf diesem Weg wurden 173 Artikel für die spätere Weiterverarbeitung im LSTM zur Aktienindexprognose extrahiert. Zur Verarbeitung des Datensatzes in Python wurden die Libraries pandas und csv eingesetzt (McKinney, 2011).

4.2 Datenaufbereitung

Um präzise Prognoseergebnisse anhand des LSTM zu erhalten, ist es unumgänglich die im vorangegangenen Abschnitt beschriebenen Finanzmarktdaten, Social Media-Daten, sowie Finanzmarktnachrichten in geeignete Form zu bringen und zu analysieren. An der Systematik des Kapitels Datenerhebung orientiert, gliedert sich das folgende Kapitel in die Abschnitte 4.2.1 Finanzmarktdaten, 4.2.2 Sentimentanalyse Reddit Daten, 4.2.3 Qualitative Inhaltsanalyse Finanzmarktnachrichten (LDA) und 4.2.4 Datenzusammenführung.

4.2.1 Finanzmarktdaten

Aus den gewonnenen Finanzmarktdaten wurden im weiteren Verlauf der Analyse gemäß der Forschungsfrage nur das Handelsdatum, sowie der Tagesschlusskurs benötigt. Für

das spätere Prototyping Verfahren wurden zudem jedoch auch der der Tagesstartkurs, der Tageshöchst-, der Tagestiefstkurs, das Handelsvolumen, sowie die Differenz aus Start- und Schlusskurs vorbereitet. Zusätzlich wurde anhand der vorliegenden Daten der Tagesverlust und -gewinn anhand der Differenz bzw. Summe aus dem täglichen Start- und Schlusskurs (unterhalb bezeichnet als „*Tages Gewinn Verlust*“), sowie der Verlust und Gewinn bezogen auf den Vortag aus Differenz bzw. Summe zwischen den Schlusskursen aus aktuellem Tag und Vortag ermittelt („*Vortag Gewinn Verlust*“). Dabei wird ein Gewinn mit positivem Vorzeichen, ein Verlust mit negativem Vorzeichen ausgewiesen. Unter dem Wert „*Vortag Gewinn“* verbirgt sich der Schlusskursgewinn bezogen auf den Vortag. Liegt kein Gewinn vor, so beträgt dieser Wert 0. Der umgekehrte Fall gilt für den Wert „*Vortag Verlust*“ im Falle eines Rückgangs des Schlusskurses. Alle weiteren Parameter wurden für das Prognosemodell nicht berücksichtigt. Aus den oberhalb gelisteten Parametern wurde sodann eine Input-CSV Datei für das LSTM generiert. Dies wird deshalb umgesetzt, um im späteren Prototyping Verfahren, die Kombination unterschiedlicher Eingabevariablen zu testen und darauf aufbauend das Prognosemodell weiter zu verbessern. Initial wird hierbei jedoch nur auf das Datum und den Aktientagesschlusskurs zurückgegriffen. Je Handelstag stehen somit die folgenden Informationen zur Verfügung:

Datum | *Erster* | *Hoch* | *Tief* | *Schlusskurs* | *Volumen* | *Tages Gewinn Verlust* | *Intraday Gewinn* | *Intraday Verlust* | *Vortag Gewinn Verlust* | *Vortag Gewinn* | *Vortag Verlust*

Ein Auszug der Finanzmarktdaten kann dem Anhang A entnommen werden.

Im Rahmen der Datenaufbereitung zur Weiterverarbeitung im LSTM mussten die Daten von der Finanzmarktplattform www.ariva.de bereinigt werden. So war es erforderlich, das Trennzeichen für Dezimalzahlen von „Komma“ auf „Punkt“ für die Python Weiterverarbeitung zu adaptieren. Für das spätere Verbinden der Finanzmarktdaten mit den weiteren Inputinformationen über das Datum als Index war es zudem notwendig, das das Datum umzuformatieren, um es in Einklang mit den anderen Eingabevariablen zu bringen.

Die oberhalb beschriebenen Berechnungs- und Aufbereitungsschritte wurden in Python anhand der NumPy Bibliothek realisiert. Bei NumPy handelt es sich um eine Bibliothek, die dem Anwender umfassende mathematische Berechnungsmöglichkeiten für Datenfelder (Arrays) zur Verfügung stellt. Mithilfe von NumPy können Operationen wie

das Wurzelziehen bis hin zu einfachen Anwendungen der deskriptiven Statistik durchgeführt werden[14] (Van Der Walt et al., 2011).

4.2.2 Sentimentanalyse Reddit Daten

Im Rahmen des Kommentarexports von der Plattform www.reddit.com, beschrieben unter Punkt 4.1.2, wurden sämtliche Kommentare aller Submissions im Subreddit /r/Wallstreetbets, im Untersuchungszeitraum von 01.01.2023 00:00:01 Uhr bis 31.12.2023 23:59:59 Uhr, auf die Suchbegriffe „dow jones“ und „djia“ geprüft und im .csv-Format gespeichert. Auf diesem Weg wurden 7.9 Mio Kommentare für die Weiterverarbeitung identifiziert und drei .csv-Datensätze zu je ca. 600 MB generiert. Jede dieser Dateien bildet in etwa ein Jahresdrittel ab. In weiterer Folge wurden diese Kommentare einer Sentimentanalyse unterzogen. Die Programmiersprache Python stellt hierbei eine breite Palette von Untersuchungsmöglichkeiten bereit. Darunter fallen etwa die Libraries NTLK (Natural Language Toolkit), die eine von der Stanford University entwickelte, breite Palette an Natural Language Processing Funktionen anbietet oder Textblob, eine Library zur Textanalyse, aufbauend auf NLTK[15] (M. Wang & Hu, 2021) (Hazarika et al., 2020). Im gegenständlichen Fall wurde für die Sentimentbeurteilung jedoch auf den lexikalischen Ansatz „Valence Aware Dictionary and sEntiment Reasoner (VADER)“ von Hutto und Gilbert zurückgegriffen (Hutto & Gilbert, 2014). Vor Einsatz des VADER-Verfahrens wurde der Text der einzelnen Kommentare anhand der NLTK-Library nach dem Saarbrückener Pipelinemodell aufbereitet, um etwaige Störfaktoren wie Emoticons oder Hyperlinks zu entfernen (M. Wang & Hu, 2021). Hutto & Gilbert entwickelten mit VADER ein Stimmungslexikon, das dezidiert für die Verwendung von Beiträgen aus sozialen Medien konzipiert wurde (Hutto & Gilbert, 2014). Bei VADER handelt es sich um ein rollenbasiertes Modell zur Sentimentanalyse, dass auf eine Kombination von qualitativen und quantitativen Methoden für die Beurteilung zurückgreift. Dabei wird anhand einer empirisch validierten “Gold Standard”-Liste von lexikalischen Eigenschaften, in Kombination mit der Sentimentintensität und einem definierten Regelsatz grammatikalischer und syntaktischer Konventionen für Ausdrucksweise und Betonung die Sentimentbestimmung durchgeführt. Hutto und Gilbert stellten dabei fest, dass sie mit Ihrem Modell bessere Ergebnisse als herkömmliche Techniken wie Naive

[14] NumPy Spezifikationen. Abgerufen am 23. Oktober 2023, von https://numpy.org/doc/stable/user/quickstart.html

[15] NLTK :: Natural Language Toolkit. Abgerufen 30. März 2024, von https://www.nltk.org/ ; TextBlob: Simplified Text Processing—TextBlob 0.18.0.post0 documentation. Abgerufen 30. März 2024, von https://textblob.readthedocs.io/en/dev/

Bayes oder Support Vector Machines erzielen. Besonders ist zudem, dass sie in Bezug auf Tweets sogar menschliche Beurteilung in einem ausgestellten Testsetting mit einer Klassifizierungsgenauigkeit von 96% übertreffen (Hutto & Gilbert, 2014). In Python gibt es von CJ Hutto selbst veröffentlicht, die entwickelte und bereitgestellte Library *vaderSentiment*, die vom Forschungsteam im Rahmen der Studie im Jahr 2014 entwickelt und seither laufend verbessert wird (Hutto, 2014/2024). Diese Library enthält neben den Funktionalitäten zur Sentimentanalyse auch Tools zur Identifikation von Emoticons. Aus dieser Bibliothek wurde für die aktuelle Untersuchung der Funktionsblock *SentimentIntensityAnalyzer* genutzt. Anhand dieser Funktionsgruppe können die Polarität und Subjektivität von Texten analysiert werden. Da gerade im Social Media-Kommentarbereich von hochsubjektiven Texten auszugehen ist, wurde im gegenständlichen Fall nur die Polarität für die Einstufung der Kommentare genutzt. Über die Funktion *polarity_scores* kann die Polarität eines Textes ermittelt werden. Dabei wird die Polarität in mehrere Segmente untergliedert: pos – die positive Ausprägung eines Texts, neg – die negative Ausprägung eines Texts, neu – die neutrale Ausprägung und compound ein normalisierter, gewichteter Wert, kombiniertet aus pos, neg und neu in Zusammenhang mit dem VADER-Lexikon (Hutto, 2014/2024). Hutto und Gilbert empfehlen für eine eindimensionale Analyse, wie im gegenständlichen Fall, den compound-Wert zu nutzen, um die Polarität (*py*) eines Kommentars zu messen. Der compund-Wert ist dabei zwischen -1 und 1 normalisiert und ist wie folgt zu interpretieren:

Positiv: $py > 0.5$

Neutral: $-0.5 < py < 0.5$

Negativ: $py < -0.5$ (Hutto, 2014/2024)

Unter Nutzung der vaderSentiment-Library wurde im Zuge der Datenaufbereitung für jedes Kommentare der compound-Polaritätswert ermittelt und gespeichert. Um die Gesamtstimmung jedes Tags einzufangen, wurden die Einzelpolaritätswerte der Kommentare eines jeden Tages summiert und durch die Anzahl der Kommentare an diesem Kalendertag dividiert. Auf diesem Weg konnte ein Gesamttages-Polaritätswert nach VADER je Kalendertag ermittelt werden. Dieser Gesamttageswert wurde dann gemeinsam mit dem Datum mithilfe der pandans-Library in einer .csv-Datei gespeichert. Je Kommentardatei wurde der obenstehende Vorgang durchgeführt und die mittels VADER-Lexikon ermittelten Sentimentinformationen, dann in einer Gesamtdatei zusammengeführt.

4.2.3 Qualitative Inhaltsanalyse Finanzmarktnachrichten (LDA)

Der dritte Inputblock für das neuronale Netz, als „digital abrufbare Expertenmeinungen", bezeichnet, wurde aus den Finanzmarktnachrichten von Wall Street Journal und New York Times extrahiert. Dahinter verbirgt sich eine qualitative Inhaltsanalyse den Kriterien nach Mayring folgend (Mayring, 2010). Da es sich hierbei um ein manuelles Vorgehen handelt, deren Anwendung für die gegenständlich vorliegende Datenmenge nicht geeignet ist, wurde das von Blei, Ng und Jordan an der University of Stanford entwickelte LDA „Latent Dirichlet Allocation"-Verfahren eingesetzt. Latent Dirichlet Allocation ist ein auf drei Ebenen aufgebautes bayeisisches Modell, in dem jedes Element eines Texts als Teil eines Themas (Topic) modelliert wird. Jedes Thema wiederum ist als Mischung von darunterliegenden Themen und Wahrscheinlichkeiten modelliert (topic probabilities). Im Kontext von Text Modelling und Topic Extraction, stellen die topic probabilities eine explizite Repräsentation eines Texts dar. In anderen Worten können mittels LDA aus einem Text die darin enthaltenen behandelten Themen extrahiert und ihnen eine entsprechende Gewichtung zugeordnet werden (Blei et al., 2003). Mittels LDA ist es somit möglich, Texte automatisiert qualitativ zu untersuchen. Wie in Kapitel 2.2.3 bereits ausgeführt konnten insbesondere im Bereich der Finanzmarktnachrichten mittels Einsatzes von LDA gute Resultate erzielt werden. So gelang es Garcia-Mendez et al 2023 mithilfe von Latent Dirichlet Allocation von Finanzmarktnachrichten darin erwähnte Marktereignisse zu identifizieren, prognostizieren und diesen Ereignissen Eintrittswahrscheinlichkeiten zuzuordnen. Im Zuge dieser Studie wurde zudem ermittelt, dass die automatisierte Kategorienbildung mittels LDA unter Anwendung von Data Clearing Schritten (vgl. Saarbrückener Pipelinemodell oder IMRAD) vergleichbare Ergebnisse wie die herkömmliche qualitative Inhaltsanalyse nach Mayring liefern kann (García-Méndez et al., 2023). Aufbauend auf die Erkenntnisse von Garcia-Mendez et al und Clark und Manning wurde für die Extraktion der Expertenmeinungen eine Latent Dirichlet Allocation in Python implementiert, die je Nachrichtenartikel, unabhängig ob von NYT oder WSJ, die zehn am stärksten gewichteten Themenblöcke bildet (Clark & Manning, 2016). Dafür wurde auf die gensim-Library zurückgegriffen. Hierbei handelt es sich um eine Bibliothek die speziell für Topic Modelling, Dokumentenindexierung und Vergleichsprüfungen zwischen Dokumenten entwickelt wurde. Speziell im Bereich der Computerlinguistik und der Informationsgewinnung ist der Einsatz der gensim-

Bibliothek weit verbreitet. Gensim selbst baut wiederum auf den Libraries NumPy und Scipy auf[16].

Initial wurden die nach Kapitel 4.1.3 aufbereiten Finanzmarktnachrichten als pandas-DataFrame eingelesen und zur Durchführung der LDA einer Zeichenkettenbereinigung unterzogen. Jene dient zur Entfernung von Sonderzeichen und weiteren Elementen, um die Texte für die LDA verarbeitbar zu machen. Die Python Library gensim stellt dafür die Funktion *preprocess_string* zur Verfügung, die die Data Clearing Schritte automatisiert abwickelt (Wang et al., 2023). Zur Durchführung der LDA mussten die so gewonnenen Textdaten in ein Dictionary gespeichert werden. Aus dem Text aller Zeitungsartikel je Fachzeitschrift wurde weiterführend für die LDA ein „bag of words" generiert, bei dem das Vorkommen der Wörter in den Texten berechnet wurde. Stopwörter wie etwa „ich" wurden bereits mittels der Funktion *„preprocess_string"* entfernt, um hier keine falschen Wortwahrscheinlichkeiten zu generieren (Blei et al., 2003) (Clark & Manning, 2016).

Mithilfe dieses bag of words wurde im nächsten Schritt für jeden Zeitungsartikel eine Latent Dirichlet Allocaiton zur Ermittlung der top zehn Themen durchgeführt. Die LDA selbst wurde mit der Funktion *gensim.models.ldamodel.LdaModel* durchgeführt. Anhand der Methode *print_topics()* wurden die entsprechenden ermittelten Top-Themen inklusive deren Gewichtung für die spätere Weiterverwendung in eine .csv-Datei geschrieben. Dabei wurde je Artikel das Datum, sowie die zugehörigen Themen mit Gewichtung und die Themen ohne Gewichtung gespeichert. Anhand der Python Library pyLDAvis und dem Funktionskonvolut *pyLDAvis.gensim_models* wurden die Ergebnisse der LDA jeweils visualisiert und die adäquate Parametersetzung für die optimalen Ergebnisse evaluiert[17] (Islam, 2019). Unter Einsatz des Prototyping-Verfahrens wurden so die passenden Parameter für die LDA identifiziert und angewandt.

Das zuvor beschriebene Vorgehen wurde jeweils für das Wall Street Journal, als auch die New York Times durchgeführt. Die Ergebnisse wurden in separaten .csv-Dateien festgehalten. Da es wiederkehrend Tage mit mehreren Zeitungsartikeln zu den gewählten Suchbegriffen gab, wurde in einem zweiten Schritt für jede Zeitschrift eine neuerliche LDA durchgeführt. Dabei wurde der Output der ersten LDA als Ausgangsbasis herangezogen. Je Tag wurden alle extrahierten Themen aus der ersten Latent Dirichlet Allocation aggregiert. Da es zu bis zu drei Artikeln pro Tag kam, ergaben sich so bis zu 30

[16] Gensim. Python; OS Independent. Abgerufen 9. März 2024, von https://radimrehurek.com/gensim/
[17] pyLDAvis: Interactive topic model visualization. Port of the R package. (3.4.1). ([Python]. Abgerufen 30. März 2024, von https://github.com/bmabey/pyLDAvis

extrahierte Themen pro Tag. Diese Daten wurden wiederum in einem Dictionary je Tag gespeichert und aus der Gesamtmenge aller Artikel ein bag of words für die zweite LDA ermittelt. Mithilfe der bereits angeführten Funktionen aus den Python-Bibliotheken gensim und pyLDAvis wurde ein adäquates LDA-Modell generiert und anhand diesem die Top zehn Themen je Kalendertag ermittelt. Die so gewonnen Top-Themen wurden ebenfalls in einer .csv-Datei unter Zuweisung des zugehörigen Datums und als Input für das spätere neuronale Netz zur Aktienindexprognose gespeichert. Das stufenweise Speichern, der Einzelergebnisse der jeweiligen LDA-Schritte erfolgte zur manuellen Nachkontrolle der Ergebnisse, um dadurch eine Erhöhung der späteren Prognosegenauigkeit für das LSTM-Modell zu erzielen. Je Zeitung wurden den Grundsätzen von Mayring zur qualitativen Inhaltsanalyse und der zugehörigen Kategorienbildung folgend, auf diesem Weg, die Top zehn Themen pro Tag ermittelt (Mayring, 2010).

```
04.01.2023,stock,climb,continu,remain,clau,santa,new,global,wednesdai,gain
05.01.2023,said,year,investor,fed,price,genter,inflat,dow,fell,wednesdai
06.01.2023,year,patten,said,job,kong,rate,hong,inflat,month,market
09.01.2023,investor,market,rose,trader,price,recent,monei,deep,share,trade
10.01.2023,year,rose,manag,bank,fed,earn,point,invest,rate,said
```

Abbildung 6: Beispielauszug LDA - Wall Street Journal csv

<u>One Hot Encoding</u>

Da ein neuronales Netz des Typs Long Short Term Memory nicht mit textuellen Daten arbeiten kann, mussten in einem weiteren Schritt die generierten Themen in nominale nummerische Daten umgewandelt werden. Dabei wurde auf das One Hot Encoding Verfahren zur Umwandlung der Textinformationen in kategoriale Daten zurückgegriffen. Studien mit Finanzmarktbezug, etwa zur Verbesserung der Kreditrisikoklassifizierung zeigen, dass das gewählte Verfahren im gegenständlichen Kontext zielführend ist (Yu et al., 2022). Mithilfe des One Hot Encodings wurden je nach Vorkommen eines Themas pro Tag ein binärer Wert von 1 für das Vorhandensein oder 0 für das nicht Vorhandensein zugewiesen. Um diese Methodik in Python zu implementieren wurde auf die Library scikit-learn und das Funktionspaket *OneHotEncoder* zurückgegriffen (Karimi, 2021). Das zuvor generierte .csv-File mit den ermittelten Themen je Finanznachrichtenplattform wurde mittels pandas als DataFrame eingelesen. Im nächsten Schritt wurde je Thema das One Hot Encoding durchgeführt. Theoretisch betrachtet ergeben sich aus diesem Verfahren bei 10 Themen pro Tag in einem gesamten Kalenderjahr 10 Themen x 365 Tage

= 3.650 kodierte binäre Werte pro Zeitschrift, die später als Inputfaktoren für das LSTM zur Dow-Jones-Industrial-Average-Prognose herangezogen werden sollen. Um die spätere Menge an Inputfaktoren zu reduzieren und die Inputgewichtung der anderen Faktoren wie Aktienschlusskurs und Reddit-Vader-Sentiment verhältnismäßig nicht zu verwässern, sowie keine Performanceeinbußen zu generieren, wurden in weiterer Folge die Top3 Topics aus den 10 Tagesthemen herangezogen und mittels One Hot Encoding in binäre Daten umgewandelt. Auf diesem Weg wurden 488 binäre Werte zu Themen zugeordnet. Beispielhaft „NYTTop1_bankman“ oder „NYTTop1_bond“, die je nach Auftreten pro Tag mit 1 oder 0 versehen wurden. Da bestimmte Themenwerte im Untersuchungszeitraum mehrmals vorkamen, wurde in einem nächsten Schritt nach doppelten Themenwerten innerhalb der Gesamtmenge aller vergebenen Binärwerte gesucht. Dabei wurden ausgehend vom Themenwert aus dem Rang Top1 nach identischen Themenwerten in den Rängen Top2 und Top3 gesucht. War an bestimmten Tagen ein Themenwert, der bereits in Top1 existiert auch in Top2 oder Top3 vorhanden (zugehöriger Binärwert = 1), wurde der Binärwert von Top1 auf 1 und der Binärwert des Rangs Top2 bzw. Top3 auf 0 gesetzt. Durch iteratives Wiederholen dieses Vorgehens gelang es, mehrere doppelte Themenwerte in Top2 und Top3 für alle Kalendertage im Untersuchungszeitraum auf 0 zu setzen (Binärwert = 0). Abschließend wurden die nullgesetzten Themenwerte in Top2 und Top3 aus dem DataFrame gelöscht. Auf diese Weise konnten bei Vorhandensein doppelter Begrifflichkeiten (Themenwerte) pro Tag (Binärwert = 1), diese jeweils „zusammengefasst“ werden, bei gleichzeitiger Reduktion der späteren Inputfaktoren für das Prognose-LSTM. Das oberhalb dargestellte Beispiel vervollständigend wurde durch diesen Algorithmus exemplarisch der Themenwert „NYTTop3_bankman“, mit Binärwert 1 am 10.03.2023 nullgesetzt, da bereits der Themenwert „NYTTop1_bankman“ im Untersuchungszeitraum existierte. Dieser wurde dann für den 10.03.2023 mit dem Binärwert 1 versehen.

Auf diesem Weg konnten beim Wall Street Journal 88 und bei der New York Times 41 Spalten des DataFrames eingespart werden. In Summe wurden für das WSJ 140 und für die NYT 219 Binärwerte zur Weiterverarbeitung im LSTM generiert. Ein Auszug der bereinigten one-hot-kodierten Daten des Wall Street Journal kann unterhalb eingesehen werden.

Tabelle 6: Auszug WSJ-CSV One Hote Encoding

Date	WSJTop1_asset	WSJTop1_bank	WSJTop1_buffer	WSJTop1_consum
01.01.2023	1	0	0	0
02.01.2023	0	0	0	0
03.01.2023	0	0	0	0
04.01.2023	0	0	0	0
05.01.2023	0	0	1	0
06.01.2023	0	0	0	0
07.01.2023	1	0	0	0

Die bereinigten Daten wurden abschließend je Zeitung in einem .csv-File unter Angabe des Datums und der Binärwerte je Tag gesichert.

4.2.4 Datenzusammenführung

In den Abschnitten 4.1 zur Datenerhebung und 4.2 zur Datenaufbereitung wurde die Einholung und Vorbereitung der jeweiligen Einzeldateien als Inputvariablen für das LSTM-Prognosemodell beschrieben. Als finaler Schritt zur Aufbereitung all dieser Einzeldaten für die Weiterverarbeitung im LSTM wurden jene in eine Gesamtinputdatei mit dem Kalenderdatum als Index zusammengeführt. Dazu wurden mithilfe der Python-pandas-Bibliothek alle Einzeldateien, jene zum Aktienindex, zu den VADER-Sentimentdaten, sowie die Binärdaten der beiden Zeitungen WSJ und NYT als DataFrames eingelesen. Um alle Daten miteinander über das Datum zu verbinden war es erforderlich das Datumsformat zu vereinheitlichen. Anschließend konnte über die *concat*-Funktion von pandas ein Gesamt-DataFrame, welcher alle Inputfaktoren enthält, erstellt werden (McKinney, 2011). Dieser DataFrame wurde abschließend als .csv-Datei gespeichert. Als Resultat aller Erhebungs- und Aufbereitungsschritte lag ein tagesfein granuliertes File mit den folgenden Informationen vor:

Datum | Erster | Hoch | Tief | Schlusskurs | Volumen | Tages Gewinn Verlust | Intraday Gewinn | Intraday Verlust | Vortag Gewinn Verlust | Vortag Gewinn | Vortag Verlust | Vader | NYTTop1_a… | … | NYTTop3_z… | WSJ_Top1_a… | … | WSJTop3_util

In Summe wurden aus den definierten vier Inputfaktoren 372 Inputvariablen gebildet.

Tabelle 7: LSTM-Inputvariablen

LSTM-Inputvariablen		
Nr.	**Inputstream**	**Variablenanzahl**
0	Datum	1
1	Historische Aktienentwicklung	11
2	VADER Sentiment	1
3	New York Times Binärwerte	219
4	Wall Street Journal Binärwerte	140
	Summe	**372**

4.3 Prototyp des Prognosemodells

Im nachfolgenden Kapitel wird auf die Konzeption und die angenommenen Ausgangsparameter des LSTM-Prognosemodells und deren zugehörige Überlegungen eingegangen. Jene werden weiterführend in Abschnitt 5 mittels Prototyping-Verfahren angewandt und optimiert.

Die aufbereiteten Daten wurden in ein Long Short Term Memory-Modell als Inputvariablen eingelesen. Das LSTM ist dabei als Typ eines rekurrenten neuronalen Netzes ausgeführt, dass über einen internen Zellzustand (Gedächtnis- „Memory“) für einen gewissen Zeitraum verfügt. Vorhandene Literatur zeigt, dass insbesondere rekurrente neuronale Netze bessere Ergebnisse in der Prognose von Daten einer Zeitreihe, wie es bei Finanzmarktdaten der Fall ist, liefern, als Feed-Forward-Netze (Vogt, 2021). Konkret erweisen sich LSTM-Modelle aufgrund des internen Zellzustandes hierfür als adäquat (Althelaya et al., 2018). Die Daten werden vorwärtsgerichtet durch das LSTM-Netz propagiert. Im Unterschied zu herkömmlichen rekurrenten neuronalen Netzen, können LSTM-Netze mithilfe des internen Zellzustandes und dessen Prozesse, Daten speichern und wieder „vergessen“ (Forget Gate) (Swathi et al., 2022). Diese Funktionalität soll sich auch im gegenständlichen Prototyp zu Nutze gemacht werden. Weitere Details zu LSTMs allgemein können dem Abschnitt 2.3 Neuronale Netze entnommen werden.

Die zugrundeliegende zu prüfende Hypothese der gegenständlichen Arbeit besagt, dass es durch Verknüpfung historischer Aktienkursinformationen des Dow Jones Industrial Average, ergänzt um Sentimentinformationen aus sozialen Medien, sowie aus Ergebnissen, gewonnen durch qualitative Inhaltsanalyse, aus Artikeln der Fachmagazine New York Times und Wall Street Journal möglich ist, ein Prognosemodell auf Basis neuronaler Netze zu entwickeln, welches die Meinungen von Kleinanlegern und Investitionsexperten, sowie dem historischen DJIA-Aktienindexkurs verbindet und zumindest temporär eine Vorhersagegenauigkeit von 60% für den Anstieg oder Abfall des Aktienindexschlusskurs am Folgetag erreicht.

Als wesentliche Messgröße wird hierbei die Prognosequote für den Folgetag (folglich auch als PQ bezeichnet) herangezogen. Die Prognosequote berechnet sich nach folgendem Schema: Es erfolgt ein Vergleich des tatsächlichen Kurses des heutigen Tages mit dem tatsächlichen Kurs des Folgetags. Ist der Kurs des Folgetags höher als der des heutigen Tages und ist auch der vorhergesagte Prognosekurs des Folgetags höher als der Prognosekurs des heutigen Tages, erfolgt eine Addition von 1 auf die Summe der

korrekten Prognosetage P_{Sum} im Testzeitraum P_{Tage}. Selbes erfolgt auch, wenn der tatsächliche Kurs am Folgetag fällt und auch der Prognosekurs am Folgetag fällt. Im umgekehrten Fall, fällt exemplarisch der tatsächliche Kurs am Folgetag, der Prognosekurs steigt allerdings am Folgetag gegenüber dem heutigen Prognosetag, erfolgt keine Addition von 1 auf die Summe der korrekten Prognosetage. Nach diesem Muster wird der gesamte Test- bzw. Prognosezeitraum durchlaufen. Am Ende erfolgt die Berechnung der Prognosequote anhand folgender Formel:

Formel 4: Prognosequote PQ Folgetag

$$PQ = \frac{P_{Sum}}{P_{Tage}} * 100$$

Der Hypothese folgend werden die zuvor genannten Inputvariablen zugrunde gelegt. Da Untersuchungen zeigen, dass eine möglichst breite Basis zu den vergangenen Aktienindexaktivitäten wie Handelsvolumen, sowie Intraday Gewinne oder Verluste positiv zur Prognosegenauigkeit beitragen können (Long, 2014), wurde initial nicht nur der historische Aktienindexschlusskurs zur Prognose ebendessen herangezogen, sondern die folgenden Daten als Inputvariablen dieses Aktienindex-Inputstreams verwendet:

Datum | *Erster* | *Hoch* | *Tief* | *Schlusskurs* | *Volumen* | *Tages Gewinn Verlust* | *Intraday Gewinn* | *Intraday Verlust* | *Vortag Gewinn Verlust* | *Vortag Gewinn* | *Vortag Verlust*

Im Rahmen des Prototyping Verfahrens wurde jedoch auch weiter untersucht, welche Auswirkung auf die Prognosegenauigkeit erzielt werden kann, wenn nur der Aktienindexschlusskurs genutzt wird.

Ergänzend zu den Finanzmarktdaten wurde der ermittelte tägliche Reddit-VADER-Sentimentwert gemäß Abschnitt 4.2.2 herangezogen.

Um die Expertenmeinungen der Hypothese zu berücksichtigen, wurden die mittels Latent Dirichlet Allocation und One Hot Encoding nach Kapitel 4.2.3 ermittelten Binärwerte der beiden Fachmagazine New York Times und Wall Street Journal für die weitere Prognose als Inputvariablen eingesetzt. Auf diese Weise ergeben sich pro Tag 372 Inputvariablen, welche für die Prognose des Aktienindexschlusskurses für den Folgetag eingesetzt wurden. Dem Effizienzgedanken nach Swathi et al 2022 folgend wurde initial ein LSTM mit folgender Struktur definiert, welches im Rahmen des Prototypings weiter untersucht wurde (Swathi et al., 2022).

Tabelle 8: LSTM-Prototyp Ausgangszeitpunkt t0

LSTM-Prototyp *zum Ausgangszeitpunkt t0*	
Inputvariablen	372
Anzahl LSTM-Layer (verborgene Schichten)	3
LSTM-Layer	**Anzahl Neuronen**
1	128
2	64
3	32
Outputvariablen	1

Als Optimierungsalgorithmus wird der Adam – Adaptive Moment Estimation – entwickelt von Kingma und Ba, eingesetzt (Kingma & Ba, 2014). Der Adam-Algorithmus basiert auf dem stochastischen Gradientenabstieg mit Impuls und Momentum und berechnet die adaptive Lernrate für jeden Parameter unter Hinzunahme von Momentum, bei dem jenes zusätzlich anhand früherer Anpassungen modifiziert wird. Kingma und Ba folgend handelt es sich hierbei um eine recheneffiziente Methode, welche wenig Speicher benötigt und sich gut für Probleme mit großer zugrundeliegende Datenmenge eignet (Kingma & Ba, 2014). Studien zeigen zudem, dass sich der Adam-Algorithmus für Deep Learning Problemstellungen universell anwenden lässt und aus diesem Grund einen guten Optimierungsausgangspunkt für die gegenständliche Untersuchung darstellt (Kingma & Ba, 2014).

Für das Modelltraining wurden initial 80 % des Inputdatensatzes herangezogen. Als Validierungsanteil wurden davon wiederum 20% des Trainingsdatensatzes festgelegt. Bezogen auf den Untersuchungszeitraum wurde daher der Zeitraum von 01.01.2023 bis 18.10.2023 als Trainingsdatensatz deklariert, wobei hiervon der Abschnitt vom 22.08.2023 bis 18.10.2023 zur Datenvalidierung eingesetzt wurde. Als Prognosezeitraum verblieb somit die Spanne zwischen 19.10.2023 und 31.12.2023, die mithilfe des gegenständlichen Modells, gemäß der zugrundeliegenden Hypothese, untersucht wurde. Um extreme Aktienindexentwicklungen, wie exorbitante kurzzeitige Anstiege oder Abfälle des DJIA zu berücksichtigen und auch prognostizieren zu können, wurden für das Prognosemodell alle Daten des gesamten Kalenderjahres zwischen 0 und 1 normalisiert (Hochreiter & Schmidhuber, 1997). Würde hier beispielhaft nur der Trainingszeitraum gewählt werden oder die Normalisierung gar nicht stattfinden, so könnte der Fall eintreten, dass das Prognosemodell nicht in der Lage ist große Kursanstiege oder -abfälle, wie es gegen Ende des Kalenderjahres 2023 beim Dow Jones Industrial Average auch der Fall war, zu prognostizieren, da im Trainingszeitraum solch hohe oder niedrige Werte nicht aufgetreten sind (Ariva, 2023).

Die Sequenzlänge, jener Zeitabschnitt in der Vergangenheit, der zur Prognose des Folgewerts genutzt werden soll, wurde initial mit zehn Tagen festgelegt. Als initiale Batchsize wurde in Anlehnung an Rokhsatyazdi et al, 2020 der Wert 64 angenommen (Rokhsatyazdi et al., 2020). Als Batchsize wird die Zahl der Trainingselemente je Trainingsdurchlauf durch das LSTM verstanden. Der Wert 64 entspricht im gegenständlichen Fall daher 64 Kalendertagen. Als Verlustfunktoin (Loss) zur Evaluierung des Modells über die Modellgewichte wurde der mittlere quadratische Fehler (Mean Squared Error – MSE) eingesetzt. Er ist als der erwartete quadratische Abstand des Schätzwerts definiert und berechnet sich wie folgt.

Formel 3: Mean Squared Error - MSE

$$MSE = \frac{1}{n}\sum_{t=1}^{t=n}(y' - y)^2$$

MSE beinhaltet dabei sowohl die Varianz als auch die Verzerrung zwischen Prognose- und wahrem Wert. y' stellt den prognostizierten und y den wahren Wert dar. n wiederum ist die Gesamtzahl aller Werte im Testsatz. Im gegenständlichen Kontext ist ein MSE-Wert umso besser je näher er sich an Null annähert (Hüttner, 1986). Dahingehend soll auch das LSTM-Modell im Rahmen des Prototyping-Verfahrens immer weiter optimiert werden - um einen möglichsten geringen MSE-Wert als Verlust (engl. Loss-Value) zu erzielen.

Zur Messung der Modellgüte des LSTM-Prognoseprototypen wurden neben dem mittleren quadratischen Fehler, auch die Quadratwurzel des mittleren quadratischen Fehlers (Root Mean Squared Error – RSME) und der mittlere absolute Fehler (MAE – mean absolute Error) eingesetzt. Wie auch für den MSE-Wert gilt für beide weiteren Messgrößen, je näher sich jene Werte an Null annähern, desto besser ist die Modellgüte und daraus folgend die Prognose (Hüttner, 1986).

Neben den oberhalb angeführten Messgrößen wurde das Prognosemodell anhand der Visualisierung einzelner Parameter validiert. Dazu kam die Darstellung der Entwicklung der Verlustfunktion von Training, Validierung und Testdatensatz zur Anwendung. Hierbei konnte die Annäherung der jeweiligen Verlustwerte je Modelldurchlauf beobachtet werden.

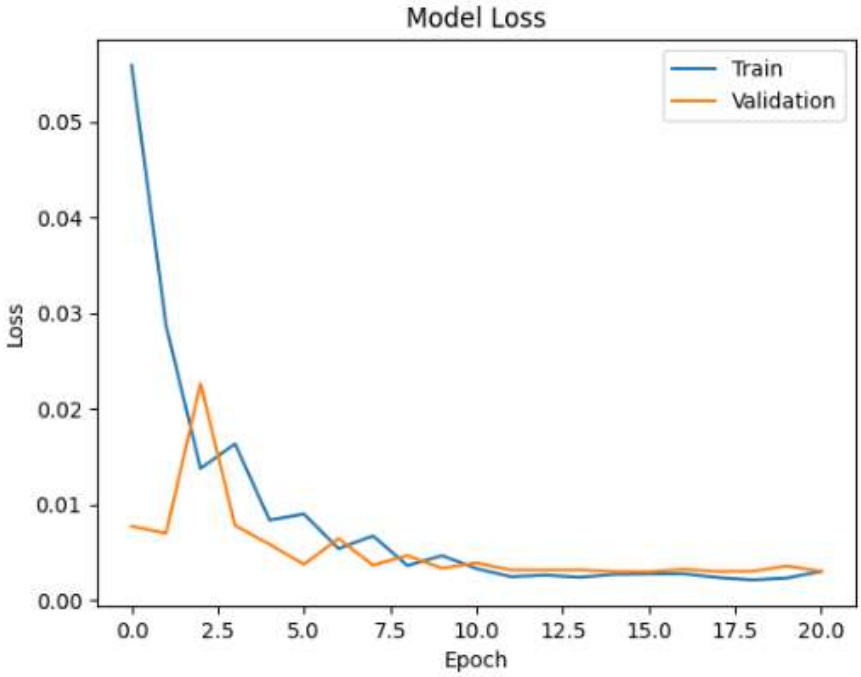


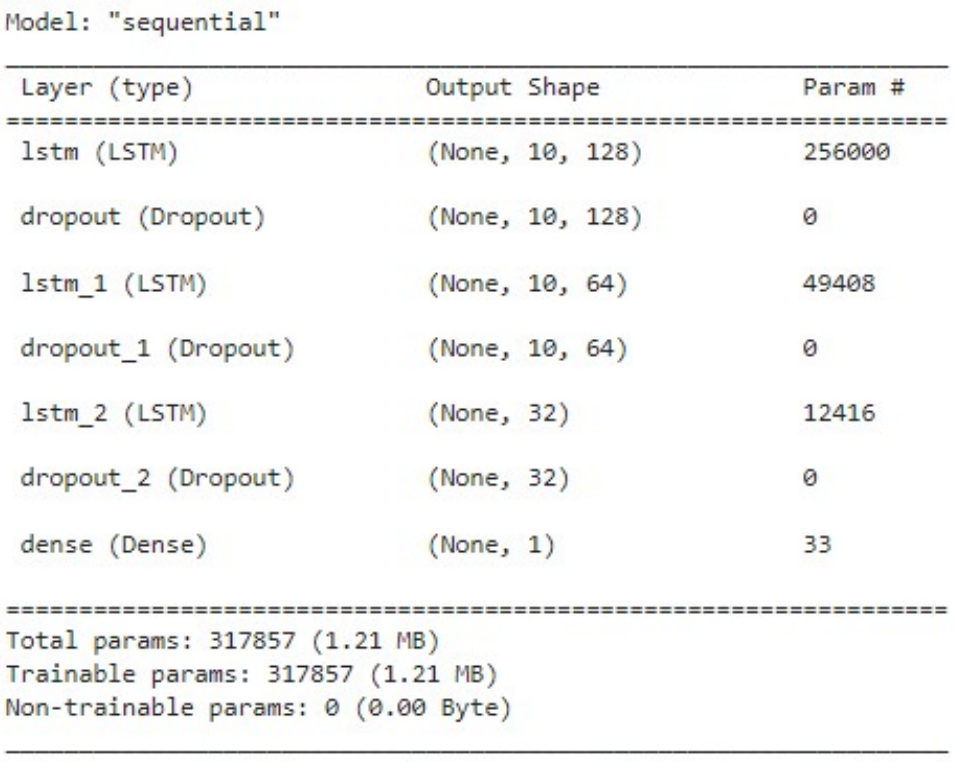
Abbildung 7: Trainings- und Validierungsverlustentwicklung Gegenüberstellung initiales Prototypmodell

Neben den Verlustkurven wurde zudem die Entwicklung des tatsächlichen Dow Jones Industrial Average Kurses dem prognostizierten Kurs im Trainings-, Validierungs-, und Testzeitraum gegenübergestellt. Insbesondere die Entwicklung und Annäherung der beiden Kurven im Testzeitraum ist für die Beurteilung im Rahmen des Prototyping-Verfahrens relevant. Abschließend wurde anhand eines Vergleichs der prognostizierten und der tatsächlichen Aktienindexschlusskurswerte die Vorhersagegenauigkeit des Modells in Prozent für den Folgetag ermittelt und dahingehend die aufgestellte Hypothese geprüft.

Zur Implementierung des oberhalb dargestellten LSTM-Prognosemodells wurde auf mehrere Python Bibliotheken zurückgegriffen. Um die benötigten Inputvariablen für das Modell einzulesen wurde erneut die Library pandas eingesetzt. Alle Eingabeinformationen, beschrieben in Kapitel 4.2, wurden in einen pandas-DataFrame eingelesen. Dabei wurde das Datum als DataFrame-Index festgelegt. Zur Berechnung der oberhalb beschriebenen Messparameter zur Modellgüte, sowie zur Formatierung der Daten als Input-Arrays für das LSTM wurde die Python Library NumPy eingesetzt. Die Normalisierung der Daten wurde mithilfe der Python Bibliothek Scikit-Learn und deren Funktionskonvolut *MinMaxScaler()* realisiert[18] (Karimi, 2021).

[18] Sklearn. Scikit-Learn. Sklearn.preprocessing.OneHotEncoder. Abgerufen 16. März 2024, von https://scikit-learn/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

Um das Prognose-LSTM zu implementieren, wurde auf TensorFlow und die Python Library Keras zurückgegriffen. Bei TensorFlow handelt es sich um eine plattformunabhängige Programmbibliothek spezialisiert auf Machine Learning Anwendungen, die ursprünglich von Google entwickelt wurde, mittlerweile jedoch unter Open-Source-Lizenz frei zur Verfügung steht. Die Software zeichnet sich durch ihre Leistungsfähigkeit und Skalierbarkeit aus und erlaubt es lernende neuronale Netze zu erstellen (Ramsundar & Zadeh, 2018). Keras stellt wiederum ein Python API für Deep Learning Anwendungen dar, welches in der Lage ist auf das TensorFlow Framework aufbauend zu arbeiten. Aufgrund dessen ermöglicht es die Keras Bibliothek in Python, Machine Learning Modelle wie neuronale Netze zu implementieren. Die Grundstruktur von Keras gliedert sich dabei in *models* und *layers* (Ramsundar & Zadeh, 2018). Ein *model* besteht aus mehreren *layers*. Für den oberhalb beschriebenen Prototyp wurde daher ein lineares *model* vom Typ *keras.Sequential()* angelegt. Mittels *model.add()* wurden dem Prognosemodell nachfolgend die drei beschriebenen LSTM-Layer mit der entsprechenden Anzahl an Neuronen (*units*) hinzugefügt. Der beschriebene Adam-Optimierungsalgorithmus (*optimizer=keras.optimizers.Adam()*), sowie die MSE-Verlustfunktion (*loss='mean_squared_error'*) wurden weiterführend im Rahmen der Modelkompilierung durch *model.compile()* festgelegt. Als Lernrate für den Adam-Algorithmus wurde initial 0.01 festgelegt. Die Modelzusammenfassung des initialen Prognose-LSTM zur Vorhersage des Dow-Jones-Industrial-Average im Betrachtungszeitraum 2023 findet sich nachfolgend.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 10, 128)           256000

 dropout (Dropout)           (None, 10, 128)           0

 lstm_1 (LSTM)               (None, 10, 64)            49408

 dropout_1 (Dropout)         (None, 10, 64)            0

 lstm_2 (LSTM)               (None, 32)                12416

 dropout_2 (Dropout)         (None, 32)                0

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 317857 (1.21 MB)
Trainable params: 317857 (1.21 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Abbildung 8: Prototyp initiales Prognosemodell zum Zeitpunkt t0

Um Overfitting zu vermeiden, wurde ein Callback mittels der *EarlyStopping*-Funktion von Keras festgelegt, welches die Entwicklung der Verlustfunktion des Validierungsdatensatzes überwacht (*monitor='val_loss')*, und nach fünf nahezu gleichbleibenden, sich nicht verbessernden Werten, das zuvor beste Modell speichert und abschließend das Modelltraining beendet (*patience=5, restore_best_weights=True).*

Auf Basis der oberhalb dargestellten Parameter wurde das initiale Prototypmodell zum Ausgangszeitpunkt (t0) in 21 Epochen angelernt. Das auf diesem Weg beste ermittelte Modell wies für den Trainingsdatensatz einen Verlustwert (MSE) von 0.0021, für den Validierungsdatensatz von 0.0030 auf. Angewandt auf den Testdatensatz stieg der MSE jedoch auf 0.0345. Die Gegenüberstellung der Entwicklung der MSE-Werte für Trainings- und Validierungsdatensatz kann der *„Abbildung 7: Trainings- und Validierungsverlustentwicklung Gegenüberstellung initiales Prototypmodell"* oberhalb entnommen werden. Die weiteren Prüfwerte für den Testdatensatzes beliefen sich auf Mean Absolute Error (MAE) 0.1566 und Root Mean Squared Error (RMSE) 0.1857.

Die Prognosegenauigkeit, die korrekte Vorhersage eines Kursanstiegs bzw. -abfalls für den Folgetag lag bei 57.90%, weshalb die Hypothese mit dem initialen Modell zum Zeitpunkt t0 zu verwerfen gewesen wäre. In *„Abbildung 9: Schlusskursvorhersage Testzeitraum initiales Prototypmodell zum Zeitpunkt t0"* wird die Visualisierung des

prognostizierten gegenüber dem tatsächlichen Aktienindex des Dow Jones für den Testdatensatz dargestellt.

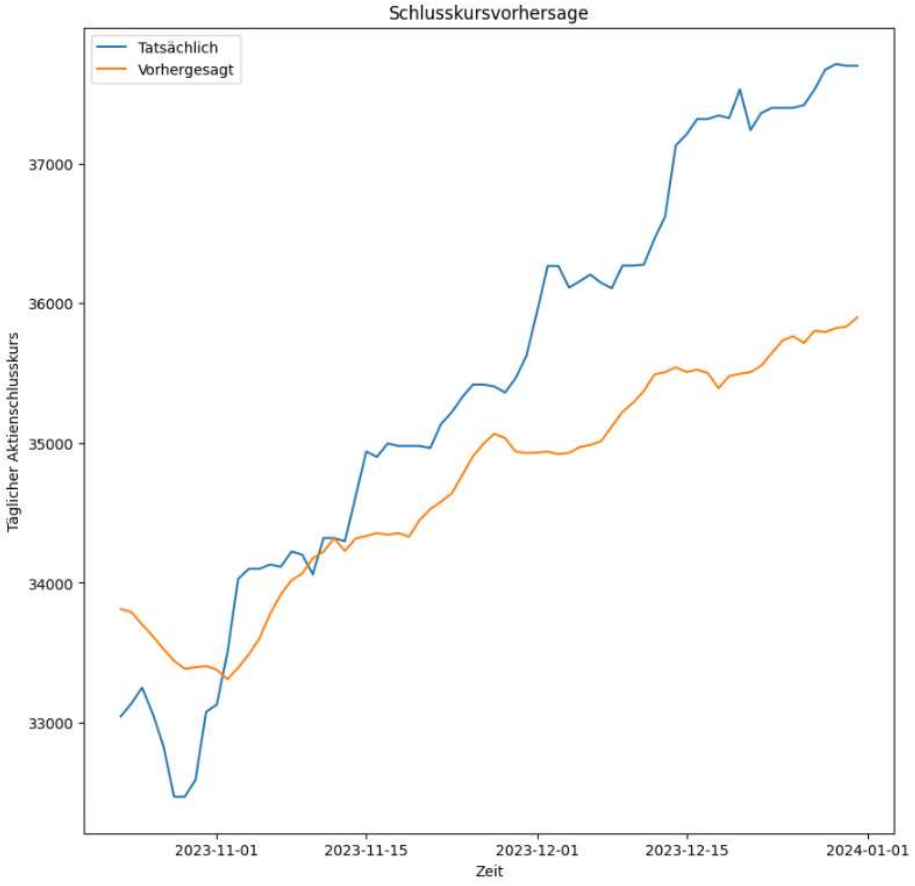


Abbildung 9: Schlusskursvorhersage Testzeitraum initiales Prototypmodell zum Zeitpunkt t0

5 Anwendung des Prognosemodellprototypen

Vorhandene Literatur zeigt, dass es unter Zuhilfenahme von rekurrenten neuronalen Netzen oder anderen Machine Learning Techniken möglich ist, Finanzmarktdaten zu prognostizieren (Nelson et al., 2017). Dabei wurde etwa wie bei Zhang et al. 2011 oder Vogt 2021 auf die Prognose anhand von Twitter-Sentimentdaten zurückgegriffen (Vogt, 2021; Zhang et al., 2011) oder rein der historische Aktienkurs herangezogen (Chakraborty et al., 2017). Wieder andere Studien konzentrierten sich auf die Prognose unter Einsatz von Interaktion mit Suchmaschinen (Vasileiou et al., 2021). Eine Untersuchung zur Prognose von Finanzmarktdaten auf Basis der in dieser Arbeit angewandten Eingabevariablen wurde jedoch noch nicht angestellt. Aus diesem Grund wurde zu Verfeinerung des Prognosemodells das Prototyping Verfahren verwendet. Das initiale Ausgangsmodell zum Zeitpunkt t0 wurde auf Basis der Erkenntnisse vorhandener wissenschaftlicher Arbeiten, sowie unter der Grundannahme zur Schaffung eines recheneffizienten Modells, wie unter Pkt. 4.3 ausgeführt, konzipiert (Swathi et al., 2022).

Im nachfolgenden Kapitel werden die durchgeführten Prototypingschritte beschrieben, die zur Ermittlung des optimalen Modells dieser Untersuchung führten. Es wird dabei auf die Anpassung der einzelnen Modellparameter, sowie deren Auswirkungen auf das Prognosemodell eingegangen. Abschließend wird das Endmodell dargestellt und erläutert.

Einleitend wird daher nochmals die zugrundeliegende Hypothese dargelegt. Jene besagt, dass es durch Verknüpfung historischer Aktienkursinformationen des Dow Jones Industrial Average, ergänzt um die VADER-Sentimentinformationen von der Plattform www.reddit.com/r/Wallstreetbets, sowie aus Ergebnissen, gewonnen durch qualitative Inhaltsanalyse, aus Artikeln der Fachmagazine New York Times und Wall Street Journal möglich ist, ein Prognosemodell auf Basis eines LSTMs zu entwickeln, welches die Meinungen von Kleinanlegern und Investitionsexperten, sowie dem historischen DJIA-Aktienindexkurs verbindet und zumindest temporär eine Vorhersagegenauigkeit von 60% für den Anstieg oder Abfall des Aktienindexschlusskurs am Folgetag erreicht.

Als wesentliche Messgröße wird hierbei die Prognosequote für den Folgetag (folglich auch als PQ bezeichnet) herangezogen. Definition und Berechnung der Prognosequote können Kapitel 4.3 entnommen werden.

Die Vorhersagequote des initialen Prognosemodells für den Folgetag mit 57,90% und einem MSE von 0,0345 beim Testdatensatz lässt darauf schließen, dass die Hypothese

durch entsprechendes Verfeinern des Modells belegbar scheint. Aus diesem Grund wurde im Rahmen des Prototyping Verfahrens initial begonnen, die Neuronenanzahl je LSTM-Layer anzupassen. In der ersten Phase wurden die Neuronenanzahl im ersten LSTM-Layer auf 256 und dann 512 erhöht. Es konnten auf diesem Weg jedoch nur geringfügige Verbesserungen des Modells erzielt werden. Aufgrund der geringen Datensatzmenge (365 Tage) wurde die passende Wahl der Batchsize im Rahmen des Prototypings als kritisch identifiziert. Mit der initial gewählten Batchsize kam es je Epoche nur zu vier Trainingsdurchläufen durch das LSTM. Im nächsten Schritt wurde daher die Batchsize halbiert und auf 32 gesetzt. Auf diesem Weg wurden im Umkehrschluss die Traingsläufe verdoppelt. Allerdings führte auch dies zu keiner merklichen Verbesserung der Prognosegenauigkeit. So wies ein Modell mit 512 Neuronen in der ersten, 64 in der zweiten und 32 in der dritten Schicht, mit einer Batchsize von 32, weiterhin einen MSE von 0,0309 und sogar eine reduzierte Prognosequote (PQ) für den Folgetag von 56,14% auf. Auch eine Erhöhung der Neuronenanzahl in den weiteren Schichten mit beispielsweise 1024-256-32 brachten nur geringe Verbesserungen in der Prognose für den Folgetag (PQ=59,65%, MSE=0,0299). Dem zuvor beschriebenen Vorgehen folgend wurden unterschiedliche Zusammensetzungen von Neuronen innerhalb der drei LSTM-Layer des Keras-Prognosemodells getestet. Es zeigte sich jedoch bei 17 unterschiedlichen Modellkonfigurationen keine ausreichende Verbesserung der Prognosegenauigkeit für den Folgetag, um die Hypothese zu belegen. Aus diesem Grund wurde im nächsten Schritt die Sequenzlänge, jener Zeitabschnitt in der Vergangenheit, der zur Prognose des Folgewerts genutzt werden soll, gekürzt. Im ersten Schritt erfolgte eine Reduktion von 10 auf 7 Kalendertage. Für die weitere Untersuchung wurde ausgehend vom besten Modell mit einer Sequenzlänge von 10 Kalendertagen (1024-256-32 Neuronen) weiter geprüft, und dessen Neuronenparameter in 23 Zyklen adaptiert. Dies führte ebenfalls zu keiner Verbesserung der Prognosewerte und Verlustfunktion. Im Zuge der Trainingsbeobachtung wurde festgestellt, dass der gesetzte *EarlyStopping*-Paramter mit *patience=5,* welcher zur Vermeidung von Overfitting eingerichtet wurde, zu streng gewählt war. Oftmals wurde dadurch das Modelltraining zu früh unterbrochen, und eine weitere Optimierung des Modells verhindert. Eine Neuparametrierung auf den Wert 10 zeigte deutliche Verbesserungen bei Trainings- und Validierungsverlust. Eine zusätzliche Reduktion der Sequenzlänger auf 5 Kalendertage und schlussendlich auf 3 Kalendertage führte aufbauend zu drastischen Verbesserungen des Modells und zur Optimierung der Prognosequote für den Folgetag auf über 60%. Ein LSTM-Modell unter Zuhilfenahme der in Kapitel 4.2 beschriebenen Eingabevariablen, mit einer LSTM-Struktur von 1024-256-32 Neuronen, einer Batchsize von 32 und einer Sequenzlänge von 3 Kalendertagen

erreicht eine Prognosegenauigkeit von 63,15% für den Folgetag. Durch eine Reduktion der Lernrate des Adam-Algorithmus von 0.001 auf 0.0001, sowie der Batchsize von 32 auf 16 Tage konnte eine weitere Verbesserung erzielt werden. Die Prognosegenauigkeit für den Folgetag änderte sich zwar nur geringfügig auf 63,16%, es konnte jedoch eine Prognosegenauigkeit von 94,29% für die Folgewoche (+7 Kalendertage), gemessen an der Aussage, ob es zu einem Schlusskursanstieg oder -abfall kommt, erzielt werden. Dieses Modell zum Zeitpunkt t57, stellte zu diesem Zeitpunkt der Untersuchung das präziseste Prognosemodell dar. Eine Visualisierung der Entwicklung der Verlustfunktion von Trainings- und Validierungsdatensatz befindet sich unterhalb. Ebenso ein Vergleich des prognostizierten und tatsächlichen DJIA-Tagesschlusskurs im Testzeitraum.

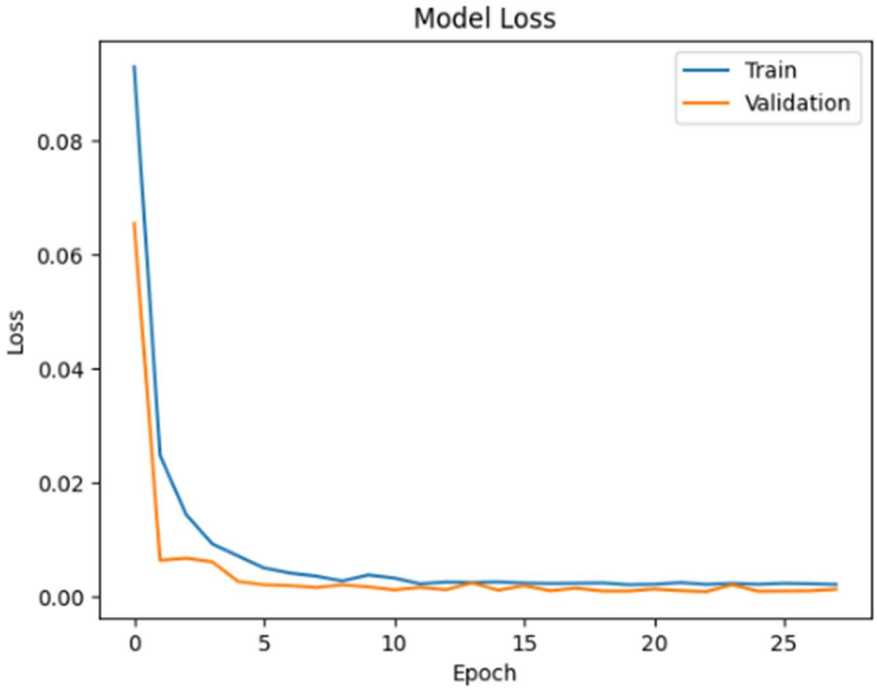


Abbildung 10: Verlustfunktion LSTM-Modell zum Zeitpunkt t57 mit Aufbau 1024-256-32

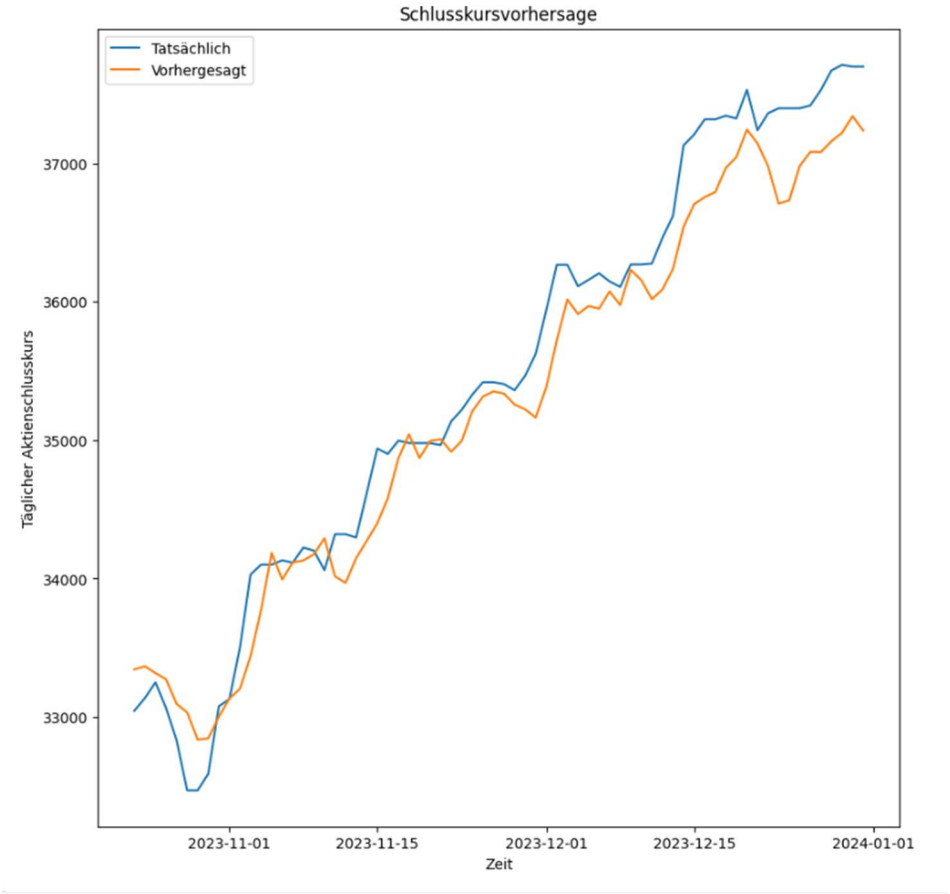


Abbildung 11: Prognose LSTM-Modell zum Zeitpunkt t57 mit Aufbau 1024-256-32

Weitere Anpassungen der Modell-Metriken führten dazu, dass die tatsächliche Prognose der vorhergesagten Dow-Jones-Industrials Tagesschlusskurswerte weiter signifikant verbessert wurde. Das präziseste Vorhersagemodell konnte mit t59 (Neuronenstruktur: 1024-512-128) erreicht werden. Die Maximalabweichung zum tatsächlichen Aktienkurs betrug max. 249 Punkte, der MSE lag bei nur 0.0013. Diese Entwicklung ging jedoch zu Lasten der Prognosequote des Schlusskursanstiegs und -abfalls für den Folgetag (PQ=61,40%). Dies lässt sich damit erklären, dass die Prognose des Aktienindexwerts zwar deutlich genauer war, jedoch an manchen Tagen den tatsächlichen Wert überstieg, obwohl dieser bereits stagnierte oder fiel. Die Prognosedarstellung unterhalb verdeutlicht dies.

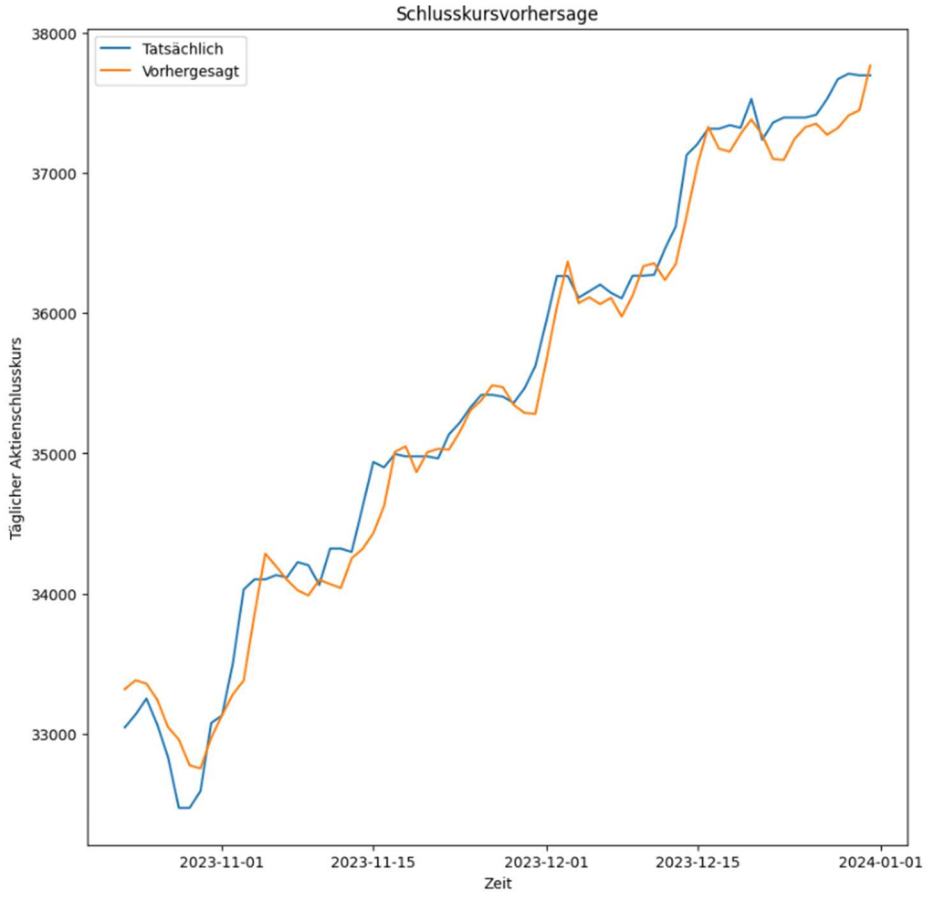


Abbildung 12: Prognosemodell zum Zeitpunkt t59

Interessanterweise wies jedoch die Prognosequote für die Folgewoche (+7 Kalendertage) dieses Modells den absoluten Höchstwert von 95,17% auf. Die höchste Prognosequote für den Folgetag konnte mit dem Modell t63 erreicht werden. Jene lag hier bei 71,93%, die Quote für die Folgewoche belief sich hier auf 91,43%. Der MSE erreichte einen Wert von 0.0046.

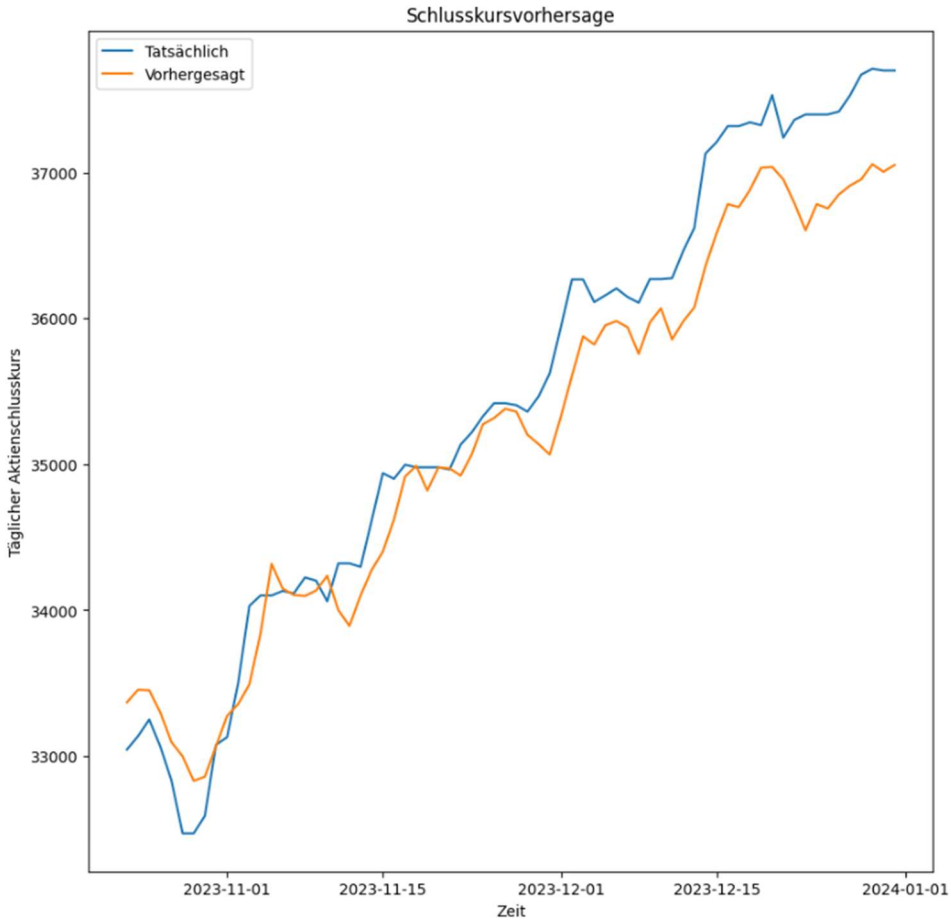


Abbildung 13: Prognose zum Zeitpunkt t63

In weiteren Prototypingzyklen wurde die Batchsize auf 12 Tage reduziert und auch die Neuronenanzahl bis zu maximal 4096 innerhalb der vorhandenen Layer adaptiert. Ergänzend wurden 23 Zyklen mit einem vierten LSTM-Layer getestet, welche jedoch keine erheblichen Verbesserungen an den Prognoseparametern hervorbrachten. Insbesondere das weitere Hinzufügen eines zusätzlichen Layers für die Prognose brachte mit den gegenständlichen Eingabevariablen keine ausschlaggebenden Verbesserungen an der Prognosequote. So erzielte ein Modell mit der Neuronen Konstellation 1024-512-256-128 eine Prognosequote von 64,51% (92,86% für die Folgewoche), bei einem MSE von 0.0065.

Weiterführend wurde geprüft, ob die Nichtberücksichtigung der folgenden Finanzmarktparameter das Prognosemodell beeinflussen würden.

Erster | Hoch | Tief | Volumen | Tages Gewinn Verlust | Intraday Gewinn | Intraday Verlust | Vortag Gewinn Verlust | Vortag Gewinn | Vortag Verlust

Dies würde die Forschungsergebnisse von Long 2014 bestätigen, welche besagen, dass eine Prognose rein auf dem Aktienschlusskurs nicht dieselbe Genauigkeit erreichen, wie Prognosen unter Zuhilfenahme mehrerer Handelsparameter wie etwa dem täglichen Handelsvolumen oder Start- und Schlusskurs. Um dies zu untersuchen wurde die idente Herangehensweise, wie schon oberhalb ausgeführt angewandt. Es wurde initial mit dem Prototypenmodell nach Kapitel 4.3 begonnen. Sukzessive wurden Anpassungen an der Neuronenanzahl, der Layeranzahl, Sequenzlänge und Batchsize durchgeführt. Ab dem ersten Modell wurden jedoch bereits die Learnings zum Adam-Optimierungsalgorithmus (Learning-Rate 0.0001) und der *EarlyStopping()*-Methode berücksichtigt. Auf diese Weise wurden weitere 52 Zyklen mit jeweils bis zu vier LSTM-Layern und bis zu 4096 Neuronen je Layer durchlaufen. Als optimales Modell in diesem Setting wurde t123 mit einer LSTM-Struktur von 2048-1024-512-256 Neuronen, einer Batchsize von 12 Tagen und einer Sequenzlänge von 3 Kalendertagen ermittelt. Jenes wies einen MSE von 0.0086, sowie eine Prognosequote von 61.70% für den Folgetag und 87.14% für die Folgewoche auf. Der MAE betrug 0.0760, der RSME 0.0927. Eine Visualisierung des Prognose- und tatsächlichen Kurses im Testzeitraum befindet sich unterhalb.

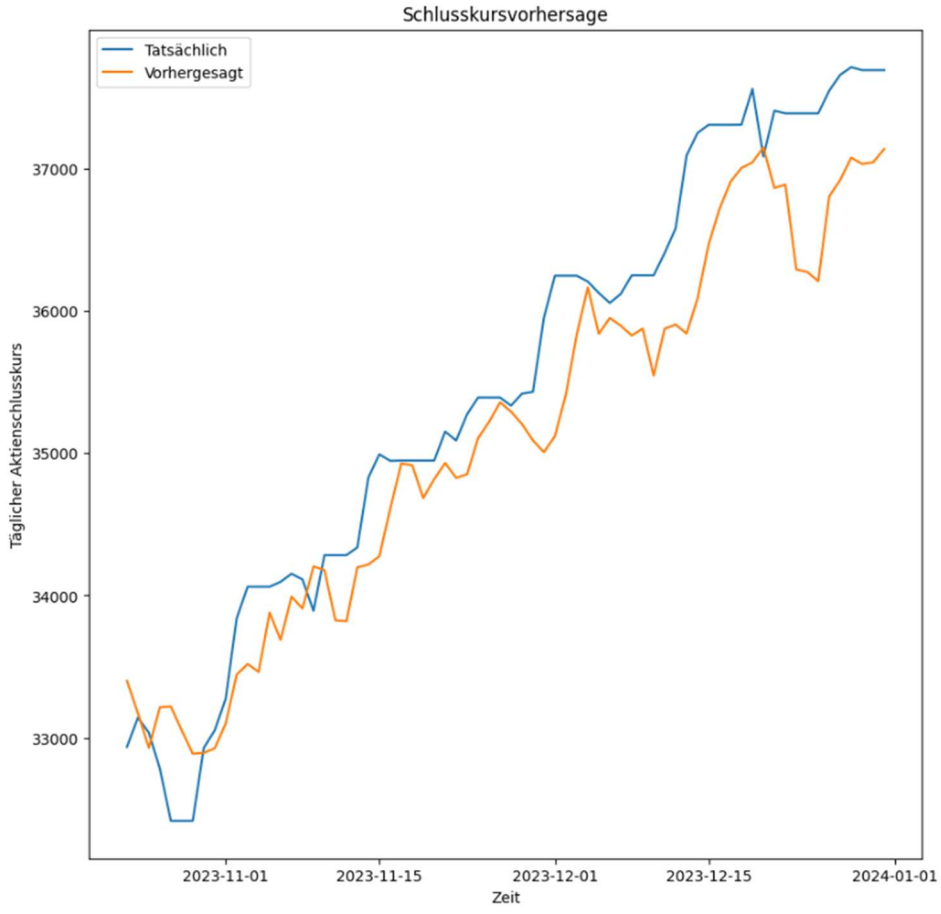


Abbildung 14: Prognosemodell zum Zeitpunkt t123

Im Vergleich zu den Prognosemodellen mit allen zur Verfügung stehenden Finanzmarkteingabevariablen zeigt sich, dass wie von Long 2014 erkannt, die Prognose nur unter Zuhilfenahme des Aktienindexschlusskurses, trotz der weiteren Inputparameter wie dem VADER-Sentimentwert und den Finanzmarktnachrichten ungenauer ausfällt als unter Einbezug mehrerer Finanzmarktparameter (Long, 2014). Hierbei ist erweiternd jedoch zu erwähnen, dass sich in dieser Eingabeparametersituation tatsächlich ein bestes Modell generieren lässt, dass sowohl die beste PQ für den Folgetag, die Folgewoche und den besten MSE liefert.

In Summe wurden im Rahmen des Prototypingverfahrens 131 Modelle in unterschiedlichen Parameterkonstellationen getestet. Dabei wurden je Prototypingzyklus unterschiedliche Parameter adaptiert. Im Zuge der Zyklen wurden Anpassungen an Trainings-, Validierungs- und Testdatensatzgröße, der Sequenzlänge, der Batchsize, der Neuronen- und LSTM-Layeranzahl, der Lernrate des Optimierungsalgorithmus, der Schranke zur Vermeidung von Overfitting (*EarlyStopping()*-Methode) und den Finanzmarkteingabevariablen vorgenommen. Es wurde erkannt, dass unterschiedliche Parameter des LSTM-Modells jenes jeweils in differenzierenden Weisen und Ausmaß beeinflussen. Unter Reduktion der Sequenzlänge auf drei Kalendertage, sowie der Reduktion der Batchsize von initial 64 auf 16 bzw. 12 Tage konnten signifikante Verbesserungen der Modelle erzielt werden. Insbesondere die Anpassung der Lernrate des Adam-Optimierungsalgorithmus auf 0.0001, sowie des EarlyStopping-Parameters führten zu präziseren Vorhersagemodellen. Es wurde zudem festgestellt, dass kein übergreifend optimales Modell existiert, welches in allen oberhalb beschriebenen Messparametern die besten Werte liefert. Unterhalb stellt daher die *Tabelle 9: Prognosemodelle* die Parameter der besten Modelle gemessen an den jeweiligen Parametern Prognosequote Folgetag, Prognosequote Folgewoche, Prognosegenauigkeit des Aktienindexschlusskurses selbst und der Prognosequoten für das Modell mit reduzierter Finanzmarktdaten dar.

Tabelle 9: Prognosemodelle

Prognosemodelle (optimiert je Messgröße)					
Nr.	**LSTM-Layer-Struktur**	**Batchsize**	**PQ (Folgetag)**	**PQ (Folgewoche)**	**MSE**
t63	1024-512-256	16	**71.93%**	91.43%	0.0046
t59	1024-512-128	16	61.40%	**95.71%**	**0.0013**
t77	1024-512-256-128	16	64.91%	92.86%	0.0065
t123*	2048-1024-512-256	12	61,70%	87,14%	0.0086
t44	1024-256-32	32	63,16%	**95,71%**	0.0160

*reduzierte Finanzmarktdaten

Gemessen an der gegenständlichen Hypothese hat sich das Modell t63 als effizientestes Prognosemodell erwiesen. Jenes erreicht eine Prognosequote von 71,93% für den Folgetag. Für einer Prognose des tatsächlichen Dow-Jones-Industrial-Average-Kurswertes, sowie für die Prognose der Folgewoche ist auf das Modell t59 zurückzugreifen. Bemerkenswert ist die Effizienz des Modells t44 hinsichtlich der PQ für die Folgewoche unter Bedachtnahme der geringeren Trainingsdurchläufe aufgrund der höheren Batchsize, sowie der reduzierten Neuronenanzahl.

6 Zusammenfassung

Im Rahmen der gegenständlichen Analyse wurde die Forschungsfrage untersucht, ob es unter Einsatz Künstlicher-Neuronaler-Netze gelingen kann, einen Prognosemodellprototypen für den Dow Jones Industrial Average Aktienindexschlusskurs zu entwickeln, welcher auf digital abrufbare Kleinanleger- und Expertenmeinungen im Prognoseprozess Rücksicht nimmt. Die zugehörige aufgestellte Hypothese lautet, dass es durch Verknüpfung historischer Aktienindexinformationen des Dow Jones Industrial Average, ergänzt um Sentimentinformationen, extrahiert aus Social Media zu diesem Aktienindex, sowie aus Ergebnissen, gewonnen durch qualitative Inhaltsanalyse, aus Artikeln der Fachmagazine New York Times und Wall Street Journal möglich ist, ein Prognosemodell auf Basis neuronaler Netze zu entwickeln, welches die Meinungen von Kleinanlegern und Investitionsexperten, sowie historischen Aktienindexkursen verbindet und zumindest temporär eine Vorhersagegenauigkeit von 60% erreicht. Jene Vorhersagegenauigkeit bezieht sich dabei auf den möglichen Schlusskursanstieg oder -abfall am Folgetag.

Zur Prüfung der aufgestellten Hypothese wurde im ersten Schritt nach geeigneten Formen von neuronalen Netzen recherchiert, welche sich für die multivariate Vorhersage von Daten in Zeitreihen eigenen. Aufgrund der breiten vorhandenen wissenschaftlichen Basis und der zugesprochenen Eignung wurde dafür ein Long Short Term Memory Modell gewählt (Swathi et al., 2022). Die Wahl der Social Media Quelle fiel auf www.reddit.com und dort dem Subreddit „Wallstreetbets“, da daraus gewonnene Sentimentinformationen bereits in der Vergangenheit erfolgreich für Finanzmarktprognosen eingesetzt werden konnten (Anand & Pathak, 2021). Die benötigten Aktienindexinformationen wurden vom deutschsprachigen Finanzportal www.ariva.de extrahiert und unter Einsatz der Programmiersprache Python für das Prognosemodell aufbereitet. Zur Extraktion der reddit-Submissions und Kommentare wurde auf das unabhängige Reddit-Archiv Pushshift.io zurückgegriffen, da seitens Reddit, die API-Policy insbesondere hinsichtlich Kosten und Requestmengen für Entwickler nachteilig angepasst wurden (Baumgartner, 2024). Über die Python-Pushshift-API wurden 7.9 Mio. Kommentare im Zeitraum von 01.01.2023 bis 31.12.2023 mithilfe der VADER-Sentimentanalyse, welche im Jahr 2014 speziell für die Social Media Analyse von Hutto und Gilbert entwickelt wurde, untersucht (Hutto & Gilbert, 2014). Aus den einzelnen Sentimentwerten je Kommentar, wurden aus allen Kommentaren eines Tages, Tagessentimentwerte generiert (siehe Kapitel 4.2.2).

Obwohl die New York Times über eine eigene kostenfreie Python-API verfügt, wurden die Finanzmarktnachrichten der New York Times über ein öffentlich zugängliches Nachrichtenarchiv und nicht über das NYT-API direkt bezogen[19]. Hintergrund dafür waren die vorhandenen Requestlimitierungen der API. Auf diesem Weg wurden 173 Artikel mit Bezug zum Dow Jones Industrial Average extrahiert. Die Artikel des Wall Street Journal wurden mittels Web Scraping Verfahren unter Zuhilfenahme der Python Library Selenium von der Website des WSJ abgerufen (García et al., 2020). Voraussetzung hierfür war das Verfügen über ein aktives Abonnement des Magazins. Nach automatisierten und manuellen Data-Clearing-Schritten verblieben 475 WSJ-Artikel zur weiteren Nutzung für das Prognosemodell (siehe Kapitel 4.1.3). Alle auf diesem Weg generierten Nachrichtenartikel der New York Times und des Wall Street Journal wurden weiterführend einer qualitativen Inhaltsanalyse unterzogen (Mayring, 2010). Dafür wurde das 2003 an der Stanford University entwickelte Latent Dirichlet Verfahren (kurz LDA) eingesetzt, um die Hauptthemen der jeweiligen Artikel zu extrahieren (Topic Modelling) (Blei et al., 2003). Auf diesem Weg wurden die Top zehn Themen je Artikel gewonnen und zwischengespeichert. Existierten an einem Tag mehr Artikel je Zeitschrift, so wurde in einem zweiten LDA-Schritt aus den Top Themen mehrerer Artikel je Zeitschrift, die Top zehn Tagesthemen gebildet. Auf diesem Weg wurden die Top zehn Tagesthemen je Kalendertag und Zeitschrift ermittelt. Die jeweiligen Top drei Tagesthemen wurden weiterführend zur späteren Verarbeitung im LSTM-Prototyp mittels One Hot Encoding kodiert und in binäre Daten umgewandelt. Dabei wurden doppelte Themen pro Tag und Zeitschrift ausgefiltert und entfernt. Nähere Details zur Datenerhebung und -aufbereitung können den Kapiteln 4.1 und 4.2 entnommen werden. Alle oberhalb geschilderten Inputvariablen für den Prognoseprototypen wurden miteinander über das Datum als Index verbunden und als Eingabevariablen in das Dow Jones Industrial Average Prognose-LSTM eingespeist. Für jenes Prognosemodell wurde nach Maßgabe vorhandener Literatur ein Ausgangsmodell zum Zeitpunkt t0 entwickelt, welches weiterführend mittels Prototyping-Verfahren evaluiert und verfeinert wurde, um die gegenständliche Forschungsfrage zu beantworten (Swathi et al., 2022). Das detaillierte Vorgehen während der Prototypingzyklen, sowie die Konzeption des Grundmodells zum Zeitpunkt t0, sowie aus dem Prototyping-Verfahren erlangte Erkenntnisse können in den Abschnitten 4.3 und 5 nachgelesen werden. Die

[19] Kaggle NYT Articles Dataset. Abgerufen 8. März 2024, von https://www.kaggle.com/datasets/aryansingh0909/nyt-articles-21m-2000-present

Schlussfolgerungen aus der gegenständlichen Untersuchung finden sich in den Folgekapiteln 5.1 und 5.2.

5.1 Schlussfolgerungen

Die Untersuchungsergebnisse der gegenständlichen Studie zeigen, dass es mit den zugrundeliegenden Eingabevariablen, bestehend aus den historischen Finanzmarktdaten des Dow Jones Industrial Average, extrahiert von der Finanzplattform www.ariva.de, sowie den Kleinanlegermeinungen, gewonnen mittels VADER-Sentimentanalyse nach Hutto und Gilbert aus den Kommentaren der Social Media Plattform www.reddit.com im Subreddit „Wallstreetbets“ (Hutto & Gilbert, 2014), sowie den anhand von Latent Dirichlet Allocation untersuchten Expertenmeinungen aus New York Times und Wall Street Journal (Blei et al., 2003) möglich ist, den Dow Jones Industrial Average präzise hervorzusagen. Es gelang, die aufgestellte Hypothese zu belegen, welche besagt, dass es anhand der oberhalb gelisteten Inputfaktoren möglich ist, eine Prognosegenauigkeit von über 60% bezogen auf den DJIA-Schlusskursanstieg oder -fall am Folgetag zu erzielen. Anhand der angewandten Datenaufbereitungsverfahren und der darauffolgenden Prototypingzyklen konnte mit dem Modell t63 ein Prognosemodell auf Basis eines LSTM entwickelt werden, welches den korrekten Kursanstieg oder -abfall im Testzeitraum von 18.10.2023 bis 31.12.2023 mit einer Prognosequote von 71.93% für den Folgetag vorhersagt. Die Arbeitsergebnisse zeigen jedoch, dass es mehrere Modelle gibt, die je nach gewähltem Prognosehorizont besser geeignet sind, als jenes mit der höchsten Prognosequote für den Folgetag. So weist zwar das t63-Modell die höchste Quote für den Folgetag auf, besitzt jedoch mit 91.43%. eine niedrigere Prognosequote für die Folgewoche als andere Modelle. Umgekehrt konnte mit dem Modell t59 ein Prognose-LSTM geschaffen werden, welches eine Prognosequote von 95.71% für die Folgewoche aufweist, jedoch lediglich eine PQ von 61.4% für den Folgetag. Es kann daher schlussgefolgert werden, dass es je nach Prognosehorizont und -ziel ratsam ist, auf unterschiedliche Modelle, unter Nutzung der gegenständlichen Inputfaktoren, zurückzugreifen. Generell wurde im Zuge des Prototyping-Verfahrens erkannt, dass sich die Prognosegenauigkeit bei Erhöhung des Prognosehorizonts, gemessen an der korrekten Vorhersage eines Kursanstiegs oder -abfalls des Tagesschlusskurses erhöht. Weiters kann erwähnt werden, dass der gewählte Adam-Optimierungsalgorithmus, sowie der mittlere quadratische Fehler als Verlustfunktion geeignet für das Training des LSTMs im gegenständlichen Kontext waren. Als Ergebnisse des Prototyping-Verfahrens aus Kapitel 5 zeigte sich zudem, dass sich die Wahl kürzerer Sequenzen maßgeblich positiv auf die angestrebte

Kurzfristprognose auswirkten, längere Sequenzen sich besser für Langfristprognosen eigneten. So konnte etwa mit dem Modell t12 mit einer geringen LSTM-Layer- und Neuronenanzahl (512-4-32) und einer Sequenzlänge von sieben Kalendertagen bereits eine Prognosequote von 92.86% für die Folgewoche erreicht werden, während die PQ für den Folgetag bei 56.14% lag. Dieser Effekt der Sequenzlängenwahl relativierte sich jedoch im gegenständlichen Setting bei steigender Neuronenanzahl in den einzelnen LSTM-Layern. Ebenfalls konnte beobachtet werden, dass ein zusätzlicher vierter LSTM-Layer keine signifikante Verbesserung der Prognosequoten und der Ergebnisse der Verlustfunktion herbeiführte. Im aktuellen Untersuchungsmodus konnte eingesehen werden, dass ein Anstieg der Neuronen bis zu einer Anzahl von 1024 Stück pro Layer zu Verbesserungen der Prognose führen kann, eine höhere Neuronananzahl pro Schicht jedoch keine weitere Optimierung mehr bringt. Deutliche Optimierungen des Prognosemodells wurden über die Reduktion der Lernrate des Optimierungsalgorithmus, sowie der Batchsize erzielt. Die optimalen Ergebnisse, nahezu aller Prognosemodelle wurden für die gegenständlichen Inputvariablen mit einer Batchsize von 16 Tagen und einer Lernrate von 0.0001 ermittelt. Außerdem zeigte sich, dass die Maßnahmen zur Vermeidung von Overfitting des LSTM mit Bedacht zu wählen sind, da im Ausgangsprototypen t0 der Eindämmungsmechanismus für Overfitting zu streng gewählt und der Trainingsprozess damit zu früh beendet wurde.

Zusammenfassend lässt sich festhalten, dass die gewählten Methoden zur Datenerhebung und -aufbereitung als Eingabevariablen für das Prognose-LSTM angemessen sind. Jene Eingabevariablen ermöglichen es, mithilfe eines durch Prototyping entwickelten Prognosemodells auf Basis eines LSTMs, die gegenständliche Hypothese zur korrekten Prognose des Dow Jones Industrial Average Aktienindextagesschlusskurs für den Folgetag mit einer Prognosequote von 71.93% zu bestätigen. Es zeigt sich, dass je größer der Prognosezeitraum gewählt wird, desto höher auch die Prognosequote steigt. Ebenfalls wurde erkannt, dass es anhand der gewählten Inputinformationen nicht, das eine beste Prognosemodell, unabhängig des Prognosehorizonts gibt, sondern, dass sich unterschiedliche Modelle je nach Prognosezeitraum unterschiedlich gut eignen. So weißt das präziseste Modell zur Vorhersage des korrekten Kursanstiegs oder -abfalls für den Folgetag eine PQ für den Folgetag von 71.93%, für die Folgewoche von 91.43%, bei einem MSE von 0.0046 auf, während das genaueste Modell für die Prognosequote für die Folgewoche eine PQ für den Folgetag von 61.40%, jedoch eine PQ für die Folgewoche von 95.71%, bei einem MSE von nur 0.0013 erreicht. Gegenständliches Modell zeichnet sich zudem durch eine hochpräzise Vorhersage der tatsächlichen Dow Jones Industrials

Tagesschlusskurswerte aus. Weiters zeigte sich, dass das Miteinbeziehen von mehr Finanzmarktinformationen als nur dem Aktienindextagesschlusskurs, die Prognosegenauigkeit weiter erhöht (Long, 2014). Abschließend kann schlussgefolgert werden, dass je nach gewünschtem Prognosehorizont differierende LSTM-Modelle einzusetzen sind, und es auf Basis der gewählten Eingabevariablen nicht möglich ist, das universelle Modell für sämtliche Vorhersageperspektiven zu ermitteln.

5.2 Ausblick

Auf Basis, der in den vorangegangenen Kapiteln erzielten Erkenntnisse, könnten sich mögliche zukünftige Arbeiten, mit der weiteren Vertiefung des Umstands der zunehmenden Genauigkeit des Prognose-LSTMs bei steigendem Prognosehorizont untersuchen. Weiterführend könnte analysiert werden, wie sich die zusätzliche Einbindung weiterer Social Media Quellen wie etwa X (Twitter) auf die Prognosegenauigkeit auswirken. So könnten Sentimentinformationen aus Tweets extrahiert werden, um sie ebenfalls, ähnlich dem aktuellen Social Media-Inputstream von Reddit, in das Prognosemodell einfließen zu lassen. Alternativ könnte selbiges für Plattformen des Meta-Konzerns durchgeführt werden. Eine weitere Optimierung des Modells durch tiefere Betrachtung der Feature Importance der Eingabevariablen scheint ebenfalls naheliegend. Vorhandene Literatur zeigte bereits im Jahr 2011, dass sich gerade X für eine solche Art von Prognose eignen kann (Bollen et al., 2011). Neben der VADER-Sentimentanalyse nach Hutto und Gilbert könnten zudem andere Sentimentlexika wie Pythons Textblob für die Beurteilung der Polaritäten in Social Media Postings herangezogen werden (Hutto & Gilbert, 2014). Auch auf Seiten der Finanzmarktnachrichten ist es möglich die Untersuchung weiter zu vertiefen, in dem mehr als nur drei Nachrichtenthemen pro Tag mittels One Hot Encoding für die spätere Prognose herangezogen werden. Die Nutzung anderer Zeitschriften wie etwa der Washington Post wäre ebenfalls denkbar. Neben der tagesaktuellen Prognose ist es zudem möglich, in zukünftigen Arbeiten auf Intraday-Kursveränderungen einzugehen um auf dieser Basis, den Prognoseprototypen weiter zu verfeinern. Auch die Einbindung von Aktienkursindikatoren ist eine weitere Möglichkeit zur Fortführung der Forschung mit dem gegenständlichen Modell.

7 Literaturverzeichnis

Acig, Bülent. (2001). Anwendung neuronaler Netze in der Finanzwirtschaft. Technische Universität Kaiserslautern.

Achkar, R., Elias-Sleiman, F., Ezzidine, H., & Haidar, N. (2018). Comparison of BPA-MLP and LSTM-RNN for Stocks Prediction. 2018 6th International Symposium on Computational and Business Intelligence (ISCBI), 48–51. https://doi.org/10.1109/ISCBI.2018.00019

Alex, B. (1998). Künstliche neuronale Netze in Management-Informationssystemen (Bd. 32). Gabler Verlag. https://doi.org/10.1007/978-3-322-91336-4

Allen, F., Haas, M., Nowak, E., & Tengulov, A. (2017). Market Efficiency and Limits to Arbitrage: Evidence from the Biggest Short Squeeze in History. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2977019

Althelaya, K. A., El-Alfy, E.-S. M., & Mohammed, S. (2018). Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU). 2018 21st Saudi Computer Society National Computer Conference (NCC), 1–7. https://doi.org/10.1109/NCG.2018.8593076

Anand, A., & Pathak, J. (2021). WallStreetBets Against Wall Street: The Role of Reddit in the GameStop Short Squeeze. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3873099

Ariva (2023). Ariva Dow Jones 2023. In Dow Jones Industrial Average 2023. Abgerufen am 14. Januar 2024. https://www.ariva.de/dow_jones_industrial_average-index/kurse/historische-kurse

Arvidsson, A. (2011). General Sentiment: How Value and Affect Converge in the Information Economy. The Sociological Review, 59, 39–59. https://doi.org/10.1111/j.1467-954X.2012.02052.x

Atkins, A., Niranjan, M., & Gerding, E. (2018). Financial news predicts stock market volatility better than close price. The Journal of Finance and Data Science, 4(2), 120–137. https://doi.org/10.1016/j.jfds.2018.02.002

Barrons (2023). Dow Jones Industrial Average Overview (DJIA) | Barron's. Abgerufen 29. Oktober 2023, von https://www.barrons.com/market-data/indexes/djia

Basiri, M. E., Naghsh-Nilchi, A. R., & Ghasem-Aghaee, N. (2014). Sentiment Prediction Based on Dempster-Shafer Theory of Evidence. Mathematical Problems in Engineering, 2014, 1–13. https://doi.org/10.1155/2014/361201

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., & Blackburn, J. (2020). The Pushshift Reddit Dataset. https://doi.org/10.48550/ARXIV.2001.08435

Baumgartner, J. M. (2024). Pushshift/api [Python]. https://github.com/pushshift/api (Original work published 2017)

Berning, L. (2016). Neuronale Netze in der Wertpapieranalyse. https://www.grin.com/document/334978

Blei, D. M., Ng, Andrew Y, & Jordan, Michael I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003).

Bloomberg (2023). Bloomberg and "The Magic" of Machine Learning | Press | Bloomberg LP. Bloomberg L.P. Abgerufen 19. Juni 2023, von https://www.bloomberg.com/company/press/bloomberg-magic-machine-learning/

Bollen, J., Mao, H., & Zeng, X.-J. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

Brody, S., & Diakopoulos, N. (2011). Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! Using Word Lengthening to Detect Sentiment in Microblogs.

Brown, J. H., Gillooly, J. F., Allen, A. P., Savage, V. M., & West, G. B. (2004). Toward a metabolic theory of ecology, 85(7), 1771–1789. https://doi.org/10.1890/03-9000

Burkov, A. (2019). Machine learning kompakt: Alles, was Sie wissen müssen (K. Lorenzen, Übers.; 1. Auflage). mitp.

Burnette, R. (2021). What Were the Factors that led to the GameStop Short Squeeze?

Chakraborty, P., Pria, U. S., Rony, Md. R. A. H., & Majumdar, M. A. (2017). Predicting stock movement using sentiment analysis of Twitter feed. 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), 1–6. https://doi.org/10.1109/ICIEV.2017.8338584

Chollet, F. (2021). Deep learning with Python (Second edition). Manning. ISBN: 9781617296864

Clark, K., & Manning, C. D. (2016). Deep Reinforcement Learning for Mention-Ranking Coreference Models (arXiv:1609.08667). arXiv. http://arxiv.org/abs/1609.08667

Da, Z., Engelberg, J., & Gao, P. (2011). In Search of Attention. The Journal of Finance, 66(5), 1461–1499. https://doi.org/10.1111/j.1540-6261.2011.01679.x

Dounis, Filippos. (2020). Elon's Golden Gift: Predicting The Stock Price Of Tesla With Twitter And Machine Learning.

Dowling, M. M., & Lucey, B. M. (2003). The Role of Feelings in Investor Decision-Making. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.346302

Engelberg, J. E., & Parsons, C. A. (2011). The Causal Impact of Media in Financial Markets. The Journal of Finance, 66(1), 67–97. https://doi.org/10.1111/j.1540-6261.2010.01626.x

Fabio, M. (2015.). Die Auswirkungen von Herdenverhalten auf Finanzmärkte. Wissenschaftliche Zuordnung und theoretische Grundlagen.

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, 25(2), 383. https://doi.org/10.2307/2325486

Füser, K. (1995). Neuronale Netze in der Finanzwirtschaft. Gabler Verlag. https://doi.org/10.1007/978-3-663-05964-6

García, B., Gallego, M., Gortázar, F., & Munoz-Organero, M. (2020). A Survey of the Selenium Ecosystem. Electronics, 9(7), 1067. https://doi.org/10.3390/electronics9071067

García-Méndez, S., De Arriba-Pérez, F., Barros-Vila, A., González-Castaño, F. J., & Costa-Montenegro, E. (2023). Automatic detection of relevant information, predictions and forecasts in financial news through topic modelling with Latent Dirichlet Allocation. Applied Intelligence, 53(16), 19610–19628. https://doi.org/10.1007/s10489-023-04452-4

Gidofalvi, G. (2001). Using News Articles to Predict Stock Price Movements. University of California, San Diego.

Gilbert, E., & Karahalios, K. (2010). Widespread Worry and the Stock Market. Proceedings of the International AAAI Conference on Web and Social Media, 4(1), 58–65. https://doi.org/10.1609/icwsm.v4i1.14023

Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision.

Grotian, K., & Beelich, K. H. (1999). Grundlagen des Lernens. In K. Grotian & K. H. Beelich, Lernen selbst managen (S. 17–38). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-98023-7_2

Gundecha, P., & Liu, H. (2012). Mining Social Media: A Brief Introduction. In 2012 TutORials in Operations Research (S. 1–17). INFORMS. https://doi.org/10.1287/educ.1120.0105

Gupta, R. K., Agarwalla, R., Naik, B. H., Evuri, J. R., Thapa, A., & Singh, T. D. (2022). Prediction of research trends using LDA based topic modeling. Global Transitions Proceedings, 3(1), 298–304. https://doi.org/10.1016/j.gltp.2022.03.015

Han, B., & Baldwin, T. (2011). Lexical Normalisation of Short Text Messages: Makn Sens a #twitter.

Hazarika, D., Konwar, G., Deb, S., & Bora, D. J. (2020). Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing. 63–67. https://doi.org/10.15439/2020KM20

Heinemann, M. (2004). DAX und Dow Jones. (2004). GRIN Verlag. ISBN 978-3-640-07125-8

Hisano, R., Sornette, D., Mizuno, T., Ohnishi, T., & Watanabe, T. (2013). High Quality Topic Extraction from Business News Explains Abnormal Financial Market Volatility. PLoS ONE, 8(6), e64846. https://doi.org/10.1371/journal.pone.0064846

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hüttner, M. (1986). Prognoseverfahren und ihre Anwendung. de Gruyter. ISBN 978-3-11-010826-2

Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. Proceedings of the International AAAI Conference on Web and Social Media, 8(1), 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

Hutto, C. (2024). Cjhutto/vaderSentiment [Python]. https://github.com/cjhutto/vaderSentiment (Original work published 2014)

Islam, T. (2019). Yoga-Veganism: Correlation Mining of Twitter Health Data. Department of Computer Science. Purdue University

Jabbar, H. K., & Khan, R. Z. (2014). Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study). Computer Science, Communication and Instrumentation Devices, 163–172. https://doi.org/10.3850/978-981-09-5247-1_017

Jiao, P., & Walther, A. (2016). Social Media, News Media and the Stock Market. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2755933

Karimi, Z. (2021). Scikit-learn-Quick-Review. https://doi.org/10.13140/RG.2.2.14605.67043

Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2022). Stock market prediction using machine learning classifiers and social media, news. Journal of Ambient Intelligence and Humanized Computing, 13(7), 3433–3456. https://doi.org/10.1007/s12652-020-01839-w

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/ARXIV.1412.6980

Kolasani, S. V., & Assaf, R. (2020). Predicting Stock Movement Using Sentiment Analysis of Twitter Feed with Neural Networks. Journal of Data Analysis and Information Processing, 8(4), Article 4. https://doi.org/10.4236/jdaip.2020.84018

Kolb, P. (2011). Was ist statistische maschinelle Übersetzung? Uinversität Pottsdam.

Kubiak, G. (1991). Vorhersage von Börsenkursen mit neuronalen Netzen. Universität Stuttgart, Institut für Parallele und Verteilte Höchstleistungsrechner.

Lanham, M. (2018). Learn ARCore: Fundamentals of Google ARCore : learn build augmented reality apps for Android, Unity, and the web with Google ARCore 1.0. Packt Publishing.

Lechelt, T. (1998). Aktienkursprognosen auf Basis Neuronaler Netzwerke. Wirtschaftswissenschaftliche Fakultät Martin-Luther-Universität Halle-Wittenberg.

Long, H. V. (2014). Dependence structure in financial time series: Applications and evidence from wavelet analysis. Victoria University of Wellington. https://www.researchgate.net/publication/303878901_Dependence_structure_in_financial_time_series_Applications_and_evidence_from_wavelet_analysis

Long, C., Lucey, B. M., & Yarovaya, L. (2021). „I Just Like the Stock“ versus „Fear and Loathing on Main Street“: The Role of Reddit Sentiment in the GameStop Short Squeeze. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3822315

Lyocsa S., Baumöhl E., & Vyrost, T. (2021). YOLO-Trading Riding with the herd during the Gamestop Episode. Econstor, 2021. http://hdl.handle.net/10419/230679

Makice, K. (2009). Twitter API: Up and running ; [learn how to build Twitter applications] (1. ed). O'Reilly.

Mayring, P. (2010). Qualitative Inhaltsanalyse. In G. Mey & K. Mruck (Hrsg.), Handbuch Qualitative Forschung in der Psychologie (S. 601–613). VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-531-92052-8_42

Mayring, P. (2014). Qualitative Content Analysis. Leibnitz Institut für Sozialwissenschaften

McKinney, W. (2011). pandas: A Foundational Python Library for Data Analysis and Statistics.

Motoyama, M., Levchenko, K., Kanich, C., McCoy, D., Voelker, G. M., & Savage, S. (2023). Re: CAPTCHAs – Understanding CAPTCHA-Solving Services in an Economic Context.

Nelson, D. M. Q., Pereira, A. C. M., & De Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. 2017 International Joint Conference on Neural Networks (IJCNN), 1419–1426. https://doi.org/10.1109/IJCNN.2017.7966019

Nofer, M., & Hinz, O. (2014). Are crowds on the internet wiser than experts? The case of a stock prediction community. Journal of Business Economics, 84(3), 303–338. https://doi.org/10.1007/s11573-014-0720-x

Nofsinger, J. R. (2005). Social Mood and Financial Economics. Journal of Behavioral Finance, 6(3), 144–160. https://doi.org/10.1207/s15427579jpfm0603_4

Oliveira, N., Cortez, P., & Areal, N. (2013). Some experiments on modeling stock market behavior using investor sentiment analysis and posting volume from Twitter. Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, 1–8. https://doi.org/10.1145/2479787.2479811

Pal, R., Kumar Shukla P., Banarasi, B., Lotfi, C., Srinivasan, S., Ertz, M., & Latrous, I. (2021). Web Scraping Techniques and Applications: A Literature Review. In Jaypee Institute of Information Technology, Noida, India, Das University, Lucknow, India, SCRS Conference Proceedings on Intelligent Systems (S. 381–394). Soft Computing Research Society. https://doi.org/10.52458/978-93-91842-08-6-38

Pan, Y. (2024). Different Types of Neural Networks and Applications: Evidence from Feedforward, Convolutional and Recurrent Neural Networks. Highlights in Science, Engineering and Technology, 85, 247–255. https://doi.org/10.54097/6rn1wd81

Piryani, R., Madhavi, D., & Singh, V. K. (2017). Analytical mapping of opinion mining and sentiment analysis research during 2000–2015. Information Processing & Management, 53(1), 122–150. https://doi.org/10.1016/j.ipm.2016.07.001

Qian, B & Rasheed, K. (2007). Stock Market Prediction with Multiple Classifiers. Applied Intelligence, 26(1), 25–33.

Ramsundar, B., & Zadeh, R. B. (2018). TensorFlow for deep learning: From linear regression to reinforcement learning (First edition). O'Reilly Media.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., & Mozetič, I. (2015). The Effects of Twitter Sentiment on Stock Price Returns. PLOS ONE, 10(9), e0138441. https://doi.org/10.1371/journal.pone.0138441

Röhner, J., & Schütz, A. (2020). Psychologie der Kommunikation. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-61338-2

Rokhsatyazdi, E., Rahnamayan, S., Amirinia, H., & Ahmed, S. (2020). Optimizing LSTM Based Network For Forecasting Stock Market. 2020 IEEE Congress on Evolutionary Computation (CEC), 1–7. https://doi.org/10.1109/CEC48606.2020.9185545

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Schniederjans, D., Cao, E. S., & Schniederjans, M. (2013). Enhancing financial performance with social media: An impression management perspective. Decision Support Systems, 55(4), 911–918. https://doi.org/10.1016/j.dss.2012.12.027

Schröter, J. (2021). Zusammenhang von Twitter-Stimmung und DAX: DAX-Vorhersage mit Twitter?

Selvi, A. A., & Arulchelvan, S. (2024). Decoding global reproductive health discourse on Reddit: themes, regions, and misinformation challenges. African journal of reproductive health, 28(1), 22–30. https://doi.org/10.29063/ajrh2024/v28i1.3

Shah, D., Campbell, W., & Zulkernine, F. H. (2018). A Comparative Study of LSTM and DNN for Stock Market Forecasting. 2018 IEEE International Conference on Big Data (Big Data), 4148–4155. https://doi.org/10.1109/BigData.2018.8622462

Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data, 7(1), 66. https://doi.org/10.1186/s40537-020-00333-6

Shilpa, S. (2023). Combined deep learning classifiers for stock market prediction: Integrating stock price and news sentiments. Kybernetes, 52(3), 748–773. https://doi.org/10.1108/K-06-2021-0457

Steinwendner, J., & Schwaiger, R. (2020). Neuronale Netze programmieren mit Python (2., aktualisierte und überarbeitete Auflage, 2., korrigierter Nachdruck). Rheinwerk Verlag.

Swathi, T., Kasiviswanath, N., & Rao, A. A. (2022). An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis. Applied Intelligence, 52(12), 13675–13688. https://doi.org/10.1007/s10489-022-03175-2

Tauchert, C., Buxmann, P., & Lambinus, J. (2020). Hawaii International Conference on System Sciences 2020, Crowdsourcing Data Science: A Qualitative Analysis of Organizations' Usage of Kaggle Competitions. ScholarSpace. https://hdl.handle.net/10125/63768

Tetzner, A., Kühne, T., Gluchowski, P., & Pfoh, M. (2021). Künstliche Neuronale Netze – Aufbau, Funktion und Nutzen. In D. Frick, A. Gadatsch, J. Kaufmann, B. Lankes, C. Quix, A. Schmidt, & U. Schmitz (Hrsg.), Data Science (S. 225–239). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-33403-1_14

Torres P., E. P., Hernández-Álvarez, M., Torres Hernández, E. A., & Yoo, S. G. (2019). Stock Market Data Prediction Using Machine Learning Techniques. In Á. Rocha, C. Ferrás, & M. Paredes (Hrsg.), Information Technology and Systems (Bd. 918, S. 539–547). Springer International Publishing. https://doi.org/10.1007/978-3-030-11890-7_52

Turing, A. M. (1950). Computing Machinery and Intelligence. Mind, New Series, 59(236), 433–460.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Vasileiou, E., Bartzou, E., & Tzanakis, P. (2021). Explaining Gamestop Short Squeeze using Intraday Data and Google Searches. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3805630

Vincent, A., & Armstrong, M. (2010). Predicting Break-Points in Trading Strategies with Twitter. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.1685150

Vogt, J. (2021). Vorhersage von Aktienkursbewegungen der Energiebranche mithilfe maschinellen Lernens und Stimmungserkennung von Beiträgen aus sozialen Medien. Hochschule Ruhr West University of Applied Sciences.

Wang, M., & Hu, F. (2021). The Application of NLTK Library for Python Natural Language Processing in Corpus Research. Theory and Practice in Language Studies, 11(9), 1041–1049. https://doi.org/10.17507/tpls.1109.09

Wang, L., Ling, Y., Yuan, Z., Shridhar, M., Bao, C., Qin, Y., Wang, B., Xu, H., & Wang, X. (2023). GenSim: Generating Robotic Simulation Tasks via Large Language Models. https://doi.org/10.48550/ARXIV.2310.01361

Watkins, B. (2003). Riding the Wave of Sentiment: An Analysis of Return Consistency as a Predictor of Future Returns. Journal of Behavioral Finance, 4(4), 191–200. https://doi.org/10.1207/s15427579jpfm0404_2

Wiesinger, S. (2021). Stock Market Forecasting mittels klassischen statistischen Verfahren und Text Mining. TU Wien.

Wu, D, Zheng, L, & Olson, D. L. (2014). Decision Support Approach for Online Stock Forum Sentiment Analysis. IEEE Transactions on Systems, Man, and Cybernetics: Systems.

Xing, F. Z., Cambria, E., & Welsch, R. E. (2018). Natural language based financial forecasting: A survey. Artificial Intelligence Review, 50(1), 49–73. https://doi.org/10.1007/s10462-017-9588-9

Yarovaya, L. (2021). Rethinking Financial Contagion: Information Transmission Mechanism during the COVID-19 pandemic.

Yu, L., Zhou, R., Chen, R., & Lai, K. K. (2022). Missing Data Preprocessing in Credit Classification: One-Hot Encoding or Imputation? Emerging Markets Finance and Trade, 58(2), 472–482. https://doi.org/10.1080/1540496X.2020.1825935

Yu, Y., Duan, W., & Cao, Q. (2013). The impact of social and conventional media on firm equity value: A sentiment analysis approach. Decision Support Systems, 55(4), 919–926. https://doi.org/10.1016/j.dss.2012.12.028

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis.

8 Abbildungsverzeichnis

9 Tabellenverzeichnis

Anhang A

A.1 Auszug Finanzmarktdaten

Datum	Erster	Hoch	Tief	Schlusskurs	Volumen	Intraday Gewinn Verlust	Intraday Gewinn	Intraday Verlust	Vortag Gewinn Verlust	Vortag Gewinn	Vortag Verlust
22.12.2023	37.349,27	37.533,99	37.276,40	37.385,97	254.544.015	36,7	36,7	0	-18,38	0	18,38
21.12.2023	37.225,32	37.409,01	37.129,14	37.404,35	258.235.722	179,03	179,03	0	322,35	322,35	0
20.12.2023	37.520,13	37.326,81	37.082	37.082	303.456.051	-438,13	0	438,13	-475,92	0	475,92
19.12.2023	37.311,82	37.557,98	37.311,82	37.557,92	277.625.458	246,1	246,1	0	251,9	251,9	0
18.12.2023	37.331,52	37.391,90	37.296,34	37.306,02	296.069.878	-25,5	0	25,5	0,86	0,86	0
17.12.2023	37.305,16	37.305,16	37.305,16	37.305,16	0	0	0	0	0	0	0
16.12.2023	37.305,16	37.305,16	37.305,16	37.305,16	0	0	0	0	0	0	0
15.12.2023	37.194,50	37.339,82	37.140,98	37.305,16	787.542.858	110,66	110,66	0	56,81	56,81	0
14.12.2023	37.115,57	37.256,12	37.194,93	37.248,35	458.101.547	132,78	132,78	0	158,11	158,11	0
13.12.2023	36.601,80	37.090,83	36.911,92	37.090,24	359.604.891	488,44	488,44	0	512,3	512,3	0
12.12.2023	36.442,10	36.590,74	36.377,79	36.577,94	292.769.414	135,84	135,84	0	173,01	173,01	0
11.12.2023	36.254,33	36.406,25	36.239,36	36.404,93	342.490.154	150,6	150,6	0	157,06	157,06	0