



**FOM Hochschule für Oekonomie & Management**

Hochschulzentrum DLS

**Projektarbeit**

im Studiengang Big Data & Business Analytics

über das Thema

**Deep Learning zur Aktienkursprognose mit multimodalen Daten**

von

Paul Hornig und Admir Dutovic

Dozent : M.Sc. Maher Hamid  
Matrikelnummer : 701650 und 733870  
Abgabedatum : 27. Februar 2025

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>II</b>
<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Codeverzeichnis</b>	<b>III</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Theoretische Grundlagen</b>	<b>2</b>
2.1 Datenverständnis . . . . .	2
2.2 Datenvorbereitung mit Pandas . . . . .	2
2.3 Modellierung mit Keras . . . . .	3
2.4 LSTM für Kursdaten . . . . .	3
2.5 Sprachmodell für Stimmungsdaten: Financial BERT . . . . .	4
2.6 CNN für Stimmungsdaten . . . . .	4
2.7 Evaluierungsmetriken . . . . .	5
<b>3 Methodik</b>	<b>6</b>
3.1 Datenbeschaffung . . . . .	6
3.2 Datenvorverarbeitung . . . . .	7
3.3 Modellierung . . . . .	12
3.3.1 Generelle Hyperparameter . . . . .	12
3.3.2 Basismodell . . . . .	13
3.3.3 Forschungsmodell . . . . .	14
3.4 Evaluierung . . . . .	16
3.4.1 Experiment 1: GOOG-Aktie . . . . .	16
3.4.2 Experiment 2: Technologie-Aktienmix . . . . .	19
3.4.3 Erkenntnisse . . . . .	21
3.5 Deployment . . . . .	22
<b>4 Fazit &amp; Ausblick</b>	<b>23</b>
4.1 Möglichkeiten . . . . .	23
4.2 Limitationen . . . . .	23
4.3 Ausblick . . . . .	24
<b>Literatur</b>	<b>25</b>
<b>Sonstige Quellen</b>	<b>27</b>

## Abbildungsverzeichnis

1	FinBERT: Beispielausgabe mit neutralem Ergebnis bei Eingabe von Tweets	4
2	Auszug aus GOOG-Kursdaten . . . . .	7
3	GOOG Tweet vom 01.01.2014 . . . . .	7
4	GOOG-Datenzeile mit vereinigten Kurs- und Tweetdaten . . . . .	8
5	GOOG-Datenzeile nach der Bereinigung von Tweets . . . . .	8
6	Resultierende GOOG-Datenzeile im Datenkorpus nach Datenvorverarbeitung . . . . .	11
7	Basismodell-Architektur mit Unterscheidung zwischen Experiment 1 und 2	13
8	Forschungsmodell-Architektur mit Unterscheidung zwischen Experiment 1 und 2 . . . . .	15
9	Experiment 1: Verlust-Vergleich von Basis- und Forschungsmodell, berechnet anhand von Validierungsdaten der GOOG-Aktie . . . . .	16
10	Experiment 1: Basismodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen . . . . .	17
11	Experiment 1: Forschungsmodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen . . . . .	18
12	Experiment 2: Verlustvergleich von Basis- und Forschungsmodell, berechnet anhand der Validierungsdaten von Technologieaktien . . . . .	19
13	Experiment 2: Basismodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen . . . . .	20
14	Experiment 2: Forschungsmodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen . . . . .	20

## Abkürzungsverzeichnis

<b>CNN</b>	Convolutional Neural Network
<b>CSV</b>	Comma-Separated Values
<b>CRISP-DM</b>	Cross Industry Standard Process for Data Mining
<b>DL</b>	Deep Learning
<b>ELU</b>	Exponential Linear Unit
<b>LSTM</b>	Long Short Term Memory
<b>MAE</b>	Mean Absolute Error
<b>MAPE</b>	Mean Absolute Percentage Error
<b>MSE</b>	Mean Squared Error
<b>ReLU</b>	Rectified Linear Unit
<b>RNN</b>	Recurrent Neural Network

## Codeverzeichnis

1	Relevanz quantifiziert als Anzahl-Tweets . . . . .	9
2	Stimmung quantifiziert als Positiv/Negativ-Score . . . . .	9
3	Umwandlung von Tweets zu Token-IDs . . . . .	10

# 1 Einleitung

Bereits seit Jahrzehnten wird dem Bereich Aktienprognose von Wissenschaftlern wie auch Investoren große Aufmerksamkeit gewidmet [1, Kap. Introduction]. Das liegt vor allem daran, dass man mit korrekten Vorhersagen sehr hohe Profite erreichen kann.

Die bisher durchgeführte Forschung hat ergeben, dass numerische Aktien-Daten allein lediglich bis zu einem gewissen Grad zur Verbesserung der Leistung von Deep Learning (DL) Modellen beitragen [1, Kap. Introduction].

Diese Arbeit widmet sich daher der Untersuchung, inwieweit Aktienprognosen durch Betrachtung von multimodalen Daten verbessert werden können. Durch den aktuellen technischen Fortschritt gibt es viele Möglichkeiten, nicht-numerische Daten einzubinden. Diese Arbeit fokussiert sich auf die Erprobung von vortrainierten Sprachmodellen, mit deren Hilfe Stimmungsdaten erzeugt werden sollen. Ein weiterer Schwerpunkt ist die Ermittlung einer geeigneten DL-Architektur, welche mit Numerik- und Stimmungsdaten trainiert wird.

Die Strukturierung dieser Arbeit orientiert sich am Cross Industry Standard Process for Data Mining (CRISP-DM) Modell. In Kapitel 2 werden zunächst wichtige theoretische Grundlagen behandelt. Anschließend erfolgt eine Beschreibung der praktischen Umsetzung mit- samt Evaluierung. Auf den umfangreichen Arbeitsschritt Deployment wird nur kurz eingegangen, indem wichtige Aspekte im Zusammenhang mit Aktienprognosen beleuchtet werden. Zum Schluss erfolgt eine zusammenfassende Analyse der Ergebnisse in Form eines Fazits und ein Ausblick auf zukünftige Arbeiten.

Das CRISP-DM-Modell erfüllt den Anspruch an Struktur und Vollständigkeit, wobei vor allem das enthaltene iterative Konzept im vorliegenden Anwendungsfall Vorteile mit sich bringt. Denn falls möglich, soll bei unzureichenden Ergebnissen der Prozessanfang bis -ende auf Verbesserungsmöglichkeiten untersucht werden.

## 2 Theoretische Grundlagen

Die Analyse von Aktienkursen mithilfe von DL kombiniert historische Kursdaten mit externen Einflussfaktoren wie Stimmungsanalysen. Dabei werden verschiedene Algorithmen genutzt, um Muster in den Daten zu erkennen und Prognosen zu erstellen. Besonders tiefe neuronale Netze wie Long Short Term Memory (LSTM) oder Convolutional Neural Network (CNN) eignen sich zur Modellierung von Zeitreihen und Textdaten [2][3].

### 2.1 Datenverständnis

Das StockNet-Dataset wurde für die Forschung zur Aktienkursprognose entwickelt und kombiniert historische Aktienkurse mit Twitter-Daten. Es wurde 2017 von Yumo Xu und Shay B. Cohen erstellt und ist unter der MIT-Lizenz öffentlich verfügbar. Das Dataset ermöglicht Analysen an der Schnittstelle zwischen Finanzmärkten und sozialer Medien.

Das Dataset besteht aus zwei Hauptkomponenten:

- **Historische Aktienkursdaten:** Enthalten tägliche Kursbewegungen wichtiger Unternehmen.
- **Twitter-Sentiment-Daten:** Enthalten Tweets basierend auf Finanzereignissen.

Das Portfolio umfasst verschiedene Unternehmen aus der Technologiebranche, darunter Apple (AAPL), Amazon (AMZN), Cisco (CSCO), Meta (FB/META) und Microsoft (MSFT) [4][5].

### 2.2 Datenvorbereitung mit Pandas

Eine sorgfältige Datenvorbereitung ist entscheidend für die Qualität von Vorhersagemodellen. In diesem Projekt werden Aktienkurs- und Tweetdaten aus verschiedenen Dateien verarbeitet, um eine strukturierte Eingabe für DL-Modelle zu erstellen.

Die Bibliothek `pandas` ist ein essenzielles Werkzeug für die Datenverarbeitung und -analyse in Python. Sie ermöglicht eine effiziente Handhabung von strukturierten Daten durch leistungsstarke Funktionen zur Bereinigung, Transformation und Aggregation.

Ein zentraler Schritt in der Datenvorbereitung ist das Einlesen und Verarbeiten von Aktienkurs- und Tweetdaten. Mit `pandas` können Comma-Separated Values (CSV)-Dateien direkt geladen und als `DataFrame` strukturiert werden. Die Spalten können gefiltert, umbenannt und mit Methoden wie `dropna()` von fehlenden Werten bereinigt werden. Zudem bietet `pandas` leistungsstarke Funktionen wie `groupby()` zur Aggregation von Daten und `merge()`, um verschiedene Datensätze miteinander zu verknüpfen.

Zeitreihenanalysen profitieren von der Möglichkeit, Datumswerte mit `to_datetime()` zu konvertieren und die Daten durch `resample()` in gleichmäßige Intervalle zu unterteilen. Diese Methoden sind essenziell, um eine konsistente und gut strukturierte Eingabe für DL-Modelle zu gewährleisten [6].

## 2.3 Modellierung mit Keras

Keras ist eine Open-Source-Deep-Learning-API, die auf TensorFlow basiert und eine benutzerfreundliche Schnittstelle zur Modellierung neuronaler Netze bietet. Für die Aktienkursprognose ermöglicht Keras die einfache Implementierung von LSTM-Netzwerken, die speziell für Zeitreihendaten geeignet sind. Durch die Verwendung von `Sequential`-Modellen können mehrere Elemente, wie LSTMs, Dropout zur Vermeidung von Overfitting und Dense-Schichten zur Vorhersage, kombiniert werden. Zudem erleichtert Keras die Hyperparameteroptimierung und das Modelltraining mit GPUs, was die Effizienz steigert. Die flexible API erlaubt es, verschiedene Architekturen schnell zu testen und mit vortrainierten Modellen zu arbeiten [7].

## 2.4 LSTM für Kursdaten

LSTM-Netzwerke sind eine spezielle Form von Recurrent Neural Networks (RNNs), die langfristige Abhängigkeiten in Zeitreihen erfassen können. Aufgrund ihrer Fähigkeit, vergangene Preisbewegungen zu speichern, werden sie häufig für Aktienkursprognosen verwendet. LSTM-Modelle bestehen aus Speicherzellen mit Eingangs-, Ausgabe- und Vergessensgattern, die den Informationsfluss regulieren. Dies verhindert das Problem des Vanishing Gradients und ermöglicht eine robuste Vorhersage von Trends in Finanzdaten. In Kombination mit Techniken wie Dropout und optimierten Aktivierungsfunktionen können LSTMs effektiv für die Aktienkursanalyse genutzt werden [3][8][9].

## 2.5 Sprachmodell für Stimmungsdaten: Financial BERT

Die Verwendung von Language Models, insbesondere Financial BERT, hat sich als wertvolles Werkzeug zur Analyse von Stimmungsdaten im Finanzsektor erwiesen. Financial BERT ist ein auf der Transformer-Architektur basierendes Modell, das speziell für die Verarbeitung und das Verständnis von Finanztexten trainiert wurde. Es kann genutzt werden, um die Stimmung in Kundenkommunikationen, Nachrichtenartikeln und sozialen Medien zu erfassen, was die Vorhersage von Kursbewegungen unterstützen kann.

Durch die Analyse von Stimmungsdaten können wertvolle Einblicke in das Verhalten und die Meinungen von Stakeholdern gewonnen werden. Positive Stimmungen oder Trends können auf potenzielle Kursanstiege hinweisen, während negative Stimmungen auf einen Kursabfall hindeuten können. Financial BERT ermöglicht es, große Mengen unstrukturierter Textdaten effizient zu verarbeiten und Stimmungsanalysen in Echtzeit durchzuführen.

**Abbildung 1:** FinBERT: Beispielausgabe mit neutralem Ergebnis bei Eingabe von Tweets

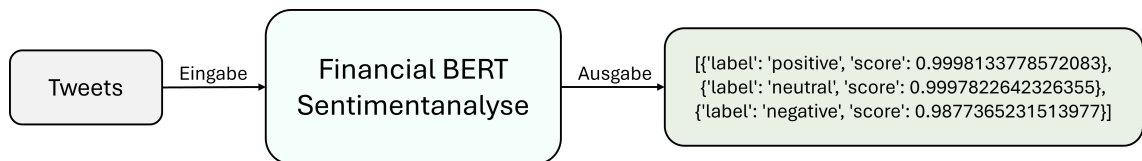


Abbildung 1 zeigt anhand eines Beispiels, wie eine Ausgabe des Modells bei der Eingabe von Tweets aussehen kann. Bei der Quantifizierung der Stimmung wird zwischen den Klassen `positive`, `neutral` und `negative` unterschieden. Diese Informationen können genutzt werden, um die Stimmungslage in Bezug auf bestimmte Unternehmen oder Branchen zu analysieren und in die Aktienkursprognose einzubeziehen. Zusätzlich zur Ausgabe von quantifizierten Stimmungsklassen lassen sich auch Embeddings ausgeben, deren tiefere Semantik ein Stimmungsbild über die Texteingabe darstellt [10][11][12].

## 2.6 CNN für Stimmungsdaten

CNNs sind nicht nur für die Bildverarbeitung geeignet, sondern haben sich auch als effektiv für die Analyse von Textembeddings erwiesen. Durch die Anwendung von Faltungsoperationen können CNNs lokale Muster und Beziehungen in vorverarbeiteten Textdaten erfassen. Diese Modelle sind besonders nützlich für die Sentiment-Analyse, da sie kontextuelle Informationen extrahieren und komplexe Muster in Zeiträumen identifizieren, die zur Klassifizierung von Stimmungen in Kundenfeedback und sozialen Medien beitragen.



Zudem ermöglichen CNNs eine parallele Verarbeitung von Daten, was die Effizienz erhöht und schnellere Trainingszeiten fördert, wodurch sie sich ideal für große Datensätze im Finanzbereich eignen [13][14].

## 2.7 Evaluierungsmetriken

Die Evaluierung von Modellen zur Aktienprognose erfordert den Einsatz geeigneter Metriken, um die Genauigkeit und Zuverlässigkeit der Vorhersagen zu bewerten. Zu den häufig verwendeten Metriken gehören der Mean Squared Error (MSE), der Mean Absolute Error (MAE) und der Mean Absolute Percentage Error (MAPE).

Der MSE misst die durchschnittlichen quadratischen Abweichungen zwischen den vorhergesagten und tatsächlichen Werten und wird durch die folgende Formel dargestellt:

**Formel 1:** MSE. Quelle: [15]

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

Der MAE hingegen berechnet die durchschnittliche absolute Abweichung:

**Formel 2:** MAE. Quelle: [16]

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

Der MAPE misst den Fehler relativ zu den tatsächlichen Werten und gibt so die Ungenauigkeit der Vorhersagen in Prozent an:

**Formel 3:** MAPE. Quelle: [17]

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (3)$$

Diese Metriken bieten wertvolle Einblicke in die Leistung von Modellen und helfen dabei, Optimierungsstrategien zur Verbesserung der Vorhersagegenauigkeit zu entwickeln [18][19].

## **3 Methodik**

### **3.1 Datenbeschaffung**

Aufgrund dessen, dass Aktienprognosen mit multimodalen Daten ein sehr belebtes Forschungsfeld sind, gibt es bereits viele sinnvoll zusammengestellte Datensätze. Um diesen Vorteil auszuschöpfen, wird ein anerkannter Finanzdatensatz gewählt, welcher in mehreren Wissenschaftsarbeiten zum Einsatz kam (Kap. 2.1)[4][20][1].

Der Datensatz umfasst Kursdaten und Tweets zu 88 Aktien, wobei jeweils eine Rohfassung und eine vorverarbeitete Variante bereitgestellt wird. In dieser Arbeit wird in beiden Fällen die Rohfassung verwendet.

Vom Initial-Beschaffer wird angegeben, dass sich die Daten auf den Zeitraum 01.01.2014 bis 01.01.2016 beziehen [4, Kap. 3]. Es gibt jedoch auch Abweichungen, wie im Fall der Aktie "BABA", bei der Kursdaten für den Bereich 19.09.2014 bis 09.01.2017 vorliegen.

Im Rahmen dieser Arbeit werden lediglich Aktien der Branche Technologie einbezogen, welche im entsprechenden Zeitraum unter den Top-10-Aktien nach Handelsvolumen vorkommen.

#### **Kursdaten**

Zu jeder Aktie liegt eine CSV-Datei vor. In Abbildung 2 werden die inkludierten Metriken dargestellt.

**Abbildung 2:** Auszug aus GOOG-Kursdaten. Quelle: [21]

```

1 Date,Open,High,Low,Close,Adj Close,Volume
2 2013-12-31,554.043335,558.406982,551.064453,558.262512,558.262512,2725900
3 2014-01-02,555.647278,556.788025,552.060730,554.481689,554.481689,3656400
4 2014-01-03,555.418152,556.379578,550.401978,550.436829,550.436829,3345800
5 2014-01-06,554.426880,557.340942,551.154114,556.573853,556.573853,3551800
6 2014-01-07,560.399475,567.717041,558.486633,567.303589,567.303589,5124300

```

## Tweets

Zu jeder Aktie liegt für jeden Tag im jeweiligen Zeitraum eine Datei mit einer Tweet-Liste vor. Abbildung 3 stellt einen Eintrag dieser Liste dar, wobei lediglich relevante Attribute einbezogen sind.

**Abbildung 3:** GOOG Tweet vom 01.01.2014. Quelle: [21]

```

{"created_at":"Wed Jan 01 03:59:03 +0000 2014", "id":418229860210057216,
"text":"RT @philstockworld: Summary of Yesterday's Webcast Featuring $AAPL $WYNN
$GOOG $LGF #TraderEducation #Options #HedgingStrategies -- http://\u2026"}

```

## 3.2 Datenvorverarbeitung

Die Datenvorverarbeitung erfolgt logisch in 4 Teilschritten und wird mit der Python-Bibliothek `pandas` durchgeführt (Kap. 2.2). Während des 1. Arbeitsschritts werden die Daten aller Aktien in einer CSV-Datei zusammengefasst, wobei zur Unterscheidung eine zusätzliche Datenspalte mit Marktkürzel als Füllwerte hinzugefügt wird (Abb. 4, Spalte "stock").

### 1. Zeitliche Ausrichtung

In diesem Arbeitsschritt werden pro Aktie Kursdaten und Tweets zu einer CSV-Datei zusammengefasst. Dabei werden zunächst alle Kursdaten-Einträge entfernt, deren Erstellungsdatum außerhalb des Zeitraums liegt, für den Tweets vorliegen. Da diese Abweichung auch entgegengesetzt auftreten kann, werden auch alle Kurznachrichten gelöscht, deren Erstellungsdatum sich außerhalb des Zeitraums der Kursdaten befindet. Abbildung 4 zeigt eine resultierende Datenzeile.

**Abbildung 4:** GOOG-Datenzeile mit vereinigten Kurs- und Tweetdaten

```

1 date,open,high,low,close,adj close,volume,tweets,stock
2 2014-01-06,554.42688,557.340942,551.154114,556.573853,556.573853,3551800,"['Google Inc :
  Charleston SEO Company Matthew Rubin Marketing Services ... $GOOG http://t.co/JgvTg01bHJ',
  ""2013's Best Smartphone\nhttp://t.co/ImopbnMjc6 $APPLE $GOOG $FB $TWTR #Samsung"",
  ""@abnormalreturns: Sunday links: reflexive buybacks, economic optimism & short vs
  long-term thinking http://t.co/4pMVcwHOkJ $AMZN $GOOG $FB"", 'Watch out world, Google
  does it again http://t.co/41MlxOacDY #aviateEUROPE $ARM $CSR $GOOG', ...]",GOOG

```

Alle Tweets, die außerhalb von Markttagen entstanden sind, werden dem nächstmöglichen Markttag zugeordnet. Somit enthält die Zeile aus Abbildung 4 auch Kurznachrichten vom 04.01 und 05.01.2014.

## 2. Textbereinigung

Um die Tweets für eine Weiterverarbeitung durch ein Sprachmodell vorzubereiten, werden mehrere Bereinigungen durchgeführt. Dabei wird darauf geachtet, dass keinerlei semantische Informationen verloren gehen. Stopp-Wörter wie beispielsweise "und" werden daher nicht entfernt. In Abbildung 4 ist zu erkennen, dass in Tweets häufig URLs eingebunden sind. Diese werden mit "URL" maskiert. Des Weiteren treten regelmäßig Referenzierungen mit Hilfe von "@" auf. Diese stellen einen Verweis auf Entitäten wie beispielsweise Personen oder Unternehmen dar und werden mit der Maskierung "AT\_ENTITY" ersetzt. Außerdem erfolgt eine Substituierung von Aktienvorkommen der Form "\$GOOG" mit "GOOG stock". Denn es wird angenommen, dass für ein Sprachmodell die Semantik des Zeichens "\$" in diesem Zusammenhang weniger klar ist als die des Begriffs "stock". In Abbildung 5 ist die bereinigte Form der Tweets-Spalte aus Abbildung 4 zu sehen.

**Abbildung 5:** GOOG-Datenzeile nach der Bereinigung von Tweets

```

1 date,open,high,low,close,adj close,volume,tweets,stock
2 2014-01-06,554.42688,557.340942,551.154114,556.573853,556.573853,3551800,"['Google Inc :
  Charleston SEO Company Matthew Rubin Marketing Services ... GOOG stock URL', ""2013's
  Best Smartphone URL APPLE stock GOOG stock FB stock TWTR stock #Samsung"", ""AT_ENTITY:
  Sunday links: reflexive buybacks, economic optimism & short vs long-term thinking URL
  AMZN stock GOOG stock FB stock"", 'Watch out world, Google does it again URL
  #aviateEUROPE ARM stock CSR stock GOOG stock', ...]",GOOG

```

## 3. Stimmungsmetriken

Dieser Arbeitsschritt sieht vor, dass pro Datenzeile anhand der enthaltenen Tweet-Liste Stimmungsmetriken extrahiert werden. Zunächst erfolgt eine Ableitung der Relevanz einer Aktie pro Tag, quantifiziert als Anzahl veröffentlichter Tweets (Code 1).

**Code 1: Relevanz quantifiziert als Anzahl-Tweets**

```
1 df['num_tweets'] = df['tweets'].apply(len)
```

Anschließend wird für jede Tweet-Liste eine durchschnittliche Quantifizierung der Metriken "positiv" und "negative" berechnet. Hierfür wird das vortrainierte Sprachmodell "Financial-BERT" eingesetzt (Kap. 2.5). Die genaue Funktionsweise ist im Code 2 beschrieben.

**Code 2: Stimmung quantifiziert als Positiv/Negativ-Score**

```
1 # Initialisierung des Sprachmodells
2 model = BertForSequenceClassification.from_pretrained("<Pfad>")
3 tokenizer = BertTokenizer.from_pretrained("<Pfad>")
4
5 # Initialisierung der Klassifikations-Pipeline
6 nlp = pipeline("sentiment-analysis", model=model, tokenizer=tokenizer)
7
8 # Funktion zur Extraktion von Sentimentmetriken über eine Tweet-Liste
9 def comp_sent(texts):
10     sent_res = []
11     for text in texts:
12         res = nlp(text)
13         # Klassifikationsergebnis für einen Text
14         # Format: [{'label': <Wert>, 'score': <Wert>}]
15         sentiment = res[0]
16
17         # Umwandlung zu Format: [positive_score, negative_score]
18         res_formatted = [float(sentiment['label'] == 'positive') *
19 sentiment['score'],
20 float(sentiment['label'] == 'negative') * sentiment['score']]
21         # Hinzufügen des Resultats zur Liste
22         sent_res.append(res_formatted)
23
24     # Berechnung des Durchschnitts der "Tuple"
25     res = list(np.mean(sent_res, axis=0)) if len(sent_res) > 0 else [0,
26 0]
27
28     # Rückgabe des Durchschnitts über alle Tweets an diesem Tag als
29     Dictionary
30     return {
31         'positive': res[0],
32         'negative': res[1]
33     }
```

Das Ergebnis dieser Operationen wird strukturiert als Spalten `num_tweets`, `positive` und `negative` in den Datenkorpus eingefügt (Abb. 6).

#### 4. Stimmung-Embeddings

Zusätzlich zu Stimmungsmetriken soll pro Tweet-Liste ein repräsentativer Embedding-Vektor erzeugt werden. Es wird angenommen, dass dadurch für ein neuronales Netz detaillierte Informationen bezüglich Tagesstimmung bereitgestellt sind. Hierfür werden pro Tag alle gesammelten Tweets einer Aktie verbunden und anschließend zur Berechnung des Embeddings an das FinancialBERT-Modell übergeben. Dazu werden zunächst alle Tweets zu einem Text zusammengefasst, wobei das Sonderzeichen "[SEP]" als Bindeglied dient. Anschließend wird dem String das Zeichen "[CLS]" vorangestellt. Dies ist eine, seitens des zugrundeliegenden "BERT"-Modells, empfohlene Vorgehensweise für Klassifikationsaufgaben, weshalb das Folgemodell "FinancialBERT" auch mit dieser Methode optimiert wurde [22, Kap. 5.3]. Anschließend wird der kombinierte Text zu einer Token-ID-Liste umgewandelt (Code 3).

**Code 3:** Umwandlung von Tweets zu Token-IDs

```
1 # Verbinden der Tweets mit Special-Token zu einem Text
2 text = ' [SEP] '.join(tweet_list)
3 # Klassifikations-Sonderzeichen voranstellen
4 text = ' [CLS] ' + text
5
6 # Text zu Token-IDs umwandeln
7 inputs = tokenizer(
8 text,
9 padding=True,
10 truncation=True,
11 return_tensors="pt"
12 )
```

Die Argumente `padding` und `truncation` in Zeile 9-10 von Code 3 stellen sicher, dass falls das Resultat die maximale Kontextlänge von 512 Token unter- respektive überschreitet, entweder mit einem Sonderzeichen aufgefüllt wird oder ein entsprechendes Zuschneiden auf 512 Token-IDs erfolgt. Im letzten Schritt werden die Tokens an das Sprachmodell übergeben, welches mit berechneten Embeddings antwortet. Während des iterativen CRISP-DM-Prozesses hat sich herausgestellt, dass ein Zuschneiden der IDs bei Längenüberschreitung für eingesetzte neuronale Netzwerke (Kap. 3.5) besser funktioniert als das Aufteilen in Chunks mit anschließender Durchschnittsberechnung der Teil-Embeddings.

## Ergebnis

Der resultierende Datenkorpus ist im CSV-Format gespeichert und enthält eine Kombination aus Kurs- und Stimmungsdaten von 10 Aktien verteilt auf 5470 Zeilen (6).

**Abbildung 6:** Resultierende GOOG-Datenzeile im Datenkorpus nach Datenvorverarbeitung

```
1 date,open,high,low,close,adj close,volume,stock,num_tweets,positive,negative,tweet_embs
2 2014-01-06,554.42688,557.340942,551.154114,556.573853,556.573853,3551800,GOOG,9,0.2161561581823561,
  0.1096002194616529,"[-1.6030482053756714, 0.28848686814308167, -0.37600037455558777, 0.
  3213549852371216, 1.8825640678405762, 2.0237033367156982, 0.075238898396492, 1.0776742696762085,
  -2.66947603225708, 0.4654025435447693, -1.4391454458236694, -2.0213849544525146, ...]"
```

### 3.3 Modellierung

In dieser Arbeit geht es darum, den Einfluss von Stimmungsdaten auf Aktienprognosen zu untersuchen. Da der Fokus auf den Sentiment-Daten gelegt ist, wird von den Kursdaten lediglich der Schlusspreis einbezogen. Als Basismodell dient also ein LSTM-basiertes Netzwerk, welches Schlusspreise als Eingabe verarbeitet. Das Forschungsmodell nimmt eine Kombination aus Schlusspreis und Stimmungsdaten entgegen.

Insgesamt werden 2 Experimente durchgeführt.

**Experiment 1:** Diese Untersuchung analysiert den Leistungsunterschied zwischen Basis- und Forschungsmodell basierend auf Trainings mit Daten der GOOG-Aktie.

**Experiment 2:** Bei dieser Analyse werden erneut beide Modelle verglichen, jedoch werden für Trainings die Daten aller Aktien verwendet.

In den nachfolgenden Abschnitten wird zunächst auf modellübergreifende Hyperparameter eingegangen und anschließend erfolgt eine Beschreibung der Architektur von Basis- und Forschungsmodell.

#### 3.3.1 Generelle Hyperparameter

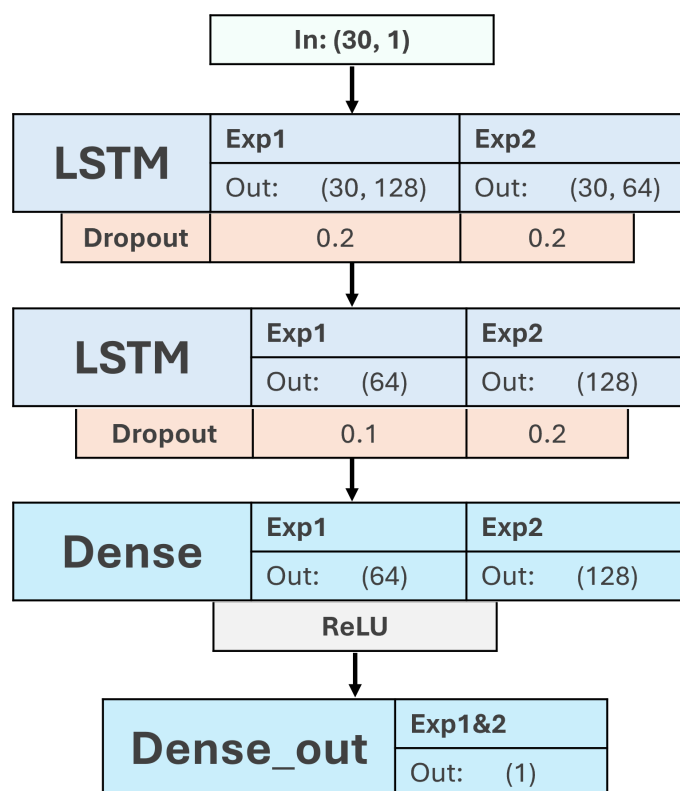
Es gibt eine Vielzahl an Hyperparametern, welche sowohl für das Basismodell als auch für das Forschungsmodell gelten. In beiden Fällen erfolgt eine Aufteilung der vorverarbeiteten Daten (Kap. 3.2) in Trainings-, Validierungs- und Testdaten. Dabei wird 20% der Gesamtmenge für Tests separiert und von den restlichen 80% werden 15% zur Validierung verwendet. Erst nach diesem Schritt erfolgt die MinMax-Skalierung der Trainingsdaten mit Ausnahme der Embeddings auf das Intervall 0 bis 1, mit Hilfe der Python-Bibliothek `sklearn` [23]. Anschließend wird der gelernte Scaler auf Test- und Validierungsdaten angewandt. So wird eine unerwünschte Übertragung von Informationen aus Validierungs- und Testdaten in den Trainingskorpus verhindert. Für Kursdaten wird ein Fenster von 30 vergangenen Markttagen als Modelleingabe verwendet, um den Preis eines nachfolgenden Tages vorherzusagen. Dieser Wert wurde in Experimenten einer Forschungsarbeit an der Universität Shaoguan in China als Optimum ermittelt [24, Tabelle 3]. Als Verlustfunktion dient typisch für Regressionsprobleme der MSE, welcher sich aus Modellausgabe ( $\hat{y}_i$ ) und tatsächlichem Preis ( $y_i$ ) ergibt (Kap. 2.7, Formel 1). Die Batch-Größe während des Trainings beträgt 64 und als Lernalgorithmus wird Adam-Optimierung eingesetzt [25]. Beide Modelle werden 50 Epochen lang trainiert, wobei immer dann eine Zwischenspeicherung erfolgt, sobald ein neuer Bestwert für die Verlustmetrik, basierend auf Validierungsdaten, erreicht wird.



### 3.3.2 Basismodell

Das Basismodell nimmt als Eingabe den Schlusspreis von 30 vorangegangenen Markttagen entgegen. Um die Charakteristik der Zeitreihendaten bestmöglich zu modellieren, wurde LSTM als Hauptbestandteil des Netzwerks gewählt (Kap. 2.4). So besteht das Modell zum einen aus LSTM- und vollständig verbundenen Schichten (Abb. 7).

**Abbildung 7:** Basismodell-Architektur mit Unterscheidung zwischen Experiment 1 und 2



Nachdem die Eingabe eine erste LSTM-Ebene durchlaufen hat, folgt eine Dropout-Einheit. Dies soll für eine bessere Generalisierbarkeit des Modells sorgen. In Anlehnung an bereits existierende Forschungsarbeiten folgt eine zweite LSTM-Schicht [26, Kap. 4.2.1]. Auch auf ihre Ausgabe wird eine Dropout-Operation durchgeführt. Anschließend werden die Daten an eine dichte Schicht übergeben. Ihr Zweck ist es, die Tiefe des Modells und somit dessen Komplexität zu erhöhen, um den komplexen Mustern in den Trainingsdaten gerecht zu werden. Damit dabei nicht nur lineare, sondern auch nichtlineare Merkmale modelliert werden können, folgt darauf die Aktivierungsfunktion Rectified Linear Unit (ReLU). Das Ergebnis dieser Operation wird abschließend an eine vollständig verbundene Ausgangsschicht übergeben. Deren Ausgabe stellt als Vorhersage den Schlusspreis des Folgetages (Tag 31) in skaliert Form dar. Für die Ermittlung von Dropout-Rate, Neuronenanzahl

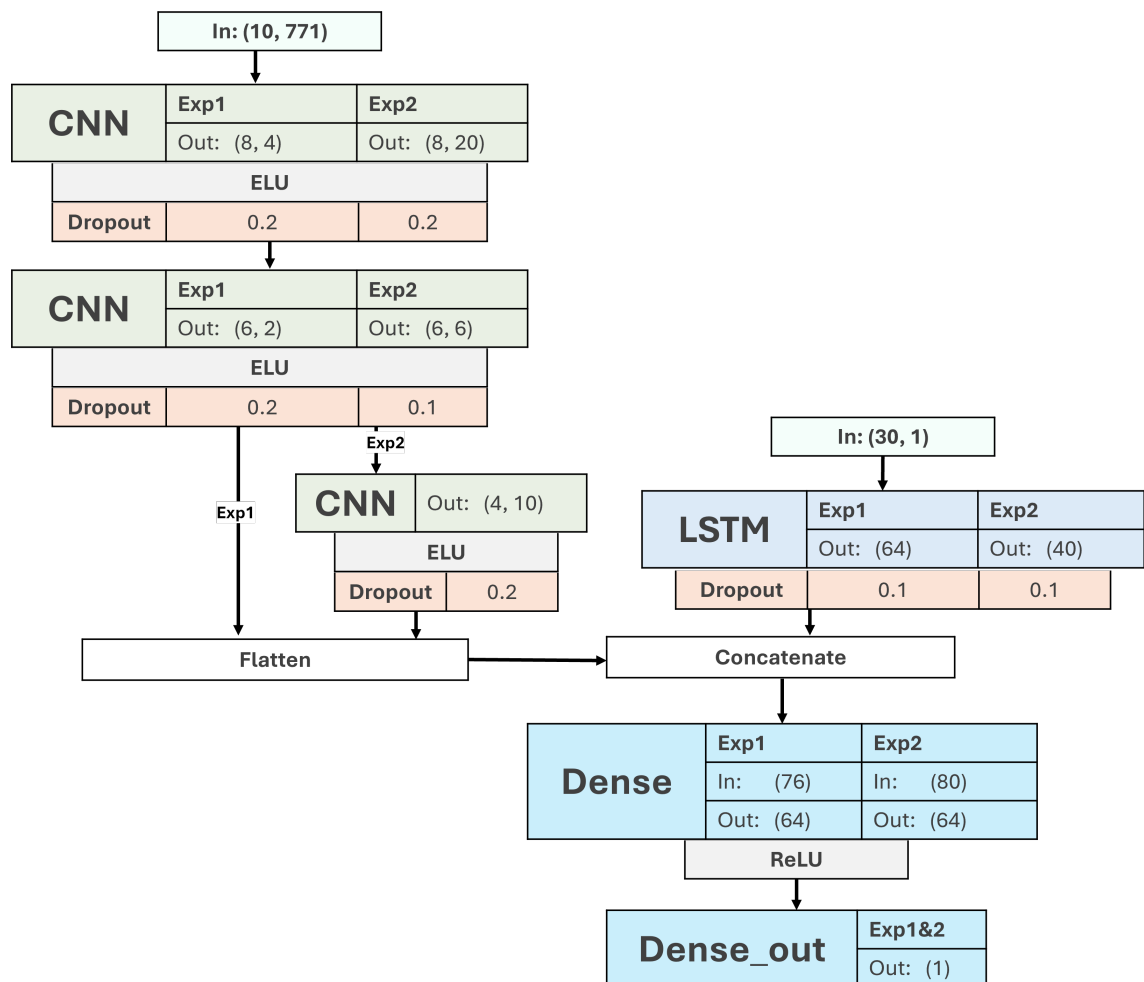
in den einzelnen Schichten und Lernrate wird mit Hilfe von `keras` eine automatisierte Hyperparameter-Optimierung durchgeführt (Kap. 2.3). Jeder einzelne Parameter wird im Vorfeld mit einem eingrenzenden Intervall versehen. Außerdem wird die Schrittgröße, mit der ein weiterer Wert evaluiert wird vorgegeben. Während der Optimierung wird mit verschiedenen Sets an Parameter-Kandidaten trainiert und die Konfiguration mit dem besten Validierungsverlust wird übernommen. Dadurch entstehen, wie in Abbildung 7 dargestellt, je nach Experiment unterschiedliche Parameterwerte. Die resultierende Lernrate beträgt bei beiden Experimenten 0.000574.

### 3.3.3 Forschungsmodell

Das Forschungsmodell besteht anfangs aus 2 Zweigen, welche im Verlauf zu einem vollständig verbundenen Netzwerk zusammenführen. Der rechte Abschnitt in Abbildung 8 ist der Architektur des Basismodells sehr ähnlich. Der größte Unterschied dabei ist die Anzahl an LSTM-Schichten. Der iterative CRISP-DM-Prozess hat ergeben, dass im vorliegenden Fall die Verwendung eines LSTM-Layers bessere Ergebnisse liefert. Zusätzlich zum Schlusspreis nimmt das Forschungsmodell im linken Eingangszweig vorverarbeitete Stimmungsdaten entgegen (Abb. 8). Dazu gehören die drei Metriken "positive", "negative" und "num\_tweets", sowie der Embedding-Vektor "tweet\_embs", zu sehen in Abbildung 6. Die Dimension des Eingangsvektors der Länge 771 ergibt sich aus dem Zusammenführen der 3 Metriken und 768 Embedding-Werte. In Anlehnung an eine Forschungsarbeit mit ähnlicher Datengrundlage wird für die Stimmungsdaten ein kleineres Fenster von 10 Markttagen verwendet [1, Kap. 4.1]. Diese mehrtägigen Sentimentdaten durchlaufen zunächst mehrere eindimensionale CNN-Schichten. Deren Filter haben stets die Größe 3 und werden mit der Schrittgröße 1 ohne Padding angewandt. Daher reduziert sich die erste Dimension des Datentupels bei jeder CNN-Schicht um den Wert 2 (z. B. bei Exp1 von 10 bei Eingabe, auf 6 nach der 2. CNN-Schicht). Die Anzahl der Filter gehört zu den variablen Hyperparametern und entspricht der 2. Dimension des CNN-Ausgabetupels (z. B. bei Exp2 im ersten CNN 20 Filter). Die Anzahl an CNN-Schichten ergab sich aus CRISP-DM-Iterationen, wobei Forschungsarbeiten mit ähnlichem Hintergrund bereits zeigten, dass 2–3 Schichten ausreichend sind [26, Kap. 8.2]. Beim Lauf der Daten durch diese Ebenen erfolgt zwischendrin jeweils zunächst die Anwendung der nichtlinearen Aktivierungsfunktion Exponential Linear Unit (ELU) und anschließend eine Dropout-Operation. Nachdem die Daten alle CNN-Layer durchlaufen haben, werden sie in einer Entfaltungsschicht zu einem 1-dimensionalen Vektor transformiert. Im nächsten Schritt erfolgt ein Zusammenführen der Ergebnisse beider Modell-Eingangszweige zu einem Vektor. Diese Vorgehensweise fand bereits Verwendung in einer wissenschaftlichen Arbeit über Aktienprognosen von Qiuyue

Zhang [1, Kap. 4.2.1]. Die nachfolgenden Schichten in Abbildung 8 gleichen den letzten Schritten des Basismodells (Abb. 7). Auch die Hyperparameteroptimierung verläuft nahezu identisch, wobei in diesem Fall die Filteranzahl in jeder CNN-Schicht einen zusätzlichen Parameter darstellt. Die aus der Optimierung resultierende Lernrate unterscheidet sich nun je Experiment leicht (Exp1: 0.000595, Exp2: 0.00047529998).

**Abbildung 8:** Forschungsmodell-Architektur mit Unterscheidung zwischen Experiment 1 und 2



### 3.4 Evaluierung

Wie bereits in Kapitel 3.5 erwähnt, wurden 2 Experimente durchgeführt. Zur Evaluierung werden verschiedene Metriken und Diagramme betrachtet. Dazu gehören zum einen der MSE als gewählte Verlustfunktion (Loss) während des Trainings, welcher anhand von skalierten Evaluierungsdaten berechnet wird und zum anderen ein Prognosediagramm und die Metriken MAE, MSE und MAPE (Kap. 2.7), welche sich bei der Verwendung von Testdaten ergeben. Für diese Metrik-Berechnungen und das Prognosediagramm werden nicht-skalierte Preise verwendet. In den nachfolgenden Abschnitten werden die Ergebnisse aufgeführt und analysiert. Im Anschluss daran erfolgt eine Zusammenfassung der gewonnenen Erkenntnisse.

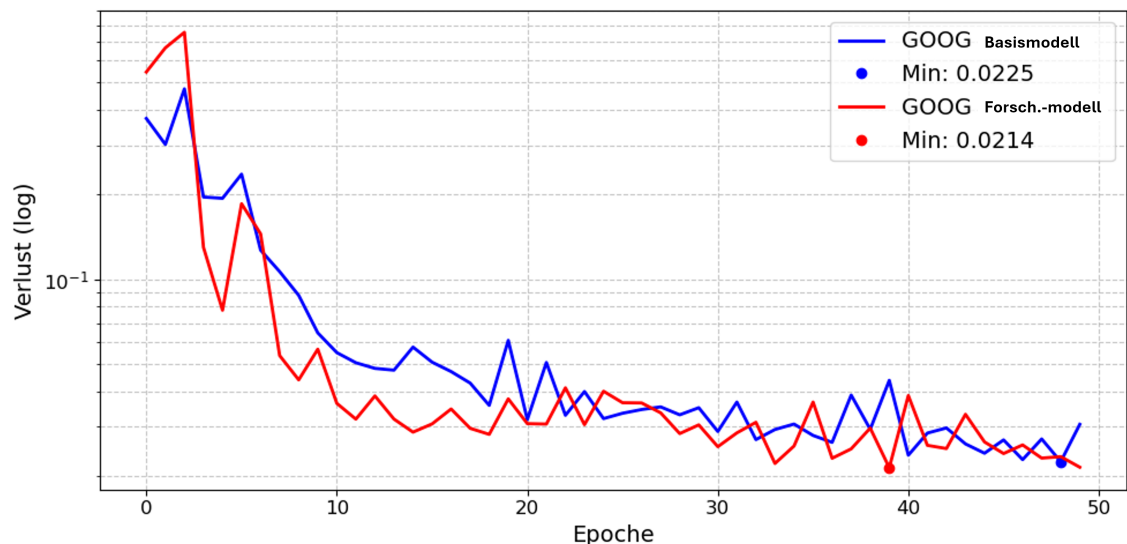
#### 3.4.1 Experiment 1: GOOG-Aktie

In Experiment 1 wird das Basismodell mit dem Forschungsmodell verglichen, wobei lediglich Daten der GOOG-Aktie berücksichtigt werden.

#### Verlust-Metrik

In Abbildung 9 wird die berechnete Verlustentwicklung während des Trainings beider Modelle gegenübergestellt.

**Abbildung 9:** Experiment 1: Verlust-Vergleich von Basis- und Forschungsmodell, berechnet anhand von Validierungsdaten der GOOG-Aktie

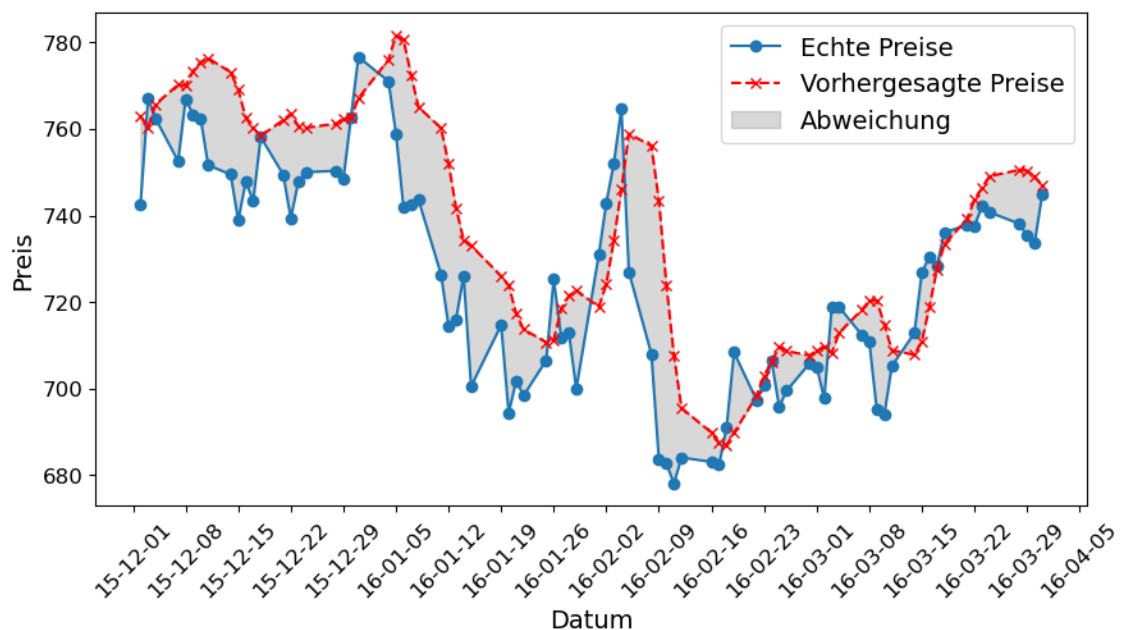


Die beiden Varianten weisen einen sehr ähnlichen Verlust-Kurvenverlauf auf. Es ist jedoch klar zu erkennen, dass die multimodale Variante leicht bessere Werte erreicht. Ihr Bestwert beträgt  $2.14 \cdot 10^{-2}$ , welcher um 4,9% geringer ausfällt als beim Vergleichsmodell.

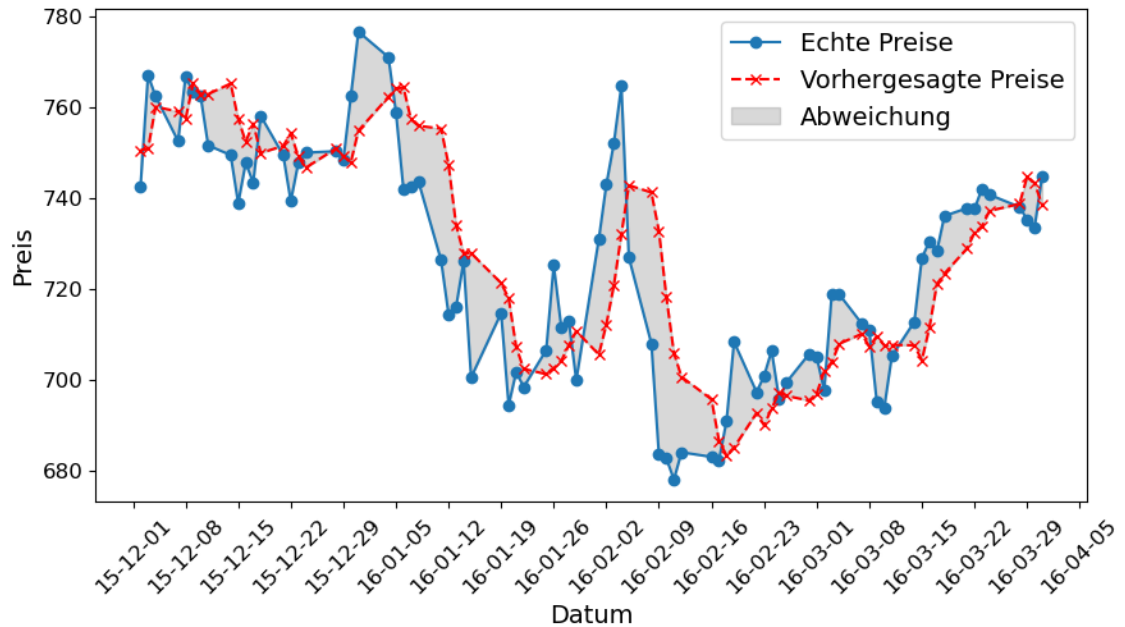
### Prognose-Metriken

In den Abbildungen 10 und 11 wird der tatsächliche Preis mit der Modellvorhersage verglichen. Die Berechnung der Modellvorhersage erfolgt schrittweise, wobei zur Berechnung des Preises eines Folgetages Realdaten der Vergangenheit an das jeweilige Modell übergeben werden. Demnach erfolgt technisch gesehen keine Langzeitprognose, sondern Tagesprognosen, die auf echten Erfahrungswerten basieren.

**Abbildung 10:** Experiment 1: Basismodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen



**Abbildung 11:** Experiment 1: Forschungsmodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen



Auf Basis der Diagramme ist eine Verbesserung nur schwer erkennbar. Ein Anhaltspunkt ist beispielsweise die Abweichung im Zeitraum "15-12-22" bis "15-12-29". In Abbildung 11 gibt es in diesem Bereich einen deutlich geringeren Anteil an Graufäche, als es bei Abbildung 10 der Fall ist.

Anhand der Metrik-Tabelle 1 werden Qualitätsunterschiede deutlicher. Dabei werden die Metriken MAE, MSE und MAPE separat für beide Modelle berechnet.

**Tabelle 1:** Experiment 1: Metrik-Vergleich zwischen Basis- und Forschungsmodell

Modell	MAE	MSE	MAPE (%)
Basis	14.78	357.0	2.05
Forschung	12.45	256.12	1.72

Die Metriken des Forschungsmodells weisen eine messbare Verbesserung im Vergleich zum Basismodell auf. Dies lässt darauf schließen, dass die Integration von Stimmungsdaten in ein DL-Modell für verbesserte Aktienprognosen sorgen kann.

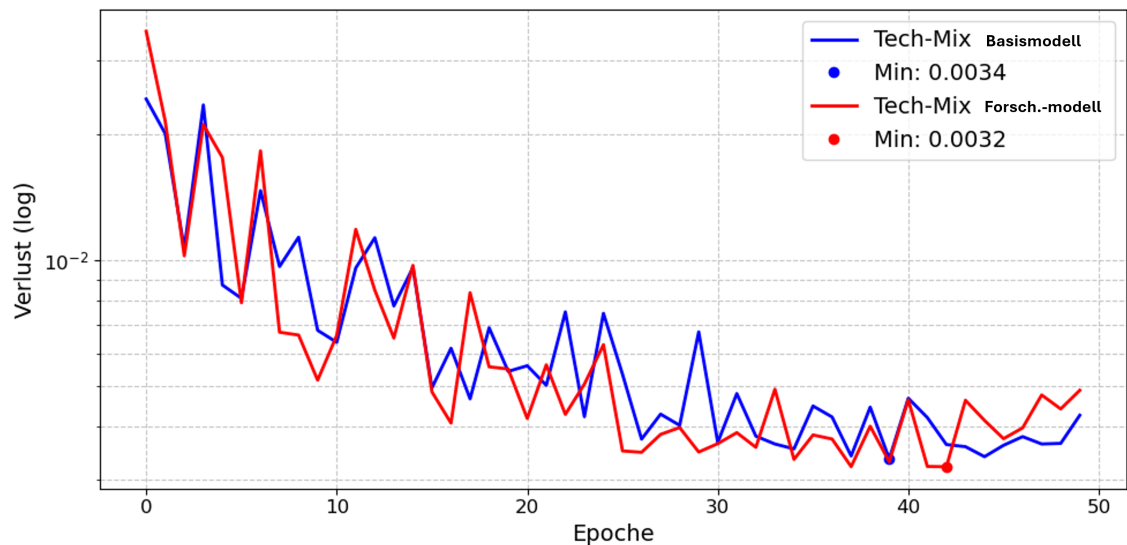
### 3.4.2 Experiment 2: Technologie-Aktienmix

In Experiment 2 werden Basis- und Forschungsmodell anhand von Daten mehrerer Aktien der Technologie-Branche trainiert und anschließend anhand von Daten der GOOG-Aktie hinsichtlich Prognose-Qualität getestet.

#### Verlust-Metrik

In Abbildung 12 wird die berechnete Verlustentwicklung während des Trainings beider Modelle gegenübergestellt.

**Abbildung 12:** Experiment 2: Verlustvergleich von Basis- und Forschungsmodell, berechnet anhand der Validierungsdaten von Technologieaktien



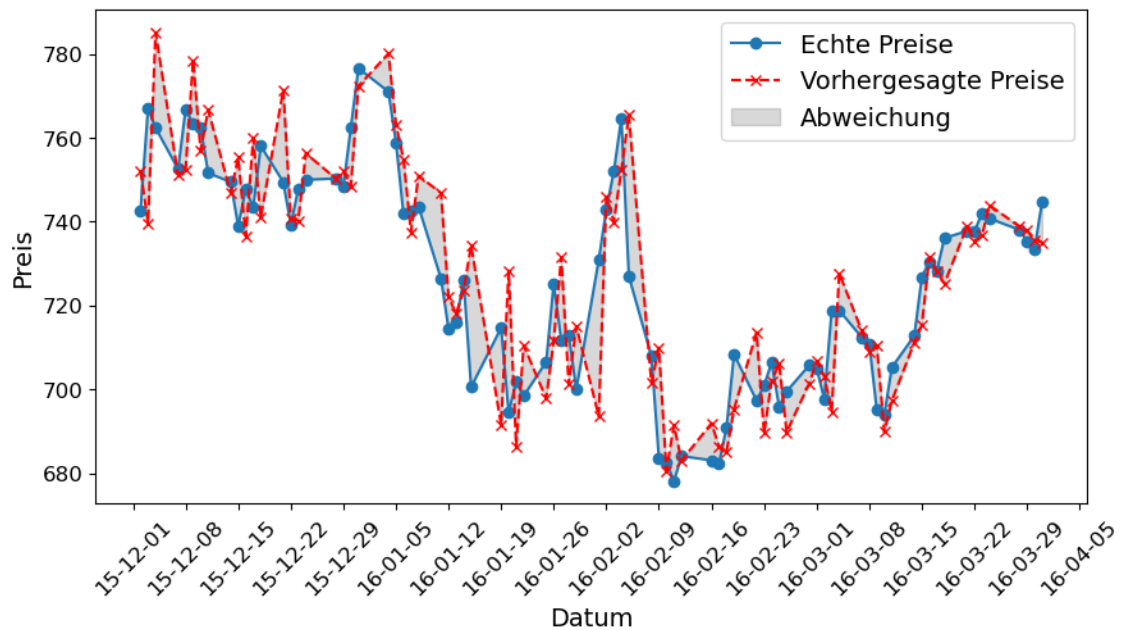
Bereits das Basismodell weist deutlich bessere Verlustwerte auf als beide Modelle aus Experiment 1. Das multimodale Forschungsmodell benötigt nun zur Erreichung des Bestwerts mehr Trainingsepochen als das Vergleichsmodell und weist erneut ein leicht besseres Ergebnis auf. Sein minimaler Verlust beträgt  $3.2 \cdot 10^{-3}$  und ist somit um 5,9% kleiner als beim Basismodell.

#### Prognose-Metriken

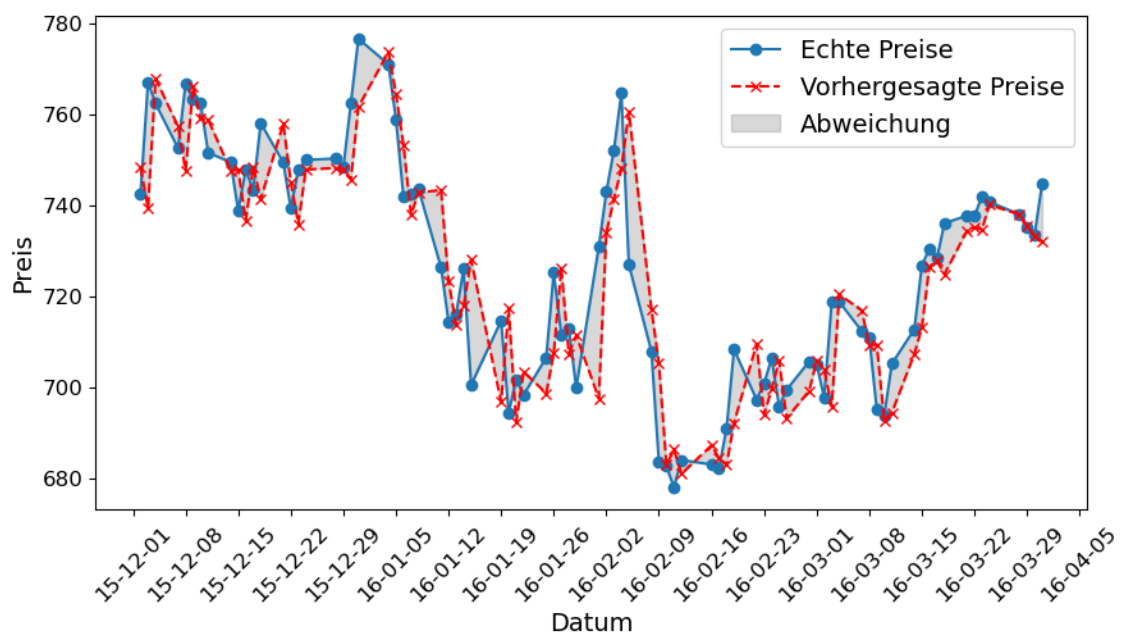
Aufgrund der Tatsache, dass die Modelle in diesem Experiment mit Daten mehrerer Aktien trainiert werden, lässt sich für jede involvierte Aktie ein Prognosediagramm erstellen. Für

die Vergleichbarkeit mit Experiment 1 erfolgt eine Fokussierung auf die GOOG-Aktie. In den Abbildungen 13 und 14 wird daher erneut, bezogen auf die GOOG-Aktie, der echte Preis mit Tagesprognosen verglichen.

**Abbildung 13:** Experiment 2: Basismodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen



**Abbildung 14:** Experiment 2: Forschungsmodell - echter vs. vorhergesagter Preis basierend auf Tagesprognosen





Beim direkten Vergleich beider Diagramme fällt auf, dass in Diagramm 13 die Positionen vorhergesagter Preise deutlich stärker in vertikaler Richtung abweichen als in Abbildung 14. Das Forschungsmodell überzeugt also mit Preisvorhersagen nahe am echten Wert. Mit Hilfe der Metrik-Tabelle 2 werden Qualitätsunterschiede noch deutlicher.

**Tabelle 2:** Experiment 2: Metrik-Vergleich

<b>Modell</b>	<b>MAE</b>	<b>MSE</b>	<b>MAPE (%)</b>
Basis	10.46	187.48	1.44
Forschung	8.97	137.72	1.24

Wie bereits durch die Verlustmetrik angedeutet, erreicht das Basismodell bessere Werte als beide Modelle aus Experiment 1. Dies spiegelt sich noch deutlicher in den Testmetriken aus Tabelle 2 wider. Darüber hinaus konnte auch in diesem Fall mit Hilfe des Forschungsmodells das Ergebnis des Basismodells übertroffen werden.

### 3.4.3 Erkenntnisse

Anhand der durchgeführten Experimente lassen sich nützliche Erkenntnisse ableiten, welche in diesem Abschnitt zusammengefasst werden. Es können zwei Strategien abgeleitet werden, welche eine Verbesserung der Aktienprognose-Qualität ermöglichen. Zum einen die Integration von Stimmungsdaten und zum anderen die Verwendung von Aktienbündeln. Die Verwendung von mehreren Aktien zur Modellierung von Aktienpreisen hat sich als sehr effektiv erwiesen. So konnte die MSE-Metrik der Tabelle 1 in Tabelle 2 beim Basismodell um 47,5% und beim Forschungsmodell um 46,2% verbessert werden. Auch die Strategie der Integration von Stimmungsdaten zeigte sich als gewinnbringend. So konnte die MSE-Metrik des Basismodells aus Experiment 1 mit Hilfe des Forschungsmodells um 28,3% verbessert werden und in Experiment 2 um 26,5%. Den größten Effekt hatte damit die Einbindung von Daten mehrerer Aktien und somit kann eine tiefgründige Extraktion von Informationen bezüglich komplexer Beziehungen zwischen Wertpapieren als sehr gewinnbringend für Aktienprognosen eingestuft werden. Bei zusätzlicher Hinzunahme von Stimmungsdaten wurden die Ergebnisse noch weiter verbessert (Exp 2 Forschungsmodell).

### 3.5 Deployment

Das Deployment von DL-Modellen ist ein entscheidender Schritt, um die entwickelten Modelle in der Praxis anzuwenden und deren Nutzen zu maximieren. Bei der Vorhersage von Aktienkursen spielt die Bereitstellung von Modellen eine zentrale Rolle. Der Prozess umfasst mehrere Phasen, von der Vorbereitung der Produktionsumgebung bis hin zur Überwachung der Modellleistung. Zur Komplexitätsreduktion wird daher im Folgenden nur kurz auf die wichtigsten Aspekte eingegangen.

Zunächst ist es wichtig, ein stabiles und skalierbares Infrastruktur-Setup zu schaffen. Cloud-basierte Plattformen wie AWS, Google Cloud oder Microsoft Azure bieten flexible Lösungen, die es Unternehmen ermöglichen, ihre Modelle in einer sicheren und skalierbaren Umgebung zu betreiben. Diese Plattformen unterstützen auch die Integration von Datenquellen in Echtzeit, was für die kontinuierliche Aktualisierung und Verbesserung der Modelle entscheidend ist.

Nach der Bereitstellung muss das entwickelte Forschungsmodell regelmäßig überwacht werden, um sicherzustellen, dass es unter verschiedenen Bedingungen konsistent funktioniert. Dies beinhaltet die Überwachung von Metriken wie der Genauigkeit der Vorhersagen sowie der Verteilung der Daten, um sicherzustellen, dass das Modell nicht unter dem Problem des "drift" leidet, bei dem sich die zugrunde liegenden Datenmuster im Laufe der Zeit ändern.

Ein weiterer wichtiger Aspekt ist die kontinuierliche Verbesserung des Modells. Durch den Einsatz von Feedback-Schleifen und das Sammeln von neuen Daten aus der Finanzwelt können Unternehmen das Modell anpassen und optimieren.

Insgesamt ist ein effektives Deployment von DL-Modellen nicht nur entscheidend für den Erfolg bei der Aktienkursprognose, sondern auch für die langfristige Wettbewerbsfähigkeit im Finanzsektor [27][28][29].

## 4 Fazit & Ausblick

Abschließend erfolgt in diesem Kapitel eine übergeordnete Analyse der Ergebnisse hinsichtlich Anwendbarkeit der entwickelten Lösung. Es erfolgt eine genaue Betrachtung der bestehenden Möglichkeiten und Limitationen. Abgeleitet davon erfolgt im letzten Abschnitt ein Ausblick auf mögliche Erweiterungen durch zukünftige Arbeiten.

### 4.1 Möglichkeiten

Die Ergebnisse aus Kapitel 3.4 zeigen deutlich, dass DL-Modelle in der Lage sind, komplexe Muster in Kurs- und Stimmungsdaten zu erlernen, welche präzise Aktienprognosen unterstützen. Das Forschungsmodell aus Experiment 2 ist in der Lage, eine Tagesprognose mit einem MAPE von 1.24% zu liefern. Je nachdem, ob diese Vorhersage des Folgepreises einen Anstieg oder Abfall darstellt, lässt sich davon eine Handelsentscheidung ableiten. Diese sollte jedoch nicht ausschließlich auf Basis der Prognose getroffen werden, sondern in Kombination mit weiteren Informationen und Analysen. Ähnlich wie die Technik Ensemble-Learning aus dem Machine-Learning-Bereich könnte die Verwendung von mehreren Modellen das Risiko einer Fehlentscheidung reduzieren [30]. Außerdem ist es wichtig, dass man sich der Limitierungen des Modells bewusst ist und diese bei der Handelsentscheidung berücksichtigt.

### 4.2 Limitationen

Das Ziel von Aktienprognosen mit Hilfe von Deep Learning besteht natürlich darin, anhand von Modellausgaben auf lange Sicht am Aktienmarkt Gewinne zu realisieren. Die Modelle aus Experiment 1 sind aufgrund der hohen Anforderung an den Umfang der Trainingsdaten lediglich auf bereits langjährig etablierte Aktien anwendbar. Experiment 2 lässt sich in einer nachfolgenden Arbeit dahingehend erweitern, dass Daten etablierter und neuer Aktien gemischt für das Training verwendet werden. Doch in beiden Fällen ist für eine optimale Vorhersage das Bestehen der Aktie seit 30 Markttagen in die Vergangenheit notwendig. Eine weitreichendere Limitation besteht darin, dass die Preisvorhersagen eine Trägheit aufweisen. Das heißt, dass in Phasen von Preisanstieg höchstwahrscheinlich auch ein weiterer Anstieg prognostiziert wird. Die Modelle weisen deutlich Schwierigkeiten bei der Vorhersage von Richtungsänderungen auf. Doch genau diese Fähigkeit ist für ein erfolgreiches Handeln am Aktienmarkt wichtig. Ein Grund dafür könnte sein, dass im Bereich der Sentiment-Daten alle Tweets mit gleichem Gewicht in die Modelle einfließen. Dies spiegelt

nicht die Realität wider, da Tweets von hochrangigen Experten oft mehr Einfluss auf die Marktentwicklung nachweisen als Tweets von Laien. Eine weitere substanzielle Limitation besteht darin, dass lediglich der Preis eines Folgetages anhand von echten Preisen 30 vorangegangener Markttage vorhergesagt wurde. Die Modelle sind nicht auf mehrtägige Prognosen optimiert und wurden nicht dahingehend getestet.

### **4.3 Ausblick**

Um eine Verbesserung der Vorhersagequalität durch Integration von Stimmungsdaten zu erreichen, sollte im Beispiel von Tweets die Tragweite der Aussagen abhängig vom Tweet-Ersteller und seiner Reichweite gewichtet werden. Darüber hinaus sollten weitere Informationsquellen wie News, Unternehmensberichte oder Interviews in die Analyse einbezogen werden. Ein aktuelles Beispiel für Kursänderung aufgrund von Expertenmeinungen ist eine Q&A-Runde mit dem Nvidia-CEO Jensen Huang und Analysten, in der sich Jensen pessimistisch gegenüber der Zukunft von Quantencomputern geäußert hatte. Dies führte zu einem sprunghaften Kursrückgang von zahlreichen Quantenaktien, welcher sich anschließend allmählich wieder erholte [31][32]. Des Weiteren sollte die Datenqualität optimiert werden, indem beispielsweise inhaltslose Tweets herausgefiltert werden. Außerdem kann die Verwendung eines großen Sprachmodells wie GPT-4o oder Deepseek-V3 zur Extraktion von Stimmungsdaten die Vorhersage-Qualität verbessern [33][34]. Ein weiterer wichtiger Aspekt, den es zu vertiefen gilt, ist die Analyse von Beziehungen zwischen verschiedenen Aktien, um so eine bestmögliche Zusammenstellung mehrerer Aktien für das Training zu erreichen. Dies könnte durch die Verwendung von Graph-Neural-Networks realisiert werden, welche in der Lage sind, Beziehungen zwischen Knoten zu erlernen [35][1, Kap. 6]. Diese Strategie ist auch ein wichtiger Schritt, um zuverlässige Prognosen für neuere Aktien zu ermöglichen.

## Literatur

- [1] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang und P. Liu, *Transformer-based attention network for stock movement prediction*. Elsevier, 2022. DOI: 10.1016/j.eswa.2022.117033.
- [2] I. Goodfellow, Y. Bengio und A. Courville, *Deep Learning*. MIT Press, 2016. Adresse: <https://www.deeplearningbook.org>.
- [3] S. Hochreiter und J. Schmidhuber, *Long Short-Term Memory*. MIT Press, 1997, S. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- [4] Y. Xu und S. B. Cohen, *Stock Movement Prediction from Tweets and Historical Prices*. Association for Computational Linguistics, Juli 2018, S. 1970–1979. DOI: 10.18653/v1/P18-1183.
- [5] K. Chen et al., *StockNet: A Comprehensive Data Set for Stock Movement Prediction*. 2018. Adresse: <https://arxiv.org/abs/1803.09470>.
- [6] M. Enache, *Data Analysis with Pandas*. Juli 2019, S. 69–74. DOI: 10.35219/eai1584040933.
- [7] J. Heaton, *Applications of Deep Neural Networks with Keras*. 2022. eprint: 2009.05673. Adresse: <https://arxiv.org/abs/2009.05673>.
- [8] T. Fischer und C. Krauss, *Deep learning with long short-term memory networks for financial market predictions*. Elsevier, 2018, S. 654–669.
- [9] S. Siامي-Namini, A. Tavakoli und A. Namin, *The performance of LSTM and BiLSTM in forecasting time series*. IEEE, 2019, S. 3285–3292.
- [10] J. Devlin, M.-W. Chang, K. Lee und K. Toutanova, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,“ in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, S. 4171–4186.
- [11] Y. Yang, M. Chen, A. Ramesh, W. Xu und L. Zhang, *FinBERT: A Pretrained Language Model for Financial Communications*. 2019.
- [12] Z. Yang, X. Li und Y. Zhang, *Financial News Predicts Stock Market Volatility*. 2020, S. 123–164.
- [13] Y. Kim, *Convolutional Neural Networks for Sentence Classification*. Sep. 2014. DOI: 10.48550/arXiv.1408.5882.
- [14] Z. Zhang, Y. Wu, L. Gao und Y. Huang, „Sensitivity of CNNs to Input Noise,“ in *Proceedings of the 2015 International Conference on Machine Learning*, 2015, S. 120–128.
- [15] D. Wallach und B. Goffinet, *Mean squared error of prediction as a criterion for evaluating and comparing system models*. Elsevier, 1989, S. 299–306. DOI: 10.1016/0304-3800(89)90035-5.

- 
- [16] C. J. Willmott und K. Matsuura, *Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance*. Inter-Research Science Center, 2005, S. 79–82. Adresse: <https://www.jstor.org/stable/24869236>.
  - [17] A. De Myttenaere, B. Golden, B. Le Grand und F. Rossi, *Mean Absolute Percentage Error for regression models*. Elsevier, 2016, S. 38–48. DOI: 10.1016/j.neucom.2015.12.114.
  - [18] R. J. Hyndman und A. B. Koehler, *Another look at forecasting accuracy*. 2006, S. 273–284.
  - [19] S. Makridakis, R. M. Hogarth und A. Gaba, *Accuracy measures: A new perspective*. John Wiley & Sons, 1993, S. 673–685.
  - [20] H. Xu, L. Chai, Z. Luo und S. Li, *Stock movement predictive network via incorporative attention mechanisms based on tweet and historical prices*. Elsevier, 2020, S. 326–339. DOI: 10.1016/j.neucom.2020.07.109.
  - [22] A. R. Hazourli, *FinancialBERT - A Pretrained Language Model for Financial Text Mining*. Feb. 2022. DOI: 10.13140/RG.2.2.34032.12803.
  - [24] L. Xie, Z. Chen und S. Yu, *Deep Convolutional Transformer Network for Stock Movement Prediction*. MDPI, Okt. 2024, S. 4225. DOI: 10.3390/electronics13214225.
  - [26] Y. Guan, P. Li und C. Lu, *Stock Price Prediction with CNN-LSTM Network*. Jan. 2020.
  - [27] T. Mason und D. Ahn, *Machine learning for financial applications: A survey*. 2018, S. 1–19.
  - [28] W. Chen, L. Zhang und Q. Wang, „Model Deployment Strategies in Machine Learning,“ in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, IEEE, 2019, S. 278–285.
  - [30] H. Wu und D. Levinson, *The ensemble approach to forecasting: A review and synthesis*. Elsevier, 2021, S. 103357. DOI: 10.1016/j.trc.2021.103357.
  - [35] H. Qian et al., *MDGNN: Multi-Relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction*. März 2024, S. 14642–14650. DOI: 10.1609/aaai.v38i13.29381.

## Sonstige Quellen

- [21] Y. Xu und dtaylor-530, *StockNet Dataset: A Comprehensive Dataset for Stock Movement Prediction from Tweets and Historical Stock Prices*, MIT License, 2017. besucht am 20. Dez. 2024. Adresse: <https://github.com/yumoxu/stocknet-dataset>.
- [23] Scikit-learn Community, *Scikit-learn*, 2024. besucht am 4. Nov. 2024. Adresse: <http://scikit-learn.org/stable/about.html>.
- [25] K. Team, *Adam Optimizer, Keras 3 API Documentation*, Keras API documentation, 2025. besucht am 3. Jan. 2025. Adresse: <https://keras.io/api/optimizers/adam/>.
- [29] M. Azure, *Machine Learning Operations (MLOps) for Azure*, 2020. besucht am 8. Feb. 2025. Adresse: <https://learn.microsoft.com/en-us/azure/architecture/microservices/implement/azure-ml-ops>.
- [31] ntv.de, *Nvidia-Chef löst Kurseinbruch bei Quantencomputer-Aktien aus*, ntv.de, Jan. 2025. besucht am 8. Feb. 2025. Adresse: [https://www.n-tv.de/wirtschaft/der\\_boersen\\_tag/Nvidia-Chef-loest-Kurseinbruch-bei-Quantencomputer-Aktien-aus-article25475642.html](https://www.n-tv.de/wirtschaft/der_boersen_tag/Nvidia-Chef-loest-Kurseinbruch-bei-Quantencomputer-Aktien-aus-article25475642.html).
- [32] R. finanzen.net, *Aktien von D-Wave Quantum, Rigetti und Co. springen hoch: Microsoft mit Durchbruch bei Quantencomputer-Chips*, finanzen.net, Feb. 2025. besucht am 21. Feb. 2025. Adresse: <https://www.finanzen.net/nachricht/aktien/majorana-1-aktien-von-d-wave-quantum-rigetti-und-co-springen-hoch-microsoft-mit-durchbruch-bei-quantencomputer-chips-14256214>.
- [33] DeepSeek Inc, *DeepSeek API Docs*, Online, Apr. 2023. besucht am 12. Feb. 2025. Adresse: [https://api-docs.deepseek.com/quick\\_start/pricing](https://api-docs.deepseek.com/quick_start/pricing).
- [34] S. Thoma, *GPT-4o Überblick*, Online, 2025. besucht am 12. Feb. 2025. Adresse: <https://chatopenai.de/gpt-4o/hallo/>.

---

## Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die angemeldete Prüfungsleistung in allen Teilen eigenständig ohne Hilfe von Dritten anfertigen und keine anderen als die in der Prüfungsleistung angegebenen Quellen und zugelassenen Hilfsmittel verwenden werde. Sämtliche wörtlichen und sinngemäßen Übernahmen inklusive KI-generierter Inhalte werde ich kenntlich machen. Diese Prüfungsleistung hat zum Zeitpunkt der Abgabe weder in gleicher noch in ähnlicher Form, auch nicht auszugsweise, bereits einer Prüfungsbehörde zur Prüfung vorgelegen; hiervon ausgenommen sind Prüfungsleistungen, für die in der Modulbeschreibung ausdrücklich andere Regelungen festgelegt sind. Mir ist bekannt, dass die Zuwiderhandlung gegen den Inhalt dieser Erklärung einen Täuschungsversuch darstellt, der das Nichtbestehen der Prüfung zur Folge hat und daneben strafrechtlich gem. § 156 StGB verfolgt werden kann. Darüber hinaus ist mir bekannt, dass ich bei schwerwiegender Täuschung exmatrikuliert und mit einer Geldbuße bis zu 50.000 EUR nach der für mich gültigen Rahmenprüfungsordnung belegt werden kann. Ich erkläre mich damit einverstanden, dass diese Prüfungsleistung zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

Magdeburg, 21.02.2025

---

(Ort, Datum)




---

(Paul Hornig, Unterschrift)

Mannheim, 14.02.2025

---

(Ort, Datum)



---

(Admir Dutovic, Unterschrift)