**Analyse der Marktstimmung von Kryptowährungen durch Reddit-Daten und Sentiment-Analyse**

Digital Business University of Applied Sciences

Data Science & Business Analytics  
PDA91 – SPIII: Praxisprojekt

**Von Dennis Reimer. Martrikelnummer: 190288**

**Zusammenfassung**  
Diese Arbeit untersucht die Analyse der Marktstimmung von Kryptowährungen auf Basis von Reddit-Diskussionen. Durch den Einsatz von Scraping-Techniken, Natural Language Processing (NLP) und Sentiment-Analyse mit CryptoBERT wird eine automatisierte Pipeline zur Erhebung und Bewertung von Stimmungsdaten entwickelt. Die ermittelten Stimmungen werden anschließend mit historischen Preisdaten von CoinGecko korreliert, um potenzielle Zusammenhänge zwischen öffentlicher Wahrnehmung und Marktbewegungen zu identifizieren. Die Umsetzung erfolgt mit einer automatisierten CI/CD-Pipeline in Jenkins und einer interaktiven Visualisierung der Ergebnisse mit Streamlit.

Inhalt

[1 Einleitung 2](#_Toc195557367)

[2 Technischer Hintergrund 2](#_Toc195557368)

[3 Implementierung und Methodik 3](#_Toc195557369)

[3.1 3](#_Toc195557370)

[3.2 Modellvergleich 4](#_Toc195557371)

[3.3 Modelloptimierung 6](#_Toc195557372)

[3.4 Label-Mapping für CryptoBERT 8](#_Toc195557373)

[4 Ergebnisse und Diskussion 9](#_Toc195557374)

[4.1 Baseline-Vergleich aller Modelle 9](#_Toc195557375)

[4.2 Optimierungsergebnisse: CryptoBERT und FinBERT 10](#_Toc195557376)

[4.2.1CryptoBERT auf Kommentaren (nach Fine-Tuning) 10](#_Toc195557377)

[4.2.2FinBERT auf Posts (nach Fine-Tuning) 11](#_Toc195557378)

[4.3 Herausforderungen und Grenzen 12](#_Toc195557379)

[5 Fazit 13](#_Toc195557380)

[5.1 Ausblick 14](#_Toc195557381)

1 Einleitung

Die Analyse von Marktstimmungen auf Basis sozialer Medien hat in den letzten Jahren zunehmend an Bedeutung gewonnen, insbesondere im Kontext volatiler Finanzmärkte wie dem der Kryptowährungen. Reddit stellt dabei eine besonders relevante Quelle dar, da es als Plattform für ungefilterte Meinungsäußerung dient und intensive Diskussionen über einzelne Kryptowährungen ermöglicht. Im Rahmen dieser Studienarbeit wird untersucht, wie sich Natural Language Processing (NLP)-Modelle zur Sentimentanalyse von Reddit-Daten systematisch vergleichen, evaluieren und verbessern lassen.

Im Fokus stehen dabei vortrainierte Sprachmodelle wie FinBERT, DeBERTa, TextBlob, VADER und CryptoBERT (Elkulako), die hinsichtlich ihrer Klassifikationsleistung auf Reddit-Posts und -Kommentaren analysiert werden. Ziel ist es, zunächst eine fundierte Vergleichsbasis der Modelle zu schaffen und darauf aufbauend gezielt Optimierungen vorzunehmen. Dabei wird das Modell CryptoBERT speziell für Kommentare weiterentwickelt, während FinBERT für längere Post-Texte optimiert wird.

Die zugrunde liegende Forschungsfrage lautet:  
Wie gut lassen sich marktbezogene Sentiments aus Reddit-Daten mithilfe spezialisierter NLP-Modelle erkennen, und wie kann ihre Genauigkeit durch gezielte Anpassungen verbessert werden?

Durch eine datengetriebene Herangehensweise und ein systematisches Fine-Tuning ausgewählter Modelle soll ein praxisnaher Beitrag zur automatisierten Stimmungsanalyse im Finanzbereich geleistet werden.

2 Technischer Hintergrund

Die Sentimentanalyse im Finanzbereich stellt besondere Anforderungen an die verwendeten Natural Language Processing (NLP)-Modelle, da finanzbezogene Begriffe oft mehrdeutig sind und in spezifischen Kontexten verwendet werden. Klassische Modelle wie VADER oder TextBlob, die auf allgemeinen Sprachmustern basieren, stoßen bei Reddit-Daten aus dem Kryptomarkt schnell an ihre Grenzen. Um präzisere Ergebnisse zu erzielen, kommen daher spezialisierte, vortrainierte Modelle wie FinBERT und CryptoBERT (Elkulako) zum Einsatz, die auf finanzspezifischen Textcorpora trainiert wurden.

Zur Analyse der Reddit-Daten wird zwischen zwei zentralen Texttypen unterschieden:

Posts: häufig längere, analytische Beiträge mit hohem Informationsgehalt,

Kommentare: kürzer, oft emotionaler, aber stark interaktionsgetrieben.

Diese Differenzierung ist entscheidend, da sie Auswirkungen auf die Wahl und Leistungsfähigkeit der eingesetzten Modelle hat. Während längere Posts tendenziell besser von Modellen mit größerem Kontextverständnis wie FinBERT verarbeitet werden, erfordern kurze und kontextarme Kommentare eine robuste Generalisierung, wie sie beispielsweise CryptoBERT bietet.

In dieser Arbeit kommen ausschließlich öffentlich verfügbare, vortrainierte Modelle aus der Hugging Face Model Library zum Einsatz. Für die Feinjustierung (Fine-Tuning) der Modelle wird die Architektur von Transformers (Wolf et al., 2020) in Kombination mit PyTorch verwendet. Die Evaluation erfolgt anhand von Standardmetriken wie Accuracy, Precision, Recall und F1-Score auf einem manuell gelabelten Testset.

Um die Klassifikation zu optimieren und eine bessere Generalisierung auf Reddit-spezifische Sprache zu erreichen, werden ausgewählte Modelle gezielt aufgeteilt und getrennt für Posts und Kommentare trainiert. Dieses Vorgehen erlaubt eine differenzierte Bewertung und gezielte Verbesserung der Sentimenterkennung im jeweiligen Subkontext.

3 Implementierung und Methodik

Dieses Kapitel beschreibt die technische Umsetzung der Sentimentanalyse, die Auswahl und Konfiguration der Modelle sowie die Schritte zur Modellverbesserung. Ziel war es, verschiedene NLP-Modelle systematisch zu vergleichen und anschließend durch gezieltes Fine-Tuning zu optimieren. Alle Implementierungsschritte wurden in Python unter Verwendung der Bibliotheken transformers, datasets, scikit-learn und PyTorch realisiert.

3.1

**3.1 Datengrundlage und Aufteilung**

Die Ausgangsdaten bestehen aus zwei selbst erstellten Datensätzen mit jeweils **300 manuell gelabelten Reddit-Posts** und **300 Kommentaren**. Jedes Textbeispiel wurde in eine von drei Sentiment-Kategorien eingeteilt:

* 0 = bullish (positiv),
* 1 = neutral,
* 2 = bearish (negativ).

Zur Sicherstellung einer fairen und konsistenten Modellbewertung wurde ein **stratifizierter Datensplit** durchgeführt. Dabei wurde gewährleistet, dass die Klassenverteilung in Trainings-, Validierungs- und Testset identisch bleibt – ein entscheidender Schritt bei unbalancierten Klassifikationsproblemen. Die Split-Strategie erzeugt ein Verhältnis von **60 % Training**, **20 % Validierung** und **20 % Testdaten**.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Diese eigene Funktion erlaubt es, den Split reproduzierbar und identisch über alle Modelltrainings hinweg zu gestalten. Insbesondere bei kleinen Datensätzen ist eine stratifizierte Aufteilung wichtig, da zufällige Verzerrungen sonst starke Auswirkungen auf das Modellverhalten haben können.

3.2 Modellvergleich

Ziel des ersten Umsetzungsschritts war es, die Performanz verschiedener vortrainierter Sentimentmodelle auf Reddit-Daten systematisch zu vergleichen. Dabei wurden zwei methodisch unterschiedliche Gruppen untersucht:

* **Transformerbasierte Modelle**, die tiefenlernende Sprachrepräsentationen nutzen
* **Lexikonbasierte Modelle**, die auf vordefinierten Wörterbüchern beruhen

**Transformer-Modelle**

Diese Modelle basieren auf der BERT-Architektur (Bidirectional Encoder Representations from Transformers) oder deren Weiterentwicklungen. Sie ermöglichen eine kontextabhängige Analyse von Texten durch das sogenannte Self-Attention-Mechanismus. In dieser Arbeit wurden vier verschiedene Varianten getestet:

* **FinBERT**  
  Ein auf Finanztexte spezialisiertes BERT-Modell, das auf einer großen Menge von SEC-Filings und anderen wirtschaftsbezogenen Dokumenten trainiert wurde. Es eignet sich besonders gut für längere Beiträge mit analytischem Charakter – typisch für viele Reddit-Posts.
* **CryptoBERT (ElKulako)**  
  Ein weiteres spezialisiertes Modell, das auf kryptowährungsbezogenen Diskussionen trainiert wurde. Es wurde auf Daten aus Foren, Social Media und Artikeln zu Krypto-Themen feinjustiert und ist somit inhaltlich eng an die verwendeten Reddit-Daten angelehnt.
* **DeBERTa (Decoding-enhanced BERT with disentangled attention)**  
  Eine Weiterentwicklung von BERT, die Kontext- und Positionsinformationen entkoppelt verarbeitet. Sie erreicht in vielen Benchmarks überlegene Ergebnisse, konnte in dieser Arbeit jedoch insbesondere bei kurzen Texten wie Kommentaren keine stabile Leistung zeigen.
* **RoBERTa**  
  Eine robuste Variante von BERT mit optimierten Trainingsparametern (größere Batch-Größen, mehr Trainingsdaten, dynamisches Masking). Das hier verwendete Modell wurde speziell auf Twitter-Daten trainiert und ist daher gut auf informelle Sprache ausgelegt, allerdings nicht auf Finanzinhalte spezialisiert.

Für alle Modelle wurde die Sentimentklassifikation einheitlich über das Hugging Face Trainer-Objekt umgesetzt. Die Testdaten wurden dazu mit dem jeweiligen Tokenizer vorverarbeitet und in das Dataset-Format überführt:

Ein Bild, das Text, Schrift, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

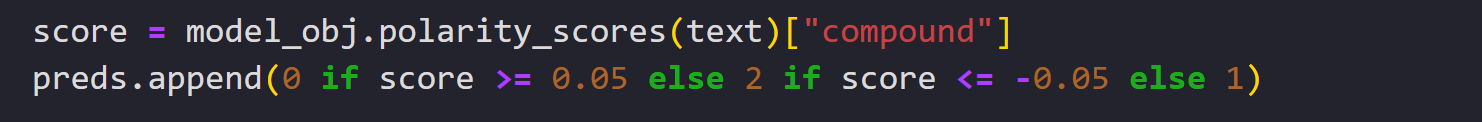
Dieser Ansatz ermöglichte eine standardisierte Batch-Inferenz für alle Transformer-Modelle. Die Ergebnisse wurden jeweils auf einem einheitlich strukturierten Testset evaluiert.

**Lexikonbasierte Modelle**

Diese Modelle beruhen auf fest definierten Wortlisten, denen bestimmte Polaritätswerte zugewiesen sind. Der Sentimentwert eines Textes ergibt sich durch Aggregation dieser Einzelwerte. Die Klassifikation erfolgt in der Regel über einfache Schwellenwerte.

* **VADER (Valence Aware Dictionary and sEntiment Reasoner)**  
  Ein speziell für soziale Medien entwickeltes Modell, das neben einem umfangreichen Lexikon auch Regeln für Emoji-Interpretation, Großschreibung und Satzzeichen berücksichtigt. Es liefert einen Polaritätswert zwischen -1 (negativ) und +1 (positiv).
* **TextBlob**  
  Ein einfaches Python-Modul, das auf einem klassischen NLP-Ansatz basiert. Es nutzt eine Sentiment-Pipeline mit einem vortrainierten Naive-Bayes-Modell für Polaritätsbewertung. TextBlob ist nicht auf Finanzsprache spezialisiert, bietet aber robuste Basisbewertungen.

Die Umrechnung der Polaritätswerte in die drei Zielklassen erfolgte über einen Schwellenwertansatz:



**Evaluation und F1-Extraktion**

Zur Vergleichbarkeit aller Modelle wurde auf Basis der vorhergesagten und der tatsächlichen Labels ein Klassifikationsreport erstellt. Daraus wurden gezielt die F1-Scores für die drei Klassen extrahiert:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Diese Auswertung ermöglichte eine differenzierte Betrachtung der Modellleistung nach Klassen und Texttypen (Posts vs. Kommentare). Die Ergebnisse werden in Kapitel 4 grafisch aufbereitet und interpretiert.

3.3 Modelloptimierung

Nach dem Modellvergleich wurden die zwei leistungsstärksten Modelle für eine gezielte Optimierung ausgewählt:

* **FinBERT** für längere Reddit-Posts
* **CryptoBERT (ElKulako)** für kurze, kontextarme Kommentare

Ziel war es, die Modellgüte durch **systematisches Fine-Tuning** weiter zu verbessern. Dazu wurden zwei zentrale Schritte implementiert:

1. Ein speziell angepasster Trainer mit gewichteter Verlustfunktion
2. Ein Vergleich mehrerer Lernraten mit Logging und Evaluation

**Custom-Trainer mit gewichteter Verlustfunktion**

Die Klassenverteilung in Reddit-Kommentaren ist häufig unausgeglichen – besonders die *bearish*-Klasse tritt seltener auf. Um dem entgegenzuwirken, wurde eine eigene Subklasse des Hugging Face Trainer definiert, in der eine **benutzerdefinierte Verlustfunktion** (loss\_fn) übergeben werden kann. Diese erlaubt z. B. die Verwendung von CrossEntropyLoss(weight=...) mit manuell definierten Klassen-Gewichtungen.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Diese Anpassung erlaubt eine differenziertere Steuerung des Lernprozesses, insbesondere bei kleineren oder verzerrten Datensätzen.

**Vergleich verschiedener Lernraten**

Ein zentraler Bestandteil des Optimierungskonzepts war ein systematischer Vergleich unterschiedlicher Lernraten. Für jede Lernrate (2e-5, 1e-5, 5e-6, 3e-6, 1e-6) wurde ein vollständiger Trainingslauf durchgeführt. Dabei wurde die F1-Makro-Metrik pro Epoche geloggt und in einem DataFrame gespeichert.

Ein Bild, das Text, Screenshot, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

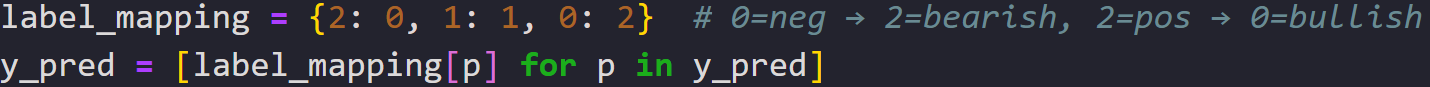
Die besten Modelle wurden auf Basis des höchsten eval\_f1-Wertes ausgewählt und auf dem Testset final evaluiert (siehe Kapitel 4).

3.4 Label-Mapping für CryptoBERT

Ein technisches Detail betraf das **Label-Format von CryptoBERT**, das in seiner Originalversion eine andere Reihenfolge der Sentimentklassen verwendet:

* 0 = negative
* 1 = neutral
* 2 = positive

Um die Auswertungen konsistent zu gestalten und mit allen anderen Modellen vergleichen zu können, wurde ein explizites **Label-Mapping** eingeführt. Dieses ordnet positive → bullish, neutral → neutral, negative → bearish:



Dieses Mapping wurde ausschließlich für CryptoBERT nach der Inferenz angewendet und stellt sicher, dass alle Metriken und Visualisierungen auf einer gemeinsamen Skala basieren.

4 Ergebnisse und Diskussion

In diesem Kapitel werden die Ergebnisse des Modellvergleichs sowie der Feinjustierung systematisch präsentiert, interpretiert und kritisch eingeordnet. Der Fokus liegt dabei auf der Bewertung der Sentiment-Klassifikationsleistung nach Texttyp (Posts vs. Kommentare), Sentimentklasse (*bullish*, *neutral*, *bearish*) und Modellart (transformerbasiert vs. lexikonbasiert).

4.1 Baseline-Vergleich aller Modelle

Zur Evaluierung der sechs getesteten Modelle wurden zunächst F1-Scores für jede Sentimentklasse berechnet. Die Testdaten umfassten jeweils 300 manuell gelabelte Reddit-Posts und 300 Kommentare. Die Ergebnisse zeigen deutliche Unterschiede – sowohl zwischen den Modellarten als auch innerhalb der Klassen.

Die besten Ergebnisse bei längeren Reddit-Posts lieferten FinBERT, TextBlob und VADER. Insbesondere FinBERT zeigte starke Werte in den Klassen *bullish* (F1: 0.5075) und *bearish* (F1: 0.5000). TextBlob erreichte fast gleichwertige Scores mit insgesamt stabiler Leistung über alle drei Klassen hinweg. Transformer wie DeBERTa und CryptoBERT schnitten auf den Post-Daten unterdurchschnittlich ab – vermutlich aufgrund mangelnder Spezialisierung auf längere Textformate. Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Bei den Kommentaren zeigte CryptoBERT eine deutlich bessere Baseline-Leistung: *bullish* (F1: 0.5143) und *neutral* (F1: 0.5806). Jedoch war die *bearish*-Klasse mit nur 0.1333 F1-Punkten sehr schwach vertreten. Auch VADER und TextBlob lieferten bei Kommentaren relativ ausgewogene Werte und schnitten hier teilweise besser ab als einige Transformer-Modelle.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Ergebnisse verdeutlichen, dass kein Modell in allen Klassen und Textarten gleichzeitig dominiert. Stattdessen zeigen sich Spezialisierungen:

* FinBERT eignet sich besonders für strukturierte Langtexte mit Finanzbezug (Posts).
* CryptoBERT zeigt Stärken bei kurzen, informellen Kommentaren – allerdings mit Schwächen in extremen Sentiments.

4.2 Optimierungsergebnisse: CryptoBERT und FinBERT

Nach dem Modellvergleich wurden die beiden leistungsfähigsten Modelle ausgewählt und gezielt optimiert:

* CryptoBERT für Reddit-Kommentare
* FinBERT für Reddit-Posts

Ziel war es, die Performance insbesondere in den schwächeren Klassen (*bearish*) zu verbessern und die Gesamtstabilität zu erhöhen. Die Evaluation erfolgte jeweils auf separaten Testsets, die zuvor nicht im Training verwendet wurden.

4.2.1CryptoBERT auf Kommentaren (nach Fine-Tuning)

Die Feinjustierung von CryptoBERT erfolgte mit gewichteter Verlustfunktion und systematischem Lernratenvergleich. Trotz dieser Maßnahmen blieb die Modellleistung hinter den Erwartungen zurück.

Evaluation auf dem Testset:

| Metrik | Wert |
| --- | --- |
| F1-Score (macro) | 0.3651 |
| Cohen’s Kappa | 0.0911 |
| Accuracy | 0.45 |

Klassifikationsreport:

| Klasse | Precision | Recall | F1-Score | Support |
| --- | --- | --- | --- | --- |
| Bullish | 0.33 | 0.17 | 0.22 | 18 |
| Neutral | 0.58 | 0.68 | 0.62 | 28 |
| Bearish | 0.21 | 0.30 | 0.25 | 10 |

Die *neutral*-Klasse konnte mit einem F1-Score von 0.62 stabil erkannt werden. Die Performance in den *bullish*- und *bearish*-Klassen blieb allerdings unzureichend. Auch der niedrige Kappa-Wert (0.0911) zeigt, dass das Modell über alle Klassen hinweg keine robuste Unterscheidung leisten konnte.

Interpretation:  
Die schwache Performance lässt sich auf mehrere Faktoren zurückführen:

* Kommentare sind oft sehr kurz und kontextlos, was das Sentiment schwer erkennbar macht.
* Ironie, Sarkasmus oder Memes sind ohne Thread-Zusammenhang kaum klassifizierbar.
* Die Trainingsdatenmenge war mit 300 Samples begrenzt – eine Skalierung wäre notwendig.

4.2.2FinBERT auf Posts (nach Fine-Tuning)

FinBERT wurde mit denselben Trainingsverfahren optimiert. Im Gegensatz zu CryptoBERT zeigte sich hier eine **signifikante Verbesserung der Modellleistung**.

**Evaluation auf dem Testset:**

| Metrik | Wert |
| --- | --- |
| F1-Score (macro) | 0.6997 |
| Cohen’s Kappa | 0.507 |
| Accuracy | 0.72 |

Klassifikationsreport:

| Klasse | Precision | Recall | F1-Score | Support |
| --- | --- | --- | --- | --- |
| Bullish | 0.69 | 0.58 | 0.63 | 19 |
| Neutral | 0.74 | 0.79 | 0.76 | 33 |
| Bearish | 0.67 | 0.75 | 0.71 | 8 |

Alle drei Klassen konnten zuverlässig klassifiziert werden. Besonders die *bearish*-Klasse, die bei anderen Modellen häufig vernachlässigt wurde, erzielte hier stabile Werte. Der Kappa-Wert von über 0.5 deutet auf eine robuste Unterscheidung zwischen allen Klassen hin.

Die Lernrate 2e-5 zeigte sich als optimal, was in der vorherigen Hyperparameteranalyse identifiziert wurde.

Der Vergleich beider Fine-Tuning-Ergebnisse zeigt:

| Modell | Datentyp | F1 (macro) | Kappa | Kommentar |
| --- | --- | --- | --- | --- |
| CryptoBERT | Kommentare | 0.3651 | 0.0911 | Schwierig, da Texte kurz, unstrukturiert, mehrdeutig |
| FinBERT | Posts | 0.6997 | 0.507 | Sehr gute Leistung, hohe Stabilität nach Tuning |

Damit konnte FinBERT durch gezieltes Fine-Tuning deutlich verbessert werden, während bei CryptoBERT weitere Kontextinformationen oder größere Datenmengen notwendig wären.

4.3 Herausforderungen und Grenzen

Die Durchführung des Modellvergleichs und der anschließenden Optimierung brachte verschiedene Herausforderungen mit sich – sowohl technischer als auch inhaltlicher Natur. Diese betreffen insbesondere die Beschaffenheit der Reddit-Daten, die Feinabstimmung der Modelle sowie die Grenzen lexikonbasierter Verfahren.

1. Begrenzter Kontext bei Kommentaren

Ein zentrales Problem war die Analyse von Reddit-Kommentaren. Diese sind oft sehr kurz, bestehen nur aus einem Satz oder gar nur einem Emoji oder Meme-Verweis. Dadurch fehlt dem Modell jeglicher inhaltlicher Kontext, um das zugrundeliegende Sentiment korrekt einzuordnen. Ironie, Sarkasmus und implizite Bewertungen können ohne Thread-Zusammenhang kaum erkannt werden. Transformer-Modelle wie CryptoBERT geraten dadurch schnell an ihre Grenzen.

Folge:  
Trotz spezialisierter Architektur und Fine-Tuning konnten bei Kommentaren keine stabilen Klassifikationsergebnisse erzielt werden. Eine mögliche Verbesserung wäre die Integration thread-basierter Features oder kontextsensitiver Zusatzinformationen.

2. Kleine Datenbasis

Die Grundlage der Analyse war ein manuell gelabelter Datensatz von jeweils 300 Posts und Kommentaren. Obwohl dies für eine erste Evaluation ausreichend ist, zeigte sich insbesondere beim Fine-Tuning, dass tiefe Modelle stark von größeren Datenmengen profitieren. Schon geringe Änderungen in den Validierungsdaten führten zu stark schwankenden Metriken – ein klassisches Anzeichen für Overfitting bzw. mangelnde Generalisierungsfähigkeit.

Lösungsansatz:  
Eine Skalierung des Datensatzes durch weiteres manuelles oder semi-automatisches Labeling ist für zukünftige Experimente essenziell. Alternativ könnten Data-Augmentation-Techniken (z. B. Back-Translation, Synonym-Ersetzungen) angewendet werden.

3. Modellabhängige Labelstruktur

Ein technisches Detail betraf die uneinheitliche Labelstruktur einzelner Modelle. CryptoBERT etwa verwendet eine andere Reihenfolge (0=negative, 2=positive) als die restlichen Modelle (0=bullish, 2=bearish). Dieses Problem wurde durch manuelles Label-Mapping behoben – dennoch macht es deutlich, dass bei der Integration fremder Modelle stets ein Abgleich der internen Klassenlogik erforderlich ist.

4. Leistungsgrenze lexikonbasierter Modelle

Die lexikonbasierten Modelle VADER und TextBlob zeigten insbesondere bei kurzen Kommentaren eine erstaunlich stabile Performance. Ihre Methodik ist jedoch nicht domänenspezifisch und basiert rein auf Schlagwortbewertung. Dadurch sind sie anfällig für irreführende Begriffe, fehlenden Kontext und unübliche Formulierungen. Langfristig stoßen solche Modelle bei feineren Sentimentunterscheidungen an ihre Grenzen.

5. Hyperparameter und Trainingsinstabilität

Beim Vergleich verschiedener Lernraten zeigten sich klare Unterschiede in der Modellleistung. Besonders bei CryptoBERT führte bereits eine kleine Änderung der Lernrate zu stark abweichenden Validierungsergebnissen. Dies verdeutlicht, wie sensibel Transformer-Modelle auf Hyperparameter reagieren – insbesondere bei kleinen Datensätzen. Auch das Early Stopping musste feinjustiert werden, da Modelle teilweise zu schnell oder zu spät abbrachen.

5 Fazit

Ziel dieser Arbeit war es, verschiedene vortrainierte Sentimentanalysemodelle auf Reddit-Daten systematisch zu vergleichen und durch gezieltes Fine-Tuning zu verbessern. Dabei wurden sowohl transformerbasierte als auch klassische lexikonbasierte Ansätze berücksichtigt. Besonderes Augenmerk lag auf der differenzierten Betrachtung zweier Texttypen – längere Reddit-Posts und kurze Kommentare – sowie der drei Sentimentklassen *bullish*, *neutral* und *bearish*.

Im Modellvergleich zeigte sich, dass kein Modell universell überlegen ist. Stattdessen variiert die Performance stark in Abhängigkeit vom Texttyp und der Sentimentklasse:

* FinBERT erwies sich als robustestes Modell für längere, analytische Reddit-Posts. Nach dem Fine-Tuning erreichte es ein F1-Score (macro) von 0.70 und klassifizierte alle drei Sentiments zuverlässig.
* CryptoBERT, obwohl thematisch auf Kryptowährungen trainiert, zeigte insbesondere bei kurzen Kommentaren Schwierigkeiten – vermutlich bedingt durch fehlenden Kontext und geringe Datenmenge.
* Die lexikonbasierten Modelle VADER und TextBlob lieferten überraschend stabile Baselines, insbesondere bei *neutralen* und *bearishen* Kommentaren, konnten aber inhaltlich nicht mit spezialisierten Modellen konkurrieren.

Die eingesetzten Optimierungsstrategien – insbesondere Custom Loss Functions, Lernratenvergleich und stratifizierte Datensplits – haben sich als wirksam erwiesen. Dennoch wurden im Rahmen der Analyse auch klare Grenzen deutlich: kleine Datenmengen, fehlende Kontextinformationen bei Kommentaren und die inhärente Unschärfe informeller Sprache sind Faktoren, die die Modellgüte einschränken.

5.1 Ausblick

Aufbauend auf den Erkenntnissen dieser Arbeit ergeben sich mehrere Ansatzpunkte für weiterführende Projekte:

1. Erweiterung des Datensatzes  
   Größere Mengen an gelabelten Daten – idealerweise >1000 pro Texttyp – würden die Trainings- und Validierungsstabilität erhöhen und feinere Modellunterschiede ermöglichen. Auch das Einbeziehen automatisierter Labelverfahren (z. B. distant supervision) könnte helfen.
2. Kontextintegration für Kommentare  
   Die Klassifikation von Reddit-Kommentaren könnte durch die Einbeziehung von Thread-Titeln, Antworten oder Upvote-Metriken deutlich verbessert werden. Die Entwicklung kontextsensitiver Modelle wäre ein lohnender nächster Schritt.
3. Domänenspezifisches Finetuning mit größeren Modellen  
   Die Analyse könnte mit größeren Foundation Models (z. B. BERTweet, LLaMA oder DeBERTa-v3-large) wiederholt werden. Diese bieten teils bessere Generalisierung – erfordern jedoch erheblich mehr Rechenleistung und Daten.
4. Einsatz in Echtzeit-Analysen  
   Die entwickelten Modelle könnten in einem Streamlit-Dashboard oder einer Webanwendung eingebunden werden, um Marktstimmungen zu Kryptowährungen kontinuierlich zu erfassen und darzustellen – etwa in Verbindung mit Preisverläufen oder Social-Media-Trends.
5. Erweiterung auf andere Plattformen  
   Eine Übertragung der Methodik auf Twitter, YouTube-Kommentare oder Telegram-Gruppen wäre möglich und könnte durch multimodale Daten (z. B. Emojis, Hashtags, Reaktionsverhalten) ergänzt werden.