Carnegie Mellon University

# Group Project: False News Detections
## Assignment 4 Report

Team 3
Sophia Kurz,Yaning Wu
(11-711) Advanced NLP
Dr. Sean Welleck
25 April 2025

# Introduction

Fake news has plagued society and the journalism industry since its advent (Rodny-Gumede, 2018), and its rapid amplification on platforms like Twitter makes timely detection especially challenging (Langin, 2018). Traditional manual fact-checking cannot scale to the volume and velocity of social-media streams, motivating automated, data-driven solutions. In this study, we explore three computational paradigms: (1) a Text-Graph Ensemble that fuses linguistic and network features, (2) a Graph Contrastive Learning framework to enhance representation robustness, and (3) a replication of a state-of-the-art Graph Neural Network method (Harby & Zulkernine, 2023) to determine which most effectively discriminates fake from real news on Twitter.

# Background

Computational approaches to fake-news detection have matured significantly over the past decade, driven by both the growing societal impact of misinformation and rapid advances in machine learning. Early methods treated deception detection as a purely linguistic task. They extracted stylistic and semantic features such as word choice, syntax patterns, sentiment polarity, and rhetorical cues from individual posts or articles (Liu, Li, & Li, 2015; Pérez-Rosas et al., 2018). Those text-only models achieved moderate performance on controlled benchmarks but often failed when faced with the diverse vernacular and deliberately manipulative language found in real-world social media streams. Purely linguistic analyses also ignored important contextual signals: who shared the content, how it propagated, and how communities responded.

To address these gaps, researchers began incorporating social and network structure into detection frameworks. They represented posts, users, and interactions as nodes and edges in a graph so that models could capture propagation patterns, user credibility, and community-level diffusion dynamics (Wu, Morstatter, Carley, & Liu, 2019; Shu et al., 2020). Graph neural networks (GNNs) emerged as an effective way to learn representations that integrate textual embeddings with relational features. Monti et al. (2019) showed that a geometric deep learning approach on social-media graphs outperformed state-of-the-art text-only baselines on Twitter and Weibo data. More recent work applied contrastive learning to graph representations. In that setup, models learn to distinguish between augmented views of the same graph, which leads to representations that generalize better under adversarial noise and limited labeled data (Wu et al., 2024; Zhu et al., 2021.)

Despite these advances, systematic comparisons across text-only, graph-based, and contrastive frameworks remain rare, especially on large Twitter data sets where misinformation campaigns are most active. Replication studies of recent GNN architectures are also scarce, leaving open questions about reproducibility and relative effectiveness in practice (Harby & Zulkernine, 2023). Our work fills this gap by evaluating three methods: a Text-Graph ensemble that fuses linguistic and relational features, a graph contrastive learning model designed for robustness, and a direct replication of a state-of-the-art GNN approach. We train and test each method on the same Twitter corpus to provide a head-to-head assessment of their strengths and weaknesses in fake-news detection.

# Methodology

## Data Collection

We replicated the data-collection framework of "A Comparative Analysis of Graph Neural Networks for Fake News Detection" (Harby & Zulkernine, 2023) under current access constraints. Because the original Twitter API endpoints are no longer publicly available, we sourced our tweets from Kaggle. We used the pre-labeled Twitter15 dataset, in which each source tweet carries one of four labels: true, false, unverified, or non-rumor.

For the Text-Graph Ensemble, tab-separated text files were ingested with Pandas and cleaned using a shared clean_text function. Labels were merged by tweet ID to align text and class, forming the dataset for the BERT, TF-IDF graph, and ensemble pipelines. Label files, also plain text with colon delimiters, are read into a second DataFrame and merged on tweet_id to align text and class. This resulting DataFrame of cleaned text and labels is fed into the BERT pipeline, the TF-IDF vectorizer, the graph builder, and the ensemble classifier.

The GNN replication method reads source-tweet text files line by line. Each tab-separated line (TSV) is split into tweet_id and raw text then passed through the shared clean_text routine to lowercase and strip non-alphanumeric characters. Labels are loaded from a separate colon-delimited text file into a dictionary mapping each tweet_id to its class. The code then filters to tweets whose identifiers appear in both streams, yielding aligned lists of IDs, cleaned texts, and labels for TF-IDF feature extraction, graph construction, and GNN training.

## Text-Graph Ensemble Methodology

First, we obtained class probability outputs from two base models, a fine-tuned BERT classifier and a graph-based GAT model, on the same set of tweets. The BERT model was trained to predict one of the four rumor classes ("true," "false," "unverified," "non-rumor") using sequence classification fine tuning, and its test-set softmax probabilities were saved. In parallel, the GAT model was trained on a TF-IDF constructed document term graph of the same tweets; its node-level softmax probabilities were saved.

Next, we aligned the two probability outputs by tweet identifier. We loaded both sets of predictions, extracted their tweet IDs, and computed the intersection of IDs present in both sets. For each shared ID, we stacked the BERT and GAT probability vectors to form a combined feature vector of length (NumClasses × 2). This produced a design matrix X and corresponding label vector y for all tweets with dual model outputs.

We then trained a logistic regression ensemble on X to learn optimal weighting of text and graph signals. Using stratified 5-fold cross validation, we evaluated macro F1 on each fold to guard against class imbalance. Hyperparameters included L2 regularization and a maximum of 1000 solver iterations. After cross validation, we refit the logistic regression on the entire dataset to produce the final ensemble model, which was serialized for downstream inference.

This ensemble procedure leverages complementary strengths of deep contextual embeddings from BERT and relational propagation patterns from GAT, learning a simple linear decision boundary in the joint probability space. Evaluation on held-out data used standard classification metrics, including accuracy, precision, recall, and macro F1, and confusion matrices to assess performance gains over each base model.

## Graph Contrastive Learning Methodology

To further explore structural learning beyond simple graph attention models, we built a bidirectional heterogeneous propagation graph by extracting tree-structured paths from the tweets file. Using this graph, we applied Graph Contrastive Learning (GraphCL) for unsupervised pre training, followed by node classification and evaluation via k-fold cross-validation.

For the construction of the heterogeneous graph, each node contains the following information: a RoBERTa-encoded vector of the tweet text, a sentiment score processed by VADER, the node's degree (i.e., the number of times it was retweeted), and its PageRank score (representing the importance of the node). The edge information includes the cosine similarity between parent and child nodes (cos_sim), the posting time difference between parent and child nodes, the propagation weight, and the propagation direction.

After constructing the heterogeneous graph, we defined the model used, Bidirectional Heterogeneous Graph Transformer. We applied two layers of HeteroConv, with each HeteroConv internally using two TransformerConv layers specifically for the Top-Down (TD) and Bottom-Up (BU) directions. Using this model, we first performed training in an unsupervised manner with GraphCL. GraphCL is an unsupervised (self-supervised) representation learning framework first proposed by You et al. at NeurIPS 2020. Its core idea is similar to SimCLR in computer vision: it treats two randomly augmented views of the same graph (or node) as positive samples, and treats all other graphs (or nodes) as negative samples. By using InfoNCE loss, it pulls the positive samples closer while pushing the negative samples apart, allowing the encoder to learn representations that are robust to both structural and attribute perturbations.

In our code, we applied mask_x( masking 10% of node features) and drop_edge_and_attr( dropping 30% of edges and their attributes), and used NT-Xent to calculate the loss and measure training progress. Finally, in the fine-tuning phase, we incorporated the tweet labels and trained the model based on the pretrained encoder, adjusting parameters accordingly. We evaluated the final model performance using accuracy and macro-F1 score through k-fold cross-validation.

## Harby & Zulkernine Replication Methodology

Building on our shared preprocessed Twitter15 splits, we first mitigated class imbalance by augmenting underrepresented labels: for any class below 90 % of the majority count, we generated synthetic tweets via random replacement of one to three words with WordNet synonyms, then appended these examples to the training set.

Next, we transformed each tweet into a TF-IDF feature vector (unigrams through trigrams, up to 2,500 features), converted the resulting dense matrix into a PyTorch tensor of node attributes, and constructed a k-nearest-neighbor graph (k = 5) by linking each node to its five most similar peers under cosine similarity—thereby producing the edge-index tensor for GNN input.

We then implemented and trained six GNN architectures, BiGCN_A, BiGCN_B, BiGAT, BiSAGE, BiARMA, and BiSGCN, using an 80/10/10 train/validation/test split. Optimization employed Adam (learning rate = 0.005, weight decay = $5\times10^{-4}$) with a cosine-annealing scheduler (up to 200 epochs, patience = 20 on validation loss). For each model, we saved the checkpoint with minimal validation loss and reported test-set accuracy, confusion matrix, and ROC curve.

This streamlined replication adheres to the original study's design while explicitly documenting our augmentation strategy, feature engineering choices, graph construction parameters, and training regimen.

# Results

## Text + Graph Ensemble Results

When applied to the held-out test tweets, the fine-tuned BERT classifier achieved 95.37 percent accuracy (macro F1 = 0.9537), confirming its strong ability to distinguish rumor from non-rumor based on linguistic patterns alone. The standalone graph attention network (GAT) delivered substantially lower performance, around 67 percent accuracy, indicating that relational diffusion signals alone are insufficient for high-fidelity fake-news detection on this dataset. When we combined the two via logistic regression ensembling, performance rose to 77 percent accuracy (macro F1 = 0.77). This ensemble accuracy sits between the two base models, demonstrating that the linear meta-classifier successfully integrates complementary text and graph signals but does not surpass BERT's text-only strength.

## Graph Contrastive Learning Results

After completing GraphCL self-supervised pre training and fine-tuning with stratified 5-fold cross-validation, our bidirectional heterogeneous graph transformer achieved an average test accuracy of 73.69% and a macro F1 score of approximately 0.75. This performance is still significantly better than that of a GAT model trained directly on the same topology (around 67%) and also outperforms simpler GCN variants that do not use bidirectional edges or rich edge attributes. However, there remains a considerable gap compared to the fine-tuned text-only BERT model (95.37%).

One key reason for this gap is that we only had access to the full text of source tweets; a large number of child nodes (retweets or comments) lacked parsable textual content and had no available labels, meaning that most nodes in the graph relied on coarse features propagated from their parent nodes, resulting in overall sparse information. Under such inherently limited conditions, achieving 73.69% accuracy relying primarily on interaction structure and sparse semantic features demonstrates that incorporating structural

information is indeed valuable, and that GraphCL pretraining effectively helped the model learn more robust representations. However, due to the limited data scale and relatively shallow model depth, the contribution of structural signals is still not sufficient to surpass the strong semantic representations provided by textual models like BERT.

### Harby & Zulkernine Replication Results

In our replication of the original GNN study, the simplest graph convolutional variant (BiGCN_A) proved optimal, achieving 84.56 percent accuracy on the Twitter15 test split. The other five architectures, BiGCN_B, BiGAT, BiSAGE, BiARMA, and BiSGCN, each fell in the 78 to 81 percent range. These results corroborate the original finding that even a two-layer GCN can capture useful propagation-based features, but they also reveal substantial variation across GNN designs. BiGCN_A's 84.56 percent suggests a robust balance between model complexity and overfitting risk, whereas deeper or more expressive convolution variants did not yield further gains under our training regimen.

## Conclusion

This project compared three paradigms for fake-news detection on Twitter: a Text-Graph Ensemble, a Graph Contrastive Learning framework, and a replication of state-of-the-art GNN architectures. Our findings show that while fine-tuned text models like BERT remain the strongest individual performers, combining text and graph signals through ensembling improves over graph baselines alone.

The Text-Graph Ensemble achieved 77% macro F1, effectively integrating linguistic and relational features. The GraphCL-based model demonstrated that self-supervised pretraining enhances robustness, achieving 73.69% accuracy despite sparse node information. Meanwhile, the BiGCN_A replication achieved 84.56% accuracy, affirming the effectiveness of lightweight GNN designs.

Overall, while textual features dominate in current datasets, incorporating structural signals meaningfully improves detection performance. Future work should further integrate text and graph modalities, scale contrastive pretraining, and address limitations from incomplete social graph data to build more resilient misinformation detection systems.

## Assignment Notes

We applied new fusion methods that combine deep language representations with graph based propagation features, creating a single meta learner that jointly optimizes both information sources. By moving beyond individual baselines and integrating transformer-based text embeddings with GNN signals, our approach captures complementary patterns that neither modality detects alone. This combined framework offers a technically sophisticated enhancement to fake news detection, illustrating how multiple advanced architectures can be orchestrated to improve performance on an established NLP task.

**This directly addresses project goal (1) by introducing a novel technique for the existing fake-news detection task.**

# Meaningful Overlap

Although both Assignments 3 and 4 addressed fake-news detection on Twitter, Assignment 4 significantly expanded the methodological scope and technical depth. Both Assignments 3 and 4 tackle fake-news detection on the Twitter datasets, so there is topical overlap. Assignment 3 was primarily a literature-review + replication exercise: we re-implemented the six bi-directional GNN variants from Harby & Zulkernine (2023) with a simple k-nearest-neighbor text graph and TF-IDF node features, then analyzed why our reproduced accuracies lagged behind the paper.

Assignment 4 moves beyond that replication in three major ways. (1) Richer graph construction: we switch from a homogeneous k-NN text graph to a *bidirectional heterogeneous propagation graph* whose nodes carry RoBERTa embeddings, VADER sentiment, degree and PageRank, and whose edges encode Δt, weight, direction and cosine similarity. (2) New learning paradigm: we introduce *GraphCL* self-supervised pre-training plus stratified fine-tuning, rather than training GNNs from scratch. (3) Multi-modal baseline: we add a Text + Graph BERT + GAT ensemble to benchmark how far structure-only models are from state-of-the-art text classifiers. Thus, while Assignment 4 builds on the domain and datasets of Assignment 3, it expands the methodology, feature set and experimental scope considerably.

# References


Harby, A. A., & Zulkernine, F. (2023). A Comparative Analysis of Graph Neural Networks for Fake News Detection. *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 1215-1222. https://doi.org/10.1109/COMPSAC57700.2023.00184

Heinis, T., & Ham, D. A. (2015). On-the-Fly data synopses. *ACM SIGMOD Record*, *44*(2), 23-28. https://doi.org/10.1145/2814710.2814715

Langin, K. (2018). Fake news spreads faster than true news on twitter—thanks to people, not bots. *Science*. https://doi.org/10.1126/science.aat5350

Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). Automatic Detection of Fake News (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1708.0710

Rodny-Gumede, Y. (2018). *Fake it till you make it: The role, impact and consequences of fake news*. Springer International. https://doi.org/10.1007/978-3-319-62057-2_13

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22-36. https://doi.org/10.1145/3137597.3137600

Wu, J., Xu, W., Liu, Q., Wu, S., & Wang, L. (2024). Adversarial contrastive learning for evidence-aware fake news detection with graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, *36*(11), 5591-5604. https://doi.org/10.1109/tkde.2023.3341640

Wu, L., Morstatter, F., Carley, K. M., & Liu, H. (2019). Misinformation in social media. *ACM SIGKDD Explorations Newsletter*, *21*(2), 80-90. https://doi.org/10.1145/3373464.3373475

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., & Wang, L. (2021, April 19). *Graph Contrastive Learning with Adaptive Augmentation* [Paper presentation]. WWW '21: Proceedings of the Web Conference 2021, Association for Computing Machinery, NY, United States. ACM. https://dl.acm.org/doi/10.1145/3442381.3449802