# Fake News Detection with GNN

Takkelapati Nagendra[1], Charan Boddapati[2]
Madamanchi Chandana[3], and Pranav Vanama[4]

IIT GUWAHATI, Assam, India

**Abstract.** Fake news has become a major challenge in the digital age. Social networks accelerate the spread of misinformation. Traditional detection methods often ignore the propagation structure. Our project leverages Graph Neural Networks (GNNs), focusing on GCN to detect fake news based on propagation behavior.

**Keywords:** Fake News · GCN · FakeNewsNet.

## 1 Introduction

Fake news on the internet is an emerging social and political issue. The propagation of inaccurate information can easily flow through the existing online social network. To combat this problem, traditional approaches that depend on human fact-checkers or NLP can fall short on their generalization or computational capacity.Inside a social network graph, we treat Twitter users and news pieces as nodes, and we model (re)tweeting behavior as edges. Sitting at the cross section of Graph ML and NLP, our GCN method propages nodes' text embeddings and pools them to make a classification. Alternatively, borrowing ideas from Convolutional Neural Network (CNN), our GNN-DP methods iteratively aggregates predictions across sub-graphs to produce the final label.

## 2 Dataset

We use the FakeNewsNet dataset, a comprehensive resource for studying fake news detection and propagation on social media. It contains two major subsets: Politifact and GossipCop, both of which provide labeled news content (True or Fake) along with their corresponding social context—such as original tweets, user retweets, and engagement patterns.

For this project, we focus specifically on the Politifact subset, which is also integrated into the PyTorch Geometric (PyG) library under the name UPFD (User Propagation Fake News Dataset).

UPFD - Politifact Subset: Total Graphs: 314
Fake News: 157 graphs
Real News: 157 graphs
Graph Structure: Each graph is a tree-structured social network.

Node Types:

Root Node: Represents the original news article (tweet).

Leaf Nodes: Represent Twitter users who have retweeted the news.

Edges represent retweet behaviors, forming a hierarchical retweet tree.
If a user retweets news from another user, an edge is formed from the source to the retweeter.

This dataset allows us to model news propagation patterns, making it well-suited for graph-based classification using Graph Neural Networks. The structure and labels enable both node-level and graph-level tasks, but in this project, we treat each graph as a sample for graph-level binary classification (fake or real).

## 3 Methodology

### 3.1 Pre-processing

The UPFD dataset is well-integrated into the PyTorch Geometric (PyG) library, making data loading and preprocessing straightforward. We perform two key preprocessing steps:

- Convert directed graphs to undirected for compatibility with standard GNN models.
- Concatenate node features: a 10-dimensional profile attribute vector with a 768-dimensional BERT embedding of past tweets, resulting in a 778-dimensional feature vector for each node.

### 3.2 Model Architecture

**Graph Convolutional Network (GCN)**

- A 2-layer GCN is implemented (with optional parameters like layers and dropout) to perform graph-level classification.
- We used 2 layers and no dropout due to the relatively shallow graph structures.
- The GCN is flexible and can be adjusted for deeper architectures when needed.

**Differentiable Pooling GNN (GNN-DP)**

- GNN-DP introduces hierarchical learning by training two GNNs in parallel: GNN A learns node embeddings, and GNN B generates cluster assignments.
- Differential pooling layers reduce graph size progressively (from 500 $\rightarrow$ 100 $\rightarrow$ 20 clusters).
- Final predictions are made using mean pooling and a softmax layer.
- We use GraphSAGE + BatchNorm layers in our custom GNN modules for node and cluster embeddings.

## 4 Results

For the GCN model, we obtained a test accuracy of 0.8371 and an F1 score of 0.8552. For the GNN with Differential Pooling model, we obtained a test accuracy of 0.8054 and an F1 score of 0.8037. The performance of the GCN model is slightly better.

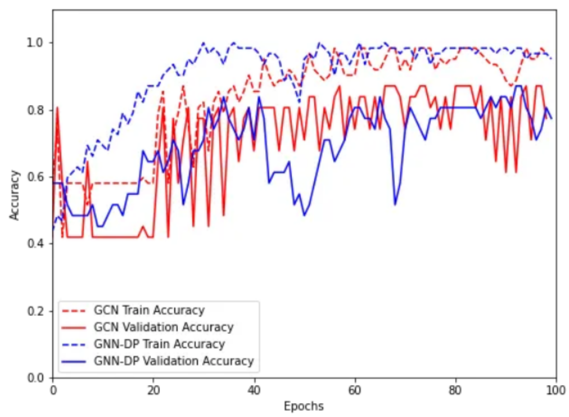


Fig. 1: Training curve.

## 5 Conclusion & Discussion

Our project demonstrates that Graph Neural Networks (GNNs) are highly effective in detecting fake news within social networks. By modeling the propagation structure of news on platforms like Twitter, GNNs can serve as a powerful complement to traditional approaches such as human fact-checking and NLP-based content analysis.

This capability is increasingly critical in the age of advanced language models like GPT-4, which can generate convincing yet misleading content at scale. As misinformation becomes more sophisticated, GNNs offer a promising defense by focusing not just on content but also on how the information spreads—making them well-suited to tackle fake news where it causes the most impact: social media.

**GitHub:** https://github.com/25-pranav-25/Fake-news-detection-with-GNN.git