Unintended bias evaluation: an analysis of hate speech detection and gender bias mitigation on social media using ensemble learning.

NASCIMENTO, Francimara, CAVALCANTI, George and DA COSTA ABREU, Marjory <http://orcid.org/0000-0001-7461-7570>

Citation:

NASCIMENTO, Francimara, CAVALCANTI, George and DA COSTA ABREU, Marjory (2022). Unintended bias evaluation: an analysis of hate speech detection and gender bias mitigation on social media using ensemble learning. Expert Systems with Applications, 201: 117032. [Article]

Unintended bias evaluation: An analysis of hate speech detection and gender bias mitigation on social media using ensemble learning

Francimaria Rayanne dos Santos Nascimento[a] (frsn2@cin.ufpe.br), George Darmiton da Cunha Cavalcanti[a] (gdcc@cin.ufpe.br), Márjory Da Costa-Abreu[b] (m.da-costa-abreu@shu.ac.uk)

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

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

Corresponding Author:

Francimaria Rayanne dos Santos Nascimento
Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Av. Jornalista Anibal Fernandes s/n, Recife, Brazil
Tel: +55 81 2126 8430
Email: frsn2@cin.ufpe.br

Unintended bias evaluation: An analysis of hate speech detection and gender bias mitigation on social media using ensemble learning

Francimaria RS Nascimento[a,*], George DC Cavalcanti[a], Márjory Da Costa-Abreu[b]

[a] *Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Av. Jornalista Anibal Fernandes s/n, Recife, Brazil*
[b] *Department of Computing, Sheffield Hallam University, Sheffield, UK*

Abstract

Hate speech on online social media platforms is now at a level that has been considered a serious concern by governments, media outlets, and scientists, especially because it is easily spread, promoting harm to individuals and society, and made it virtually impossible to tackle with using just human analysis. Automatic approaches using machine learning and natural language processing are helpful for detection. For such applications, amongst several different approaches, it is essential to investigate the systems' robustness to deal with biases toward identity terms (gender, race, religion, for example). In this work, we analyse gender bias in different datasets and proposed a ensemble learning approach based on different feature spaces for hate speech detection with the aim that the model can learn from different abstractions of the problem, namely **unintended bias evaluation metrics**. We have used nine different feature spaces to train the pool of classifiers and evaluated our approach on a publicly available corpus, and our results demonstrate its effectiveness compared to state-of-the-art solutions.

Keywords: Hate speech detection, Ensemble learning, Gender bias, Multi-features

[*] Corresponding author
Email addresses: `frsn2@cin.ufpe.br` (Francimaria RS Nascimento), `gdcc@cin.ufpe.br` (George DC Cavalcanti), `m.da-costa-abreu@shu.ac.uk` (Márjory Da Costa-Abreu)

1. Introduction

The popularisation of social media platforms has driven the exponential growth of the number of textual contents, making manual moderation of such content unsustainable (Cao et al., 2020). In particular, social media platforms allow users to express themselves freely, giving them a false sense of 'no man's land' and promoting a fertile ground for hate speech cases and offensive language usage. Despite its scarcity compared to other contents, the easy dissemination of abusive content on these platforms can be potentially harmful to target individuals, society, governments, and social media (Miškolci et al., 2020).

Hate speech is not a trivial phenomenon due to its subjective nature. Fortuna & Nunes (2018) defined it 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*". It should be noted that hate speech is usually expressed against a group or a community and may cause potential harm to individuals and society.

In this context, sexist hate speech has a large space on online social media, usually used against women (Chiril et al., 2020). This type of speech discriminates or harms against a person or group based on a person's gender. Sexism often is based on a belief in the superiority of a specific sex or gender. Its dissemination can be potentially harmful, and we cannot underestimate its impact on online social media. As an example, widespread sexist hate speech on social media can disseminate gender stereotypes.

Several works have proposed methods to perform automatic hate speech detection on benchmark datasets using Natural Language Processing (NLP) with classic Machine Learning (ML) (Salminen et al., 2020; Senarath & Purohit, 2020; Watanabe et al., 2018) and Deep Learning techniques (Zhang & Luo, 2019). So far, this task has been designed in the majority of cases using classic supervised machine learning approaches using metadata, user-based features, text mining-based features, such as lexical approaches, *n*-grams, bag-of-words, text embedding, sentiment, etc., which require a previous definition of the feature extraction methods employed. Deep learning models have explored these approaches for both feature extraction, and classification (Kapil & Ekbal, 2020; Santosh & Aravind, 2019). However, deep learning models require a significant amount of labelled data to perform well. Ensemble learning also has presented robust results, although few explored in the context of hate speech detection (Agarwal & Chowdary, 2021; Al-Makhadmeh & Tolba, 2019; Pitsilis et al., 2018). Even though different contributions have been dedicated to investigating these contents and presented high classification scores, the datasets and algorithms' potential biases did not receive attention in these researches.

The skewed distribution of specific terms in the training data can induce questionable trends for particular statements, and the representation learned by the model can not generalise well enough for practical use (Badjatiya et al., 2019; Dixon et al., 2018; Park et al., 2018). Hence, the supervised model can give unreasonable high hateful scores to clearly non-hateful text, such as "*You are a great woman*". The source of this bias can be associate with the highly frequent use of the word "*woman*" in hateful comments, which the model overgeneralised and associated this word with hateful comments. Dixon et al. (2018) stated this phenomenon as *false positive bias* and defined this behaviour of recognition models as *unintended bias.* In particular, they said: *"a model contains unintended bias if it performs better for comments containing some particular identity terms than for comments containing others"*.

Despite previous efforts, recent studies have investigated concerns about systems' robustness and discuss the impact of unintended bias in the dataset (Badjatiya et al., 2019; Dixon et al., 2018; Nozza et al., 2019). Some studies investigated bias regarding sensitive words (e.g., lesbian, gay, bisexual, transgender, trans, and so on) and try to mitigate bias based on balancing the training dataset (Dixon et al., 2018) or using replacement strategies (Badjatiya et al., 2019). Moreover, some works presented evidence of racial and dialect biases in several corpora annotated for toxic content, based on the correlation between words related to African American English dialect (AAE) and toxicity ratings (Mozafari et al., 2020; Sap et al., 2019). Gender stereotypes present in benchmark datasets are also a serious concern, in which a model can perform better with determinate identity terms than comments with others (Park et al., 2018). Therefore, it is essential to consider the bias in the datasets and algorithms for hate speech detection. These biases in datasets or classifiers lead to unfairness against target groups, which the classifiers are usually designed to protect.

In this work, we proposed an ensemble learning method based on different feature spaces for unintended gender bias mitigation in the context of hate speech detection on online social media. The model combines base classifiers, each trained with a different feature representation. Each feature extraction method captures a different abstraction about the data and can present a different classification performance. Therefore, even though one method of feature extraction might fail due to inconsistencies in the data samples (Sajjad et al., 2019) the system can still achieve a good performance as the system also considers other features. We analyse and mitigate gender bias in the datasets using bias-sensitive words and a replacement strategy to bias mitigation.

We believe that it will revolutionise the fight against gender-based hate speech if we can automatically detect messages of this nature and therefore deal with gender stereotypes present in the system. Thus, we analyse model biases, particularly gender identities (gender bias) present in hate speech datasets. We also propose an approach based on ensemble learning to classify hate speech on online social media and investigate the impact of gender bias in our ensemble method. Hence, this study aims to answer the following research questions: (1) Does the proposed multi-view stacked classifier combined with template-based mitigation outperform current techniques for hate speech detection in the context of unintended gender bias? (2) Can the bias mitigation method deal with gender biases in datasets without compromising the performance of the ensemble learning model?

In essence, the main contributions of this research are:

- Evaluation of a multi-view stacked classifier using nine different feature spaces combined with template-based mitigation for hate speech detection and gender bias mitigation.
- We perform our experiments in four real-world datasets in the context of gender bias mitigation.
- We explore the model's behaviour using three base classifiers while considering the unintended gender bias.

This work is organised as follows: Section 2 describes related work. Section 3 presents the problem statement, Section 4 describe the proposed methodology, Section 5 present the experimental setup, and Section 6 discusses the results. Section 7 concludes the work with the final remarks.

2. Related work

This section presents a comprehensive study of automatic hate speech detection and bias detection and mitigation in hate speech models and later specifically for gender-related hate speech.

2.1. Automatic hate speech detection

Hate speech is a complex problem that expresses the explicit intention to promote hatred or incites harm against a person or a targeted group. Several approaches have been proposed to hate speech detection on online social media platforms using classic machine learning methods, ensemble learning, and deep learning techniques. Twitter has attracted a significant part of the researches due to the increasing number of available data and free tools for data collection (Davidson et al., 2017; Waseem & Hovy, 2016; Waseem, 2016; Watanabe et al., 2018).

Classic supervised machine learning methods with different techniques for feature extraction have been frequently used in the literature for hate speech detection (Almatarneh et al., 2019; Santosh & Aravind, 2019). General feature representation methods of text mining have been successfully adapted to the problem of hate speech detection, such as Bag-of-Words (BoW) (Burnap & Williams, 2016; Nobata et al., 2016), n-grams (Corazza et al., 2020; Santosh & Aravind, 2019), dictionaries or lexical resources (Gitari et al., 2015; Mathew et al., 2019), etc. Regarding classification perspective, different algorithms have been employed, such as Logistic Regression (Davidson et al., 2017), Support Vector Machine (SVM) (Salminen et al., 2018), Random Forest (Elisabeth et al., 2020), Decision tree (Plaza-Del-Arco et al., 2020).

Davidson et al. (2017) addressed the problem of hate speech detection on Twitter, focusing on distinguishing between hate speech and offensive language. They exhibited that the presence of offensive words does not necessarily represent hate speech. The researchers evaluated their own hate speech dataset with the Logistic Regression classifier that achieved an F1-score of 0.90. However, the classifier had difficulty differentiating tweets labelled as hate speech, mislabeling almost 40%.

The deep learning techniques learn abstract feature representations from the data, and different models can be used as feature extractors and classifiers for hate speech detection. Recently, several works have applied pre-trained word embedding approaches, such as Word2Vec, GloVe, and FastText, because of the semantic information extracted from the text (Cao et al., 2020; Founta et al., 2019; Miok et al., 2019; Salminen et al., 2020). Regarding classification models, the most popular models are the Convolutional Neural Network (CNN) (Zhang & Luo, 2019; Del Vigna et al., 2017), Long Short-Term Memory Network (LSTM) (Cao et al., 2020; Zhang et al., 2018), Gated Recurrent Unit (GRU) (Corazza et al., 2020), and Bidirectional Encoder Representations from Transformers (BERT) (Mozafari et al., 2020).

Ensemble learning, or multiple classifier systems, have proven robust and improve the results of different classification tasks. In (Al-Makhadmeh & Tolba, 2019), (Pitsilis et al., 2018), and (Zimmerman et al., 2018), the researchers explored the combination of deep neural networks. Even though the models achieve slightly higher classification results than the current state-of-art, these techniques are time-consuming compared to the combination of other algorithms such as Logistic Regression and Decision Tree classifiers. In (Risch & Krestel, 2020), the researchers proposed an ensemble of BERT models based on bootstrap aggregation (bagging) and used soft majority voting to combine the predictions. Liu et al. (2019) investigated the hate speech detection problem as multi-task learning. For the classification task, they proposed a fuzzy ensemble approach. The experimental results showed that the proposed method outperforms SVM and deep neural networks, using embeddings features.

2.2. Bias detection and mitigation in hate speech models

Recently, great efforts have been taken to detect and mitigate bias in hate speech detection models. Dixon et al. (2018) investigated unintended bias in abusive detection models and evaluated the proposed method using a synthetic test set and an annotated dataset from Wikipedia Talk pages. The authors manually created a list of general identity terms (e.g., gay, transgender, feminist, and so on) to quantify the bias. Similarly, Nozza et al. (Nozza et al., 2019) also used a list of terms to quantify and mitigate unintended bias.

In (Badjatiya et al., 2019), the researchers proposed a two-stage method for unintended stereotype bias detection and mitigation. Firstly, they design different heuristics to identify a set of bias-sensitive words. Further, in the second stage, the researchers proposed replacement strategies in training data to mitigate the bias. The results show that the proposed procedures can reduce the bias without compromising the model performance significantly.

Bolukbasi et al. (2016) demonstrated gender stereotypes in word2vec (Mikolov et al., 2013) and introduced an algorithm to reduce gender biases in the word embeddings. Park et al. (2018) investigated gender bias on abusive language detection models. The authors used different methods to measure and debias gender bias, such as Debiased Word Embeddings, Gender Swap, and Bias fine-tuning strategies. Although the strategies to gender bias mitigation explored have reduced the performance of the classifiers, the authors stated that the method applied reduced the gender biases by 90-98%. In (Kiritchenko & Mohammad, 2018), the researchers evaluated gender and race bias in 219 automatic sentiment analysis systems from SemEval-2018 Task 1. The study provided an Equity Evaluation Corpus (EEC) to evaluate those systems' gender and racial bias.

Sap et al. (2019) investigated the unintended racial bias against speech produced by African Americans in two benchmark datasets widely used for hate speech detection. They used the AAE dialect to quantify the toxicity rating and stated that AAE tweets have a higher probability of being associate with offensive classes than the other tweets. In (Mozafari et al., 2020), the researchers addressed the problem of racial bias on the trained classifier. They introduced a transfer learning approach based on the BERT using the fine-tuning of the algorithm to mitigate racial bias. The results achieved demonstrated evidence of racial bias in the trained classifier against tweets written in AAE.

In this study, we investigated a list of potential bias-sensitive words (available in Section 4.1.1) and looked for disproportionate representations, focus on gender bias. We mitigated the gender bias based on a replacement strategy. Firstly, we evaluate the distribution of the bias-sensitive words in the hateful classes and overall. Then, we use a template strategy to replace the potential bias-sensitive words.

Despite different contributions for hate speech detection on online social media, it is relevant to highlight that a challenging task in hate detection is to select the best feature space for the classification. Furthermore, different features spaces can capture different abstractions of the problem. However, classification models for detecting hate speech using multi-view learning are seldom explored. In this paper, we proposed an ensemble learning method based on several feature spaces and different classifiers using public datasets to fill this gap. Moreover, we address unintended gender bias in the training set.

3. Problem formulation

In this section, we formulate the problem statement and describe the datasets used. Furthermore, we discuss the strategy employed for gender bias analysis and mitigation and ensemble learning for hate speech detection.

3.1. Dataset description

We analyse public annotated datasets for hate speech detection. We limited our data source using four criteria: (a) Twitter as the data source because it is the third most popular online social media (Antonakaki et al., 2021). Furthermore, Twitter is one of the most exploited sources for hate speech detection due to its policy on publicly available data and its free tools for data collection (Poletto et al., 2020). (b) The dataset was available at the time of performing research. (c) Written in the English language. (d) Described in previous studies. Thus, we obtained four datasets, described below and summarised in Table 1.

- **Waseem-Hovy (WH)** (Waseem & Hovy, 2016): The corpus contains data collected from Twitter over the two months. The authors collected 130k tweets and performing an initial manual search with potential terms or phrases[1] they considered hateful. The authors then manually annotated a subset of these data 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*" to check annotator bias. The original dataset consists of 16,906 tweets annotated as sexism, racism, or neither.

- **Waseem (WS)** (Waseem, 2016): This dataset explored an overlap of the dataset described in (Waseem & Hovy, 2016) to investigate the influence of annotator in the labelling of data. Thus, the authors relabelled 2,876 tweets. The authors provide 6,909 labelled tweets by annotators

[1] 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".

domain experts (feminist and anti-racist activists) and amateur recruited on CrowdFlower. The authors also included a new label (racism and sexism) to identify tweets with both types of hate speech. However, we do not consider the new label (both) because it represents only 1% of the samples.

- **Davidson (DV)** (Davidson et al., 2017): The authors used a hate speech lexicon from *Hatebase.org* to collect the corpus. The first sample was collected, resulting in 85.4 million tweets from the timeline of 33k Twitter users. Then, the authors selected a random sample of the 25k tweets using the lexicon. The CrowdFlower (CF) workers manually annotated the corpus as hate speech, offensive but not hate speech, or neither (neither offensive nor hate speech). In this process, the authors instructed the CF workers to think about the words and inferred context to avoid false-positive. Thus, it has resulted in a dataset with 24,802 labelled tweets.

- **HatEval (HE)** (Basile et al., 2019): collected the HatEval dataset for task 5 at SemEval-2019. They explored two categories of hate speech: misogyny and xenophobia. Different approaches were employed to compile potential hate speech and a lexicon of more frequent terms. The authors annotated the dataset from the crowdsourcing platform Figure Eight (F8) and two experts based on majority voting. The final dataset includes 19,600 tweets, 6,600 for Spanish, and 13,000 for English. The data was annotated based on three categories: Hate Speech (hateful or non-hateful); Target Range (individual or generic target); and Aggressiveness (aggressive or non-aggressive). However, we used only English tweets and the category Hate Speech.

In the first phase, we pre-process the tweets for noise reduction. It includes removing the URLs (which starts with “$http[s]://$”), the mentions (“i.e.,@$user$”), numbers, punctuation and stopwords, and make all text lowercase and have used stemming. Several works performed the pre-processing step

Table 1: Description of the datasets.

Dataset	Distribution	Number of instances	Label (%)	Target/Categories	Annotators
WH	GitHub repository	16,906	sexism (20%) racism (12%) neither (68%)	sexism, racism	1
WS	GitHub repository	6, 909	sexism (13%) racism (2%) neither(85%)	sexism, racism	4 or more
DV	GitHub repository	24,783	hateful (6%) offensive (77%) neither (17%)	general	3 or more
HE	GitHub repository	13,000	hateful (43%) non-hateful (57%)	misogyny, xenophobia	3

before the feature extraction (Dorris et al., 2020; Zhang et al., 2018; Watanabe et al., 2018; DeSouza & Da-Costa-Abreu, 2020) because the informal language used on social media and a diversity of elements of the tweets (for example, user names, URLs) can introduce noise and confuse a text classifier. Furthermore, these data pre-processing reduce the feature dimensionality of different feature extraction methods.

3.2. Unintended gender bias mitigation

Text-related models can extract strong insights about the significant association of determinate terms and a label. These associations can be positive in some cases and help the model improve performance. Nevertheless, it is not suitable for a hate speech detection model to depend on strong insight from individual word occurrences, but the combination of such words (Badjatiya et al., 2019). For example, "*Mary is a beautiful woman*". In this case, it might be beneficial for the classification model to use the knowledge extracted from the significant association between the "woman" and the "female" label. However, it is not good to relate the word "woman" with a "hateful" label, which might have unintended learned from the training pattern.

Hate speech detection models tend to present gender biases toward specific identity terms (Park et al., 2018). This issue can be motivated by the imbalanced nature of hate speech datasets and the disproportionate use of identity terms in hate speech sentences. For instance, some keywords such as “women” and “feminism” are highly associated with sexist comments in benchmarks datasets (Mozafari et al., 2020; Park et al., 2018). These factors can contribute to overfitting the original hate speech detection model. Consequently, the model can make generalisations such as associating the word “women” and a “hateful” label.

Different studies have investigated potential bias-sensitive words (BSWs). Dixon et al. (2018) manually create a list of 51 common identity terms and further analyse them from the training data. In (Badjatiya et al., 2019), the researchers also used the list of words proposed in (Dixon et al., 2018), besides, the authors proposed two new approaches to select the words, called Skewed Occurrence Across Classes (SOAC), which select the word that is used significantly in a particular class (’Hateful’); and Skewed Predicted Class Probability Distribution (SPCPD), which select the word based on the probability distributions. In this study, we investigate a list of bias sensitive words describe in Section 4.1 based on the literature (Nozza et al., 2019; Kiritchenko & Mohammad, 2018), because we focus on a particular bias (gender bias), and we investigate disproportionate distribution among labelled classes.

Regarding bias correction, different strategies can be employed, such as statistical correction (Dixon et al., 2018), model correction or post-processing (Mozafari et al., 2020; Park et al., 2018), and data correction (Badjatiya et al., 2019). The statistical correction includes techniques that try to distribute terms across the training set classes uniformly to balance the samples. In the model correction or post-processing, the mitigation of bias in the training set can either make during the model fine-tuning in the post-processing or by modifying the word embeddings. The data correction strategy consists of generalising some attributes that the model should not use to classify the sentence as hateful, thus reducing the number of information in the training set available to the classifier.

In this paper, we employed the data correction strategy to mitigate unintended bias, focusing on gender bias. The data correction was employed because: (i) In the statistic correction, selecting appropriate samples for bias correction is challenging. Furthermore, the balancing strategies used can introduce new skew distribution of terms in the training set; (ii) It does not require specific classifier models as the model correction strategy. Therefore, we can use it with any classification model; (iii) It has been successfully used as an unintended bias mitigation strategy for hate speech detection without compromising model performance (Badjatiya et al., 2019).

3.3. Ensemble learning

Hate speech datasets usually disproportionately use of determinate terms (say, bias sensitive words) highly correlated to minority class ('hateful'), enhancing bias stereotypes in the machine learning model. In this way, the classifier trained with biased data can deal with an increase in false-positive instances. Generally, different training data or feature spaces can emphasise different aspects of the problem, even with the same method, each learning algorithm presents its own weaknesses and strengths (Zhou et al., 2020).

Ensemble learning, or multiple classifier systems (MCS), is a machine learning technique that extracts the knowledge from the combination of several methods to increase the recognition accuracy in pattern recognition systems (Kuncheva, 2014). Bagging Algorithm (bootstrap aggregating) and Boosting are popular ensemble methods (Walmsley et al., 2018; Risch & Krestel, 2020). These algorithms are based on a homogeneous set of weak learners and build diversity by sub-sampling or re-weighting the existing training examples. However, these methods used the same feature spaces for all classifiers, and a challenging issue in the hate speech detection task is to determine the right features for classification (Fortuna & Nunes, 2018).

The hate speech detection on social media is a complex classification task in which different feature spaces can significantly change the performance. Moreover, a single classifier usually performs worse using a single feature space to handle inconsistencies, and various data (Sajjad et al., 2019). Several studies have argued that combining different feature spaces presents better results (Burnap & Williams, 2016; Watanabe et al., 2018), but combined spaces in a vector can deal with large dimensions.

Therefore, we choose to combine the classifiers using different feature spaces. Each feature space represents a different view of the problem to capture different abstractions about the data. Thus, although one method might fail due to data inconsistencies, the system still can consider other feature spaces and perform well. The multi-view learning optimises the model by learning one function based on different abstractions of the data that a single-view cannot comprehensively represent for all examples (Cruz et al., 2013; Zhao et al., 2017).

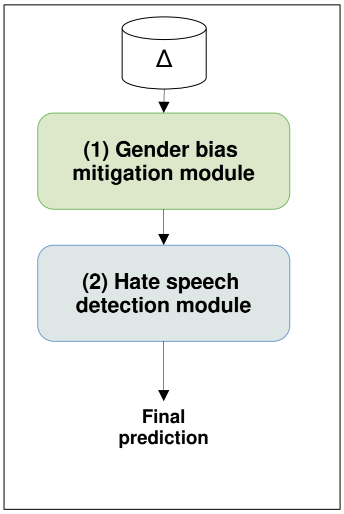

Δ
(1) Gender bias mitigation module
(2) Hate speech detection module
Final prediction

Figure 1: Overview of the proposed methodology. Δ is the training set.

4. Proposed methodology

This section introduces our methodology for hate speech detection and gender bias analysis and mitigation. The proposed model (Figure 1) consists of two main modules: **(1) Gender bias mitigation module** and **(2) Hate speech**

detection module. These two modules are described in Sections 4.1 and 4.2 respectively.

4.1. Gender bias mitigation

For gender bias mitigation, we investigate the disproportional distribution of specific terms on the datasets. Thus, we evaluate whether the model incorrectly predicted the sample's label based on specific words. The gender bias mitigation module is divided into two stages (see Figure 2): (1) Bias detection; and (2) Replacement of bias-sensitive words (BSWs).

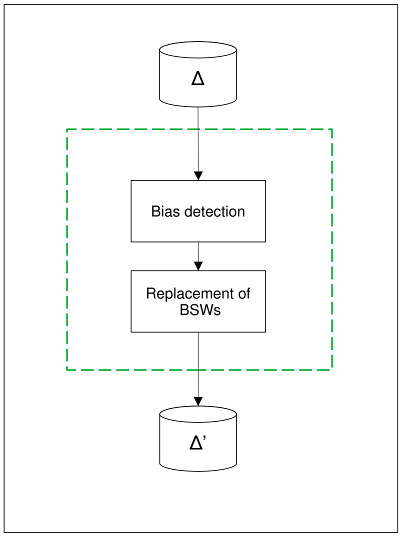


Figure 2: Gender bias mitigation module. Δ and Δ' are the training set before and after the gender bias mitigation module, respectively. BSWs: bias-sensitive words.

4.1.1. Bias detection

In this stage, we evaluate the distribution of the bias-sensitive words in hateful tweets and the entire dataset to investigate disproportionate representations. In order to simplify our analysis, we only consider a binary gender. Table 2 presents the list of nouns used in our study representing females and males.

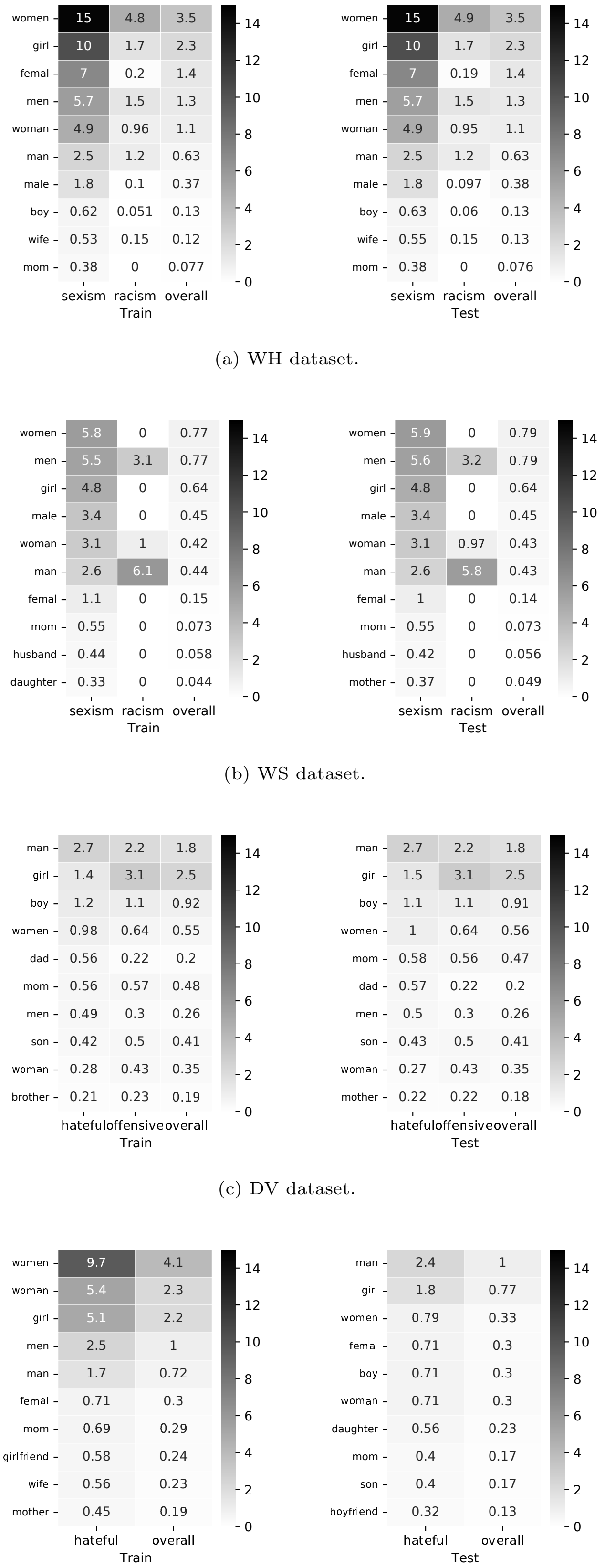
Different nouns, such as ‘she’, ‘her’, ‘he’, and ‘him’, were disregarded because of the pre-processing step as we remove stop-words and, consequently, exclude these words. Besides, the word ‘female’ was written as ‘femal’ because of the pre-processing step.

Table 2: Pairs of nouns representing a female or a male person used in this study.

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

Bias in the training sets is a serious concern, and the high scores due to it can overestimate the models (Wiegand et al., 2019). The significant occurrence of a word in a determinate class (say Hateful) can likely introduce unintended bias in the classifier model, which can probably learn this pattern and classify a sentence with that word into that class (Badjatiya et al., 2019). Therefore, we investigate the distribution of tweets with specific words. To do so, we compute $p(w|c)$ to measure the likelihood of the sentences in the class c contain the word w, and $p(w)$, which denote the likelihood of the sentences in the entire training/test set contain the word w. We analyse the disproportional distribution of determinate words in the hateful class and its overall distribution in the training and test sets. We used a cross-validation strategy to split three of the datasets in training, validation, and test sets (described in Section 5.1). Therefore, we present the average results for WH, WS, and DV datasets.

Figure 3 illustrates the top 10 average likelihood of word in the hateful comments and its overall likelihood with WH, WS, and DV datasets in the training and test sets, respectively. For the HE dataset, we have used the partitions provided in (Basile et al., 2019). The “class name”, e.g. sexism, racism, hateful, and so on, in the columns represents $p(w|c)$ and “overall” represents $p(w)$. We use a heat map with a grey colour bar where the legend indicates the likelihood values in colours.



(a) WH dataset.

(b) WS dataset.

(c) DV dataset.

(d) SE dataset.

Figure 3: The top 10 likelihood of tweets with the terms related to the gender terms in each dataset (WH, WS, DV, and SE). Sorted by the first column in descending order. Average results of cross-validation for WH, WS, and DV datasets.

Note that the terms such as 'women' and 'girl' appear more frequently in 'sexism' and 'hateful' comments than overall comments with WH, WS, and HE datasets. On the other hand, even though terms such as 'man' and 'girl' have been more frequent in 'hateful' comments with the DV dataset, the amount of hateful comments containing these terms is not disproportional to the other classes. The term 'man' also occurred more frequently in 'racism' comments with the WS dataset. These behaviours occurred for both training and test sets' samples for WH, WS, and DV datasets and only in training set in the HE dataset.

It is relevant to observe that the high disproportional distribution of the term 'women' usually occurs in datasets composed of tweets related to sexism or misogyny categories. Furthermore, even though the term 'man' occurred with a higher frequency with DV dataset in 'hateful' comments, the distribution of the term is much smaller than the word 'women' in other datasets.

4.1.2. Replacement of BSWs

In the replacement stage, we use a template strategy based on (Badjatiya et al., 2019), to replace the potential bias sensitive words (listed in Table 2) for the $< identity >$ tag and reduce gender bias introduced by these terms without compromise the model accuracy. This process masks some of the information available in the training set, inhibiting bias through these BSWs in the classification model. Different examples are illustrate in Table 3.

Table 3: Examples of sentences using the replacement strategy.

Tweet	**After replacement strategy**
RT @user: I'm not sexist, but girl fights just plain s*ck.	RT @user: I'm not sexist, but $< identity >$ fights just plain s*ck.
I'm not sexist but I hate serving women!	I'm not sexist but I hate serving $< identity >$!
This boy is an idiot followed by a bunch of idiots, this is a lack of leadership and direction.	This $< identity >$ is an idiot followed by a bunch of idiots, this is a lack of leadership and direction.

The idea is to reduce the differentiation of similar terms related to gender, such as ‘women’ and ‘men’. In the hate speech domain, the term ‘women’ is usually more frequently used than ‘men’, although they represent a similar group. Hence, the significantly high use of a term in a specific class (say Hateful) can likely introduce bias in the model.

4.2. Hate speech detection

The hate speech detection module consists of two phases (Figure 4):

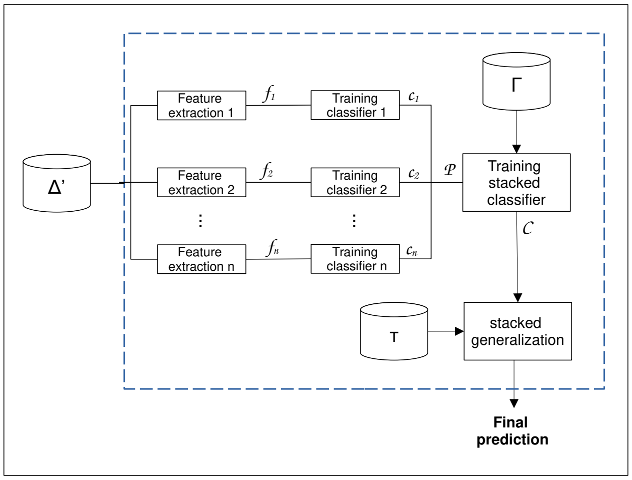


Figure 4: Hate speech detection module. Δ', Γ, and τ are the training after the bias mitigation module, validation, and test sets, respectively.

1. *Pool generation phase*, where the pool of classifiers P is generated using the training instances based on the combination of the classifier c_i with each feature of the feature space $F : \{f_1, f_2, ..., f_n\}$, composed of n feature extraction methods; Then, $P : \{f_1c_1, f_2c_2, ..., f_nc_n\}$.
2. *Combination phase*, where the predictions are combined using the stacked generalization to give the final prediction.

4.2.1. Pool generation

The pool generation phase is performed using a heterogeneous approach since each model is trained with different feature spaces. We investigated three different base classifiers: Logistic Regression (LR) (Davidson et al., 2017; Unsvåg & Gambäck, 2018), Decision Tree (DT) (Plaza-Del-Arco et al., 2020; Salminen et al., 2018), and Support Vector Machine (SVM) (MacAvaney et al., 2019; Senarath & Purohit, 2020). We have selected these models because they are frequently used for hate speech classification. Although recent works have addressed to use of Deep Learning models, these techniques are data hungry and time-consuming compared to algorithms such as LR and DT.

Each feature set f_i captures a different representation of the instances and can capture different properties about the dataset. Thus, using distinct sets of features, even though one instance representation might fail due to feature space, the model can consider the other data representation. We selected nine different feature representations currently used in the literature (Watanabe et al., 2018; Mozafari et al., 2020; Corazza et al., 2020; Cao et al., 2020; Fortuna & Nunes, 2018). Table 4 presents a summary of the feature methods used.

We selected three popular embedding methods, GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017), and BERT (Devlin et al., 2019), for f_1, f_2, and f_3 representations, respectively. These embedding methods had been used in several studies and proven effective for hate speech classification (Founta et al., 2019; Mozafari et al., 2020; Rizos et al., 2019; Sajjad et al., 2019). We choose the highest dimension (200) available for GloVe embedding trained over the Twitter data, as it produced the best results in the literature (Founta et al., 2019). For FastText embedding, the dimension is 300. The BERT has two models, and both have uncased (only lowercase letters) and cased versions, named BERT$_{BASE}$ and BERT$_{LARGE}$. The BERT$_{BASE}$ model contains 12 layers, 12 self-attention heads, and 110 million parameters and the BERT$_{LARGE}$ model has 24 layers, 16 attention heads, and 340 million parameters. In this work, we use the uncased version of the pre-trained BERT$_{BASE}$ model because training BERT is computationally expensive. Moreover, this model was effective in (Mozafari et al., 2020; Risch & Krestel, 2020; Salminen et al., 2020) for hate speech detection. For word embedding, we used the implementation from Transformers library (Wolf et al., 2020) for the BERT

Table 4: Feature extraction methods. The N is the number of different sequences of words/characters across the dataset.

Name	**Feature**	**Description**	**Vector dimension**
f_1	GloVe	Global Vectors for Word Representation. Pre-trained word embedding.	200
f_2	FastText	Pre-trained word embedding.	300
f_3	BERT	Bidirectional Encoder Representations from Transformers (BERT). Pre-trained embedding method.	768
f_4	TF	Term Frequency.	vocabulary size
f_5	TF-IDF	Term Frequency-Inverse Document Frequency.	vocabulary size
f_6	Word bi-grams	Count vector of word bigrams.	N sequences of two adjacent words
f_7	Word tri-grams	Count vector of word trigrams.	N sequences of three adjacent words
f_8	Char bi-grams	Count vector of character bigrams.	N sequences of two adjacent characters
f_9	Char tri-grams	Count vector of character trigrams.	N sequences of three adjacent characters

model and Zeugma library[2] for the other word embedding methods.

For the representations f_4 to f_9, we selected traditional feature extraction methods used for hate speech detection (Almatarneh et al., 2019; Corazza et al., 2020; Elisabeth et al., 2020; Salminen et al., 2020; Senarath & Purohit, 2020; Santosh & Aravind, 2019). These methods are based on the Bag-of-Words (BoW) technique. Thus, for the TF and TF-IDF, the feature vector's size used depends on the dataset vocabulary size. The n-grams technique combine the n adjacent items (words, characters, syllables, etc.) into a list of size N. We selected two approaches 'word n-grams' and 'character n-grams', with n equal 2 and 3. We used the implementations from scikit-learn (Pedregosa et al., 2011) for the extraction of these features.

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

It is relevant to highlight that the proposed framework is general to work with different features extraction methods and classifiers. The new techniques added to the system only need to be careful with the feature representation standard required by the classifier selected. Therefore, the proposed methodology can be continuously refined and improve the classification results with new features extraction methods and new classification models.

4.2.2. Combination phase

In the combination phase, the outputs of the classifiers are aggregated to obtain the final decision. The aggregation of the models can be performed based on different strategies, such as non-trainable, trainable and dynamic weighting (Cruz et al., 2018). In this work, we used a trainable aggregation strategy (Wolpert, 1992). The Stacked Generalization (or "stacking") consists of two levels of learning (Oriola, 2020). At the first level, different base learning algorithms learn from the training dataset. Each trained algorithm is then used to create a new dataset from the predictions collected using the validation dataset. Then, at the second level, another learning algorithm, also called meta-learner, is fitted based on the new dataset, which learns the aggregation function to provide the final prediction.

This architecture presents more robust than non-trainable ones as it does not require assumptions about the base model. Furthermore, the stacked generalisation does not use fixed rules and can be adjusted to the characteristics of the problem (Cruz et al., 2018). It has also been successfully used as a fusion rule in different classification problems, for instance, sentiment analysis (Al-Azani & El-Alfy, 2017) and hate speech classification (MacAvaney et al., 2019; Montani & Schüller, 2018; Paschalides et al., 2020). We use the StackedClassifier implementation provide by the Deslib Python library (Cruz et al., 2020).

5. Experimental setup

5.1. Datasets

The experiments were conducted using four public datasets for hate speech detection described in Section 3.1. We used stratified 5-fold cross-validation to evaluate the model. However, we need to partition the data into training, validation, and test because we used the validation set predictions for fitting the stacked model. The cross-validation scheme partitioned the dataset into 5 disjoint subsets (1 fold for test and 4 folds for training/validation). Then, we applied a stratified 4-fold cross-validation in the training/validation folds divided into 3 folds for training and 1 for validation. Resulting in 20 executions with 3 folds for training, 1 for validation, and 1 for test. We used a stratified division because this strategy preserves the prior percentage of samples for each class. For the HE dataset, we used the partition provided in (Basile et al., 2019) to compare the results with the literature easily.

5.2. Parameters setting

As stated in Section 4.2.1, we consider three base classifiers in this study: LR, SVM, and DT. These models were selected because they are the most used for hate speech detection. We trained each classifier with a different feature space resulting in nine models using each classifier.

Table 5: Hyperparameters of the models evaluated for all datasets.

Classifier	**Hyperparameter**
LR	'penalty': ['l1', 'l2']
DT	'criterion': ['gini', 'entropy'], 'splitter': ['best','random']
SVM	'kernel': ['linear', 'sigmoid', 'rbf', 'poly']

We used a grid search to select the best hyperparameters of the models for all datasets. Table 5 shows the hyperparameters evaluated for each classifier. We fitted the algorithms using the training set and evaluated their performance using the validation set. Then, we selected the hyperparameters setup with the best performance for each model based on the macro F1-score metric.

5.3. Evaluation Metrics

We evaluated the overall **performance** of the classification with the macro F1-score. The F1-score measures the harmonic mean of the Precision and Recall. The precision is computed by the number of samples correctly classified positives divided by the total of samples predicted as positives. The recall is the number of samples correctly classified positives divided by the total samples identified as positives, including the false negatives. In multi-class problems, the F1-score often is applied to each class and aggregated using micro-average or macro-average to give a final result. In this work, we use the macro-average due to the imbalanced nature of the datasets. Furthermore, the micro-averaging can mask the real performance of minority classes (Charitidis et al., 2020).

We look at divergences between the terms to measure the performance of the bias mitigation module, which we are calling as **Unintended bias evaluation metrics**. However, the disproportional distribution in the original dataset can be followed by the test set and influence the de-bias evaluation (Park et al., 2018; Nozza et al., 2019). Therefore, we generated a synthetic unbiased test set, with similar comments distribution for each term, based on the identity term template method proposed in (Dixon et al., 2018; Park et al., 2018; Nozza et al., 2019).

The idea is to build sentences changing only the identity term, for instance, "*Women* should be protected" and "*Men* should be protected". We define several templates filled with the terms described previously (see Table 2). Thus, each identity term occurs in the same context. Table 6 presents examples of templates. Several studies have also used this approach to evaluate unintended gender bias in different abusive language datasets (Nozza et al., 2019; Park et al., 2018). The synthetic test set[3] created comprises $1,248$ samples, of which 648 are non-hateful, and 600 are hateful, and all identity terms appear in the same contexts.

Table 6: Template examples used to generate the synthetic test set.

Example template	**Label**
I am $< identity >$	non-hateful
$< identity >$ should be protected	non-hateful
You are a good $< identity >$	non-hateful
Filthy $< identity >$	hateful
I hate all $< identity >$	hateful
$< identity >$ should be killed	hateful

For evaluation of the unintended bias, we use metrics introduced in a recent state-of-the-art work (Dixon et al., 2018). The *Error Rate Equality Differences* compute the aggregation of the difference between the false positive rate (FPR) or false negative rate (FNR) on the entire test set and the per-term values, FPR_t and FNR_t. False Positive Equality Difference (FPED) and False Negative Equality Difference (FNED) are defined in Equations 1 and 2, respectively, where $T = \{female, male, girl, boy, ...\}$.

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

$$FNED = \sum_{t \epsilon T} |FNR - FNR_t| \tag{2}$$

The error rate equality differences measure the model's fairness based on the hypothesis that a model without unintended bias has a similar error rate across all identity terms. Therefore, for these metrics, the ideal result is zero. It is relevant to mention that these metrics aim to evaluate bias. Thus, the punctual values of these metrics are not necessary, but rather whether they have similar values across all terms. Hence, we want to evaluate whether a specific term influences the error rates and, consequently, is subjected to unintended bias.

[3] https://github.com/Francimaria/Hate_speech_gender_bias

The *Pinned AUC Equality Difference* (*pAUC*) metric is also investigated in the literature to measure unintended bias (Dixon et al., 2018). However, we decided not to apply the *pAUC* metric because it suffer from several limitations (Borkan et al., 2019). Moreover, its competence to measure unintended bias depends on the sampling procedure used (Badjatiya et al., 2019).

5.4. Statistical analysis

For statistic analysis of the classifiers, we used the non-parametric Friedman test to compare the classification performance of all classifiers over the datasets as recommended in (Demšar, 2006). The Friedman test ranks each algorithm for each dataset. The best performing algorithm gets the rank 1, the second-best rank 2, and so on. In the case of ties, average ranks are used. Then, the average rank is computed using all datasets. We performed the tests with 95% confidence, i.e. the level of significance $\alpha = 0.05$.

We also performed a post-hoc Bonferroni-Dunn test for pairwise comparison between the average ranks for each classifier over the datasets. The critical difference is measured to evaluate whether the performance of the two classifiers is significantly different. The performance of the two classifiers is considered significantly different when the average rank is higher than the critical difference. The critical difference (CD) is defined in Equation 3. The critical value q_α is based on the Studentized range statistic divided by $\sqrt{2}$.

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{3}$$

We used the critical difference diagram proposed in (Demšar, 2006) to describe post-hoc test results projected onto the average rank axis. The thick horizontal line connects classifiers that are not significantly different on the CD diagram.

Furthermore, we also investigated a second pairwise statistical analysis test to examine whether there is a significant difference between the classification methods. We use the Wilcoxon non-parametric signed-rank test with the level of significance $\alpha = 0.05$. Demšar (2006) stated that this method is robust for pairwise comparison between classification algorithms.

6. Results and discussion

In this work, we divided the experimental study into four parts. In Section 6.1, the base classifiers are evaluated for each dataset using the test set. In Section 6.2, evaluate the proposed model performance with the test set and the unintended bias mitigation using the synthetic test set. In Section 6.3, we compare the proposed methodology against the state-of-art in other to evaluate the classification performance using the test set and the bias toward identity terms using the synthetic test set. Then, in Section 6.4, we analyse the case of studies to evaluate the effectiveness of the proposed methodology for gender bias mitigation using the synthetic test set.

6.1. Base classifiers evaluation

Firstly, we analysed the behaviour of each base classifier (LR, DT, and SVM) across different feature extractors. We used the macro F-measure scores to compare the general model performance. Table 7 presents the average and standard deviation results for the datasets evaluated. The best results are highlighted in bold, and the second-best results are underlined for each dataset.

As seen in Table 7a, the LR classifier obtained the highest results with the TF feature extractor in WH, WS, and DV datasets. The pair LR and TF-IDF presented the best scores for the DV dataset. For the HE dataset, the monolithic classifier analysed presented a different behaviour. The LR classifier achieved the highest macro F-score with FastText word embedding and word 2-grams.

The DT classifier achieved slightly lower results than the other classifiers analysed for the four datasets evaluated (see Table 7b). This classifier performed better for the WH dataset with the TF-IDF feature extractor. Different feature extractors presented better results for the WS dataset as TF, TF-IDF, and character 3-grams. For the HE dataset, this classifier obtained the highest macro F-score with Glove word embedding and word 2-grams.

Table 7: Performance of the base classifiers varying the feature spaces. Average and standard deviation results of the macro F-score. The best results are highlighted in bold, and the second-best results are underlined for each dataset. We present the results of the average rank at the column named 'Avg rank' of the tables.

LR					
Feature	**WH**	**WS**	**DV**	**HE**	**Avg. rank**
BERT	0.72 (0.01)	0.61 (0.03)	0.59 (0.02)	0.49	5.00
Glove	0.66 (0.01)	0.48 (0.02)	0.65 (0.02)	0.54	4.75
FastText	0.65 (0.01)	0.44 (0.01)	0.59 (0.02)	**0.56**	5.63
TF	**0.76 (0.01)**	**0.70 (0.02)**	**0.71 (0.02)**	0.46	2.50
TF-IDF	0.73 (0.01)	0.65 (0.05)	**0.71 (0.02)**	0.45	3.63
w2grams	0.57 (0.01)	0.44 (0.03)	0.44 (0.02)	**0.56**	6.25
w3grams	0.38 (0.01)	0.41 (0.02)	0.30 (0.01)	0.42	9.00
c2grams	0.72 (0.01)	0.64 (0.03)	0.64 (0.02)	0.43	5.38
c3grams	0.75 (0.01)	0.70 (0.04)	0.70 (0.02)	0.47	2.88

(a) Results obtained with Logistic Regression classifier (LR).

DT					
Feature	**WH**	**WS**	**DV**	**HE**	**Avg. rank**
BERT	0.52 (0.01)	0.46 (0.02)	0.46 (0.01)	0.52	6.5
Glove	0.55 (0.01)	0.44 (0.02)	0.54 (0.01)	**0.54**	5.5
FastText	0.56 (0.01)	0.44 (0.01)	0.52 (0.01)	0.53	5.875
TF	0.71 (0.01)	**0.69 (0.04)**	**0.70 (0.01)**	0.41	3.5
TF-IDF	**0.72 (0.01)**	**0.69 (0.04)**	0.69 (0.01)	0.43	3.25
w2grams	0.60 (0.01)	0.52 (0.04)	0.49 (0.02)	**0.54**	4.625
w3grams	0.42 (0.01)	0.45 (0.03)	0.33 (0.01)	0.50	7.5
c2grams	0.65 (0.01)	0.66 (0.03)	0.62 (0.01)	0.46	4.5
c3grams	0.70 (0.01)	**0.69 (0.04)**	0.67 (0.01)	0.44	3.75

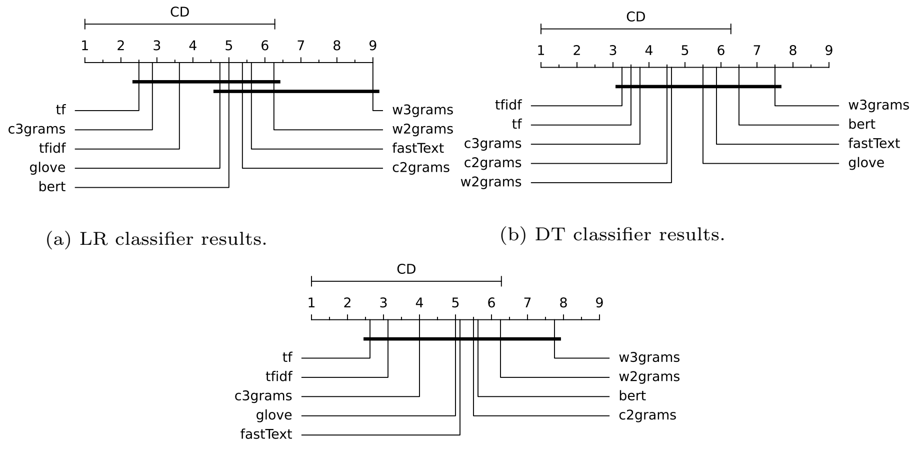
(b) Results obtained with Decision Tree classifier (DT).

SVM					
Feature	**WH**	**WS**	**DV**	**HE**	**Avg. rank**
BERT	0.71 (0.01)	0.63 (0.03)	0.57 (0.02)	0.48	5.625
Glove	0.70 (0.01)	0.48 (0.02)	0.61 (0.02)	0.55	5
FastText	0.71 (0.01)	0.47 (0.01)	0.59 (0.02)	**0.56**	5.125
TF	**0.75 (0.01)**	**0.70 (0.03)**	**0.71 (0.02)**	0.42	2.625
TF-IDF	**0.75 (0.01)**	0.68 (0.03)	0.68 (0.02)	0.44	3.125
w2grams	0.56 (0.01)	0.47 (0.03)	0.45 (0.02)	**0.56**	6.25
w3grams	0.38 (0.01)	0.44 (0.02)	0.31 (0.01)	0.49	7.75
c2grams	0.72 (0.01)	0.65 (0.03)	0.60 (0.02)	0.39	5.5
c3grams	0.74 (0.01)	0.69 (0.02)	0.67 (0.01)	0.40	4

(c) Results obtained with the Support Vector Machine classifier (SVM).

For the SVM classifier (Table 7c), the TF feature extractor obtained the highest classification performance for WH, WS, and DV datasets. On the other hand, the SVM with FastText word embedding and word 2-grams feature extractors presented the best classification performance for the HE dataset, similarly to the LR classifier.

The Friedman statistic test shows that there is a significant difference between the classification performance of each algorithm with the nine feature extraction techniques. Then, we evaluated a pairwise comparison using a post-hoc test. Figure 5 shows the Critical Difference (CD) diagram of the statistical test. The TF, TF-IDF, and character 3-grams feature extraction algorithms presented the highest rank values with the three classifiers. These results demonstrated that the vocabulary used is similar in these datasets, and the BoW and character n-grams approaches are still relevant in this context. Moreover, the experiments showed that the performance of the classifiers is highly dependent on the selected feature space and the dataset under analysis.



(a) LR classifier results.

(b) DT classifier results.

(c) SVM classifier results.

Figure 5: Graphical representation of the average rank for each classifier over all datasets. For each classifier, we evaluated the performance with nine different feature extraction techniques. We used Bonferroni-Dunn post-hoc test to compute the critical difference (CD). Techniques with no statistical difference are connected by horizontal lines.

6.2. Proposed model evaluation

In this section, we will analyse the results obtained with the proposed methodology using three different base classifiers (LR, DT, and SVM) and our focus is to evaluate whether the bias mitigation model compromises the performance of the ensemble learning model. For the stacking generalisation (Wolpert, 1992), we have used the Logistic Regression algorithm as the meta-classifier. We selected this classifier because it is simpler and quicker than SVM and obtained better performance than the DT classifier (over the WH, WS, and DV datasets). Moreover, the Wilcoxon signed-rank test results demonstrate that there is not a significant difference between the performance of the LR and SVM classifiers for three of the datasets analysed (WH, DV, and HE).

Table 8 describes the results obtained with the proposed methodology using the original training set and the bias mitigation module. We presented the average and standard deviation results for WH, WS, and DV datasets. For the HE dataset, we used the partitions proposed in (Basile et al., 2019). The best results are highlighted in bold for each metric. For each dataset, we performed a pairwise comparison of the proposed methodology with the original training set and after using the bias mitigation module. We used the Wilcoxon statistical test to compare the models, and significantly better results are marked with a $*$.

For the WH dataset (Table 8a), our proposed methodology obtained the best macro F-score, FPDE, and FNDE results. These results suggest that the proposed methodology reduces the unintended gender bias without compromise the model performance in this dataset. However, the bias mitigation scores tended to have a slightly increased with the SVM classifier.

Table 8b presents the results of the WS dataset. The proposed classifier obtained similar results with the original training set and with the bias mitigation module. On the other hand, for the DV dataset (Table 8c), the proposed classifier achieved macro F-score slightly inferior with the bias mitigation module. The HE dataset (see Table 8d)) also was evaluated in task 5-A at SemEval-2019, the mean of the baseline results with the dataset in English were 0.451

Table 8: Results obtained by the proposed method. Before and after applying the bias mitigation module. The best results are highlighted in bold for each metric. Results that are significantly better are marked with ∗.

	Original training set			Bias Mitigation module		
Model	**F-score**	**FPED**	**FNED**	**F-score**	**FPED**	**FNED**
proposed (LR)	0.77 (0.009)	1.07 (0.486)	1.49 (0.499)	∗ **0.79 (0.011)**	∗**0.20 (0.237)**	∗**0.47 (0.318)**
proposed (DT)	0.74 (0.010)	0.91 (0.628)	1.00 (0.377)	∗ **0.77 (0.011)**	∗ **0.29 (0.343)**	∗**0.37 (0.329)**
proposed (SVM)	0.77 (0.008)	∗**0.84 (0.313)**	∗**1.43 (0.445)**	∗ **0.79 (0.012)**	2.94 (1.183)	3.03 (1.091)

(a) Results obtained by the proposed method with the WH dataset.

	Original training set			Bias Mitigation module		
Model	**F-score**	**FPED**	**FNED**	**F-score**	**FPED**	**FNED**
proposed (LR)	**0.71 (0.026)**	**0.07 (0.129)**	∗**0.17 (0.339)**	0.71 (0.034)	0.08 (0.159)	0.47 (0.592)
proposed (DT)	0.68 (0.043)	0.01 (0.031)	**0.00 (0.017)**	**0.69 (0.039)**	**0.00 (0.000)**	**0.00 (0.000)**
proposed (SVM)	0.70 (0.024)	∗**0.10 (0.193)**	∗**0.53 (0.418)**	0.70 (0.034)	0.24 (0.391)	0.90 (1.112)

(b) Results obtained by the proposed method with the WS dataset.

	Original training set			Bias Mitigation module		
Model	**F-score**	**FPED**	**FNED**	**F-score**	**FPED**	**FNED**
proposed (LR)	∗**0.72 (0.022)**	5.72 (1.584)	3.99 (1.320)	0.71 (0.023)	∗**4.39 (1.495)**	**3.60 (0.730)**
proposed (DT)	**0.67 (0.017)**	2.37 (1.561)	2.40 (1.572)	0.66 (0.014)	∗ **0.80 (0.868)**	∗**0.84 (0.597)**
proposed (SVM)	∗**0.71 (0.025)**	∗**5.62 (0.873)**	∗**3.82 (0.604)**	0.70 (0.024)	6.03 (0.894)	4.29 (0.612)

(c) Results obtained by the proposed method with the DV dataset.

	Original training set			Bias Mitigation module		
Model	**F-score**	**FPED**	**FNED**	**F-score**	**FPED**	**FNED**
proposed (LR)	**0.46**	0.27	3.82	0.45	**0.00**	**1.54**
proposed (DT)	**0.44**	4.02	4.69	0.42	**0.00**	**0.15**
proposed (SVM)	0.42	0.14	3.00	0.42	**0.00**	**1.92**

(d) Results obtained by the proposed method with the HE dataset.

and 0.367, with the SVM and MFC (this assigns the most frequent labels), respectively, and the proposed model obtained similar results with LR classifier. Moreover, for this dataset, the proposed methodology reduces the unintended gender bias.

In order to improve the general classification performance of the proposed method for the HE dataset, we also evaluated the proposed model with different feature extractor combinations. After conducting empirical tests, we found a better trade-off between macro F-score and gender bias mitigation using three feature extraction methods: the word embedding FastText, Glove, and word 2-grams. The performance of the monolithic models with the other feature extractors can have influenced the results obtained. The results are described in Table 9. Although the better classification performance, the proposed method using all features obtained better bias mitigation with the LR and SVM classifiers than using a subset of the features.

Table 9: Results obtained by the proposed method adapted for HE dataset. Before and after applying the bias mitigation module. *In the pool of classifiers, we used only three feature extractors (FastText, Glove, and word 2-grams). The best results are highlighted in bold for each metric.

	Original training set			**Bias Mitigation module**		
Model*	**F-score**	**FPED**	**FNED**	**F-score**	**FPED**	**FNED**
proposed (LR)	0.55	4.25	7.43	0.55	**2.85**	**5.76**
proposed (DT)	0.54	0.41	0.28	**0.55**	**0.00**	**0.00**
proposed (SVM)	0.55	0.76	3.87	0.55	**0.45**	**3.31**

It has been shown in the literature the use of monolithic classifiers for hate speech classification task, such as (Waseem & Hovy, 2016) and (Davidson et al., 2017) used LR; (Salminen et al., 2018) used LR, DT, SVM and also used ensemble models; and (Senarath & Purohit, 2020) used SVM. We have evaluated these classifiers with different feature extraction methods (see Section 6.1) and the proposed method achieved better performance than these methods for WH, WS, and DV datasets (Table 8). Even though we employed a simple strategy for bias mitigation and classical machine learning classifiers, the proposed methodology proved robust to unintended gender bias mitigation without compromising the model performance.

For the HE dataset, the method proposed using only three feature extractors (see Table 9) would be placed at the third position out of 69 submissions to the English Subtask A[4]. It is relevant to highlight that even though the team in the second position obtained a 0.571 macro f-score, they did not provide the system descriptions for a fair comparison. Moreover, our method also deals with unintended bias mitigation, which is in addition to classification performance.

Despite the ensemble learning techniques being time-consuming compared to monolithic models, the base models of the proposed stacked classifier can be executed simultaneously, reducing the processing time of the proposed model. Furthermore, once trained, its prediction is faster.

6.3. Proposed methodology versus the best base classifier

This section compares the results obtained by the proposed methodology against the best results obtained by the LR base classifier (Table 10). We evaluated the classification performance of the models using the macro F-score, while the FPED and FNED metrics are employed for bias evaluation.

Table 10: Performance of the proposed method and the LR classifier. *In the pool of classifiers, we used only three feature extractors (FastText, Glove, and word 2-grams) for the HE dataset marked with ∗.

Dataset	**Model**	**F-score**	**FPED**	**FNED**
WH	TF + LR	0.76 (0.01)	0.61 (0.68)	1.41 (0.64)
	proposed (LR)	**0.79 (0.01)**	**0.20 (0.24)**	**0.47 (0.32)**
WS	TF + LR	0.70 (0.02)	**0.00 (0.00)**	**0.00 (0.00)**
	proposed (LR)	**0.71 (0.03)**	0.08 (0.16)	0.47 (0.59)
DV	TF + LR	0.71 (0.02)	**2.36 (1.83)**	**2.43 (1.48)**
	TF-IDF + LR	0.71 (0.02)	2.91 (1.47)	2.50 (0.77)
	proposed (LR)	0.71 (0.02)	4.39 (1.50)	3.60 (0.73)
HE	FastText + LR	**0.56**	6.24	6.62
	w2grams + LR	**0.56**	0.19	**0.22**
	proposed (LR)	0.45	**0.00**	1.54
	proposed (LR)∗	0.55	2.85	5.76

The proposed method obtained better classification performance and bias mitigation results for the WH dataset than with the best monolithic classifier evaluated. Even though the proposed method obtained higher macro F-score re-

[4] https://docs.google.com/spreadsheets/d/1wSFKh1hvwwQIoY8_XBVkhjxacDmwXFpkshYzLx4bw-0/edit#gid=0

sults than the LR classifier, it presented a slightly higher gender bias for the WS dataset. For the DV dataset, the proposed method presented the same macro F-score result as the LR classifier. Although the proposed method presented a slightly inferior macro F-score for the HE dataset, it achieved better bias mitigation results. Furthermore, even though the pre-trained word embedding FastText had better classification performance for the HE dataset (see Table 10), the FPED and FNED metrics obtained higher values. This behaviour of word embedding confirms the results in (Bolukbasi et al., 2016), stating that even word embeddings trained with millions of data can present bias.

The WH and HE datasets presented the highest frequency of specific identity terms in the “sexism” and “hateful” labelled classes (see Section 4.1.1), respectively. We can infer that the disproportionate representation of identity terms in the training set influenced the performance of the monolithic models for particular identity terms in these datasets.

For statistical analysis of the proposed methodology performance and the monolithic classifiers, we used the Friedman statistic test that shows that there is a significant difference in the performance of the classifiers. Then, we performed a pairwise comparison using a post-hoc test. Figure 6 presents the CD diagram of the statistical test. The pairwise comparison between the models presented that the proposed model presents a better average rank than different monolithic classifiers. However, its performance there is not a significant difference of some classifiers. Even though the proposed methodology only presented a minor classification improvement in contrast with some classifiers, the main objective of the proposed model is to reduce the unintended gender bias without compromising the classification model performance.

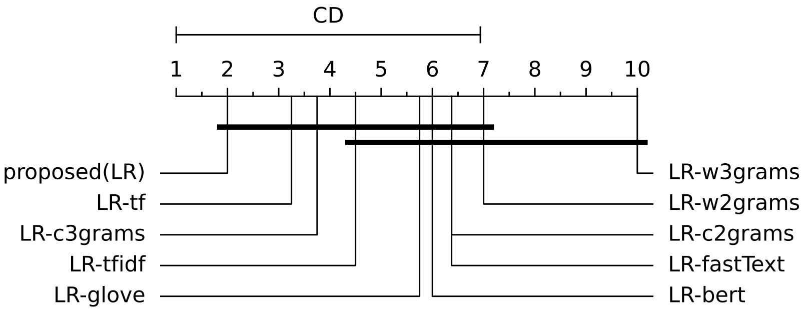


Figure 6: Graphical representation of the average rank for each model using the LR classifier over all datasets. For the HE dataset was used the proposed methodology results with only three features. The Bonferroni-Dunn post-hoc test was used to compute the critical difference (CD). Horizontal lines connect techniques with no statistical difference. The best classifier is the one presenting the lowest average rank.

The unintended gender bias also has been investigated in the literature. Park et al. (2018) analysed three different strategies for gender bias mitigation (Debiased Word Embeddings, Gender Swap, and Bias fine-tuning) for the WS dataset. The methods analysed presented values between 0.006 and 0.333 for the FNED metric, and between 0.027 and 0.337 for the FPED, with different models and bias mitigation method combinations. However, the methods used have affected the classification performance evaluated with the AUC metric. Although our proposed model has presented FNED higher than the presented in (Park et al., 2018), it has reduced the unintended gender bias without compromising the classification model performance.

6.4. Case Studies

This section evaluates the effectiveness of the proposed methodology using different pairs of examples from the proposed synthetic dataset. Table 11 presents the hateful score predicted by the classifiers for each pair of samples using the LR classifier trained on the WH dataset. This dataset was selected because it presented a higher disproportionate representation of identity terms (see Section 4.1.1). The examples presented are clearly non-hateful tweets. For instance, the first sample "*You are a great woman*", the Logistic Regression classifier with Term Frequency feature extractor (LR + TF) predicted the hateful label (score) of 0.33 while the proposed model after the bias mitigation gave the probability score of 0.18. We can infer that the significant frequency of particular identity terms in hateful comments and the imbalance nature of the training data used for hate speech detection can contribute to the increase of *false positive bias*, in which the model can give unreasonable high hateful score to the clearly non-hateful sentence due to the use of particular identity terms, similar to the reports in (Dixon et al., 2018).

We also showed a boxplot (Figure 7) of these examples' hateful score to better visualisation because we collected the score across the k-fold cross-validation. Thus, we used the scores from the 20 executions. Each example is identified by the BSW used. The obtained results show the effectiveness of the proposed methodology. Even though using a simple method for bias mitigation, it performed well. Moreover, the proposed classifier demonstrated be less sensitivity to unintended gender bias than monolithic models.

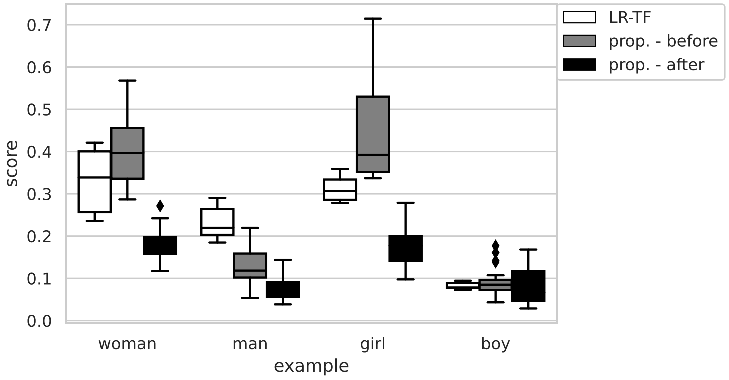


Figure 7: Case studies sentences predictions across the k-fold cross-validation. Logistic Regression classifier with Term Frequency feature extractor (LR-TF), proposed model before bias removal with original data (prop. - before), and proposed model after bias removal (prop. - after).

Table 11: Sentence predictions obtained by a monolithic classifier and the proposed method before and after the bias mitigation stage. The bias sensitive words are highlighted in bold. All examples are non-hateful. Bias Sensitive Words (BSWs).

		LR + TF	**Before bias removal**	**After bias removal**
BSW	**Examples**	***sexism***	***sexism***	***sexism***
woman	You are a great **woman**	0.33(0.067)	0.40(0.079)	0.18(0.038)
man	You are a great **man**	0.23(0.037)	0.13(0.044)	0.08(0.032)
girl	I am **girl**	0.31(0.027)	0.44(0.115)	0.17(0.044)
boy	I am **boy**	0.08(0.007)	0.09(0.036)	0.09(0.045)

7. Conclusions and future work

In this paper, we have discussed how to identify and analyse bias mitigation, particularly toward gender identity terms, in the hate speech detection task, namely **unintended gender bias**. We have proposed a methodology based on two different modules to address the problem. In the first module, we proposed a gender bias mitigation strategy. Then, in the second module, a multi-view stacked classifier for hate speech detection. We selected nine different feature extraction methods, and we evaluated the proposed methodology with three base classifiers (LR, DT, and SVM).

Overall, the proposed classifier outperforms different models using several feature extractors using the WH, WS, and DV datasets. Furthermore, the proposed methodology reduced the unintended gender bias without compromising the performance in the WH dataset. The dataset presented the highest disproportionate in the identity terms representation. Although some results are slightly inferior, the proposed methodology demonstrates to be effective compared to state-of-the-art solutions.

It is relevant to highlight that the proposed multi-view stacked classifier is general enough to work with different feature extractors and classification models. Therefore, the proposed classifier can be extended and continuously improve the classification results.

As future work, we intend to explore complementary feature extraction techniques that better fitting for each dataset and newer ensemble learning strategies as dynamic selection methods (Cruz et al., 2018). Furthermore, we also purpose to investigate other strategies to select potential bias-sensitive words related to gender stereotypes. Although the proposed methodology focuses on gender terms, the method proposed can be expanded to work with other identity problems as racial stereotypes.

Acknowledgements

This work was supported by the Brazilian CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, in portuguese) (Scholarship no. 88887.484211/2020-00).

References

Agarwal, S., & Chowdary, C. R. (2021). Combating hate speech using an adaptive ensemble learning model with a case study on covid-19. *Expert Systems with Applications*, *185*, 115632. doi:https://doi.org/10.1016/j.eswa.2021.115632.

Al-Azani, S., & El-Alfy, E.-S. M. (2017). Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text. *Procedia Computer Science*, *109*, 359–366.

Al-Makhadmeh, Z., & Tolba, A. (2019). Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. *Computing*, *102*, 501–522.

Almatarneh, S., Gamallo, P., Pena, F. J. R., & 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.

Antonakaki, D., Fragopoulou, P., & Ioannidis, S. (2021). A survey of twitter research: Data model, graph structure, sentiment analysis and attacks. *Expert Systems with Applications*, *164*, 114006.

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). New York, NY, USA: ACM.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F. M. R., 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* (pp. 54–63). Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *30th Conference on Neural Information Processing Systems* (pp. 4349–4357). Barcelona, Spain: Advances in Neural Information Processing Systems.

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.

Burnap, P., & Williams, M. L. (2016). Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data science*, *5*, 11.

Cao, R., Lee, R. K.-W., & Hoang, T.-A. (2020). Deephate: Hate speech detection via multi-faceted text representations. In *12th ACM Conference on Web Science* WebSci '20 (p. 11–20). New York, NY, USA: ACM.

Charitidis, P., Doropoulos, S., Vologiannidis, S., Papastergiou, I., & Karakeva, S. (2020). Towards countering hate speech against journalists on social media. *Online Social Networks and Media*, *17*, 100071.

Chiril, P., Moriceau, V., Benamara, F., Mari, A., Origgi, G., & Coulomb-Gully, M. (2020). He said "who's gonna take care of your children when you are at acl?": Reported sexist acts are not sexist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4055–4066). online: ACL.

Corazza, M., Menini, S., Cabrio, E., Tonelli, S., & Villata, S. (2020). A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology*, *20*, 1–22.

Cruz, R. M., Cavalcanti, G. D., Tsang, I. R., & Sabourin, R. (2013). Feature representation selection based on classifier projection space and oracle analysis. *Expert Systems with Applications*, *40*, 3813–3827.

Cruz, R. M., Hafemann, L. G., Sabourin, R., & Cavalcanti, G. D. (2020). Deslib: A dynamic ensemble selection library in python. *Journal of Machine Learning Research*, *21*, 1–5.

Cruz, R. M., Sabourin, R., & Cavalcanti, G. D. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, *41*, 195–216.

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.

Del Vigna, F., Cimino, A., Dell'Orletta, F., Petrocchi, M., & Tesconi, M. (2017). Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity* (pp. 86–95). CEUR–WS.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, *7*, 1–30.

DeSouza, G., & 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). Glasgow, UK: IEEE.

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). Minneapolis, Minnesota: ACL.

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 (p. 67–73). New York, NY, USA: ACM. URL: `https://doi.org/10.1145/3278721.3278729`.

Dorris, W., Hu, R. R., Vishwamitra, N., Luo, F., & Costello, M. (2020). Towards automatic detection and explanation of hate speech and offensive language. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics* (p. 23–29). New York, NY, USA: ACM.

Elisabeth, D., Budi, I., & Ibrohim, M. O. (2020). Hate code detection in indonesian tweets using machine learning approach: A dataset and preliminary study. In *2020 8th International Conference on Information and Communication Technology* (pp. 1–6). Yogyakarta, Indonesia: IEEE.

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

Founta, A. M., 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). New York, NY, USA: ACM.

Gitari, N. D., Zuping, Z., Damien, H., & Long, J. (2015). A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, *10*, 215–230.

Kapil, P., & Ekbal, A. (2020). A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, *210*, 106458.

Kiritchenko, S., & Mohammad, S. M. (2018). Examining gender and race bias in two hundred sentiment analysis systems. *NAACL HLT 2018*, (p. 43).

Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. (2nd ed.). John Wiley & Sons.

Liu, H., Burnap, P., Alorainy, W., & Williams, M. L. (2019). Fuzzy multi-task learning for hate speech type identification. In *The World Wide Web Conference* (pp. 3006–3012). New York, NY, USA: ACM.

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PLOS ONE*, *14*, 1–16.

Mathew, B., Dutt, R., Goyal, P., & Mukherjee, A. (2019). Spread of hate speech in online social media. In *Proceedings of the 10th ACM conference on web science* (pp. 173–182). New York, NY, USA: ACM.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR*.

Miok, K., Nguyen-Doan, D., Škrlj, B., Zaharie, D., & Robnik-Šikonja, M. (2019). Prediction uncertainty estimation for hate speech classification. In *International Conference on Statistical Language and Speech Processing* (pp. 286–298). Cham: Springer.

Miškolci, J., Kováčová, L., & Rigová, E. (2020). Countering hate speech on facebook: The case of the roma minority in slovakia. *Social Science Computer Review*, *38*, 128–146.

Montani, J. P., & Schüller, P. (2018). Tuwienkbs at germeval 2018: German abusive tweet detection. In *14th Conference on Natural Language Processing KONVENS* (p. 45). Vienna, Austria: Austrian Academy of Sciences.

Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). Hate speech detection and racial bias mitigation in social media based on bert model. *PLOS ONE*, *15*, 1–26.

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). Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.

Nozza, D., Volpetti, C., & Fersini, E. (2019). Unintended bias in misogyny detection. In *IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 149–155). New York, NY, USA: ACM.

Oriola, O. (2020). A stacked generalization ensemble approach for improved intrusion detection. *International Journal of Computer Science and Information Security*, *18*, 62–67.

Park, J. H., 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). Brussels, Belgium: ACL.

Paschalides, D., Stephanidis, D., Andreou, A., Orphanou, K., Pallis, G., Dikaiakos, M. D., & Markatos, E. (2020). Mandola: A big-data processing and visualization platform for monitoring and detecting online hate speech. *ACM Transactions on Internet Technology*, *20*, 1–21.

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. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, *48*, 4730–4742.

Plaza-Del-Arco, F.-M., Molina-González, M. D., Ureña-López, L. A., & Martín-Valdivia, M. T. (2020). Detecting misogyny and xenophobia in spanish tweets using language technologies. *ACM Transactions on Internet Technology (TOIT)*, *20*, 1–19.

Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., & Patti, V. (2020). Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, (pp. 1–47).

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). Marseille, France: ELRA.

Rizos, G., Hemker, K., & Schuller, B. (2019). Augment to prevent: short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 991–1000). New York, NY, USA: ACM.

Sajjad, M., Zulifqar, F., Khan, M. U. G., & Azeem, M. (2019). Hate speech detection using fusion approach. In *2019 International Conference on Applied and Engineering Mathematics (ICAEM)* (pp. 251–255). IEEE.

Salminen, J., Almerekhi, H., Milenkovic, M., Jung, S.-g., An, J., Kwak, H., & Jansen, B. J. (2018). Anatomy of online hate: Developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In *ICWSM* (pp. 330–339).

Salminen, J., Hopf, M., Chowdhury, S. A., Jung, S.-g., Almerekhi, H., & Jansen, B. J. (2020). Developing an online hate classifier for multiple social media platforms. *Human-centric Computing and Information Sciences*, *10*, 1.

Santosh, T., & Aravind, K. (2019). Hate speech detection in hindi-english code-mixed social media text. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data* (pp. 310–313). New York, NY, USA: ACM.

Sap, M., Card, D., Gabriel, S., Choi, Y., & Smith, N. A. (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).

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.

Unsvåg, E. F., & Gambäck, B. (2018). The effects of user features on twitter hate speech detection. In *Proceedings of the 2nd workshop on abusive language online (ALW2)* (pp. 75–85).

Walmsley, F. N., Cavalcanti, G. D., Oliveira, D. V., Cruz, R. M., & Sabourin, R. (2018). An ensemble generation method based on instance hardness. In *2018 International Joint Conference on Neural Networks* (pp. 1–8). IEEE.

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). Austin, Texas: ACL.

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). San Diego, California: ACL.

Watanabe, H., Bouazizi, M., & Ohtsuki, T. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, *6*, 13825–13835.

Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019). Detection of abusive language: the problem of biased datasets. In *Proceedings of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: human language technologies, volume 1 (long and short papers)* (pp. 602–608). Minneapolis, Minnesota: ACL.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. M. (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). Online: ACL.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*, 241–259.

Zhang, Z., & Luo, L. (2019). Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, *10*, 925–945.

Zhang, Z., Robinson, D., & Tepper, J. (2018). Hate speech detection using a convolution-lstm based deep neural network. *ESWC 2018: The semantic web*, .

Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, *38*, 43–54.

Zhou, Y., Yang, Y., Liu, H., Liu, X., & Savage, N. (2020). Deep learning based fusion approach for hate speech detection. *IEEE Access*, *8*, 128923–128929.

Zimmerman, S., Kruschwitz, U., & Fox, C. (2018). Improving hate speech detection with deep learning ensembles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*. ELRA.