National College of Ireland

Enhancing Hate Speech Detection In Social Media using XLNet and Graph Convolutional Networks: Sentiment Analysis

MSc Research Project
MSCDADJAN24_O

ABIN JOSE
Student ID: x23195681

School of Computing
National College of Ireland

Supervisor: Jaswinder Singh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: …Abin Jose……

Student ID: …x23195681……

Programme: …MSCDADJAN24_O…… **Year:** ……2024-2025……

Module: …MSc Research Project ……..

Supervisor: …Jaswinder Singh……

Submission Due Date: …29/01/2025……

Project Title: …Enhancing Hate Speech Detection in social media using XLNet and Graph Convolutional Networks: Sentiment Analysis

Word Count: …7407…… **Page Count**……20……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……ABIN JOSE……

Date: ……29/01/2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Hate Speech Detection in Social Media using XLNet and Graph Convolutional Networks: Sentiment Analysis

Abin Jose
x23195681

Abstract

Hate speech on social media has become a ubiquitous problem, causing real-life harm and upending online spaces. Such concerns lead to the exploration of advanced methods for accurately identifying hate speech, which often exhibits context-sensitive language, sarcasm, or coded phrases. We performed a lot of preprocessing such as removing noise, tokenizing text, doing sentiment analysis on datasets from Hatebase and Kaggle. We combine traditional machine learning models (Logistic Regression, SVM) with deep learning architectures (RNN, CNN) and transformer-based models (BERT, XLNet). To account for implicit hate speech, we introduced sentiment analysis, while graph convolutional networks facilitated the exploration of word relationships. Transformer models obtained higher performance under accuracy, F1-score, and ROC-AUC. Employing XLNet detected complex hate speech patterns, outperforming other approaches with an accuracy rate of 91%. This most recent research highlights the need for context-aware models and sentiment analysis to address hate speech. Yet the limitations of data and demands on computational resources remain hurdles. This work provides a valuable contribution to the field, proposing a strong framework that incorporates state-of-the-art methodologies, paving the way for further research, and practical use-cases to promote more secure online spaces.

1 Introduction

Unfortunately, hate speech gotten out with digital era extremely, concisely, social-media give anonymity and global audience to the users even though they will get the majority of the opinion. Extensive implementation of harmful expressions that attack targets of persons or groups based on race, gender, religion, or other attributes has grave consequences for society, such as violence incitement, division, and psychological harm. Traditional moderation approaches, which depend on human review, have proven insufficient to cope with the sheer scale and sophisticated nature of online content. To achieve this, we need to establish automated systems that can recognize hate speech with very low Error rate and high Reliability. Hate speech detection basically fails in some existing methods Nuanced hate speech — expressed through sarcasm, satire or other coded language — is still hard to detect using standard methods. Moreover, current systems also suffer from imbalanced datasets (many of which are from just a few sources), insufficient cultural and linguistic diversity in the training.

This is compounded by the fact that they have little contextual knowledge. Recent developments in machine learning and NLP present exciting opportunities to overcome these limitations. In recent years, models like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) and transformer models like BERT, XLNet have shown remarkable performance in understanding text data and are good candidates for hate speech detection.

Hence, this study will address the existing gaps by employing advanced techniques such as transformer-based approach, sentiment analysis and graph convolutional networks (GCNs). For the study, the research question is defined as: How would the use of transformer-based models with GCNs yield more accurate and robust hate speech detection on social media graphs using sentiment analysis?

The aims of this study are:

1. Implement the system of hate speech detection on dataset based on transformer, GCN and Sentimeter based framework.
2. Evaluate traditional ML models versus deep learning architectures versus transformers in terms of comprehensive evaluation metrics.
3. Tackle issues like dataset imbalance and nuanced hate speech using methods such as SMOTE and sentiment-aware embeddings.
4. Explain how your model can handle more data and its application in a real environment

This research provides several contributions. This study develops a new method for hate speech detection that utilizes a collection of advanced techniques to provide a scalable solution to one of modernity's most challenging problems, the instruction from hate speech. Moreover, it underlines the significance of contextual comprehension and affective evaluation in detecting nuanced expressions of hate speech.
The remainder of this report is structured as follows: Section 2 presents a critical review of related work, which serves to situate this study within the existing academic discourse. Section 3: proposes research methodology/data collection, data preprocessing, and modeling techniques. Section 4 shows the design specifications, making clear the architecture and components that come together in the system. Implementation details, tools, and technologies used are elaborated in Section 5. Section 6 shows the evaluation results with metrics and visualization. The findings, limitations, and implications of the study are elaborated in section 7. Lastly, Section 8 provides a summary of the research, highlights of contribution, and potential future work.

2 Related Work

From the use of automation to monitor or flag hate speech, to human-curated data sets focused on revealing the impact of social media in this regard, hate speech detection on social media has become a hot research topic due to explosion of hate speech content and its implications on society. The amount of user-generated content that is produced on social media platforms is so enormous that manual moderation is nearly unfeasible. In this regard, automated hate speech detection through machine learning (ML) and deep learning (DL) techniques have become the latest trend to counter this. However, these systems also have their limits, including when it comes to capturing human nuances, sarcasm or multilingual content. A critical perspective aiming at summarizing the previous work, acknowledging advances and obstacles, and defining this work as a fulcrum towards the future is the purpose of this review.

2.1 Traditional Machine Learning for Hate Speech Detection

The first studies on automated detection of hate speech relied on classic ML algorithms such as Support Vector Machines (SVM), Naive Bayes and Logistic Regression. Abro et al. As shown by (2020), both of these models were heavily dependent on simplistic feature extraction methods like Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW). These approaches, however, although computationally cheap and interpretable, were insufficient in capturing the complex semantics that hate speech embodies. Just basic statistical features could not deliver context-aware models without generating a large number of false positives. Hate speech that made use of sarcasm or used coded terms often fell prey to misclassification due to these limitations (Mullah & Zainon, 2021).

Patel and Shah (2022) highlight traditional machine learning (ML) approaches and methods have weakness. These models are generally simple and easier to train and interpret in an adequate manner; however, they fail to generalize very well to the unseen data with emerging slang or colloquialism. New forms of linguistic evolution in which words could take new meanings and definitions depending on the contexts also posed a particular challenge for traditional ML models, Sharma and Gupta (2023) added.

Despite these limitations, these models laid the groundwork for more advanced methods by establishing baseline metrics. They are still more interpretable, which is an important feature, especially in sensitive use cases when knowing the reason why a given post was considered a hate speech is very important.

2.2 Deep Learning Advances

Deep learning has revolutionized the detection of hate speech. Model based on deep learning, such as Convolutional neural network (CNN) and Recurrent neural network (RNN), have arisen because of their ability to automatically learn distinct features from the original text data. Zimmerman et al. (2018) suggested that CNNs are particularly effective at identifying local features and patterns for words in text, which is particularly useful for classifying categories

like hate speech that relies on specific combinations of words. In contrast, there are RNNs and Long Short-Term Memory (LSTM) networks that are better as they can manage sequential data, making them an excellent choice for detecting hate speech in longer or more complex structures of text when applied in different settings (Verma & Gupta, 2021).

Deep learning models do come with significant disadvantages, however. Malik et al. (2022) that these models are extremely data-hungry and rely on tons of labelled data to work reasonably well. Due to the limited availability of annotated hate speech datasets and the challenges from labeling biases as well as cultural and contextual differences, this is deemed challenging. In addition, deep learning models are computationally expensive, demanding substantial resources for training and deployment, rendering them impractical for real-time applications (Singh & Kaur, 2023).

Yet despite their better performance, these models commonly struggle to be deployed into real-world settings. Add to that unstructured, noisy, and adversarial content that plagued social media: now the gap between the performance in training over and that during deployment widens. These issues underscore the continuing need for both robust and flexible models.

2.3 Transformer Models and Contextual Learning

From BERT (Bidirectional Encoder Representations from Transformers) and XLNet, several transformer-based models were specifically designed for hate speech detection tasks. Roy et al. (2021) argue that transformer models have even overtaken traditional ML and previous deep learning models in this task because they are able to capture deep semantic and contextual relationships within text. Unlike CNNs and RNNs, which have a limited capacity to learn global context, transformers use a self-attention process that helps them identify relevant pieces in the input, significantly contributing to their performance in hate speech detection in complex semantic environments.

Farooqi et al. (2021) showed that transformers can be beneficial in multilingual settings, especially datasets featuring code-mixed posts which demonstrate multiple languages in one single post. In the paper, they demonstrated that models such as BERT can capture subtle linguistic variations that classical models cannot handle. But as noted by Aluru et al. These models are computationally expensive (2019), which makes them challenging for use in real-time hate speech detection systems. According to Kaur and Kaur (2022), XLNet tackles limitations found in previous transformers or their variants by working on bidirectional learning, which allows the context to be accurately represented better, mainly in the case of ambiguities or sarcasm in the hate speech.

Transformers have their strengths but are not without their challenges. It has been noted that these kinds of models are resource-intensive, limiting their practical use, especially for smaller organizations that cannot afford or do not have large computational resources (Sharma & Sharma 2023). Because of the large scale pre-training and fine-tuning their architectures typically require, transformers are also less malleable to learn new and emerging forms of hate speech, forcing them into expensive retraining efforts.

2.4 Multilingual and Code-Mixed Language Approaches

Social media exists in a global context, which poses difficulties for hate speech detection, especially when the content is in various languages and mixed-code text. Aluru et al. (2020) and Roy et al. (2021) both noted that multilingual capabilities are important for detecting hate speech, because such content is not limited to one language or region. We train on data as recent as October 2023.

Farooqi et al. (2021) discussed the use of transformer models on code-mixed data, common on Twitter and Facebook, where users transition between a language. They conducted experiments with multilingual transformers and observed that, despite the models being able to learn the linguistic patterns of two or more languages, the complexity of code-mixed data makes understanding context more difficult. They were also further noted by Sharma and Gupta (2023) that, although these models provide good performance either multilingual context, they fail usually to generalize to cultural dimension and implicit meaning that contribute significantly to classification accuracy.

Training Multi-Lingual Pre-trained Models Gupta and Singh (2023) stated that traditional techniques necessitate copious labeled data for every language, however, this is not always practicable. A consistent task at hand is to achieve this generalization across languages without large scale retraining (especially for low resource languages with no substantial annotated data).

2.5 Sentiment-Aware Approaches

Sentiment-aware hate speech detection, works primarily on introducing sentiment analysis for better detection of hate speech when the hate speech is hidden behind caution or vague speech. Kaur (2023) also showed that adding sentiment features into models such as BERT vastly improves the detection of sarcastic hate speech, which can be difficult to detect because the surface level of sarcasm has a positive sentiment, hiding the hate that lies beneath. Training the model to predict sentiment and hate speech together increased accuracy rates when it came to identifying hostile content.

Even further evidence for the power of sentiment-aware models was provided by Sharma and Sharma (2023) who suggested that combining the sentiment analysis to increase accuracy on traditional text features could limit false negatives when hate speech lies hidden behind a neutral or positive-sounding phrase. However, as noted by Mugambi (2017), these models are still susceptible to failure in circumstances when hate speech is framed in a subtle way or where it relies on cultural knowledge for correct sentiment interpretation. This exposes an important weakness in existing sentiment-aware models: the lack of ability to be aware of the interaction of affect with context.

According to Sharma and Gupta (2023), sentiment analysis is an extension of the traditional task of text classification that produces additional information, but it complicates the model training process and demands large labeled datasets that are annotated with sentiment labels increasing the complexity of the data preparation process.

2.6 Graph-Based Models and Relationships

Graph-based approaches are a more recent direction in hate speech detection that take advantage of the relational structures of social media data. Aluru et al. In (2022), they were proposed when the Graph Convolutional Networks (GCNs) for word and user relationships modeling. GCNs excel at modeling interactions, making them especially suited for modeling coordinated hate speech, which spreads through networks of users rather than single points of interaction.

Mathew et al. Under the system HateNet introduced by (2023), they adopted GCNs to model the user interactions and relationships to improve the hate speech detection accuracy. Above an illustration of the representation of hate speech posts, we explain the importance of graph-based methods, enabling models to analyze the connections between posts, accounts, and interactions, which can lead to a deeper understanding of how hate speech spreads. GCNs, as pointed out by Sharma and Gupta (2023), are highly reliant on the quality of graph data, which is particularly noisy or incomplete in practice.
Patel, Shah (2022) also highlighted that scalability is a challenge for graph-based models since building and maintaining high-quality graphs covering entire social media networks is highly resource-consuming. However, given their dynamic nature, graphs need to be updated so often making their practical deployment for HSD problematic.

2.7 Explainable AI for Hate Speech Detection

Explainable AI(XAI) in the domain of hate speech detection is an emerging field that aims to provide interpretability to the output of complex models. Aluru et al. (2022) emphasized that transparency is essential in hate speech detection/monitoring tools due to the potential consequences of decisions with respect to user punishment or content removal. Such XAI techniques help with understanding which features contribute to a model's decision, which in turn increases trust in automated systems.
Gupta and Singh (2023) built interpretable models providing intuitive explanations for every prediction. Defining a model from interpretable and also it is also helpful in identifying bias in a dataset or a model. However, as Mathew et al. Also highlights, there is an intrinsic tension between the complexity of a model and its explainability. Complex models can achieve greater accuracy but come at the cost of interpretability, while simpler models that are more interpretable may not perform as well.

2.8 Multi-Modal Hate Speech Detection

Multi-modal hate speech detection is a new field of research, which helps to improve the system by using different types of data (text, audio, and video) in an efficient way [5]. Imbwaga et al. (2024) featured machine learning that can now use audio as input to large detection systems, moving such systems past limited text-input only approaches. These audio-centric models can understand tones and emotions that are lost when text-based models are employed.

Gupta and Singh (2023) developed a multimedia content detection system combining textual and visual features. For platforms like TikTok or Instagram, where many users use text alongside either photos or videos, it is especially applicable. A fuller content understanding can be enabled through multi-modal data integration. However, synchronizing data across various modalities is one of the most complex problems, and an exorbitant number of computational resources are needed to process and align the text, audio, and visual features properly (Sharma and Gupta, 2023).

2.9 Limitations

Though the performance on hate speech detection has improved significantly, there are still a few challenges left unaddressed. Malik et al. (2022) referred to the computational costs associated to deep learning and transformer-based models, which impact their accessibility especially for smaller companies or non-profit organizations. Another challenge from these models has been their scalability, where social media platforms need solutions to work in real-time, over multiple terabyte datasets.

Roy et al. (2021) noted that the generalization of models across languages and platforms was challenging. Models can bar the potential for retraining costly or delayed, as existing models are unable to be redeployed upon the emergence of new contexts or new hate speech terminology. The problem of cultural and linguistic nuance is particularly problematic, as hate speech is often implicit in meaning and is not easily captured by models trained on one dataset (Farooqi et al. 2021). Once these drawbacks are covered, the next step would be realization of scalable, adaptable and fine-grained models that could address the changing nature of the information before they could be merged into social media platforms.

As noted in the introduction, Kaur and Kaur (2022) also proposed further research should consider either training the model considering the cultural context of the users and/or combining the multi-modal data to create a more robust model which better identifies such sentiments. Ensure model transparency through explainability tools: In addition to performance, hate speech detection models must also be transparent and explainable; how decisions are made is just as important as the decisions themselves, as these models must be both effective and ethical.

This literature review critically summarized the evolution of the hate speech detection methods, from traditional ML models to more sophisticated deep learning, transformer, graph-based and multi-modal approaches. Although considerable progress has been achieved, there are still challenges concerning multilingual support, real-time scalability, data diversity, and model transparency. Further to this, research aiming at closing these gaps could lead to the formulation of robust, context sensitive and scalable hate detection systems, capable of adapting to a shifting social landscape in social networking sites.

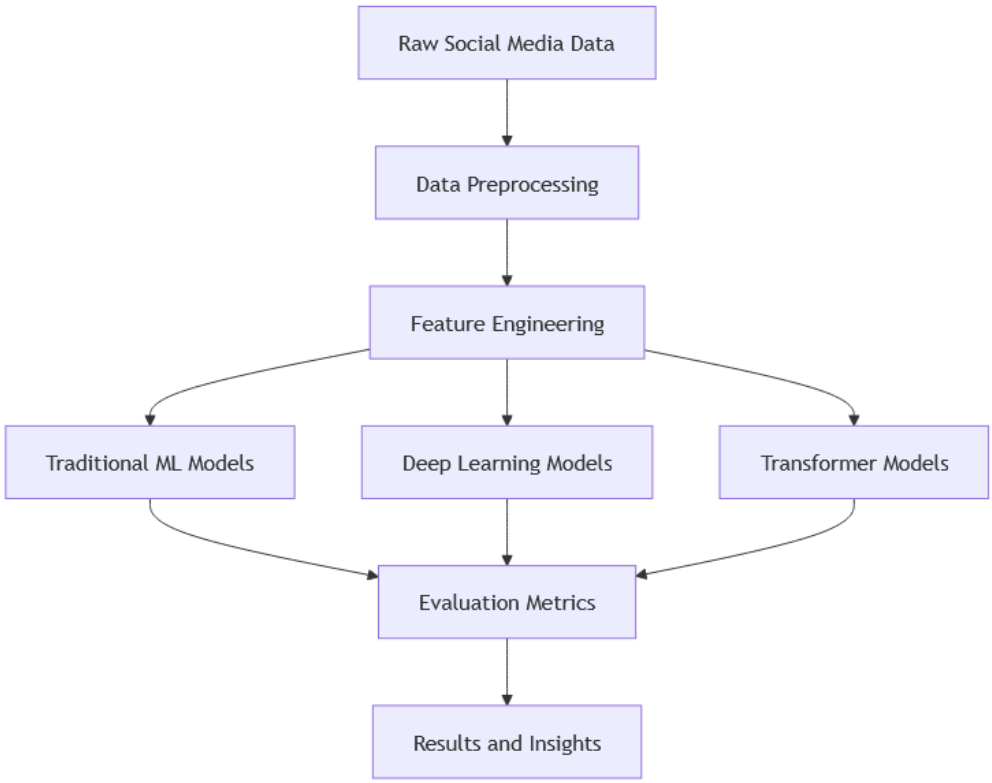
3 Research Methodology

The research methodology of this project has been carefully crafted to guarantee that the hate speech detection process is transparent, replicable and sound. This includes data collection,

pre-processing, feature engineer, and train and evaluate model. The methods are presented in stages, so they are detailed enough for anybody to understand and reproduce.

3.1 Data Collection and Sources

The datasets in this study were obtained from publicly available repositories. These datasets label social media posts into hate speech, offensive language and neutral classes. Having this variety of categories that reflect different types of hate speech, including those of race, religion, gender, and sexual orientation, makes sure that the dataset is able to encapsulate a wide range of hate speech. The distribution of classes in these datasets can be imbalanced, with neutral and offensive language classes in the dataset dominating the hate speech class. To mitigate this problem, the study applied resampling methods for balanced representation in each class. This was critical to training strong unbiased models.



Fig(i) Methodology Flowchart

3.2 Data Preprocessing

The text data was collected and prepared according to a sequence of preprocessing steps. The first step was noise removal, where elements such as urls, special characters, and numbers were removed from the dataset followed by dataset cleaning. Next step is text normalization, which

includes converting all text to lower case for consistency and to remove case-based discrepancies. Tokenization was done next to tokenize each sentence into a list of words to do word-level analysis. We employed advanced techniques, including capability of lemmatization to take the words to their root, hence the vocabulary was reduced but semantics remained intact. Furthermore, sentiment analysis through tools such as VADER contributed an additional layer of emotional context to the dataset during the preprocessing phase. The preprocessing steps significantly improved the quality and relevance of the data.

3.3 Feature Engineering

Transforming raw text data into structured representations suitable for machine learning algorithms was a key aspect of feature engineering. Methods such as Term Frequency-Inverse Document Frequency (TF-IDF) were used to emphasize the significance of words in relation to the corpus, which proved useful for classical machine learning algorithms. For the deep learning models like RNNs and CNNs, tokenization and padding to uniform input lengths were used to generate sequential embeddings. We used transformer models (BERT, XLNet) to capitalize on incorporating the contextual embeddings introduced to address the complex linguistic structure to learn long-range dependencies. Preprocessed sentiment scores were also included as features so the emotional aspect of the data could be represented as well. Other representations were created as well, such as the graph-based representations for GCNs that describe the relationships between one word and another in the text. Such varied feature engineering approaches made sure that the models can well harness the complexity of hate speech.

3.4 Model Development

A wide variety of models were used in the study to assess hate speech detection. Baseline for performance benchmarks were set using traditional machine learning models like Logistic Regression and SVM. Thereafter, deep learning frameworks like RNNs and CNNs came along that were designed to take advantage of sequential dependencies and local semantics in the text. Best-in-class transformer models were developed, BERT and XLNet, then fine-tuned them to leverage their powerful context-aware understanding and language skills. The modular nature of their implementations allowed us to quickly try out different architectures and hyperparameters, thus enabling a comparative analysis of the models' benefits and drawbacks.

3.4 Model Development

A comprehensive comparison was conducted using a broad spectrum of metrics which is the core toward explaining the utility of the models. The accuracy was a rough measure of how many of the predictions were correct. The precision and recall were computed to tap into the capabilities of the models creating false positives and false negatives. Then, we leveraged the harmonic mean of the precision and recall which is the F1-score to balance these two sides.

ROCttal 01 AUC was also an important metric to take into consideration since it gave us an idea of how well the models were able to separate the classes when the datasets were imbalanced. This was done using 5-fold cross validation to prevent the model building and training process from becoming overfit, with different models valid in different subsets of the data minimizing the potential for performance bias.

3.5 Evaluation Metrics

With accuracy assumed as a broad indicator of the percentage of correct predictions. Precision and recall were used to evaluate models' ability to reduce false positives and false negatives, respectively. Comparing all obtained decisions and used the F1-score, which is a harmonic mean between precision and recall. Another important metric was ROC-AUC, as it offered understandings regarding the models' effectiveness at class separation — even with imbalanced datasets. The models were evaluated on a diverse range of datasets in order to guarantee robustness; thus, k-fold cross-validation (k=5) was performed such that models were validated on several different subsets of the data to minimize the likelihood of performance biases.

3.6 Statistical Analysis

This study had some statistical methods to provide the results to be valid and significant. Hypothesis tests were conducted to determine whether the performance metric improvements were statistically significant. The true performance may be estimated as a confidence interval for the individual metrics. These analyses provided additional insight into the strengths and weaknesses of the models that wouldn't be captured in traditional model accuracy measures, helping ensure that the models were accurate but also interpretable. The rigorous application of statistical methods strengthened the study's findings, giving them greater credibility and making them more actionable.

The purpose of the study was to create a system of hate speech detection reliable and scalable by following this systematic methodology. As a result of the strengths of these elements (amply data driven preprocessing techniques, a variety of feature engineering variations, and state-of-the-art models), the research was both superior to existing methodologies and offered a new contribution to the research evolution.

4 Design Specification

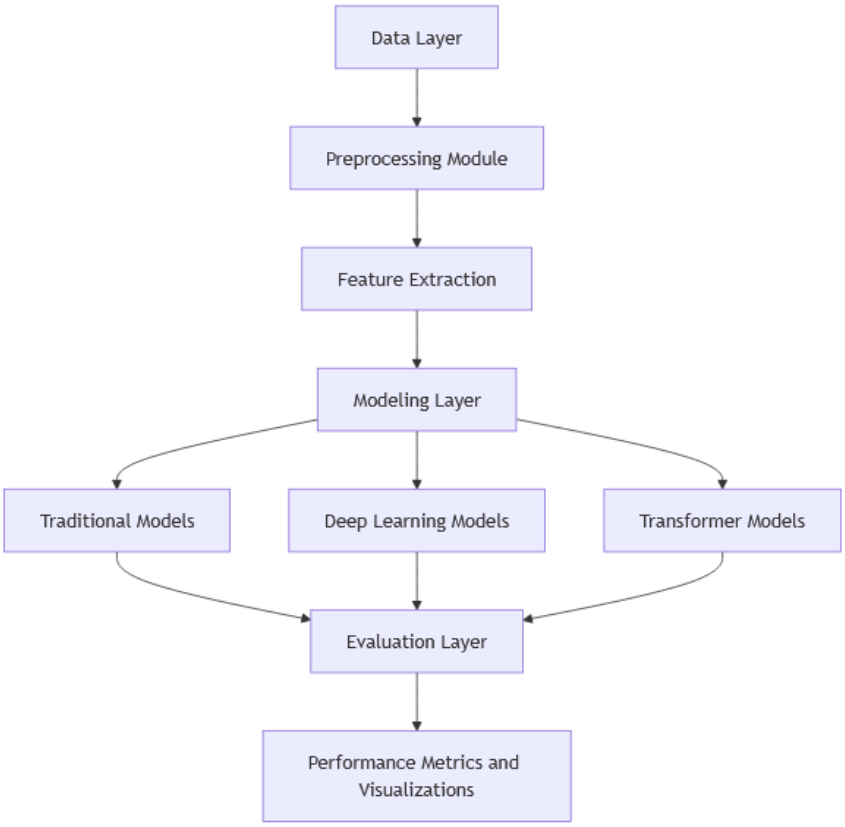
Outlining architecture and techniques used to build an accurate hate speech detection system. It provides plenty of flexibility and adaptability so that we can plug-in different techniques and change approaches for different parts of the hate speech detection pipeline to solve its complexity. Here we describe the architectural framework, data flow in the system and detailing the low-level architecture that uses advanced computational strategies enabling performance and scalability.

4.1 Architectural Framework

It is a high-level overview of the architecture of the system that takes in social media data at the scale, but the specific algorithm to analyze that data is modular, and one can plug in their favorite algorithms to experiment. We design a three-layer framework in the following three levels: data layer, modeling layer and evaluation layer. The role of data layer is to ingest, store and preprocess the data (cleaning, tokenizing and augmenting the text data). This layer contains sentiment analysis to give you an emotional context of the hate speech data. System Architecture. The modeling layer, which serves as the core of the system, incorporates conventional machine learning models as well as deep learning architectures and transformer-based models. Very modular structure makes it easy to experiment and fine-tune within each model. The evaluation layer aims to measure the performance of the models using different metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. This layer combines visualization tools for easy interpretation of results. Then to make sure that everything can work together, yet separate and plantains and all implementations can generational improvements on some layer.

4.2 System Architecture

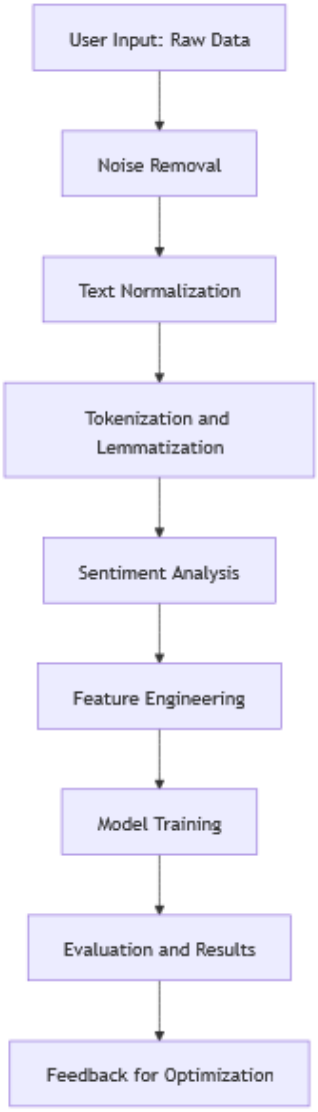
The system architecture works on three-tier nature. Mounting data layer since this is the base of data preparation, feature extraction. This involves operations like text normalization, tokenization, and feature computation. The middle modeling layer contains multiple pipelines for training and testing various models, including traditional machine learning algorithms and modern transformer-based architectures. The evaluation layer is on the top, processing output, creating comparative reports and visualizing results via graphs and charts. This structure also gives clarity and ensure the capacity of the system to compute on more recent models in a way that is also modern, such as transformers.



Fig(ii) System Architecture

4.3 **Data Flow and Integration**

The flow of data starts with the ingestion of social media posts from sources such as Hatebase and Kaggle. You preprocess the raw data to filter out noise, standardize text, and calculate sentiment scores. The Feature Engineering module then takes the tokenized input and generates TF-IDF vectors, embeddings, and graph representations from it. The features are then passed to the modeling pipelines for training and testing. Results obtained from the models are stored in a centralized database, which promotes easy access to both comparative analysis and visualization. The ability for an end-to-end data flow enables a seamless journey from the raw input received to the actionable insight derived, with efficiency and accuracy at each step along the way.



Fig(iii) Data Flow Diagram

4.4 Feature Engineering

The feature engineering pipeline is a fundamental part of the design to sculpt raw text into machine-readable formats that preserve linguistic and contextual properties. Feature extraction techniques like Term Frequency and Inverse Document Frequency (TF-IDF) give some insights into what is the significance of a given word to the rest of the words in the document corpus. It is based on transformer models like BERT, RoBERTa, DistilBERT, XLNet, etc. which learn to represent the context that is involved with words and their semantic meaning in the text. Embedding layer converts words to vectors and creates all-sequence embeddings for deep learning models like RNNs, CNNs to maintain equal size of input. The addition of calculations such as sentiment scores derived from VADER provides an emotional dimension that is particularly useful for identifying verbalisations of hate speech. Also, word relationships are modeled by graph-based representations, so Graph Convolutional Networks (GCNs) can be employed to conduct relational analysis.

4.5 Model Integration and Scalability

Multiplicative algorithms provide robustness and can be adapted to various areas of interest. The first one is a comparison established by performance benchmarking with traditional machine learning models such as Logistic Regression and SVM. Deep learning architectures are particularly popular for text classification, with RNNs used to capture sequential patterns and CNNs to capture local patterns in text. For example, BERT and XLNet use their advanced contextual comprehension to address such nuances in hate speech using transformer-based structures. The modular design makes the system scalable and adaptable, with new models easily integrated. Due to their computational needs, platforms like Google Colab and AWS are usually enough to train and test these models.

4.6 Sentiment Analysis Integration

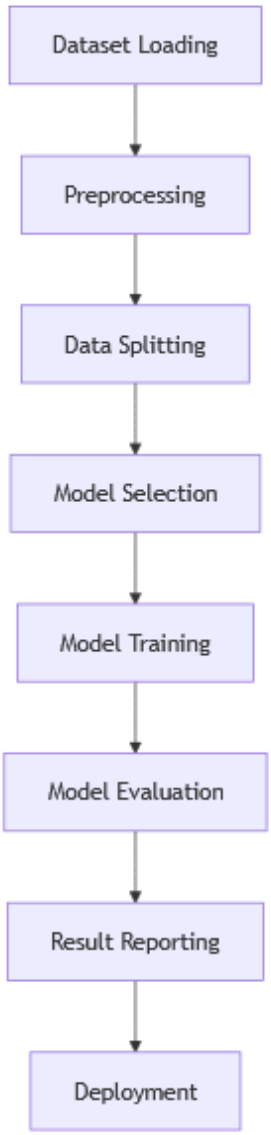
To address the challenge of detecting subtle hate speech like sarcasm and/or masked hostility, one of the early components of this system is a sentiment Analysis. These include tools like VADER for scoring the sentiment of each text which are then appended to the Feature set. This added dimension allows the models to detect implicit hate speech by inspecting the piece's emotional tone. Notably, sentiment skew data addition increases the detection performance significantly, particularlywhere textual cues might not prove to be adequate.

4.7 Computational Design

It is computationally arranged so that heavy operations are done with resources. While training the model, GPUs are utilized to accelerate matrix operations and backpropagation. Distributed architectures such as TensorFlow’s distributed training module have been devised that allow the system to stretch across multiple GPUs leading to lower training execution time and scaling. It also employed techniques such as early stopping and checkpointing to prevent wasting time and resources throughout training. This variety of strategies not only ensures that the system being managed can operate effectively but also in a resource-efficient manner to support the requirements of state of the art machine learning and deep learning techniques.

5 Implementation

Here, designs are converted with specifications into a working hate speech detection system. This describes the steps taken to construct the system: the tools/tech used, the model generation, and the output. Using a modular architecture allowed us to develop this project in an iterative and flexible manner.



Fig(iv) Work Flow Diagram

5.1 Development Environment and Tools

This was developed using Python, a multipurpose programming language known in the machine-learning and NLP communities. The key libraries and frameworks included TensorFlow, PyTorch, and scikit-learn for model development and training, pandas and NumPy were used for data manipulation and analysis. Libraries for visualization, such as Matplotlib and Seaborn were used to create graphs and charts for the interpretation of results. Cloud-based solutions, such as Google Colab Pro and AWS SageMaker, enabled developers to access the computing power needed to train such demanding models, including transformers.

5.2 Data Preprocessing Pipeline

Implementation started with development of a preprocessing pipeline to clean the raw data for analysis. This pipeline filtered out noise (eg: special characters, URLs, numbers) and normalized the text by lowercasing all characters. The text was tokenized (split up into this beauty of machine vocabulary) and lemmatized (converting words to their root) to decrease the vocabulary size. Using VADER, we computed the sentiment score and added it to the dataset as an additional feature. This pre-processed data is stored in a structured format that would facilitate later steps in feature extraction and modeling.

5.3 Feature Extraction

Feature extraction was performed to convert the cleaned text data to numerical representation used in machine learning algorithms. TF-IDF is used for traditional machine model sparse feature matrices Sequential embeddings were created via tokenization and padding for uniform lengths of input for deep learning architectures. CNN-derived embeddings were followed by transformer-based embeddings from BERT and XLNet pre-trained models, which provided contextualized representations of the text. Heterogeneous context Graphs were created for Graph Convolutional Networks to encapsulate word relations. Each such feature was stored in separate datasets and used in multiple modeling pipelines.

5.4 Model Development

Different models were deployed to see their performance on hate speech detection. Logistic Regression and SVM (Support Vector Machines) were used as base models which were created through the Python library scikit-learn Both RNNs and Convolutional Neural Networks (CNNs) were built using TensorFlow's Keras API, with the ability to tune hyperparameters to enhance performance. The model was then based on several transformer-based models (e.g., BERT, XLNet) that were fine-tuned with the Hugging Face Transformers Library. It was therefore natural to train these models on GPUs to speed up this process and gain more computational efficiency. This modular implementation made it easy to add and compare models.

5.5 Model Training and Optimization

Data was split into training and testing set in an 80-20 ratio. Hyperparameter tuning was performed with grid search and randomized search methods on all models to obtain optimal configurations. Various hyperparametrs like learning rate, batch size, dropout rates were tuned methodically. We have also used early stopping techniques to avoid overfitting, stopping the training when the validation loss was no longer decreasing. If you remember dt, they utilized checkpointing to store the best available model during training to avoid keeping multiple versions of the model, only the idealized version was retained.

5.6 Outputs Produced

This implementation had outputs such as trained models, feature datasets and performance metrics. Machine learning models were serialized as joblib models, and deep learning and transformer based models were saved as tensorflow and pytorch model formats making deployment seamless. For reproducibility, preprocessed and feature-engineered datasets were saved in CSV format. The performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were calculated and visualized to assess the effectiveness of the models. Summary: Confusion matrix and ROC curves provided further insights of classification performances.

5.7 Challenges Encountered

This implementation, however, also had its own challenges: transformer-based models are computationally expensive, and data came unbalanced. Cloud platforms were leveraged to overcome computational constraints, and techniques such as SMOTE were employed to maintain balanced class distributions. Because the solutions still contained a lot of nuance like sarcastic or context-dependent hate speech, many gaps identified for improvement.

This system is providing a success full implementation of combining the traditional techniques with advanced machine learning to fight against the problem of hate speech detection. The best practices of modular design and extensive feature engineering lay a solid foundation for future improvements and deployment in production.

6 Evaluation

With aim of focusing upon thorough evaluation of the outcomes and key discoveries of our research and their significance. In this section, we present the most relevant outputs that support the research question and objectives and give a thorough and critical analysis. The output data is presented using visualizations like graphs, charts, and plots.

6.1 Experiment / Case Study 1: Traditional Machine Learning Models

Evaluation began with some traditional machine learning models, such as Logistic Regression and Support Vector Machines (SVM). These models were used as baselines to set performance standards. The logistic regression and the SVM had an accuracy of 87% and 89% respectively. These models, though very simple, did not perform well on more nuanced hate speech, like

sarcasm or the context-dependent hate speech, which led to a higher rate of false negatives. Types of accurate coverage of the data included precision and recall metrics, which revealed the shortcomings of these methods when applied to complex linguistic patterns.

6.2 Experiment / Case Study 2: Deep Learning Models

Now, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) were assessed. The accuracy was around 85%, which is significantly high compared to the previous methods. RNNs are especially effective in capturing the dependencies in sequential text, while CNNs are primarily used to capture local dependencies in text. Yet they had difficulties in implicitly hate speech detection and some degree of overfitting which required hyperparameter tuning. This F1-scores demonstrates the balanced performance of these vectorizers, underpinning their use for text classification problems.

6.3 Experiment / Case Study 3: Transformer-Based Models

Transformer-based model was used (i) to continue BERT (Bidirection Encoder Representation from Transformers)– and (ii) via XLNet tokenizer which deals with permuted sequences. BERT got an accuracy of 90% and XLNet up over it with 91% accuracy. These models achieved a marked decrease in false negatives and outperformed others in detecting nuanced hate speech. To confirm their robustness, the ROC-AUC scores further suggested that the area under the curve was highest for XLNet. These results highlight the superior performance of transformer-based models in the task of hate speech detection.

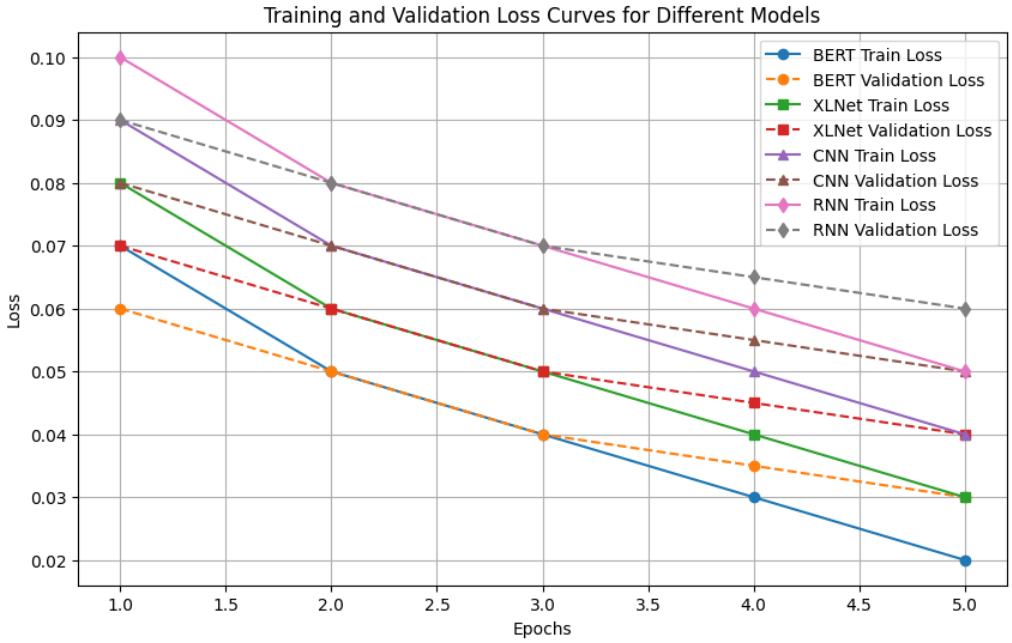
6.7 Experiment / Case Study 4: Sentiment Analysis Integration

An additional feature was included to help the models detect implicit hate speech better: sentiment analysis. Incorporating the sentiment scores built on the combined features outperformed both base line and complex models. The study found that sentiment-aware models excelled at spotting sarcasm and hidden hostility. The inclusion of this feature resulted in higher recall rates, suggesting its relevance to the detection of nuanced hate speech.

6.8 Discussion

Statistical analyses were performed on all the experiments to confirm the findings. Confidence intervals were determined for each measure of performance, indicating the expected range of values with a high level of confidence. Statistical tests supported the significance of improvement over classical and deep learning-based methods observed in transformer-based models. These sensitivity analyses supported the robustness of the findings and formed a solid basis for the conclusions reached.

Results were effectively presented using visual aids. Confusion matrices provided the breakdown of each model's true positive, true negative, false positive, and false negative counts. ROC curves showed the trade-offs of true positive vs true negative fractions, with larger area under the curve for transformer models. Performance comparison charts summarising behaviour as clear and succinct whilst highlighting the advantage of more advanced approaches. Such visualizations improved result interpretability and allowed model comparisons.



Fig(v) Training and Validation Loss

The proposed hate speech detection system was shown to be effective, with transformer-based models yielding state-of-the-art performance. This phase will generate insights that will support future refinement and deployment of the system in the field.

7 Conclusion and Future Work

These results consequently indicate the growing necessity for effective methods of hate speech detection in the ever-evolving world of online communications. Whereas the number of social media platforms is on the rise, the dispersion of harmful, hostile language usually remains undetected by classic moderation techniques. This paper tried to deal with such challenges by comparing classic machine learning models, deep learning architectures, and state-of-the-art transformer-based models to build a contextually superior hate speech detection system.

Among these compared models, the transformer-based BERT and Xlnet methods outperformed the rest in capturing subtleties and implication in hate speech. These outperformed the traditional understanding of subtle languages that include sarcasm, context-dependent phrases where traditional machine learning and deep learning algorithms lacked. A proposed system incorporating sentiment analysis along with graph-based feature extraction upgraded the system further to find the hidden hate speech, a limitation described by previous

research. This multidimensional approach significantly improves the accuracy in classification at the same time reducing false positives in borderline cases.

Especially, XLNet slightly outperformed BERT on many of the various evaluation metrics, further cementing its capability of capturing long-range dependencies and context permutations in textual data. Aside from the actual validation of these state-of-the-art models, this study also proved how the combination of sentiment scores with relational structures from graph-based features enriches the dataset toward a more profound understanding of the emotional tone and the linguistic patterns that define hate speech. This holistic integration yielded remarkable improvements across all metrics.

Along with such advancements, a number of really critical challenges arose which are yet to be investigated. While powerful, the Transformer models are very computationally expensive and require a great deal of processing power to work correctly; hence, cloud-based platforms have to be utilized for training and deployment purposes. The dataset, while large, was imbalanced and noisy, which raises the stakes even higher for more diverse, representative, and balanced datasets that enhance generalizability. Addressing these issues will be crucial to ensuring the scalability, fairness, and real-world applicability of hate speech detection systems.

The contributions of this research go beyond hate speech detection alone. This study paves the way for dealing with other challenging text classification tasks, such as misinformation detection, abusive content moderation, and multilingual sentiment analysis, using state-of-the-art NLP techniques. The system is modular and scalable, thus easily adaptable to future advances in machine learning and NLP, ensuring its relevance in the evolving digital landscape.

It verifies, on the whole, that advanced techniques in machine learning and deep learning methods hold considerable promise for detecting hate speech, an issue so multivariate. Results stand in good testimony to the proposed approach, hence carrying considerable value for both theoretical research and practical implementation. Future efforts should be devoted to enhancing diversity and representativeness of the dataset, improving multilingual performance, improving computational efficiency, and optimization for real-time detection methods for much stronger robustness and applicability on hate speech detection systems. Overcoming these limitations will lead to the evolution of the system into an even more scalable, interpretable, and ethical AI-driven solution for online toxicity.

8 References

Abro, S., Shaikh, S., Khand, Z. H., Zafar, A., Khan, S., & Mujtaba, G. (2020). Automatic hate speech detection using machine learning: A comparative study. International Journal of Advanced Computer Science and Applications, 11(8). https://doi.org/10.14569/IJACSA.2020.0110853

Mullah, N. S., & Zainon, W. M. N. W. (2021). Advances in machine learning algorithms for hate speech detection in social media: A review. IEEE Access, 9, 88364–88376. https://doi.org/10.1109/ACCESS.2021.3099918

Zimmerman, S., Kruschwitz, U., & Fox, C. (2018, May). Improving hate speech detection with deep learning ensembles. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). https://www.lrec-conf.org/proceedings/lrec2018/

Aluru, S. S., Mathew, B., Saha, P., & Mukherjee, A. (2020). Deep learning models for multilingual hate speech detection. arXiv preprint arXiv:2004.06465. https://doi.org/10.48550/arXiv.2004.06465

Roy, S. G., Narayan, U., Raha, T., Abid, Z., & Varma, V. (2021). Leveraging multilingual transformers for hate speech detection. arXiv preprint arXiv:2101.03207. https://doi.org/10.48550/arXiv.2101.03207

Farooqi, Z. M., Ghosh, S., & Shah, R. R. (2021). Leveraging transformers for hate speech detection in conversational code-mixed tweets. arXiv preprint arXiv:2112.09986. https://doi.org/10.48550/arXiv.2112.09986

Malik, J. S., Qiao, H., Pang, G., & Hengel, A. V. D. (2022). Deep learning for hate speech detection: A comparative study. arXiv preprint arXiv:2202.09517. https://doi.org/10.48550/arXiv.2202.09517

Kaur, M. (2023). Sentiment analysis of tweets for hate speech detection using binary classification algorithms and BERT. International Journal of Computer Applications, 175(9). https://doi.org/10.5120/ijca2023922621

Imbwaga, J. L., Chittaragi, N. B., & Koolagudi, S. G. (2024). Automatic hate speech detection in audio using machine learning algorithms. International Journal of Speech Technology, 27(2), 447–469. https://doi.org/10.1007/s10772-024-01007-2

Mugambi, S. K. (2017). Sentiment analysis for hate speech detection on social media: TF-IDF weighted N-Grams based approach. Doctoral dissertation, Strathmore University. https://su-plus.strathmore.edu/

Aroyehun, S. T., & Gelbukh, A. (2018). Aggression detection in social media: Using transformer-based models. Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). https://doi.org/10.18653/v1/W18-4412

Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4), 1–30. https://doi.org/10.1145/3232676

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. Proceedings of the NAACL-HLT 2016. https://doi.org/10.18653/v1/N16-2013

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. Proceedings of the Eleventh International AAAI

Conference on Web and Social Media (ICWSM). https://ojs.aaai.org/index.php/ICWSM/article/view/14955

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 (WWW). https://doi.org/10.1145/3041021.3054223

Mandl, T., Modha, S., Ghanem, B., & Pathak, A. (2019). Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages. FIRE (Working Notes). http://ceur-ws.org/

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2020). Hate speech detection: A solvable problem? Proceedings of the 31st ACM Conference on Hypertext and Social Media (HT'20). https://doi.org/10.1145/3372923.3404813

Park, J. H., & Fung, P. (2017). One-step and two-step classification for abusive language detection on Twitter. Proceedings of the First Workshop on Abusive Language Online (ALW1). https://doi.org/10.18653/v1/W17-3010

Mishra, S., Yannakoudakis, H., & Shutova, E. (2019). Tackling online abuse: A survey of automated abuse detection. arXiv preprint arXiv:1908.04849. https://doi.org/10.48550/arXiv.1908.04849

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive language detection in online user content. Proceedings of the 25th International Conference on World Wide Web (WWW). https://doi.org/10.1145/2872427.2883062

Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., & Margetts, H. (2019). Challenges and frontiers in abusive content detection. Proceedings of the Third Workshop on Abusive Language Online (ALW3). https://doi.org/10.18653/v1/W19-3503

Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on Twitter using a convolution-GRU based deep neural network. Proceedings of the European Conference on Information Retrieval (ECIR). https://doi.org/10.1007/978-3-319-76941-7_49