Department of Computer Science and Engineering (Data Science)
Subject: Social Network Analysis Laboratory (DJ19DSL8014)

AY: 2024-25

Investigating Rumors on Social Media Using GCNs and SNA

By

60009210189 - Om Uskaikar

60009210191 - Bhuvi Ghosh

60009210192 - Mihir Randive

60009210195 – Hetvi Bhanushali

60009210199 – Isha Shah

A. Introduction:

This project tackles the urgent issue of rumor detection on social media by introducing a hybrid dual-encoder framework that combines transformer models for analyzing user preferences with Graph Convolutional Networks (GCNs) for interpreting social context. Traditional methods often fall short in capturing the complexity of user-content interactions, so this system leverages both exogenous (social network) and endogenous (content-based) information to improve detection accuracy. By integrating user behavior, community dynamics, and content analysis into a unified architecture, the model offers a scalable and comprehensive tool for identifying misinformation, aiding platforms, moderators, and fact-checkers in curbing the spread of false information online.

B. Objective:

- **Content-Based Sentiment and Context Analysis:** Use a Transformer model to analyze tweet content, focusing on sentiment and contextual cues to understand emotional tone and linguistic patterns that influence rumor credibility and spread.
- **Social Network Analysis (SNA) Feature Extraction:** Extract key SNA features—such as user centrality, clustering coefficients, and community structures—to capture user influence and rumor-spreading community dynamics within the Twitter network.
- **Graph Convolutional Neural Network (GCN) for User Interaction Modeling:**

Develop a GCN model to encode user interactions like retweets and replies, creating social embeddings that represent engagement patterns and the structure of information diffusion.

• **Fusion of Content and Social Embeddings:** Implement a fusion mechanism that combines Transformer-based content embeddings with GCN-derived social embeddings, enhancing the model's ability to detect rumors by integrating behavioral and textual signals.

• **Rumor Classification with Soft Parameter Sharing and Community-Level Insights:** Employ a classification layer that uses the fused embeddings to label tweets as rumors or non-rumors, introducing soft parameter sharing between components for coherence, and perform community detection to identify clusters prone to spreading misinformation.

C. Dataset:

This study uses the UPFD (User Profile Fake News Detection) pipeline, specifically the GossipCop dataset, which includes graph-based data for fake news detection on social media. Each entry represents a news article and its propagation structure via tweets.

The dataset includes:

• **gossipcop_fake.csv** and **gossipcop_real.csv** – Contain fake and real news samples respectively.

Each file contains the following fields:

• **id** – Unique identifier for each news item.
• **url** – Source URL of the article.
• **title** – Title of the article.
• **tweet_ids** – Tab-separated list of tweet IDs that shared the article.

Statistics of the UPFD – GossipCop Dataset

• #Graphs: 5464
• #Fake News: 2732
• #Total Nodes: 314,262
• #Total Edges: 308,798
• Avg. Nodes/Graph: 58

For preprocessing, edge weights were assigned based on the frequency of (source, target) interactions. Duplicate edges were merged and weighted to reflect repeated interactions, improving graph quality and training stability. Among the available node features (profile, spacy, bert, content), the **content** feature was selected, combining a 300-dimensional spaCy vector with a 10-dimensional profile vector to form a 310-dimensional node representation. Dataset splits (train, val, test) used native UPFD configurations, with optional transformation and filtering applied as needed.

D. Methodology:

This chapter presents a hybrid architecture that fuses **Graph Convolutional Networks (GCNs)** and **Transformer-based content encoders** for fake news detection. It leverages both **user interaction patterns** and **news semantics**, enhancing interpretability and detection accuracy.

#Architecture Overview

- Dual-pathway system:
  - **Social Graph**: Models user interactions via GCN.
  - **News Content**: Processes textual data with a Transformer.
- Both pathways are fused for classification.

Graph Convolutional Network (GCN)

- **Encodes social structures** using user interaction graphs.
- Components:
  - GCN layers for neighborhood aggregation.
  - Engagement embeddings for user behavior.
- Mathematical update:

$$E_U^{(l+1)} = \sigma(D^{-1/2} A D^{-1/2} E_U^{(l)} W^{(l)})$$

Content Encoder: Transformer

- Extracts semantic features from news content.
- Components:
  - **Self-Attention** for context capturing.
  - **Textual Embedding** for semantic representation.

- Key equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Dynamic Feature Fusion

- Combines GCN and Transformer embeddings dynamically.

Equation:

$$F_{\text{final}} = \sigma(W_s E_U + W_c E_T + b)$$

Classification Layer

- Fused embedding is input to a neural classifier.

Loss:

• **Binary Cross-Entropy (BCE)**.

• **Total Loss**:

$$L = L_T + L_{GCN} + \lambda\Omega(\theta_s, \theta_c)$$

Community and Propagator Detection

Community Detection

- Clusters users based on interactions using **Louvain Method**.
- Objective: Localize rumor-prone groups.

Common Influence Score (CIS)

- Combines metrics:
  - Spread Proportion (SP)
  - Spread Efficiency (SE)
  - Normalized Avg. Path Length (APL)
  - Centrality scores (BC, EC)

- Equation:

$$CIS = SP + SE + \frac{1 - APL}{\text{Max Path}} + \frac{\text{Avg BC}}{\text{Max BC}} + \frac{\text{Avg EC}}{\text{Max EC}}$$

Independent Cascade Model

- Simulates probabilistic spread of fake news over time.
- Captures real-world uncertainty in influence dynamics.

Influence Backtracking

- Uses **Dijkstra's algorithm** to trace fake news origin paths.
- Reverses edge weights for least-cost influence path.

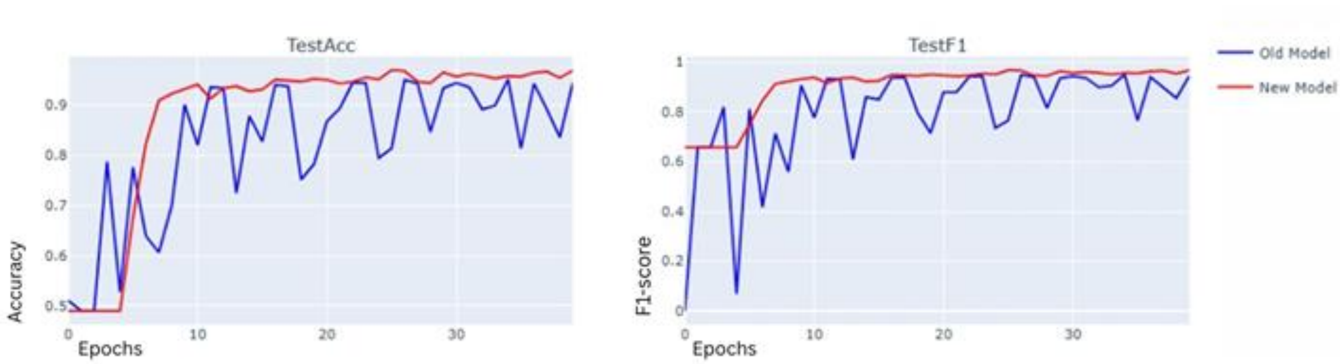
This architecture synergizes **graph-based user modeling** and **semantic analysis**, supported by **dynamic fusion**, **community detection**, and **influence tracing**, providing a comprehensive framework for fake news detection and propagator localization.

E. Results:

This section presents an in-depth analysis of the performance of the proposed GCN+Transformer model, comparing it with the baseline model across several key metrics, and highlighting its efficiency and accuracy.

<u>Model Performance Comparison</u>

Figure 6.1: Test Accuracy and F1 Score Comparison



As illustrated in the above figure, the GCN + Transformer model (represented in red) consistently outperforms the baseline model (blue) in terms of both Test Accuracy and F1 Score. The curves for the proposed model are smoother, indicating better generalization over training epochs.

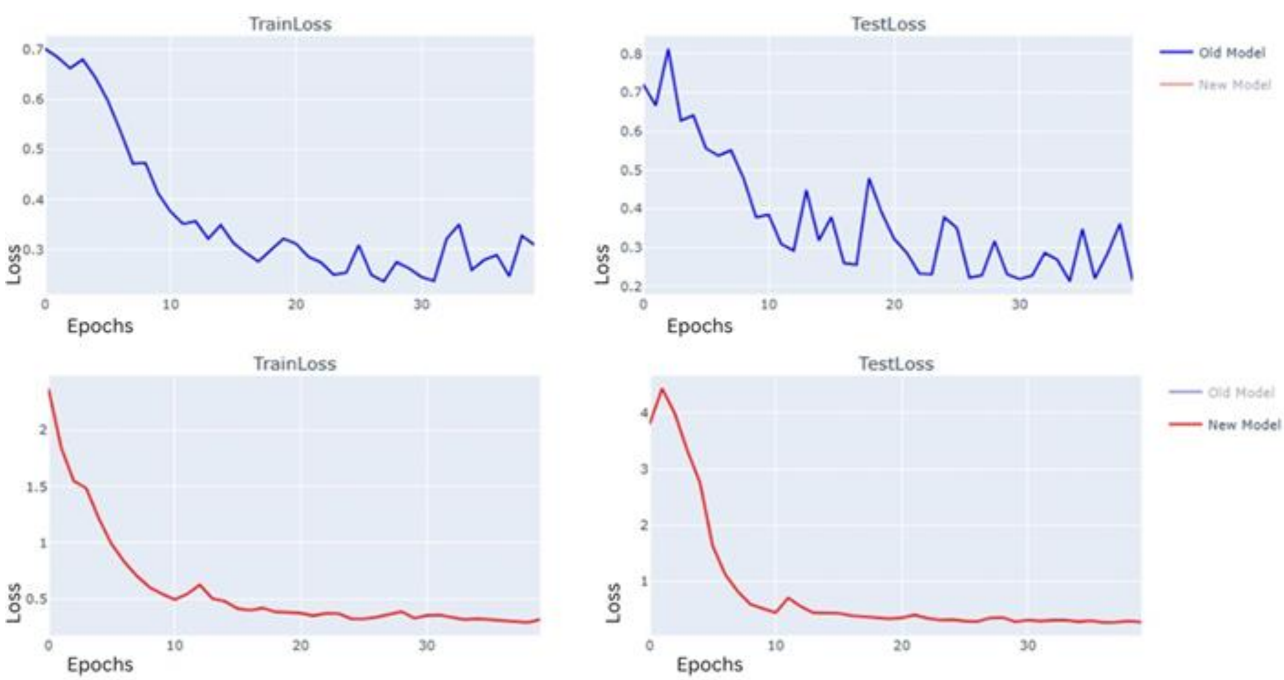
Performance Metrics Comparison:

Metric	Baseline Model	GCN+Transformer Model
Accuracy	0.9377	0.9679
F1 Score	0.934	0.967
Precision	0.972	0.9670
Recall	0.899	0.972

The GCN+Transformer model improves:

- Test Accuracy from 93.77% to 96.79%
- F1 Score from 93.4% to 96.7%
- Recall from 89.9% to 97.2%, showcasing a significant enhancement in the model's ability to identify true positives.

Training and Test Loss Comparison



In the above figure, the comparison of training and test losses over 35 epochs highlights the differences between the baseline (blue curves) and the proposed model (red curves).

- Baseline Model: The training loss decreases from around 0.7 to about 0.25–0.3 by epoch 15 but starts to fluctuate after that. The test loss, starting at 0.8, reduces steadily to 0.2–0.3 but shows noticeable noise in later epochs, suggesting overfitting.
- GCN+Transformer Model: The training loss begins at 2.3, drops rapidly to below 0.4 by epoch 15, and stabilizes around 0.2 in later epochs. The test loss, initially high at 4.2, decreases sharply to under 0.5 in the first 10 epochs and stays consistently below 0.3. This significant improvement in loss behavior underscores the model's ability to generalize better and avoid overfitting.

Analysis of Results

To validate the superior performance of the GCN+Transformer model, statistical tests were conducted to compare it with the baseline. These include McNemar's Test and one-tailed t-tests for F1 Score and Accuracy over the epochs, all at a significance level of α = 0.05.

Statistical Results:

Test	Null Hypothesis (H_0)	Alternative Hypothesis (H_a)	p-value	Result
McNemar's Test	Performance of New Model = Old Model	Performance of New Model > Old Model	0.000003	H_0 Rejected, H_a Accepted
t-Test (F1 Score)	μF1 (New) = μF1 (Old)	μF1 (New) > μF1 (Old)	0.0002	H_0 Rejected, H_a Accepted
t-Test (Accuracy)	μAcc (New) = μAcc (Old)	μAcc (New) > μAcc (Old)	0.0001	H_0 Rejected, H_a Accepted

McNemar's Test for Classification Outcomes

- Null Hypothesis (H_0): The two models have equal classification performance.
- Alternative Hypothesis (H_a): The new model performs better than the baseline.

The p-value of 0.000003 strongly rejects H_0, confirming that the GCN+Transformer model significantly outperforms the baseline in classification.

One-Tailed t-Test for F1 Score Over Epochs

- Null Hypothesis (H_0,F1): No difference in average F1 scores.
- Alternative Hypothesis (H_a,F1): The new model has a higher average F1 score.

The p-value of 0.0002 confirms that the GCN+Transformer model achieves a significantly higher average F1 score, reflecting its better precision-recall balance.

One-Tailed t-Test for Accuracy Over Epochs

- Null Hypothesis (H_0,Acc): No significant difference in accuracy.
- Alternative Hypothesis (H_a,Acc): The new model has higher average accuracy.

The p-value of 0.0001 supports rejecting H_0, indicating a meaningful improvement in accuracy across epochs for the GCN+Transformer model.

Backtracking Results

The backtracking algorithm is used to identify the origin of influence in a diffusion process modeled by the Independent Cascade (IC). In two test cases, the algorithm successfully traced back to Node 4, the source node that initiated the information spread at T = 0. The algorithm's ability to accurately identify the source node in multiple cases validates its effectiveness in reconstructing diffusion paths and understanding how influence spreads.

Test Case 1 (Node 194):

The algorithm identifies the influence path as Node 194 → Node 102 → Node 6 → Node 4, with Node 4 being the origin.

Test Case 2 (Node 196):

The path is reconstructed as Node 196 → Node 193 → Node 94 → Node 4, again confirming Node 4 as the source.

The ability of the backtracking method to consistently trace to the true source node reinforces its value for applications such as misinformation detection, where identifying the origin of false information is crucial.

F. Conclusion:

This chapter presents a comprehensive overview of the outcomes, limitations, and future directions for the GCN+Transformer model. The integration of Graph Convolutional Networks with a Transformer architecture resulted in significant improvements over the baseline, achieving a test accuracy of 0.9679, high recall (0.972), F1-score (0.970), and precision (0.976), along with more stable training dynamics and improved generalization. The model also demonstrated effectiveness in detecting influential nodes and accurately tracing the sources of misinformation. However, limitations remain, including its evaluation on static datasets, limited capacity to detect subtly framed or implicit misinformation, and reduced performance in multilingual or dynamic network environments. Community detection and source tracing may also be less reliable in sparse or evolving networks. To address these gaps, future work will focus on incorporating temporal dynamics, enhancing detection of implicit misinformation, improving robustness in complex network structures, and enabling real-time misinformation tracking and alerting systems.