



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 : 1. Januar 2025

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Abkürzungsverzeichnis	III
1 Einleitung	1
2 Theoretische Grundlagen	2
2.1 Datenverständnis	2
2.2 Aktienprognosen mit Deep Learning	2
2.2.1 LSTM für Kursdaten	2
2.2.2 LLM für Stimmungsdaten	2
2.2.3 CNN für Stimmungsdaten	2
2.2.4 Evaluierungsmetriken	2
3 Methodik	4
3.1 Datenbeschaffung	4
3.2 Datenvorverarbeitung	5
3.3 Modellierung	10
3.4 Evaluierung	12
3.4.1 Experiment 1: GOOG-Aktie	12
3.4.2 Experiment 2: Technologie-Aktienmix	14
3.4.3 Limitationen	16
3.4.4 Erkenntnisse	17
3.5 Deployment	18
4 Fazit & Zukünftige Arbeiten	19
Quellenverzeichnis	20

Abbildungsverzeichnis

1	GOOG Kursdaten	5
2	GOOG Tweet vom 01.01.2014	5
3	GOOG Datenzeile nach Schritt 1	6
4	GOOG Datenzeile nach Schritt 2	6
5	GOOG Datenzeile im Datenkorpus	9
6	Verlust-Vergleich von Basis- und Forschungsmodell	12
7	Basismodell - Echte vs Vorhergesagte Preise	13
8	Forschungsmodell - Echte vs Vorhergesagte Preise	13
9	Verlust-Vergleich von Basis- und Forschungsmodell	14
10	Basismodell - Echte vs Vorhergesagte Preise	15
11	Forschungsmodell - Echte vs Vorhergesagte Preise	16

Abkürzungsverzeichnis

MAE	Mean Absolute Error
MSE	Mean Squared Error
MAPE	Mean Absolute Percentage Error
CRISP-DM	Cross Industry Standard Process for Data Mining

1 Einleitung

Bereits seit Jahrzehnten wird dem Bereich Aktienprognose von Wissenschaftlern als auch Investoren große Aufmerksamkeit gewidmet¹. 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 Modellen beitragen².

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 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.

¹ Zhang, Q. et al., 2022, Kap. Introduction.

² Ebd., Kap. Introduction.

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).

___³

2.2 Aktienprognosen mit Deep Learning

2.2.1 LSTM für Kursdaten

2.2.2 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. ...

___⁴

2.2.3 CNN für Stimmungsdaten

2.2.4 Evaluierungsmetriken

- Mean Absolute Error (MAE), Mean Squared Error (MSE) und Mean Absolute Percentage Error (MAPE)⁵

³ Xu, Y., Cohen, S. B., 2018, Kap. 3.

⁴ Hazourli, A. R., 2022.

⁵ Xie, L., Chen, Z., Yu, S., 2024, 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 Cross Industry Standard Process for Data Mining (CRISP-DM).

3.1 Datenbeschaffung

Aufgrund dessen, dass Aktienprognosen mit multimodalen Daten ein sehr belebtes Forschungsfeld ist, gibt es bereits viele sinnvoll zusammengestellte Datensätze. Um diesen Vorteil auszuschöpfen wird ein anerkannter Finanzdatensatz gewählt, welcher in mehreren Wissenschaftsarbeiten zum Einsatz kam^{6,7,8}. Die Daten werden auf GitHub unter der MIT-Lizenz zur Verfügung gestellt⁹.

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 beziehen¹⁰. 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`-Datei vor. In Abbildung 1 werden die inkludierten Metriken dargestellt.

⁶ Xu, Y., Cohen, S. B., 2018.

⁷ Xu, H. et al., 2020.

⁸ Zhang, Q. et al., 2022.

⁹ Xu, Y., dtaylor-530, 2017.

¹⁰ Xu, Y., Cohen, S. B., 2018, Kap. 3.

Abbildung 1: GOOG Kursdaten

```

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

```

Quelle: Eigene Darstellung

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 werden.

Abbildung 2: GOOG Tweet vom 01.01.2014

```

{"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"}

```

Quelle: Eigene Darstellung

3.2 Datenvorverarbeitung

Die Datenvorverarbeitung erfolgt logisch in 4 Teilschritten. Während des 1. Arbeitsschritts werden die Daten aller Aktien in eine `csv`-Datei zusammengefasst, wobei zur Unterscheidung eine zusätzliche Datenspalte mit Marktkürzel als Werte hinzugefügt wird (Abb. 3, Spalte "stock").

1. Zeitliche Ausrichtung der Daten

In diesem Arbeitsschritt werden pro Aktie Kursdaten und Tweets zu einer `csv`-Datei zusammengefasst. Dabei werden zunächst alle Kursdaten-Einträge entfernt, dessen 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 Bereichs der Kursdaten befindet. Abbildung 3 zeigt eine resultierende Datenzeile.

Abbildung 3: GOOG Datenzeile nach Schritt 1

```

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

```

Quelle: Eigene Darstellung

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 symantische Informationen verloren gehen. Stopp-Wörter wie beispielsweise "und" werden daher nicht entfernt. In Abbildung 3 ist zu erkennen, dass in Tweets häufig URL's eingebunden sind. Diese werden mit "URL" maskiert. Des weiteren treten regelmäßig Referenzierungen mittel "@" auf. Diese werden mit der Maskierung "AT_ENTITY" ersetzt. Außerdem erfolgt eine Substituierung von Aktienvorkommen der Form "\$<Kürzel>" mit "<Kürzel> stock". In Abbildung 4 ist die bereinigte Form der Tweets-Spalte aus Abbildung 3 zu sehen.

Abbildung 4: GOOG Datenzeile nach Schritt 2

```

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

```

Quelle: Eigene Darstellung

3. Stimmungsmetriken

– The maximum numbers of messages and words in a single message are set to 30 and 40, respectively. –¹¹

¹¹ Zhang, Q. et al., 2022, Kap. 5.2.

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)
```

Quelle: Eigene Darstellung

Anschließend wird für jede Tweet-Liste eine durchschnittliche Quantifizierung der Metriken "positiv" und "negative" berechnet. Hierfür wird das vortrainierte Sprachmodell "Financial-BERT" eingesetzt (Kap. 2.2.2). 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 Sentiment Metriken über eine
    Tweet-Liste
9 def comp_sent(texts):
10     sent_res = []
11     for text in texts:
12         res = nlp(text)
13         # Klassifikation-Resultat 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     }

```

Quelle: Eigene Darstellung

4. Stimmung-Embeddings

Zusätzlich zu Stimmungsmetriken soll pro Tweet-Liste ein repräsentative 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 wurde¹². 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

```

¹² Hazourli, A. R., 2022, Kap. 5.3.

```

7  inputs = tokenizer(
8  text,
9  padding=True,
10 truncation=True,
11 return_tensors="pt"
12 )

```

Quelle: Eigene Darstellung

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 ID's 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, ...]"

```

Quelle: Eigene Darstellung

3.3 Modellierung

In dieser Arbeit geht es darum den Einfluss von Stimmungsdaten bei 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 wird der gleiche Unterschied betrachtet, 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. Diese werden jeweils mit Hilfe der Python-Bibliothek `keras` entwickelt.

Generelle Hyperparameter

Es gibt eine Vielzahl an Hyperparameter, 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 unerwünschten 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 ermittelt¹³. In Anlehnung an eine weitere Forschungsarbeit wird für Stimmungsdaten ein kleineres Fenster von 10 Markttagen verwendet¹⁴. Als Verlustfunktion dient typisch für Regressionsprobleme der MSE, welcher

¹³ Xie, L., Chen, Z., Yu, S., 2024, Tabelle 3.

¹⁴ Zhang, Q. et al., 2022, Kap. 4.1.

sich aus Modellausgabe (\hat{y}_i) und tatsächlichem Preis (y_i) ergibt (Kap. 2.2.4, Formel 1). Die Batch-Größe während des Trainings beträgt 64.

Basismodell

D

- lstm
- spezifische hyper paramter optimierung mit keras

Forschungsmodell

- Basismodell (etwas abgewandelt) + CNN
- spezifische hyper paramter optimierung mit keras
- arch unterschied bei 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. 2.2.4), welche sich anhand 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.

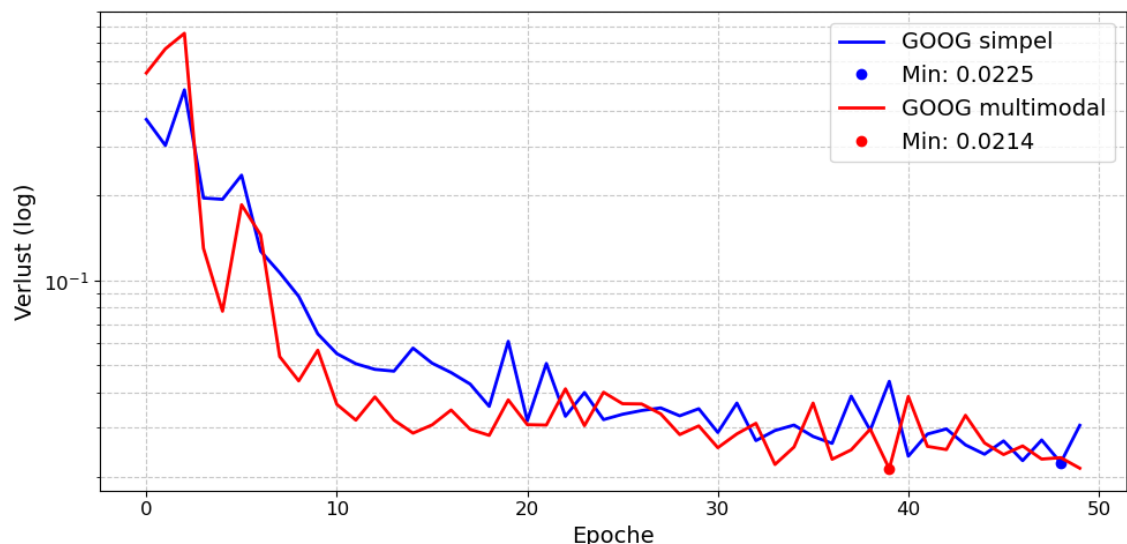
3.4.1 Experiment 1: GOOG-Aktie

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

Verlust-Metrik

In Abbildung 6 wird der berechnete Verlust beider Modelle gegenüber gestellt.

Abbildung 6: Verlust-Vergleich von Basis- und Forschungsmodell



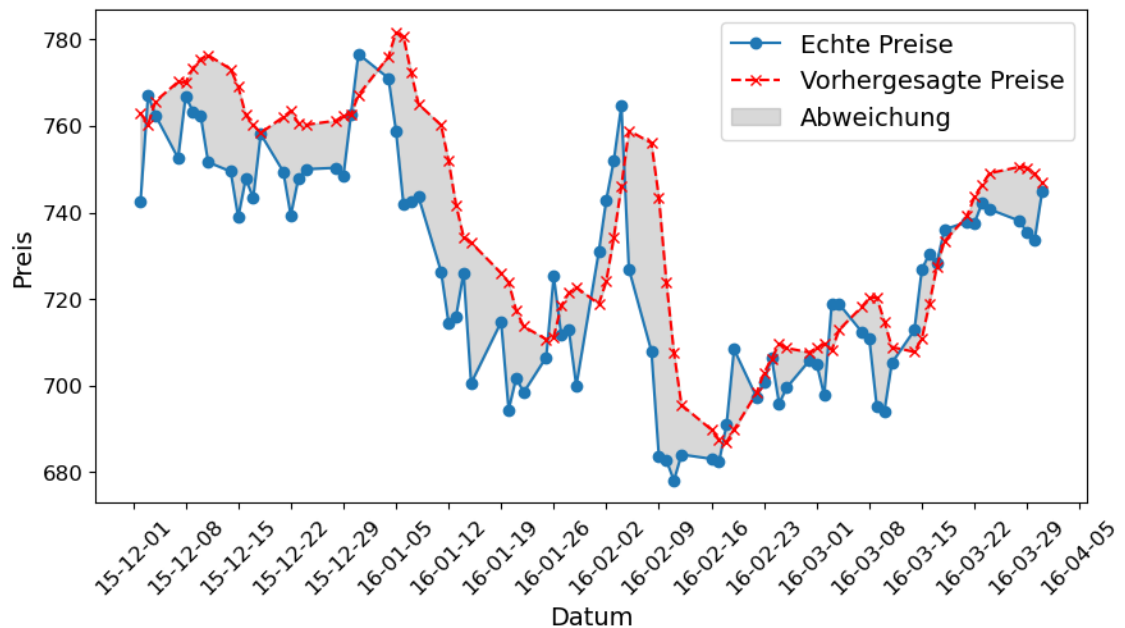
Quelle: Eigene Darstellung

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

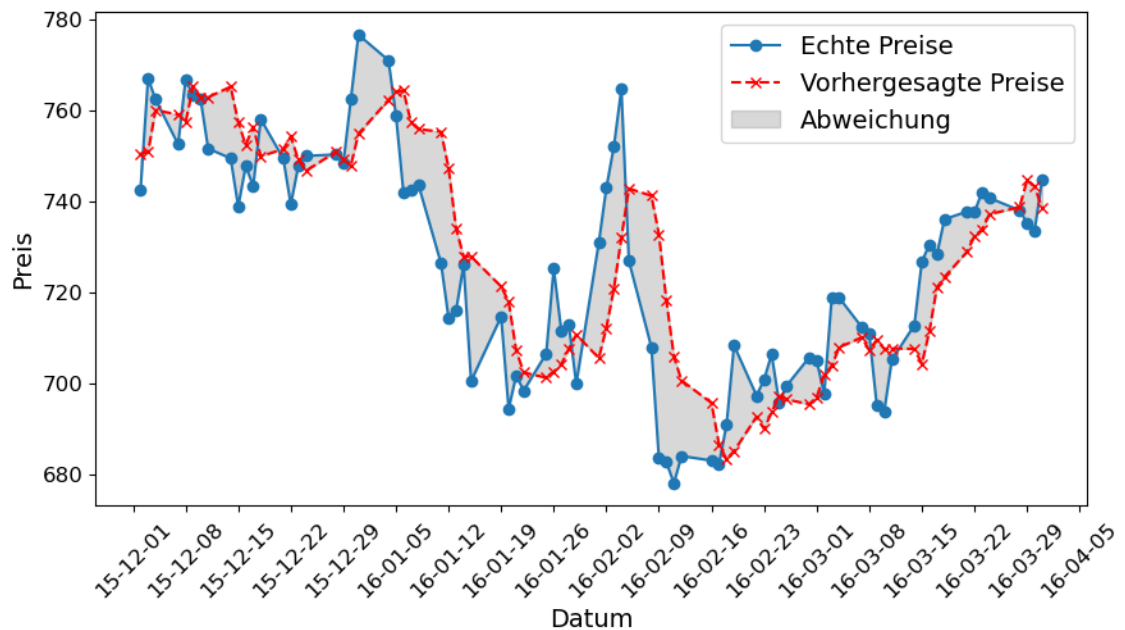
In Abbildung 7 und 8 wird der tatsächliche Preis mit der Modellvorhersage verglichen.

Abbildung 7: Basismodell - Echte vs Vorhergesagte Preise



Quelle: Eigene Darstellung

Abbildung 8: Forschungsmodell - Echte vs Vorhergesagte Preise



Quelle: Eigene Darstellung

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 8 gibt es in diesem Bereich einen deutlichen geringeren Anteil an Graufäche, als es bei Abbildung 7 der Fall ist.

Anhand der Metrik-Tabelle 1 werden Qualitätsunterschiede deutlicher.

Tabelle 1: Metrik-Vergleich

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 nach.

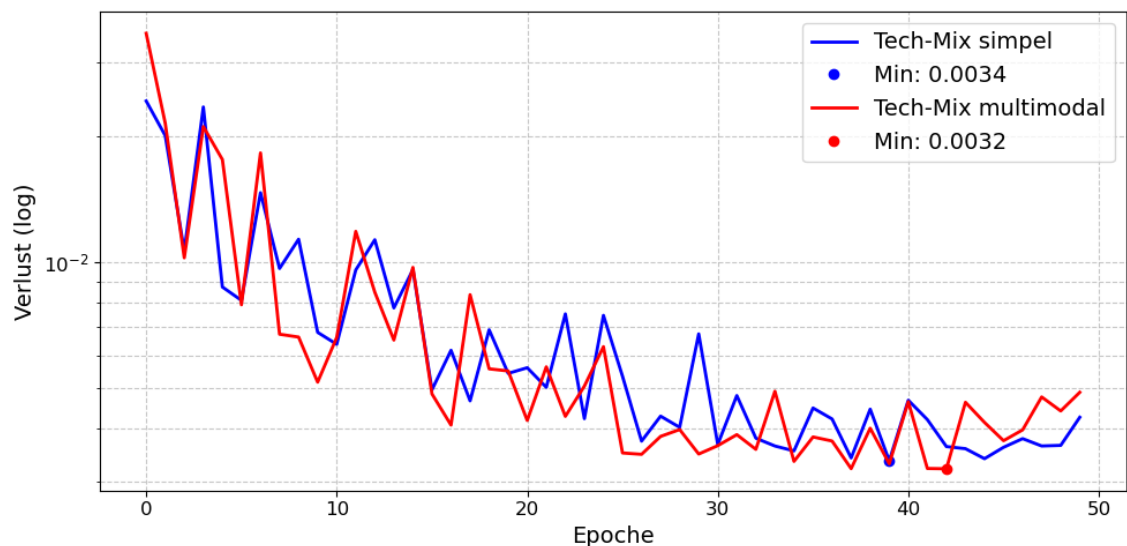
3.4.2 Experiment 2: Technologie-Aktienmix

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

Verlust-Metrik

In Abbildung 9 wird der berechnete Verlust beider Modelle gegenüber gestellt.

Abbildung 9: Verlust-Vergleich von Basis- und Forschungsmodell



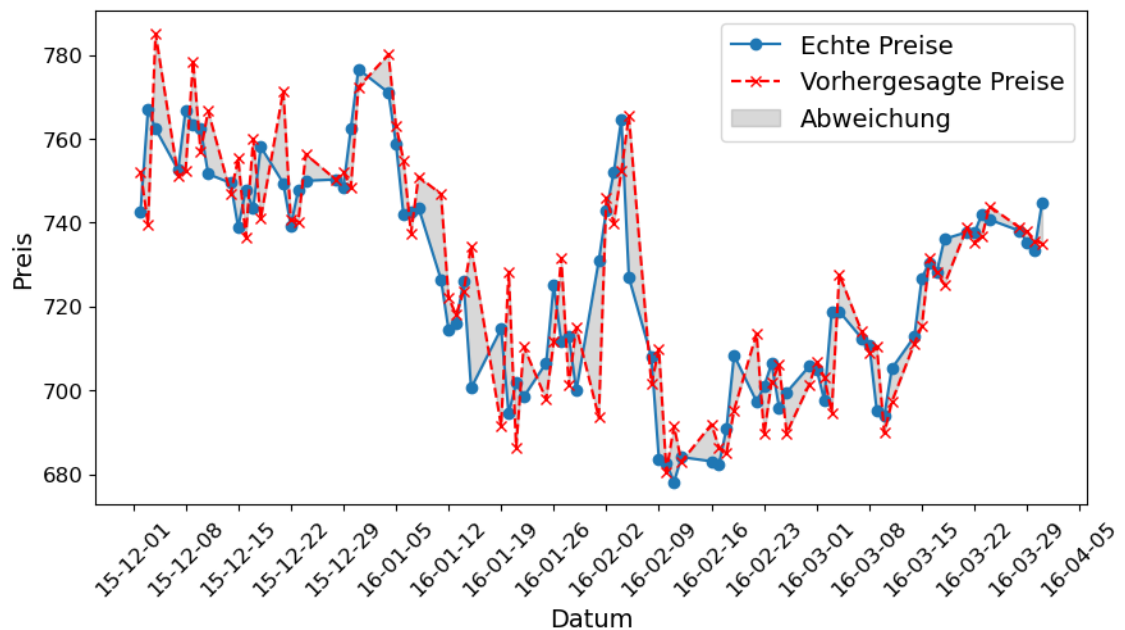
Quelle: Eigene Darstellung

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.

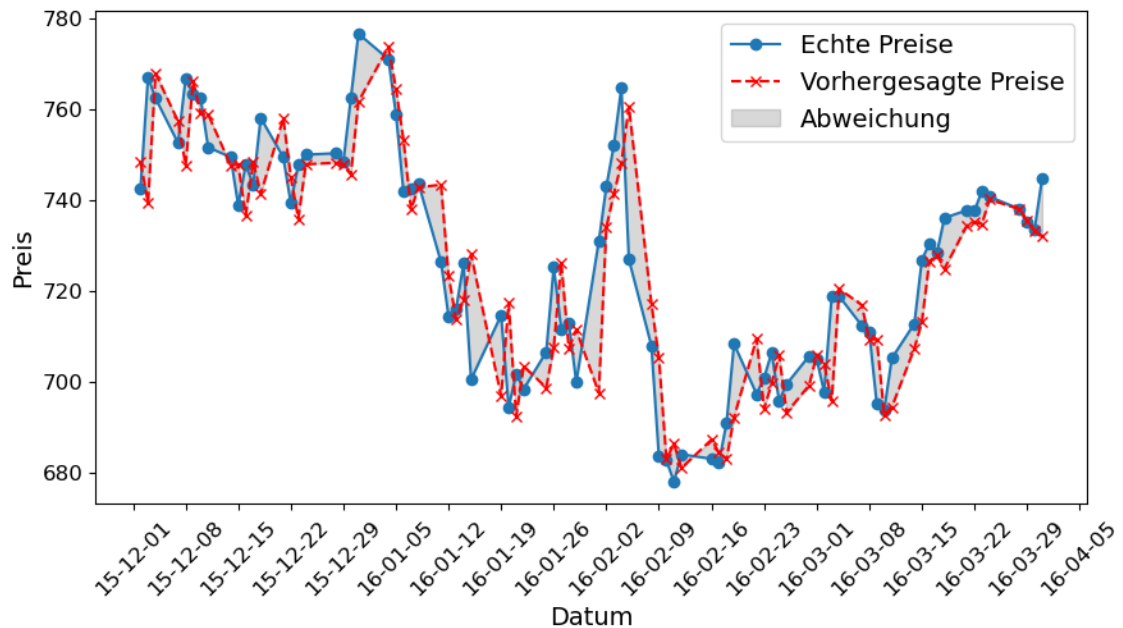
Prognose-Metriken

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 10 und 11 wird daher erneut, bezogen auf die GOOG-Aktie, der echte Preis mit Vorhersagen verglichen.

Abbildung 10: Basismodell - Echte vs Vorhergesagte Preise



Quelle: Eigene Darstellung

Abbildung 11: Forschungsmodell - Echte vs Vorhergesagte Preise

Quelle: Eigene Darstellung

Beim direkten Vergleich beider Diagramme fällt auf, dass in Diagramm 10 die Positionen vorhergesagter Preise deutlich stärker in vertikaler Richtung abweichen als in Abbildung 11. Das Forschungsmodell überzeugt also mit Preisvorhersagen nahe am echten Wert. Mit Hilfe der Metrik-Tabelle 2 werden Qualitätsunterschiede noch deutlicher.

Tabelle 2: Metrik-Vergleich

Modell	MAE	MSE	MAPE (%)
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-4¹⁵.

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.

¹⁵ [vincent2023openai](#).

Quellenverzeichnis

- Hazourli, Ahmed Rachid* (2022): FinancialBERT - A Pretrained Language Model for Financial Text Mining, in: Preprint on ResearchGate (2022), CC BY 4.0 License
- Xie, Li, Chen, Zhengming, Yu, Sheng* (2024): Deep Convolutional Transformer Network for Stock Movement Prediction, in: Electronics, 13 (2024), CC BY 4.0 License, S. 4225
- Xu, Hongfeng, Chai, Lei, Luo, Zhiming, Li, Shaozi* (2020): Stock movement predictive network via incorporative attention mechanisms based on tweet and historical prices, in: Neurocomputing, 418 (2020), S. 326–339
- Xu, Yumo, Cohen, Shay B.* (2018): Stock Movement Prediction from Tweets and Historical Prices, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (2018), hrsg. von *Gurevych, Iryna, Miyao, Yusuke*, S. 1970–1979
- Xu, Yumo, dtaylor-530* (2017): StockNet Dataset: A Comprehensive Dataset for Stock Movement Prediction from Tweets and Historical Stock Prices, MIT License, <<https://github.com/yumoxu/stocknet-dataset>> (2017) [Zugriff: 2024-12-20]
- Zhang, Qiuyue, Qin, Chao, Zhang, Yunfeng, Bao, Fangxun, Zhang, Caiming, Liu, Peide* (2022): Transformer-based attention network for stock movement prediction, in: Expert Systems with Applications, 202 (2022)