MisinfoTeleGraph: Network-driven Misinformation Detection for German Telegram Messages

Lu Kalkbrenner[1], Veronika Solopova[2], Steffen Zeiler[2], Robert Nickel[3], Dorothea Kolossa[2]

[1]CeMAS›, [2]Technische Universität Berlin, [3]Bucknell University

Correspondence: lu@kalkbrenner.in

Abstract

Connectivity and message propagation are central, yet often underutilised, sources of information in misinformation detection—especially on poorly moderated platforms such as Telegram, which has become a critical channel for misinformation dissemination, namely in the German electoral context. In this paper, we introduce Misinfo-TeleGraph, the first German-language Telegram-based graph dataset for misinformation detection. It includes over 5 million messages from public channels, enriched with metadata, channel relationships, and both weak and strong labels. These labels are derived via semantic similarity to fact-checks and news articles using M3-embeddings, as well as manual annotation. To establish reproducible baselines, we evaluate both text-only models and graph neural networks (GNNs) that incorporate message forwarding as a network structure. Our results show that GraphSAGE with LSTM aggregation significantly outperforms text-only baselines in terms of Matthews Correlation Coefficient (MCC) and F1-score. We further evaluate the impact of subscribers, view counts, and automatically versus human-created labels on performance, and highlight both the potential and challenges of weak supervision in this domain. This work provides a reproducible benchmark and open dataset for future research on misinformation detection in German-language Telegram networks and other low-moderation social platforms.

1 Introduction

Disinformation and misinformation, with their proven impact on democratic elections, have become one of the most harmful online phenomena of our age (Howard et al., 2019). Ever since mainstream social media platforms implemented more thorough content moderation policies against harmful speech and misinformation, many users migrated to Telegram (Rogers, 2020). For instance, it was shown that around 30% of adults use the Telegram messenger as a news source in Germany (Holnburger, 2023). Telegram has become a key platform for spreading misinformation, conspiracy theories and far-right ideologies in Germany, while largely remaining unmoderated (Urman and Katz, 2022; Holnburger, 2023), and solidifying false beliefs with the echo chamber effect (Bovet and Grindrod, 2020). Already in 2017, the Council of Europe reported that conventional fact-checking was becoming unable to respond to such data volumes to identify check-worthy content and verify it in a timely manner (Wardle and Derakhshan, 2017). Therefore, in recent years, extensive research has been conducted on identifying misinformation using machine learning methods. However, most studies focused on data from X (formerly Twitter) and on the English language, while for other languages, including German, mostly simple text-based methods were investigated. In this study, we present our **Misinfo-TeleGraph** Dataset[1], which is a German Telegram misinformation graph dataset including 13.845 German Telegram channels and their messages from October 2022 to May 2024, including the forwarding information and metadata regarding views and likes. 742 messages are weakly labeled by corresponding fact-checks and newspaper articles using similarity scores from M3-embeddings. We trained a Graph Neural Network (GNN) to detect misinformation and analyzed how the incorporation of network information improves the model's performance in comparison to a text-only approach. We make our code available in GitHub[2].

2 Related Work

While multiple successful methods were developed to detect the factual correctness of news purely rely-

[1] https://zenodo.org/records/13362123

[2] https://github.com/kalkbrennerei/MisinfoTeleGraph

Proceedings of the The 9th Workshop on Online Abuse and Harms (WOAH), pages 179–191
August 1, 2025 ©2025 Association for Computational Linguistics

ing on textual content (Tanvir et al., 2020; Hiriyannaiah et al., 2020; Kaliyar et al., 2021; Zhou et al., 2020), such models were shown to be language-dependent (Monti et al., 2019), prone to adversarial attacks (Han et al., 2020; Goodfellow et al., 2015), and generalize badly to new data due to over-reliance on linguistic patterns and keywords (Solopova et al., 2024). Recent works have been focusing on including social context and propagation patterns (Shu et al., 2017). Approaches based on social context often focus on user demographics, account authenticity and political bias of the thread participants (Uppada et al., 2022), location and profile pictures (Shu et al., 2019). Other approaches look at social network structure, and user reactions such as likes and shares (Monti et al., 2019; Li et al., 2020; Yang et al., 2020). Zhang et al. (2019) used message view counts and information about the Telegram channels in which messages have been shared, including the number of subscribers for each channel.

Liu and Wu (2018) used multivariate time series with recurrent and convolutional networks. Wu and Liu (2018) inferred user embeddings with social network structures and classified them using an LSTM-RNN. Mishra (2020) analyzed user-to-user interaction propagation paths over multiple hops using a transformer architecture, while Hamdi et al. (2020) used node2vec to create graph embeddings from the follower-followee relationship.

Motivated by the graph structure of social networks, *Graph Neural Networks (GNNs)* were identified as a promising technique within propagation-based approaches. Monti et al. (2019) applied a GNN for misinformation detection based on data from X, including content, social context, and propagation features. Han et al. (2020) extended this approach by leveraging continual learning techniques to improve the performance on unseen data. Dou et al. (2021) extracted node features from news articles and user preferences from X using BERT and node2vec embeddings, and compared Graph Convolutional Network (GCN) and GraphSAGE architectures, while also explicitly separating endogenous and exogenous user preferences. Comparing multiple types of GNNs for this task, Mahmud et al. (2022) showed that GraphSAGE (Hamilton et al., 2017) performed best, with a test accuracy of 96.99%. Nielsen and McConville (2022), which serves as the main inspiration for our work, implemented a heterogeneous version of the GraphSAGE model as a baseline for their MuMiN dataset of multi-lingual tweets, achieving an F1 score of 61.45% compared to the LaBSE (Language-Agnostic BERT Sentence Embedding) text-only baseline of 57.90%.

Most existing graph-based misinformation detection datasets, like the MuMiN and FakeNewsNet (Shu et al., 2020), are primarily derived from X, with limited options from other social networks. While there are non-specific datasets from platforms like Telegram, such as TGDataset (Morgia et al., 2023) and the Pushshift dataset (Baumgartner et al., 2020), research on graph neural networks for misinformation detection in Telegram data is notably absent. While Zhang et al. (2021) utilized Telegram threads to train a GNN for a node classification task, to the best of our knowledge, ours is the first work implementing GNNs with Telegram data for misinformation detection, and also the first on employing these for the German language.

3 Methods

To create the Telegram graph dataset, we used *weak annotation* on data that we received from Data4Transparency (2024). From this dataset, we constructed a graph using network information, including messages, channels, views, likes and cross-channel message forwarding. Statistics of the dataset are depicted in Figure 1. The annotated training graph is used to train a Graph Neural Network, where node embeddings are computed based on their neighborhood representation using the GraphSAGE architecture.

Our methods are inspired by Nielsen and McConville (2022), who trained a GNN on a graph dataset from X. Since Telegram and X are very different platforms, the creation of our graph dataset differs considerably from the work of Nielsen and McConville (2022). However, we were able to reuse some of their code, and we employed the same approach to train a baseline GNN model on the data.

# Telegram channels	13,845
# Telegram messages	5,727,631
Similarity threshold	0.7
# weakly linked message-claim pairs	742
# weak pairs in the factual class	110
# weak pairs in the misinfo. class	632
# weak pairs in the 'other' class	542
# strongly linked message-claim pairs	651
# strong pairs in the factual class	94
# strong pairs in the misinfo. class	557

Table 1: Statistics of the *MisinfoTeleGraph* dataset.

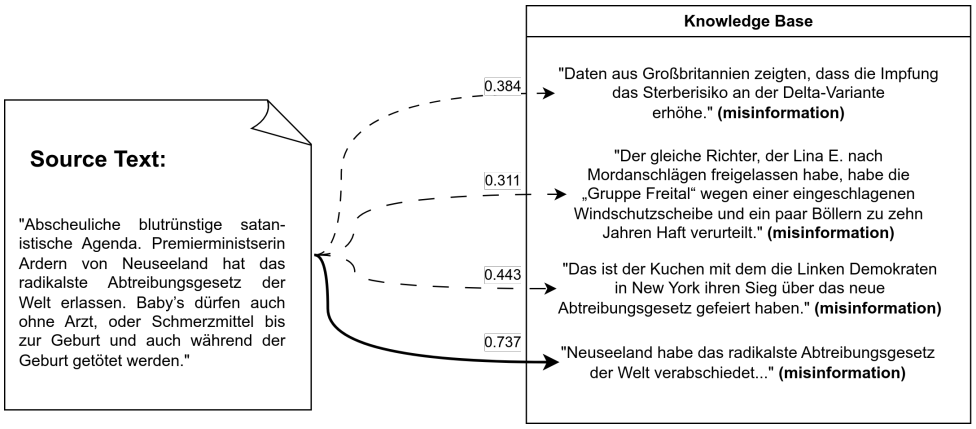


Figure 1: Weak Supervision using M3-embeddings and a knowledge base. A source text (on the left) is compared to claims contained in a knowledge base of fact-checks and news articles (on the right). Similarity scores are computed based on the M3-embeddings of the text and claims. The source text is linked with a claim, if the similarity score exceeds a threshold of 0.7, as is the case for the last claim in the knowledge base with a score of 0.737. The source text inherits the label (factual or misinformation) of the claim that it is matched with.

3.1 Telegram Data Source

For training our model, we created a dataset based on data provided from Data4Transparency (2024) (D4T). Their data contains information about which channel messages are posted in and which channel messages are being forwarded to. From this message-forwarding network information, we constructed a graph dataset as described in Section 3.3.

3.2 Training Data Annotation

Since training data annotation remains a costly task, *weak annotation* is a promising approach to annotate data sets of misinformation from online social networks. Manual data annotation often requires skilled human annotators, who are knowledgeable in their domain, such as professional fact-checkers in the case of the detection of misinformation. In this work, we use semantic similarity based on M3-embeddings (Chen et al., 2024) to pre-select Telegram messages that potentially contain misinformation and manually annotate the pre-selected collection. This approach is shown in Figure 1. For the weak annotation, we use a knowledge base of newspaper articles and fact-checking articles that contain texts from the sources in Table 2. The fact-checks were fetched from the Google Fact Check Tools API (API, 2024) and the newspaper articles were fetched from WoldNewsAPI [3].

The texts from the knowledge base are compared to the telegram messages using semantic similarity. We compared different semantic similarity thresholds by precision. We found that a threshold of 0.7 matches enough message-claim pairs with an acceptable precision of 67.86%. The resulting 868 weakly annotated message-claim pairs were annotated by hand to obtain 589 strongly annotated message-claim pairs. The precision was computed by dividing the number of strongly annotated message-claim pairs by the number of weakly annotated pairs.

Source	# articles
BR (Bayrischer Rundfunk)	343
CORRECTIV	2568
DPA (Deutsche Presseagentur)	2271
AFP (Agence France-Presse)	1012
presseportal.de	378
Zeit	2396
Taz	1293
Süddeutsche	655

Table 2: German knowledge base sources. Fact-checking articles on top and newspaper articles below.

3.3 Network Information from message forwarding

To feed both textual information and network information into a graph neural network, we created a graph $\mathcal{G}$ with two node classes for Telegram channels and Telegram messages. We use the following two edge classes to describe the information about messages being forwarded across channels:

- `IS_PART_OF` describes the relationship of a message being posted in a channel

[3] https://worldnewsapi.com/

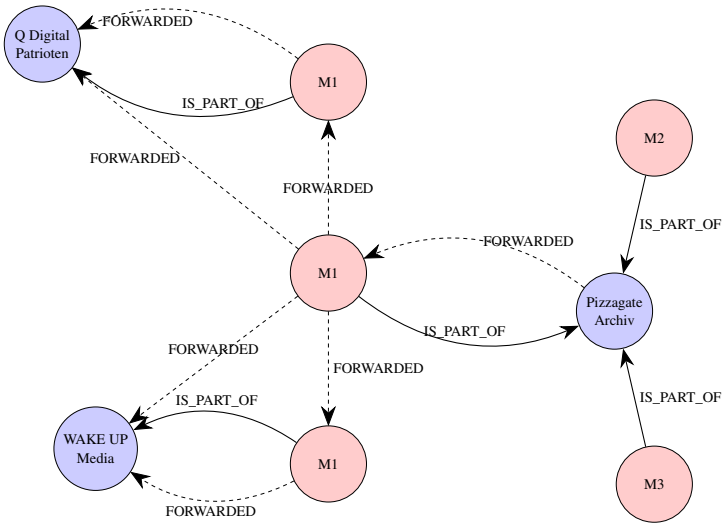


Figure 2: Example of a Social Network Graph. The example shows the "Pizzzagate Archiv" Telegram channel depicted in purple on the right. There are three messages (depicted in red) that were posted in the "Pizzagate Archiv" channel and are thus connected via an IS_PART_OF relation. One of the messages (depicted in the middle) has also been forwarded to two other channels, namely the "Q Digital Patrioten" channel and the "WAKE UP Media" channel. To preserve the information in which channel a message has been posted first, the messages are duplicated and linked by a FORWARDED relationship to the original messages when they are being forwarded. This is why the message in the middle appears three times in the middle. Every message only has one IS_PART_OF relationship with one channel.

- `FORWARDED` describes the relationship of a message being forwarded to a channel.

To preserve the information in which channel a message has been posted first, the messages are duplicated and linked by a `FORWARDED` relationship to the original messages when they are being forwarded.

Every node n has a feature vector X_n that contains the M3-embedding of the message text or the channel description concatenated with metadata.

A subgraph of the graph that we created can be seen in Figure 2.

3.4 Training of the GNN

To train the GNN model, we followed the procedure of (Nielsen and McConville, 2022), using a GraphSAGE architecture as proposed by (Hamilton et al., 2017). We experimented with different numbers of GraphSAGE layers and different aggregation functions. The GraphSAGE architecture setup is depicted in Figure 3. We set the learning rate to 1e-3 with a learning rate scheduler that starts at 1e-3 and ends at 1e-5 after 100 iterations, using a weight decay of 1e-5 for all experiments.

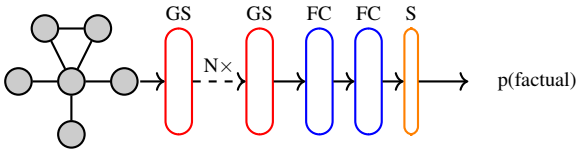


Figure 3: GraphSAGE model architecture. The network takes a node to be classified and its surrounding graph as an input. The graph is passed through N GraphSAGE layers (GS). The node embedding of the node to be classified is then passed through two fully connected layers (FC). A Sigmoid function (S) is applied to the resulting logits to compute the probabilities of belonging to the factual or the misinformation class.

4 Experimental Setup

For the GNN architecture depicted in Figure 3, we experimented with different numbers of GraphSAGE layers, different aggregator architectures and number of epochs. We then used the best-performing combination to verify our main hypotheses:

1. Including additional graph information (forwarding information) to train a GNN has an edge over the text-only misinformation classification baseline.

2. Including view and subscriber counts improves the GNN baseline.

3. Using weak labels for GNN training does not result in significantly poorer performance and calibration compared to strong labels.

For the text-only baseline, we use an architecture based on M3 that is depicted in Figure 4.

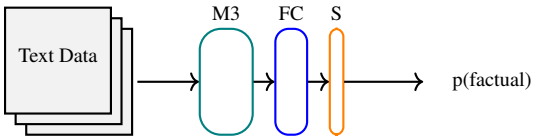


Figure 4: Text-based model architecture. Based on the text data of the messages to be classified, M3-embeddings are computed (M3). The embeddings are then classified by a fully connected layer (FC) and a Sigmoid function (S).

5 Results

5.1 Metrics

As standard evaluation metrics, we use Precision, Recall and their harmonic average F1-score, considering these indicators separately for misinformation and true samples. We also use the Matthews correlation coefficient (MCC), which is robust to unbalanced datasets, as a combination of precision and recall. Finally, we calculate the Expected Calibration Error (ECE) from (Nixon et al., 2019), which measures if a model's predicted output probabilities reflect the accuracy of its decision, to assess whether the model is exhibiting over-confidence or under-confidence. It is computed by

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} |\mathrm{acc}(b) - \mathrm{conf}(b)|, \qquad (1)$$

where B is the number of bins, n_b is the number of predictions in bin b, and N is the total number of data points. Each prediction is assigned to a bin based on its confidence score (i.e., the predicted probability of the top class), and $\mathrm{acc}(b)$ and $\mathrm{conf}(b)$ denote the average accuracy and average confidence within bin b, respectively.

5.2 Qualitative findings during annotation

While manually annotating message–claim pairs generated by the weak annotator model, we observed that it performs surprisingly well in cross-lingual contexts. Despite the dataset being composed of German-language Telegram channels, several English and Russian messages that were also contained in the channels were matched correctly with German claims. For example, an English message about the U.S. deploying Marines to Israel was successfully paired with a German-language claim falsely reporting that thousands of U.S. soldiers had landed in Israel (see Table 8). Further examples can be found in Appendix A.

However, the model often failed to capture logical specificity. For instance, it confused claims about vaccine-related deaths with those referring to COVID-19 fatalities, and did not consistently distinguish between adverse effects and death. Similarly, in messages related to the Gaza conflict, the model was unable to identify which actor—Israel or Hamas—was described as initiating violence.

These cases suggest that, while cross-lingual matching is a strength, the weak annotator model struggles with logical entailment and causal nuance, highlighting a key area for improvement in future work.

5.3 GNN architecture

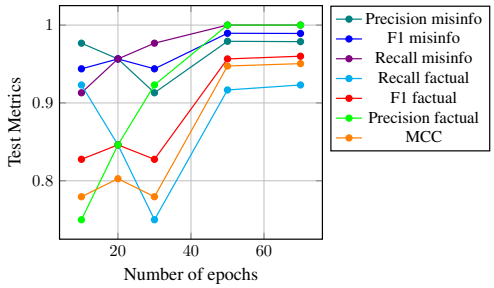
We compare different numbers of GraphSAGE layers, LSTM and mean aggregation, and different numbers of epochs.

Similar to Nielsen and McConville (2022), we are able to verify that LSTM aggregation performs best in terms of all considered metrics as depicted in Table 3. This is likely due to the ability of LSTM to remember long-term dependencies over multiple "hops" of the graph.

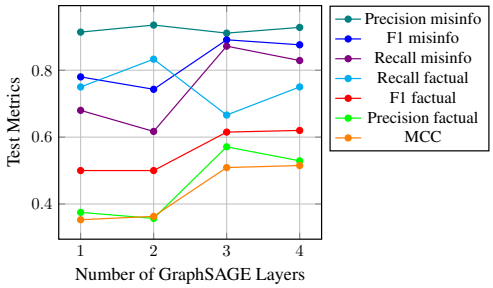
	mean agg.	LSTM agg.
factual precision	0.357	**0.75**
misinfo precision	0.935	**1.0**
factual recall	0.833	**1.0**
misinfo recall	0.617	**0.914**
factual F_1	0.5	**0.857**
misinfo F_1	0.744	**0.956**
MCC	0.363	**0.828**

Table 3: Test set metrics for mean and LSTM aggregators on the weak training set and 4 GraphSAGE layers after 10 epochs.

Unlike Nielsen and McConville (2022), which achieved the best performance for two GraphSAGE layers, we are able to observe the best performance in terms of almost all metrics measured for four GraphSAGE layers. Four GraphSAGE layers correspond to four "hops" in the graph depicted in Figure 2. This is likely due to the graph structure, as channel nodes are never directly connected. The four-hop neighborhood of a Telegram message



(a) Varying the number of epochs.



(b) Varying the number of GraphSAGE layers.

Figure 5: Test set metrics across two training configurations: (a) different numbers of training epochs using LSTM aggregators and 4 GraphSAGE layers. after 100 iterations, and a weight decay of 1e-5; (b) different numbers of GraphSAGE layers using mean aggregation. 10 epochs, learning rate 1e-3 with the same learning rate scheduler.

contains all messages of the same channel and all messages of the channels they are being forwarded to. We were unable to test more than four GraphSAGE layers due to hardware restrictions. Figure 5 depicts the results for different numbers of GraphSAGE layers and numbers of epochs.

Hence, the overall best-performing architecture is the one using LSTM aggregation and 4 GraphSAGE layers. Due to hardware restrictions, we use 10 epochs for the experiments in the following sections.

5.4 Comparison of GNN and Text-only Model

	graph-based	text-based
factual precision	**1.0**	0.714
misinfo precision	**0.979**	0.943
factual recall	**0.923**	0.833
misinfo recall	**1.0**	0.893
factual F_1	**0.960**	0.769
misinfo F_1	**0.989**	0.917
MCC	**0.950**	0.691

Table 4: Test set metrics for the text-only baseline in comparison with the graph baseline. The graph baseline uses an LSTM aggregator and 4 GraphSAGE layers.

In this Section, we compare the GNN model depicted in Figure 3 to the text-based model depicted in Figure 4. Table 4 shows the different metrics for the two baselines. The GNN model outperforms the text-based model for all metrics. We achieved a 95% MCC score for the graph-based model and 69.1% MCC for the text-based model. Our results are comparable to those of Mahmud et al. (2022), who achieve 78.12% test accuracy for a text-based model and 96.99% test accuracy for a GraphSAGE model for the classification of misinformation. The results from this section verify our hypothesis (1) from Section 4 that taking additional network information into account improves performance over the text-only misinformation classification baseline.

5.5 Effect of using View and Subscriber Counts

	incl. counts	w/o counts
factual precision	**1.0**	0.923
misinfo precision	**0.979**	0.978
factual recall	**0.923**	**0.923**
misinfo recall	**1.0**	0.978
factual F_1	**0.960**	0.923
misinfo F_1	**0.989**	0.978
MCC	**0.950**	0.901

Table 5: GNN applied to the datasets including subscriber and view counts and without including them compared by their metrics. The model uses an LSTM aggregator and 4 GraphSAGE layers.

	weak data	strong data
factual precision	**1.0**	**1.0**
misinfo precision	**0.979**	0.974
factual recall	**0.923**	0.875
misinfo recall	**1.0**	**1.0**
factual F_1	**0.960**	0.933
misinfo F_1	**0.989**	0.987
MCC	**0.950**	0.923
ECE	**0.033**	0.051

Table 6: GNNs trained on the weak and strong datasets, compared by their test metrics. The model uses an LSTM aggregator and 4 GraphSAGE layers.

To compute the node features used in the GNN model in all previous experiments, we concatenated the M3-embedding with additional metadata.

Channel	C_{D_f}	Out	In
Eva Herman Offiziell	17,522	16,420	1,102
Tagesereignisse der Offenbarung	13,617	1,084	12,533
AUF1	12,969	12,966	3
Impfen-nein-danke.de	11,424	437	10,987
Freie Sachsen	11,290	11,157	133

Table 7: Top 5 channels by forward-degree centrality (C_{D_f})

The embedding of the channel name was concatenated with the number of subscribers. The message embedding was concatenated with the number of views. In this Section, we removed the view and subscriber counts to test if the model performs worse. The results can be seen in Table 5. The model that uses only M3-embeddings and does not have access to view and subscriber counts on the right performs slightly worse for all metrics except factual recall. This verifies the hypothesis (2) from Section 4.

5.6 Weak and Strong Labels

In this section, we compare the weak and strong datasets. Weakly annotated datasets introduce some noise because there are training examples that have incorrect labels. In some cases, this may lead to perturbations of the classifier (Dehghani et al., 2017), but in many cases, the results are still promising (Tekumalla and Banda, 2023). To test if the weak dataset perturbs the classifier, we manually annotated the test set. Table 6 shows a comparison of the metrics over the weak and strong datasets. The classifier performs similarly on both datasets, which suggests that there are no strong perturbations, verifying our hypothesis (3) from Section 4.

In this experiment, we also computed the ECE for both classifiers, trained on the weak and the strong dataset. Both values are below 0.1, which implies that both models are calibrated well. It remains to be seen in future work if we can confirm these results with a larger weakly-annotated dataset.

6 Additional network analysis

To illustrate the potential of the *MisinfoTeleGraph* dataset for network analysis, we explore structural properties of the message forwarding graph. The dataset includes forwarding relations between Telegram messages and channels, allowing for classic social network analysis such as centrality computations.

Inspired by Das et al. (2018) and Landherr et al. (2010), we computed several centrality measures using the Neo4j Graph Data Science (GDS)[4] library, including variations of *degree centrality* and *betweenness centrality*. These measures highlight the most influential Telegram channels in terms of content dissemination and information flow. Since degree centrality takes all edges into account, we introduce a variant of degree centrality, which we named *forward-degree centrality*. This metric specifically counts the number of edges that represent content forwarding actions. Unlike traditional degree centrality, which includes all edge classes, forward-degree centrality captures only the edges from the `FORWARDED` class. This measure allows to capture information propagation across the platform, reflecting how actively a Telegram channel participates in origination and redistribution patterns of misinformation-related messages.
Table 7 shows the top-ranked channels according to this measure. We also differentiate between in-degree and outdegree as is usually done for degree centrality.

Notably, channels like *AUF1* and *Freie Sachsen* act as *broadcast hubs* with high outgoing edge counts, while others like *Tagesereignisse der Offenbarung* mostly redistribute external content. This asymmetry illustrates distinct roles in the misinformation ecosystem — original content creators versus amplifiers — and offers interpretable context for GNN-based classification. Additional centrality metrics, extended tables, and Cypher queries are available in Appendix B.

7 Discussion

The evaluation of our GNN-based misinformation detection model on the MisinfoTeleGraph dataset has yielded several key insights.

Quantitative evaluation showed that our GNN-based model outperformed a text-only baseline.

[4]https://github.com/neo4j/graph-data-science

The graph-based approach achieved an MCC of 0.95 compared to 0.69 for the text-only model, confirming that incorporating network structure improves misinformation classification. Additional experiments with different numbers of GraphSAGE layers indicated that four layers provided the best performance, likely due to the specific Telegram message forwarding network structure. Moreover, the use of an LSTM aggregator consistently outperformed mean aggregation, underscoring the importance of long-term dependency capture in graph-based misinformation detection.

Additionally, we identified cross-lingual capacities of the chosen embeddings, successfully matching German claims with messages in English and Russian on multiple samples. However, qualitative evaluation revealed limitations in handling logical entailment, particularly in differentiating specific statistical claims related to COVID-19 and distinguishing actors in conflicts like the Gaza war. Finally, we noticed that statistical claims that are often found in health-related misinformation, this topic remains hard to classify.

8 Conclusion

This study shows that integrating network information into misinformation detection models improves performance over text-only approaches. We present the MisinfoTeleGraph dataset and a reproducible baseline to support future research. Our findings highlight AI's potential in fact-checking, while acknowledging its limits in logical entailment and bias.

AI should assist, not replace, human verification, especially as its generative power still outpaces its detection, reinforcing the need for media literacy and broader misinformation countermeasures. Future work should focus on multi-modal detection, better weak annotations, and ethical deployment in sensitive contexts to build more robust misinformation detection systems.

Limitations

One of the main limitations of this study is the relatively small dataset size. The weakly annotated dataset contains 873 message-claim pairs, and the strongly annotated dataset consists of only 651 pairs. The small dataset size may contribute to potential overfitting and could lead to inflated performance metrics. Future work should aim to scale up the dataset by increasing the number of similarity scores computed per claim and exploring additional sources for annotation. Additionally, misinformation often spreads through multi-modal content such as images and videos, which were not considered in this study. Integrating multi-modal features into the dataset could further improve misinformation detection models.

Another issue relates to data redundancy. During annotation, many messages were found to be thematically similar due to message forwarding and minor text modifications. This raises concerns about potential data leakage, where similar messages may appear in both the training and test sets. Implementing stricter data-splitting techniques, such as clustering similar messages before partitioning, could help mitigate this risk.

The weak annotation approach used in this study was computationally intensive and thereby renders a future expansion of the dataset difficult. The current method relies on computing similarity scores between messages and claims using M3-embeddings, which is effective but slow. Future research should explore hybrid retrieval methods, such as combining BM25 for fast pre-selection with M3-embeddings for precise matching. While GraphSAGE was effective in capturing network structures, alternative GNN architectures could further enhance performance. Graph Attention Networks (GAT) or Graph Isomorphic Networks (GIN) may provide improvements by learning more complex interactions within the network. Additionally, techniques such as neighborhood extension via k-nearest neighbors could help address issues related to low-degree nodes, ensuring that nodes with fewer connections still receive sufficient contextual information.

Finally, deploying a GNN-based misinformation detection model in real-world settings presents challenges due to the need for network information. Unlike text-based models that require only message input, GNNs rely on the surrounding network structure. To facilitate deployment, a continuously updated graph database representing the Telegram ecosystem would be necessary. However, integrating the model into an online fact-checking system or browser extension could provide users with real-time misinformation alerts and be used for selecting check-worthy occurrences for fact-checkers to consider. It would be especially valuable to create cross-lingual and cross-platform graphs to identify coordinated campaigns across languages and different social media websites.

Acknowledgement

The work on this paper was mainly performed in the scope of the "noFake" project funded by the German Federal Ministry of Research, Technology and Space (BMFTR) under Award Identifier F16KIS1519, while the first author was still affiliated with TU Berlin.The manuscript was completed under "news-polygraph" project (BMFTR, reference: 03RU2U151C).

References

Google Fact Check Tools API. 2024. Google fact check tools api. https://developers.google.com/fact-check/tools/api/. Accessed: 2025-04-07.

Jason Baumgartner, Savvas Zannettou, Megan Squire, and Jeremy Blackburn. 2020. The pushshift telegram dataset. In *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*, pages 840–847. AAAI Press.

Alexandre Bovet and Peter Grindrod. 2020. The activity of the far right on telegram. *ResearchGate preprint*, pages 1–19.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216.

Kousik Das, Sovan Samanta, and Madhumangal Pal. 2018. Study on centrality measures in social networks: a survey. *Soc. Netw. Anal. Min.*, 8(1):13.

Data4Transparency. 2024. Data4transparency. https://data4transparency.com/. Accessed: 2024-08-17.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 65–74. ACM.

Yingtong Dou, Kai Shu, Congying Xia, Philip S. Yu, and Lichao Sun. 2021. User preference-aware fake news detection. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2051–2055. ACM.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tarek Hamdi, Hamda Slimi, Ibrahim Bounhas, and Yahya Slimani. 2020. A hybrid approach for fake news detection in twitter based on user features and graph embedding. In *Distributed Computing and Internet Technology - 16th International Conference, ICDCIT 2020, Bhubaneswar, India, January 9-12, 2020, Proceedings*, volume 11969 of *Lecture Notes in Computer Science*, pages 266–280. Springer.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.

Yi Han, Shanika Karunasekera, and Christopher Leckie. 2020. Graph neural networks with continual learning for fake news detection from social media. *CoRR*, abs/2007.03316.

Srinidhi Hiriyannaiah, A.M.D. Srinivas, Gagan K. Shetty, Siddesh G.M., and K.G. Srinivasa. 2020. Chapter 4 - a computationally intelligent agent for detecting fake news using generative adversarial networks. In Siddhartha Bhattacharyya, Václav Snášel, Deepak Gupta, and Ashish Khanna, editors, *Hybrid Computational Intelligence*, Hybrid Computational Intelligence for Pattern Analysis and Understanding, pages 69–96. Academic Press.

Josef Holnburger. 2023. *Chronologie einer Radikalisierung - Wie Telegram zur wichtigsten Plattform für Verschwörungsideologien und Rechtsextremismus wurde*. CeMAS.

Philip N. Howard, Bharath Ganesh, Dimitra Liotsiou, John Kelly, and Camille François. 2019. The IRA, social media and political polarization in the united states, 2012-2018. Technical report, U.S. Senate Documents.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multim. Tools Appl.*, 80(8):11765–11788.

Andrea Landherr, Bettina Friedl, and Julia Heidemann. 2010. A critical review of centrality measures in social networks. *Bus. Inf. Syst. Eng.*, 2(6):371–385.

Jiawen Li, Yudianto Sujana, and Hung-Yu Kao. 2020. Exploiting microblog conversation structures to detect rumors. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5420–5429, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yang Liu and Yi-fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the*

30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 354–361. AAAI Press.

Fahim Belal Mahmud, Mahi Md. Sadek Rayhan, Mahdi Hasan Shuvo, Islam Sadia, and Md. Kishor Morol. 2022. A comparative analysis of graph neural networks and commonly used machine learning algorithms on fake news detection. *CoRR*, abs/2203.14132.

Rahul Mishra. 2020. Fake news detection using higher-order user to user mutual-attention progression in propagation paths. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 2775–2783. Computer Vision Foundation / IEEE.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M. Bronstein. 2019. Fake news detection on social media using geometric deep learning. *CoRR*, abs/1902.06673.

Massimo La Morgia, Alessandro Mei, and Alberto Maria Mongardini. 2023. Tgdataset: a collection of over one hundred thousand telegram channels. *CoRR*, abs/2303.05345.

Dan Saattrup Nielsen and Ryan McConville. 2022. Mumin: A large-scale multilingual multimodal fact-checked misinformation social network dataset. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3141–3153. ACM.

Jeremy Nixon, Mike Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring calibration in deep learning. *CoRR*, abs/1904.01685.

Richard Rogers. 2020. Deplatforming: Following extreme internet celebrities to telegram and alternative social media. *European Journal of Communication*, 35(3):213–229.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3):171–188.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36.

Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. 2019. The role of user profiles for fake news detection. In *ASONAM '19: International Conference on Advances in Social Networks Analysis and Mining, Vancouver, British Columbia, Canada, 27-30 August, 2019*, pages 436–439. ACM.

Veronika Solopova, Viktoriia Herman, Christoph Benzmüller, and Tim Landgraf. 2024. Check news in one click: NLP-empowered pro-kremlin propaganda detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 44–51, St. Julians, Malta. Association for Computational Linguistics.

Abdullah All Tanvir, Ehesas Mia Mahir, S M Asiful Huda, and Shuvo Barua. 2020. A hybrid approach for identifying authentic news using deep learning methods on popular twitter threads. In *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, pages 1–6.

Ramya Tekumalla and Juan M. Banda. 2023. Leveraging large language models and weak supervision for social media data annotation: An evaluation using COVID-19 self-reported vaccination tweets. In *HCI International 2023 - Late Breaking Papers - 25th International Conference on Human-Computer Interaction, Proceedings, Part III*, volume 14056 of *Lecture Notes in Computer Science*, pages 356–366. Springer.

Santosh Kumar Uppada, K. Manasa, B. Vidhathri, R. Harini, and B. Sivaselvan. 2022. Novel approaches to fake news and fake account detection in osns: user social engagement and visual content centric model. *Soc. Netw. Anal. Min.*, 12(1):52.

Aleksandra Urman and Stefan Katz. 2022. What they do in the shadows: examining the far-right networks on telegram. *Information, communication & society*, 25(7):904–923.

Claire Wardle and Hossein Derakhshan. 2017. Information disorder: Toward an interdisciplinary framework for research and policymaking. Technical report, Council of Europe.

Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 637–645. ACM.

Xiaoyu Yang, Yuefei Lyu, Tian Tian, Yifei Liu, Yudong Liu, and Xi Zhang. 2020. Rumor detection on social media with graph structured adversarial learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1417–1423. International Joint Conferences on Artificial Intelligence Organization. Main track.

Qiang Zhang, Aldo Lipani, Shangsong Liang, and Emine Yilmaz. 2019. Reply-aided detection of misinformation via bayesian deep learning. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2333–2343. ACM.

Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew J. Hirn. 2021. Magnet: A neural network for directed graphs. In *Advances in*

Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 27003–27015.

Xinyi Zhou, Jindi Wu, and Reza Zafarani. 2020. SAFE: similarity-aware multi-modal fake news detection. In *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part II*, volume 12085 of *Lecture Notes in Computer Science*, pages 354–367. Springer.

A Examples for Cross-lingual Message-claim Pairs

Message	Claim
The United States is sending 2 thousand marines from the rapid reaction brigade to the shores of Israel...	Tausende von US-Marines oder Soldaten sind gerade in Israel gelandet (misinformation – DPA).
NEW – Large German health insurance company analyzed data from 10.9 million insured individuals regarding vaccination complications. "According to our calculations, we consider 400,000 visits to the doctor by our policyholders because of vaccination complications...	Bei der Techniker Krankenkasse seien im Jahr 2021 knapp 440.000 Fälle von Impfnebenwirkungen erfasst worden. In Blog-Artikeln werden die Zahlen mit Werten für 2019 und 2020 verglichen und mit Impfschäden in Verbindung gebracht. (misinformation – CORRECTIV)
NEW – U.S. CDC has quietly deleted the statement that the "mRNA and the spike protein do not last long in the body" from their website...	US-Behörde CDC gibt, dass mRNA und Spikeprotein lange im Körper verbleiben und löscht Entwarnung zu Corona-Impfstoffen von ihrer Webseite. (misinformation – DPA)
Экономика России приходит в упадок – Путин загоняет свою страну в пропасть ("Russia's economy is in decline – Putin is driving his country into the abyss" – DeepL translation)	Russlands Kriegswirtschaft: Putin ruiniert sein Land (newspaper article – taz)

Table 8: Cross-lingual message–claim pairs

B Graph Network Analysis

In this appendix, we propose exemplary graph network analyses that can be done using the MisinfoTeleGraph dataset.

Channel	Subscribers	Degree Centrality
AUF1	252,897	12,969
Eva Herman Offiziell	185,259	34,835
Freie Sachsen	148,628	11,290
Tagesereignisse der Offenbarung	2,045	46,925
WELT	547	86,962

Table 9: Top 5 Telegram channels by subscriber count and degree centrality

B.1 Degree Centrality Analysis

Degree Centrality captures the immediate influence of a node by counting its direct connections (Das et al., 2018). Formally, it is defined as

$$C_D(x) = d_x \tag{2}$$

where d_x is the degree of the node. It is one of the centrality measures that is the easiest to compute in $\mathcal{O}(n)$ time because the algorithm iterates over every node once and counts the number of nodes to which the node is linked.

The following cypher query was used to compute the Degree Centralities.

1: Cypher query for computing Degree Centrality

```
CALL gds.degree.write(
  'messages_and_channels',
  { writeProperty: 'degree' }
) YIELD centralityDistribution,
    nodePropertiesWritten
RETURN centralityDistribution.min as minScore,
      centralityDistribution.mean as meanScore,
      nodePropertiesWritten
```

Running the degree centrality query took 41992 ms. The minimum score was 0, which means that there are isolated nodes that do not have neighbors, and the mean score was 2.96. Since a Telegram message can only be part of one Telegram channel, a high degree means that a Telegram message has been forwarded many times.

Table 10 depicts the ten Telegram channels with the highest degrees, where the "WELT" channel is by far the channel with the highest degree. Note that there are different edge types that channels have that are counted here. They are ingoing and outgoing `FORWARDED` edges and `IS_PART_OF` edges and not all Telegram messages for each channel are included in the dataset. That means that the Degree Centrality of a Telegram channel can be interpreted

C_D	Channel	Subscribers
86962	WELT	547
46925	Tagesereignisse der Offenbarung	2045
46827	impfen-nein-danke.de offiziell	11093
45146	BILD	492
39649	OutoftheBoxTV_DerIrrsinnhatProgramm	4548
35872	Schuberts Lagemeldung - Stefan Schubert Offiziell	36247
34835	Eva Herman Offiziell	185259
33814	Aufgewacht	75
32844	Alternative News	694
31728	Nyx News | Ukraine	1123

Table 10: Top 10 Degree Centrality (C_D) of Telegram channels

as how influential a Telegram channel is according to the topics in the fact-checking and news articles.

B.2 Forward-Degree Centrality Analysis

To determine influential Telegram channels, the number of how many messages forwarded from or to them, the `FORWARDED` edge type is an interesting feature. Therefore, we computed a new measure, forward-degree centrality, which is defined as

$$C_{D_f}(x) = d_{x_f} \tag{3}$$

where d_{x_f} is the degree of x when only taking the `FORWARDED` edge type into account. The measure can be interpreted as how influential a Telegram channel is in spreading information that is related to the fact-checking articles. We used the following Cypher projection where the `IS_PART_OF` relations are dropped, and computed the degree centrality on the graph projection:

2: Cypher query for computing forward-degree centrality

```
CALL gds.graph.create.cypher(
    'messages_and_channels_forwards',
    'MATCH (n) where (n:TGMessage and n.degree >
        1) or n:TGChannel
    RETURN id(n) AS id',
    'MATCH (n)-[e:IS_FORWARDED_FROM |
        IS_FORWARDED_TO]-(m)
    RETURN id(n) AS source, e.weight AS weight,
        id(m) AS target')
```

B.3 Ingoing and outgoing edges

The number of outgoing and ingoing edges can vary a lot in some cases, as can be seen in Table 11, where many channels have only a few ingoing edges, but a lot of outgoing edges. This might indicate that they create a lot of content that gets frequently forwarded, but do not typically forward messages from other channels themselves. On the other side, e.g. “Impfen Nein Danke“, “Tagesereignisse der Offenbarung“ are channels that often forward information but do not create new original content.

B.4 Betweenness centrality

Betweenness centrality is a measure that determines the actor that controls information among other nodes via connecting paths (Das et al., 2018).

The Betweenness centrality $C_B(x)$ of a node x is defined by

$$C_B(x) = \sum_{u \neq v \in \mathcal{V}(G)} \frac{\sigma_{uv}(x)}{\sigma_{uv}} \tag{4}$$

where σ_{uv} is the number of shortest $u - v$ paths and $\sigma_{uv}(x)$ is the number of shortest $u - v$ paths that contain x. Computing the Betweenness centrality for a graph with n nodes and m edges has a time complexity of $\mathcal{O}(nm)$ (Das et al., 2018). For the graph created from our data, computing the Betweenness scores took around 2 months.

The following cypher query was used to compute the Betweenness Centralities:

3: Cypher query for computing Betweenness Centralities

```
CALL gds.betweenness.write('
     messages_and_channels',
    { writeProperty: 'betweenness' })
    YIELD centralityDistribution,
        nodePropertiesWritten
    RETURN centralityDistribution.min AS
        minimumScore,
    centralityDistribution.mean AS meanScore,
        nodePropertiesWritten
```

The results are depicted in Table 12. The ten Telegram channels with the highest Betweenness centrality are either part of the 10 channels with the highest degree centrality or with the highest forward-degree centrality.

C_B	Channel	Subscribers
1815135347746	Tagesereignisse der Offenbarung	2045
1540030673515	AUF1	252897
1401343526977	Eva Herman Offiziell	185259
1344917542040	Aufgewacht	75
988463077140	WELT	547
914524232655	Freie Sachsen	148628
811428932945	impfen-nein-danke.de offiziell	11093
810381539513	henning rosenbusch - channel	65474
709972834297	Mäckle macht gute Laune	130755
689059267090	OutoftheBoxTV	4548

Table 11: Top 10 Forward-Degree Centrality (C_{D_f}) of Telegram channels

C_{D_f}	**Out**	**In**	**Channel**	**Subscribers**
17522	16420	1102	Eva Herman Offiziell	185259
13617	1084	12533	Tagesereignisse der Offenbarung	2045
12969	12966	3	AUF1	252897
11424	437	10987	impfen-nein-danke.de offiziell	11093
11290	11157	133	Freie Sachsen	148628
10139	7537	2602	Mäckle macht gute Laune	130755
10192	30	10162	OutoftheBoxTV_DerIrrsinnhatProgramm	4548
9576	9549	27	henning rosenbusch - channel	65474
8575	7499	1076	Haintz.Media #FreeAssange	81527
8135	511	7624	RBK - Ceterum censeo NATO esse delendam! Raus aus der NATO!	2045

Table 12: Top 10 Betweenness Centrality (C_B) of Telegram channels