Comparative analysis of the effectiveness of NLP algorithms in unveiling criminal intentions in tweets

Nisha P. Shetty*, Adriteyo Das*, Naman Joshi*, and Gautham Manuru Prabhu†

Abstract

Purpose

The aim of this project is to address the issue of identifying criminal intentions, which have significant real-world impact of societal-division. The paper focuses on implementing advances in Natural Language Processing (NLP) to achieve the same.

Design/Methodology/Approach

Unlike the prevalent use of machine and deep learning architectures for detecting hate content, this study investigates and analyzes a feature-engineered NLP model in conjunction with three advanced transformer models: DistilBERT, RoBERTa, and ELECTRA, as well as a fine-tuned variant of DistilBERT.

Findings

The proposed model demonstrated superior performance compared to all existing architectures, surpassing base NLP models with a significant accuracy improvement of approximately 15%. Additionally, the study highlights enhanced text classification results achieved by fine-tuning the classifier layers or improving learning through pretraining with Hugging Face transformers.

Originality

The originality in this research lies in its comprehensive approach to feature engineering, combining advanced embedding techniques, multi-scale context extraction, and sophisticated feature selection and weighting methods. **Research Limitations/Implications**

The study acknowledges potential limitations in the scope of language and cultural contexts covered by the models. Future research could expand to more diverse datasets to universalize the applicability of the findings.

Social Implications

Current work through detection of harmful intentions with architectures

*Department of Information Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, 576104, Karnataka, India

†Department of Computer Science Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, 576104, Karnataka, India

capable of handling vast data contributes to creating a safer, more inclusive environment. This has significant impact on reducing real-world consequences of online hate.

Keywords— Natural Language Processing, Transformers, Lexical Semantics

1 Introduction

In today's time, social networks serve as prevalent platforms for information dispersion and social engagement. This has also led to the same acting as a medium for communications that signal criminal intentions. Analyzing text data from these platforms to identify potential threats or criminal actions is vital to protecting digital and physical environments. The paper aims to explore, assess and improvise methodologies in natural language processing(NLP) to navigate these networks and help identify criminal intentions. Research in this area can be valuable, particularly for law enforcement officials and civil society members, to help prevent and detect crimes. Papcunov et al. (Papcunová et al. (2023)) proposed a definition of hate speech utilizing 10 hate speech indicator parameters, the primary ones being a denial of human rights and promotion of violent behavior. They define hate speech as "as any text that promotes violent behavior, denies human rights, contains slurs, vulgarisms, or ad hominem attacks, uses negative stereotypes, or purposefully manipulates the truth or historical facts." Hate crime refers to criminal acts motivated by prejudice or bias against a particular group based on race, religion, ethnicity, sexual orientation, or gender identity. Social media platforms have witnessed instances where individuals or groups openly express hatred towards specific communities, incite violence, or organize discriminatory actions. Twitter, being a widely used platform for public discourse, has seen several notable cases of discourse which in turn has led to hate crimes.

1.1 Role of Natural Language Processing (NLP)

NLP is a powerful form of learning that enables computers to understand and interpret human language, especially when it comes to analyzing vast volumes of textual data on social media. With the help of NLP techniques, researchers and law enforcement can efficiently analyze large-scale data and uncover patterns and insights indicative of criminal intentions. NLP goes beyond the surface level of text and delves deep into linguistic aspects such as identifying sentiments, language patterns, and semantic relationships. This capability allows it to capture the broader context of words or phrases, which is crucial for detecting concealed criminal intents (Khurana et al. (2023) Alqahtani et al. (2023)).

1.2 Existing Landscape and Rationale behind Model selection

Earliest approaches used for detecting hate in text relied on dictionaries or some relevant knowledge bases which didn't offer good precision when the keyword was not identified. Additionally,without the knowledge of context just flagging the text as hate speech due to presence of keyword caused false alarms. This detection was further augmented by gathering information about the user and such as demographics and statistics which led to potential bias in the detection system. A pleothra of machine and deep learning classifiers have been used in earlier literature for hate detection or for detecting emotional contents in texts.However, there is a dearth of studies focused on specific emotions of hate crimes like abuse or aggressiveness detection as proposed in our study. Models like Bag of words, Term Frequency-Inverse Document Frequency (TF-IDF) etc. also results in high false positives as they flag the text based on presence of certain keywords without differentiating between different uses of the same word in varying contexts. The proposed transformer based models emphasize more on textual based features and is proven to outperform traditional Machine learning(ML) or Deep Learning classifiers(DL) in terms of contextual understanding which is vital for hate detection.The attention mechanism of transformers aid in focussing on relevant parts of the input text, enhancing their ability to identify crucial elements that indicate hate speech. These models excel at understanding complex linguistic structures and can recognize patterns in the data that simpler models miss. They can detect hate speech even when it is not explicitly stated but implied through context and tone (Alrehili (2019) Kovács et al. (2021) Nandi et al. (2024) Jahan & Oussalah (2023)).

1.3 Problem Statement

The research problem at hand is how to effectively detect criminal intentions on social networks, with a specific focus on utilizing NLP techniques. The aim is to develop computational methods that can analyze and interpret textual data from social media platforms to identify potential criminal activities, such as hate speech, abuse and aggression. Particulary, the proposed research aims to answer the question "Which among the investigated transformer based architectures fare the best in detection of criminal intentions in social media text?"

This paper is structured as follows: Section 2 comprises the literature review, the methodology is entailed in Section 3, and the result analysis is carried out in Section 4. Discussion is provided in Section 5, Conclusions and Future work are done in sections 6 and 7, respectively.

2 Literature Review

There have been a lot of studies on how to recognize harmful language like hate speech, cyberbullying, aggressiveness, and toxic comments as a result of

the increase of hate speech on social media platforms. Researchers Han et al. (Han et al. (2024))investigated the effectiveness of various machine learning approaches in detecting online trolling behavior. Their findings revealed that ensemble methods, such as Random Forest, showed promising performance in classifying these types of posts.Their research produced identical results for Logistic Regression, Support Vector Machines, and Naive Bayes, as evidenced by similar confusion matrices, highlighting the significance of model interpretability. The researchers also suggested that future research should explore the potential of deep learning and transformer-based models to further advance this field of study.

Sharif and Hoque's (Sharif & Hoque (2022)) innovative approach combined multiple transformer models to identify aggressive content in Bengali texts. They classified hate speech into four distinct categories: religious, political, verbal, and gender-based. Their research utilized various preprocessing methods, including TF-IDF, word embeddings, and FastText, to extract meaningful features. The unique weighting strategy they developed outperformed several individual machine learning and deep learning classifiers. Their error analysis revealed that the models struggled to detect implicit aggression and the presence of ambiguous words in the text hindered the ability to accurately identify the content.

Baruah et al.(Baruah et al. (2024)) modelled an architecture for detecting abusive language in Khasi texts. They built a collection of Khasi language data and used it to create specialized word representations using word2vec and fastText. Their study explored multiple modeling techniques, encompassing deep learning, conventional machine learning, and ensemble approaches. They experimented with different vector representations, including word2vec, fastText, and topic vectors derived from Latent Dirichlet Allocation (LDA). Moreover, they explored whether language models such as LaBSE and LASER, which can generalize knowledge across languages, could effectively detect abusive content in Khasi texts. The research uncovered a crucial discovery - employing feature selection methods and equalizing the dataset resulted in improved performance across various machine learning classification models.

Rehman et al.(Zia Ur Rehman et al. (2023)) combined insights from social and textual elements to enhance the detection of harmful content. They investigated the social context surrounding posts, considering factors like the number of likes and reports, a post's tendency to attract abusive comments, and the user's propensity to make such comments. This comprehensive analysis provided a deeper understanding of the dynamic interactions between users and posts. To further refine the text-based assessment, the authors employed state-of-the-art language models, including Muril, XLM-R, and M-Bert, to extract contextual information from the posts. By fusing these individual features, they created a robust joint feature set that was then fed into an ensemble classifier to determine whether a post was abusive or not. Despite these advancements, the proposed method encountered limitations. It struggled to recognize abusive comments when the spelling of a non-abusive word matched that of an abusive word, and it lacked the scalability to handle multilingual data where certain words may

have different connotations across languages. These challenges highlight the complexities involved in accurately capturing the nuanced context of language, a critical factor in effectively addressing abusive content.

Song et al.'s(Song et al. (2022)) study utilized both the content of posts and the user's interaction network to categorize abusive language on social media platforms. Their method employed a fine-tuned BERT model for contextual classification, which they enhanced with graph-based learning to capture the interactions and semantic similarities between posts. This combined approach allowed their model to learn from the emotions, attitudes, and other factors that shape abusive language within the network, and how they influence each other. While their method demonstrated effectiveness in networks with higher levels of interaction, the authors suggest future exploration of detecting abusive language in the absence of labeled datasets.

Khan et al.(Khan et al. (2022)) recognized the shortcomings of natural language processing techniques in detecting aggression in social media posts, particularly in capturing emotional cues. To address this, they incorporated emotional features alongside text-based features to enhance the ability to identify aggressive content. They employed a novel feature selection method, mutual information with correlation coefficient, to uncover the relationships between the different characteristics. Their experiments showed that a compact deep neural network(DNN) model outperformed other classification approaches when using this combined feature set. The authors aim to further their work by expanding the detection capabilities to diverse multilingual datasets.

MacAvaney et al.(MacAvaney et al. (n.d.)) proposed a new classifier called the multi-view SVM approach. This method achieved near better perfomance and interpretability compared to neural networks. In their study, the authors defined hate in various contexts and platforms, and they listed definitions of hate provided by notable figures. They also analyzed prominent datasets used by various researchers in this field. Their classifier combined two types of SVMs: one focused on feature extraction (view classifier) and the other dedicated to predicting the probability of which class the input should be assigned to based on the extracted features (meta classifier). With each view classifier focused on the impact of a single feature rather than a feature set on the classification result, the proposed method proved to be highly interpretable in identifying the key features that contribute the most to the hate speech label. However, the authors acknowledged the shortcomings of their approach in recognizing hate in the absence of explicit mentions of the user's intent and context. They also noted the challenges in recognizing hate emotions conveyed through images.

Bacha et al.(Bacha et al. (2023)) evaluated the effectiveness of 3 computer vision models - YOLOv4, YOLOv5, and SSD MobileNet V2 - in identifying offensive text within memes. Their pre-processing stage involved addressing image backgrounds, normalizing text orientation and size, color backgrounds, and blurriness to enhance readability. Additionally, they filtered out duplicate images and memes containing only text. However, their model had limitations, such as poor performance when the meme contained lesser text. The proposed classification was also binary and could be expanded to detect other aspects of

hate.

Arya et al.(Arya et al. (2024)) utilized Open AI's versatile Contrastive Language-Image Pre-Training (CLIP) model, which combined text and image data to detect hatred in a Facebook meme dataset. This study framed the categorization of multimodal hostile memes as a classification task, where the model predicted whether a multimodal meme was hateful or non-hateful based on the accompanying image and text. The authors suggest exploring more sophisticated machine learning techniques as a next step to better manage these multimodal data.

Qureshi and Sabih (Qureshi & Sabih (2021)) in their study explored different forms of text mining algorithms to appropriately categorize various forms hate based on extracted feature sets. They evaluated their data sets using tSNE plots and employed Latent semantic analysis(LSA) to deal with high dimensional datasets. Their study concluded with identification of issues like the need for appropriate discriminating features, complex data overlaps, and non-linearity. The authors leave the limitations of addressing model interpretability in complex and overlapping hate categories as a future scope. They advocate the use of transformers for the same.

Asif et al. (Asif et al. (2024)) modelled a Graph convolutional neural network that used cosine similarity to form connections between the words. Augmented with detection through GloVe embeddings their model aims to identify troll behavior in social media. Albeit having good performance the proposed approach suffered from sublime precision and recall rates which can be mitigated through enhacing contextual understanding of the model. The authors further elaborate the need to address model bias and improve model explainabilty as a future scope in this research.

Hashmi and Yayilgan (Hashmi & Yayilgan (2024)) created a stacked classifier with Bi-LSTM and GRU with Fast embeddings and regularization to detect multi-class hate in Norwegian language. The employed fast text embeddings excelled when handling languages with intricate grammatical structures and diverse variations. The results were further augmented with application prompt-based fine-tuning, including both few-shot and full fine-tuning, also LIME was employed to make the model more interpretable. For future investigations, the researchers propose exploring advanced multilingual transformer models, such as mT5 and GPT, particularly for languages with limited resources.

Jahan and Oussalah (Jahan & Oussalah (2023)) conducted a systematic and structured review using the PRISMA protocol. The authors critically assessed the current state-of-the-art techniques, datasets, and performance metrics. Munthuli et al.(Munthuli et al. (2023)) explored the application of transformer-based models, specifically XLM-RoBERTaBASE, for classifying intents in Thai Supreme Court decisions related to sexual violence laws. The model was found to handle multi-intent classification, albeit with misclassifications in overlapping intents. Kumar et al. (Kumar et al. (2020)) demonstrated the effectiveness of using a fine-tuned BERT model to detect hate speech and offensive content on social media. Authors Fortuna et al. (Fortuna et al. (2022)), however, explained the difficulties in using NLP to handle hate speech. The paper puts forth the

challenge in modeling the same as a classification problem. It further analyses how hate speech is a complicated, context-dependent problem that NLP techniques are ill-suited to handle. In summary, these research papers contribute to different aspects of tweet classification and analysis. They focus on topics such as hate crime identification, detection of harmful language, sentiment analysis, crime-related tweet classification, and the development of frameworks for investigating suspicious posts on social media. Transformer models, including BERT and ELECTRA, are utilized in some of these studies to enhance the classification and analysis of tweets, allowing for more accurate and nuanced results. The survey papers researched are briefed in the Table 1.

2.1 Gap in work

The study of previous research papers highlights a significant gap in the existing literature on tweet classification, specifically focusing on the under utilization of transformer models, such as BERT and ELECTRA. The detection rate of subtle hate speech in sentences has been considerably influenced by complicated morphological structures, ambiguous words, various accents, and a wide range of phrase components. This identifies as a crucial research gap. Transformer models, with their extensive contextual awareness and capacity to capture intricate speech patterns, are ideally suited to meet these issues. Their capacity to analyse complete phrases bidirectionally enables them to better manage the complexities of language, setting them as a viable tool for detecting subtle hate speech.

3 Methodology

The classification of tweets into categories such as clean, abusive, and hate crime using transformers and aggressiveness index classification involves leveraging advanced techniques to analyze and categorize tweets automatically based on their content and intent. Figure 1 shows the outline of the study.

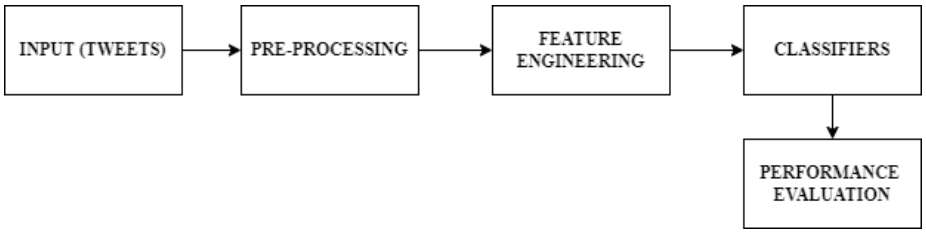

Figure 1: Framework of the study

Table 1: Comparision with existing architectures

Paper	Data Type	Data Set Used	Models	Results	Classes
Bacha et al. (2023)	Memes	KAU-Memes dataset	YOLOv4, YOLOv5, and SSD MobileNetV2	YOLOv5: mAP: 91.40 %	Offensive Non Offensive
Arya et al. (2024)	multimodal meme with assosciated text and images	CIFAR	CLIP model with cosine similarity and prompt engineering	Proposed Model: Accuracy: 87.42%	Hateful vs Non-hateful
Qureshi & Sabih (2021)	Tweets	HatebaseTwitter, HatEval , WassemA WaseemB ,MLMA	Text mining models with LSA and tSNE tested on ML models	CatBoost: Accuracy: 89.03%	Ethinic, Religion, Racism Gender, Sexism, Refugees
Khan et al. (2022)	Tweets and Emotions	Cyber-Troll Dataset	ML and DL models	Proposed DNN: Accuracy: 96.07%	Aggressive Non Aggressive
Baruah et al. (2024)	Text	Khasi Abusive Language Dataset (KALD)	ML, DL and Multilingual models	XGBoost: Accuracy: 94.42%	Abusive Non Abusive
MacAvaney et al. (n.d.)	Text	Stormfront , TRAC , HatEval, and HatebaseTwitter	multi-view SVM approach	Proposed Model: Accuracy: 80.03%	Hate No Hate
Asif et al. (2024)	Text and metadata	Dataset for Detection of Cyber-Trolls	Graph Convolutional Network (GCN) with Cosine Similarity	Proposed Model: Accuracy:92%	Trolls and toxic content
Song et al. (2022)	Text and Interaction data	HatEval, Davids,FNUC,	Combination of BERT and Relationship special Graph neural network	Proposed Model: Accuracy 70-77%	Abusive Non Abusive
Zia Ur Rehman et al. (2023)	Social Textual	SCIDN and MACI	Multimodal Architecture	Proposed Model: Accuracy: 90.12 % and 95.40 %	Abusive Non Abusive
Sharif & Hoque (2022)	Text	Bengali Aggressive Text dataset	Weighted Ensemble	Proposed Model: Weighted F1 Score: 93.43%	Aggressive Non Aggressive
Han et al. (2024)	Text	Cyber-Troll dataset	ML models	Random Forest Accuracy: 94%	Aggressive Non Aggressive
Hashmi & Yayilgan (2024)	Text	Detecting and grading hateful messages in the Norwegian language	Fast RNN, Transformers DL architectures	Proposed Model: F1 Score: 91-97%	Multiclass Hate
Proposed	Text	Cyber-Troll and HSOL	BERT based Transformers	Proposed Model Accuracy:80%	Abuse Aggressiveness Non Aggressive Hate Neither

3.1 Pre-processing

To ensure accurate tweet classification, several pre-processing steps are necessary. Firstly noise-inducing elements such as URLs,hash tags,and mentions are removed, as these make the analysis difficult. Then punctuation marks are removed to focus solely on the text. Subsequently, the text is standardized into lowercase. Common spelling corrections and normalization of various Unicode patterns are performed. Contractions in the text like 'aren't' is expanded to 'are not' to maintain uniformity. For tweets that contain multiple sentences or clauses, proper segmentation is ensured to maintain contextual meaning. Common words like domain specific words which do not add much to the analysis are subsequently removed.If multilingual data is present, such words were detected and filtered to focus on the target English language. Patterns like dates are analysed to gather more contextual information from them.Techniques like stemming and lemmatization which simplify the words are incorporated. Finally tokenization is performed to make the words easier for the transformer models to predict. The NLP libraries are used for this process. Additionally, Named Entity Recognition (NER) is employed to identify and possibly replace named entities with generic tags. The text length to is adjusted fit model input requirements by padding shorter texts and truncating longer ones.Furthermore, emojis are interpreted suitably using python packages like emoji and twitter slangs are normalized using preexisting APIs and dictionaries. Word embeddings like GloVE are applied for synonym replacements and NLP libraries like TextAugment, and nlpaug augmented the training sets. The augmentation process involves modifications at the character, word, and sentence levels to enhance data diversity. Character-level augmentation employs keyboard distance to simulate realistic typing errors. At the word level, techniques include back-translation, where text is translated to another language and back to create varied expressions; random insertion or deletion of words to diversify the text; word splitting, which breaks words into smaller parts; and thesaurus-based synonym replacement to generate different versions of the text. Sentence-level augmentation involves next-sentence prediction to enrich the context and abstractive text summarization to create concise summaries, providing alternative expressions of the content.Figure 2 summarizes the process.

3.2 Feature extraction

In the general feature extraction approach used by transformers, tokens obtained in the tokenization step are converted into embedding vectors, capturing their semantic meanings. Positional encodings are added to maintain the order of tokens. Multi-headed self-attention is then applied to these embeddings, enabling each token to interact with others and gather contextual information. Following self-attention, residual connections add the input of each layer to its output, and layer normalization stabilizes activations. Each attention layer is further refined by feed-forward neural networks. For downstream tasks, the text is processed through the transformer model to generate embeddings, which are

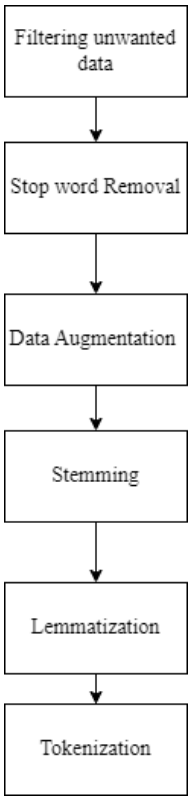

Figure 2: Steps involved in preprocessing

then aggregated into a fixed-size vector through pooling. This vector is mapped to a simpler space using a linear layer, and a softmax function transforms the output into probabilities, leading to the final prediction.

3.3 Transformers

Transformers, such as the BERT model (Devlin et al. (2018)), have become a dominant architecture in NLP, outpacing traditional methods like LSTM and neural networks. This is because they possess several key features. Firstly, their attention mechanism expertly captures long-range dependencies, allowing a better understanding of the global context that LSTM's sequential focus cannot provide. Secondly, their design allows for efficient parallel processing, which speeds up training and inference. Additionally, they produce contextualized word representations, which provide a comprehensive understanding of word relationships within a given context, an improvement over older models' static embeddings. Another significant advantage is that they can be pre-trained on vast corpora using unsupervised objectives, which not only grants them a

rich language comprehension but also permits further fine-tuning on smaller, task-specific datasets, often resulting in better performance with fewer labeled samples. Lastly, their intrinsic handling of variable-length sequences, using positional encodings.

3.3.1 DistilBERT

DistilBERT (Sanh et al. (2019)), a variant of the BERT architecture, has gained popularity due to its streamlined nature. It achieves model compactness through knowledge distillation, where a smaller model is trained to emulate the performance of the more expansive BERT. This results in a reduction in parameters, facilitating expedited inference and making it suitable for resource-limited settings. Despite its smaller size, DistilBERT doesn't compromise on performance. This is due to its preserved essence from BERT, which enables its effective application in transfer learning. DistilBERT's efficiency extends beyond inference to training as well, making it a time and cost-effective option, especially on large datasets. Its performance-competitive nature and resource-efficient characteristics make it a promising alternative to BERT in the NLP domain.

3.3.2 ELECTRA

The traditional pretraining approach of BERT involves using masked language modeling (MLM). In contrast, ELECTRA (Clark et al. (2020)) employs a generative-discriminative setup, where a small MLM is used to corrupt the input text. The corrupted text is then passed through the model, which is tasked with identifying the replaced tokens. This approach enables the model to learn from errors and patterns in the input text, resulting in enhanced contextual comprehension and superior token representations. The effectiveness of ELECTRA's method has been demonstrated by its superior performance in various downstream NLP tasks. The model's ability to discern real from replaced tokens during training has improved performance in tasks such as question answering, sentiment analysis, and named entity recognition. Additionally, decoupling the embedding size from the hidden size enhances the computational efficiency of the model, thereby reducing the time and resources required for training. Furthermore, the compatibility of ELECTRA with different NLP components and applications ensures its versatility and applicability across various domains. This compatibility also enables the model to be easily integrated into existing NLP pipelines without requiring significant modifications or reconfigurations.

3.3.3 RoBERTa

RoBERTa, a robustly optimized variant of the BERT model, has gained immense popularity in the NLP community due to its superior performance in text classification tasks. The model has been known for its enhanced training methodology encompassing larger batch sizes, prolonged training schedules, and more extensive training data, which has helped it achieve impressive results

in various NLP tasks (Liu et al. (2019)). Furthermore, the model's ability to fine-tune on specific datasets leverages its proficiency in text classification, making it highly adaptable to various scenarios. However, despite its advantages, including the incorporation of pretrained language model weights, expansive training on voluminous text data, and resultant robustness and generalizability, RoBERTa has its challenges. The model's considerable computational demands, prolonged training durations, and inherent interpretability issues might impede its applicability in environments with limited computational resources or where model transparency is paramount. Therefore, while it is undoubtedly a significant advancement in NLP, researchers must consider these limitations while designing their experiments and applications. Table 2 highlights the difference between the transformers used in the study in terms of the advantages and limitations. Table 3 puts forth the configurations used in the work for each of the transformer models.

3.4 Proposed Architecture

To obtain contextual and subword information, post-processing BERT and fastText embeddings (K et al. (2024)) are used.A Multi-CNN architecture is used to analyse these embeddings, extracting multi-scale context through the use of parallel CNN layers with different filter sizes (El-Rashidy et al. (2023)). A rich, integrated feature representation is produced by merging and transforming the outputs from the Multi-CNN, FastText, and BERT. Next, an Enhanced Information Gain (EIG) is used to identify crucial features that expresses strong discriminative power, and an Ant Colony Optimisation (ACO) is used to fine-tune feature weights in order to accomplish optimised feature selection. These features are given relevance weights by the application of an attention mechanism, and then dimensionality reduction is carried out using the Pearson Correlation Coefficient (PCC) in order to retain less correlated and more diversified features. Finally for classification,ensemble classifiers is used, that involves training many base classifiers on a reduced feature set and combining their predictions using a meta-classifier.

3.4.1 Feature Extraction

1. **BERT Embedding:**The preprocessed text are fed into a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model. BERT is a deep learning model notable for extracting rich contextual information from text. It generates embeddings that represent each token in relation to the complete phrase, capturing both left-to-right and right-to-left dependencies. This allows the model to recognise the subtle meaning of words based on their context, which is very beneficial for identifying hate speech and hostility.

2. **FastText Embedding:** Unlike traditional word embeddings, FastText represents words as a collection of character n-grams, allowing it to create

embeddings even for words not seen during training. This is crucial for understanding variations in language often present in hate speech.

3.4.2 Multi-Scale Context Extraction

Both BERT and FastText embeddings are fed into multiple parallel CNN layers, each with different filter sizes. The architecture utilizes multiple parallel CNN layers, each equipped with filters of varying sizes. The rationale behind using different filter sizes is to capture linguistic features at multiple granularities:

- Small Filters (e.g., 2-3 words): These filters are designed to capture fine-grained details, such as word pairs or short phrases. This allows the model to detect local context, such as idiomatic expressions, word associations, or specific phrases that may indicate hate speech or aggression.
- Medium Filters (e.g., 4-5 words): Filters of medium size focus on slightly larger contexts, such as clauses or short sentences. This helps the model understand relationships between words that are close to each other but not directly adjacent, such as subject-verb-object structures.
- Large Filters (e.g., 6-7 words or more): Larger filters are employed to capture broader context, such as entire sentences or even multiple sentences. This is crucial for understanding the overall tone, sentiment, and intent behind the text, as well as for detecting more subtle forms of hate speech that might span across multiple words or sentences.

Each CNN layer operates by applying its filters across the entire length of the embeddings using a sliding window approach. As the filters move across the embeddings, they perform convolutional operations that aggregate and summarize the information within each window. This results in feature maps that highlight important patterns and structures in the text.The convolutional operation produces a set of feature maps for each filter size, where each map captures different aspects of the text based on the size of the filter used. For instance, a feature map generated by a small filter might highlight specific word pairings indicative of aggressive language, while a feature map from a larger filter might capture the overall sentiment of a sentence.After convolution, each feature map is passed through an activation function, typically ReLU (Rectified Linear Unit), which introduces non-linearity into the model. ReLU helps in activating only the most relevant features while suppressing less important ones. This step is essential for allowing the model to focus on key patterns that are indicative of hate speech or aggression, while ignoring irrelevant or noise features.Following activation, max-pooling is applied to reduce the dimensionality of the feature maps while retaining the most critical information. Max-pooling selects the maximum value from each region of the feature map, which corresponds to the most prominent feature detected by the filter. This operation helps in summarizing the information and making the model more computationally efficient without losing significant context.The outputs from the various CNN layers, each corresponding to different filter sizes, are then combined to form a comprehensive set of

multi-scale contextual features. These features capture a wide range of information—from the fine-grained details of specific phrases to the broader patterns spanning entire sentences.To prepare these features for subsequent processing stages, they are often flattened and concatenated into a single vector that represents the entire text at multiple contextual levels. This vector serves as a rich, integrated feature set that encapsulates the diverse linguistic elements present in the text.

3.4.3 Feature Merging

After extracting multi-scale contextual features, the outputs from BERT embeddings, FastText embeddings, and the Multi-CNN layers are concatenated into a single vector that integrates the strengths of each embedding method and the multi-scale CNN outputs. This concatenated vector then undergoes a non-linear transformation, typically through a fully connected neural network layer with an activation function like ReLU (Rectified Linear Unit). This step enhances the model's ability to learn complex interactions between features, creating a richer, more informative representation for the classification task.

3.4.4 Optimized Feature Selection

Enhanced Information Gain (EIG) is an advanced feature selection method that builds upon the traditional Information Gain (IG) concept. It's particularly useful in this context because it can capture both linear and non-linear relationships between features and the target classes (hate speech, aggressiveness, and neutral content). The steps through which EIG is implemented is as follows:

1. **Entropy Calculation:**
   For each feature F_i, calculate its entropy $E(F_i)$:

   $$E(F_i) = -\sum p(f_i) \log_2(p(f_i)) \tag{1}$$

   where $p(f_i)$ is the probability of value f_i in feature F_i. This measures the uncertainty or randomness in the feature's distribution.

2. **Conditional Entropy:**
   Calculate the conditional entropy $CE(C|F_i)$ of the class variable C given feature F_i:

   $$CE(C|F_i) = -\sum p(f_i) \sum p(c|f_i) \log_2(p(c|f_i)) \tag{2}$$

   where $p(c|f_i)$ is the conditional probability of class c given feature value f_i.

3. **Information Gain:**
   Compute the Information Gain $IG(C, F_i)$:

   $$IG(C, F_i) = E(C) - CE(C|F_i) \tag{3}$$

This measures how much information feature F_i provides about the class C.

4. **Enhanced Information Gain:**
   EIG extends this concept by considering feature interactions. For each pair of features F_i and F_j:
   (a) Calculate joint entropy $E(F_i, F_j)$.
   (b) Compute conditional entropy $CE(C|F_i, F_j)$.
   (c) Calculate EIG:

$$EIG(C, F_i, F_j) = E(C) - CE(C|F_i, F_j) \tag{4}$$

1. Feature Ranking: Rank features based on their EIG scores. Higher EIG indicates stronger predictive power for hate speech and aggressiveness classification.

2. Feature Selection: Select the top K features with the highest EIG scores.

Furthermore,Ant Colony Optimization (ACO) is used to fine-tune the feature weights, optimizing the feature subset selected by EIG.

3.4.5 Pearson Correlation Coefficient (PCC) Based Feature Reduction

The correlation matrix of the features is calculated using the Pearson Correlation Coefficient, which measures the linear relationship between pairs of features. Features that are highly correlated with each other (with absolute PCC value close to +1 or -1) are likely to carry redundant information, so a subset of features that are less correlated is selected. This step reduces the dimensionality of the feature space, improving computational efficiency while retaining diverse and informative features for classification.

3.4.6 Attention-Based Weighting

Once the optimized feature set is determined, an attention mechanism is applied to assign importance weights to the features. The steps followed are as follows:

Step 1: Feature Representation

- **Input:** Optimized set of features $\mathbf{F} = \{f_1, f_2, \ldots, f_n\}$, where each feature f_i is a vector representing a specific aspect of the input text.
- **Output:** A set of feature vectors.

Step 2: Compute Importance Scores

- For each feature vector f_i:
  - Pass f_i through a small neural network or a linear layer to compute an importance score s_i.

- Equation: $s_i = \text{NN}(f_i)$ or $s_i = \mathbf{w}^T f_i + b$, where $\mathbf{w}$ and b are learnable parameters.

Step 3: Normalize Scores Using Softmax

- **Input:** Set of importance scores $\{s_1, s_2, \ldots, s_n\}$.
- **Compute:** Attention weights α_i by applying the softmax function to the scores.
- Equation: $\alpha_i = \frac{\exp(s_i)}{\sum_{j=1}^{n} \exp(s_j)}$.

Step 4: Apply Attention Weights

- For each feature vector f_i:
  - Multiply the feature vector by its corresponding attention weight α_i to emphasize its importance.
  - Equation: $f_i' = \alpha_i \times f_i$.

Step 5: Aggregate Weighted Features

- Aggregate the weighted feature vectors to form a final feature representation $\mathbf{F}'$ for classification.
- Equation: $\mathbf{F}' = \sum_{i=1}^{n} f_i'$.

Step 6: Use for Classification

- **Input:** Aggregated feature vector $\mathbf{F}'$.
- **Output:** Pass $\mathbf{F}'$ to the classifier for final hate speech and aggressiveness prediction.

3.4.7 Ensemble Classification

With the reduced and optimized feature set, the model employs an ensemble classification approach to improve accuracy and robustness.

Multiple Base Classifiers: Several base classifiers are trained on the reduced feature set. The classifiers used here are Random Forest, Support Vector Machines (SVM), Extreme Gradient Boosting (XGB), and Neural Networks, each bringing different strengths to the table. Random Forests are robust to overfitting, SVMs are effective in high-dimensional spaces, and Neural Networks excel at capturing complex patterns.

Meta-Classifier: Once the base classifiers are trained, their predictions are combined using a meta-classifier. This approach is often referred to as stacking. The meta-classifier learns to weigh the predictions from the different base classifiers to produce a final, more accurate classification.The meta-classifier used here is logistic regression.

Predictions from Base Classifiers: Each base classifier C_i makes a prediction for a given input x:

$$p_i = C_i(x) \tag{5}$$

The vector of predictions from all base classifiers is

$$\mathbf{p} = [p_1, p_2, \ldots, p_m] \tag{6}$$

where m is the number of base classifiers.

Training the Meta-Classifier: The meta-classifier M is trained on the output $\mathbf{p}$ of the base classifiers. A common choice for the meta-classifier is logistic regression, which models the probability of the target class as:

$$M(\mathbf{p}) = \frac{1}{1 + \exp\left(-(\mathbf{w} \cdot \mathbf{p} + b)\right)} \tag{7}$$

where $\mathbf{w}$ is the weight vector learned by the meta-classifier, and b is the bias term.

Final Prediction: The final classification result y is given by:

$$y = \arg\max_{c \in \{0,1\}} M(\mathbf{p}_c) \tag{8}$$

where $\mathbf{p}_c$ represents the prediction vector for class c.

Table 2: Comparison of BERT, DistilBERT, ELECTRA, and RoBERTa

Aspect	**BERT**	**DistilBERT**	**ELECTRA**	**RoBERTa**
Advantages	High performance with bidirectional context understanding.	More efficient with reduced parameters. Faster training.	Competitive performance. Efficient training. Discriminative pre-training method.	Often outperforms BERT. Trained on more data. No next-sentence prediction.
Disadvantages	Large model size. Moderate training efficiency.	Slightly lower performance than BERT.	Complex training setup. Needs careful tuning.	Large model size. High computational cost.
Model Configs	12 layers, 768 hidden size, 110M params.	6 layers, 768 hidden size, 66M params.	12 layers, 768 hidden size, 110M params.	12 layers, 768 hidden size, 125M params.

Table 3: Hyperparameter configurations used in the study

Parameter	**ELECTRA**	**RoBERTa**	**Fine Tuned DistilBERT**	**DistilBERT**
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning Rate	2eˆ{-5}	2eˆ{-5}	0.0004	3eˆ{-5}
Batch Size	32	32	16	16
Maxlen	128	256	128	100
Epochs	3	3	10	4
Random Seed	42	42	42	42
Learning Rate Scheduler	Warm up over first 10%	Warm up over first 10%	Warm up over first 10%	Warm up over first 10%
Loss Function	Cross- Entropy Loss	Cross-Entropy Loss	Cross-Entropy Loss	Cross-Entropy Loss
Validation method	5 fold cross validation and Early stopping based on validation loss	5 fold cross validation and Early stopping based on validation loss	5 fold cross validation and Early stopping based on validation loss	5 fold cross validation and Early stopping based on validation loss
Data Sets used for pre-training			Hugging Face Twitter Hate Speech dataset (Sachdeva et al. (2022)) and Tweets reporting Abuse classification task (Reddy et al. (2021)) (Mishra (2023))	

4 Results and Analysis

4.1 Exploratory Data Analysis

4.1.1 EDA on Hate Speech and Offensive Language Dataset(HSOL) dataset

HSOL (Davidson et al. (2017), Davidson & Others (2017))is a dataset used to train models for detecting hate speech. The writers started with a dictionary of hate speech lexicons that was put together by Hatebase.org and contains terms and phrases that have been classified as hate speech by internet users. They found a sample of tweets by using the Twitter API to look for tweets that contained terms from the dictionary. They selected a random sample of 25k tweets from this corpus that contained phrases from the glossary and had CrowdFlower (CF) employees manually code them. Each tweet was classified in one of three categories: clean, objectionable but not hate speech or hate speech. Our dataset consists of 24,788 rows of tweets and 3 emotions that are: hate crime, abuse or neither. We created another column called emotion, marking each tweet as one of the 3 classes. The following is the dataset distribution with respect to the three classes: 0 for Neither, 1 for Abuse and 2 for Hate Crime. Figure 3 shows the distribution of the dataset.

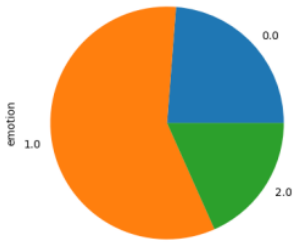

Figure 3: Class distribution of the dataset

PMI stands for Pointwise Mutual Information, and it is a statistical measure used to quantify the association between two discrete random variables. In the context of NLP, PMI is often used to measure the correlation between a class (e.g., a category or topic) and a word or group of words. The obtained results on our dataset are tabulated in Table 4 and 5.

The dataset is a mix of various words but focused on certain trigger words. In our case, Class 1.0 has the highest number of instances (14347), while Class 2.0 has the lowest number (4543). This indicates a class imbalance, where one or more classes have disproportionately more or fewer instances than others.

Class imbalances can adversely affect the performance of the text classification model. Models may become biased towards the majority class (Class 1.0 in this case) and might struggle to correctly classify instances from the minority classes (Class 0.0 and Class 2.0)

Word	PMI
b∗tch	1.65×10^{-6}
f∗gg∗t	32.47×10^{-6}
f∗ck	5.86×10^{-6}
n∗gg∗	6.31×10^{-6}
n∗gg∗r	35.91×10^{-6}

Table 4: PMI for Most Frequent Words Indicating Hate Crime

Word	PMI
b∗tch	5.08×10^{-6}
h∗∗	4.97×10^{-6}
p∗ss∗	5.08×10^{-6}
f∗ck	4.01×10^{-6}
n∗gg∗	3.92×10^{-6}

Table 5: PMI for Most Frequent Words Indicating Abuse

4.2 EDA on Aggressiveness Dataset

In this research, the Cyber-Troll publicly available dataset (Dataturks (n.d.)) was used. This dataset was produced by Data-Turk with the purpose of detecting hostility. The dataset includes tweets in the English language that have been divided into two categories by the Data-Turk society: cyber-aggressive (CA) and non-cyber-aggressive (NCA). The messages in cyber-aggressive tweets are meant to offend or harm someone online. On the other hand, non-cyberaggressive tweets are those that have no malicious intent and do not hurt other people. 20,001 tweets make up the dataset, of which 12,179 are NCA and 7822 are CA.

4.3 Metrics Used

1. Accuracy: To measure the overall correctness of the models in classifying tweets into their respective categories.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{9}$$

2. F1 Score: To assess the models' abilities to balance precision and recall for hate crime and abusive class predictions.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

where

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

and

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

where TP, TN, FP, FN stand for True Positives, True Negatives, False Positives, False Negatives respectively.

3. Specificity:

$$\text{Specificity } (\%) = \left(\frac{\text{TN}}{\text{TN} + \text{FP}} \right) \times 100 \tag{13}$$

4. Matthews Correlation Coefficient (MCC):

$$\text{MCC } (\%) = \left(\frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \times (\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TN} + \text{FN})}} \right) \times 100 \tag{14}$$

5. Jaccard Index:

$$\text{Jaccard Index} = \frac{|A \cap B|}{|A \cup B|} \tag{15}$$

6. Parameters Used: To analyze the computational cost and resource requirements of each model.

7. Inference Speed: The time taken by the model to process one input instance during inference is crucial, especially in real-time applications. This metric is measured in terms of inference time per sample.

   - Inference time per sample:

$$\text{Inference time per sample} = \frac{\text{Total time taken}}{\text{Number of samples}} \tag{16}$$

4.4 Transformer results

The results of the classification task are displayed in Table 6.

Table 6: Performance Comparison of Different Classifiers

Metric	DistilBERT	ELECTRA	RoBERTa	DistilBERT (HF)	Improved RoBERTa	Proposed Model
Accuracy (%)	72	76	71	80	89	**91.52**
F1 Score (%)	75	75	72	81.25	89.32	**92.91**
Precision (%)	75	78	70.59	78	89.32	**94.95**
Recall (%)	68.18	74	71.42	82	88.65	**83.85**
Specificity (%)	75	76.47	71.28	79.59	89.32	**93.92**
MCC (%)	43.18	52.04	42.01	60.04	77.98	**79.94**
Jaccard Index (%)	60.00	65.00	55.38	66.10	80.70	**70.37**
Parameters (M)	66	110	123	66	66	**94**
Inference Speed (ms)	50	300	15	50	70	**78**

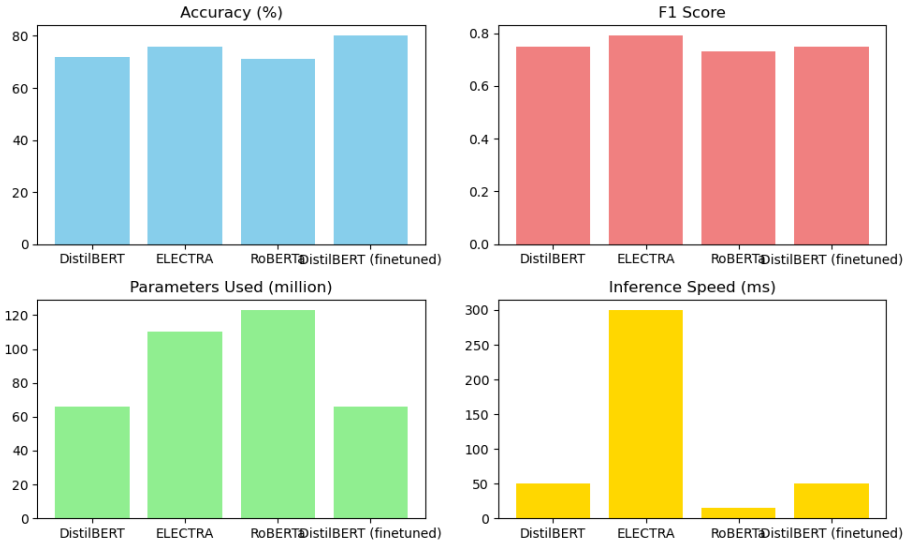

Figure 4: Graphs comparing the models employed in the study

Figure 4 shows the graphical representation of the study.

5 Discussion

The findings clearly demonstrate the suggested method's better capacity to detect hate speech, abuse, and other damaging words on social media sites. By combining numerous embeddings, the approach excels at detecting subtle and diverse kinds of hate speech. The multi-scale CNN technique allows the model to analyse both fine-grained (word-level) and wider (sentence or document-level) contexts at the same time, increasing its capacity to reliably categorise complicated and context-dependent hate speech instances. Unlike basic NLP models, which depend on simpler or less effective feature selection approaches, the proposed method's enhanced feature selection produces a more targeted and potent collection of features, directly improving classification performance. This, in conjunction with the attention process, guarantees that even subtle or indirect instances of hate speech are captured.

Furthermore, fine-tuning DistilBERT using Hugging Face datasets or improving RoBERTa with enhanced layer normalisation results in better performance than basic models. These transformer-based models, which have been pre-trained on large text corpora, are extremely efficient at capturing deep contextual links between words, which is critical for recognising the subtle signs and complex linguistic patterns that are characteristic of hate speech. When fine-tuned with domain-specific data, these models improve their ability to recognise the intricacies of hostile content on platforms such as social media. The enhanced layer normalisation in RoBERTa, in particular, improves model sta-

bility and learning efficiency, resulting in higher generalisation, particularly in diverse and dynamic contexts. DistilBERT's simplified design facilitates quicker training and inference while maintaining high performance, making it ideal for real-time applications. Overall, these models provide richer, more accurate feature representations that capture both syntactic and semantic features, as well as the subtle patterns associated with hate speech, improving detection accuracy.

In addition to individual model performance, ensemble learning played a crucial role in enhancing classification reliability. As shown in the ROC curves (Figure 5), the meta-classifier—an ensemble built on top of NN, RF, SVM, and XGB using logistic regression as the combiner—achieved an AUC of 0.96, matching the highest-performing base models (NN and XGB). This indicates that the ensemble was able to leverage the complementary strengths of each constituent classifier. While Random Forest (AUC = 0.92) and SVM (AUC = 0.90) underperformed slightly compared to the neural and boosting models, their inclusion in the ensemble contributed to robustness by capturing different patterns of bias and variance.

The ROC analysis further reveals that the ensemble exhibits strong discriminative ability across a range of thresholds, maintaining a high true positive rate with a very low false positive rate. This is particularly valuable in hate speech detection, where the cost of false negatives (missing harmful content) can be severe. The steep ascent in the ROC curve near the origin and the wide margin over the diagonal line confirm that the classifier makes decisions with high confidence in most cases. Thus, integrating a meta-classifier enhances both stability and generalisation, reinforcing the practical utility of ensemble techniques in real-world, adversarial social media environments.

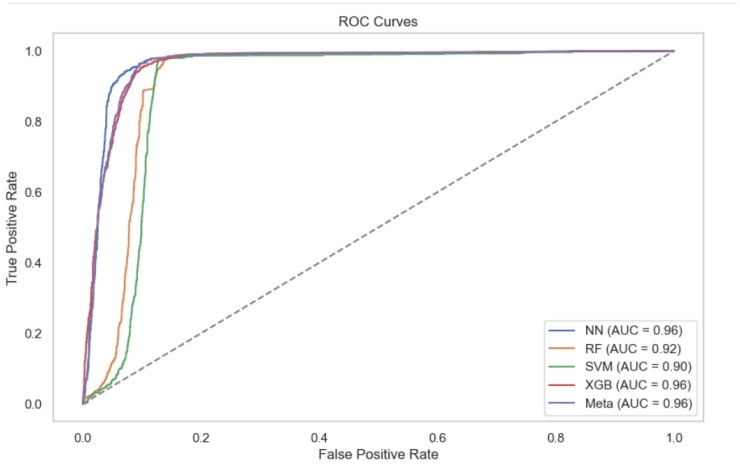

Figure 5: ROC curves of base models (NN, RF, SVM, XGB) and the meta-classifier (ensemble). The meta-classifier achieves an AUC of 0.96, demonstrating high discriminative power.

6 Conclusions

This study effectively demonstrated the suggested NLP model's ability to recognise hate speech, abusive language, and other detrimental data on social media sites. By combining feature engineering with effective ensemble learning, the model outperforms fundamental NLP models by about 15%. DistilBERT fine-tuning and RoBERTa enhancement through enhanced layer normalisation have been demonstrated to increase performance, emphasising the relevance of domain-specific training and model stability when dealing with complicated language patterns associated with hate speech.The findings highlight the potential of advanced NLP approaches to help create a safer and more inclusive online environment by properly recognising and dampening abusive speech.

6.1 Study Limitations and discussion on the findings

There are a few drawbacks to this study that should be noted. In the first place, it demonstrates how important user intent and context are for determining if something is abusive or aggressive. This study merely classifies violent and abusive language on social media at a superficial level; a thorough literature assessment from psychological and linguistic perspectives is necessary to fully comprehend these occurrences. While other studies in this field cover more comprehensive views of violence from texts, pictures, videos, and images, this work just addresses textual material. The analysis of inaccurate predictions demonstrated the difficulties in recognizing writings that implicitly promote or convey abuse and aggressiveness. Such incidents lack obvious abusive or violent references or language, making them difficult to identify. In contrast, some

texts employ violent or abusive phrases ironically, with no intention of causing damage, yet the model incorrectly identifies them as hostile or abusive. This emphasizes the difficulty of detecting and categorizing such text samples without a better grasp of context. Furthermore, to further complicate matters, some terms are used often in both abusive/aggressive and non-abusive/non-aggressive classes. The use of these imprecise terms (syntactic ambiguity) in the text makes categorization even more difficult. The model's classification performance may be enhanced by using more training data, including their meta-information, and including contextual analysis of violent and abusive communications. The literature's quantitative and qualitative error assessments highlight how challenging it is to recognize abuse and aggression that is communicated subtly or ironically, which is a serious obstacle for the proposed framework. The imbalance in the classes is one of the major setbacks which affects the accuracy. The identified imbalances can further be augmented upon using sampling techniques like Synthetic Minority Over-sampling Technique (SMOTE) in future research. Thus,the observed findings underlines the significance of user intent and context in identifying abusive language, emphasising the necessity for models that grasp subtleties such as sarcasm, ambiguity and implied abuse. Addressing class imbalances and combining psychological and linguistic insights might help enhance model accuracy. Future study should concentrate on broader data sets and improved algorithms to improve identification and moderation.

6.2 Significance of the Results Obtained

6.2.1 Implications of the findings

Studies in this field contribute to a more positive online community by swiftly and efficiently detecting and removing offensive language from social media. This enhances the security and quality of online interactions, ensuring civil and productive conversations. By shielding users from toxic language and establishing standards for acceptable behavior, these systems can positively impact public attitudes towards online interactions. Promoting respectful and considerate communication helps decrease the incidence of online harassment and discrimination, creating a more welcoming digital space for all users. This improvement can result in increased engagement and participation from users who might otherwise be deterred by a hostile environment. Additionally, by preventing the spread of misinformation and extremist content, effective hate speech detection supports a more informed and balanced public discourse, which is crucial for a healthy democratic society. Automated detection of abusive content can also assist law enforcement and internet marketing by protecting their reputations and enhancing the user experience on their platforms. Compared to traditional machine learning models, transformers—specifically, models like BERT, GPT, and their derivatives—have shown superior performance in natural language interpretation tasks, leading to more accurate identification of hate speech, abuse, and aggression. Transformers can also efficiently handle large volumes of data, enabling real-time monitoring of social media platforms. This capability allows

for quicker moderation, reducing the spread of harmful content and safeguarding users from exposure to such material.

6.2.2 Practical Applications and Resource considerations

Identifying hate crime and abusive content in social media posts is crucial for creating safer online spaces. The successful application of these transformer-based models in detecting harmful language demonstrates their potential to be deployed as part of automated content moderation systems on various social media platforms. The low inference time of RoBERTa (15 ms) makes it particularly suitable for real-time applications where swift content analysis is essential. The analysis of parameters used by each model (66 million for DistilBERT, 110 million for ELECTRA, and 123 million for RoBERTa) can help organizations make informed decisions regarding computational costs and resource requirements. Smaller models like DistilBERT offer a good trade-off between performance and resource utilization, while RoBERTa might be preferred for scenarios where low inference time is critical.

In conclusion, the project's results highlight the effectiveness of transformer-based models in detecting harmful language on social media platforms. Fine-tuning DistilBERT proved to be a powerful approach for achieving superior performance, while RoBERTa demonstrated excellent inference speed. These findings provide valuable insights into text classification tasks, demonstrating the significance of customizing pre-trained models to specific domains and their potential impact on content moderation and safer online communities.

7 Future Scope of the Work

The future directions for refining models to detect harmful social media language underscore several critical aspects. Primarily, future work should focus on enhancing data collection methodologies, ensuring the diversity of labeled data by accounting for regional variations and representing the demographics of the population accurately. While current research has predominantly utilized BERT-based transformer models, future efforts could incorporate newer models such as GPT-4, T5, and BART (Lewis et al. (2019)) . Exploring variations in deep learning models can also improve detection accuracy.

Present methods are limited to textual data only. Expanding to other forms of data like images and memes can significantly enhance detection capabilities. Additionally, a multimodal analysis that leverages text, social media interactions, and other data forms can provide additional clues about intent and context, reducing false positives and improving accuracy. This approach is particularly useful in situations with data sparsity.

It is proposed to create an intuitive, user-friendly web-based application to expand the accessibility and practical use of these models to a larger audience. Incorporating multilingual capabilities, however difficult owing to scarce labelled data in many languages, is another worthwhile addition. Using in-

terpretable and explainable AI is advantageous, especially in hate detection settings when actions based on model judgments are required. A cross-domain generalized strategy that supports a variety of platforms and languages can also be investigated.

Improving the model by including deeper contextual understanding, such as analyzing bigger conversation threads, user profiles, and past interactions, can offer additional insight about the intent behind the language used. More complex Natural Language Processing (NLP) tools, such as integration of sentiment analysis, emotion detection, and discourse analysis, can assist discern true animosity from irony or sarcasm. These components are ubiquitous in online communication but provide difficulties for automated detection systems. Finally, collaborative filtering, which makes suggestions based on user similarities, may be used to identify probable cases of abuse or violence by analyzing interaction patterns.

References

Alqahtani, T., Badreldin, H. A., Alrashed, M., Alshaya, A. I., Alghamdi, S. S., bin Saleh, K., Alowais, S. A., Alshaya, O. A., Rahman, I., Al Yami, M. S. & Albekairy, A. M. (2023), 'The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research', *Research in Social and Administrative Pharmacy* **19**(8), 1236–1242.
URL: *https://www.sciencedirect.com/science/article/pii/S1551741123002802*

Alrehili, A. (2019), Automatic hate speech detection on social media: A brief survey, *in* '2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)', pp. 1–6.

Arya, G., Hasan, M. K., Bagwari, A., Safie, N., Islam, S., Ahmed, F. R. A., De, A., Khan, M. A. & Ghazal, T. M. (2024), 'Multimodal hate speech detection in memes using contrastive language-image pre-training', *IEEE Access* **12**, 22359–22375.

Asif, M., Al-Razgan, M., Ali, Y. A. & Yunrong, L. (2024), 'Graph convolution networks for social media trolls detection use deep feature extraction', *Journal of Cloud Computing* **13**(1), 33.
URL: *https://doi.org/10.1186/s13677-024-00600-4*

Bacha, J., Ullah, F., Khan, J., Sardar, A. W. & Lee, S. (2023), 'A deep learning-based framework for offensive text detection in unstructured data for heterogeneous social media', *IEEE Access* **11**, 124484–124498.

Baruah, A., Wahlang, L., Jyrwa, F., Shadap, F., Barbhuiya, F. & Dey, K. (2024), 'Abusive language detection in khasi social media comments', *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* . Just Accepted.
URL: *https://doi.org/10.1145/3664285*

Clark, K., Luong, M.-T., Le, Q. V. & Manning, C. D. (2020), 'Electra: Pre-training text encoders as discriminators rather than generators', *arXiv preprint arXiv:2003.10555* .

Dataturks (n.d.), 'Tweets Dataset for Detection of Cyber-Trolls — kaggle.com', `https://www.kaggle.com/datasets/dataturks/dataset-for-detection-of-cybertrolls`. [Accessed 27-08-2024].

Davidson, T. & Others (2017), 'Hate speech and offensive language dataset', `https://github.com/t-davidson/hate-speech-and-offensive-language`.

Davidson, T., Warmsley, D., Macy, M. & Weber, I. (2017), Automated hate speech detection and the problem of offensive language, *in* 'Proceedings of the international AAAI conference on web and social media', Vol. 11, pp. 512–515.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018), 'Bert: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

El-Rashidy, M. A., Farouk, A., El-Fishawy, N. A., Aslan, H. K. & Khodeir, N. A. (2023), 'New weighted bert features and multi-cnn models to enhance the performance of mooc posts classification', *Neural Computing and Applications* **35**(24), 18019–18033.
URL: *https://doi.org/10.1007/s00521-023-08673-z*

Fortuna, P., Domínguez, M., Wanner, L. & Talat, Z. (2022), Directions for nlp practices applied to online hate speech detection, *in* 'Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing', pp. 11794–11805.

Han, H., Asif, M., Awwad, E. M., Sarhan, N., Ghadi, Y. Y. & Xu, B. (2024), 'Innovative deep learning techniques for monitoring aggressive behavior in social media posts', *Journal of Cloud Computing* **13**(1), 19.
URL: *https://doi.org/10.1186/s13677-023-00577-6*

Hashmi, E. & Yayilgan, S. Y. (2024), 'Multi-class hate speech detection in the norwegian language using fast-rnn and multilingual fine-tuned transformers', *Complex & Intelligent Systems* **10**(3), 4535–4556.
URL: *https://doi.org/10.1007/s40747-024-01392-5*

Jahan, M. S. & Oussalah, M. (2023), 'A systematic review of hate speech automatic detection using natural language processing', *Neurocomputing* **546**, 126232.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231223003557*

K, N., M, K., Easwaramoorthy, S. V., C R, D., Yoo, S. & Cho, J. (2024), 'Hybrid approach of deep feature extraction using bert- opcnn & fiac with customized bi-lstm for rumor text classification', *Alexandria Engineering*

Journal **90**, 65–75.
URL: *https://www.sciencedirect.com/science/article/pii/S1110016824000759*

Khan, U., Khan, S., Rizwan, A., Atteia, G., Jamjoom, M. M. & Samee, N. A. (2022), 'Aggression detection in social media from textual data using deep learning models', *Applied Sciences* **12**(10).
URL: *https://www.mdpi.com/2076-3417/12/10/5083*

Khurana, D., Koli, A., Khatter, K. & Singh, S. (2023), 'Natural language processing: state of the art, current trends and challenges', *Multimedia Tools and Applications* **82**(3), 3713–3744.
URL: *https://doi.org/10.1007/s11042-022-13428-4*

Kovács, G., Alonso, P. & Saini, R. (2021), 'Challenges of hate speech detection in social media', *SN Computer Science* **2**(2), 95.
URL: *https://doi.org/10.1007/s42979-021-00457-3*

Kumar, A., Saumya, S. & Singh, J. P. (2020), Nitp-ai-nlp@ hasoc-fire2020: Fine tuned bert for the hate speech and offensive content identification from social media., *in* 'FIRE (Working Notes)', pp. 266–273.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. & Zettlemoyer, L. (2019), 'Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension', *arXiv preprint arXiv:1910.13461* .

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* .

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N. & Frieder, O. (n.d.), 'Hate speech detection: Challenges and solutions.', **14**(8), e0221152.
URL: *https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0221152*

Mishra, P. (2023), 'Text classification with distilbert - 92% accuracy', Kaggle notebook. Accessed: 2023-11-03.
URL: *https://www.kaggle.com/code/pritishmishra/text-classification-with-distilbert-92-accuracy/notebook*

Munthuli, A., Socatiyanurak, V., Sangchocanonta, S., Kovudhikulrungsri, L., Saksakulkunakorn, N., Chairuangsri, P. & Tantibundhit, C. (2023), 'Transformers for multi-intent classification and slot filling of supreme court decisions related to sexual violence law', *IEEE Access* .

Nandi, A., Sarkar, K., Mallick, A. & De, A. (2024), 'A survey of hate speech detection in indian languages', *Social Network Analysis and Mining* **14**(1), 70.
URL: *https://doi.org/10.1007/s13278-024-01223-y*

Papcunová, J., Martončik, M., Fedáková, D., Kentoš, M., Bozogáňová, M., Srba, I., Moro, R., Pikuliak, M., Šimko, M. & Adamkovič, M. (2023), 'Hate speech operationalization: a preliminary examination of hate speech indicators and their structure', *Complex & intelligent systems* **9**(3), 2827–2842.

Qureshi, K. A. & Sabih, M. (2021), 'Un-compromised credibility: Social media based multi-class hate speech classification for text', *IEEE Access* **9**, 109465–109477.

Reddy, S. M., Tyagi, K., Tripathi, A. A., Pawar, A. & Kotecha, K. (2021), Tweets reporting abuse classification task: Tract, *in* 'Congress on Intelligent Systems: Proceedings of CIS 2020, Volume 2', Springer, pp. 305–314.

Sachdeva, P., Barreto, R., Bacon, G., Sahn, A., von Vacano, C. & Kennedy, C. (2022), The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism, *in* G. Abercrombie, V. Basile, S. Tonelli, V. Rieser & A. Uma, eds, 'Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022', European Language Resources Association, Marseille, France, pp. 83–94.
URL: *https://aclanthology.org/2022.nlperspectives-1.11*

Sanh, V., Debut, L., Chaumond, J. & Wolf, T. (2019), 'Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter', *arXiv preprint arXiv:1910.01108* .

Sharif, O. & Hoque, M. M. (2022), 'Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers', *Neurocomputing* **490**, 462–481.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231221018567*

Song, R., Giunchiglia, F., Shen, Q., Li, N. & Xu, H. (2022), 'Improving abusive language detection with online interaction network', *Inf. Process. Manage.* **59**(5).
URL: *https://doi.org/10.1016/j.ipm.2022.103009*

Zia Ur Rehman, M., Mehta, S., Singh, K., Kaushik, K. & Kumar, N. (2023), 'User-aware multilingual abusive content detection in social media', *Inf. Process. Manage.* **60**(5).
URL: *https://doi.org/10.1016/j.ipm.2023.103450*