Gender bias detection on hate speech classification: an analysis at feature-level

NASCIMENTO, Francimaria R. S., CAVALCANTI, George D. C. and COSTA-ABREU, Marjory Da <http://orcid.org/0000-0001-7461-7570>

Neural Computing and Applications (2025) 37:3887–3905
https://doi.org/10.1007/s00521-024-10841-8

ORIGINAL ARTICLE

Gender bias detection on hate speech classification: an analysis at feature-level

Francimaria R. S. Nascimento[1] · George D. C. Cavalcanti[1] · Marjory Da Costa-Abreu[2]

Received: 28 November 2023 / Accepted: 21 October 2024 / Published online: 17 December 2024

Abstract

Hate speech is a growing problem on social media due to the larger volume of content being shared. Recent works demonstrated the usefulness of distinct machine learning algorithms combined with natural language processing techniques to detect hateful content. However, when not constructed with the necessary care, learning models can magnify discriminatory behaviour and lead the model to incorrectly associate comments with specific identity terms (e.g., woman, black, and gay) with a particular class, such as hate speech. Moreover, some specific characteristics should be considered in the test set when evaluating the presence of bias, considering that the test set can follow the same biased distribution of the training set and compromise the results obtained by the bias metrics. This work argues that considering the potential bias in hate speech detection is needed and focuses on developing an intelligent system to address these limitations. Firstly, we proposed a comprehensive, **unbiased dataset** to unintended gender bias evaluation. Secondly, we propose a framework to help analyse bias from feature extraction techniques. Then, we evaluate several state-of-the-art feature extraction techniques, specifically focusing on the bias towards identity terms. We consider six feature extraction techniques, including TF, TF-IDF, FastText, GloVe, BERT, and RoBERTa, and six classifiers, LR, DT, SVM, XGB, MLP, and RF. The experimental study across hate speech datasets and a range of classification and unintended bias metrics demonstrates that the choice of the feature extraction technique can impact the bias on predictions, and its effectiveness can depend on the dataset analysed. For instance, combining TF and TF-IDF with DT and MLP resulted in higher bias, while BERT and RoBERTa showed lower bias with the same classifier for the HE and WH datasets. The proposed dataset and source code will be publicly available when the paper is published.

Keywords Hate speech detection · Unintended gender bias · Feature extraction · Social media · Unbiased dataset · Machine learning techniques

1 Introduction

Social media platforms power user-generated content about various subjects to spread quickly and easily. As a result, the easy dissemination of content and anonymity on social

George D.C. Cavalcanti and Marjory Da Costa-Abreu These authors contributed equally to this work.

https://github.com/Francimaria/hate_speech_bias_feature.

✉ Marjory Da Costa-Abreu
md0948@exchange.shu.ac.uk

Francimaria R. S. Nascimento
frsn2@cin.ufpe.br

George D. C. Cavalcanti
gdcc@cin.ufpe.br

[1] Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Av. Jornalista Anibal Fernandes s/n, Recife, Pernambuco 50740-600, Brazil

[2] Department of Computing, Sheffield Hallam University, Sheffield, UK

media platforms has facilitated online hate speech to proliferate. In [1], hate speech is defined as "*Hate speech is language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humour is used*". The dissemination of hate speech on these platforms is potentially harmful and causes serious impacts on the victims. However, the enormous amount of content generated makes human moderation slow, expensive, and ineffective.

Recent studies have proposed several methods using distinct machine learning (ML) models, such as deep learning (DL) algorithms combined with natural language processing (NLP) techniques to detect hate speech content automatically [2–6]. However, when badly designed, learning models can exhibit? unintended unfair behaviours and lead the model to make decisions based on identity terms, such as woman, gay and black [7]. As [8] pointed out, "*a model contains unintended bias if it performs better for comments containing some particular identity terms than for comments containing others*". Figure 1 shows an example of unintended bias from the model evaluated in [9] before the bias mitigation procedures. This example illustrates the model's behaviour when it overgeneralises the association of a specific term ("woman") and the hate label. It results in a high probability of the model classifying as hate a non-hate sample. In this example, the classifier predicted the samples using the word "man" with a hate label score of 0.13, while the same example with the word "woman" with a higher score of 0.40.

The potential bias in learning models raises concerns regarding the robustness of the systems and the impact of this unintended bias on the generalisation of the systems in practical applications [10, 11]. Different studies have exhibited bias associated with identity terms (e.g., lesbian, gay, transgender, and so on) in benchmark datasets [8, 10]. Moreover, racial and dialectic biases have been proven in trained classifiers for hate speech detection, as evidenced by the correlation between words associated with African American English dialect (AAE) and the hate speech label

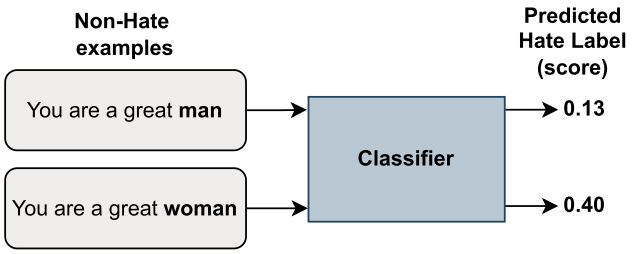

Fig. 1 Example of unintended bias in non-hateful tweets

[12, 13]. Studies also demonstrated the presence of gender bias in trained classifiers for hate speech detection [9, 14]. Therefore, it is essential to consider the potential bias in the model development process since it can cause unfairness towards specific groups that these classifiers are usually designed to protect.

The development of machine learning models can lead to unintended bias at different stages [15]. An essential aspect of developing a machine learning solution for text classification is extracting meaningful features from data [16]. The feature representation used as input is a relevant factor contributing to a machine learning model's effectiveness. Several feature extraction techniques have been applied in the hate speech detection context, including methods based on Bag-of-Words [17], lexical resources [18], and text embedding and deep learning approaches [19]. The embedding techniques have improved the classification performance for abusive and hate speech detection [19, 20]. However, a comparative study analysing the impact of feature extraction techniques for unintended bias in the classification of hate speech is still an open research question.

This problem's understanding is essential because it can help mitigate stereotypes in hate speech detection systems and related domains. For instance, there is an increasing number of applications based on textual content analysis, such as machine translation systems [21], recommendation systems [22], and large language models like ChatGPT[1] (Chat Generative Pre-trained Transformer), that can be influenced by biases as well. The bias can manifest in various forms and negatively impact the effectiveness of these systems. Therefore, it is essential to consider the potential bias when designing and implementing text-based technologies.

In this study, we investigate unintended bias, specifically related to gender identity (gender bias). Gender bias can result in the model showing preference or prejudice towards a particular gender. Its dissemination can reinforce harmful stereotypes in the systems, resulting in real-world consequences [23]. For instance, concerns have been raised about sexist behaviours of Artificial Intelligence (AI) tools for resume filtering systems penalising women in the hiring process based solely on their gender[2] [24, 25]. Although these, few studies have addressed this issue related to the feature extraction technique in the hate speech detection context.

We performed a comprehensive analysis, considering six feature extraction methods TF (Term Frequency), TF-IDF (Term Frequency-Inverse Document Frequency), RoBERTa (Robustly Optimized BERT Pre-training

[1] https://openai.com/.

[2] https://www.bbc.co.uk/news/technology-45809919.

Approach), BERT (Bidirectional Encoder Representations from Transformers), FastText, and GloVe (Global Vectors for Word Representation) used for feature extraction and different classification algorithms. To understand whether the feature extraction method impacts the unintended gender bias learned by the model and if this behaviour is followed in different datasets. Hence, this study aims to answer the following research questions: (1) Does the choice of the feature extraction technique impact the presence of unintended gender bias on the model's prediction? (2) Do feature extraction techniques tend to present bias when dealing with different datasets? (3) Can the bias affect the performance of the models? Experiments on three real-world English datasets for hate speech detection demonstrate that feature extraction techniques can impact the bias on predictions. Moreover, we explored the behaviour of the feature extraction techniques using several classifiers with various metrics. It allows us to explore different nuances of the bias problem.

The main contributions of this paper are:

1. The design of a framework to help analyse the biased behaviour of feature extraction techniques.
2. The proposal of an unbiased dataset for assessing unintended gender bias in the context of hate speech detection, while existing studies mainly focused on debiasing datasets. This dataset comprises all identity terms in the same context to ensure a fair and unbiased evaluation.
3. Our experiments show that feature extraction techniques can impact unintended gender bias in predictions. For instance: TF and TF-IDF presented more biased behaviour than FastText, GloVe, BERT, and RoBERTa for the FPED and FNED metrics.

Thus, we aim to achieve these contributions by presenting our work which is organised as follows: Sect. 2 presents related work. Section 3 discusses the proposed methodology and the proposed unbiased dataset. The experimental setups are described in Sect. 4. Section 5 presents the results. Section 6 provides a comprehensive discussion. Section 7 concludes the work with the final remarks.

2 Related work

Several approaches for hate speech detection have been proposed based on classic machine learning models [5, 17], ensemble learning [26], and deep learning techniques [4] combined with different techniques for feature extraction. General feature extraction techniques for text mining have been applied to the hate speech detection problem. The word embedding methods have been more frequently used than classical methods, such as bag-of-words (BoW) and n-grams. These techniques can capture semantic information from the text and the syntactic relationship between the words [27].

Cruz et al. [3] proposed a framework that evaluates and selects multiple feature extraction techniques and classification models that complement each other to design a robust multiple classifier system. The authors demonstrated the effectiveness of the proposed methodology in four hate speech datasets.

Firmino et al. [28] proposed a method based on Cross-Lingual Learning for detecting hate speech in texts. The method used Pre-Trained Language Models in English, Italian, and Portuguese, and showed better performance for the OffComBr-2 corpus.

Even though these contributions have improved the performance of hate speech detection models, they did not explicitly consider the potential bias in the models. Dixon et al. [8] introduced the concept of unintended bias in the toxicity language datasets. The authors investigate unintended bias regarding identity terms (atheist, gay, transgender, etc.) and try to mitigate the bias using statistic correction to balance the data with the most disproportional distribution. In [7], the authors investigated the subjectivity level of a comment and the presence of identity terms to mitigate its bias.

The hate speech detection models can present different types of unintended bias, such as racial, annotation, cross-geographic, political, and so on [29, 30]. In [29], the authors investigated social stereotypes in the automated detection of hate speech. The authors pointed out that the bias can be developed due to limited perspective and repeated exposure to similar behaviour.

Gender stereotypes hosted in hate speech datasets are also a serious concern, in which a model can perform better with determinate gender identity terms than comments with others [14]. In [9], the authors proposed a multi-view ensemble learning approach to learn distinct abstractions of the problem and found effective results compared to the literature. In [31], the authors analysed the effect of debiased embedding for mitigating gender bias in English and Turkish tweets. They concluded that the classification performance of hate speech detection models based on neural embeddings could be improved by removing the gender-related bias. In [32], the researchers proposed an approach to gender bias evaluation using word embeddings. They analysed Twitter data from Hong Kong and demonstrated the temporal trend and spatial distribution of these biases.

Table 1 summarises the related works that address hate speech detection and investigate concepts related to bias. This table shows the reference of the paper and its publication year, the datasets, the feature extraction technique, and the classifiers evaluated. The column "gender bias"

Table 1 Related works summary

Year	References	Dataset	Feature	Classifier	Gender bias	Unbiased test set
2018	[8]	Wikipedia talk pages	Convolutional neural network (CNN)	CNN	×	✓
2018	[14]	Twitter WH [33], FN [34]	Word2Vec, fast text, randomly initialised embeddings (random)	CNN,gated recurrent units (GRU), α-GRU	✓	✓
2019	[27]	Twitter HE [35]	InferSent, concatenated power mean word embedding, lexical vectors, universal sentence encoder, embeddings from language model (ELMo)	Logistic regression (LR), random forest (RF), Support vector machines - radial basis function (SVM-RBF), extreme gradient boosted decision trees (XGBoost)	×	×
2020	[5]	YouTube [36], Reddit [37], Wikipedia [38], Twitter dataset [39]	BoW, term frequency-inverse document frequency (TF-IDF), Word2Vec, BERT, and all combined	LR, naïve bayes,, XGBoost, and neural networks	×	×
2020	[17]	WHO, 2021 Twitter DV [39], FN [34]	BoW, TF-IDF, n-grams, dictionary (Hatebase), FrameNet, Word2Vec	SVM	×	×
2020	[4]	Twitter DV [39], WH [33], Hindi-English, OLID [40], Harassment [41]	word and char embeddings, CNN	Deep Multi-task Learning (MTL), CNN, LSTM, stacking of CNN+GRU, and CNN_{α}+GRU	×	×
2022	[3]	Twitter DV [39], WH [33], HE [35], DV + WH	GLoVe, Word2Vec, FastTex, Term Frequency (TF), TF-IDF	Ensemble learning	×	×
2022	[7]	Stormfront [42], Twitter WH [33], FN [34], Kaggle-Wikipedia[3]	BERT	Subdentity-BERT (SS-BERT)	×	×
2022	[9]	Twitter WH [33], WS [43], DV [39], HE [35]	GloVe, FastText, BERT, TF, TF-IDF, char and word n-grams	Ensemble learning	✓	✓
2022	[31]	Twitter [44]	BoW, fastText, BERT	SVM, BiLSTM	✓	✓
2023	[26]	Kaggle-Wikipedia[4]	GloVe, fastText, BERT	BERT-based ensemble learning	×	×
2023	[29]	Gab Hate Corpus	BERT, RoBERTa, and TF-IDF	BERT, RoBERTa, and SVM	✓	×
2024	[28]	OffComBr-2, OffComBr-2 translated into English,Evalita 2018, and WH	BERTimbau, Italian BERT, BERT, and XLM-R	Zero-shot transfer, Joint Learning, and Cascade Learning	×	×
2024	[32]	Twitter data from Hong Kong	Word2Vec		✓	×
2024	**Our**	Twitter DV [39], WH [33], HE [35]	TF, TF-IDF, RoBERTa, GloVe, FastText, BERT	SVM, LR,Decision Tree (DT), XGBoost, Multi-Layer Perceptron (MLP), RF	✓	✓

[3] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[4] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

denotes whether the work considered the unintended gender bias in the proposed model, and the column "unbiased test set" indicates whether the dataset used to evaluate the bias follows an impartial data distribution in the context applied.

Despite the previous contributions to hate speech detection, the potential biases did not receive attention in

most works [3–5, 17, 26, 27]. Some efforts have investigated related bias concepts in hate speech detection, addressing the analysis only considering one input vector or features from the same families [7, 8]. Although the analysis of different features has been performed in some studies [9, 14, 31], these studies usually investigate the impact of the proposed bias mitigation model without considering the potential bias introduced by the original feature.

Moreover, none of these works presents a clear methodology for analysing the impact of unintended gender bias from multiple feature representations and how it affected the performance of classifiers. To fill this gap, this paper proposed a methodology for analysing the relationship between the feature extraction technique and the unintended bias in different hate speech datasets. The proposed methodology is presented in the following section.

3 Proposed methodology and unbiased dataset

This work investigates the relationship between the feature extraction technique and the unintended gender bias measured in the predictions of state-of-the-art machine learning techniques. Hate speech detection models are usually designed to classify the data in binary labels as Hate/Non-hate or multi-classes, and the model performance is calculated using the predictions from a test set. However, it is essential to consider the potential bias in the trained model against identity terms.

It is crucial to remark that while the original test set can be used to assess traditional metrics such as accuracy, it should not be evaluated to assess the bias since it may have the same biased distribution of identity terms as the training set, making bias identification challenging.

Therefore, a dataset with all identity terms in a similar context, the **unbiased dataset**, is necessary to properly evaluate the bias metrics, as these metrics depend on the identity term information. Considering the relevance and necessity of this dataset with these specific characteristics, we proposed a new unbiased dataset described in Sect. 3.1.

Figure 2 presents the proposed framework. The proposed includes three main stages: Feature extraction, Training, and Prediction, which receive three datasets: training (Δ), test (τ), and unbiased test (Γ) as input. The first step is the feature extraction, thus, given a training set Δ with text in natural language, the feature extractor F transforms the text in numeric feature spaces f_Δ. The second step is the training, in which the training set's data representation (f_Δ) is used to train a classifier. The trained classifier C predicts the classes from the unbiased test set numeric feature spaces f_Γ generated using F for bias evaluation. Then, in the last step, C predicts the classes from the test set numeric feature spaces f_τ also generated using F for classification performance evaluation. The outputs are the hate/non-hate prediction accuracy computed from τ and the unintended bias assessment using Γ.

Feature extraction. In the context of hate speech, datasets are usually available as raw text for analysis. Therefore, feature extraction aims to transform the natural language text into a numerical vector space suitable as model inputs. Several feature extraction techniques can be applied, such as Bag-of-Words (BoW) techniques [45, 46], lexical resources [47], and embedding methods [12, 26]. The feature extractor F transforms the raw text in numerical feature spaces, and each dataset is represented as a set of feature vectors denoted by f_Δ, f_τ, and f_Γ.

Training. The training process is responsible for the learning task, where the input vector (f_Δ) and its respective labels are used to fit a classification model, resulting in the trained classifier C. Several classifiers, such as classical machine learning and deep learning algorithms, can be applied to this task. We investigate a diverse set of state-of-art classifiers.

Prediction. As shown in Fig. 2, the prediction module receives as input the trained classifier C and the feature vectors f_τ and f_Γ, which represent the datasets τ and Γ, respectively. Thus, the same model C is used to predict the instances from two datasets; C predicts the hateful labels from f_τ and the predictions from f_Γ are used to calculate the unintended gender bias metrics since the test set τ may exhibit a biased distribution similar to the training set, which can impact the bias evaluation, as highlighted in [8].

3.1 Unbiased dataset

In the field of machine learning, it is common to split the dataset into at least two parts: training and testing. The evaluation of the models is performed using the test dataset. However, this dataset will likely follow the biased distribution of identity terms from the training set, so it should not be considered to evaluate the unintended bias accurately [8, 9]. To overcome this issue, a relevant strategy is to create a dataset with equal comment distribution for each term using the identity term template. Moreover, the bias metrics that are based on the difference in error rate equality need all identity terms in the same context. Therefore, we proposed a dataset with equal comment distribution for each term using the identity term template.

Table 2 shows examples of the identity term templates used. The idea is to generate sentences that change only the identity term, for example "you are a good woman" and "you are a good man". Therefore, a model without bias is expected to generalise well and generate similar

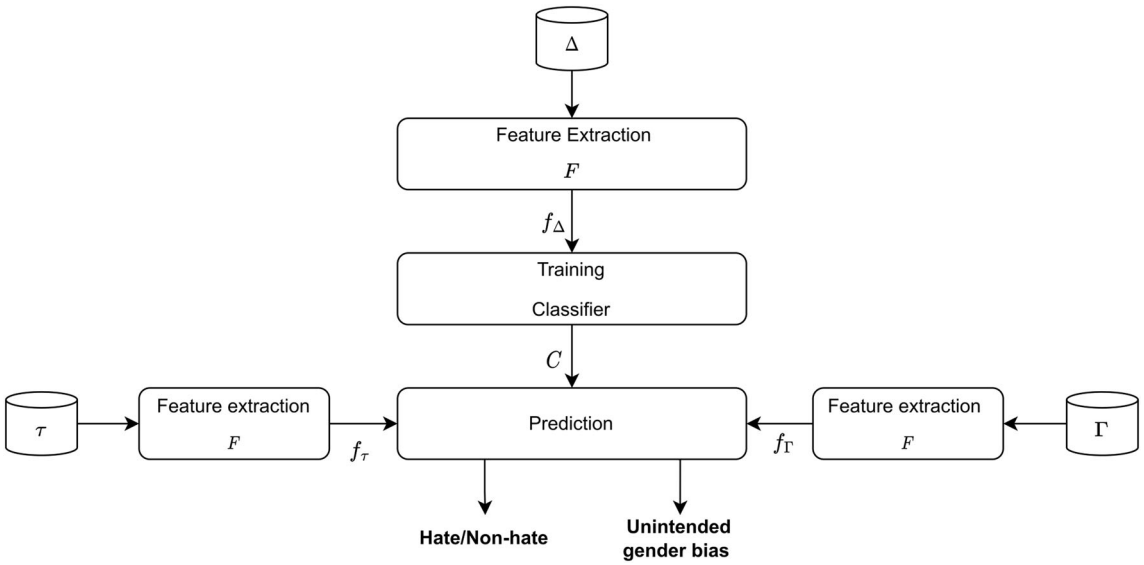

Fig. 2 Proposed methodology. Δ, τ, and Γ are the training, test, and unbiased test sets. F is the feature extractor. f_Δ, f_τ, and f_Γ are the matrices generated by F using the training, test, and unbiased test sets, respectively. C is the trained classifier

Table 2 Examples of templates. $<identity>$ denotes an identity term

Example template	Label
$<identity>$ think they're special. I hate them	Hateful
$<identity>$ should all be killed	Hateful
I hate all $<identity>$	Hateful
$<identity>$ should be protect	Non-hateful
We should not let $<identity>$ suffer	Non-hateful
I am $<identity>$	Non-hateful

predictions for the same sentence, modifying only the identity term. The availability of reliable labels across a range of terms is crucial in assessing bias effectively. Therefore, to generate a more extensive data sample and cover different expressions of hate, we investigated the combination of the dataset proposed in [9] and the HateCheck dataset [48].

In [9], the authors proposed templates filled with 24 identity terms (Table 3) to generate 1248 samples, of which 600 are hateful, and 648 are non-hateful. The HateCheck is a functional hate speech test dataset, meaning that this dataset is designed to test different functionalities of a model with targeted test cases, which was validated through a structured annotation process. The dataset comprises different cases of hate speech that can be particularly challenging cases for classification models. It is composed of 421 templates using identity terms. We filled the templates using a list of identity terms (Table 3) to ensure all identity terms appear in the same context.

Table 3 Identity terms. The word 'female' was spelled as 'femal' due to the pre-processing step

Identity terms
Woman, women, girl, sister, daughter, wife, girlfriend, mother, aunt, mom, grandmother, femal, man, men, boy, brother, son, husband, boyfriend, father, uncle, dad, grandfather, male

However, we used only 400 templates because some changed only the identity term style since we did not distinguish the templates by the identity term style, e.g. plural ($<identity\_p>$), singular ($<identity\_s>$), plural with spaces ($<identity\_p\_space\_add>$), and so on. For example, the templates "$<identity\_p>$ are disgusting." and "$<identity\_p\_space\_add>$ are disgusting." are considered as "$<identity>$ are disgusting." As the idea is to build sentences by changing only the identity term, these templates can generate repeated samples. It resulted in 9600 samples, of which 7296 are hateful and 2304 non-hateful. The combination of the datasets resulted in a dataset with 10848 examples. However, some examples presented the same text content after the pre-processing step (described in Sect. 4.2). Therefore, we remove these examples resulting in 10,728 instances, of which 7776 are hateful and 2952 non-hateful. The proposed dataset will be publicly available when the paper is published.[3]

[3] https://github.com/Francimaria/hate_speech_bias_feature/tree/main/dataset/UB

4 Experimental methodology

This section describes the experimental setup used in this study. We describe the datasets, pre-processing steps, feature extraction, training classifier, evaluation metrics, and parameter settings.

The Python programming language was used to conduct the experiments on a computer with an processor Intel Core i7-10510U CPU @ 1.80 GHz x 4, 15.3 GiB of memory, and an Intel Corporation UHD Graphics card.

4.1 Datasets

Table 4 summarises the datasets selected. We analyse three (WH, DV, and HE) widely used English Twitter datasets to evaluate the proposed methodology. Furthermore, considering that the test set from the original dataset can hold the same biased distribution as the training set and affect the bias evaluation [8], we use an unbiased dataset (**UB**) (described in Sect. 3.1) for bias evaluation because this dataset includes all identity terms in the same context, ensuring non-bias towards identity terms.

Waseem-Hovy (WH) [33]: The corpus has more than 16k samples collected from Twitter. The initial search used a list of potential hateful terms and phrases.[4] The authors manually annotated the dataset based on guidelines inspired by critical race theory. The annotation was reviewed by "*a 25-year-old woman studying gender studies and a non-activist feminist*". The dataset consists of tweets labelled as sexist, racist or neither.

Davidson (DV) [39]: The authors used a lexicon from *Hatebase.org* to search the tweet and extracted the timeline for each user. They then selected random samples, and the CrowdFlower (CF) workers manually annotated. They labelled the corpus as hate speech, offensive or neither (neither offensive nor hate speech). The authors instructed the CF workers to consider the words and the inferred context to avoid false positives in this process. The final dataset has resulted in 24,802 labelled tweets.

HatEval (HE) [35]: The HatEval dataset is a multilingual corpus for hate speech detection against women and immigrants. The authors used different gathering strategies based on previous studies proposed in the literature to collect the dataset. Figure Eight (F8) workers and two experts annotated the dataset labelled based on majority voting. The final dataset comprises 19,600 tweets, 6,600 for Spanish and 13,000 for English. However, we used only English tweets. The data was annotated based on three categories: first, Hate Speech (hateful and non-hateful); second, Target Range (individual target and generic target); and Aggressiveness (aggressive or non-aggressive).

We selected those datasets because they address different nuances of hate speech problems, such as sexism, racism, and xenophobia. Moreover, they have distinct data collection and annotation processes. Therefore, we can use a diverse set of datasets to evaluate the proposed methodology under different hate speech detection scenarios.

We conducted our experiments using stratified 5-fold cross-validation to divide the WH and DV datasets in 4 folds for training (Δ) and 1 fold for testing (τ). So, we used 15% of the training set as the validation set for the classifiers' parameter tuning. This strategy is used to compute the mean and standard deviation of the results achieved and thus helps us find more precise estimators of the model performance [3]. Moreover, we used the stratified version of cross-validation to ensure the proportion of each class is represented as in the original dataset across each fold to avoid class bias.

For the HE dataset, we used the original training (Δ), validation, and testing (τ) division used in the competition [35]. For the unbiased dataset (UB), we used the complete dataset as a test set (Γ) to evaluate the bias on predictions.

4.2 Pre-processing

In the context of Twitter, the text often contains elements such as URLs, hashtags, slang, mentions, RT, etc. This content can raise noise in the classification task [12, 49, 50]. Therefore, we performed different pre-processing criteria to clean our model's input for clarity and generality. It includes: converting all text to lowercase, remove the mentions ("i.e.,*@user*"), URLs (which start with "*http*[*s*] : //"), RT symbols (Retweet), numbers, punctuation marks, stopwords, and redundant white spaces, and stemming the text to reduce word flexions.

4.3 Feature extraction

For feature extraction, we considered six methods in this study:

- **TF**: Term Frequency (TF) [51, 52], also called of count vectoriser, represents the textual features based on the occurrence and frequency of words in a document. This feature extraction method is relatively simple. However, in the case of large textual datasets, the representation matrix can become exceedingly sparse, necessitating a significant computational effort. Our work used the maximum features equal to 2000, as used in the literature [53]. We used the implementation from the Scikit-learn Python library [54].

[4] Terms queried for: "MKR", "asian drive", "feminazi", "immigrant", "nigger", "sjw", "WomenAgainstFeminism", "blameonenotall", "islam terrorism", "notallmen", "victimcard", "victim card", "arab terror", "gamergate", "jsil", "racecard", "race card".

Table 4 Summary of datasets

Name	Available	Tweets	Label (%)	Target	Annotator
WH	GitHub repository	16,906	Sexism (20%) Racism (12%) Neither (68%)	Sexism, racism	1
DV	GitHub repository	24,783	Hate (6%) Offensive (77%) Neither (17%)	General	3 or more
HE	GitHub repository	13,000	Hate (43%) Non-hate (57%)	Misogyny, xenophobia	3
UB	GitHub repository	10,728	Hate (76%)	General	–

- **TF-IDF**: Term Frequency-Inverse Document Frequency (TF-IDF) [55] is a widely used statistical feature representation technique from textual data. This method scores and weighs words that occur in a document. The primary objective is to identify the relevant and significant words that impact the document most meaning and relevance. The length of the feature vector depends on the document vocabulary size, and the representation matrix can become sparse as with the TF method. Our work used the maximum features equal to 2000, as used in the literature [53]. We used the implementation from the Scikit-learn Python library [54].
- **GloVe**: GloVe, an acronym for Global Vectors for Word Representation [56], learns word representations by incorporating global statistics (word-word co-occurrence counts). In essence, it is a global log-bilinear model with a weighted least-squares objective for the unsupervised learning of word representations. Our work used GloVe embeddings trained on Twitter data (2B tweets, 27B tokens, 1.2M vocab, uncased) with a feature dimension of 200. We used the implementation from the Zeugma library.[5]
- **FastText**: FastText model [57] learns word representations based on character n-grams, which assumes that each word is the sum of the n-grams vectors. Thus, considering subword information that helps the model build word vectors for out-of-vocabulary words. For the current work, we use the FastText embedding with a feature dimension of 300 (implementation from the Zeugma library) pre-trained with 1 million word vectors trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (16B tokens).
- **BERT**: BERT, an abbreviation for Bidirectional Encoder Representations from Transformers [58]. The BERT is a pre-trained embedding method defined in two models, BERT_{BASE} and BERT_{LARGE}, both with uncased (only lowercase letters) and cased versions. The BERT_{LARGE} model consist of 24 layers, 16 attention heads, and 340 million parameters and the BERT_{BASE} model consists of 12 layers, 12 self-attention heads, and 110 million parameters. We selected the pre-trained BERT_{BASE} uncased because the training process of a BERT model is computationally expensive. Furthermore, it has presented promising results for hate speech detection in [5, 12, 59]. We used the implementation from Transformers library [60] with a feature dimension of 768.
- **RoBERTa**: RoBERTa, an acronym to Robustly Optimised BERT Pre-training Approach [61]. RoBERTa is a language model developed based on BERT architecture. This model was designed to improve its results by adjusting key hyperparameters of the BERT model, such as longer sequences, changes in the length of batch size, and removal of the next sentence prediction objective. We selected the pre-trained RoBERTa_{BASE} for this study. We used the implementation from Transformers library [60] with a feature dimension of 768.

4.4 Training classifier

We selected various classification algorithms to evaluate the different feature extraction techniques. Our objective is to analyse different classifiers to investigate if the techniques' biased behaviour is generalised for a wide range of classification algorithms.

This study includes the following algorithms in the experiments: Support Vector Machine (SVM) [62], Logistic Regression Classifier (LR), Decision Tree Classifier (DT), Extreme Gradient Boosted Decision Trees (XGBoost) [63], Multi-Layer Perceptron Neural Network (MLP) [64], and Random Forest (RF).

[5] https://zeugma.readthedocs.io/en/latest/.

4.5 Evaluation

We assessed all methods using distinct evaluation metrics to provide different performance perspectives. The objective is to analyse the possible advantages and limitations of each technique. Table 5 summarises the selected metrics for bias and classification evaluation.

Regarding the unintended bias evaluation, we investigated different metrics widely used in the literature [9, 14]. These metrics measure the bias based on the outputs of the algorithms. We selected a threshold agnostic metric derived from the ROC-AUC (or AUC) metric [65], called, **subgroup AUC**. To facilitate the assessment of the bias in the context of our analysis, we measure the average across all identity terms. The equation for the subgroup AUC is defined in Eq. 1.

$$subgroup\ AUC = \frac{1}{|T|}\sum_{t\epsilon T} AUC\left(D_t^- + D_t^+\right) \tag{1}$$

where D_t^- denotes the negative examples (non-hate speech) and D_t^+ the positive one (hate speech) that mention the identity term $t\ \epsilon\ T$, where $T = [woman, ..., male]$ (complete list in Table 3) and $|T|$ denotes the number of identity terms in T.

The subgroup AUC measures the model performance of each subset that mentions a specific identity term, so we compute the average value of these results. Therefore, low results indicate that the model had difficulty distinguishing the labels of the samples in the context of identity terms.

In [65], the authors also proposed other metrics based on AUC with different objectives. But we decided to use only the subgroup AUC since this paper focuses on investigating the feature extraction biased behaviour against identity terms. Moreover, we measured the average value for the subgroup AUC across all identity terms.

In addition, we also used two metrics based on the **Error Rate Equality Difference** introduced in [8]. The **False Positive Equality Difference (FPED)** and **False Negative Equality Difference (FNED)** defined in Eqs. 2 and 3, respectively.

$$FPED = \frac{1}{|T|}\sum_{t\epsilon T}\left|FPR - FPR_t\right| \tag{2}$$

$$FNED = \frac{1}{|T|}\sum_{t\epsilon T}\left|FNR - FNR_t\right| \tag{3}$$

The FPED (or FNED) computes the sum of the difference between the False Positive Rate (*FPR*) or False Negative Rate (*FNR*) on the complete dataset and each subset containing a specific identity term, FPR_t and FNR_t. As for the AUC subgroup, we also calculate the average value to normalise the metric values between 0 and 1. To facilitate the understanding and contrast of different metrics.

The FPED and FNED measure the bias based on the error rate equality differences. Therefore, a model without unintended bias is expected to present similar values across all terms, where $FPR = FPR_t$ and $FNR = FNR_t$ for all identity terms. The wide divergence in these values across the identity terms suggests a high unintended bias, so the best result is zero.

On the other hand, a partial objective of this experiment is to evaluate the classification performance. Therefore, we evaluated the general performance of the models using the macro F-score and area under the ROC curve (AUC).

F1, or F-score, is measured based on the precision and recall harmonic mean, which are defined as in Eqs. 4 and 5. The number of instances correctly classified is defined as TP (True Positives) and TN (True Negatives). In contrast, FP (False Positives) and FN (False Negatives) represent the number of those incorrectly classified. Then, F1 can be defined as in Eq. 6.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = 2 \cdot \frac{\mathrm{Pr}\,ecision \times \mathrm{Re}call}{\mathrm{Pr}\,ecision + \mathrm{Re}call} \tag{6}$$

The F1, usually in multiclass problems, can be aggregated using micro or macro averages. In this paper, we selected the macro-average due to the imbalance nature observed in the hate speech datasets evaluated. In imbalanced datasets, the micro-averaging can mask the model performance for minority classes [66].

The AUC is computed as the area underneath the receiver operating characteristic curve. This probability curve plots the True Positive Rate (synonym for recall)

Table 5 Summary of the selected metrics

Evaluation	Metric	Meaning
Bias	Subgroup AUC	Compute the AUC from examples with identity terms
	FPED	False-positive equality difference
	FNED	False-negative equality difference
Classification	F1	Harmonic mean of the precision and recall
	AUC	Area under the ROC curve

Table 6 Enumeration of parameters used throughout the experiments

Method	Hyperparameters	Library
SVM	Kernel = [linear, rbf]	Sklearn v1.2.2[9]
LR	Penalty = [l1, l2]	Sklearn v1.2.2
MLP	Activation = [relu, logistic]	Sklearn v1.2.2
DT	Criterion = [gini, entropy]	Sklearn v1.2.2
XGBoost	n_estimators = [50,100]	Xgboost 1.7.5[10]
RF	n_estimators = [50,100]	Sklearn v1.2.2

against the False Positive Rate (FPR), defined in Eq. 7, at various threshold values from 0 to 1. The AUC is designed for binary classification problems. However, we can use it for multiclass problems using the One-vs-Rest technique. It computes the AUC of each class against the rest.

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

4.6 Parameters setting

Table 6 presents the parameters considered in this study for each classification model. We selected the best set of hyperparameters using grid-search and the macro F1-score as evaluation metric. The classifier was trained with the training set, and its performance was measured with the validation set. The column Library defines the library of each model and its version. The parameters for each classifier are in the Github repository.[6]

Regarding the MLP, it requires defining the network architecture. Therefore, we use a standard architecture with a single hidden layer containing 100 neurons as in [3].[7][8].

5 Experimental results

This section presents the experimental results aiming to answer the research questions defined in the Introduction section. The experiments evaluate the feature extraction techniques to analyse the unintended gender bias on the predictions using an unbiased test set (Sect. 5.1) and to investigate the bias impact on the classification performance using the test set (Sect. 5.2). In addition, the results with the standard deviation for all metrics are available in the supplementary material.

[6] https://github.com/Francimaria/hate_speech_bias_feature

[7] Scikit-learn Python library [54],

[8] https://xgboost.readthedocs.io/en/stable/install.html

5.1 Unintended gender bias

To answer the research question **RQ1** - Does the choice of the feature extraction technique impact the presence of unintended gender bias on the model predictions? - for each dataset, we compared the results of the feature extraction techniques using the unbiased test dataset. As mentioned previously, the unbiased test dataset uses the strategy of identity term templates to generate a data sample where all identity terms appear in the same context to evaluate the unintended bias from identity terms.

The unbiased test dataset was labelled as hate and non-hate. Therefore, to analyse the unintended bias metrics, we consider the predictions of "racism" and "sexism" as "hate" and "neither" as "non-hate" for the WH dataset. For the HE dataset, we did not perform modifications. For the DV dataset, we assume "hate" and "offensive" as "hate" and "neither" as "non-hate", as in related work [5].

In all tables, we abbreviated the name of the classifier as XGB – XGBoost. We highlighted the best results in bold for each classifier and underlined ties. For each dataset, we compare the feature extractors per classification model using the Wilcoxon statistical test, and the significantly better result is marked with $*$. The significance level adopted was 0.05. We selected the Wilcoxon statistical test because, as stated in [67], this test is robust for pairwise comparison between models.

Table 7 presents the results obtained with the FNED metric, which measures the bias based on the false negative rate, in which the closer the result is to zero, the lower the bias. For the HE dataset, it is important to note that the FastText demonstrated more biased behaviour when combined with MLP, which obtained 0.214, and the TF with DT achieved 0.223. For the WH dataset, the classifiers presented more biased behaviour with GloVe and FastText, especially the LR classifier, which found results bigger than 0.20 when combined with these feature extractors. Moreover, for WH dataset, presented more bias on prediction with five of the six classifiers analysed. For the DV dataset, TF and TF-IDF presented more biased results for the DV dataset when combined with LR, SVM, and MLP, finding results bigger than 0.20. These results evidenced that specific hate speech samples were considered non-hate speech when the samples contained some identity terms but not others.

Table 8 presents the results obtained with the FPED metric, which measures the bias based on the false positive rate. For this metric, as for FNED, results closer to zero present less bias. The BERT and RoBERTa presented less biased behaviour for all datasets evaluated for most classifiers. These results were statistically better for the WH and DV datasets. As for the metric FNED, the DT classifier

presented more bias in predictions for the HE dataset with the TF and TF-IDF (0.228 and 0.237) and the MLP with TF-IDF (0.190). Contrasting these results with those obtained through BERT and RoBERTa with DT, MLP, and RF, it is possible to note that these feature extractors present almost twice the result for HE and WH datasets. These results evidenced that TF and TF-IDF, combined with DT, MLP, and RF, consider more non-hate samples as hate when the samples contain some identity terms but not others.

Table 9 presents the results obtained with the subgroup AUC metric that measures the classifier performance in the context of identity terms. For the HE dataset, FastText presented the best results for SVM, XGB, and RF. Moreover, it is relevant highlight for this dataset GloVe with MLP (0.572) found better results than more complex more models, such as BERT and RoBERTa. For the WH dataset, BERT presented less biased behaviour for DT, SVM, and RF classifiers. However, it was statistically better only for the SVM classifier. The best results for the DV dataset were found with GloVe for SVM, XGB, and RF and with BERT for LR and MLP. In contrast with FNED and FPED, which analyse whether the model presents different performance among the identity terms, this metric measures the average model effectiveness in the samples evaluated with all identity terms.

In addition, it is relevant to notice that most of the results from the Subgroup AUC metric (Table 9) are close to 0.5, meaning that the algorithms had difficulty classifying the examples with identity terms. However, as previously mentioned (see Sect. 4.1), the unbiased test dataset used to evaluate the unintended bias comprises different cases of hate speech, the majority challenging cases for classification models. We obtained the best results with the models trained with the HE dataset. These results also can evidence a context-dependence of these models.

Based on all the evidence presented above, we can answer the research question **RQ1: Yes, the choice of the feature extraction technique impacts the presence of unintended gender bias on the model predictions**. We could verify that some classifiers presented more bias on predictions with some feature extraction techniques. For instance, the DT using the TF and TF-IDF as a feature extractor found a result higher than 0.20 with the bias metric FNED and using BERT, the same classifier found results lower than 0.06 for the HE and WH dataset (see Table 7). For this metric, the ideal value is zero, so the higher the value, the more bias. TF and TF-IDF are textual features that score and weight words based on their occurrence and frequency in a document. Thus, it may lead to a bias in relation to terms that are merely common in the dataset rather than truly informative for classification.

In addition, we can also answer the research question **RQ2 – Do feature extraction techniques tend to present bias when dealing with different datasets?** – The results in Tables 7, 8, and 9 endorse the need to properly and wisely select the feature extraction technique for each dataset matters for the effectiveness of the unbiased behaviour on the model predictions. The BERT and RoBERTa as input vectors achieved the best results for the FPED and FNED metrics with most classification models. However, it presented the best result only for some classifiers for the Subgroup AUC and in most cases, the results were not significantly better. Moreover, the analysis with the metric subgroup AUC showed different performances of the feature extractors in distinct datasets.

5.2 Classification performance

To answer the research question **RQ3 - Can the bias affect the performance of the models?** - for each dataset, we compare the results of each feature extraction technique with different classifiers and contrast them with the results in Sect. 5.1. We then aim to answer if the bias in the model predictions impacts the classification performance.

Tables 10 and 11 present the AUC and macro F1 metric results. For the HE dataset, the classification models presented the best AUC performance with GloVe for the DT and RF, FastText for the SVM and MLP, RoBERTa for the LR, and BERT for the XGB. With F1, FastText presented the best results for the LR, SVM, XGB, and RF, while GloVe with DT and MLP. In contrast with the results obtained with the bias metrics evaluated in Table 9, GloVe with DT presented more bias on prediction than the other feature extractors for the subgroup AUC metric, finding results under 0.5. The TF-IDF presented the best classification performance with both metrics for the WH dataset with the DT, SVM, and RF, while FastText with XGB. We achieved the best classification performance for the DV dataset using TF with all classifiers for the macro F1 metric. This feature extractor also presented more bias on predictions than FastText, BERT, and RoBERTa for different classifiers with the metrics FPED and FNED, as shown in Sect. 5.1.

Based on all the above evidence, we can answer the research question **RQ3: It depends on the analysed dataset**. For the HE, the feature extraction techniques that present more bias on predictions also present the best classification performance. We can infer that the test dataset can follow the same biased behaviour as the training set and influence these results, similar to the conclusions in [8]. Therefore, evaluating the model with an unbiased test is relevant and can help investigate different insights into the problem.

Table 7 Results obtained using FNED bias metrics for all datasets

Feature	LR	DT	SVM	XGB	MLP	RF
(a) HE dataset						
TF	**0.041**	0.223	**0.025**	0.000	0.204	0.111
TF-IDF	0.142	0.227	0.154	0.000	0.231	0.122
GloVe	0.175	**0.034**	0.139	0.105	0.158	0.052
FastText	0.132	0.065	0.137	0.082	0.214	0.062
BERT	0.131	0.037	0.069	0.042	0.125	0.026
RoBERTa	0.053	0.037	0.069	0.031	**0.115**	**0.016**
(b) WH dataset						
TF	0.084	0.215	0.098	0.000	0.167	0.181
TF-IDF	0.193	0.209	0.128	0.000	0.238	0.204
GloVe	0.250	0.069	0.198	0.128	0.138	0.067
FastText	0.200	0.084	0.159	0.142	0.174	0.085
BERT	0.131	0.051	**0.057***	0.028	0.113	**0.017**
RoBERTa	**0.083**	0.051	0.078	0.036	**0.098**	0.020
(c) DV dataset						
TF	0.237	0.107	0.130	0.000	0.245	0.081
TF-IDF	0.132	0.075	0.215	0.000	0.217	0.079
GloVe	0.152	0.058	0.133	0.084	0.129	0.106
FastText	**0.067**	0.057	0.078	0.060	0.097	0.074
BERT	0.105	**0.036**	**0.065***	0.037	**0.081**	0.050
RoBERTa	0.101	0.051	0.103	0.068	0.124	**0.041***

The table shows the average obtained from the k-fold for each feature extractor combined with each classifier. The best feature extractor result for each classifier is boldfaced, and ties are underlined. Significantly better results are marked with ∗

Table 8 Results obtained using FPED bias metrics for all datasets

Feature	LR	DT	SVM	XGB	MLP	RF
(a) HE dataset						
TF	0.028	0.228	**0.009**	0.000	0.173	0.103
TF-IDF	0.095	0.237	0.096	0.000	0.190	0.118
GloVe	0.182	0.053	0.103	0.072	0.108	0.036
FastText	0.125	0.052	0.097	0.068	0.174	0.049
BERT	0.094	**0.035**	0.028	0.018	0.067	0.015
RoBERTa	**0.021**	0.047	0.031	0.025	**0.064**	**0.012**
(b) WH dataset						
TF	0.092	0.212	0.103	0.000	0.167	0.166
TF-IDF	0.184	0.209	0.126	0.000	0.217	0.194
GloVe	0.251	0.078	0.215	0.117	0.138	0.065
FastText	0.232	0.094	0.162	0.151	0.171	0.083
BERT	0.112	**0.043**	**0.024***	0.016	0.095	**0.013***
RoBERTa	**0.070***	0.055	0.070	0.039	**0.088**	0.023
(c) DV dataset						
TF	0.243	0.115	0.123	0.000	0.263	0.077
TF-IDF	0.138	0.076	0.212	0.000	0.232	0.077
GloVe	0.206	0.072	0.160	0.106	0.151	0.115
FastText	**0.084**	0.062	0.105	0.065	0.120	0.078
BERT	0.117	**0.045**	**0.081***	0.048	**0.102**	0.064
RoBERTa	0.128	0.064	0.127	0.077	0.163	**0.047***

The table shows the average obtained from the k-fold for each feature extractor combined with each classifier. The best feature extractor result for each classifier is boldfaced, and ties are underlined. Significantly better results are marked with ∗

6 Discussion

We evaluated the unintended gender bias from TF, TF-IDF, BERT, RoBERTa, GloVe, and FastText in the predictions of six state-of-the-art machine learning classifiers in hate speech datasets. Based on the proposed analysis, we identified three main aspects: (1) feature extractor choice matters from a biased perspective, (2) training and testing based on the same dataset cannot properly estimate the bias in the predictions, and (3) the bias influence is dataset-dependent in the classification performance. Section 6.1 presents the models execution time evaluation, Sect. 6.2 discusses the overall relationship between bias and classification performance metrics, and a more profound analysis using the classifier's AUC and Subgroup AUC is addressed in Sect. 6.3.

6.1 Models execution time evaluation

This section analyses the average training time for each feature extraction method, including the training time and the representation step. The analysis was performed on the HE dataset, and we executed each experiment five times. Table 12 presents the execution time (in seconds) for each feature extraction method in each set. The BERT and RoBERTa presented a higher execution time for the train, test, and validation (val) sets. These results are due to the higher complexity of these models in contrast with TF, TF-IDF, GloVe, and FastText. Furthermore, the execution time of BERT and RoBERTa were similar, as expected, considering that they have similar architectures.

We also calculate the classifiers' training time without considering the representation step to make the models comparable. Table 13 presents the model's execution time evaluation for the classification step. The classifiers presented the most cost-effective with GloVe as a feature extractor for five of the six classifiers analysed.

6.2 Classification performance metrics versus unintended bias metrics

This analysis was performed based on five metrics: AUC and macro F1 for performance evaluation and FNED, FPED, and Subgroup AUC for bias evaluation. As

Table 9 Results obtained using Subgroup AUC bias metrics for all datasets

Feature	LR	DT	SVM	XGB	MLP	RF
(a) HE dataset						
TF	0.508	0.518	0.511	0.502	0.519	0.499
TF-IDF	0.529	0.517	0.541	0.502	0.531	0.517
GloVe	**0.554**	0.498	0.547	0.520	**0.572**	0.518
FastText	0.552	0.532	**0.553**	**0.538**	0.565	**0.528**
BERT	0.515	**0.533**	0.530	0.533	0.533	0.516
RoBERTa	0.523	0.516	0.528	0.515	0.546	0.508
(b) WH dataset						
TF	0.497	0.508	0.491	0.500	0.493	0.492
TF-IDF	0.505	0.507	0.502	0.500	0.499	0.498
GloVe	0.498	0.502	0.490	**0.513**	0.480	0.501
FastText	**0.518**	0.507	0.504	0.503	**0.517**	0.501
BERT	0.510	**0.510**	**0.515***	0.508	0.513	**0.504**
RoBERTa	0.508	0.504	0.501	0.499	0.505	0.497
(c) DV dataset						
TF	0.533	**0.530**	0.530	0.508	0.535	0.518
TF-IDF	0.516	0.508	0.531	0.508	0.534	0.500
GloVe	0.524	0.515	**0.538**	**0.525**	0.529	**0.531**
FastText	0.500	0.505	0.511	0.495	0.515	0.507
BERT	**0.534**	0.525	0.526	0.523	**0.551**	0.521
RoBERTa	0.495	0.509	0.497	0.507	0.526	0.500

The table shows the average obtained from the k-fold for each feature extractor combined with each classifier. The best feature extractor result for each classifier is boldfaced, and ties are underlined. Significantly better results are marked with $*$

Table 10 Results obtained using AUC for all datasets

Feature	LR	DT	SVM	XGB	MLP	RF
(a) HE dataset						
TF	0.584	0.536	0.616	0.502	0.588	0.555
TF-IDF	0.626	0.538	0.638	0.518	0.591	0.548
GloVe	0.599	**0.562**	0.626	0.599	0.611	**0.647**
FastText	0.622	0.555	**0.648**	0.623	**0.647**	0.646
BERT	0.626	0.559	0.638	**0.624**	0.605	0.624
RoBERTa	**0.632**	0.529	0.631	0.590	0.617	0.618
(b) WH dataset						
TF	**0.901**	0.761	0.887	0.832	0.878	0.891
TF-IDF	0.899	**0.770**	**0.898***	0.809	**0.887**	**0.901***
GloVe	0.862	0.659	0.885	0.840	0.871	0.833
FastText	0.864	0.653	0.885	**0.841**	0.884	0.829
BERT	0.867	0.630	0.870	0.828	0.864	0.813
RoBERTa	0.871	0.652	0.873	0.840	0.875	0.834
(c) DV dataset						
TF	0.924	**0.800***	0.906	0.899	0.903	0.915
TF-IDF	**0.933***	0.783	**0.917***	0.899	0.900	**0.916**
GloVe	0.913	0.672	0.908	0.859	0.911	0.856
FastText	0.903	0.655	0.899	0.863	**0.915**	0.850
BERT	0.875	0.610	0.862	0.806	0.870	0.785
RoBERTa	0.890	0.630	0.877	0.833	0.900	0.827

The table shows the average obtained from the k-fold for each feature extractor combined with each classifier. The best feature extractor result for each classifier is boldfaced, and ties are underlined. Significantly better results are marked with $*$

previously mentioned, the feature extractors that performed well regarding FNED and FPED had similar false-negative and -positive rates for different identity terms. On the other hand, the models with higher Subgroup AUC scores found less difficulty in classifying samples containing identity terms. Figure 3 shows the results for all metrics for each dataset. This graphic represents the number of times each feature extractor wins for each metric independent of the six classifiers evaluated; so, the maximum number is six. In the case of ties, all who tie are considered winners.

As we can observe from the results reported in Fig. 3, none of the feature extractors achieved the best results for all metrics. In addition, in some cases, the feature extractor that found the best overall classification performance also presented more bias on prediction.

For instance, FastText presented more bias on prediction for the HE dataset than the other feature extractors for the FNED and FPED metrics, even though it had achieved better results than the other feature extractor techniques for the AUC and macro F1 metrics. These results suggest that when the model is trained using this feature extractor, it presents different performances for examples that mention distinct identity terms. As expected, these results also suggest that the test set may have followed the same biased distribution as the training set.

For the WH dataset, the GloVe, in contrast with the other feature extractor analysed, presented the worst results for the bias metrics. In addition, it is relevant to note that BERT achieved the best result for the bias metrics and TF-IDF for the classification performance metrics.

Moreover, an interesting behaviour can be observed for the DV dataset. For the classification performance metrics, BERT, RoBERTa, and GloVe achieved presented worst results with the AUC and macro F1. In addition, for the bias metrics, RoBERTa also presents poor results, especially with the metric subgroup AUC. These results can evidence the classifiers' poor performance across all identity terms when combined with RoBERTa. On the other hand, GloVe presented better results only with subgroup AUC bias metric.

Table 11 Results obtained using macro F1 for all datasets

Feature	LR	DT	SVM	XGB	MLP	RF
(a) HE dataset						
TF	0.489	0.435	0.446	0.421	0.487	0.409
TF-IDF	0.495	0.462	0.475	0.420	0.504	0.420
GloVe	0.525	**0.541**	0.539	0.544	**0.527**	0.579
FastText	**0.566**	0.538	**0.555**	**0.571**	0.517	**0.589**
BERT	0.500	0.535	0.500	0.532	0.512	0.541
RoBERTa	0.502	0.500	0.493	0.515	0.460	0.520
(b) WH dataset						
TF	**0.749***	0.698	0.741	**0.701**	0.721	0.742
TF-IDF	0.730	**0.709**	**0.747**	0.700	0.725	**0.762***
GloVe	0.661	0.556	0.703	0.622	0.707	0.615
FastText	0.640	0.551	0.704	0.623	**0.726**	0.606
BERT	0.702	0.515	0.679	0.604	0.692	0.577
RoBERTa	0.684	0.545	0.695	0.629	0.688	0.598
(c) DV dataset						
TF	**0.707**	**0.693**	**0.706***	**0.703**	**0.698**	**0.715***
TF-IDF	0.702	0.681	0.681	0.696	0.691	0.682
GloVe	0.642	0.536	0.606	0.569	0.690	0.579
FastText	0.574	0.515	0.581	0.546	0.682	0.547
BERT	0.593	0.458	0.512	0.478	0.611	0.483
RoBERTa	0.571	0.487	0.543	0.499	0.629	0.489

The table shows the average obtained from the k-fold for each feature extractor combined with each classifier. The best feature extractor result for each classifier is boldfaced, and ties are underlined. Significantly better results are marked with $*$

Table 12 Models execution time evaluation in seconds for the representation step

Feature	Train	Test	Val
TF	0.238	**0.132**	0.172
TF-IDF	**0.198**	0.263	0.158
GloVe	0.420	0.156	0.063
FastText	0.370	0.139	**0.056**
BERT	398.348	140.224	46.328
RoBERTa	386.538	136.511	44.620

The feature extraction with the lowest execution time for each classifier is highlighted in bold

6.3 Case studies

This section evaluates the relationship between classification performance and the unintended gender bias metric. For this analysis, we consider two metrics, AUC and Subgroup AUC. However, the results with all combinations of metrics are available in the GitHub repository available in supplementary information.

Table 13 Models execution time evaluation in seconds for the classification step

Feature	LR	DT	SVM	XGB	MLP	RF
TF	0.465	5.712	410.910	18.700	51.795	4.288
TF-IDF	**0.347**	7.076	430.032	18.790	47.957	4.500
GloVe	0.348	**2.084**	**60.432**	**7.557**	**8.353**	**2.741**
FastText	0.386	2.800	83.708	13.282	19.754	3.303
BERT	4.256	7.354	158.321	37.468	18.666	5.407
RoBERTa	1.472	9.141	191.830	36.169	19.168	5.502

The feature extraction with the lowest execution time for each classifier is highlighted in bold

Figures 4, 5 and 6 present the results using the pair of metrics for the HE, WH, and DV datasets, respectively. For all datasets, the DT classifier achieved the worst results related to bias and classification performance. In contrast with the other datasets, for the HE dataset, the combination of classifiers and feature extractors presented less biased behaviour (for more details, see Sect. 5). Considering the trade-off between the bias metric (subgroup AUC) and the performance metric (AUC), FastText presented the best results for the HE and WH datasets when combined with SVM, RF, and MLP, while GloVe and BERT presented the best results for the DV dataset when combined with LR, MLP, and SVM.

7 Conclusion

In this study, we conducted a thorough analysis to explore how unintended gender bias from various feature extractors can influence classification performance. We carried out a wide-ranging experiment using six feature extractors, six classification methods, and three hate speech datasets. The results were assessed using multiple metrics to examine various aspects of the issue.

The outcomes of our analysis reveal that the feature extraction method plays a crucial role in determining the occurrence of unintended gender bias in model predictions. The experiments demonstrate that TF and TF-IDF exhibited more bias in predictions compared to BERT, FastText, GloVe, and RoBERTa. Consequently, selecting the best feature extraction technique for each dataset is essential to ensure that the model predictions remain unbiased. Our findings emphasise the significance of such analyses as a critical tool in model selection.

Researchers face complex difficulties due to the growing incidence of hate speech in modern media. Although

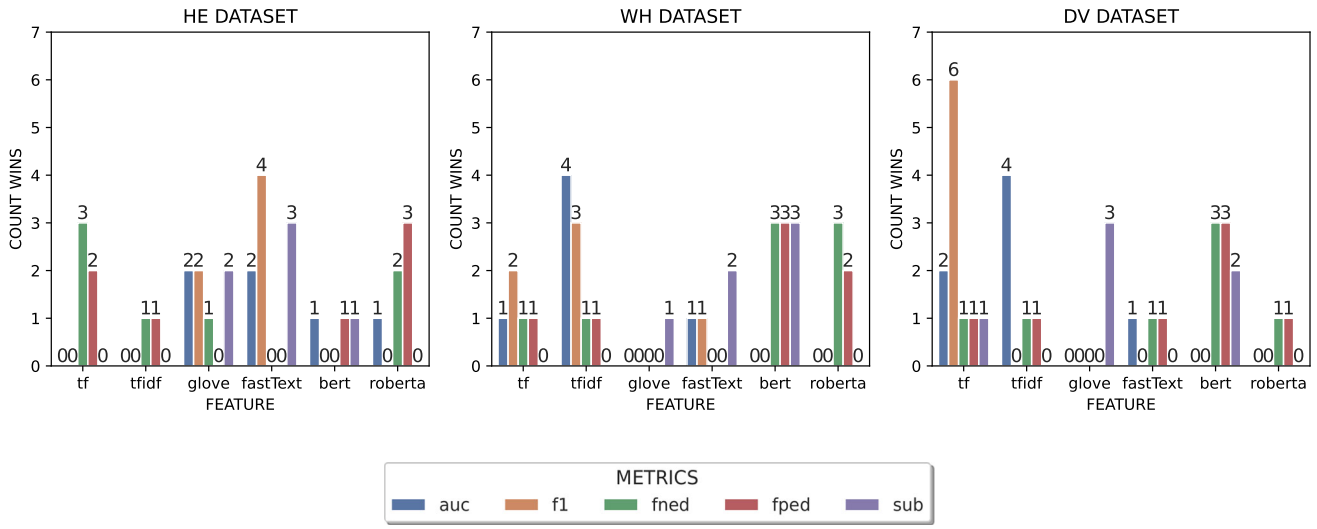

Fig. 3 Classification performance metrics versus unintended bias metrics. f1_score is macro F1-score, and subgroup denotes Subgroup AUC

significant progress has been made in automatic hate speech detection, hate speech detection methods face challenges and limitations. The results obtained in this paper demonstrate that some feature extractors can lead to gender bias. Moreover, there are numerous biases within the context of hate speech, including racial, cross-geographic, and political biases [30]. To address this issue, conducting a comprehensive analysis of various biases may provide valuable insights into developing procedures that improve the model's generalisation power.

The data collection and annotation can also impact the dataset's characteristics and lead to bias in the model. In the context of hate speech detection, the real-world distribution of non-hate is tiny, which makes collecting hate speech comments hard. The researchers usually use specific topics, hashtags, or users to increase the hate speech content [39]. Consequently, it introduces unintended biases into the dataset and the modelling pipeline. Therefore, in future work cross-dataset analysis can be used to identify and address dataset biases.

7.1 Future work

Detecting hate speech is a complex task, even for humans, due to its subjective nature. Therefore, hate speech detection methods have faced challenges and limitations, such as gender bias. In future work, we intend to investigate further the dataset annotation process to understand its influence

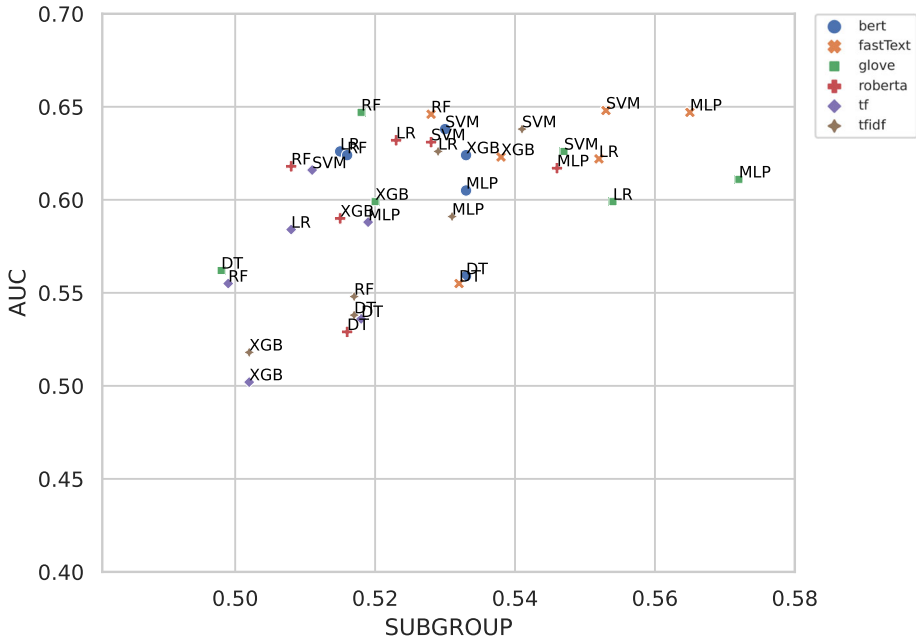

Fig. 4 AUC versus Subgroup AUC for the HE dataset

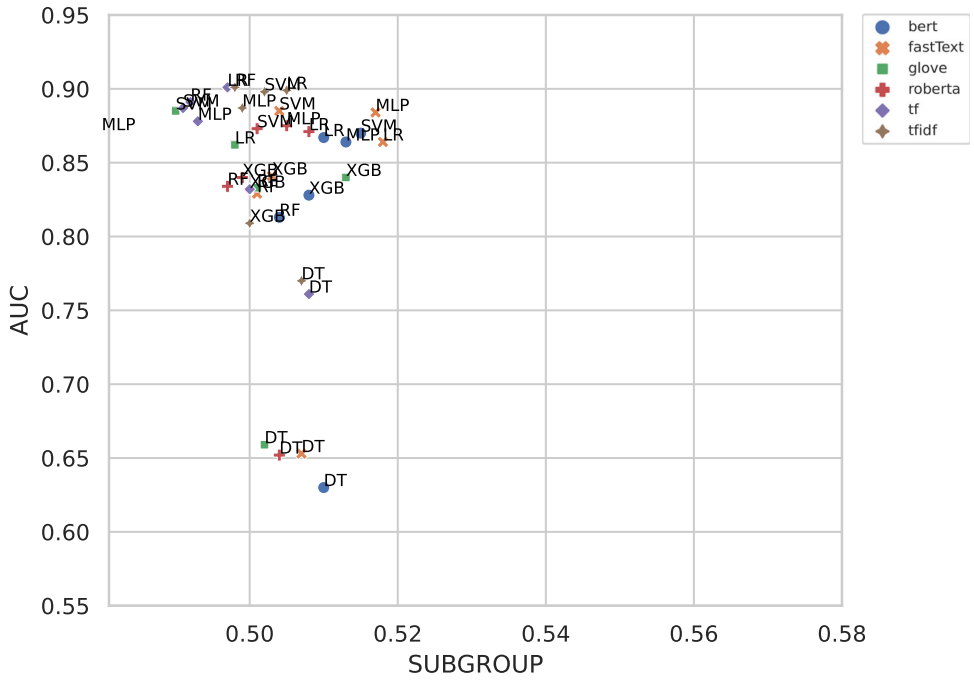

Fig. 5 AUC versus Subgroup AUC for the WH dataset

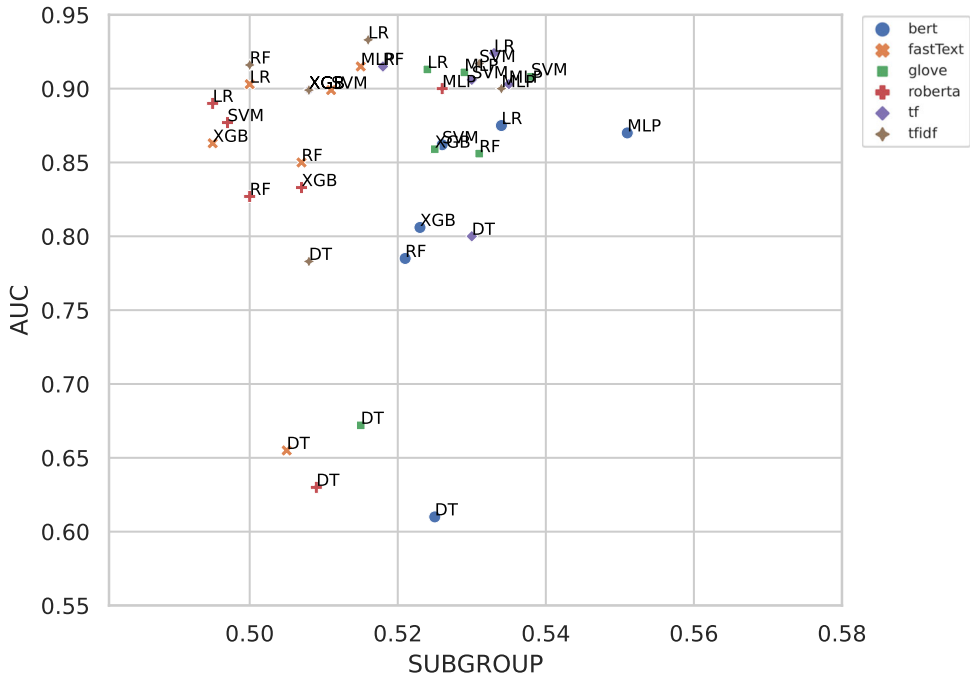

Fig. 6 AUC versus Subgroup AUC for the DV dataset

on the bias. In addition, we intend to extend this study to deep learning classifiers, such as CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), and so on. This analysis also can be extended to work with other identity problems, such as racial, religious, and xenophobic stereotypes.

Extracting relevant features from data is crucial for text classification using machine learning algorithms. Several methods have been proposed for feature extraction and significant progress has been made, as discussed in [16], including Bag-of-Words techniques, Large Language Models (LLMs), DNN. However, properly selecting the adequate method can be a complex task.

According to the experimental study conducted in [3], the combination of different methods for feature extraction can improve the performance of classification models. Moreover, our study evidenced that the feature selection matters in the context of unintended gender bias.

Therefore, multiple features can extract different abstractions of the data and introduce complementary information for the model to deal with inconsistencies.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-024-10841-8.

Acknowledgements This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* - Brazil (CAPES) - Finance Code 001 and Scholarship no. 88887.484211/2020-00.

Data availability All data supporting the findings of this study are available in the GitHub repository: https://github.com/Francimaria/hate_speech_bias_feature/tree/main/dataset. Table 4 shows the datasets.

Code availability Source code and supplementary data can be found in the GitHub repository: https://github.com/Francimaria/hate_speech_bias_feature.

Declarations

Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Fortuna P, Nunes S (2018) A survey on automatic detection of hate speech in text. ACM Comput Surv 51(4):1–30
2. Balouchzahi F, Shashirekha HL, Sidorov G, Gelbukh A (2022) A comparative study of syllables and character level n-grams for dravidian multi-script and code-mixed offensive language identification. J Intell Fuzzy Syst 43(6):6995–7005
3. Cruz R.M.O., de Sousa W.V., Cavalcanti G.D.C. (2022) Selecting and combining complementary feature representations and classifiers for hate speech detection. Online Soc Netw Med 28:100194. https://doi.org/10.1016/j.osnem.2021.100194
4. Kapil P, Ekbal A (2020) A deep neural network based multi-task learning approach to hate speech detection. Knowl-Based Syst 210:106458
5. Salminen J, Hopf M, Chowdhury SA, Jung S-G, Almerekhi H, Jansen BJ (2020) Developing an online hate classifier for multiple social media platforms. HCIS 10(1):1
6. Sengupta A, Bhattacharjee SK, Akhtar MS, Chakraborty T (2022) Does aggression lead to hate? detecting and reasoning offensive traits in hinglish code-mixed texts. Neurocomputing 488:598–617. https://doi.org/10.1016/j.neucom.2021.11.053
7. Zhao Z, Zhang Z, Hopfgartner F (2022) Utilizing subjectivity level to mitigate identity term bias in toxic comments classification. Online Soci Netw Med 29:100205
8. Dixon L, Li J, Sorensen J, Thain N, Vasserman L (2018) Measuring and mitigating unintended bias in text classification. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. AIES '18, pp. 67–73. ACM, New York, NY, USA . https://doi.org/10.1145/3278721.3278729
9. Nascimento FRS, Cavalcanti GDC, Costa-Abreu MD (2022) Unintended bias evaluation: an analysis of hate speech detection and gender bias mitigation on social media using ensemble learning. Exp Syst Appl 201:117032. https://doi.org/10.1016/j.eswa.2022.117032
10. Badjatiya P, Gupta M, Varma V (2019) Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In: The World Wide Web Conference, pp. 49–59. ACM, New York, NY, USA
11. Jahan MS, Oussalah M (2023) A systematic review of hate speech automatic detection using natural language processing. Neurocomputing 546:126232. https://doi.org/10.1016/j.neucom.2023.126232
12. Mozafari M, Farahbakhsh R, Crespi N (2020) Hate speech detection and racial bias mitigation in social media based on bert model. PLoS ONE 15(8):1–26
13. Sap M, Card D, Gabriel S, Choi Y, Smith NA (2019) The risk of racial bias in hate speech detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1668–1678
14. Park JH, Shin J, Fung P (2018) Reducing gender bias in abusive language detection. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2799–2804. ACL, Brussels, Belgium
15. Lee MSA, Singh J (2021) Risk identification questionnaire for detecting unintended bias in the machine learning development lifecycle. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. AIES '21, pp. 704–714. ACM, New York, NY, USA . https://doi.org/10.1145/3461702.3462572
16. Nascimento FRS, Cavalcanti GDC, Costa-Abreu MD (2023) Exploring automatic hate speech detection on social media: A focus on content-based analysis. SAGE Open 13(2):21582440231181310.https://doi.org/10.1177/21582440231181311
17. Senarath Y, Purohit H (2020) Evaluating semantic feature representations to efficiently detect hate intent on social media. In: 2020 IEEE 14th International Conference on Semantic Computing, pp. 199–202. IEEE, San Diego, CA, USA
18. Nobata C, Tetreault J, Thomas A, Mehdad Y, Chang Y (2016) Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, pp. 145–153. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE
19. Cao R, Lee RK-W, Hoang T-A (2020) Deephate: Hate speech detection via multi-faceted text representations. 12th ACM Conference on Web Science. WebSci '20. ACM, New York, NY, USA, pp 11–20
20. Founta AM, Chatzakou D, Kourtellis N, Blackburn J, Vakali A, Leontiadis I (2019) A unified deep learning architecture for abuse detection. In: Proceedings of the 10th ACM Conference on Web Science, pp. 105–114. ACM, New York, NY, USA
21. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, et al (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144

22. Karn AL, Karna RK, Kondamudi BR, Bagale G, Pustokhin DA, Pustokhina IV, Sengan S (2023) Customer centric hybrid recommendation system for e-commerce applications by integrating hybrid sentiment analysis. Electron Commer Res 23(1):279–314
23. Sun T, Gaut A, Tang S, Huang Y, ElSherief M, Zhao J, Mirza D, Belding E, Chang K-W, Wang WY (2019) Mitigating gender bias in natural language processing: Literature review. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1630–1640. ACL, Florence, Italy . https://doi.org/10.18653/v1/P19-1159
24. Dastin J (2018) Amazon scraps secret ai recruiting tool that showed bias against women. Ethics of data and analytics. Auerbach Publications, San Francisco, USA, pp 296–299
25. Deshpande KV, Pan S, Foulds JR (2020) Mitigating demographic bias in ai-based resume filtering. In: Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization, pp. 268–275. ACM, New York, NY, USA . https://doi.org/10.1145/3386392.3399569
26. Mazari AC, Boudoukhani N, Djeffal A (2023) Bert-based ensemble learning for multi-aspect hate speech detection. Cluster Comput 27:1–15. https://doi.org/10.1007/s10586-022-03956-x
27. Indurthi V, Syed B, Shrivastava M, Chakravartula N, Gupta M, Varma V (2019) FERMI at SemEval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in Twitter. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 70–74. Association for Computational Linguistics, Minneapolis, Minnesota, USA . https://doi.org/10.18653/v1/S19-2009 . https://aclanthology.org/S19-2009
28. Firmino AA, Souza Baptista C, Paiva AC (2024) Improving hate speech detection using cross-lingual learning. Expert Syst Appl 235:121115
29. Davani AM, Atari M, Kennedy B, Dehghani M (2023) Hate speech classifiers learn normative social stereotypes. Trans Assoc Comput Linguist 11:300–319
30. Garg T, Masud S, Suresh T, Chakraborty T (2023) Handling bias in toxic speech detection: a survey. ACM Comput Surv 55(13s):1–32
31. Şahinuç F, Yilmaz EH, Toraman C, Koç A (2022) The effect of gender bias on hate speech detection. Signal, Image and Video Processing, 1–7
32. Shen K, Ding L, Kong L, Liu X (2024) From physical space to cyberspace: recessive gender biases in social media mirror the real world. Cities 152:105149
33. Waseem Z, Hovy D (2016) Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: Proceedings of the NAACL Student Research Workshop, pp. 88–93. ACL, San Diego, California
34. Founta AM, Djouvas C, Chatzakou D, Leontiadis I, Blackburn J, Stringhini G, Vakali A, Sirivianos M, Kourtellis N (2018) Large scale crowdsourcing and characterization of twitter abusive behavior. In: Twelfth International AAAI Conference on Web and Social Media, pp. 491–500. AAAI Press, California, USA
35. Basile V, Bosco C, Fersini E, Nozza D, Patti V, Pardo FMR, Rosso P, Sanguinetti M (2019) Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In: Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, Minnesota, USA, pp. 54–63 . Association for Computational Linguistics
36. Salminen J, Almerekhi H, Milenkovic M, Jung S-g, An J, Kwak H, Jansen BJ (2018) Anatomy of online hate: Developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In: Proceedings of the International AAAI Conference on Web and Social Media, California, USA, pp. 330–339
37. Almerekhi, H., Kwak, H., Jansen, B.J., Salminen, J.: Detecting toxicity triggers in online discussions. In: Proceedings of the 30th ACM Conference on Hypertext and Social Media, pp. 291–292. ACM, New York, NY, USA (2019). https://doi.org/10.1145/3342220.3344933
38. Wulczyn E, Thain N, Dixon L (2017) Ex machina: Personal attacks seen at scale. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1391–1399. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE . https://doi.org/10.1145/3038912.3052591
39. Davidson T, Warmsley D, Macy M, Weber I (2017) Automated hate speech detection and the problem of offensive language. In: Eleventh International AAAI Conference on Web and Social Media. AAAI Press, Montreal, Canada
40. Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N, Kumar R (2019) Predicting the type and target of offensive posts in social media. In: Proceedings of NAACL-HLT, pp. 1415–1420. ACL, Minneapolis, Minnesota. https://doi.org/10.18653/v1/N19-1144
41. Golbeck J, Ashktorab Z, Banjo RO, Berlinger A, Bhagwan S, Buntain C, Cheakalos P, Geller AA, Gergory Q, Gnanasekaran RK (2017) *et al.:* A large labeled corpus for online harassment research. In: Proceedings of the 2017 ACM on Web Science Conference, pp. 229–233. ACM, New York, NY, USA . https://doi.org/10.1145/3091478.3091509
42. Gibert O, Perez N, Garcıa-Pablos A, Cuadros M (2018) Hate speech dataset from a white supremacy forum. In: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pp. 11–20. ACL, Brussels, Belgium. https://doi.org/10.18653/v1/W18-5102
43. Waseem Z (2016) Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In: Proceedings of the First Workshop on NLP and Computational Social Science, pp. 138–142. ACL, Austin, Texas
44. Toraman C, Şahinuç F, Yilmaz E (2022) Large-scale hate speech detection with cross-domain transfer. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 2215–2225. European Language Resources Association, Marseille, France . https://aclanthology.org/2022.lrec-1.238
45. Almatarneh S, Gamallo P, Pena FJR, Alexeev A(2019) Supervised classifiers to identify hate speech on english and spanish tweets. In: International Conference on Asian Digital Libraries, pp. 23–30. Springer, Berlin, Heidelberg . https://doi.org/10.1007/978-3-030-34058-2_3
46. Corazza M, Menini S, Cabrio E, Tonelli S, Villata S (2020) A multilingual evaluation for online hate speech detection. ACM Trans Internet Technol 20(2):1–22
47. Gitari ND, Zuping Z, Damien H, Long J (2015) A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering 10(4):215–230
48. Röttger P, Vidgen B, Nguyen D, Waseem Z, Margetts H, Pierrehumbert J (2021) *et al.:* Hatecheck: Functional tests for hate speech detection models. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), p. 41 . Association for Computational Linguistics
49. Asiri Y, Halawani HT, Alghamdi HM, Abdalaha Hamza SH, Abdel-Khalek S, Mansour RF (2022) Enhanced seagull optimization with natural language processing based hate speech detection and classification. Appl Sci 12(16):8000
50. DeSouza GA, Da-Costa-Abreu M (2020) Automatic offensive language detection from twitter data using machine learning and feature selection of metadata. In: 2020 International Joint Conference on Neural Networks, pp. 1–6. IEEE, Glasgow, UK

51. Farhangian F, Cruz RM, Cavalcanti GD (2024) Fake news detection: taxonomy and comparative study. Information Fusion 103:102140
52. Plaza-Del-Arco F-M, Molina-González MD, Ureña-López LA, Martín-Valdivia MT (2020) Detecting misogyny and xenophobia in spanish tweets using language technologies. ACM Trans Int Technol (TOIT) 20(2):1–19
53. Kumari K, Jamatia A (2022) An approach of hate speech identification on twitter corpus. In: International Conference on Frontiers of Intelligent Computing: Theory and Applications, pp. 115–125 . Springer
54. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
55. Shmueli G, Bruce PC, Yahav I, Patel NR, Lichtendahl KC Jr (2017) Data mining for business analytics: concepts, techniques, and applications in R. Wiley, USA
56. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543
57. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146
58. Devlin J, Chang M.-W, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186. ACL, Minneapolis, Minnesota
59. Risch J, Krestel R (2020) Bagging bert models for robust aggression identification. In: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pp. 55–61. ELRA, Marseille, France
60. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, Platen P, Ma C, Jernite Y, Plu J, Xu C, Scao TL, Gugger S, Drame M, Lhoest Q, Rush AM (2020) Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. ACL, Online
61. 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
62. Cortes C, Vapnik V (1995) Support-vector networks. Machine learning 20:273–297
63. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16, pp. 785–794. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/2939672.2939785
64. Aggarwal CC et al (2018) Neural networks and deep learning. Springer 10(978):3
65. Borkan D, Dixon L, Sorensen J, Thain N, Vasserman L (2019) Nuanced metrics for measuring unintended bias with real data for text classification. In: Companion Proceedings of the 2019 World Wide Web Conference, pp. 491–500. ACM, New York, NY, USA . https://doi.org/10.1145/3308560.3317593
66. Charitidis P, Doropoulos S, Vologiannidis S, Papastergiou I, Karakeva S (2020) Towards countering hate speech against journalists on social media. Online Soci Netw Med 17:100071
67. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30