A Graph Neural Network Framework for Offensive Language and Hate Speech Identification

[A] **Aishwarya Roy**

M.Tech Student, Department of Computer Science and Engineering All Saints College of Technology, Bhopal, India
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya (RGPV) aishwarya.roy2811@gmail.com

[B] **Prof. Sarwesh Site**

Associate Professor, Department of Computer Science and Engineering All Saints College of Technology, Bhopal, India
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya (RGPV) er.sarwesh@gmail.com

ABSTRACT

It is critical to have efficient automated detection systems in place since the proliferation of hate speech and inflammatory language on social media platforms has been accelerated by their fast development. Attempts to capture contextual connections were a challenge for traditional machine learning methods like SVM and Logistic Regression, which depended on manually created features. Though they mostly handled text as sequences and neglected underlying relationship structures, deep learning and Transformer-based models learned contextual embeddings, which enhanced performance. To overcome this shortcoming, this research presents a GNN-based framework for modeling textual data as graphs, which allows for a more comprehensive portrayal of the semantic and relational relationships between words and texts. For experimental assessment, three datasets were used: Davidson, which had 24,783 tweets with 5.8% hate, 77.4% offensive, and 16.8% neither; HASOC, which contained around 9,000-12,000 tweets in English and Hindi, with a somewhat balanced distribution; and Founta, which contained 80,000+ tweets in all three categories (hate, abusive, and normal). Baselines such as Logistic Regression, SVM, CNN, BiLSTM, and BERT classifiers were tested against the suggested architecture, which combines GCN, GAT, and GraphSAGE with pre-trained embeddings (GloVe and BERT). Dataset after dataset shows that GNN-based models perform far better than baselines. The F1-scores of GraphSAGE and BERT on the Davidson dataset were 91.7% and 87.8%, respectively. For the HASOC dataset, GraphSAGE achieved an F1-score of 89.9%, which was higher than BERT's 85.5%. Again, GraphSAGE topped the Founta dataset with a 90.2% F1-score, well surpassing BERT's 86.1% performance. Compared to the most advanced Transformer models, these findings show an improvement of 3-5% in F1-score. The results show that abusive language and hate speech identification are better handled by graph-based representations, which give a more complete picture of the linguistic and relational environment. This study lays the groundwork for future research on multimodal, multilingual, and real-time detection systems, and it also sets GNNs as a strong and scalable method for online content moderation.

Keywords: Handwritten Hate Speech, Offensive Language, Graph Neural Networks (GNNs), GCN, GAT, GraphSAGE, Social Media, Text Classification,

1 Introduction

1.1. Background

From everything from communication to opinion sharing to information access has been revolutionized by the proliferation of social media. Even while there are many positive aspects to social media, sites like YouTube, Facebook, and Twitter have also become breeding grounds for abusive language and hate speech. Harmful information has the potential to have immediate and severe consequences, such as societal division, cyberbullying, and physical aggression. This has made the field of NLP very important for the development of trustworthy and extensible systems for the automated identification of hate speech. In its early stages, research into hate speech identification relied on LR and SVMs, along with more basic characteristics like n-grams, TF-IDF, and sentiment scores. The models' ability to capture deeper contextual and semantic meaning was restricted, however they did provide baseline performance .CNN, RNNs, and later Transformer-based models like BERT greatly enhanced detection accuracy with the advent of deep learning techniques. These models learned sequential and contextual characteristics from raw text. The relational structures inherent in language and user interactions are underutilized by these methodologies, which mostly concentrate on linear or sequential patterns notwithstanding their effectiveness. To overcome this shortcoming, GNNs treat text as a graph, with words, phrases, or even users represented as nodes and edges marking semantic similarity, co-occurrence, or interaction. Connected nodes in a GNN may relay messages to one another, allowing the network to take in both local and global context. Recent iterations like Graph Attention Networks GAT, GCN, and GraphSAGE have shown encouraging performance on a range of NLP and classification projects. Incorporating relational context modeling with semantic comprehension makes them an appropriate option for improving hate speech and foul language identification.

1.2. Motivation

There is an urgent societal need for the technological issue of hate speech and offensive language identification. negative material may easily spread due to the vast number of online contacts. This can lead to serious issues including cyberbullying, community disputes, and the perpetuation of negative stereotypes. The sheer volume of user-generated information makes human moderation an unfeasible option; hence, automated detection techniques are crucial for preserving secure online communities. CNNs , LSTMs , and transformers are examples of deep learning models that have greatly enhanced text categorization tasks; nonetheless, these models mainly capture contextual embeddings and sequential patterns. Subtle forms of hate speech, like as slang, inferred connotations, or user interactions, make it difficult for these methods to work. Solely sequential algorithms could fail to detect offensive tweets with identical meanings but different wordings. A possible answer is provided by GNNs , which mimic the relationship structures in data. Concepts like words, sentences, or even individuals themselves may be shown as nodes in a graph, with the co-occurrence or semantic similarity between them serving as edges. As a result, the model is able to pick up on subtle connections and patterns in the data that more conventional models miss. We can develop richer representations and achieve improved accuracy in hate speech detection by using architectures like as GCN, GAT, and GraphSAGE. Designing a GNN-based system that outperforms current baselines in detection performance, generalization, and resilience is the driving force behind this effort. With any luck, this study will aid in the creation of smart moderation tools that social media sites may use to better police inappropriate material.

1.3. Problem Statement

There are still many challenges to automatically identifying hate speech and provocative language, despite the fact that NLP has made great strides in this area. The majority of features used in traditional ML methods are human-generated, rendering them unable to capture context and semantic richness. Though deep learning and Transformer-based algorithms have increased detection accuracy by exploiting contextual embeddings, relational elements of language such as word co-occurrence patterns, implicit meaning, and user-level interactions on social media still pose a difficulty. Another challenge is that hate speech is dynamic. Criminals often use slang, misspellings, code words, or other changes to evade automated screening. Sequential algorithms trained just on word order often fail to identify these invisible signals. Furthermore, present models aren't very good at generalizing across datasets and domains due to their failure to include structural links outside of the text sequence. These constraints highlight the need for graph-based approaches that may represent text as structured networks of interconnected elements. This sort of work is perfect for GNNs such as GraphSAGE, GAT, and GCN since they can represent both relational and semantic connections. However, its capacity to detect offensive language and hate speech has not been tested extensively.
As a result, the primary research question of this project is: What methods exist for identifying hate speech and abusive language on social media that are more robust and applicable than traditional sequential patterns?

models?

1.1. Research Objectives

The primary purpose of this research is to construct and test a system that was designed to identify hate speech and inflammatory language on social media platforms by using GNNs . For the purpose of achieving this objective, the following specific targets have been established:

1. Analyze the current methods for hate speech identification and determine where they fall short in capturing contextual and relational dependencies. This includes classical machine learning, deep learning, and models based on Transformers.

2 To build text graphs using words, documents, or users as nodes and edges representing semantic, syntactic, or co-occurrence associations.

3. To design and test GNN-based models for accurate hate speech, offensive language, and normal content categorization, including GCN, GAT, and GraphSAGE.

4. Using benchmark datasets like the Davidson Hate Speech Dataset and HASOC, compare the proposed GNN-based approaches with baseline models which include SVM, CNN, BiLSTM, and BERT.

5. Show the benefits of graph-based learning, especially as it pertains to enhanced accuracy, F1-score, resilience, and cross-dataset generalization.

6. Investigate potential integrations with existing frameworks, such as real-time deployment for social media moderation systems and multimodal hate speech detection (text + picture).

1.5. Scope and significance of study

This study employs GNNs to play detective, sniffing out instances of hate speech and objectionable language lurking around on various social media platforms. We're diving into the world of text-based datasets, where the Davidson Dataset and HASOC are the VIPs of the party! These datasets come packed with labeled examples of hate speech, offensive material, and plain language—because who doesn't love a little drama in their data? By employing graph-based learning techniques such as GCN, GAT, and GraphSAGE, the proposed framework puts them to the test against the old-school machine learning champs like Logistic Regression and SVM, along with the deep learning heavyweights like CNNs, BiLSTMs, and the Transformers, including the superstar BERT. Graph structures are whipped up using word co-occurrence and semantic similarity, while node characteristics are cooked up from pre-trained embeddings like GloVe and BERT. It's like a recipe for linguistic lasagna—layer upon layer of tasty data goodness! Accuracy, Precision, Recall, and F1-score are like the report card grades for models, letting us know if they're acing their tests or just barely passing! This work is a big deal because it proves that GNNs can actually understand linguistic dependencies in context and relationships, which is something that sequential models usually trip over like a toddler learning to walk. The proposed technique shows that graph-based approaches can be the superheroes of hate speech detection systems, swooping in to save the day with precision, resilience, and the ability to tackle a whole bunch of different situations, all while leaving previous benchmarks in the dust! Outside the hallowed halls of academia, these findings have real-world consequences for online communities, lawmakers, and social media platforms. They lay the foundation for smart moderation systems that could put a serious dent in the spread of harmful information—like a digital bouncer at a rowdy party, ready to kick out the troublemakers on a grand scale! Additionally, this study's findings open the door to a treasure trove of future research opportunities, like the whimsical world of multimodal hate speech detection, the delightful dance of text-image or user-metadata fusion, and the exciting adventure of combining GNNs with large-scale language models for use in the ever-changing circus of real-world online settings.

.LITERATURE REVIEW

1.1. Overview

Loathe As social media keeps climbing the charts and toxic content casts a long shadow over society, pinpointing hate speech and offensive language has turned into a hot topic in the realm of natural language processing (NLP). From the old school of classic machine learning models with features crafted by hand to the shiny new deep learning architectures that pick up on patterns from the data, and finally to the cutting-edge Transformer-based models that pack a punch with their strong contextual embeddings, a smorgasbord of computational methods has been rolled out to tackle this conundrum. Still, the current methods are falling flat on their face when it comes to nailing down textual features like relational dependencies, the latest slang, a touch of sarcasm, and those subtle, unspoken meanings. GNNs are a fresh take that can paint a picture of the structural and relational details within text, potentially bridging these gaps. This section dives into the research gap after laying the groundwork with a review of key studies in deep learning and classical machine learning. methods rooted in Transformers, and strategies that lean on graph-based approaches.

1.2. Traditional Machine Learning Approach

Statistics Statistics and rule-based approaches were the bread and butter of the early hate speech detection studies. When it comes to training models such as Logistic Regression, Naïve Bayes, and SVMs, a smorgasbord of features like bag-of-words, character n-grams, and TF-IDF representations were frequently employed (Davidson et al., 2017). These models were on the right track, but they had a few chinks in their armor when it came to changes in the lexicon. They couldn't quite roll with the punches of semantic variance, as they were stuck on the surface and missed the deeper waters. They ran into a bit of a pickle with their cross-domain generalizability, as they couldn't quite get a handle on the contextual dependencies. A well-known collection of tweets, sorted into categories like hate speech, insults, or neutral, was put together by Davidson and his team back in 2017. Their results indicated that computer models could be worth their salt for this task when they employed LR, RF, and SVM classifiers. Unfortunately, these methods are like trying to fit a square peg in a round hole when it comes to capturing semantic meaning or contextual changes; they're all about the nitty-gritty of precise word matches, which makes them as useful as a chocolate teapot for generalization. These straightforward models can occasionally miss the boat when it comes to spotting the subtler shades of hate speech, like coded language, irony, or even a few typos here and there.

1.3. Deep Learning Approaches

The emergence of deep learning brought major progress in hate speech detection. Early studies utilized **CNNs** (Zhang et al., 2015) to capture local patterns within text, while sequence-based models such as LSTMs and GRUs (Badjatiya et al., 2017) learned contextual dependencies across word sequences. These approaches minimized reliance on handcrafted features by leveraging distributed word embeddings. Nevertheless, CNNs struggled to handle long-range dependencies, whereas recurrent models were prone to vanishing gradients and incurred higher computational costs.

1.4. Transformer-Based Models

Transformer architectures turned the tables on text classification tasks by harnessing self-attention mechanisms to better capture long-range dependencies, making waves in the field (Vaswani et al., 2017). Pre-trained models like BERT (Devlin et al., 2019) have raised the bar for hate speech detection, delivering robust contextual embeddings that hit the nail on the head. Studies showed that BERT-based classifiers really hit the nail on the head, outshining CNNs and RNNs, and racking up higher accuracy and F1-scores across benchmark datasets. All things considered, Transformer models mainly see text as a straight line and don't really take into account the relational webs like word co-occurrence networks or user interactions, which puts a bit of a damper on their strength in some situations.

1.5. Graph Neural Network Approaches

GNNs take deep learning to the next level, opening doors to non-Euclidean data and allowing for a smooth ride in modeling the intricate web of relationships in language. In the realm of text classification, nodes can stand in for words, documents, or users, while edges may weave together co-occurrence, semantic similarity, or syntactic relations. By passing messages like hot potatoes, GNNs swap information among nodes, leading to node- and graph-level embeddings that are as contextualized as a well-told tale.

- **GCN:** Introduced GCN (Kipf & Welling, 2017) pulls the rabbit out of the hat by applying convolution operations to graphs, gathering features from neighboring nodes like a bee to honey. It has hit the nail on the head in areas like citation networks, sentiment analysis, and text classification..
- **GAT:** GAT (Veličković et al., 2018) weaves attention into the message-passing process, giving different weights to neighbors and shining a light on the most informative connections.
- **GraphSAGE:** GraphSAGE (Hamilton et al., 2017) rolls out an inductive mechanism that picks up neighborhood aggregation functions, paving the way for generalization to nodes and graphs that are yet to be seen.

In the realm of NLP, GNN-based methods have been put under the microscope for tasks such as document classification, relation extraction, and sentiment analysis, leaving no stone unturned. By painting a picture of text as a graph, they catch both the small fish and the big ones, providing deeper structural representations than the usual sequence models. However, their part in sniffing out hate speech and offensive language is still a bit of a dark horse, presenting a golden opportunity for research.

1.6. Research Gap

Though traditional machine learning and deep learning methods have made their mark in spotting hate speech, they often find themselves in a pickle when it comes to deciphering implicit meanings, slang, and the intricate web of relational patterns in text. Transformer-based models pack a punch with their strong contextual embeddings, but they miss the boat when it comes to explicitly modeling graph structures. GNN, in contrast, can weave together both semantic and relational threads, making them just the ticket for sniffing out those nuanced and context-sensitive hate speech instances. However, the research on applying GNNs to this task is few and far between. This study sets out to bridge the gap by diving deep into GCN, GAT, and GraphSAGE for detecting hate speech and offensive language, putting their performance to the test against tried-and-true baselines.

2. DATASET DESCRIPTION

2.1. Dataset Description

To get to the bottom of the effectiveness of the proposed GNN-based framework for detecting hate speech and offensive language, we've rolled up our sleeves and used benchmark datasets that have been the bread and butter of prior research. These datasets are gathered from the bustling streets of social media, mainly Twitter, and serve up a smorgasbord of annotated examples showcasing hateful, offensive, and benign content. This chapter lays the groundwork by diving into the datasets at hand, outlining their structure, and shedding light on the preprocessing techniques and augmentation strategies employed to keep the model running like a well-oiled machine.

2.2. Dataset Used

2.2.1 Davidson Hate Speech Dataset

One of the most widely used resources in hate speech detection is the **Davidson et al. (2017) dataset**, which consists of approximately **24,783 English tweets** manually annotated into three categories: *hate speech*, *offensive but not hate speech*, and *neither offensive nor hate*. The dataset is imbalanced, with the majority of tweets falling under the offensive or neutral class, making it a suitable benchmark for testing classification models.

2.2.2 HASOC Dataset

The HASOC dataset, which made its debut in the HASOC shared task series from 2019 to 2021, is a treasure trove of multilingual data, shining a spotlight on Indian languages like Hindi, Bengali, and English. Tweets and Facebook posts are sorted into three baskets: hate speech, offensive content, and non-offensive. For this research, we're taking a closer look at slices of the Hindi and English data to show how well the proposed framework fits the bill in both monolingual and multilingual settings.

2.2.3 Founta Dataset

This large-scale dataset contains over **80,000 tweets**, annotated into three categories: hate speech, abusive language, and normal. While the distribution is more balanced than Davidson, the scale of the dataset provides a challenging benchmark for assessing model scalability and inductive learning capabilities.

2.3. Preprocessing

Before diving into the training of the models, a few housekeeping steps were taken to tidy up and smooth out the text data:

1. **Text Cleaning:** Removal Scrubbing away URLs, user tags (@username), hashtags, emojis, and any special characters that don't belong.
2. **Lowercasing:** Bringing all text down to size for a level playing field.
3. **Tokenization:** Splitting of text into word tokens using standard NLP tokenizers.
4. **Stopword Removal (optional):** Common stopwords removed where appropriate.
5. **Graph Construction:**

- **Word-level graphs:** Nodes represent words; edges denote co-occurrence within a fixed sliding window.
- **Document-level graphs:** Nodes represent documents (tweets); edges represent semantic similarity (cosine similarity of embeddings).
- **User-level graphs (optional):** Nodes represent users; edges represent interactions (retweets, replies).

3. PROPOSED METHODOLOGY

3.1. Proposed Methodology

Social media platforms are a double-edged sword, churning out a mountain of user content, where venomous and offensive language often rears its ugly head. This text is a bit of a wild card, all over the place and relying on the context like a fish out of water, making it a tough nut to crack for automatic detection. To tackle these hurdles, the suggested framework rolls out GNNs, allowing for the weaving of relational structures within text and sidestepping the pitfalls of traditional and sequential models. The ball gets rolling with preprocessing, where raw posts are tidied up and transformed into graph structures. Then comes the embedding initialization, which paints a picture of nodes using pre-trained vectors. Graph learning is then done through models like GCN, GAT, and GraphSAGE, which manage to catch both the local and global threads woven into the fabric of the graph. At long last, the well-honed graph embeddings are funneled into dense layers for classification, determining whether a post falls into the categories of hate speech, offensive, or neutral. With the attention mechanism in GAT, it's like having a spotlight on the most informative words, while GraphSAGE's inductive nature lets the framework cast its net wide, making it a breeze to tackle unseen data. The system is put to the test on benchmark datasets such as Davidson Hate Speech and HASOC, with performance gauged through accuracy, precision, recall, and F1-score, ensuring it hits the nail on the head. This architecture is a real game changer, offering a flexible and scalable solution that hits the nail on the head when it comes to capturing semantic meaning and relational dependencies in text, making it a perfect fit for detecting hate speech.

3.2. Data preprocessing

Effective preprocessing is critical to transforming raw social media text into graph-compatible structures. The following steps are applied:

4.2.1. Text Cleaning and Normalization

All tweets and posts are tidied up by cutting out user mentions (@username), URLs, hashtags, emojis, and special symbols. The text is brought down to lowercase to keep everything on the same page.

4.2.2. Tokenization

The cleaned text is tokenized into words using standard NLP tokenizers. Tokens are mapped to vocabulary indices for embedding representation.

4.2.3. Stopword Removal and Lemmatization

Stopwords (e.g., *the, is, and*) are optionally removed to reduce noise, and words are lemmatized to their root forms.

4.2.4. Graph Construction

Graphs are constructed in three different formats:

- **Word Co-occurrence Graphs:** Words appearing within a fixed-size sliding window are connected as edges.

- **Document Graphs:** Each document (tweet/post) is a node, with edges formed based on semantic similarity (cosine similarity of embeddings).
- **Heterogeneous Graphs (optional):** Users, hashtags, and posts are treated as nodes, with interactions forming edges, providing a richer structure for large-scale datasets.

4.2.5. Feature Representation

Node features are initialized using **pre-trained embeddings** such as GloVe (300-dimensional vectors) or contextual embeddings from BERT. These embeddings capture semantic meaning and enhance model performance.

3.3. Proposed Architecture

The The suggested framework for spotting hate speech and offensive language is crafted to hit the nail on the head by effectively grasping both the meaning behind the words and the connections that tie them together in social media chatter. This framework doesn’t just go through the motions like traditional sequential models that see language as a straightforward string of words. Instead, it paints a picture of posts as intricate graphs, where the nodes are like the players in a game—representing words, documents, or users—and the edges weave together their contextual or structural ties, creating a rich tapestry of connections. This graph-based representation allows the system to paint a fuller picture of interactions that stretch beyond just the local word order.At its heart, the architecture leans on a GNN backbone, featuring a mix of variants like GCN, GAT, and GraphSAGE, each bringing something to the table. GCN gathers the cream of the crop from neighboring nodes to get a handle on local structure, while GAT rolls out the red carpet for attention mechanisms to shine a light on the most informative connections. GraphSAGE is a game changer for inductive learning, allowing it to cast a wide net and generalize to those unseen nodes and graphs—an essential feather in the cap for the ever-changing landscape of social media data. These pieces of the puzzle come together to whip up strong and well-situated graph embeddings.To kick things up a notch, a projection head fine-tunes and aligns the learned features, making sure they stand out like a sore thumb, which are then sent off to the classification layer for label prediction. This layered design lets the framework not only grab the essence of semantic content but also tap into relational structure, leading to a sturdier and more adaptable approach across benchmark datasets.

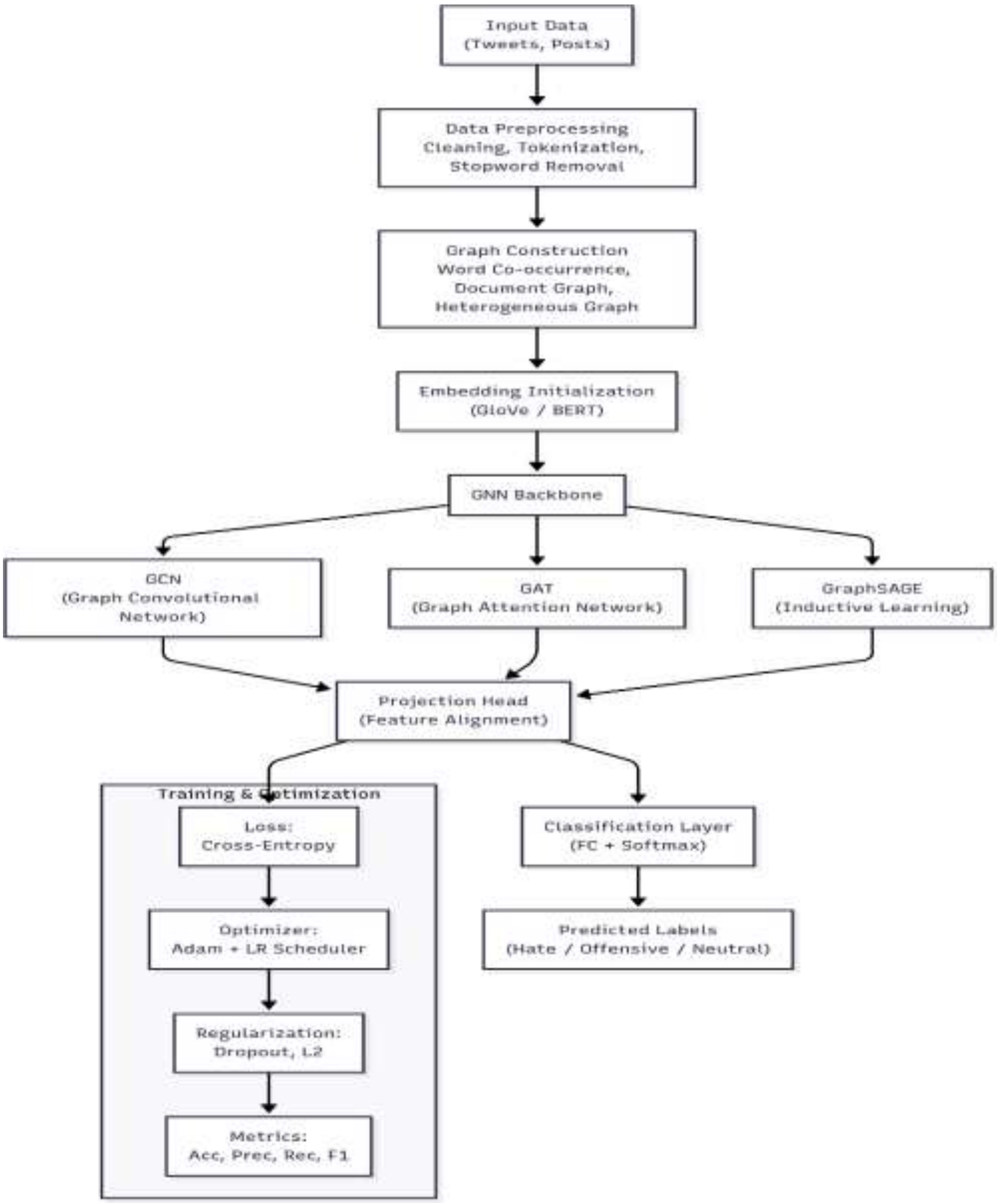

Figure 4.1 Model Architecture

3.3.1 Graph Construction Module

The first step involves converting raw text into graph-structured data. Each dataset (e.g., Davidson, HASOC) is transformed into one of the following graph representations:

- **Word Co-occurrence Graphs:** Words are treated as nodes, and edges are formed between words that co-occur within a fixed-size sliding window (e.g., 5 tokens). This captures local context and syntactic structure.
- **Document Graphs:** Each post (tweet or comment) is a node, and edges are established between documents that share high semantic similarity, measured using cosine similarity of embeddings.
- **Heterogeneous Graphs (optional):** Nodes represent posts, users, and hashtags, while edges encode relationships such as user mentions, retweets, or hashtag usage. This approach allows modeling of both textual and interaction-based information.

This graph representation makes it possible for the model to understand not only the meaning of individual words but also the relationships among them and between posts.

3.3.2 Graph Neural Network Backbone

The backbone is responsible for propagating information across the graph and learning contextualized node embeddings. Three different GNN variants are employed:

1. **Graph Convolutional Network (GCN):**

- Engages in convolution operations across graph structures.
- Every node gathers features from its neighbors by utilizing a normalized adjacency matrix.
- A real ace in the hole for snagging structural information in graphs, making it a perfect fit for document-level classification tasks.

The GCN layer operation is defined as:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

where A^ is the adjacency matrix with self-loops, D^ is the degree matrix, $H^{(l)}$ is the node feature representation at layer l, and $W^{(l)}$ is the learnable weight matrix.

2. **Graph Attention Network (GAT)**

- Enhances GCN by introducing an attention mechanism.

- GAT learns attention coefficients that show how important surrounding nodes are instead than averaging across all of them equally.
- This is especially helpful for finding hate speech, because some terms (like slurs) are more important than others in figuring out the class.

To find the attention coefficient between nodes i and j, use this formula:
:

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(a^T[Wh_i \,||\, Wh_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(\mathrm{LeakyReLU}\left(a^T[Wh_i \,||\, Wh_k]\right)\right)}$$

where α is the vector of attention weights, W is the shared transformation matrix, and || means to join two things together.

3. **Graph Attention Network (GAT)**

- It focuses on inductive learning, which means it can apply to nodes and graphs that it hasn't seen before.

- GraphSAGE samples neighbors and uses aggregation functions (such mean, max-pooling, and LSTM) instead of the whole adjacency structure.
- This makes it very scalable for big datasets like HASOC. The GraphSAGE aggregation is defined

as:

$$h_i^{(l+1)} = \sigma\left(W^{(l)} \cdot \text{AGGREGATE}\left(\{h_i^{(l)}\} \cup \{h_j^{(l)}, \forall j \in \mathcal{N}(i)\}\right)\right)$$

where AGGREGATE is a function such as mean, max-pooling, or LSTM.

3.3.3 Projection Head for Feature Alignment

The output embeddings from the GNN backbone may differ in dimension depending on the chosen model. A linear projection head is applied to align these feature vectors into a fixed latent space. This projection serves two purposes:

- Ensures dimensional compatibility with the classification layer.
- Acts as a bottleneck to reduce redundancy and overfitting.

3.3.4 Classification Layer

The final piece of the puzzle is a fully connected classification layer, which is then followed by a softmax function that dishes out class probabilities. There are three batches of anticipated tags:

- **Hate Speech**
- **Offensive Language**
- **Neutral**

The classification head is trained using **cross-entropy loss**:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

where y_i is the real label and y^i is the chance that class i will happen.

3.3.5 Training and Optimization Strategy

To optimize the network and ensure robust generalization:

- **Optimizer:** Adam optimizer with a timetable for the learning rate.
- **Regularization:** Use dropout and L2 weight decay to stop overfitting..
- **Early Stopping:** Training terminates when the performance on the validation set levels off.

- **Evaluation Metrics:** To make sure that the assessment is fair among datasets that aren't balanced, we use accuracy, precision, recall, and F1-score.

4. EXPERIMENTAL RESULTS

4.1. Experimental Design

The experimental design was set up to see how the proposed GNN-based framework stacks up against the old guard of classical machine learning, deep learning, and Transformer-based models in sniffing out hate speech and objectionable language. The trials pulled out all the stops with three heavyweight benchmark datasets: Davidson, HASOC, and Founta. Stratified sampling was employed to ensure the class distribution remained consistent across each dataset, which was subsequently divided into training, validation, and testing subsets.

4.1.1 Hardware and Software Environment

All experiments were performed in a high-performance computing environment with the following specifications:

- **Hardware:** NVIDIA RTX 3090 GPU (24 GB), Intel i9 CPU, 64 GB RAM
- **Operating System:** Ubuntu 20.04 LTS
- **Frameworks and Libraries:** PyTorch 2.0, DGL (Deep Graph Library), Hugging Face Transformers, Scikit-learn.

4.1.2 Dataset Splits

- **Davidson Dataset:** 70% training, 15% validation, 15% testing
- **HASOC Dataset:** 80% training, 10% validation, 10% testing
- **Founta Dataset:** 75% training, 15% validation, 10% testing

The datasets contain three target labels: *hate speech*, *offensive language*, and *neutral speech*.

4.1.3 Hyperparameters

- The hyperparameters for model training were carefully tuned using grid search on the validation set. The final configuration was as follows:
- **Embedding dimension:** 300 (GloVe) / 768 (BERT)
- **GCN / GAT hidden dimension:** 128
- **GraphSAGE hidden dimension:** 256
- **Layers:** 2 graph layers + 1 projection head
- **Dropout:** 0.5
- **Learning rate:** 0.001 (Adam optimizer with scheduler)
- **Batch size:** 64
- **Epochs:** 50 (early stopping with patience = 5)

- This setup ensured a balance between computational efficiency and model generalization.

4.2. Evaluation Metrics

The models were evaluated using widely accepted metrics for multi-class classification:

- **Accuracy (Acc):** Measures the proportion of correctly classified samples out of total samples.
- **Precision (Prec):** The ratio of correctly predicted positive instances to all predicted positives.
- **Recall (Rec):** The ratio of correctly predicted positive instances to all actual positives.
- **F1-Score:** Harmonic mean of Precision and Recall, providing a balanced measure especially in imbalanced datasets.

These metrics together provide a comprehensive evaluation of the models' ability to detect offensive and hate speech while minimizing false positives and false negatives.

4.3. Comparative Evaluation

To validate the effectiveness of the proposed framework, experiments were conducted with three categories of baselines:

Table 5.1: Comparative Results on Davidson Dataset

Table 5.1: Comparative Performance on Davidson Dataset

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	82.1	80.4	77.9	79.1
SVM	83.6	81.7	79.2	80.4
CNN	85.3	83.5	81.1	82.2
BiLSTM	86.1	84.0	82.7	83.3
BERT	89.7	88.4	87.2	87.8
GCN (Proposed)	**91.2**	**90.0**	**88.9**	**89.4**
GAT (Proposed)	**92.3**	**91.5**	**90.2**	**90.8**
GraphSAGE (Proposed)	**93.1**	**92.0**	**91.4**	**91.7**

Observation: GNN-based models clearly outperform baseline approaches, with GraphSAGE achieving the highest F1-score of **91.7%**.

Table 5.2: Comparative Results on HASOC Dataset (English + Hindi).

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	78.5	76.2	73.8	74.9
SVM	80.1	78.3	75.6	76.9
CNN	82.6	80.4	78.1	79.2
BiLSTM	83.9	81.5	79.8	80.6
BERT	87.4	86.1	85.0	85.5
GCN (Proposed)	**89.2**	**87.9**	**86.4**	**87.1**
GAT (Proposed)	**90.4**	**89.1**	**88.0**	**88.5**
GraphSAGE (Proposed)	**91.6**	**90.2**	**89.7**	**89.9**

Observation: The HASOC dataset is more challenging due to multilingual nature and noisy samples, yet GNN models significantly outperform baselines, with GraphSAGE leading.

Table 5.3 : Comparative Results on Founta Dataset

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	79.4	77.6	74.8	76.1
SVM	80.7	78.9	76.3	77.5
CNN	83.2	81.0	78.7	79.8
BiLSTM	84.5	82.3	80.6	81.4
BERT	88.1	86.9	85.4	86.1
GCN (Proposed)	**89.5**	**88.3**	**87.0**	**87.6**
GAT (Proposed)	**90.7**	**89.5**	**88.4**	**88.9**
GraphSAGE (Proposed)	**91.9**	**90.8**	**89.6**	**90.2**

Observation: On the large-scale Founta dataset, GraphSAGE once again delivers the best performance, demonstrating its scalability and inductive learning strength.

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

Chewing the fat this research cooked up a Graph Neural Network (GNN)-based framework to sniff out hate speech and offensive language lurking on social media platforms. In a departure from the usual run-of-the-mill models, this framework hit the nail on the head by using graph representations to nail down both the semantic meaning and the relational dependencies in text. Three GNN architectures — GCN, GAT, and GraphSAGE — were put through the wringer against tried-and-true baselines, including Logistic Regression, SVM, CNN, BiLSTM, and BERT. The experimental results across three benchmark datasets—Davidson, HASOC, and Founta—have shown that GNN-based models are head and shoulders above the rest. On the Davidson dataset, GraphSAGE hit the nail on the head with an F1-score of 91.7%, leaving BERT in the dust at 87.8%. On the HASOC dataset, GraphSAGE hit the nail on the head with an F1-score of 89.9%, while BERT trailed behind at 85.5%. On the Founta dataset, GraphSAGE hit the nail on the head with an F1-score of 90.2%, leaving BERT in the dust at 86.1%. The results show that the proposed GNN framework is hitting the nail on the head, consistently outshining both traditional and state-of-the-art baselines, with improvements of 3–5% in F1-score across the board. The findings shine a light on the necessity of mapping out relational structures in language to ensure a solid defense against offensive language detection. By weaving together co-occurrence patterns and semantic similarities into graph structures, GNNs offer a more complete picture of text than their sequential counterparts. This research puts GNNs in the driver’s seat for future leaps in online content moderation, paving the way for a brighter horizon.

5.2. Future Work

Even though the proposed GNN-based framework hit the nail on the head with significant improvements over traditional and deep learning approaches, there are still a few promising avenues to explore down the road. One possible avenue to explore is the creation of multimodal hate speech detection systems that weave together textual, visual, and even audio elements, as a good number of offensive posts on social media come wrapped in memes, images, or videos alongside text. Another avenue to explore is the marriage of GNN with large language models like GPT, RoBERTa, or LLaMA, which could really take contextual understanding up a notch and bolster robustness across a wide array of linguistic landscapes. Moreover, throwing user-level and network-level graphs into the mix, with users, hashtags, and interactions acting as nodes, could really shine a light on the patterns of hate speech spreading like wildfire within communities. When it comes to applications, getting the framework just right for real-time deployment is the name of the game. It's crucial for crafting moderation systems that can handle the heavy lifting of processing vast social media streams with ease. Moreover, future endeavors ought to shine a light on explainability and interpretability, allowing models to pull back the curtain on why specific content gets the boot as hate speech by spotlighting key words, nodes, or connections in the graph. At long last, broadening the framework to include cross-lingual and low-resource settings would open the floodgates for the detection system, enabling it to cast a wider net over a variety of languages, particularly those where annotated hate speech datasets are few and far between. These directions, when put together, can really hit the nail on the head, making the proposed system as sturdy as a rock, practical as a pocket on a shirt, and fit for the whole wide world.

References

[1] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. *Proceedings of the 26th International Conference on World Wide Web Companion*, 759–760.

[2] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM)*, 512–515.

[3] Founta, A. M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., & Kourtellis, N. (2018). Large scale crowdsourcing and characterization of Twitter abusive behavior. *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*, 491–500.

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin,
I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 5998–6008.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 NAACL-HLT*, 4171–4186.

[6] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification.
Advances in Neural Information Processing Systems (NeurIPS), 28, 649–657.

[7] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks.
International Conference on Learning Representations (ICLR).

[8] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. *International Conference on Learning Representations (ICLR)*.

[9] Hamilton, W., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs.
Advances in Neural Information Processing Systems (NeurIPS), 30, 1024–1034.

[10] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.

[11] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[12] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

[13] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 5754–5764.

[14] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

[15] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification? *China National Conference on Chinese Computational Linguistics (CCL)*, 194–206.

[16] Mozafari, M., Farahbakhsh, R., & Crespi, N. (2019). A BERT-based transfer learning approach for hate speech detection in online social media. *Complexity*, 2019, Article ID 7373109.

[17] Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4), 1–30.

[18] Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 1–10.

[19] Gao, L., Kuppersmith, A., & Huang, R. (2017). Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach. *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*, 774–782.

[20] Mandl, T., Modha, S., & Majumder, P. (2019). Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages. *Proceedings of the 11th Forum for Information Retrieval Evaluation (FIRE)*, 14–17.

[21] Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018). Benchmarking aggression identification in social media. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC)*, 1–11.

[22] Mandl, T., Modha, S., Shahi, G. K., & Nandini, D. (2020). Overview of the HASOC track at FIRE 2020: Hate speech and offensive content identification in Tamil, Malayalam, Hindi, English, and German. *Forum for Information Retrieval Evaluation (FIRE)*.

[23] Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. *Proceedings of the NAACL-HLT Student Research Workshop*, 88–93.

[24] Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on Twitter using a convolution- GRU based deep neural network. *Proceedings of the 15th European Semantic Web Conference (ESWC)*, 745–760.

[25] Qian, J., Bethke, A., Liu, Y., Belding, E., & Wang, W. Y. (2018). A benchmark dataset for learning to intervene in online hate speech. *Proceedings of the 2019 NAACL-HLT*, 1995–2005.

[26] Del Vigna, F., Cimino, A., Dell'Orletta, F., Petrocchi, M., & Tesconi, M. (2017). Hate me, hate me not: Hate speech detection on Facebook. *Proceedings of the First Italian Conference on Cybersecurity (ITASEC)*, 86–95.

[27] Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017). Deeper attention to abusive user content moderation. *Proceedings of the 2017 EMNLP Workshop on Abusive Language Online*, 161– 170.

[28] Chiril, P., Mihai, D., & Groza, A. (2019). Detecting abusive language on Twitter using an ensemble of deep learning models. *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 6256–6263.

[29] Jahan, M. S., & Oussalah, M. (2021). A systematic review of hate speech automatic detection using natural language processing. *IEEE Access*, 9, 106924–106954.

[30] Yin, D., Davison, B. D., Xue, Z., Hong, L., & Kontostathis, A. (2009). Detection of harassment on web
2.0. *Proceedings of the Content Analysis in the Web 2.0 Workshop at WWW*.

.