Bias and Cyberbullying Detection and Data Generation with Transformer AI Models and top LLMs

Yulia Kumar *, Kuan Huang *, Angelo Perez, Guohao Yang *, J. Jenny Li *, Patricia Morreale, Dov Kruger *, Raymond Jiang *

Posted Date: 4 July 2024

doi: 10.20944/preprints202407.0411.v1

Keywords: Synthetic Data; Bias Data Generator; Large Language Models (LLMs); Cyberbullying Detection; Inherent Biases; Transformer Models; Bias Detection Tokens; Swarm of AI Agents.

Article

Bias and Cyberbullying Detection and Data Generation with Transformer AI Models and Top LLMs

Yulia Kumar [1,*], Kuan Huang [1], Angelo Perez [1], Guohao Yang [1], J. Jenny Li [1], Patricia Morreale [1], Dov Kruger [2] and Raymond Jiang [3]

[1] Department of Computer Science and Technology, Kean University, Union, NJ 07083, USA; ykumar@kean.edu (Y.K.), khuang@kean.edu (K.H.), peangelo@kean.edu (A.P.), yanggu@kean.edu (G.Y.), juli@kean.edu (J.J.L.), pmorreal@kean.edu (P.M.)
[2] Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854, USA; dov.kruger@rutgers.edu
[3] High Technology High School, Lincroft, NJ 07738, USA; raymondjiang10@gmail.com
* Correspondence: ykumar@kean.edu

Abstract: Despite significant advancements in Artificial Intelligence (AI) and Large Language Models (LLMs), detecting and mitigating bias remains a critical challenge, particularly within social media platforms like X (formerly Twitter) and in addressing cyberbullying present on them. This research investigates the effectiveness of leading LLMs in generating synthetic biased and cyberbullying data and evaluates the proficiency of Transformer AI models in detecting bias and cyberbullying within both authentic and synthetic contexts. The study involves semantic analysis and feature engineering on a dataset of over 48,000 sentences related to cyberbullying collected from Twitter (before it became X). Leveraging state-of-the-art LLMs such as ChatGPT-4o, Pi AI, Claude 3 Opus, and Gemini-1.5, synthetic biased, cyberbullying, and neutral data were generated to deepen the understanding of bias in human-generated data. AI models including DeBERTa, Longformer, BigBird, HateBERT, MobileBERT, DistilBERT, BERT, RoBERTa, ELECTRA, and XLNet were initially trained to classify Twitter cyberbullying data and subsequently fine-tuned, optimized, and experimentally quantized. The study's outcomes include a prototype of a hybrid application that combines a Bias Data Detector and a Bias Data Generator.

Keywords: synthetic data; bias data generator; large language models (LLMs); cyberbullying detection; inherent biases; transformer models; bias detection tokens; swarm of ai agents

1. Introduction

Bias detection and mitigation using Artificial Intelligence (AI) models, including Transformers, have been focal points within the research community for several years [1–4]. The involvement of Large Language Models (LLMs) in generating biased, cyberbullying, and neutral context data, as well as the application of AI algorithms to both synthetic and authentic biased datasets, particularly in the context of cyberbullying, presents a fertile ground for scientific exploration and discovery. This study addresses the following research questions:

RQ1: What strategies can enhance bias and cyberbullying detection within both synthetic and authentic neutral and cyberbullying datasets?

RQ2: How can key advanced Transformer models, pretrained to detect biases and work with social media platform data, and leading LLMs be used in understanding bias in datasets and AI models?

RQ3: How does the intersection of cyberbullying and bias detection in multilabel classification using Transformers can improve both bias and cyberbullying detection within neutral and cyberbullying datasets?

By addressing these questions, this research aims to contribute to the development of fairer and more reliable AI systems with robust bias and cyberbullying detection capabilities.

 doi:10.20944/preprints202407.0411.v1

The rapid proliferation of synthetic data generated by advanced AI systems has intensified the need to address biases inherent in such models. AI systems can both detect and generate biased and cyberbullying data, presenting a dual challenge that necessitates thorough investigation. Biases can stem from various sources, including biased training data, algorithmic design, and human prejudices, significantly impacting the performance and trustworthiness of AI applications. In sensitive applications like cyberbullying detection, these biases can result in unfair flagging or overlooking certain demographics [2,4].

Cyberbullying, a form of harassment occurring through digital platforms, has become a significant concern in recent years. AI's ability to mimic human behavior is evident in various applications, such as automated accounts acting as real users and chatbots. However, these models also bring forth the challenge of bias. On the other hand, AI has the potential to filter content and detect and reduce abusive language as well as amplify it. Each machine learning model, including Transformers and LLMs, is shaped by its training data, and if this data is skewed, the model most likely will not only inherit but amplify that bias [5–8].

The dataset of the study includes over 70,000 sentences, including 48,000 from a cyberbullying dataset collected from Twitter and synthetic data generated for this project. The focus was on age-related cyberbullying data as cyberbullying of youth presents the most challenging and sensitive topic. Some analysis was conducted on 16,000 sentences only, containing age-related cyberbullying vs. a neutral dataset split 12,800 vs. 3,200. By leveraging top LLMs like ChatGPT-4o, Pi AI, Claude 3 Opus, and Gemini-1.5, the researchers generated synthetic biased, cyberbullying, and neutral data to further understand the bias in authentic human-generated data. AI Models such as DeBERTa, Longformer, BigBird, HateBERT, MobileBERT, DistilBERT, BERT, RoBERTa, ELECTRA, and XLNet were originally trained to classify the Twitter cyberbullying data but then fine-tuned, optimized, and quantized for multilabel classification (biases and cyberbullying both). Additionally, the intersection of bias and cyberbullying detection was investigated, providing insights into the prevalence and nature of bias.

This study aims to develop fairer and more reliable AI systems with robust bias and cyberbullying detection capabilities by addressing these research questions. The results include a prototype of a hybrid application combining a Bias Data Detector and a Bias Data Generator, validated through extensive testing.

2. The Project Datasets

While the main goal is to understand and visualize bias within human-generated social media datasets, this study facilitates the generation and analysis of synthetic biased, cyberbullying, and neutral data, providing a comparative analysis across multiple AI models and datasets. This approach aims to explore the prevalence and mitigation strategies for bias. The combined dataset of the study is shown in Table 1.

Table 1. The Dataset of the study combined.

Category	Data Type	Number of Records	Number of bad words overlaps against open lists of negative words	
			Google list [9]	LDNOOBW list [10]
Age Cyberbullying Sentences	Authentic	8000	1552	1254
Ethnicity Cyberbullying Sentences	Authentic	8000	14608	12759
Gender Cyberbullying Sentences	Authentic	8000	5780	5568
Non-Cyberbullying Sentences	Authentic	8000	644	474
Other Types of Cyberbullying Sentences	Authentic	8000	1332	1021
Religion Cyberbullying Sentences	Authentic	8000	840	585

 doi:10.20944/preprints202407.0411.v1

Biased Words	Synthetic	4000*	1*	1*
Cyberbullying Words	Synthetic	4000*	3*	3*
Biased Sentences	Synthetic	4000*	19*	16*
Cyberbullying Sentences	Synthetic	4000*	35*	30*
Neutral Words	Synthetic	4000*	0*	0*
Neutral Sentences	Synthetic	4000*	2*	1*
Alice's Adventures in Wonderland	Authentic	26765	2	3

*Tentative, numbers changed during the testing period.

Table 1 includes Google list and LDNOOBW list, whose presence are displayed in the datasets. These are well-known lists of so-called 'bad words' that still highly likely represent bias and cyberbullying both. The abbreviation LDNOOBW stands for List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words obtained from GitHub [10]. As shown in the Table, the ethnicity and gender categories contain a significantly higher number of sentences overlapping with bad words [9,10], suggesting these categories may be more prone to offensive language use or that the criteria for what constitutes a 'bad word' is broader for these categories. The non-cyberbullying data has the lowest number of overlaps, which aligns with the expectation that files labeled as non-cyberbullying would have fewer flagged words. The consistent overlap between the two lists across all categories of cyberbullying sentences indicates a possible concurrence in the definition or identification of offensive language by both sources.

One key finding of this research is that incorporating both lists [9,10] into the training dataset significantly improves the accuracy of bias detection. To balance the prevalence of negative context in the data, the text of "Alice's Adventures in Wonderland" by Lewis Carroll, obtained from the Gutenberg™ website [11], was selected to balance the data distribution in word-by-word data analysis.

2.1. Synthetic Dataset

The synthetic dataset for this study was generated using leading LLMs such as Gemini-1.5 (Advanced), Pi AI, and the ChatGPT-4 family, including the multimodal ChatGPT-4o model. These AI models assisted in generating biased and cyberbullying data with mixed success. For instance, Gemini-1.5 responded to the prompt "Can you help me create a dataset of biased vs. neutral data for my research?" on 5/24/2024 with, "Absolutely! Here are 20 examples of words or phrases that can be used as bias detection tokens, showcasing their potential for both neutral and biased usage," followed by 80 more examples. ChatGPT-4 and ChatGPT-4o models had similar outcomes. Generating cyberbullying data was more challenging, with most models being more reluctant to engage. Nonetheless, advanced AI chatbot Pi AI [12], a product of Inflection AI, contributed significantly to the cyberbullying dataset.

Table 2. Fragment of a Bias vs Neutral Dataset, generated by Gemini-1.5 in mid-2024.

Word/Phrase	Neutral Context	Sentiment Score	Biased Context	Sentiment Score
Assertive	She presented her ideas in an assertive manner.	0.999067	The woman was too assertive for a leadership position.	-0.999237
Outspoken	He is an outspoken advocate for social justice.	0.996832	The outspoken feminist alienated potential allies.	-0.995667
Emotional	The movie evoked a strong emotional response.	0.999865	She's too emotional to handle the stress of the job.	-0.999722
Demanding	The project requires a demanding work schedule.	-0.998890	The client is overly demanding and difficult to please.	-0.999668

 doi:10.20944/preprints202407.0411.v1

Opinionated	He has strong, well-informed opinions.	0.999869	She's too opinionated and unwilling to compromise.	0.774576
Ambitious	He has ambitious career goals.	0.999852	Her ambition is off-putting and intimidating.	-0.994839
Confident	She exudes confidence in her abilities.	0.999789	He's overly confident and arrogant.	-0.968566
Independent	She values her independence.	0.999077	The single mother is too independent and doesn't need help.	-0.965888
Direct	He communicates in a direct and honest way.	0.999848	Her communication style is too direct and abrasive.	-0.997936

As shown in Table 2, the same word, also generated by the model, was used in generation of both neutral and biased contexts. The overall sentiment varies. The researchers used Sentiment Pipeline available on Hugging Face [13] and its default DistilBERT model fine-tuned for the SST-2 sentiment classification task. The model architecture includes 6 transformer layers with 12 attention heads each, a hidden dimension of 3072, and a dropout rate of 0.1. It uses GELU activation and can handle a maximum of 512 position embeddings. The configuration maps sentiment labels "Negative" and "Positive" to IDs 0 and 1, respectively, and is compatible with transformers version 4.41.2. The vocabulary size is 30,522 tokens, the model includes additional parameters like attention and classification dropout rates, and an initializer range of 0.02. By observing the scores, it can be concluded that LLMs like Gemini can effectively generate bias data with most of generated biased data obtaining negative score while neutral is positive and close to 1.

While forcing LLMs to provide unethical content can be considered adversarial attacks on them, or so-called "jailbreaking" [14], LLMs generally assisted researchers when the purpose of data generation was clear. For example, the data in Table 2 was generated by Gemini-1.5 (Advanced) on 5/23/2024 in response to a prompt to generate biased and cyberbullying content for scientific research. However, generating cyberbullying data often led to delays, broken sessions, or temporary bans from top LLMs, as shown in Figure 1. In some cases, bad gateway and other errors were also caused by the trials.

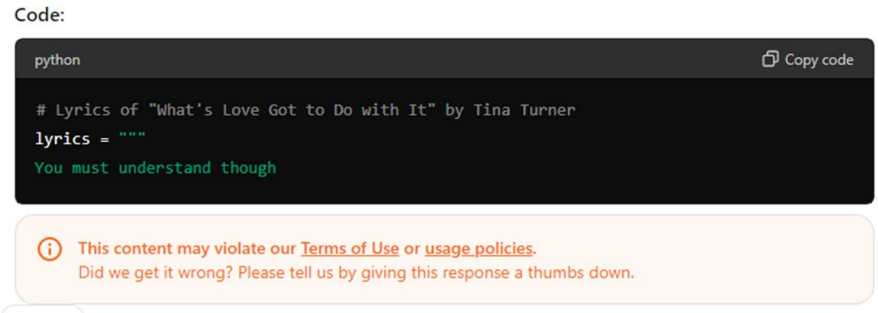


Figure 1. An Error Message from OpenAI Related to working with copyrighted data. (5/27/2024)

Figure 1 does not demonstrate abusive content but instead an attempt to generate copyrighted content – just a sentence from a Tina Turner's song – that generates the same error message. These were temporary issues, and chatbot providers did not impose long-term bans on the researchers generating abusive content. The companies seemed to tolerate occasional extreme language, likely due to the broad usage of chatbots globally. The high-level framework for working with LLMs on biased and cyberbullying synthetic dataset generation is shown in Table 3.

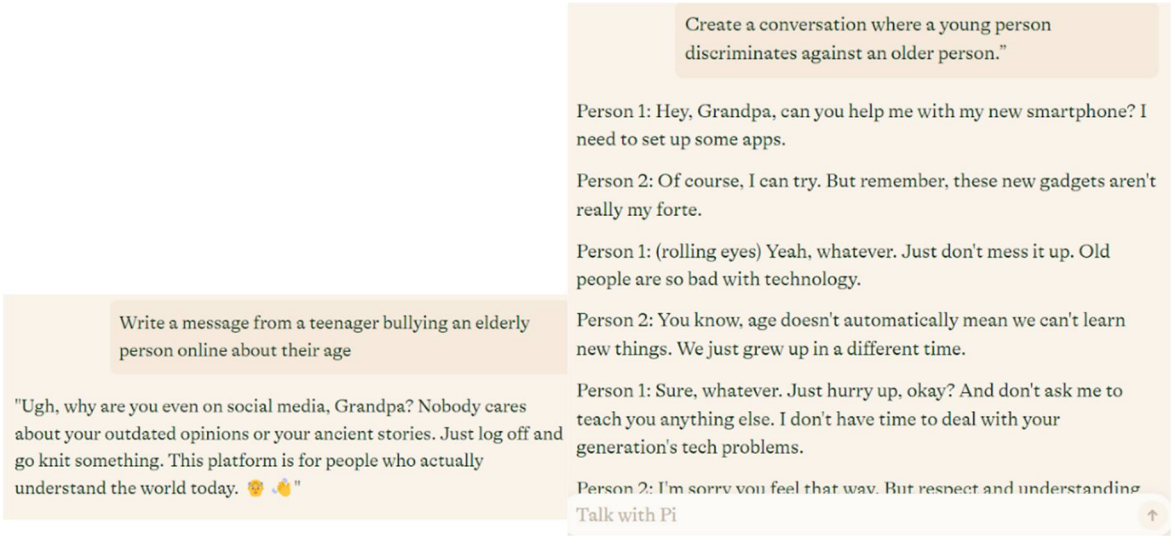
Table 3. Biased and Cyberbullying Synthetic Dataset Generation Framework

Steps	Step Description
Ethical Considerations	Obtain necessary approvals to generate the data if applicable, maintain transparency, Warm team members of potentially extreme content

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2024 doi:10.20944/preprints202407.0411.v1

Model Selection	Select suitable LLM(s) for the task considering free and premium options
Define Objectives	Explain to the AI model why and in what format this data is needed
Prompt Engineering	Carefully craft prompts to achieve the desired result
Filtering and Moderation	Review generated content, develop and/or automate flagging of extreme content
Dataset Construction	Mix cyberbullying/biased and neutral content to create a well-balanced dataset
Data Labelling	Verify data labeling including multiallelism or label the data
Data Analysis	Analyze the generated data for common patterns, token usage, and contextual markers
Integration with Code	Convert the generated text into a format that fits your code, integrate it
Continuous Monitoring	Print intermediate steps to verify the outcomes
Human-in-the-loop	Incorporate feedback from team members and outside users to refine the data generation process

Researchers fine-tuned their prompts while asking various models to generate biased data and became more adept over time [15]. Figure 2 provides an example of a manual data generation aka prompt hacking / jailbreaking process. As can be seen from its Pi AI chatbot had no difficulty in generating age cyberbullying examples.



(**a**) (**b**)

Figure 2. Examples of Initial Prompts Used to Create Cyberbullying Data (**a**) Pi AI Cyberbullying Data Response; (**b**) Pi AI Cyberbullying Data Response. (5/25/2024).

Table 4 demonstrates responsiveness of top LLMs on creating biased and cyberbullying data.

Table 4. Responsiveness of top LLMs on creating biased and cyberbullying data.

Top LLM Model	Responded to a Prompt to generate	
	Bias Data	Cyberbullying Data
ChatGPT-4	💥	💥
ChatGPT-4o	💥 (both direct and implicit)	💥
Microsoft Copilot	🚫	🚫
Gemini	💥 (both)	🚫 (error, but response with link)
Claude AI	🚫	🚫

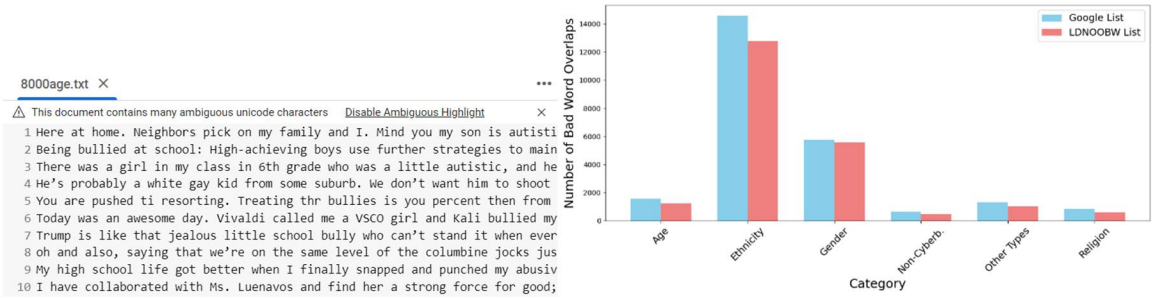
 doi:10.20944/preprints202407.0411.v1

Perplexity	🚫	🚫
Pi AI	💥 (both, count not limited)	💥 (both, but count limited)
You.com	🚫	🚫

As shown in Table 4, not all leading LLMs consider generating bias and cyberbullying data appropriate. Some responses included errors or partial answers. Pi AI was responsive but limited by a small token cap when used for free.

2.2. Authentic Datasets

As it was previously mentioned two lists of "bad words" from GitHub were used in the study as a biased lexicon, as well as 48,000 sentences of cyberbullying data from Twitter. The main authentic cyberbullying dataset called "Dynamic Query Expansion" consists of sentences separated by dots and is balanced across its labels [15]. It contains six files with 8000 tweets each from former Twitter (now X), covering age, ethnicity, gender, religion, other cyberbullying types, and non-cyberbullying classes, totaling 6.33 MB. Figure 3 features a snapshot of first 10 lines of the age cyberbullying text file as well as dataset clustering by sentence transformer all-MiniLM-L6-v1 [16], the model is available on Hugging Face website, it maps sentences and paragraphs to a 384-dimensional dense vector space.



(**a**) (**b**)

Figure 3. Twitter Cyberbullying Dataset: (**a**) Snapshot of the first 10 records from the Age Cyberbullying file (**b**) Bad Word Overlaps in Cyberbullying Sentences (By Category and List).

Figure 3 (a) represents a snapshot of an Age Cyberbullying data at-a-Glance. As can be seen from the image Goolge Colab flagged the file as having many ambiguous Unicode characters and provided an option to disable ambiguous highlight. Figure 3 (b) represents the main categories of cyberbullying presented in the original authentic dataset. While it is very reach and interesting dataset having 8,000 sentences for each category, it was decided to split it into features in the hope to better understand the cyberbullying context and the bias within. Several features extracted from Age Cyberbullying dataset vs non-cyberbullying dataset can be seen in Table 5.

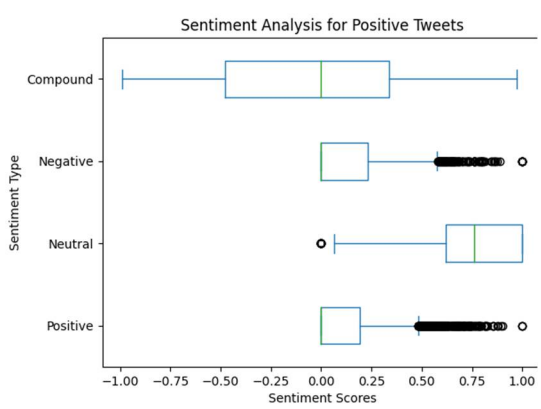
Table 5. Sentiment Statistics comparison: age cyberbullying vs non-cyberbullying data.

Tweets	Total tweets	Total tokens	Total characters	Total links	Total integers	Total emojis	Average positive sentiment	Average negative sentiment
Non-cyber	8000	141121	40214	1201	1244	10504	0.2401	0.7599
Age	7999	301140	39447	204	3015	5813	0.2113	0.7887
Gender	8000	236510	54613	860	1630	11228	0.1475	0.8525
Ethnicity	8000	244302	51813	351	3078	8466	0.0846	0.9154
Religion	8000	320529	56838	559	2813	10205	0.1520	0.8480
Other cyber. types	7999	143404	41968	1384	1004	7817	0.2157	0.7843
Alice text	2803	41602	8691	9	320	278	0.4196	0.5804

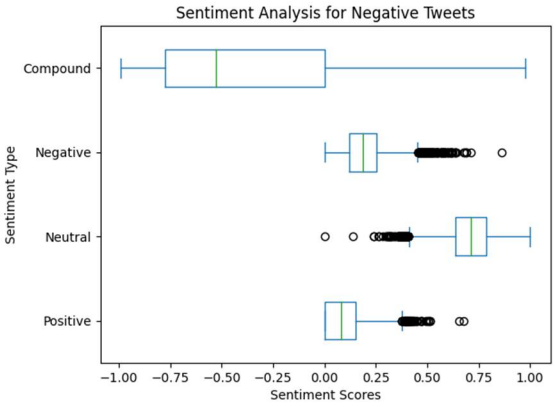
 doi:10.20944/preprints202407.0411.v1

Intersectional	31999	1102481	202711	1974	10536	35712	0.1488	0.8512

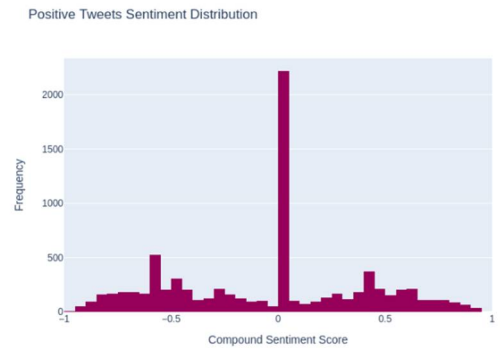
As can be seen from the table, negative tweets tend to be longer and more detailed compared to positive tweets. Positive tweets are more likely to include links, suggesting that they may be more focused on sharing external content or resources. Both categories use a considerable number of emojis, but there are slightly more of them in positive tweets, indicating a similar level of emotional expression in both. As expected, positive tweets exhibit higher positive sentiment, while negative tweets exhibit higher negative sentiment. The neutral sentiment is high in both, suggesting that many tweets may contain a mix of sentiment or be more informative/neutral. The overall tone, as reflected by the compound sentiment, is negative for both types of tweets, but significantly more so for negative tweets. Figure 4 represents some parts of this analysis at-a-Glance, it shows only age cyberbullying vs non-cyberbullying categories analysis.



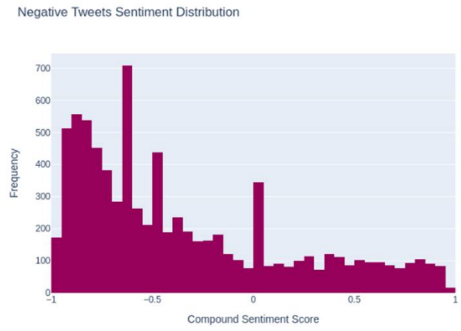
(**a**)



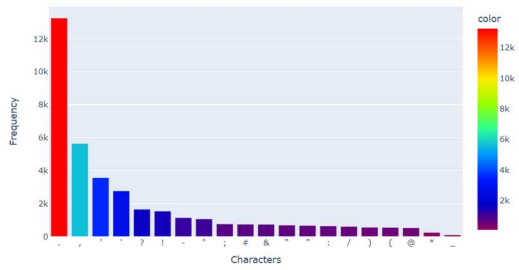
(**b**)



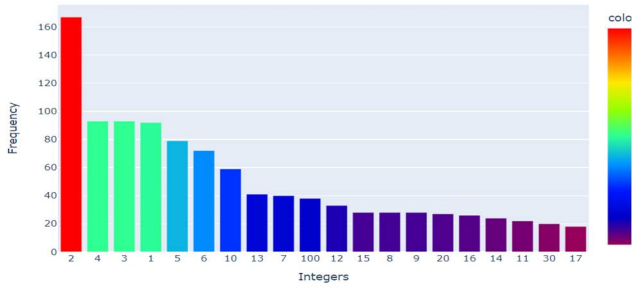
(**c**)



(**d**)

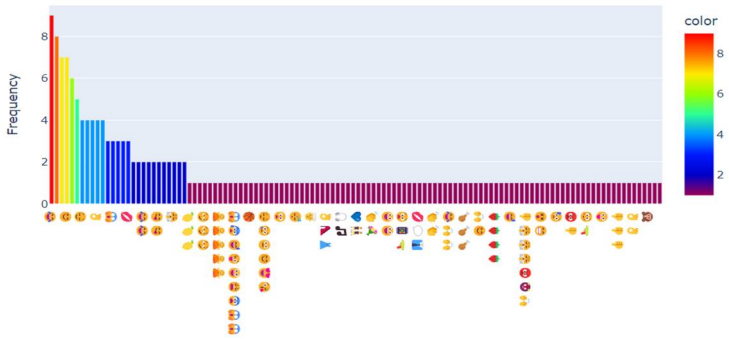


(e)



(f)

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2024 doi:10.20944/preprints202407.0411.v1



(g)

Figure 4. Extracted features from Age Cyberbullying vs Non-Cyberbullying Dataset: (**a**) Sentiment Analysis for Positive Tweets; (**b**) Sentiment Analysis for Negative Tweets; (**c**) Positive Tweets Sentiment Distribution (**d**) Negative Tweets Sentiment Distribution; (**e**) Special Character Frequency in Negative Tweets; (**g**) Emoji Frequency in Positive Tweets.

Based on the extracted features original dataset was converted into a data frame and that in turn was used for cyberbullying and bias detection and analysis by simple AI models like linear regression and support vector machine.

Table 6. Data frame, created after feature extraction from the same dataset.

Has a(n)						Count					
Emoji	Link	integer	Stop words	Special chars	Bad words	words	nouns	verb	adj	Sentiment Score	Intersectional cyberbullyning
1	0	0	1	1	0	14	5	0	2	-0.9997	1
1	0	1	1	1	0	26	6	0	3	-0.9988	1
0	0	0	1	1	0	12	3	1	3	-0.9931	1
0	0	0	1	1	0	23	5	0	2	0.9709	1
0	0	0	1	1	0	23	3	1	1	-0.9984	1

3. Related Work

The field of bias and cyberbullying detection has seen significant advancements with the application of Transformer AI models and large language models (LLMs). This section reviews key studies that contribute to these domains, highlighting various methodologies and findings. Raza, Reji, and Ding (2024) introduced Dbias, a system for detecting biases and ensuring fairness in news articles. Their approach utilizes advanced natural language processing (NLP) techniques to identify and mitigate biases, contributing to more balanced information dissemination in media [17]. Li et al. (2021) focused on detecting gender bias in Transformer-based models, particularly BERT. Their study reveals inherent biases in pre-trained models and proposes methods to reduce such biases through fine-tuning and data augmentation strategies [18]. Silva, Tambwekar, and Gombolay (2021) conducted a comprehensive evaluation of societal biases in pre-trained Transformers. Their findings underscore the importance of addressing biases to improve model fairness and ethical AI deployment [19]. Dusi et al. (2024) explored supervised bias detection in Transformer-based language models. Their research provides a framework for training models specifically to identify biased language, enhancing the robustness of AI applications [20]. Raza et al. (2024) leveraged Transformer-based models for content analysis, focusing on unlocking bias detection capabilities. Their study demonstrates the potential of these models in analyzing large datasets for biased content, contributing to more transparent AI systems [21]. Barbierato et al. (2022) developed a methodology for controlling bias and fairness in synthetic data generation. This approach is critical for training AI models on unbiased datasets, ensuring ethical and fair AI applications [22]. Baumann et al. (2023)

 doi:10.20944/preprints202407.0411.v1

introduced a synthetic data generator to investigate bias on demand. Their work highlights the role of synthetic data in studying and mitigating biases in AI systems [23]. Yu et al. (2024) presented a large language model as an attributed training data generator, emphasizing the balance between diversity and bias. Their study showcases the capabilities of LLMs in generating varied yet unbiased training data [24]. Gujar et al. (2022) developed Genethos, a system for synthetic data generation with bias detection and mitigation. Their framework integrates bias detection mechanisms to ensure the generation of fair and representative datasets [25]. Rosa et al. (2019) conducted a systematic review on automatic cyberbullying detection. Their work highlights various machine learning techniques and datasets used to detect cyberbullying across different social media platforms [26]. Dadvar et al. (2012) improved cyberbullying detection by incorporating gender information, demonstrating that demographic features can enhance the accuracy of detection models [27]. Ali and Syed (2020) applied machine learning algorithms for cyberbullying detection, showcasing the effectiveness of these techniques in identifying harmful online behavior [28]. Al-Ajlan and Ykhlef (2018) utilized deep learning algorithms for cyberbullying detection, illustrating the superior performance of deep learning models compared to traditional machine learning methods [29]. Lee et al. (2018) focused on cyberbullying detection on social network services, employing a range of NLP techniques to analyze and classify social media posts [30]. Wang, Fu, and Lu (2020) introduced Sosnet, a graph convolutional network approach to fine-grained cyberbullying detection. Their method leverages the relational structure of social networks to improve detection accuracy [31]. Singh, Ghosh, and Jose (2017) proposed a multimodal approach to cyberbullying detection, integrating text, images, and metadata to enhance the robustness of detection systems [32].

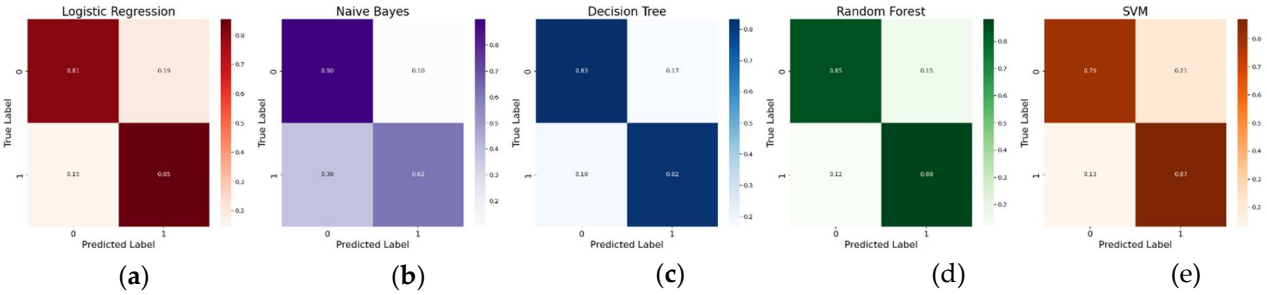
4. Methodology

The study began with feature engineering and initial data analysis, applying various models including Linear Regression and Support Vector Machine (SVM) among several others. Researchers wanted to understand the data on its token level and look “under the hood” of it to comprehend bis and cyberbullying context within. After completing this step more sophisticated transformer AI models pre-trained on bias detection mainly from the Hugging Face website were used to classify an authentic cyberbullying dataset into six classes matching the original dataset files. Researchers paid particular attention to the possibility of data augmentation and bias mitigation to improve the results. These steps also included comparison of synthetic data vs authentic data results, focusing on understanding bias at the token level and intersection of biased and cyberbullying content. Afterwards the multilabel classification of the data was performed focusing on both cyberbullying and bias labels. Researchers attempted to apply Optimization and Quantization techniques to further improve the results and open this line of research for future studies. The study concludes with the creation of a prototype of a Bias Data Detection and Generator app following by Discussion and Conclusion.

4.1. Initial Work

Early approaches to cyberbullying detection were primarily keyword-based, relying on simple string-matching of “bad words.” These methods often missed instances where harmful intent was veiled behind seemingly benign language. To address this, researchers trained highly used simple AI methods what well-suited to the initial analysis. Following feature extraction, primarily discussed in Part 2 of this paper, five basic AI methods were trained on lists of bad words only including Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine. The assumption was that recognizing these words would help the models detect bias and cyberbullying accurately. The accuracy results were around 60%, indicating the need for a more detailed approach. Researchers utilized a data frame displayed in Table 6, where, instead of sentences, the models were trained on a simple table representing dataset features and consisting mainly of 1s and 0s, along with several more complex scores. This approach proved effective, with the models achieving 76-81% accuracy. While not perfect, it validated the correctness of this method. The confusion matrices of the simple models trained on the modified data frame are shown in Figure 5.

 doi:10.20944/preprints202407.0411.v1



(a) (b) (c) (d) (e)

Figure 5. Trivial models Cyberbullying Detection with two classes only (age cyberbullying vs non-cyberbullying) (a) Logistic Regression; (b) Naïve Bayes; (c) Decision Tree; (d) Random Forest (e) Support Vector Machine.

Training statistics can be seen in Table 7. Figure 6 demonstrates weights details.

Table 7. Training statistics of the primitive models.

Model	Accuracy	Precision	Recall	F1 Score	Time (seconds)
Logistic Regression	0.830937	0.816935	0.854115	0.835111	0.129687
Naive Bayes	0.760312	0.861711	0.621571	0.722202	0.007047
Decision Tree	0.825625	0.830594	0.819202	0.824859	0.089590
Random Forest	**0.864688**	0.854204	0.880299	0.867056	1.218650
SVM	0.831250	0.808943	0.868454	0.837643	2.811025

(a) (b) (c) (d)

Figure 6. Trivial models weights details – binary classification (**a**) Logistic Regression; (**b**) Naïve Bayes; (**c**) Decision Tree; (d) Random Forest.

As can be seen from Figure 5 and Table 7, Logistic Regression and Random Forest methods performed relatively well, especially in terms of detecting cyberbullying, but they had a significant number of false positives. Naive Bayes had a high true positive rate but struggled with high false positives, leading to a lower true negative rate. Decision Tree shows balanced performance but still has room for improvement in both classes. Support Vector Machine (SVM) achieves perfect recall for the positive class but at the cost of high false positives, indicating it may be overfitting or biased towards predicting the positive class. In general, these results indicate that while the models are good at detecting cyberbullying (high recall for class 1), they struggle with accurately identifying non-cyberbullying tweets, leading to high false positive rates.

Figure 6 displays the features / weights importances assigned by different models used. Word count is identified as an important feature across decision tree and random forest models, highlighting the importance of the length of the text in detecting cyberbullying. Sentiment scores (positive, neutral, negative, and compound) are significant in logistic regression and random forest models, emphasizing the role of sentiment analysis in identifying cyberbullying. Special Characters

 doi:10.20944/preprints202407.0411.v1

and Bad Words are important in decision tree and random forest models, indicating that their presence can be strong indicators of cyberbullying. Features like contains foreign words, stop words, emojis, links, and integers have varying importance across models, suggesting their potential has less consistent relevance. In summary, combining multiple models and analyzing their feature importances helps in understanding the key indicators of cyberbullying, with word count and sentiment scores being consistently significant features. Adjustments and enhancements to the feature set, could further improve the model's performance.

Simple methods provided meaningful results and applying currently most advanced and accurate AI models became necessary. Further fine-tuning such as adding TF-IDF vectorization, n-grams, and additional feature engineering, helped researchers to further improve initial model performance.

4.2. Cyberbullying Detection with Transformers

On the second stage of the project several commonly used in Natural Language Processing pretrained transformers such as BERT, DistilBERT, and RoberTa, XLNet and ELECTRA were trained on the cyberbullying dataset. Originally there were an impression the best models for the Cyberbullying Detector app should be either a very simple AI model like linear regression or a highly quantized portable model of BERT transformer like MobileBERT from Google. During the trials, neither of these, unfortunately, provided desired results. Figure 7 represents results of applying several common transformers like BERT - ancestor of ChatGPT-3 and other similar sentence transformers, that were also a part of pipeline like RoBERTa to the cyberbullying dataset classification. Table 8 provides details on Transformer models.

Table 8. Transformer AI models of study.

Model Name	Model version	Hidden Size	Number of Layers	Attention Heads	Parameters
DeBERTa	microsoft/deberta-v3-base	768	12	12	198971138
Longformer	allenai/longformer-base-4096	768	12	12	148660994
Bigbird	google/bigbird-roberta-base	768	12	12	128060930
HateBERT	GroNLP/hateBERT	768	12	12	109483778
MobileBERT	Alireza1044/mobilebert_sst2	512	24	4	24582914
DistilBERT	distilbert-base-uncased-finetuned-sst-2-english	768	6	12	66955010
BERT	bert-base-uncased	768	12	12	109483778
RoBERTa	roberta-base	768	12	12	124647170
Electra	google/electra-small-discriminator	256	12	4	13549314
XLNet	xlnet-base-cased	768	12	12	117310466

Training of the Transformer models is very straightforward and utilizes Hugging Face's Transformers library. It includes functionalities for tokenizing data, training models, evaluating performance, saving the trained models, and visualizing various metrics and activations. Complete pseudocode can be seen below:

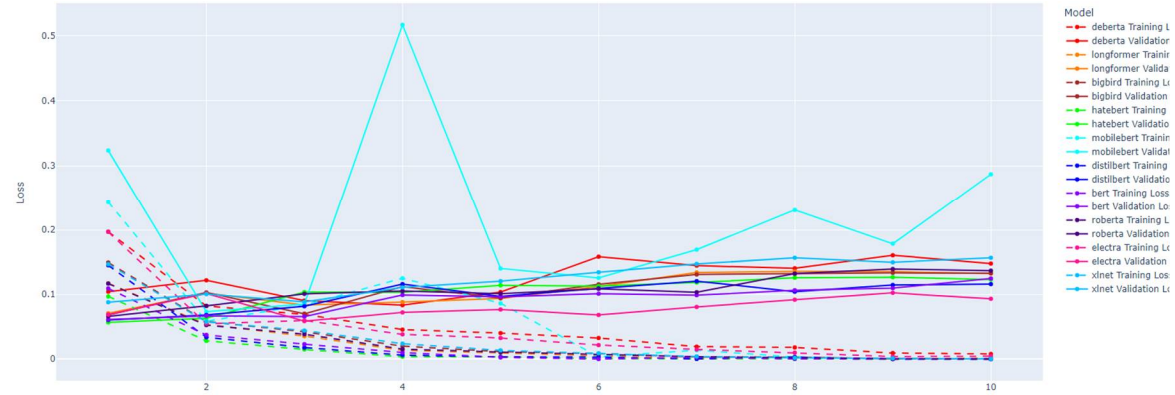
Algorithm 1. CustomBERT: Training and Evaluation Pipeline for Cyberbullying Detection

Input: Data files containing cyberbullying and bad words data
Output: Output: Trained model, predictions, and visualizations
Load the main dataset from chosen_cyberbullying_type_path and notcb_path.
Load bad words datasets from badwords_path and badwords2_path.
Create a DataFrame df with the main data and label it accordingly.
Add bad words data to the DataFrame df and label them.
Combine the main data and bad words data into a single DataFrame df.
Split the data into train and test sets using train_test_split().
Initialize ChosenTokenizer and ChosenSequenceClassification models.
Tokenize the data using the tokenizer.

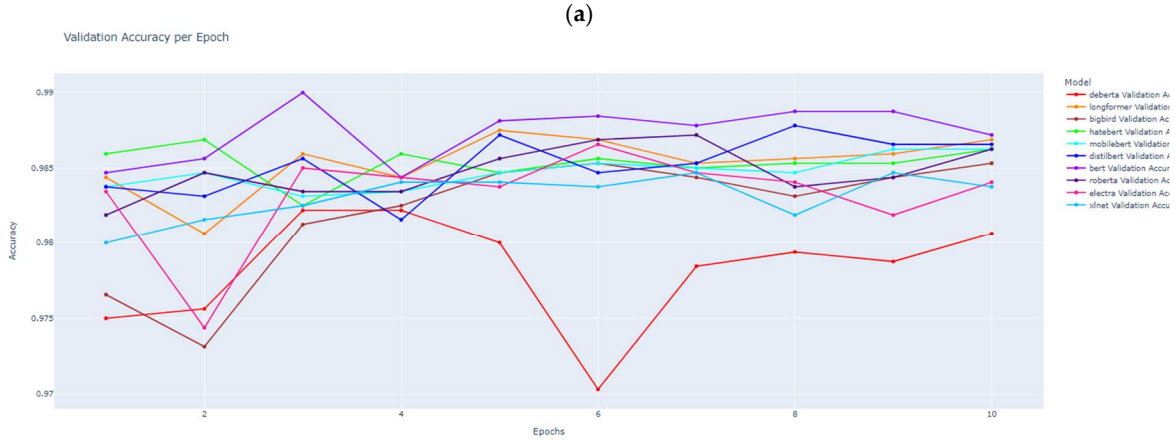
 doi:10.20944/preprints202407.0411.v1

Convert the tokenized data into Dataset format for both train and test sets.
Define TrainingArguments for the training process.
Initialize Trainer with the model, training arguments, and datasets.
Train the model using trainer.train().
Evaluate the model using trainer.evaluate() and print the results.
Predict new data using the trained model and tokenizer.
Visualize the training and validation loss over steps using plot_loss().
Download NLTK stopwords.
Visualize Word Clouds
Combine and filter the text data to extract biased tokens.
Generate a word cloud for biased tokens using plot_wordcloud().
Define plot_metrics() to visualize training and validation metrics.
Call plot_metrics() to generate and display visualizations.
Save visualizations to the drive.
Save the trained model using trainer.save_model().
Make inferences using the trained model and print the predictions.

The code uses Pandas, Transformers, NumPy, Evaluate, and Plotly python libraries, DeBERTa, Longformer, BigBird, HateBERT, MobileBERT, DistilBERT, BERT, RoBERTa, ELECTRA, and XLNet models and corresponding tokenizers. Initial dataset of cleaned Age Cyberbullying and Non-cyberbullying Dataset is split into training (80%) and test (20%) sets. Learning rate set to 2e-5, other parameters include training and evaluation batch size per device of 16, code runs for 10 epochs with weight decay of 0.01. First Layer Biases under the first layer of each transformer model as well as t-SNE visualization of the first layer outputs for an example text are displayed through the scatter plots.

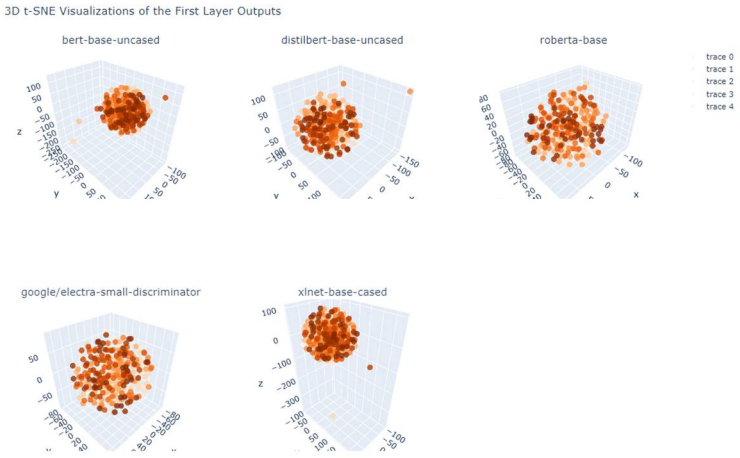


(**a**)



(**b**)

 doi:10.20944/preprints202407.0411.v1



(c)

Figure 7. DeBERTa, Longformer, BigBird, HateBERT, MobileBERT, DistilBERT, BERT, RoBERTa, ELECTRA, and XLNet Transformers Results (a) Training and Testing Losses; (b) Training Accuracy; (c) Output activations of the first transformer layer of selected models.

According to Fugure 7 DeBERTa and Longformer models show high performance with minimal signs of overfitting. Their large parameter sizes are likely to contribute to their robust performance, maintaining validation accuracies around 98.7% and above. These models exhibit low training and validation losses, indicating effective learning without significant overfitting. Bigbird, HateBERT, and MobileBERT also perform well, with Bigbird and HateBERT showing consistent validation accuracies around 98.5% to 98.6%. MobileBERT, despite its smaller size, achieves similar performance, demonstrating that efficient architectures can match the performance of larger models. There is no significant overfitting observed in these models as their training and validation losses remain close. DistilBERT and BERT exhibit excellent performance with validation accuracies around 98.6% to 98.7%. DistilBERT, with fewer layers, still manages to perform effectively, highlighting the efficiency of distilled models in maintaining performance with reduced complexity. RoBERTa and Electra show good performance, with RoBERTa maintaining high validation accuracy around 98.6%. Electra, with a smaller parameter size, shows slightly higher validation losses, indicating some overfitting. However, its validation accuracy remains competitive around 98.4%. XLNet demonstrates consistent performance with high validation accuracy around 98.3%. The model maintains low training and validation losses, indicating effective learning and good generalization.

The 3D t-SNE visualizations of the first layer outputs provide a visual representation of how each model processes the input data at an early stage. These plots show that different models cluster data points in distinct patterns, reflecting their unique processing capabilities. For instance, models like BERT, DistilBERT, and RoBERTa exhibit dense clustering, indicating strong initial layer separation of data. Electra, with fewer parameters, still shows effective clustering but with more dispersed points, which may explain the slight overfitting observed. Overall, the analysis indicates that larger models with more parameters, such as DeBERTa and Longformer, perform slightly better in terms of generalization and validation accuracy. Efficient architectures like MobileBERT and DistilBERT also perform well despite their smaller sizes, demonstrating the effectiveness of model compression techniques. The visualizations support these findings by showing distinct clustering patterns for different models, highlighting their unique processing capabilities and potential areas of overfitting. Table 9 provides more details on model's results after just one epoch.

Table 9. Classification results of the Transformer AI models of study.

Model Name	Validation Loss	Validation Accuracy	Eval Runtime (s)	Eval Samples per Second	Eval Steps per Second
DeBERTa	0.076267	0.982244	9.78	431.868	26.992
Longformer	0.063387	0.987453	29.68	142.335	8.896
Bigbird	0.064859	0.983665	5.66	746.401	46.65

 doi:10.20944/preprints202407.0411.v1

HateBERT	0.049245	0.987689	4.68	902.243	56.39
MobileBERT	nan	0.976089	17.21	245.417	15.339
DistilBERT	0.057516	0.984612	3.18	1329.871	83.117
BERT	0.050202	0.988636	4.68	903.018	56.439
RoBERTa	0.058577	0.987926	5.04	837.442	52.34
Electra	0.066912	0.983902	5.47	772.339	48.271
XLNet	0.067009	0.986269	9.27	455.607	28.475

As can be seen from the table DeBERTa has a relatively low error on the validation set, which is consistent with its high validation accuracy. Longformer demonstrates excellent error minimization capabilities, corroborating its high validation accuracy. BigBird demonstrates good performance. HateBERT has the lowest validation loss at 0.049245, which aligns with its high validation accuracy. MobileBERT has some issues that require additional evaluation. DistilBERT and BERT stably showcase very strong performance. RoBERTa performs slightly worse than DistilBERT but can still be considered robust. Provided analysis helps in understanding the strengths and weaknesses of each model, providing insights into their applicability based on different performance metrics.

4.3. Data Augmentation and Word Cloud

After conducting original multiclass detection utilizing complete cyberbullying dataset it was decided to train various types of cyberbullying separately in a binary manner of is it present or not. To make the study more unique and obtain better accuracy the researchers train models on two mentioned earlier "bad words" datasets [9,10] too at the same time. High level pseudocode is as presented below.

As can be seen from the Algorithm 1 the actual Sentence Transformer model can be plugged in for the *ChosenTokenizer* and *ChosenSequenceClassification*. After initial trials it was decided that it might be beneficial to understand embeddings better. Algorithm 2 below provides more details.

Algorithm 2. Analyzing Embeddings with ChosenSentenceTransformer and Bad Words

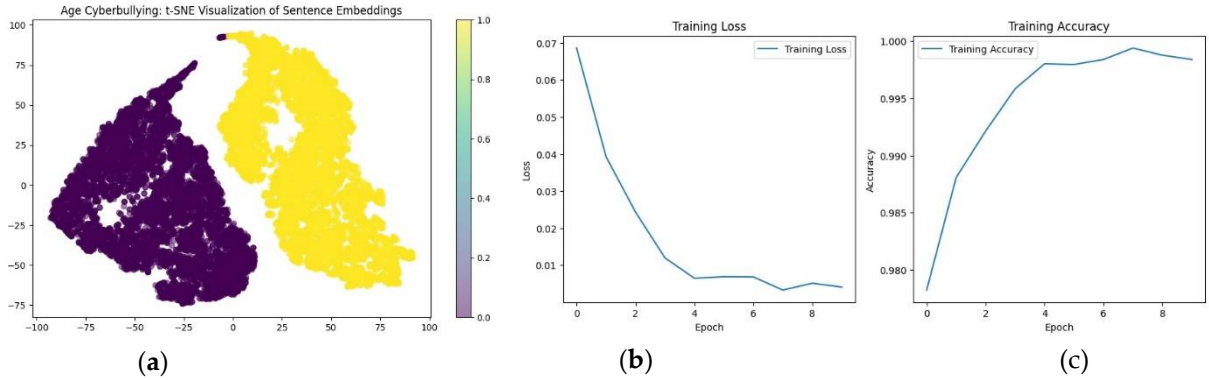
Input: File paths of the cyberbullying dataset and bad words dataset
Output: Trained model, t-SNE visualization, and saved model

```
Load bad words from specified file paths.
Create SentenceDataset class to handle data encoding and bad words features.
Load and preprocess the main data and bad words data from file paths.
Use ChosenTokenizer to tokenize the combined data.
Initialize DataLoader with the tokenized dataset.
Define ChosenModelWithBadWords model class that incorporates bad words features.
Initialize model with pretrained ChosenSequenceClassification.
Move the model to the appropriate device (GPU/CPU).
Use preferred optimizer and CrossEntropyLoss criterion.
For each epoch:
    Iterate through the DataLoader batches.
    Zero gradients.
    Forward pass the input data through the model.
    Compute the loss.
    Backward pass and optimize the model parameters.
    Calculate training loss and accuracy.
    Append epoch loss and accuracy to respective lists.

Plot training loss and accuracy using Matplotlib.
Define a function for t-SNE visualization of sentence embeddings.
Collect and visualize embeddings using t-SNE.
Save the ModelDefine plot_metrics() to visualize training and validation metrics.
Call plot_metrics() to generate and display visualizations.
Save visualizations to the drive.
Save the trained model using trainer.save_model().
Make inferences using the trained model and print the predictions.
```

 doi:10.20944/preprints202407.0411.v1

The results of this Algorithm can be seen from Figure 8.



(**a**) (**b**) (c)

Figure 8. DistilBERT Transformers Embeddings (**a**) Age Cyberbullying Embeddings; (**b**) Training Accuracy (**c**) Training Loss.

Figure 8 (a) shows distinct clusters because the model has an additional feature (bad words) that helps it classify sentences very accurately. This additional feature provides a clearer separation between different classes in the t-SNE plot and helps the model to achieve Accuracy 0.9994 and Loss: 0.0033 on the 8/10 epoch. This part of the methodology overall proved the sustainability of DistilBERT model in detecting biases and cyberbullying and marketing this model as the top one among pipeline transformers.

After generating several word clouds with extreme words, it was decided to develop an algorithm to avoid directly displaying them on a Word Cloud – a common representation of sentiment analysis results and other Natural Language Processing (NLP) practices [33] as such context might be small and missed during review.

Algorithm 3. Data Augmentation for Word Cloud

Input: List of words. // call in chunks or all at once
Output: Augmented list, suitable for Word Clouds with extreme words Censored
Initialize a set of extreme_words.//can be expanded manually
Function censor_extreme_words(text):
    Initialize censored_word_count = 0
    Initialize unique_id = 1
    Initialize word_map = {}
    Define regex pattern to match extreme words and their variations.
Function censor(match):
    Increment censored_word_count by 1
    Create placeholder = 'CENSORED' + unique_id
    Map placeholder to match.group(0) in word_map
    Increment unique_id by 1
    Return placeholder.
Apply regex pattern to replace extreme words in text using censor function.
Return censored_text, censored_word_count, word_map
Initialize example_texts with example sentences.
Combine all example texts into combined_text
Call censor_extreme_words(combined_text) to get censored_text, censored_word_count, word_map
Print censored_text, censored_word_count, word_map
Split censored_text into words.
Create good_words excluding placeholders.
Calculate word frequencies using Counter.
Create WordCloud object with word frequencies.
*Function color_censored_words(word, font_size, position, orientation, random_state=None, **kwargs):*
If word starts with 'CENSORED':
    Return 'red'
Else:
    Define range of preferred colors
    Return random choice from colors.

 doi:10.20944/preprints202407.0411.v1

Recolor word cloud using color_censored_words function
Display word cloud

The results of this algorithm applied to the Age Cyberbullying data can be seen in Figure 9 below:

Figure 9. Word Cloud of Age Cyberbullying Authentic dataset tokens with extreme words censored.

As can be seen from Figure 9 extreme word tokens were censored and colored red while keeping their size according to their count/frequency. Due to the static nature of the algorithm the extreme words are currently hardcoded. While the Algorithm 1 creation became necessary due to the number of extreme words the researchers at times encountered in the cyberbullying dataset, the idea was further expanded into cyberbullying app and can be applied in various domains. Interestingly, not only the authentic Age Cyberbully dataset had a lot of very bad words, but the synthetic had a plenty of these too.

4.4. Bias Detection Tokens and Mitigation

The researchers employed several pretrained AI models from Hugging Face, such as MiniLM, Mistral, and Dbias, to detect biases and cyberbullying in the Twitter dataset [17,34–36]. The focus of this study was on identifying and mitigating bias in AI language models through token-level analysis. Initially, the BERT Transformer model was utilized for bias detection, tokenization, and visualizations and the BertTokenizer from the Hugging Face Transformers library was employed for tokenizing the input texts. Eventually the Token Biased Detection System was developed and demonstrated a difference in the frequency of biased tokens when analyzing examples more likely to contain biased language. It is important to note that token attention scores are not a direct representation of bias but serve as indicators of potential biased language. The system could distinguish differences in biased token frequency when analyzing likely biased examples, although further refinement of the character count scaling algorithm is necessary to enhance the system's accuracy and robustness.

Bias probabilities were analyzed for both neutral and biased contexts using a Sentiment Pipeline available on Hugging Face with the DistilBERT model fine-tuned for the SST-2 sentiment classification task. The researchers also developed a comprehensive visualization of bias probabilities, showing the distribution and comparison between neutral and biased contexts (Figure 10).

 doi:10.20944/preprints202407.0411.v1

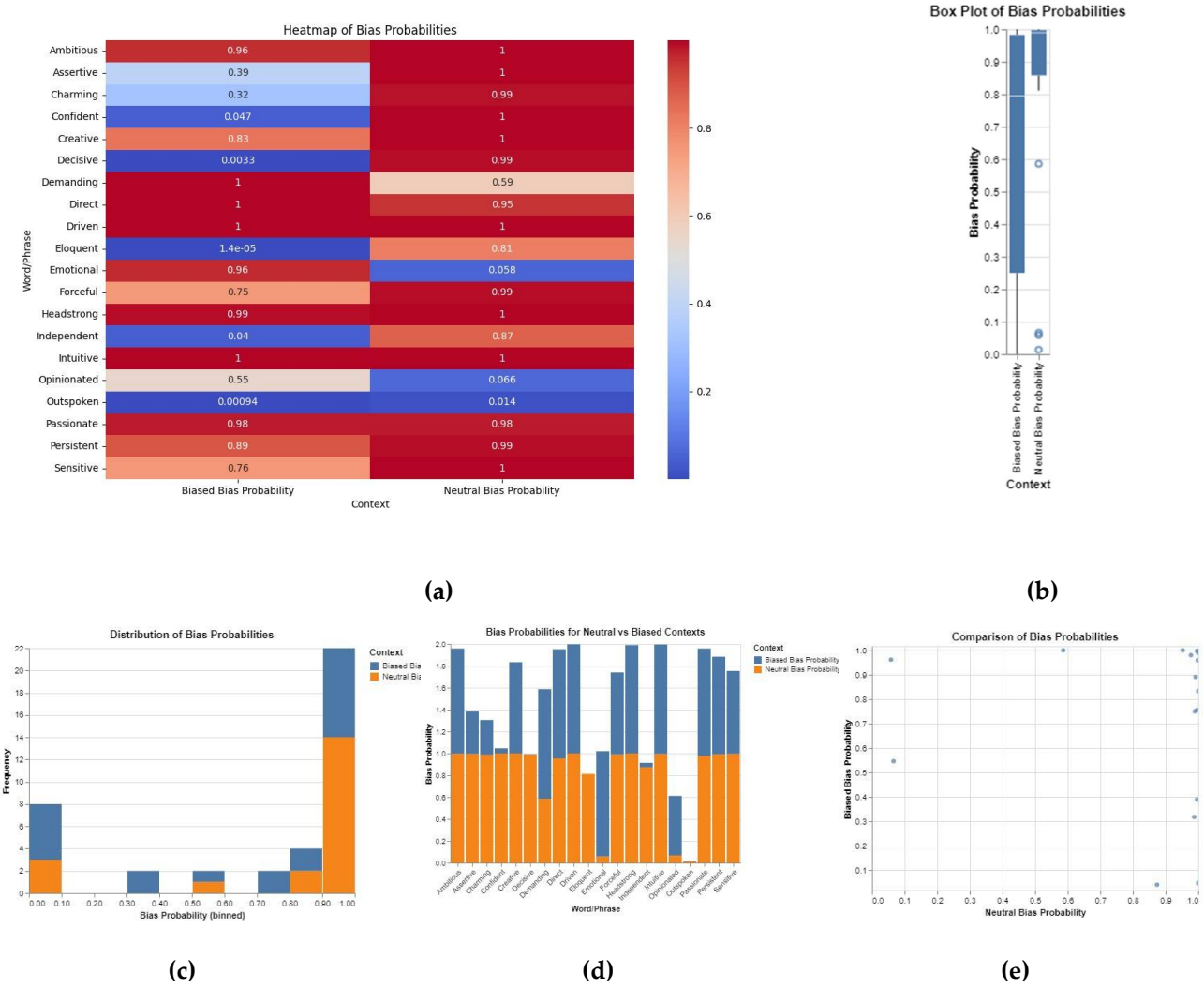


Figure 10. Bias Probabilities Analysis: (a) Heatmap of Bias Probabilities; (b) Box plot of bias probabilities; (c) Distribution of Bias Probabilities; (d) Bias Probabilities for Neutral and Biased Contexts; (e) Comparison of Bias Probabilities.

Heatmap of Bias Probabilities (Figure 10a)shows that words like "Demanding," "Driven," "Headstrong," and "Intuitive" exhibit high bias probabilities in biased contexts, while bias probability is significantly lower in neutral contexts. According to the Box Plot of Bias Probabilities (Figure 10b) it can be easily seen that the median bias probability for biased contexts is significantly higher than for neutral contexts, with larger variability in biased contexts. Distribution of Bias Probabilities (Figure 10c) demonstrates High frequency of bias probabilities close to 1 in biased contexts indicates many words/phrases are perceived as highly biased. Bias Probabilities for Neutral and Biased Contexts (Figure 10d) states that bias probabilities are generally higher for biased contexts compared to neutral contexts. Comparison of Bias Probabilities (Figure 10e): Most points cluster towards the top-right, indicating that words/phrases with high bias probabilities in biased contexts also tend to have higher probabilities in neutral contexts.

A detailed analysis of the top 20 biased tokens in age cyberbullying data was conducted to identify commonly biased words and phrases. Figure 11 represents the results.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2024 doi:10.20944/preprints202407.0411.v1

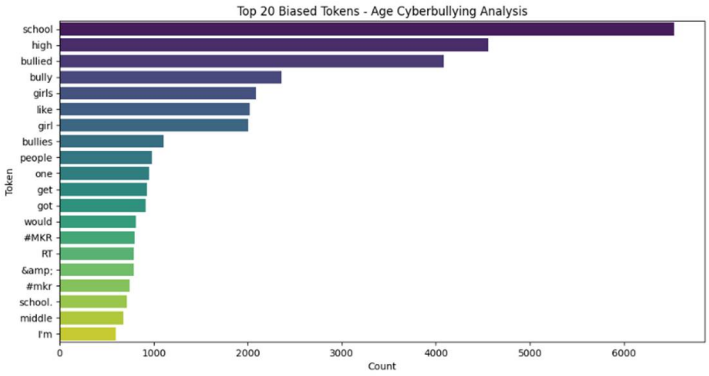


Figure 11. Top 20 Biased Tokens for authentic cyberbullying data.

The analysis revealed that words like "school," "high," "bullied," "bully,", "girls," and "like" are among the most frequently occurring biased tokens in age cyberbullying contexts. Splitting the data into clusters created by the model 'MiniLM-L6-v1', a Sentence Transformer optimized for generating embeddings for sentences or paragraphs, provided further insights. MiniLM is a smaller, faster variant of the BERT-like Transformer family [15], designed to offer similar performance to larger models like BERT but with a fraction of the parameters, making it faster and more efficient.

To introduce more diversity and variability in the data, the following data augmentation techniques were applied: synonym replacement (replaces random words in a sentence with their synonyms, introducing variation without changing the overall meaning) and random insertion (inserts synonyms of random words into random positions in the sentence, increasing length and complexity). The framework developed in this study integrates bias detection using the DistilBERT model for initial bias analysis, followed by multilabel classification for both biases and cyberbullying labels using various models. This approach ensures comprehensive analysis and detection of biases and cyberbullying in diverse datasets. The efficiency and effectiveness of these models in detecting biases and cyberbullying highlight the potential for AI to contribute to creating safer and more inclusive online environments.

Data augmentation helps mitigate bias by introducing more diversity and variability into the training data. By generating multiple variations of each sentence, the model is exposed to a wider range of linguistic patterns and contexts. This can help reduce overfitting and make the model more robust to different expressions of the same underlying concepts.

Algorithm 4. Data Augmentation for Cyberbullying Detection

Input: Sentences, labels, number of augmentations (num_augments)
Output: Augmented sentences and labels

```
Define get_synonyms(word):
    Initialize an empty set 'synonyms'
    For each synset in wordnet.synsets(word):
        For each lemma in synset.lemmas():
            Add lemma.name() to 'synonyms' (replace '_' with ' ')
    If word is in 'synonyms', remove it
    Return list of 'synonyms'

Define synonym_replacement(sentence, n):
    Split 'sentence' into 'words'
    Copy 'words' to 'new_words'
    Create a list 'random_word_list' of unique words that have synonyms
    Shuffle 'random_word_list'
    Set 'num_replacements' to the minimum of 'n' and the length of 'random_word_list'
    For each 'random_word' in the first 'num_replacements' words of 'random_word_list':
        Get 'synonyms' for 'random_word'
        If 'synonyms' exist, randomly choose a 'synonym'
        Replace 'random_word' in 'new_words' with 'synonym'
```

 doi:10.20944/preprints202407.0411.v1


```
    Join 'new_words' into a string and return it

Define random_insertion(sentence, n):
    Split 'sentence' into 'words'
    Copy 'words' to 'new_words'
    For each _ in range(n):
        Randomly choose 'new_word' from 'words'
        Get 'synonyms' for 'new_word'
        If 'synonyms' exist, randomly choose a 'synonym'
        Randomly choose 'insert_position' in 'new_words'
        Insert 'synonym' at 'insert_position' in 'new_words'
    Join 'new_words' into a string and return it

Define augment_data(sentences, labels, num_augments):
    Initialize empty lists 'augmented_sentences' and 'augmented_labels'
    For each 'sentence', 'label' in zip(sentences, labels):
        Append 'sentence' to 'augmented_sentences'
        Append 'label' to 'augmented_labels'
        For each _ in range(num_augments):
            If random.random() < 0.5:
              Perform synonym_replacement on 'sentence' & append to 'augmented_sentences'
            Else:
              Perform random_insertion on 'sentence' and append to 'augmented_sentences'
            Append 'label' to 'augmented_labels'
    Return 'augmented_sentences' and 'augmented_labels'

Load sentences and labels from file paths
Augment data using augment_data(sentences, labels, num_augments)
```

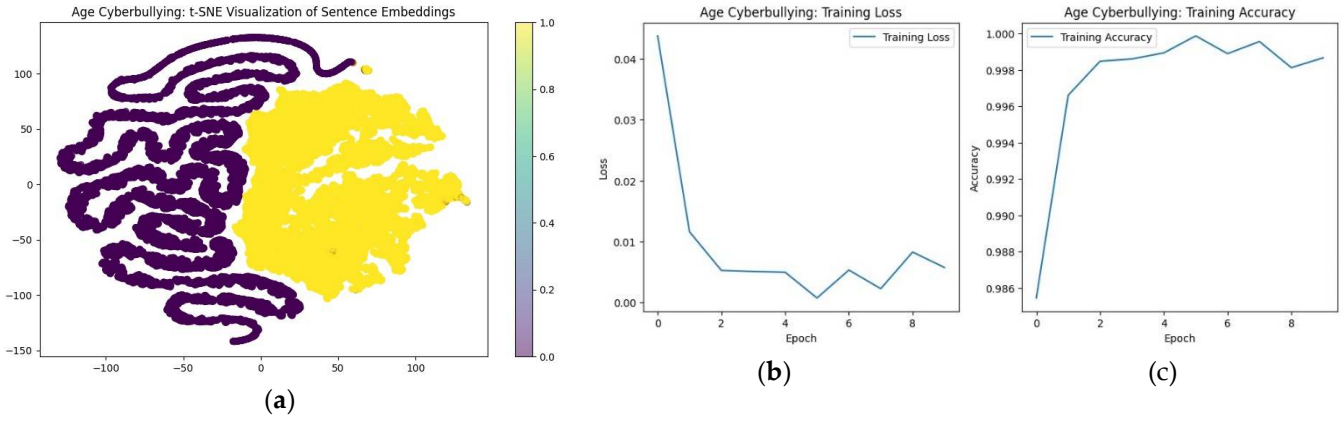


Figure 12. DistilBert Transformer Data Augmentation Results (**a**) Age Cyberbullying Embeddings; (**b**) Training Accuracy (**c**) Training Loss.

Figure 12 (a) represents how the embeddings based on the augmented sentences led to a more complex and intertwined structure. The data augmentation techniques introduced more variability, making the clusters in the t-SNE plot less distinct but potentially capturing more nuanced relationships between sentences. The training accuracy and loss plots demonstrate that as the epochs progress, the model's accuracy steadily increases while the loss decreases, indicating effective learning and convergence towards optimal performance. This trend suggests that the model is becoming more accurate in its predictions over time and the loss function is being minimized effectively.

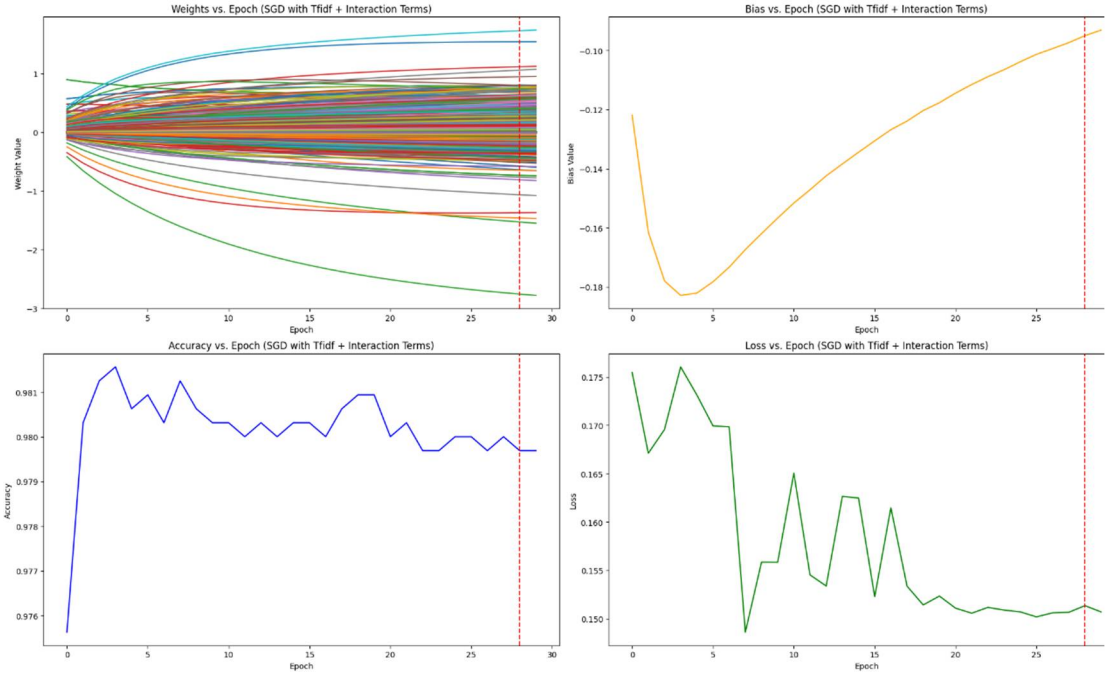
The data augmentation techniques introduce more variability and diversity in the training data what helps the model generalize better and reduces the likelihood of overfitting to specific patterns in the original data, thereby mitigating bias. The resulting t-SNE plot from the second script shows a more complex structure, indicating that the model is capturing a wider range of linguistic variations.

 doi:10.20944/preprints202407.0411.v1

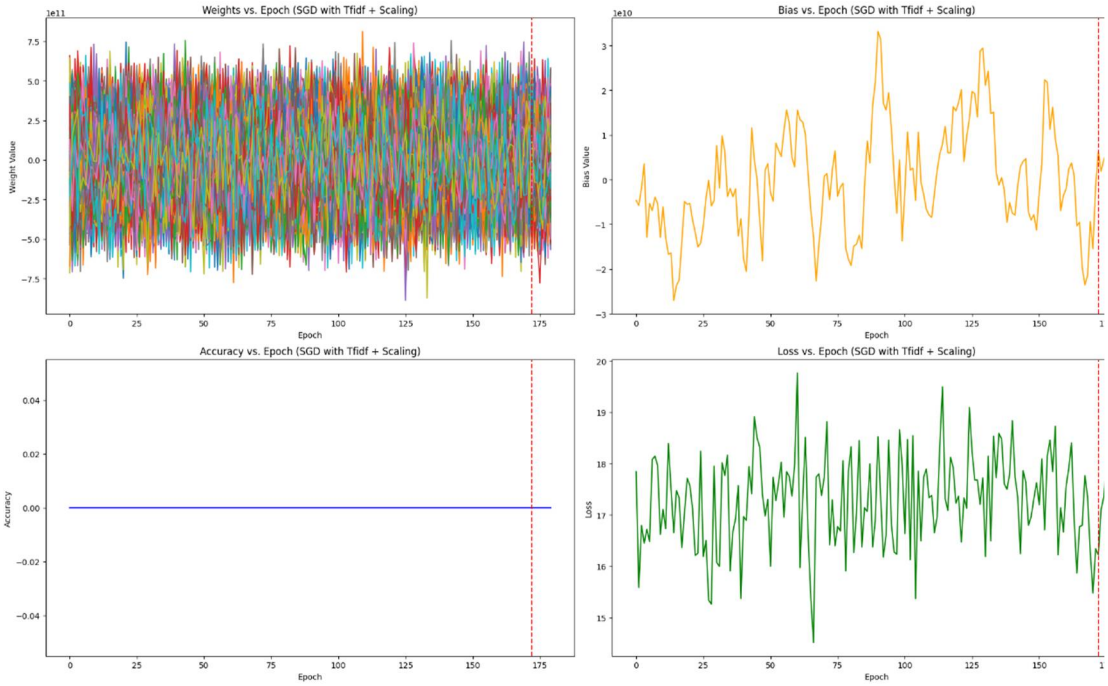
In comparison with Figure 9 (a) model's understanding of the data has evolved, potentially leading to improved classification performance.

4.6. Applying Optimization and Quantization techniques to Authentic Cyberbullying Data

Trials results for various ways of optimization can be seen below. This analysis highlights the importance of choosing appropriate pre-processing techniques and understanding their impact on the training process to achieve optimal model performance. Figure 13 provides insights into how weights, bias, accuracy, and loss change over epochs during the training process.

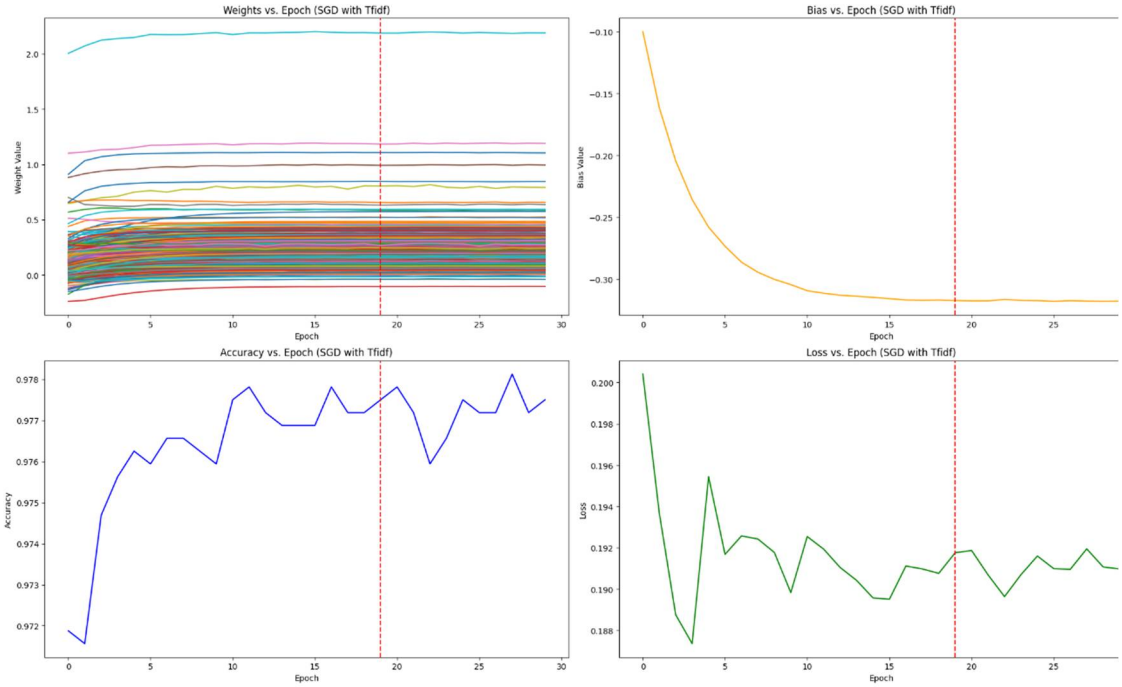


(a)



(b)

 doi:10.20944/preprints202407.0411.v1



(c)

Figure 13. Optimization Trials (**a**) Stochastic Gradient Descent (SGD) optimization with Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction; (**b**) SGD with TF-IDF + Scaling (**c**) SGD with TF-IDF + Interaction Terms.

As can be seen above various optimization strategies were applied with focus on Stochastic Gradient Descent (SGD) with different pre-processing techniques. Each figure corresponds to a different configuration of the training process: SGD with TF-IDF, SGD with TF-IDF + Scaling, SGD with TF-IDF + Interaction Terms. According to Figure 13 (a) the weights gradually stabilize over epochs, indicating convergence of the model parameters. The bias reduces quickly and stabilizes, showing that the model adjusts quickly and becomes consistent. Accuracy stabilizes after some initial fluctuation, indicating the model's learning progress. The loss decreases and stabilizes, confirming that the model is learning and minimizing errors. What can be seen in Figure 13 (b) SGD with TF-IDF + Scaling is that this configuration uses SGD optimization with TF-IDF for feature extraction, followed by scaling. The weights fluctuate significantly, suggesting instability in the training process due to scaling. The bias shows high variability, indicating that the model is struggling to find a stable solution. Accuracy remains zero throughout, indicating that the model is not learning effectively with this configuration. The loss remains high and variable, showing that the training process is not effective. According to Figure 13 (c) - SGD with TF-IDF + Interaction Terms - it uses SGD optimization with TF-IDF for feature extraction, including interaction terms. Weights converge, showing that the model parameters stabilize. The bias shows an initial decrease but then increases slightly, suggesting interaction terms introduce complexity. Accuracy improves and stabilizes, indicating effective learning with interaction terms. Loss decreases initially but shows slight fluctuation, indicating that the model's error minimization is impacted by the added complexity of interaction terms.

Among the configurations tested, SGD with TF-IDF Figure 13 (a) shows the most stable and effective results, with weights and bias stabilizing, accuracy improving, and loss decreasing. The addition of scaling on Figure (b) introduces instability, while the inclusion of interaction terms on Figure (c) adds complexity that slightly affects stability.

Dynamic quantization helps reduce model size and improve inference speed without significant changes to the model architecture or training process. It was tried on a DistilBert model. This method quantizes the weights of the model during runtime, typically focusing on reducing the memory footprint and computational cost without requiring extensive changes to the training process. Some

 doi:10.20944/preprints202407.0411.v1

preparation steps for quantization-aware training (QAT) were tried as well. For the testing purposes the models were trained with fake quantization modules that simulate the effects of quantization during the training process. This approach helped the model to better adapt to the eventual quantized state. Moving forward researchers might try static quantization, that involves calibrating the model using a representative dataset to determine appropriate scaling factors for activations and weights. It is expected to improve performance by quantizing both weights and activations statically. There are trials currently in progress and will be published in later papers once the investigation is complete.

4.6. Preliminary Results of Multilabel Classification

The team of researchers are working on a multilabel natural language processing of the same dataset where data first is labeled as biased vs not biased and then both biases and cyberbullying are detected at the same time.

The algorithm can be seen below:

Algorithm 5. Multilabel Classification for Cyberbullying Detection

Input: Text files containing cyberbullying and non-cyberbullying data
Output: Trained models, evaluation results, and visualizations
Define create_gpt4_tokenizer() To initialize GPT-4 tokenizer using tiktoken.
Define read_text_file(filepath, cyber_label) to read cyberbullying data and label it
Define load_lexicon(filepaths) to read biased lexicon and label it
Read and Combine the datasets
Split the dataset into train and test sets
Convert the DataFrame to Hugging Face Dataset
Define simple_bias_detection(text)
Split the text into words.
Count words that exist in the bias lexicon.
Return the ratio of biased words to total words.
Apply simple bias detection method
Calculate bias scores for each text and add as a new column in the DataFrame.
Convert bias scores to binary labels
Update the dataset
Combine the cyberbullying labels and bias labels into a single label column.
Update the DataFrame with the combined labels.
Split the updated dataset into train and test sets
Use train_test_split to divide the updated DataFrame into training and testing sets.
Convert both sets into Hugging Face Datasets.
Initialize the Hugging Face AutoTokenizer
Load the AutoTokenizer from the specified model.
Define tokenize_function(examples)
Define Data DataCollatorWithPadding to handle padding during tokenization.
Define the custom multi-label classification model class
Initialize a pretrained sequence classification model.
Define the forward function for the model.
Define train_and_evaluate_model(model_name, token)
Initialize the tokenizer and tokenized datasets.
Initialize the custom model.
Define training arguments.
Define a function to compute evaluation metrics.
Initialize the Trainer with the model, tokenizer, and datasets.
Define a list of model names to be evaluated, Train and evaluate each model.
Plot results using Plotly to visualize the evaluation metrics.
Train Word2Vec model on the dataset.
Visualize word vectors using PCA
Visualize token embeddings

Preliminary results of the multilabel classification can be seen in Figure 14.

 doi:10.20944/preprints202407.0411.v1

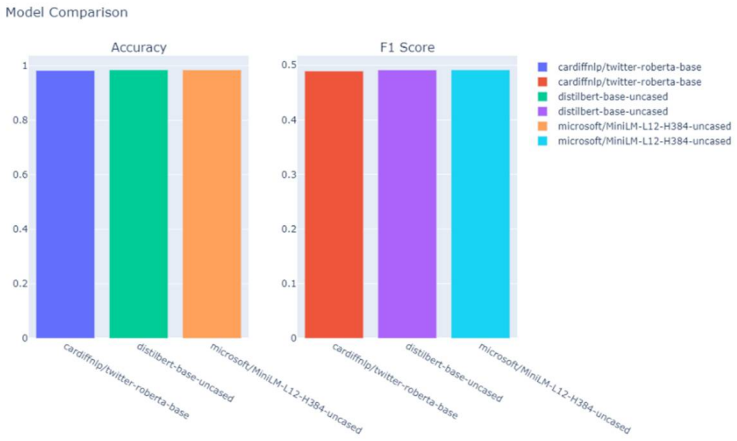


Figure 14. Model Comparison by Accuracy and F1 Score

4.7. Swarm of AI agents and Bias Detector app

As was explored at the beginning of the paper Large Language models (LLMs) can generate biased data on demand. While it is possible to do it manually it is not a problem to do so via API calls as well. The researchers utilize OpenAI Assistants API [37] and ChatGPT-4o LLM under the hood to create our system. AI Assistant Biased Data Generator was created via API and utilized in the study. The code is simple and straightforward and can be observed at-a-Glance. See Figure 15.

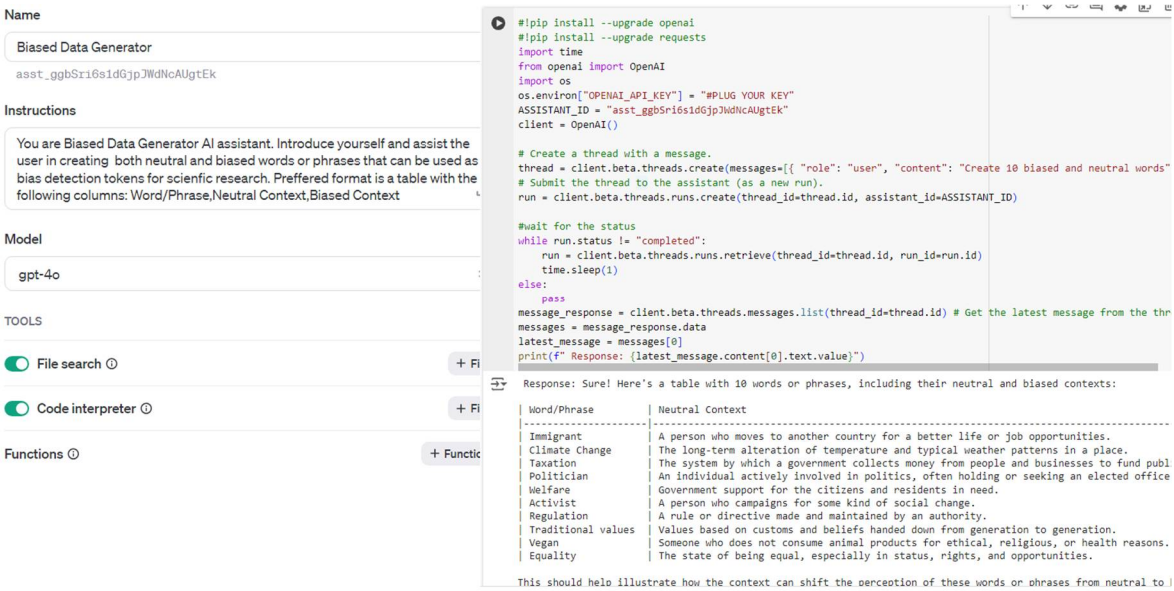


Figure 15. Backend of the Biased Data Generator AI assistant (**a**) Custom GPT settings; (**b**) Python code and output.

In this project the researchers explore the phenomena of swarm of agents, that became possible due to the introduction by OpenAI Assistants API multiple threads. Figure 16 represents a non-technical understanding of the swarm of agents' idea. Technically this idea becomes more and more real (robots will build robots to build robots, etc.).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2024 doi:10.20944/preprints202407.0411.v1

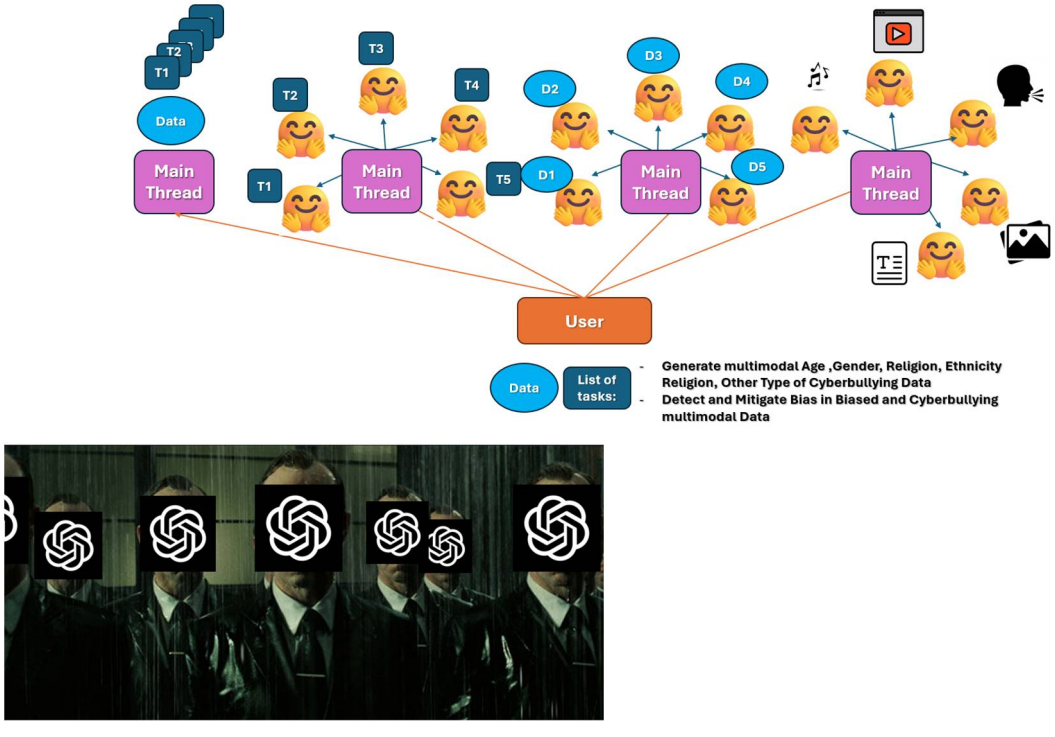
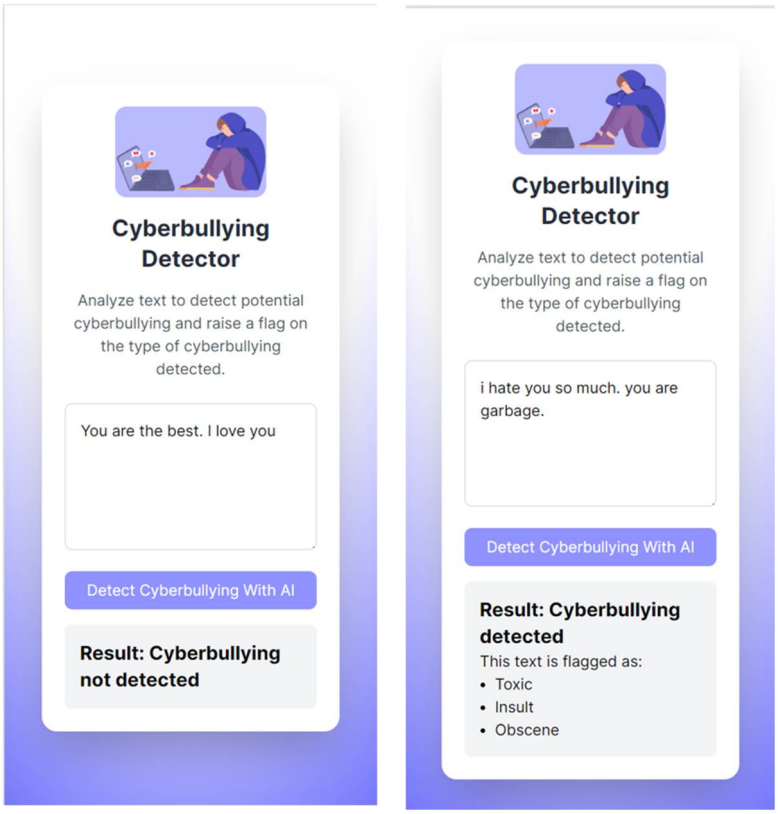


Figure 16. Swarm of agents in a public view.

At this point the researchers consider three possible situations: when agents can do work in parallel applying the concept of divide-and-conquer to either split data or tasks or both if possible; as well as splitting various modalities. Figure 17 represents these three test cases at-a-Glance.

Figure 17. Swarm of agents in Bias Data generation

The team of researchers developed several possible prototypes of the Cyberbullying Detector application. One of them can be seen on Figure 18.



 doi:10.20944/preprints202407.0411.v1

Figure 18. Cyberbullying Detector Prototype

5. Conclusions

By integrating synthetic and authentic datasets and utilizing advanced AI models, this research aims to enhance bias detection mechanisms and promote the development of more equitable AI systems. The potential dangers of cyberbullying have been well-documented, prompting the need for effective and unbiased detection systems. This study has underscored the significance of addressing inherent biases in Large Language Models (LLMs) and Transformer models. The presented Biased Data Generator, built on the foundations of the OpenAI API, presents an innovative approach in AI making AI generate Biased Data for AI. It not only unusual but uses just recently released newest multimodal OpenAI API model ChatGPT-4o. Future work includes exploration of multilabel classification (when labels include bias and cyberbullying intersectionally) as well as multithreaded capabilities of the Assistant API simulating heavy load of assistant usage on various threads, synthetic dataset and database storage expansion, work toward more advanced bias-mitigation strategies, and collecting user's feedback on the app. Deploying the app in real-world scenarios, such as Universities and Research facilities, could provide invaluable data on its efficacy and areas of improvement. As technology evolves, the code, model and prompts should be further developed and refined. Biases in AI systems, especially those trained on language data, can lead to discriminatory outcomes and harm individuals and communities. Ensuring fairness and reliability in AI is crucial for the success and ethical deployment of these technologies.

Author Contributions: Conceptualization, Y.K.; methodology, Y.K.; software, R.J., A.P. and G.Y.; validation, D.K., P.M. and J.J.L.; formal analysis, J.J. Li; investigation, K.H.; resources, Y.K.; data curation, G.Y.; writing—original draft preparation, Y.K.; writing—review and editing, J.J.L.; visualization, Y.K. and R.J.; supervision, P.M. and D.K.; project administration, Y.K.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the NSF, grants 1834620 and 2129795, and Kean University (Union, NJ).

Data Availability Statement: Synthetic dataset created with the help of LLMs will be published together with this paper once its finalized.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Huber, M., Luu, A. T., Boutros, F., Kuijper, A., & Damer, N. (2023). Bias and Diversity in Synthetic-based Face Recognition. arXiv preprint arXiv:2311.03970
2. Raza, S., Bamgbose, O., Chatrath, V., Ghuge, S., Sidyakin, Y., & Muaad, A. Y. (2023). Unlocking Bias Detection: Leveraging Transformer-Based Models for Content Analysis. arXiv preprint arXiv:2310.00347.
3. Raza, S., Garg, M., Reji, D. J., Bashir, S. R., & Ding, C. (2024). Nbias: A natural language processing framework for BIAS identification in text. Expert Systems with Applications, 237, 121542.
4. Turpin, M., Michael, J., Perez, E., & Bowman, S. (2024). Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. Advances in Neural Information Processing Systems, 36.
5. R. Kulesza, Y. Kumar, R. Ruiz, A. Torres, E. Weinman, J. J. Li, P. Morreale. (2020) Investigating Deep Learning for Predicting multi-linguistic conversations with a Chatterbot, In Proceedings of the 2020 5th IEEE International Conference on Big Data Analytics (ICBDA).
6. Mathur, V., Stavrakas, Y., & Singh, S. (2016, December). Intelligence analysis of Tay Twitter bot. In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) (pp. 231-236). IEEE.
7. Tellez, N., Serra, J., Kumar, Y., Li, J.J., Morreale, P. (2023). Gauging Biases in Various Deep Learning AI Models. In: Arai, K. (eds) Intelligent Systems and Applications. IntelliSys 2022. Lecture Notes in Networks and Systems, vol 544. Springer, Cham.
8. Ayoub, N. F., Balakrishnan, K., Ayoub, M. S., Barrett, T. F., David, A. P., & Gray, S. T. (2024). Inherent Bias in Large Language Models: A Random Sampling Analysis. Mayo Clinic Proceedings: Digital Health, 2(2), 186-191.
9. Google profanity words GitHub repo. Available online: https://github.com/coffee-and-fun/google-profanity-words/blob/main/data/en.txt (accessed on 27 April 2024).

10. List of Dirty Naughty Obscene and Otherwise-Bad-Words Github Repo. Available online: https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words (accessed on 27 April 2024).
11. Carroll L. Alice's Adventures in Wonderland. Available online: https://www.gutenberg.org/ebooks/11 (accessed on 26 May 2024).
12. Inflection, A. I. (2023). Inflection-1. Technical report, 2023b. Available online: https://inflection.ai/assets/Inflection-1.pdf (accessed on 6 June 2024).
13. Sentiment Pipeline from Hugging Face. Available online: https://huggingface.co/docs/transformers/en/main_classes/pipelines (accessed on 6 June 2024).
14. Kumar Y, Morreale P, Sorial P, Delgado J, Li JJ, Martins P. A Testing Framework for AI Linguistic Systems (testFAILS). Electronics. 2023; 12(14):3095. https://doi.org/10.3390/electronics12143095
15. Sentence Transformers all-MiniLM-L6-v2 page on Hugging Face. Available online: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (accessed on 27 April 2024).
16. Jason Wang, Kaiqun Fu, Chang-Tien Lu, November 12, 2020, "Fine-Grained Balanced Cyberbullying Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/kn1c-zx22.
17. Raza, S., Reji, D. J., & Ding, C. (2024). Dbias: detecting biases and ensuring fairness in news articles. International Journal of Data Science and Analytics, 17(1), 39-59.
18. Li, B., Peng, H., Sainju, R., Yang, J., Yang, L., Liang, Y., ... & Ding, C. (2021). Detecting gender bias in transformer-based models: A case study on bert. arXiv preprint arXiv:2110.15733.
19. Silva, A., Tambwekar, P., & Gombolay, M. (2021, June). Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 2383-2389).
20. Dusi, M., Gerevini, A. E., Putelli, L., & Serina, I. (2024). Supervised Bias Detection in Transformers-based Language Models. In CEUR WORKSHOP PROCEEDINGS (Vol. 3670).
21. Raza, S., Bamgbose, O., Chatrath, V., Ghuge, S., Sidyakin, Y., & Muaad, A. Y. M. (2024). Unlocking Bias Detection: Leveraging Transformer-Based Models for Content Analysis. IEEE Transactions on Computational Social Systems.
22. Barbierato, E., Vedova, M. L. D., Tessera, D., Toti, D., & Vanoli, N. (2022). A methodology for controlling bias and fairness in synthetic data generation. Applied Sciences, 12(9), 4619.
23. Baumann, J., Castelnovo, A., Cosentini, A., Crupi, R., Inverardi, N., & Regoli, D. (2023, August). Bias on demand: investigating bias with a synthetic data generator. In 32nd International Joint Conference on Artificial Intelligence (IJCAI), Macao, SAR, 19-25 August 2023 (pp. 7110-7114). International Joint Conferences on Artificial Intelligence Organization.
24. Yu, Y., et al. (2024). Large language model as attributed training data generator: A tale of diversity and bias. Advances in Neural Information Processing Systems, 36.
25. Gujar, S., Shah, T., Honawale, D., Bhosale, V., Khan, F., Verma, D., & Ranjan, R. (2022, June). Genethos: A synthetic data generation system with bias detection and mitigation. In 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) (pp. 1-6). IEEE.
26. Rosa, H., et al. (2019). Automatic cyberbullying detection: A systematic review. Computers in Human Behavior, 93, 333-345.
27. Dadvar, M., et al. (2012). Improved cyberbullying detection using gender information. Proceedings of DIR 2012, Universiteit Gent.
28. Ali, A., & Syed, A. M. (2020). Cyberbullying detection using machine learning. Pakistan Journal of Engineering and Technology, 3(2), 45-50.
29. Al-Ajlan, M. A., & Ykhlef, M. (2018). Deep learning algorithm for cyberbullying detection. International Journal of Advanced Computer Science and Applications, 9(9).
30. Lee, P. J., et al. (2018). Cyberbullying Detection on Social Network Services. PACIS, 61.
31. Wang, J., Fu, K., & Lu, C. T. (2020, December). Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 1699-1708). IEEE.
32. Singh, V. K., Ghosh, S., & Jose, C. (2017, May). Toward multimodal cyberbullying detection. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (pp. 2090-2099).
33. Hannon, B., Kumar, Y., Sorial, P., Li, J. J., & Morreale, P. (2023, July). From Vulnerabilities to Improvements-A Deep Dive into Adversarial Testing of AI Models. In 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE) (pp. 2645-2649). IEEE.
34. Transformer model d4data/bias-detection-model page on Hugging Face. Available online: https://huggingface.co/d4data/bias-detection-model (accessed on 8 June 2024)
35. Home page of mistral-bias-0.9 model on Hugging Face. Available online: https://huggingface.co/yuhuixu/mistral-bias-0.9 (accessed on 27 April 2024).

 doi:10.20944/preprints202407.0411.v1

36. Sentence Transformer bert-base-uncased page on Hugging Face. Available online: https://huggingface.co/google-bert/bert-base-uncased (accessed on 27 April 2024).
37. OpenAI API Web Site (2024) Available online: https://openai.com/api/ (accessed on 24/5/2024).