



**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 <AdmirNr>  
Abgabedatum : 21. Februar 2025

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>II</b>
<b>Abkürzungsverzeichnis</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 Datenvorverarbeitung mit Pandas . . . . .	2
2.3 Aktienprognosen mit Deep Learning . . . . .	2
2.3.1 Modellierung mit Keras . . . . .	2
2.3.2 LSTM für Kursdaten . . . . .	2
2.3.3 LLM für Stimmungsdaten . . . . .	2
2.3.4 CNN für Stimmungsdaten . . . . .	3
2.3.5 Evaluierungsmetriken . . . . .	3
<b>3 Methodik</b>	<b>4</b>
3.1 Datenbeschaffung . . . . .	4
3.2 Datenvorverarbeitung . . . . .	5
3.3 Modellierung . . . . .	10
3.3.1 Generelle Hyperparameter . . . . .	10
3.3.2 Basismodell . . . . .	11
3.3.3 Forschungsmodell . . . . .	12
3.4 Evaluierung . . . . .	14
3.4.1 Experiment 1: GOOG-Aktie . . . . .	14
3.4.2 Experiment 2: Technologie-Aktienmix . . . . .	16
3.4.3 Limitationen . . . . .	19
3.4.4 Erkenntnisse . . . . .	19
3.5 Deployment . . . . .	20
<b>4 Fazit &amp; Zukünftige Arbeiten</b>	<b>21</b>
<b>Quellenverzeichnis</b>	

## Abbildungsverzeichnis

1	Auszug aus GOOG-Kursdaten . . . . .	5
2	GOOG Tweet vom 01.01.2014 . . . . .	5
3	GOOG-Datenzeile mit vereinigten Kurs- und Tweetdaten . . . . .	6
4	GOOG-Datenzeile nach der Bereinigung von Tweets . . . . .	6
5	GOOG-Datenzeile im Datenkorpus . . . . .	9
6	Basismodell-Architektur mit Unterscheidung zwischen Experiment 1 und 2	11
7	Forschungsmodell-Architektur mit Unterscheidung zwischen Experiment 1 und 2 . . . . .	13
8	Verlust-Vergleich von Basis- und Forschungsmodell, berechnet anhand von Validierungsdaten der GOOG-Aktie . . . . .	14
9	Basismodell - echter vs vorhergesagter Preis basierend auf Tagesprognosen	15
10	Forschungsmodell - echter vs vorhergesagter Preis basierend auf Tages- prognosen . . . . .	16
11	Verlust-Vergleich von Basis- und ForschungsmodellII, berechnet anhand der Validierungsdaten von Technologieaktien . . . . .	17
12	Basismodell - Echte vs Vorhergesagte Preise . . . . .	18
13	Forschungsmodell - Echte vs Vorhergesagte Preise . . . . .	18

## Abkürzungsverzeichnis

<b>CNN</b>	Convolutional Neural Network
<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

# 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 großen Sprachmodellen, mit deren Hilfe Stimmungsdaten erzeugt werden sollen. Ein weiterer Schwerpunkt ist die Ermittlung einer geeigneten DL-Architektur, welche mit Numerik- und Textdaten trainiert wird.

Die Strukturierung dieser Arbeit orientiert sich am Cross Industry Standard Process for Data Mining (CRISP-DM) Modell. In Kapitel ?? wird daher zunächst wichtiges Domänenwissen behandelt. Anschließend erfolgt eine Beschreibung der praktischen Umsetzung mitsamt Evaluierung. Darüber hinaus wird auf Möglichkeiten eines Deployment eingegangen. Zum Schluss erfolgt eine zusammenfassende Analyse in Form eines Fazits.

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

## 2 Theoretische Grundlagen

### 2.1 Datenverständnis

—

Here's the German translation: Im Finanzbereich werden Aktien in 9 Branchen kategorisiert: Grundstoffe, Konsumgüter, Gesundheitswesen, Dienstleistungen, Versorgungsunternehmen, Mischkonzerne, Finanzwesen, Industriegüter und Technologie. Da Aktien mit hohem Handelsvolumen tendenziell häufiger auf Twitter diskutiert werden, wählen wir die zweijährigen Kursbewegungen von 88 Aktien vom 01.01.2014 bis 01.01.2016 als Ziele aus, bestehend aus allen 8 Aktien der Mischkonzerne und den Top 10 Aktien nach Kapitalvolumen aus jeder der anderen 8 Branchen (siehe ergänzendes Material).

— [2, Kap. 3]

### 2.2 Datenvorverarbeitung mit Pandas

### 2.3 Aktienprognosen mit Deep Learning

#### 2.3.1 Modellierung mit Keras

- hyperparameter optimisierung mit "hyperband" [3]

- lern algorithmus adam [4] [5]

#### 2.3.2 LSTM für Kursdaten

gute quelle für formeln: [6]

#### 2.3.3 LLM für Stimmungsdaten

—

FinancialBERT applies domain-specific language understanding to financial text analysis. Built by ahmedrachid, this model stands alongside other financial sentiment analyzers like finbert-tone and finbert. The model was fine-tuned on the Financial PhraseBank dataset, achieving 98% weighted average precision across sentiment categories. ...

— [7]

### 2.3.4 CNN für Stimmungsdaten

- begründen warum filter der gröÙe 3 gut gute quelle für formeln: [6]

### 2.3.5 Evaluierungsmetriken

- Mean Absolute Error (MAE), Mean Squared Error (MSE) und Mean Absolute Percentage Error (MAPE) [8, Kap. 4.3]

mse:

**Formel 1: MSE**

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

Quelle: ...

## 3 Methodik

Die Umsetzung erfolgt in logischen Schritten und orientiert sich dabei am Prozessmodell CRISP-DM.

### 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 [2][9][1]. Die Daten werden auf GitHub unter der MIT-Lizenz zur Verfügung gestellt [10].

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 [2, 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 (Kap. 2.1), welche im entsprechenden Zeitraum unter den Top-10-Aktien nach Handelsvolumen vorkommen.

#### Kursdaten

Zu jeder Aktie liegt eine **csv!** (**csv!**)-Datei vor. In Abbildung 1 werden die inkludierten Metriken dargestellt.



**Abbildung 1:** Auszug aus GOOG-Kursdaten. Quelle: [10]

```

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 2 stellt einen Eintrag dieser Liste dar, wobei lediglich relevante Attribute einbezogen sind.

**Abbildung 2:** GOOG Tweet vom 01.01.2014. Quelle: [10]

```

{"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. 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. 3, 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 3 zeigt eine resultierende Datenzeile.

**Abbildung 3:** 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 3 auch Kurznachrichten vom 04.01 und 05.01.2014.

## 2. Textbereinigung

Um die Tweets für eine Weiterverarbeitung durch ein großes 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 3 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 4 ist die bereinigte Form der Tweets-Spalte aus Abbildung 3 zu sehen.

**Abbildung 4:** 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. 2.3.3). 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') * sentiment['score'],
19             float(sentiment['label'] == 'negative') *
                sentiment['score']]
20     ]
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, 0]
26
27     # Rückgabe des Durchschnitts über alle Tweets an diesem
        Tag als Dictionary
28     return {
29         'positive': res[0],
30         'negative': res[1]
31     }

```

Zusammenfassend wird für jede Tweet-Liste ein Durchschnittswert für die Metriken "positiv" und "negative" berechnet. Diese Werte werden in den Datenkorpus eingefügt (Abb. 5).

#### 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 [7, 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 8 – 9 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 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 (5).

### Abbildung 5: GOOG-Datenzeile im Datenkorpus

```

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,G00G,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 anhand der GOOG-Aktie.

**Experiment 2:** Bei dieser Analyse werden erneut beide Modelle verglichen, jedoch unter Einbeziehung aller Aktiendaten.

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

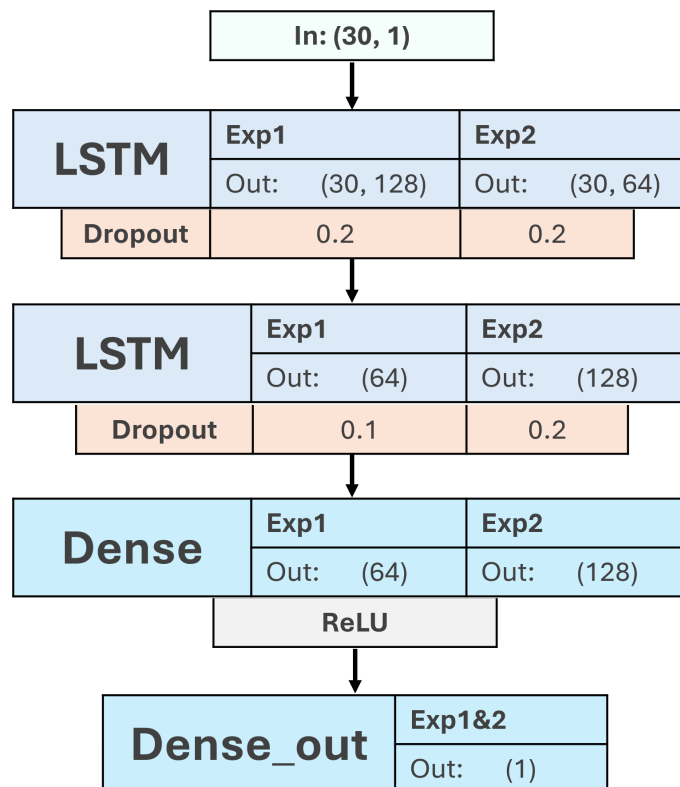
#### 3.3.1 Generelle Hyperparameter

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`. 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 [8, Tabelle 3]. Als Verlustfunktion dient typisch für Regressionsprobleme der MSE, welcher sich aus Modellausgabe ( $\hat{y}_i$ ) und tatsächlichem Preis ( $y_i$ ) ergibt (Kap. 2.3.5, Formel 1). Die Batch-Größe während des Trainings beträgt 64 und als Lernalgorithmus wird Adam-Optimierung eingesetzt (Kap. 2.3.1). 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 Long Short Term Memory (LSTM) als Hauptbestandteil des Netzwerks gewählt (Kap. 2.3.2). So besteht das Modell zum einen aus LSTM- und vollständig verbundenen Schichten (Abb. 6).

**Abbildung 6:** 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 [6, 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 Folgeta-

ges (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. 2.3.1). 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 6 dargestellt, je nach Experiment unterschiedliche Parameterwerte. Die resultierende Lernrate beträgt bei beiden Experimenten 0.000574.

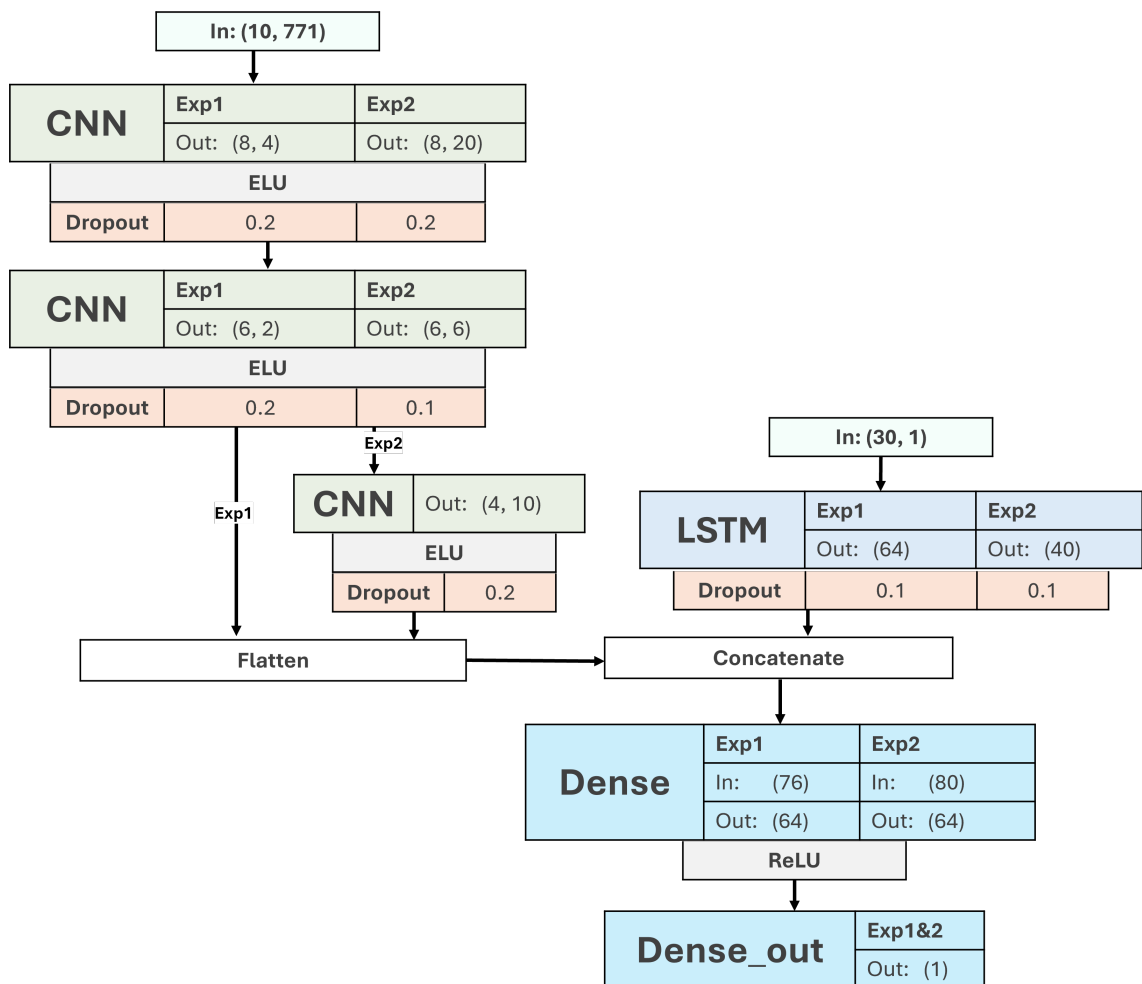
### 3.3.3 Forschungsmodell

Das Forschungsmodell besteht anfangs aus 2 Zweigen, welche im Verlauf zu einem vollständig verbundenen Netzwerk zusammenführen. Der rechte Abschnitt in Abbildung 7 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. 7). Dazu gehören die drei Metriken "positive", "negative" und "num\_tweets", sowie der Embedding-Vektor "tweet\_embs", zu sehen in Abbildung 5. 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 Convolutional Neural Network (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 [6, Kap. 8.2]. Beim Lauf der Daten durch diese Ebenen erfolgt zwischendrin jeweils zunächst die Anwendung einer 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 ei-



nem 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 7 gleichen den letzten Schritten des Basismodells (Abb. 6). 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 7:** 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 gehört zum einen der MSE als gewählte Verlustfunktion (Loss) während des Trainings, welcher anhand von Evaluierungsdaten berechnet wird und zum anderen ein Prognosediagramm und die Metriken MAE, MSE und MAPE (Kap. 2.3.5), welche sich bei der Verwendung von Testdaten ergeben. In den nachfolgenden Abschnitten werden die Ergebnisse aufgeführt und analysiert. Im Anschluss daran erfolgt eine kritische Betrachtung der Limitationen und abschließend werden gewonnene Erkenntnisse zusammengefasst.

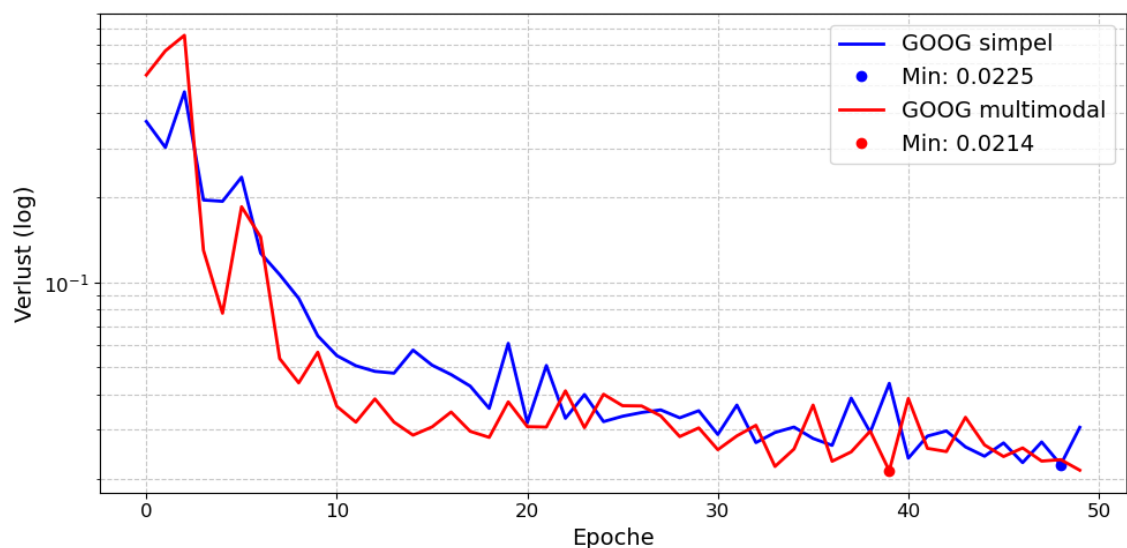
#### 3.4.1 Experiment 1: GOOG-Aktie

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

##### Verlust-Metrik

In Abbildung 8 wird der berechnete Verlust beider Modelle gegenüber gestellt.

**Abbildung 8:** Verlust-Vergleich von Basis- und Forschungsmodell, berechnet anhand von Validierungsdaten der GOOG-Aktie

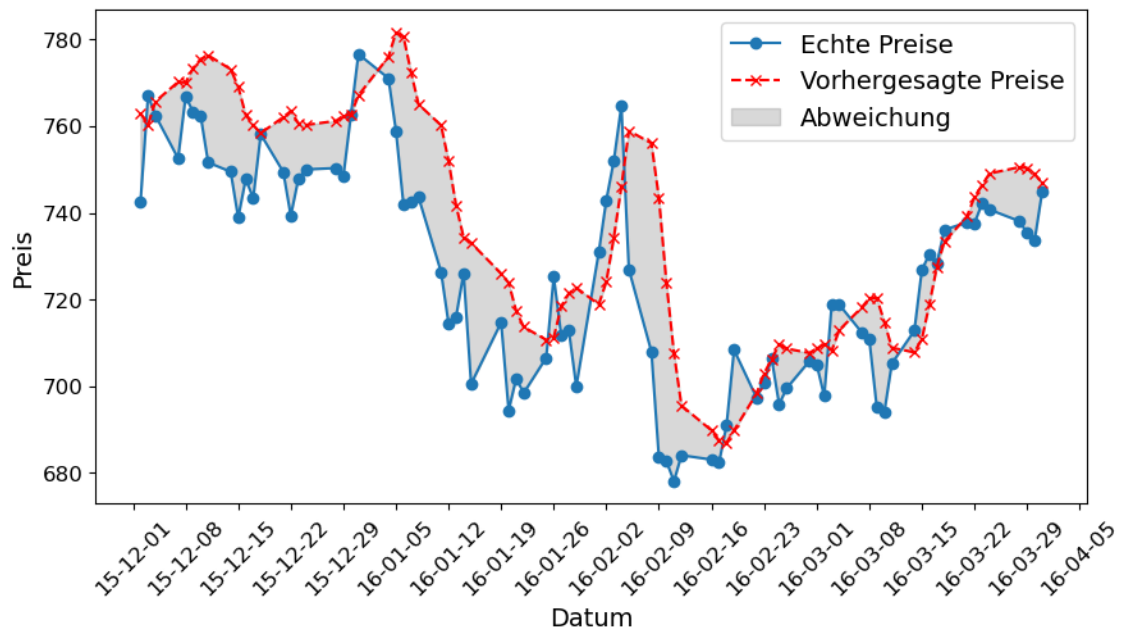


Die beiden Varianten weisen einen sehr ähnlichen Verlust-Kurvenverlauf auf. Es ist jedoch klar zu erkennen, dass die multimodale Variante leicht bessere Werte erreicht. Ihr Bestwert beträgt  $2.14 \cdot 10^{-2}$ , welcher um 4,9% geringer ausfällt als beim Vergleichsmodell.

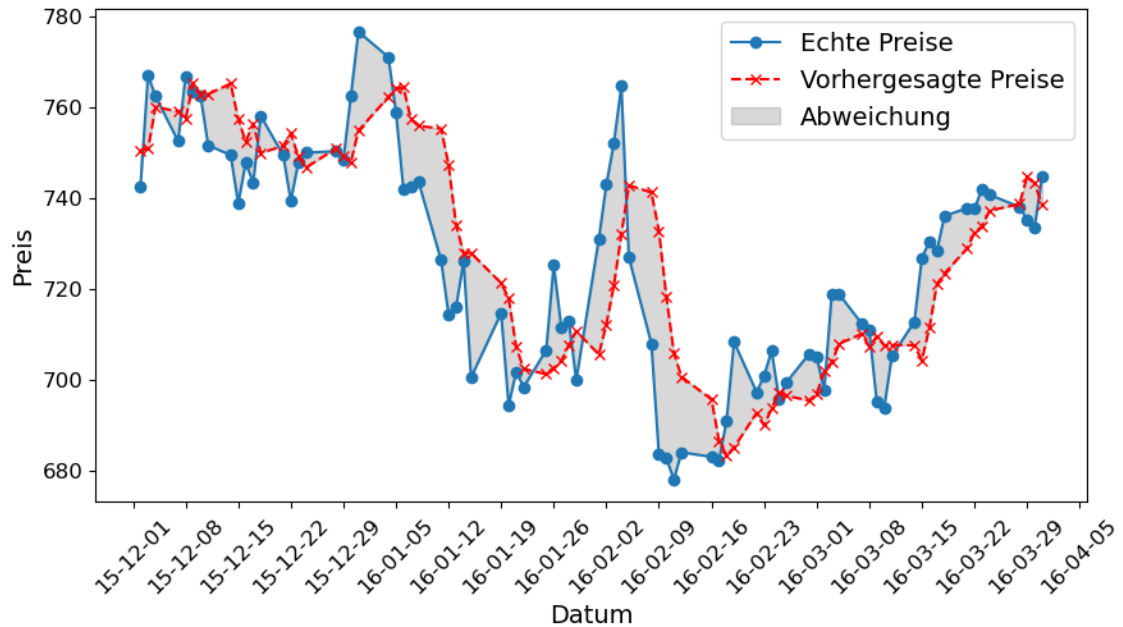
## Prognose-Metriken

In Abbildung 9 und 10 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 9:** Basismodell - echter vs vorhergesagter Preis basierend auf Tagesprognosen



**Abbildung 10:** 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 "22-12-2015" bis "29-12-2015". In Abbildung 10 gibt es in diesem Bereich einen deutlich geringeren Anteil an Graufäche, als es bei Abbildung 9 der Fall ist.

Anhand der Metrik-Tabelle 1 werden Qualitätsunterschiede deutlicher.

**Tabelle 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 Validierungsmetriken des Forschungsmodells weisen demnach eine messbare Verbesserung im Vergleich zum Basismodell auf.

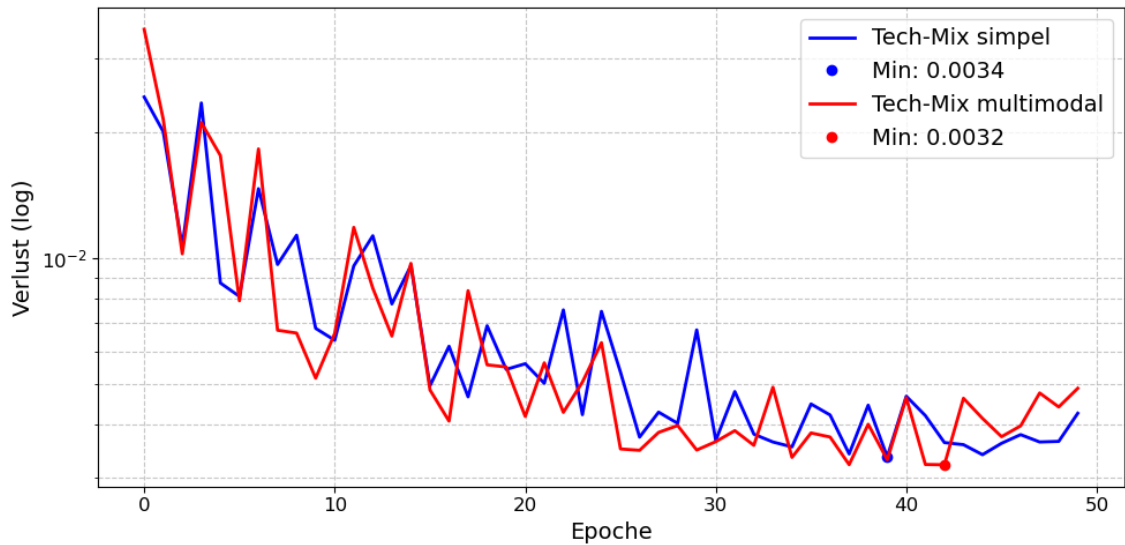
### 3.4.2 Experiment 2: Technologie-Aktienmix

In Experiment 2 werden Basis- und Forschungsmodell anhand von Daten mehrerer Aktien der Technologie-Branche trainiert.

## Verlust-Metrik

In Abbildung 11 wird der berechnete Verlust beider Modelle gegenüber gestellt.

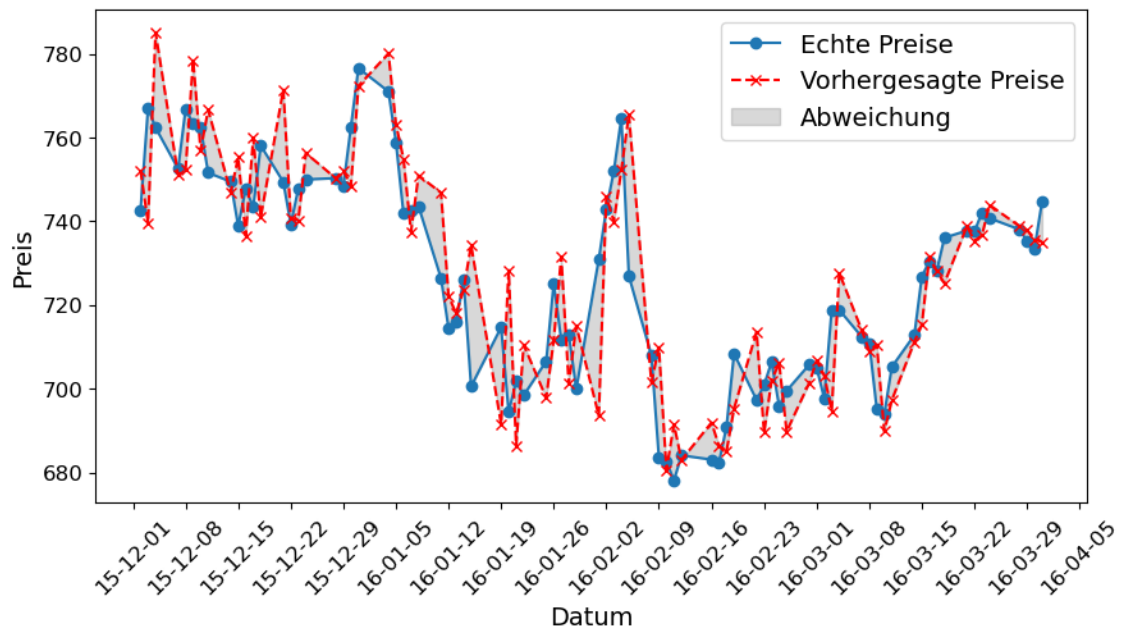
**Abbildung 11:** Verlust-Vergleich von Basis- und Forschungsmodell, berechnet anhand der Validierungsdaten von Technologieaktien



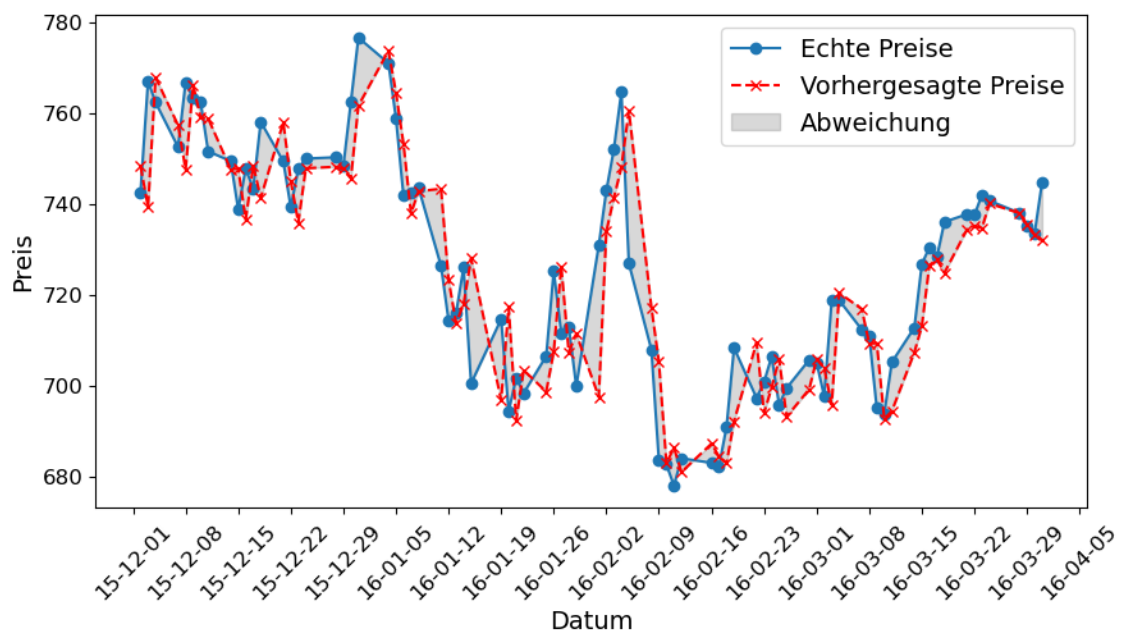
Bereits das Basismodell weist deutlich bessere Verlust-Werte auf als beide Modelle aus Experiment 1. Die multimodale Variante benötigt nun zur Erreichung des Bestwerts mehr Trainingsepochen als das Vergleichsmodell und weist erneut ein leicht besseres Ergebnis auf. Ihr minimaler Verlust beträgt  $3.2 \cdot 10^{-3}$  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 Abbildung 12 und 13 wird daher erneut, bezogen auf die GOOG-Aktie, der echte Preis mit Vorhersagen verglichen.

**Abbildung 12:** Basismodell - Echte vs Vorhergesagte Preise

Quelle: Eigene Darstellung

**Abbildung 13:** Forschungsmodell - Echte vs Vorhergesagte Preise

Quelle: Eigene Darstellung

Beim direkten Vergleich beider Diagramme fällt auf, dass in Diagramm 12 die Positionen vorhergesagter Preise deutlich stärker in vertikaler Richtung abweichen als in Abbildung 13. Das Forschungsmodell überzeugt also mit Preisvorhersagen nahe am echten Wert. Mit Hilfe der Metrik-Tabelle 2 werden Qualitätsunterschiede noch deutlicher.

**Tabelle 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 Verlust-Metrik angedeutet, erreicht das Basismodell bessere Werte als beide Modelle aus Experiment 1. Doch auch in diesem Fall konnte mit Hilfe des Forschungsmodells eine Verbesserung verzeichnet werden.

### 3.4.3 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. Eine weitere substanzielle Limitation besteht darin, dass lediglich der Preis eines Folgetages anhand von echten Preisen 30 vorangegangener Markttage vorhergesagt wurde. Die Modelle wurden nicht auf mehrtägige Prognosen getestet.

### 3.4.4 Erkenntnisse

Anhand der Experimente dieser Arbeit lassen sich nützliche Erkenntnisse ableiten. Im Vergleich zum Basismodell aus Experiment 1 konnten Verbesserungen sowohl durch Integration von Stimmungsdaten (Exp. 1 Forschungsmodell) als auch Aktienbündel (Exp. 2 Basismodell) erzielt werden. Zweiteres brachte jedoch den größeren Erfolg. Eine tiefgründige Extraktion von Informationen bezüglich komplexer Beziehungen zwischen Wertpapieren kann daher als sehr gewinnbringend für Aktienprognosen eingestuft werden. Bei zusätzlicher Hinzunahme von Stimmungsdaten wurden die Ergebnisse noch weiter verbessert (Exp. 2 Forschungsmodell).

## **3.5 Deployment**



## 4 Fazit & Zukünftige Arbeiten

Bei der Trendanalyse ist zu beachten, dass aktuell veröffentlichte Patente (2024-Q2) Innovationen der jüngsten Vergangenheit (bei ungeprüften i.d.R. 18 Monate) darstellen. Die zeitliche Dimension in den Abbildungen ??-?? und ??-?? ist daher im Hinblick auf Innovationsstärke leicht verzerrt.

Die Tabelle ?? im Kapitel 3.1 macht deutlich, dass China mit Abstand die meisten Innovationen im Robotik-Kontext hervorbringt. Der aktuelle Trend weist jedoch einen plötzlichen, deutlichen Rückgang auf. Diese Entwicklung ist ungewöhnlich und die Gründe dafür können vielfältig sein. Es kann zum Beispiel möglich sein, dass chinesische Unternehmen zunehmend mehr Wert darauf legen, Innovationen im Bereich Robotik und KI vollständig verdeckt zu halten, ähnlich zu der Vorgehensweise von OpenAI mit GPT-4 [vincent2023openai].

Europa ist, wie zu erwarten, das Schlusslicht. Auch nach Anwendung eines, auf Beschäftigtenzahlen je Region basierenden, Äquivalenzfaktors blieb die Überlegenheit Chinas in nahezu allen Bereichen bestehen (Abb. ??, ?? u. ??). Die Analysen ergaben, dass die USA im Bereich Medizin und Teleoperation im Vergleich zu Europa und China am intensivsten an Neuerungen gearbeitet hatten, so dass dort der größte gewichtete Anteil an Patenten veröffentlicht wurde.

Die Überlegenheit Chinas bei der Patentanzahl muss nicht zwingend bedeuten, dass dieses Land die größte Innovationskraft aufweist. In dieser Arbeit wurden veröffentlichte Patente einbezogen, aber deren Neuartigkeit ist nicht zwangsläufig geprüft und auch eine Bewertung des Einfallsreichtums ist nicht gegeben. So kann es durchaus sein, dass in den USA bei Patenten mehr Wert auf Qualität bzw. Einfallsreichtum der Erfindung gelegt wird, so dass banalere Neuerungen bei der Prüfung abgelehnt werden.

---

## Quellenverzeichnis

- [1] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang und P. Liu, „Transformer-based attention network for stock movement prediction,“ *Expert Systems with Applications*, Jg. 202, 2022. DOI: 10.1016/j.eswa.2022.117033. Adresse: <https://www.sciencedirect.com/science/article/abs/pii/S0957417422006170?via%3Dihub>.
- [2] Y. Xu und S. B. Cohen, „Stock Movement Prediction from Tweets and Historical Prices,“ *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, I. Gurevych und Y. Miyao, Hrsg., S. 1970–1979, Juli 2018. DOI: 10.18653/v1/P18-1183. Adresse: <https://aclanthology.org/P18-1183>.
- [3] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh und A. Talwalkar. „Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization.“ *Journal of Machine Learning Research*, Volume 18, Pages 152, besucht am 20. Dez. 2024. Adresse: <https://jmlr.org/papers/v18/16-558.html>.
- [4] K. Team. „Adam Optimizer - Keras 3 API Documentation.“ Keras API documentation, besucht am 3. Jan. 2025. Adresse: <https://keras.io/api/optimizers/adam/>.
- [5] D. P. Kingma und J. Ba, „Adam: A Method for Stochastic Optimization,“ *arXiv preprint arXiv:1412.6980*, Jg. v9, S. 1–15, Dez. 2014, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. DOI: 10.48550/arXiv.1412.6980. Adresse: <https://arxiv.org/abs/1412.6980>.
- [6] Y. Guan, P. Li und C. Lu, „Stock Price Prediction with CNN-LSTM Network,“ *University of Toronto Publications*, Jan. 2020.
- [7] A. R. Hazourli, „FinancialBERT - A Pretrained Language Model for Financial Text Mining,“ *Preprint on ResearchGate*, Feb. 2022, CC BY 4.0 License. DOI: 10.13140/RG.2.2.34032.12803.
- [8] L. Xie, Z. Chen und S. Yu, „Deep Convolutional Transformer Network for Stock Movement Prediction,“ *Electronics*, Jg. 13, S. 4225, Okt. 2024, CC BY 4.0 License. DOI: 10.3390/electronics13214225. Adresse: [https://www.researchgate.net/publication/385328306\\_Deep\\_Convolutional\\_Transformer\\_Network\\_for\\_Stock\\_Movement\\_Prediction](https://www.researchgate.net/publication/385328306_Deep_Convolutional_Transformer_Network_for_Stock_Movement_Prediction).
- [9] H. Xu, L. Chai, Z. Luo und S. Li, „Stock movement predictive network via incorporative attention mechanisms based on tweet and historical prices,“ *Neurocomputing*, Jg. 418, S. 326–339, 2020. DOI: 10.1016/j.neucom.2020.07.109. Adresse: <https://www.sciencedirect.com/science/article/abs/pii/S0925231220313060?via%3Dihub>.
- [10] Y. Xu und dtaylor-530. „StockNet Dataset: A Comprehensive Dataset for Stock Movement Prediction from Tweets and Historical Stock Prices.“ MIT License, besucht am 20. Dez. 2024. Adresse: <https://github.com/yumoxu/stocknet-dataset>.

---

## 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)

Dieburg, 14.02.2025

---

(Ort, Datum)



---

(Alexander Lahn, Unterschrift)