

# Vertrag zur Abschlussarbeit

Vertrag zur Bachelorarbeit von **Kristoffer Schaaf** bei **Prof. Dr. Stefan Sarstedt**

**Vorläufiger Titel:**

„Entwicklung einer Software zur Erkennung von Fake News auf Nachrichtenportalen.“

## 1. Problemstellung

### Beschreibung und Einordnung des Problems

Fake News sind ein allgegenwärtiges Problem. Als politischer Schachzug bereits im Amerikanischen Bürgerkrieg 1779 genutzt, hat der Term auch in der US-Wahl 2016 wieder eine besondere Rolle gespielt [Sharma2024].

Sie können die Form von Text, Fotos, Filmen oder Audio annehmen und sind dementsprechend auf jeder Plattform auffindbar, die die Verbreitung nicht unterbindet. Die momentan populärste Plattform zum Teilen der Fake News ist WhatsApp [Ashish2024].

Fake News definieren sich wie folgt:

*Fake News sind bewusst erstellte Online-Falschmeldungen, die teilweise oder vollständig unwahre Inhalte verbreiten, um Leser\*innen gezielt zu täuschen oder zu manipulieren. Sie imitieren klassische Nachrichtenformate, nutzen auffällige Titel, emotionale Bilder und strategisch gestaltete Inhalte, um Glaubwürdigkeit zu erzeugen und Aufmerksamkeit zu gewinnen. Ziel ist es, durch das Verbreiten dieser Inhalte Klicks, Reichweite und damit finanzielle oder ideologische Vorteile zu erzielen. [Baptista2020].*

Zur Identifizierung von Fake News gibt es fünf verschiedene Ansätze:

Kategorie	Kurzbeschreibung	Beispiele / Methoden
<b>Language Approach</b>	Analyse sprachlicher Merkmale (Wortwahl, Grammatik, Syntax).	<ul style="list-style-type: none"><li>- Bag of Words (n-Gramme)</li><li>- Semantische Analyse</li><li>- Deep Syntax (PCFG, Parse Trees)</li></ul>
<b>Topic-Agnostic</b>	Bewertet formale Eigenschaften statt Inhalte.	<ul style="list-style-type: none"><li>- Viele Werbeanzeigen</li><li>- Lange, reißerische Headlines</li><li>- Autorennamen vorhanden</li></ul>
<b>Machine Learning</b>	Einsatz trainierter Algorithmen auf Fake-News-Datensätzen.	<ul style="list-style-type: none"><li>- Twitter Crawler</li><li>- Rumor Identification Framework</li><li>- Datensätze via Crowdsourcing</li></ul>
<b>Knowledge-Based</b>	Abgleich mit externen Faktenquellen zur Validierung.	<ul style="list-style-type: none"><li>- Expert Fact Checking</li><li>- ClaimBuster Tool</li><li>- Crowdsourcing über Kiskkit</li></ul>
<b>Hybrid Approach</b>	Kombination mehrerer Ansätze (Text, Quelle, Reaktionen).	<ul style="list-style-type: none"><li>- CSI-Modell (Capture, Score, Integrate)</li><li>- Kombi aus menschlicher &amp; KI-Analyse</li></ul>

[Matthee2021]

Im Folgenden wird ausschließlich der sogenannte „Machine Learning“-Ansatz behandelt.

Relevante Literatur zu diesem Bereich ist unter anderem hier zu finden:

- [Sharma2024]
- [Ashish2024]
- [Gupta2022]

Verschiedene Arbeiten zu diesem Thema im deutschsprachigen Raum sind:

- TH Ingolstadt [Willweber2023]
- University of Applied Sciences Upper Austria [Simone2022]
- RWTH Aachen [Mattern2021]
- TU Darmstadt [Hartwig2021]

Aus Betrachtung dieser geht hervor, dass die allgemeine Aufarbeitung dieses Themas bereits umfassend bearbeitet wurde. Es existieren verschiedenste „Machine Learning“-Modelle zum Erkennen von Fake News mit teils sehr guten Metriken. Auch stehen verschiedene Datensätze mit deutschen Fake und nicht Fake Artikeln zur Verfügung.

Im Paper der University of Applied Sciences Upper Austria wird die Qualität verschiedener deutscher Nachrichtenportale mit „Machine Learning“-Modellen getestet. Das Ergebnis zeigt, dass Spiegel, Die Zeit und Süddeutsche die besten Portale sind und Express, BZ-Berlin und Bild die schlechtesten - also auch am meisten Fake News verbreiten.

Das Paper der TU Darmstadt stellt das White-Box Browser Plugin TrustyTweet vor. Dieses Plugin unterstützt Twitter-Nutzer\*innen beim Erkennen von Falschinformationen. Es gibt politisch neutrale, transparente und verständliche Hinweise, ohne das eigene Urteilsvermögen einzuschränken. Ziel ist es, die Medienkompetenz der Nutzer\*innen langfristig zu stärken.

## Hauptziel der Arbeit

Motiviert durch die Arbeiten der University of Applied Sciences Upper Austria und der TU Darmstadt werde ich in dieser Arbeit die Entwicklung eines weiteren Tools dokumentieren. Dieses Tool soll wie auch das Browser Plugin TrustyTweet eine Unterstützung zum Erkennen von Fake News anbieten.

Ob dieses Tool als Browser Plugin oder als eine andere Form der Software implementiert wird, steht zum jetzigen Zeitpunkt noch nicht fest.

Auch ob eine Black Box oder White Box - das heißt, kann der User zum Beispiel sehen, warum der Artikel als Fake News deklariert wird - Architektur genutzt wird, wird im Laufe der Arbeit entschieden.

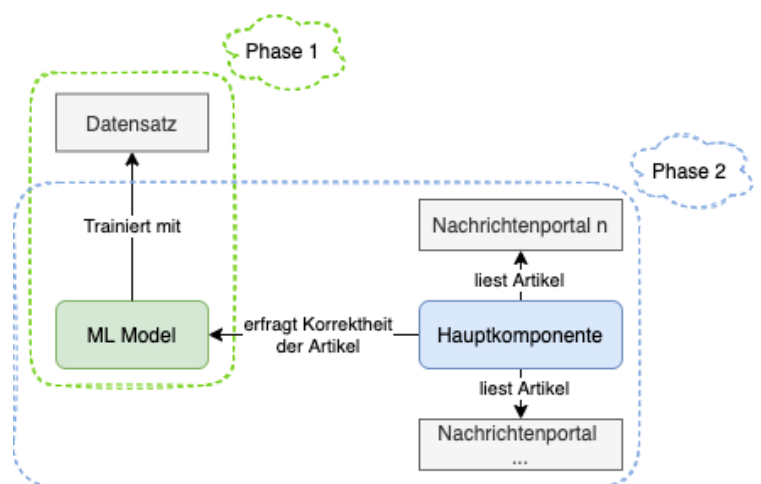
Ziel ist es, dass das Tool nicht wie TrustyTweet auf Twitter eingesetzt wird, sondern auf verschiedenen Nachrichtenportalen.

Welche Portale hierfür genutzt werden, hängt von der noch ausstehenden Entscheidung der genutzten Datensätze ab.

Es ergeben sich zwei wichtige Softwarekomponenten:

(1) Ein „Machine Learning“-Model, welches durch in „Fake“ und „nicht Fake“ klassifizierte, deutsche Artikel trainiert wird.

(2) Eine Hauptkomponente, welche die Schnittstelle zum Endnutzer anbietet und außerdem mit dem „Machine Learning“-Model kommuniziert um die zu analysierenden Artikel auf den verschiedenen Nachrichtenportalen auszuwerten.



## Datensatzwahl

Folgende Datensätze enthalten deutsche Fake News Artikel:

Datensatz	Quelle	Anzahl Zeilen	Anteil Fake (%)	Besonderheiten / Einschränkungen	Eignung
<b>Fake News Dataset German</b>	University of Applied Sciences Upper Austria	63.868	7,24 %	Geringer Anteil an Fake-News → unausgewogene Klassenverteilung	Weniger geeignet
<b>GermanFake NC</b>	Fraunhofer-Institut für Sichere Informationstechnologie SIT	489	100 %	Enthält nur Referenzen, keine Texte → keine direkte Textauswertung möglich	Nicht geeignet
<b>FANG-COVID</b>	Association for Computational Linguistics	41.242	31,97 %	Ausgewogene Klassen, vollständige Texte, viele Metadaten (Titel, Header, Label)	<b>Sehr geeignet</b>
<b>DeFaktS</b>	FZI Forschungszentrum Informatik	–	–	Kein Zugang → Nutzung ausgeschlossen	Nicht verfügbar

Für das Trainieren meines Modells kommt am ehesten der FANG-COVID-Datensatz in Frage. Dieser bietet mit über 41.000 Einträgen eine gute Datengrundlage. Außerdem hat dieser mit einem Fake-News-Anteil von ca. 32 % eine im Vergleich zu anderen verfügbaren Datensätzen deutlich ausgewogenere Klassenverteilung. Ob diese Klassenverteilung ausreicht, kann zum jetzigen Zeitpunkt noch nicht bestätigt werden. Deshalb ist diese Festlegung auf diesen Datensatz erst einmal temporär.

Darüber hinaus enthält der Datensatz neben der Label-Kennzeichnung („fake“ oder „real“) auch Titel, Header und den gesamten Beitrag der Artikel, was eine tiefergehende textanalytische Auswertung ermöglicht.

Im „[German News Dataset](#)“. Sind außerdem zahlreichen Artikel verschiedener Zeitungen enthalten. Diese erleichtern das Testen der Hauptkomponente während der Entwicklung.

## 2. Einzelaufgaben und angestrebte Ergebnisse

	Einzelaufgabe	Angestrebtes Ergebnis
1.1	Wählen eines geeigneten Algorithmus für die Implementierung und eines geeigneten Datensatzes für das Trainieren des „Machine Learning“-Modells	Das Model hat eine Genauigkeit von > 0.85
1.2	Das „Machine Learning“-Model bietet eine Schnittstelle für die Hauptkomponente an	Die Schnittstelle kann über API-Tests getestet werden
1.3	Das "Machine Learning"-Model erzielt mit dem German News Datensatz sinnvolle Ergebnisse	Die Genauigkeit liegt bei > 0.80
1.4	Das „Machine Learning“-Model wird als Container zur Verfügung gestellt	Der Container kann über Docker gestartet und über einen Port auf <i>localhost</i> angesprochen werden
2.1	Die Hauptkomponente lässt sich in den Chrome Browser integrieren und erkennt einen Artikel, wenn dieser geöffnet wird.	Ein simples „Hello World!“ Pop-up zur Demonstration, welches nach Erfolg aus der Implementierung entfernt wird

	<b>Einzelaufgabe</b>	<b>Angestrebtes Ergebnis</b>
<b>2.2</b>	Die Hauptkomponente gibt die Artikel an das „Machine Learning“-Model weiter und empfängt die entsprechenden Rückgabewerte	Die Funktionalität der Hauptkomponente wird über Integrations- oder E2E-Tests getestet.
<b>2.3</b>	Die Hauptkomponente ergänzt im Browser entsprechendes html zur Validierung, bzw. nicht Validierung des Artikels	Erfolgreiche Live Demonstration

### 3. Vereinbarungen über den Ablauf

#### Richtlinien für Software Engineering:

- Die Software wird gepflegt in dem GitHub Repository:  
[https://github.com/krisschaaf/ml\\_zum\\_erkennen\\_von\\_fake\\_news](https://github.com/krisschaaf/ml_zum_erkennen_von_fake_news)  
Für die Zugriffsrechte auf das Repository bitte eine Mail an:  
[kristoffer.schaaf@haw-hamburg.de](mailto:kristoffer.schaaf@haw-hamburg.de)
- Sämtlicher kopierter Code wird gekennzeichnet (sofern möglich)

#### Verpflichtungen zur Aktualisierung verwendeter Software:

- Die genutzten Base Images werden zu Beginn der Arbeit festgelegt. Sollten während der Bearbeitung keine kritischen Vulnerabilities zu diesen Images bekannt werden, wird dieses Image nicht aktualisiert.
- Der genutzte Browser wird zu Beginn der Arbeit festgelegt. Sollten während der Bearbeitung Minor Updates released werden, wird der Browser entsprechend aktualisiert. Sollten während der Bearbeitung Major Upates released werden, werden diese nur dann installiert, wenn der bestehende Code für die Hauptkomponente nicht aufwändig angepasst werden muss.

#### Abgabe der Arbeit in Dateiform:

- Die Arbeit wird als .zip Archiv abgegeben. In diesem Archiv befinden sich zwei Ordner. Ein Ordner für das „Machine Learning“-Model und ein Ordner für die Hauptkomponente.
- Zusätzlich wird eine Anleitung zum Aufsetzen angefügt.
- **Eventuell** wird die Hauptkomponente released (zum Beispiel in Form eines Browser Plugins) und das „Machine Learning“-Model deployed.

### 4. Organisatorisches

#### Zeitraum der Arbeit:

Voraussichtliche Fertigstellung innerhalb des Sommersemesters 2025 (Vor Beginn der Semesterferien)

**Bearbeiter:**

Name	Schaaf
Vorname	Kristoffer
Email HAW	<a href="mailto:kristoffer.schaaf@haw-hamburg.de">kristoffer.schaaf@haw-hamburg.de</a>
Email Privat	<a href="mailto:kris-schaaf@gmx.de">kris-schaaf@gmx.de</a>
Telefon	+49 175 1207539
Adresse	Marschblick 6, 25821 Breklum

**Erstbetreuung:**

Prof. Dr. Stefan Sarstedt ([stefan.sarstedt@haw-hamburg.de](mailto:stefan.sarstedt@haw-hamburg.de))

**Zweitbegutachtung:**

## 5. Notwendige Voraussetzungen

**Hardware:**

- Ein Rechner mit Zugriff auf das Internet

**Inhaltlich:**

- Keine Abhängigkeiten

**Software:**

- Ein erweiterbarer Browser (zum Beispiel Google Chrome)
- Mind. 2 passende Datensätze zum Trainieren des „Machine Learning“-Modells

## 6. Vorläufige Gliederung der Arbeit

**1. Einleitung**

- Hintergrund: Die zunehmende Verbreitung von Fake News und deren gesellschaftliche Auswirkungen
- Wahl der Nachrichtenportale
- Zielsetzung: Entwicklung einer Software zur automatisierten Fake-News-Erkennung
- Aufbau der Arbeit

**2. Grundlagen und Begriffsdefinitionen**

- Definition „Fake News“: Merkmale, Ziele, Beispiele
- Kategorisierung der Fake News Detection-Ansätze
- Warum der Fokus auf Machine Learning?
- Überblick über relevante Plattformen und deren Rolle im Medienkonsum

**3. Maschinelles Lernen zur Fake News-Erkennung**

- Grundlagen von ML
  - Überwachtes vs. unüberwachtes Lernen
  - Klassische ML-Modelle (Logistic Regression, Naive Bayes, SVM, Entscheidungsbäume)
- Performance-Metriken: Accuracy, Precision, Recall, F1-Score
- Herausforderungen im Kontext von Fake News

#### 4. Natural Language Processing & Deep Learning (noch unsicher über Notwendigkeit)

- Text-Vorverarbeitung
- Word Embeddings
- Deep Learning-Modelle
- Diskussion: White Box vs. Black Box-Ansätze zur Erklärbarkeit

#### 5. Relevante Datensätze und Auswahlkriterien

- Vorstellung verfügbarer deutscher Fake-News-Datensätze
  - FANG-COVID, German News Dataset (Kaggle), GermanFakeNC etc.
- Auswahl und Begründung des finalen Datensatzes
- Herausforderungen bei deutschen Textdaten (Klassenverteilung, Quellenvielfalt)

#### 6. Konzeption der Softwarelösung

- Zielsystem: Tool zur Fake News-Erkennung auf Nachrichtenportalen
- Entscheidung für Architektur
- Systemkomponenten:
  - ML-Modell zur Klassifikation
  - Hauptkomponente/UI zur Analyse von Artikeln
  - Schnittstellenkommunikation

#### 7. Umsetzung des Prototyps

- Implementierung des ML-Modells: Architektur, Frameworks, Training
- Preprocessing und Feature Engineering
- Entwicklung der Benutzeroberfläche (wenn nötig/sinnvoll)
- Integration der Datenbasis

#### 8. Evaluation und Ergebnisse

- Testumgebung und Testdaten
- Ergebnisse: z.B. Genauigkeit, Fehlklassifikationen, Laufzeit
- Vergleich mit existierenden Ansätzen
- Feedback zur Nutzbarkeit (ggf. Usability-Test)

#### 9. Diskussion

- Bewertung der Ergebnisse
- Stärken und Schwächen der gewählten Architektur
- Ethische Aspekte: z.B. Transparenz, Verantwortung, Missbrauch
- Grenzen maschineller Fake News Detection

#### 10. Fazit und Ausblick

- Zusammenfassung der zentralen Erkenntnisse
- Bewertung des entwickelten Tools
- Potenzial für Weiterentwicklungen (z. B. Multimodale Analyse, Einsatz von Transformers, Echtzeiterkennung)
- Ausblick auf zukünftige Forschungsrichtungen

### 11. Meilensteinplan

KW / Zeitraum	Meilenstein / Aufgabe	Ziel / Ergebnis
KW 15 (07.04. – 14.04.)	Abgabe & Besprechung des Exposés	Feedback vom Betreuer einholen, ggf. Anpassung des Projektumfangs
KW 16–17	Literaturrecherche & Theorieteil vorbereiten	Sammlung & Sichtung relevanter Literatur zu ML, Fake News, Nachrichtenportale, Datensätzen etc.
KW 18–19	Technologie-Stack festlegen & Datensatz final wählen	Entscheidung über Frameworks und Datensatz

<b>KW 20</b>	Text Preprocessing + Explorative Datenanalyse	Analyse von Klassenverteilung und Artikelstruktur
<b>KW 21-22</b>	Entwicklung & Training des ML-Modells	Erstes Modell trainieren, Evaluation mit Basismetriken
<b>KW 22</b>	Zwischenpräsentation / Abstimmung mit Betreuer	Vorläufiges Modell vorstellen, Feedback einholen, ggf. Justierung der Architektur
<b>KW 23-25</b>	Implementierung der Hauptkomponente (UI / Tool-Frontend)	Erste UI, Anbindung an Modell
<b>KW 25</b>	Integration: Modell & Oberfläche verbinden	Prototyp funktionsfähig; Artikelanalyse auf einem Nachrichtenportal möglich
<b>KW 26</b>	Systemtests, Evaluation & Feintuning	Tests mit realen Artikeln (z. B. aus German News Dataset), Performanceverbesserung
<b>KW 27</b>	Dokumentation der Software & Vorbereitung der Ergebnisdiskussion	Funktion, Architektur, Entscheidungswege dokumentiert
<b>KW 28</b>	Verfassen des Fazits & Ausblicks	Reflexion über Vorgehen, Herausforderungen, mögliche Weiterentwicklungen
<b>KW 29</b>	Kompletter Feinschliff der Arbeit (Korrekturen, Layout etc.)	Abgabe-Ready-Version
<b>KW 29</b>	<b>Abgabe der Bachelorarbeit</b> (je nach Abgabetermin verschiebbar)	Finale Version abgeben

## 12. Literatur

[Sharma2024]	Sharma, U., & Singh, J. (2024–6). A comprehensive overview of fake news detection on social networks. <i>Social Network Analysis and Mining</i> , 14(1), 120. <a href="https://doi.org/10.1007/s13278-024-01280-3">https://doi.org/10.1007/s13278-024-01280-3</a>
[Ashish2024]	Ashish, Sonia, Arora, M., Hemraj, Rana, A., & Gupta, G. (2024). An Analysis and Identification of Fake News using Machine Learning Techniques. 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), 634–638. <a href="https://doi.org/10.23919/INDIACom61295.2024.10498879">https://doi.org/10.23919/INDIACom61295.2024.10498879</a>
[Basptista2020]	Baptista, J. P., & Gradim, A. (2020). Understanding Fake News Consumption: A Review. <i>Social Sciences</i> , 9(10). <a href="https://doi.org/10.3390/socsci9100185">https://doi.org/10.3390/socsci9100185</a>
[Simone2022]	Simone, S., Oliver, K., Clara, D., & Andreas, S. (2022). Detecting Fake News and Performing Quality Ranking of German Newspapers Using Machine Learning. 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 1–5. <a href="https://doi.org/10.1109/ICECCME55909.2022.9987851">https://doi.org/10.1109/ICECCME55909.2022.9987851</a>
[Matthee2021]	de Beer, D., & Matthee, M. (2021). Approaches to Identify Fake News: A Systematic Literature Review. In T. Antipova (Ed.), <i>Integrated Science in Digital Age 2020</i> (pp. 13–22). Springer International Publishing.
[Hartwig2021]	Hartwig, K., & Reuter, C. (2021). Fake News technisch begegnen – Detektions- und Behandlungsansätze zur Unterstützung von NutzerInnen. In P. Klimczak & T. Zoglauer (Eds.), <i>Wahrheit und Fake im postfaktisch-digitalen Zeitalter: Distinktionen in den Geistes- und IT-Wissenschaften</i> (pp. 133–149). Springer Fachmedien Wiesbaden. <a href="https://doi.org/10.1007/978-3-658-32957-0_7">https://doi.org/10.1007/978-3-658-32957-0_7</a>
[Willweber2023]	Willweber, J. T. D. R. (2023). Comparison of deep learning and word embedding approaches for German fake news detection [Phdthesis]. Technische Hochschule Ingolstadt.

[Mattern2021]	Mattern, J., Qiao, Y., Kerz, E., Wiechmann, D., & Strohmaier, M. (2021). FANG-COVID: A New Large-Scale Benchmark Dataset for Fake News Detection in German. In R. Aly, C. Christodoulopoulos, O. Cocarascu, Z. Guo, A. Mittal, M. Schlichtkrull, J. Thorne, & A. Vlachos (Eds.), Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER) (pp. 78–91). Association for Computational Linguistics. <a href="https://doi.org/10.18653/v1/2021.fever-1.9">https://doi.org/10.18653/v1/2021.fever-1.9</a>
[Gupta2022]	V. Gupta, R. S. Mathur, T. Bansal and A. Goyal, "Fake News Detection using Machine Learning," 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India, 2022, pp. 84-89, doi: 10.1109/COM-IT-CON54601.2022.9850560. keywords: {Measurement;Support vector machines;Machine learning algorithms;Social networking (online);Natural language processing;Web sites;Fake news;Natural Language Processing;Random Forest;Fake News Detection;Logistic Regression;k-Nearest Neighbor;Machine Learning;SVC},
	Mehta, D., Dwivedi, A., Patra, A., & Anand Kumar, M. (2021–4). A transformer-based architecture for fake news classification. Social Network Analysis and Mining, 11(1), 39. <a href="https://doi.org/10.1007/s13278-021-00738-y">https://doi.org/10.1007/s13278-021-00738-y</a>
	Mi, S. (2020). German News Dataset. <a href="https://www.kaggle.com/datasets/pqbsbk/german-news-dataset">https://www.kaggle.com/datasets/pqbsbk/german-news-dataset</a> . <a href="https://www.kaggle.com/datasets/pqbsbk/german-news-dataset">https://www.kaggle.com/datasets/pqbsbk/german-news-dataset</a>
	Olan, F. (2022). Fake news on Social Media the Impact on Society. Information Systems Frontiers : A Journal of Research and Innovation, 1. <a href="http://dx.doi.org/10.1007/s10796-022-10242-z">http://dx.doi.org/10.1007/s10796-022-10242-z</a>
	Raza, S. (2022). Fake news detection based on news content and social contexts a transformer-based approach. International Journal of Data Science and Analytics, 13(4), 335. <a href="http://dx.doi.org/10.1007/s41060-021-00302-z">http://dx.doi.org/10.1007/s41060-021-00302-z</a>
	Bondielli, A. (2024). Dataset for multimodal fake news detection and verification tasks. Data in Brief, 54, 110440. <a href="http://dx.doi.org/10.1016/j.dib.2024.110440">http://dx.doi.org/10.1016/j.dib.2024.110440</a>
	Sheikhi, S. (2021). An effective fake news detection method using WOA-xgbTree algorithm and content-based features. Applied Soft Computing, 109, 107559. <a href="https://doi.org/https://doi.org/10.1016/j.asoc.2021.107559">https://doi.org/https://doi.org/10.1016/j.asoc.2021.107559</a>
	Lan, D. H., & and, T. M. T. (2024). Exploring fake news awareness and trust in the age of social media among university student TikTok users. Cogent Social Sciences, 10(1), 2302216. <a href="https://doi.org/10.1080/23311886.2024.2302216">https://doi.org/10.1080/23311886.2024.2302216</a>

Unterschriften

---

Bearbeiter: Kristoffer Schaaf

Betreuer: Prof. Dr. Stefan Sarstedt

Zweitbegutachtung: