Journal of Computer Science

Research Article

Optimizing Bias Detection in Tweets Using Bayesian Probabilistic Model

Prasanth G Rao, Harsha Chigurupati, Krish Hashia, Thriveni J, P Deepa Shenoy and Venugopal K R

Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore, 560001, India

Article history
Received: 25-01-2026
Revised: 20-03-2026
Accepted: 07-04-2026

Corresponding Author:
Prasanth G Rao
Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore, 560001, India
Email: prasanthgrao@gmail.com

Abstract: Content moderation on social media faces persistent challenges from inconsistent evaluation shaped by subjective judgment and subtle semantic variations. This work proposes a Bayesian probabilistic framework for detecting bias in tweets using WordNet-based vocabulary filtering, statistical normalization via z-scores, and threshold optimization. The system is stateless, scalable, and dataset-agnostic, requiring no session-specific information. Unlike complex models such as Support Vector Machines (SVM), Multi-Layer Perceptrons (MLP), and AdaBoost, which tend to exhibit skewed classification patterns, the proposed approach achieves balanced confusion matrices and competitive F1 scores. Experimental evaluation across three benchmark datasets covering hate speech, political partisanship, and racial and gender-based discrimination demonstrates accuracy ranging from 71 to 82.4%, with the highest F1 score of 0.859 on Dataset 1, confirming the framework's effectiveness for interpretable and balanced bias detection.

Keywords: X, Bias, Fairness, SentiWordNet

Introduction

Social media platforms such as X (formerly Twitter), Facebook, Instagram, and Reddit have become primary channels for information exchange and opinion formation. Users increasingly rely on these platforms for news consumption (Statista, 2024b), with millennials preferring social media over traditional media (Statista, 2024a). This shift amplifies the impact of biased content; as influential users can shape real-world outcomes through their reach. Consequently, automated, scalable systems for bias detection are essential for maintaining the integrity of public discourse.

Bias, defined as a predisposition that obstructs objective judgment (Mehrabi et al., 2021), manifests in social media through multiple forms: gender prejudice, racial bias, confirmation bias, conformity bias, and sampling bias, among others. These biases affect data collection, analysis, and downstream decision-making, with adverse real-world consequences (Morstatter and Liu, 2017). From an algorithmic perspective, models trained on biased data risk perpetuating and amplifying these biases, as demonstrated by challenges in toxicity detection (Sap et al., 2019) and content moderation systems.

Sentiment analysis, particularly through lexical resources such as SentiWordNet 3.0 (Baccianella et al., 2010), has been widely adopted for bias-related tasks. SentiWordNet assigns positivity, negativity, and objectivity scores at the synset level, enabling fine-grained sentiment classification. However, sentiment and bias are not equivalent: as Sap et al. (2019) demonstrated, sentiment-based approaches can exhibit systematic racial bias, particularly against African American English. Several recent approaches have been proposed, including the BABE framework (Spinde et al., 2021) achieving 0.804 F1-Score using Bidirectional Encoder Representations from Transformers (BERT) models, and various deep learning architectures. Despite these advances, significant challenges persist in generalizability across diverse datasets, maintaining model accuracy while mitigating bias, and capturing nuanced or contextual forms of bias beyond binary classification.

In our previous work (Rao et al., 2026), we evaluated multiple machine learning algorithms for bias detection across curated datasets and found that Bayesian approaches, particularly Gaussian Naive Bayes (GNB), demonstrated consistent performance. Building on this finding, the present work proposes an optimized Bayesian

probabilistic framework that combines WordNet-based vocabulary filtering, z-score normalization of bias ratios, and threshold optimization. The main contributions of this work are as follows:

1. A Bayesian scoring mechanism that integrates word-level z-scores with prior bias probabilities to quantify bias association for each word, grounded in the principle that a word's bias contribution is proportional to its statistical deviation weighted by the inverse of its occurrence probability
2. A threshold optimization strategy that adapts classification boundaries to dataset characteristics, achieving balanced confusion matrices without the overfitting tendencies observed in complex models.
3. A comprehensive comparative evaluation against five established classifiers (GNB, SVM, MLP, AdaBoost, Decision Tree) across three diverse benchmark datasets spanning political, racial, and gender-based bias

Related Work

Wewelwala and Sumanathilaka (2024) surveyed various hybrid methods for emotion recognition utilizing both audio and textual data, highlighting the challenges and limitations of neural networks and deep learning in handling diverse and biased datasets from social media. Although hybrid models improve performance, they do not completely eliminate biases, especially those stemming from diverse social media sources.

Dai et al. (2023) investigated inadvertent biases in toxicity detection systems, specifically those employing LSTM and attention processes. The authors offered an enhanced model that integrates attention mechanisms to address and reduce bias more effectively. The proposed model mitigates certain biases; however, it does not completely eliminate them, underscoring the intricacy of the situation. The method remains inadequate for addressing specific biases, especially those arising from the subjective characteristics of social media content.

Researchers of Shu et al. (2017) first delineated the concept of fake news stories and then characterized the issue as a classification task. Their proposed solution consisted of two phases: feature extraction and model development. Two distinct models are presented: The Social Context Model and the News Media Content Model. The initial technique emphasizes the linguistic subtleties of the article composition, whereas the subsequent approach scrutinizes the factual veracity of the content offered. The researchers are also developing a dataset called FakeNewsNet, which is aimed at improving false news detection. They investigate many strategies, encompassing crowdsourcing, machine learning models such as Convolutional Neural Networks (CNN), and context-building algorithms that leverage resources like DBpedia and the Google Relation Extraction corpus. The report also delineates pertinent research domains and examines prospective trajectories in the discipline.

Kulshrestha et al. (2017) sought to detect bias by examining users' historical interests, follower lists, and tweet keywords. The interest vectors were ranked from highest to lowest for each user. The vectors were transformed into TF-IDF and analyzed against the Democrat and Republican X datasets utilizing cosine similarity. This study assessed input text bias, indicative of X query bias, ranking algorithm bias, and output bias. The ranking bias was determined by subtracting the input bias from output bias, assuming that the ranking mechanism functions as a black box (search results). The authors acknowledge that tweet content may induce bias, although they did not propose a remedy. They asserted that the brevity of tweets may complicate the identification of bias, potentially resulting in suboptimal conclusions. This study presupposes that users possess strong political affiliations (Democrat, neutral, or Republican) while disregarding moderate perspectives. The involvement of hand-polled individuals increased the probability of bias during the verification process.

Alsaad et al. (2018) utilized a Structural Equation Modeling (SEM) methodology to develop indices for racism and religious bias. Their findings indicated that racism is not influenced by social media usage, whereas a negative correlation exists between social media use and religious bias. The authors promoted the incorporation of contrasting perspectives to enhance the comprehension of diversity, potentially resulting in diminished discrimination. They employed data from the World Values Survey and Global Information Technology Report for their analysis.

Hamborg et al. (2019) conducted an interdisciplinary literature review to explore the automated detection of media bias in news articles. They integrated findings from social science with developments in computer science.

This study highlights that many methodologies from social science remain underutilized in the computational field. The authors classified several types of bias observed in social media and examined their effects on businesses and individuals. They also evaluated computational methods, including sentiment analysis and CNNs, and emphasized their advantages and drawbacks.

Ganguly et al. (2020) empirically assessed three assumptions prevalent in the development of political media bias datasets. Initially, they illustrated that in specific instances, the political orientation of the evaluator can affect their assessment of the political bias of a news piece. The analysis revealed that the political orientation of news stories does not consistently correspond to the political orientation of the publisher. Finally, they demonstrated that the publisher's political position may vary when addressing

diverse subjects. The authors utilized highly rated MTurk workers in the study, emphasizing the biases inherent in the dataset development process.

Hargittai (2020) investigated potential biases in big data by analyzing the representation of several demographic categories, including age, gender, race, and income, using data obtained from social media platforms such as X and Reddit. In contrast to earlier research that concentrated on platforms, the author examined the representation of each group inside the data sample. A logistic regression model was employed to plot these characteristics against group representation and to compare them with diverse survey findings in order to evaluate temporal changes in representation. This paper offers an overview of the many types of data biases that may occur when utilizing large amounts of data from social media networks.

Papakyriakopoulos et al. (2020) showed that word embeddings possess intrinsic biases and suggested techniques for identifying and alleviating these biases. The authors emphasized on the difficulties in mitigating gender bias using Support Vector Machines (SVM) in languages such as German. This study demonstrates how word embeddings mirror cultural biases, exemplified by the linkage of professions such as nursing and secretarial work with women. Furthermore, the researchers proposed that these intrinsic biases may be utilized to identify additional types of bias.

Iacus et al. (2020) addressed the problem of selection bias in data obtained from Social Networking Sites (SNS). The authors emphasize that this prejudice occurs because not all people utilize the Internet, and among those who do, not everyone participates in these platforms. To address this bias, they suggested a mathematical approach that utilizes national statistics to modify datasets, guaranteeing that they more accurately represent the population's Internet usage and social networking service involvement.

Ghosh et al. (2021) underscored the necessity for toxicity detection models to uncover and mitigate intrinsic biases arising from the nuanced, yet non-toxic, biases inherent in the discourse of the data's temporal context. The study comprises two sections. The first section concentrates on identifying overrepresented words from seven English-speaking nations with unique cultural backgrounds. K-means clustering was subsequently utilized to categorize these phrases through a substitution and evaluation approach, with the Perspective API implemented for toxicity modeling. The ultimate toxicity scores were juxtaposed with those of a control group to authenticate the model using real-world events. Their approach of substituting prevalent words with alternative meanings uncovers both nation-specific toxic languages and newly developed expressions that could otherwise go unnoticed.

Mozafari et al. (2020) employed a publicly accessible pre-trained BERT model for the identification of hate speech and mitigation of racial bias in social media. Text preprocessing was conducted with WORDPIECE tokenization, with a learning rate of 2e −5 implemented via the Adam optimizer, and a dropout probability of 0.1 established for all layers. The authors claimed to be the first to employ classifiers to address social media bias by experimenting with BERT models. Their model was engineered to facilitate the incorporation and training of new values to alleviate bias. Moreover, they utilized the Log Likelihood Ratio (LLR) to examine commonly employed n-grams and discerned racial bias, observing that tweets in African American English (AAE) demonstrated elevated levels of racism and sexism relative to Standard American English (SAE). Nonetheless, this adversely affected the predictive accuracy of their models. The authors do not address how their methodology would apply to other racial groups, genders, or English dialects except for AAE and SAE.

In their theoretical study, Datta et al. (2021) explored the creation of AI models that avoid amplifying the internal biases that individuals rely on when processing the overwhelming amount of information from social media. The authors argue that by proposing information that matches biases, present AI systems risk reinforcing them and contributing to echo chambers. To counteract this, the study advocates for development of AI models that can mitigate bias by removing anonymity and identifying anchoring biases, thereby preventing the spread of misinformation and fake news. Guo et al. (2022) investigated the relationship between COVID-19 news, such as the number of cases and influx of immigrants, and the introduction of bias into social media by analyzing Chinese tweet datasets. The authors employed a deep learning technique called Attention-based Channels-LSTM Multitask-learning Model (ACLMM). In this method, input data is first converted into word vector representations using the Continuous Bag of Words (CBOW) model, which is then passed through bidirectional LSTM layers, complete with hidden and output layers. A final attention layer was added to specialize in different emotions in parallel. In most cases, the model achieved an accuracy of approximately 80%.

Kaiser et al. (2022) explored the disproportionate effects of sharing information between politically dissimilar individuals on social media platforms. The study highlights that, in such cases, users often unfollow each other in an attempt to stop the spread of misinformation. However, the authors argue that this approach is misguided, as it contributes to the creation of echo chambers where differing viewpoints are less likely to enter the discourse, further exacerbating polarization.

Hollingshead et al. (2022) reviewed the challenges in social media research, highlighting demographic

discrepancies across social platforms such as X, Instagram, and Reddit. The results from social media analysis exhibit inconsistencies and are shown to inadequately reflect the entire community, primarily due to biases linked to platform demographics and computational limitations. Users of certain platforms such as X and LinkedIn tend to be more affluent, leading to skewed representations. The study also emphasizes the selection bias that arises from the use of manual surveys with a limited participant pool.

Hollingshead et al. (2022) reviewed the challenges in social media research, highlighting demographic discrepancies across social platforms such as X, Instagram, and Reddit. The results from social media analysis exhibit inconsistencies and are shown to inadequately reflect the entire community, primarily due to biases linked to platform demographics and computational limitations. Users of certain platforms such as X and LinkedIn tend to be more affluent, leading to skewed representations. The study also emphasizes the selection bias that arises from the use of manual surveys with a limited participant pool.

In their systematic study of media bias detection, Rodrigo-Gines et al. (2024) defined bias as an unjust preference based on subjective judgment. They divided bias into two categories: spin and ideology. They investigated both human-centric methods such as crowdsourcing and automated approaches such as web crawling and machine learning. The authors provide a comprehensive guide to the utilization of NLP techniques, word embeddings, lexicon-based approaches, and models such as SVM and deep learning for bias detection. The importance of diversifying datasets is underscored, and additional research in this field is encouraged.

Singh et al. (2023) introduced Gender Bias Identification and Extraction(GLIdE) for the analysis of gender-biased languages in social media. The model uses TransE to produce entity representations and is further analyzed using BERT to capture lexical associations. The vectors were input into a Graph Convolutional Network (GCN) and attention mechanisms to focus on emotions. A pretrained BERT model facilitates semantic encoding, whereas a T2G graph assists in analyzing contextual word usage. The model was evaluated in comparison with established models such as BERT and Bi-RNNs.

Recent literature has further highlighted emerging challenges in bias identification and mitigation within modern AI systems. Wei et al. (2025) examined bias in generative AI from an information management perspective, emphasizing how data curation practices, model feedback loops, and governance limitations contribute to persistent and amplified biases. In high-stakes domains, (Koçak et al., 2024) presented a comprehensive analysis of bias in medical imaging AI, detailing sources, detection strategies, mitigation techniques, and ethical concerns, while underscoring the risks of biased decision making in clinical contexts. Complementing these studies, Lin et al. (2025) investigated LLM-based bias detection and revealed notable disparities between model judgments and human perception, raising concerns about the reliability and generalizability of automated bias assessment systems.

Despite major developments in bias detection in social media, some unresolved difficulties remain. Contemporary models frequently encounter difficulties with generalizability, particularly when utilized with heterogeneous datasets and in multiple situations. Although initiatives to alleviate biases, including gender and racial biases, have demonstrated potential, these strategies frequently result in diminished model accuracy. Many models tend to oversimplify bias into binary categories, which presents another challenge in capturing nuanced or moderate viewpoints. Furthermore, the challenge of overcoming the biases inherent in various cultural contexts and languages persists. These constraints highlight the need for more flexible and holistic models that can address the complexity and multi-faceted nature of bias in social media content.

Table 1 summarizes the key related works, highlighting their methods, merits, and limitations.

Research Gaps and Problem Identification

Based on the analysis of existing literature, the following research gaps have been identified:

1. Generalizability: Most models are evaluated on narrow domains (e.g., gender-only or race-only datasets) and fail to generalize across diverse bias types. Complex models such as SVM and MLP tend to overfit to dataset-specific patterns, as evidenced by their skewed confusion matrices
2. Accuracy-Fairness Trade-off: Efforts to mitigate specific biases often reduce overall classification accuracy. Existing approaches lack mechanisms to balance detection performance with classification fairness
3. Interpretability: Deep learning approaches (BERT, LSTM, GCN) achieve competitive performance but offer limited transparency in how individual words or features contribute to bias decisions
4. Binary Oversimplification: Many models reduce bias to a binary label without providing quantitative word-level bias scores that enable further analysis
5. Threshold Sensitivity: Few approaches address the critical role of classification thresholds and their adaptation to dataset characteristic

Table 1: Summary of Related Work: Methods, Merits, and Limitations

Study	Method	Merit	Limitation
Shu et al. (2017)	CNN, crowdsourcing, knowledge graphs	Comprehensive multi-model framework with FakeNewsNet dataset	Does not directly address bias; focused on fake news
Kulshrestha et al. (2017)	TF-IDF, cosine similarity	Quantifies input, ranking, and output bias separately	Assumes strong political affiliations; ignores moderate views
Alsaad et al. (2018)	SEM	Differentiates racism from religious bias using large-scale surveys	Limited to survey-based data; no automated detection
Mozafari et al. (2020)	BERT, LLR	First to use BERT classifiers for social media bias with n-gram analysis	Limited to AAE vs SAE; reduced accuracy on racial subgroups
Papakyriakopoulos et al. (2020)	Word embeddings, SVM	Demonstrates cultural bias in embeddings; proposes mitigation	Gender debiasing difficult in morphologically rich languages
Ghosh et al. (2021)	K-means, Perspective API	Detects nation-specific toxic language and emerging expressions	Substitution approach may miss context dependent bias
Guo et al. (2022)	ACLMM (Bi-LSTM + attention)	Multi-task emotion classification at ~80% accuracy	Domain-specific to COVID-19 Chinese tweets
Singh et al. (2023)	GLIdE (TransE + BERT + GCN)	Multi-component architecture for gender bias extraction	High model complexity; limited to gender bias
Lin et al. (2025)	LLM-based detection	Reveals disparities between LLM and human bias perception	Questions reliability of automated bias assessment

The proposed framework addresses these gaps by providing an interpretable, Bayesian word-level scoring mechanism with adaptive threshold optimization, designed for consistent performance across heterogeneous bias datasets.

Problem Definition

Given social media feed data, we formulated a bias detection algorithm that identifies biased entities and provides a measure of bias for the entity. The objectives of the models are as follows:

1. Identify and assign bias scores to the words used in the messages
2. Allow fine-tuning of weights to better mitigate bias in the dataset

Materials and Methods

Figure 1 outlines the various stages of the proposed bias-detection model. The system comprises the following key stages:

- Data Preprocessing: The input tweets were cleaned, lemmatized, and stopwords were removed to prepare the data for analysis. A word-frequency dictionary was computed to capture the occurrence of unique words
- Bias Ratio Computation: The frequency of words in biased and non-biased tweets was compared to calculate a bias ratio for each word
- Normalization and Statistical Analysis: The calculated bias ratios were normalized using Min-Max scaling, and statistical measures such as mean, standard deviation, and z-scores were derived for each word
- Bayesian Scoring: Bayesian probability was combined with word frequencies and z-scores to compute a final bias score for each word
- Performance Evaluation: The model's performance was assessed using metrics such as accuracy, precision, recall, F1-score, and ROC AUC. An optimal threshold for bias detection was identified to classify tweets as biased or non-biased effectively

Certain standard symbols have been used to define key elements and collections used throughout all algorithms. These include:

- $\mathbb{N}$: The set of positive integers, used to store word frequencies
- v : The vocabulary of all distinct words in the training dataset D_{train}
- $\mathbb{R}$: The set of real numbers, used to define the range of bias ratios

The following subsections delve into each of these stages, providing detailed algorithms and explanations.

Preprocessing and Word Frequency Computation

The preprocessing step involved cleaning and organizing the text data for analysis. This includes removing stopwords, lemmatizing words, and calculating word frequencies. Below, we define the key elements used in Algorithm 1.

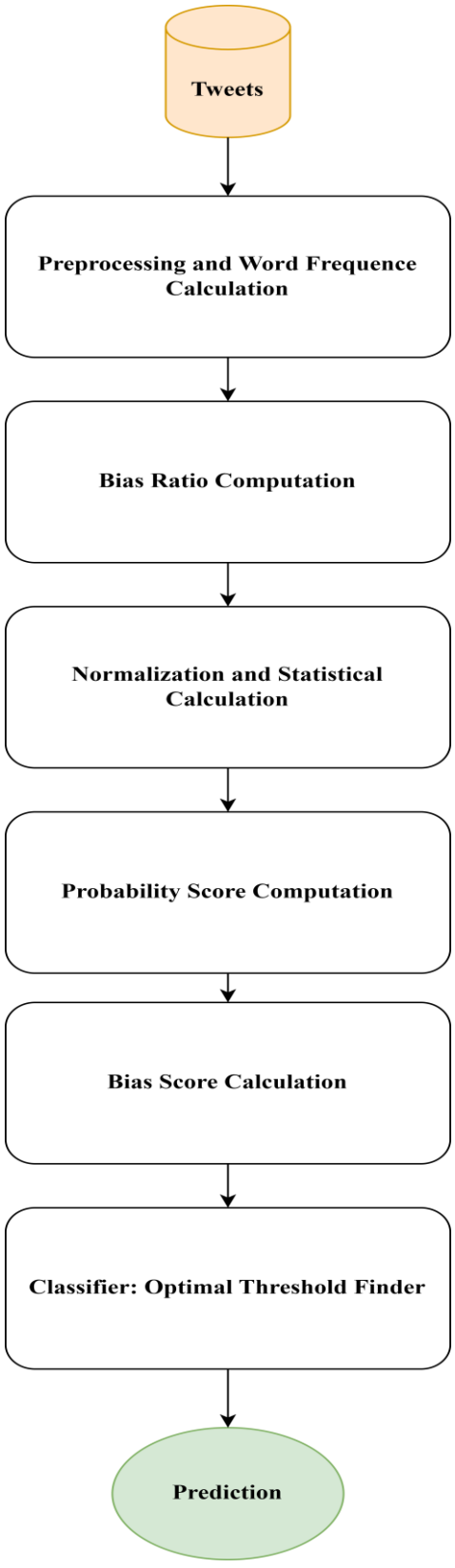

Fig. 1: System Architecture

Algorithm 1: Preprocessing and Word Frequency Calculation

1: **Input**:

- Training dataset D_{train}
- Test dataset D_{test}

2: **Output**:

- Word frequency dictionary $F : V \rightarrow \mathbb{N}$

3: Load D_{train} and D_{test}
4: Define stopword set $S \subset v$
5: Initialize lemmatizer $L : v \rightarrow B$; B is the set of base forms of words
6: Initialize empty word frequency dictionary $F : v \rightarrow \mathbb{N}$
7: **for** each text instance $t_i \in D_{\text{train}}$ **do**
8: **for** each word $w_j \in t_i$ **do**
9: Convert to lowercase and lemmatize:
10: $w'_j \leftarrow \mathscr{L}(w_j.\text{lowercase})$
11: **if** $w'_j \notin \mathscr{S}$ then
12: if $w_j \in /$ domain(F) **then**
13: $F(w'_j) \leftarrow 1$
14: else
15: $F(w'_j) \leftarrow F(w'_j)+1$
16: **end if**
17: **end if**
18: **end for**
19: **end for**
20: Sort F by descending frequency
21: Load Wordnet dataset D_{Wordnet}
22: Initialize Wordnet Hashtable H_W

- B: The set of base forms of words, used in lemmatization to standardize inflected forms
- S $\subset$ v: The set of stopwords, i.e., common uninformative words to be excluded from analysis
- L: $v \rightarrow$ B: The lemmatizer, which maps a word to its base form (root word or lemma). For example, L ("running") = "run"

Word Frequency Calculation: The word frequency dictionary F is constructed by iterating each text instance t_i in the text column T of the training dataset D_{train}. For each word w_j in t_i

1. Convert $w'j$ to lowercase and apply lemmatization using lemmatizer L
2. If the resulting word $w'j$ is not a stopword, update its frequency in F:

- If $w'j \in /$ domain(F), initialize $F(w'j) = 1$

DOI: 10.3844/jcssp.2026.1552.1568

- Otherwise, increment the frequency

The final word frequency dictionary F is sorted in descending order by word frequency.

Wordnet Hashtable: To ensure the standardization of terminology, we employed the SentiWordNet dataset ($D_{WordNet}$), which is essentially WordNet enhanced with corresponding sentiment scores. As a sequitur to our discussion in the introduction, it is important to note that while the sentiment scores (PosScore and NegScore) are not relevant to our objectives, we use the wordlist exclusively as a reference standard for comparison. The hashtable thus constructed shall be referred to as H_W.

Bias Ratio Computation

The Bias Ratio Computation performed in Algorithm 2 quantifies the relative frequency of words in biased versus unbiased text instances within the training dataset D_{train}. The following additional components have been defined in the algorithm:

- Cbias: $F_y\ (w'_j) \rightarrow \mathbb{N}$: A dictionary storing the frequency of each word in biased samples (y = 1)
- Cnon-bias: $F_y\ (w'_j) \rightarrow \mathbb{N}$: A dictionary storing the frequency of each word in non-biased samples (y = 0)
- R (w'_j): $F\ (w'_j)\ 7\rightarrow\ \mathbb{R}$: The bias ratio dictionary, storing the ratio of biased to non-biased word frequencies, for each unique word w'_j encountered in the training dataset

The algorithm iterates through D_{train}, separating text instances by their labels (y = 1 for biased and y = 0 for non-biased). For each word w'_j , its lowercase lemmatized form w'_j = L (w_j. lowercase) is verified. If w'_j exists in H_W (WordNet Hashtable), its count is updated in the respective frequency dictionary. (C_{bias} or $C_{non\text{-}bias}$)

Bias Ratio Computation: For each word w ′ j present in both C_{bias} and $C_{non\text{-}bias}$, the bias ratio $R(w'_j)$ is computed as in (1):

$$R(w'_j) = \frac{C_{bias}(w'_j)}{C_{non-bias}(w'_j)} \quad (1)$$

Sorting and Output: The resulting dictionary $\mathbb{R}$ is sorted by bias ratio in descending order to highlight words with the highest relative bias. Only words with valid bias and non-bias counts are considered.

Algorithm 2: Bias Ratio Computation

1: **Input**:
- Training dataset D_{train}
- WordNet Hashtable H_W

2: **Output**:
- Word ratio dictionary $R : F(w'_j) \rightarrow \mathbb{R}$

3: Initialize empty frequency dictionaries:
- $C_{bias} : F_y(w'_j)(y = 1) \rightarrow \mathbb{N}$
- $C_{non\text{-}bias} : F_y(w'_j)(y = 0) \rightarrow \mathbb{N}$

4: **for** each text instance $t_i \in D_{train}$ with label y = 1 (biased samples) **do**
5: **for** each word $w_j \in t_i$ **do**
6: Convert w_j to lowercase and lemmatize: $^{w'}{}_j \leftarrow L(w_j.$ lowercase)
7: **if** $w'j \in$ domain(H_W) **then**
8: Increment Cbias($w'j$) ←Cbias($w'j$)+1
9: **end if**
10: **end for**
11: **end for**
12: **for** each text instance $t_i \in D_{train}$ with label y = 0 (nonbiased samples) **do**
13: **for** each word $w_j \in t_i$ **do**
14: Convert w_j to lowercase and lemmatize: $w'_j \leftarrow L(w_j.$lowercase)
15: **if** $w'_j \in$ domain(H_W) **then**
16: Increment $C_{non\text{-}bias(}w'j) \leftarrow C_{non\text{-}bias}(w'j)$+1
17: **end if**
18: **end for**
19: **end for**
20: Initialize empty ratio dictionary $R : F(^{w'}{}_j) \rightarrow \mathbb{R}$
21: for each word $w'_j \in$ domain(C_{bias}) do
22: if $w'j \in$ domain(Cnon-bias) then
23: Compute bias ratio R(w'_j)
24: **end if**
25: **end for**
26: Sort R by values in descending order
27: **return** R

This helps in identifying words that disproportionately appear in biased text instances, which can be valuable for understanding linguistic patterns associated with bias. The sorted bias ratio dictionary R provides a ranked list of such words, enabling further analysis or visualization.

This helps in identifying words that disproportionately appear in biased text instances, which can be valuable for understanding linguistic patterns associated with bias. The sorted bias ratio dictionary R provides a ranked list of such words, enabling further analysis or visualization.

Algorithm 3: Normalization and Statistical Calculation

1: **Input**: Sorted word ratio dictionary $R : F(w'_j) \rightarrow \mathbb{R}$ (from Bias Ratio Computation)
2: **Output**: Z-scores dictionary $Z : F(w'_j) \rightarrow \mathbb{R}$
3: Initialize empty dictionary $R_{norm} : F(^{w}{}_j) \rightarrow \mathbb{R}$
4: **for** each word $w'_j \in$ domain (R) **do**
5: Append normalized word ratio $R_{norm}(w'_j)$ of word $^{w'}{}_j$ in (R) using Min-Max normalization
6: **end for**

7: Compute the mean μ and standard deviation σ of R_{norm}
8: Initialize empty dictionary $Z : F(w'_j) \rightarrow \mathbb{R}$
9: **for** each word $w'_j \in \text{domain}(R_{\text{norm}})$ **do**
10: Compute z-score $z(w'_j)$
11: **end for**
12: **return** z

Normalization and Statistical Calculation

Algorithm 3 normalizes the word ratios and computes statistical measures such as the mean, standard deviation, and z-scores.

The algorithm applies Min-Max normalization to each word ratio, transforming the values into the range [0,1] using (2):

$$R_{norm}(w'_j) = \frac{R(w'_j) - \min(R(w'_j))}{\max(R(w'_j)) - \min(R(w'_j))} \tag{2}$$

The mean (μ) was calculated as the arithmetic average of the normalized word ratios, defined in (3):

$$\mu = \frac{1}{|domain(R_{norm})|} \sum_{w \in domain(R_{norm})} R_{norm}(w'_j) \tag{3}$$

The standard deviation (σ) measures the spread of the normalized word ratios around the mean and is given by (4):

$$\sigma = \sqrt{\frac{1}{|domain(R_{norm})|} \sum_{w \in domain(R_{norm})} (R_{norm}(w'_j) - \mu)^2} \tag{4}$$

After normalization, the z-score for each word w is computed using (5):

$$z(w'_j) = \frac{R_{norm}(w'_j) - \mu}{\sigma} \tag{5}$$

The resulting z-scores indicate the extent to which a word's normalized ratio deviates from the mean, expressed in terms of standard deviations. The algorithm returns dictionary z: $W \rightarrow \mathbb{R}$ containing the z-scores for each word.

Algorithm 4: Bayesian Score Computation

1: **Input**:
- Word frequency dictionary $f : V \rightarrow \mathbb{N}$
- Z-scores dictionary $z : F(w'_j) \rightarrow \mathbb{R}$
- Total word frequency $P(w'_j)$

2: **Output**:
- Bayesian scores dictionary $b : F(w'_j) \rightarrow \mathbb{R}$
- Tweet scores dictionary $t : D_{\text{test}} \rightarrow \mathbb{R}$

3: Initialize empty dictionary $b : F(w'_j)$ $\mathbb{R}$
4: **for** each word $w'_j \in \text{domain}(z)$ **do**
5: Compute the Bayesian score $b(w'_j)$
6: Add $b(w'_j)$ to the dictionary b
7: **end for**
8: Initialize empty dictionary $t : D_{\text{test}} \rightarrow \mathbb{R}$
9: **for** each tweet $t_i \in D_{\text{test}}$ **do**
10: Convert t_i to lowercase and tokenize
11: Compute tweet_score(t_i):
12: Add tweet_score(t_i) to the dictionary t
13: **end for**
14: **return** b and t

Bayesian Score Computation

Algorithm 4 computes the Bayesian score for each word to quantify its association with bias in the dataset. The following steps are performed:

1. Compute P_{biased} the probability of a tweet being biased is defined as (6):

$$p_{biased} = \frac{l_{bias}}{l_{total}} \tag{6}$$

Where l_{bias} is the number of biased tweets, and l_{total} is the total number of tweets:

2. Compute $P(w'_j)$ The probability of a word w'_j occurring in the dataset is given by (7):

$$P(w'_j) = \frac{f(w'_j)}{l_{total}} \tag{7}$$

Where $f(w'_j)$ is the frequency of word w'_j in the dataset:

3. Compute Bayesian Score for Each Word For each word $w'_j \in$ domain (z), the Bayesian score is calculated as (8):

$$b(w'_j) = z(w'_j) \times \frac{P_{biased}}{p(w'_j)} \tag{8}$$

4. Compute Tweet Scores The total score for a tweet t is computed as (9):

$$tweet\_score_t = \begin{cases} \sum_{0} w'_j \in_t b\left[w'_j\right] & if\ w'_j \in b \\ & otherwise \end{cases} \tag{9}$$

Where w'_j represents each word present in the tweet t. Certain words from t may not have been encountered while training. The bayesian bias scores for such words were presumed to be 0:

5. Output Results The algorithm outputs the Bayesian scores dictionary b and tweet scores dictionary t

The Bayesian score formulation in (8) is motivated by Bayes' theorem. In a standard Naive Bayes framework,

the posterior probability of a tweet being biased given a word w'_j in (10):

$$P(biased \mid w'_j) = \frac{P(w'_j \mid biased) \bullet P(biased)}{P(w'_j)} \quad (10)$$

The z-score $z(w'_j)$ serves as a normalized proxy for the likelihood $P(w'_j \mid \text{biased})$, capturing how strongly a word's bias ratio deviates from the population mean. Multiplying by $\frac{P_{biased}}{P(w'_j)}$ mirrors the prior-to-evidence ratio, weighting each word's bias signal by the base rate of bias relative to the word's frequency. This formulation shares the conditional independence assumption with Naive Bayes: Each word contributes independently to the overall tweet score. While this is a simplification (as word co-occurrences carry contextual information), this assumption enables interpretable, word-level attribution and is empirically validated by the competitive performance demonstrated in Section 5. The additive aggregation in (9) corresponds to the log-space product of independent word-level posterior contributions, consistent with the Naive Bayes scoring paradigm.

Performance Evaluation and Optimal Threshold Detection

After assigning agglomerative scores to each tweet in the dataset, the Algorithm 5 identifies the best threshold tˆ that separates biased from non-biased tweets by maximizing various performance metrics, such as accuracy, F1 score, and ROC-AUC score.

Threshold Exploration: The algorithm performs a grid search over threshold values θ within the range [−400,400] in steps of 50. This range was selected to encompass the full observed range of tweet scores across all three datasets, ensuring that no viable threshold is excluded. The step size of 50 provides a practical trade-off between search resolution and computational cost while remaining consistent with the magnitude of tweet score variations. The Accuracy vs. Threshold plots in Figure 2 empirically confirm that the optimal thresholds lie well within the search range and that the accuracy curves exhibit smooth, unimodal peaks, indicating that small perturbations around the optimal threshold do not cause sharp performance degradation. For each tweet Ti ∈ domain(t), if the tweet score t(Ti) exceeds the threshold θ, it is classified as biased:

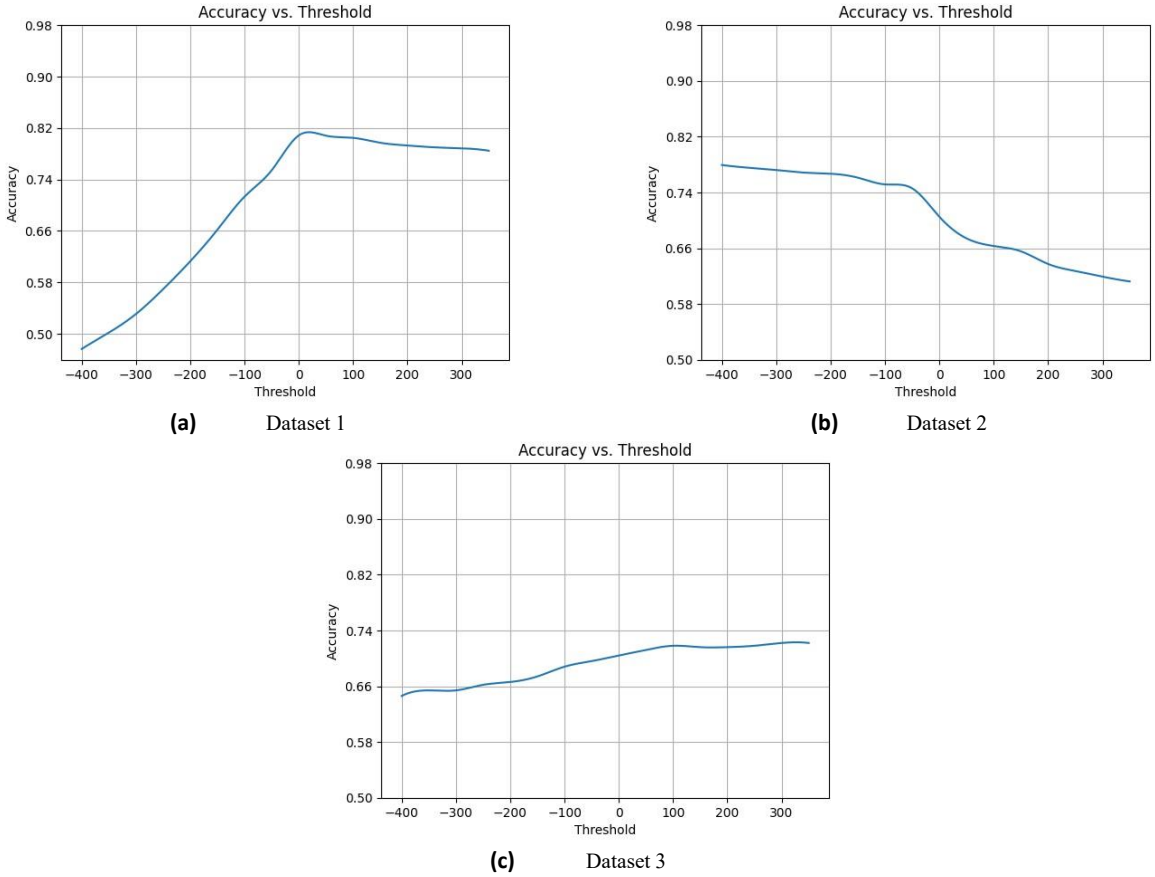

Fig. 2: Accuracy vs Threshold

Algorithm 5: Performance Evaluation and Optimal Threshold Detection

1: **Input**: Tweet scores dictionary $t : D_{\text{test}} \rightarrow \mathbb{R}$, Threshold values T
2: **Output**: Best threshold $t^{\wedge}$, Evaluation metrics: Accuracy, F1 score, Precision, Recall, Confusion Matrix, ROC Curve
3: Initialize variables for best threshold $t^{\wedge}$, accuracy, F1 score, precision, recall, and confusion matrix
4: **for** each threshold $\theta \in T$ **do**
5: **for** each tweet $T_i \in$ domain(t) **do**
6: Assign bias label as bias_label_{Ti}
7: **end for**
8: Compute evaluation metrics such as Accuracy F1 Score, etc. for current threshold θ:
9: **if** current accuracy and F1 score exceed previous best values **then**
10: Update best threshold $t^{\wedge}$, accuracy, precision, recall, and F1 score
11: **end if**
12: **end for**
13: Plot ROC curve and Accuracy vs. Threshold graphs
14: **return** $t^{\wedge}$, Accuracy, F1 score, Precision, Recall, Confusion Matrix, ROC Curve

$$bias\_label_{Ti} = \begin{cases} 1 & if\ t(Ti) > \theta \\ 0 & otherwise \end{cases}$$

The results are stored at each threshold. Alternative threshold selection strategies such as Youden's J statistic (J = Sensitivity + Specificity−1) could be employed; however, the grid-search approach was preferred as it jointly optimizes accuracy and F1 score rather than relying solely on ROC-derived metrics. Refinement of threshold granularity is identified as a direction for future optimization:

Performance Evaluation: For each threshold θ, the algorithm calculates several performance metrics.

a) *Accuracy*: The proportion of correctly classified tweets
b) *Precision*: The number of true positives divided by the total number of positive predictions (including both true and false positives)
c) *Recall*: The number of true positives divided by the total number of true samples (including true positives and false negatives)
d) *F1 Score*: The harmonic mean of precision and recall
e) *Confusion Matrix*: A matrix used to determine the performance of the classification model for a given set of test data
f) *ROC AUC Score*: A metric that measures the classifier's ability to distinguish between biased and non-biased tweets

1. Optimal Threshold Selection: The threshold t^ with the highest accuracy score is selected. The final tweet classification is based on this optimal threshold
2. ROC Curve and Accuracy Plot: The ROC curve was plotted, and accuracy scores for various thresholds were visualized

The algorithm returns the best threshold t^, the best accuracy score, the best ROC AUC score, and the best F1 score corresponding to the selected threshold.

Datasets Used

Table 2 summarizes the datasets used in the simulations. Dataset 1 contains 24,783 tweets, categorized as hate speech (5%), offensive language (77%), and neutral content (16%). The first two categories include homophobic, sexist, and racist tweets (i.e., offensive content), whereas the neutral class contains non-offensive tweets. Because our algorithm performs binary classification, the hate-speech and offensive-language classes were merged into a single offensive class.

Dataset 2 includes 5,000 social media posts from political entities, with labels for neutral (74%) and partisan (26%) content. Dataset 3 comprises 20,989 tweets aggregated from multiple sources (Mandl et al., 2019; Dalvi, 2024; Founta et al., 2018). It focuses on hate speech related to racial and gender-based discrimination, with 30% labeled as biased and 70% as unbiased.

Table 2: Dataset description

ID	Dataset	Records	Data Distribution
1	Labeled Data	24783	Hate Speech: 1430 Offensive Language: 19190 Neither: 4163
2	Political Social Media Posts	5000	Neutral: 74% Partisan: 26%
3	X DF Dataset	20989	Biased: 30% Unbiased: 70%

Together, these three datasets provide a varied and balanced evaluation setting across topics, scenarios, dataset sizes, and class distributions. A 90:10 train–test split was used for all experiments, consistent with our prior comparative study (Rao et al., 2026). This split maximizes training data for word-level frequency estimation; smaller training sets can underrepresent rare but informative words, reducing the reliability of bias-ratio computation. The same split was applied uniformly across all six algorithms (GNB, SVM, MLP, AdaBoost, Decision Tree, and the proposed method), ensuring fair comparison. Consistent behavior across datasets that differ in size (5,000–24,783 records), domain (political

content, hate speech, racial/gender discrimination), and class balance (majority class from 70 to 84%) also provides an implicit robustness check.

Metric Analysis

We compared our proposed algorithm with the standard machine learning algorithms listed below:

- GNB (Gaussian Naive Bayes): A Naive Bayes Classifier that assumes data points (features) are independent of each other, and that input features follow a continuous Gaussian distribution. Posterior beliefs are updated after observing new data
- AdaBoost: An ensemble-based classifier that combines the output of several weak classifiers. It iteratively focuses on the errors made by the current classifier to improve future classifiers. The final output is obtained through majority voting
- MLP (Multi-Layer Perceptron): A Multilayer Perceptron-based Neural Network algorithm, using the ReLU activation function and backpropagation to adjust errors
- SVC (Support Vector Classifier): A Support Vector Machine-based classifier that attempts to find an optimal hyperplane that separates the data into different classes. It uses kernel functions like Radial Basis Function (RBF) to project lower-dimensional data into higher dimensions where separation is possible
- Decision Tree Classifier: Classification and Regression Trees (CART) recursively splits the data into branches based on specific features. Leaf nodes represent target values, and metrics like Gini Impurity and Information Gain are used to decide the best features to split on

Results and Discussion

This section presents a comparative performance analysis of the algorithms and the thresholding optimization in the proposed algorithm across the three different datasets referred in Table 2. Referring to Table 3 for metrics comparison, the Accuracy vs Threshold graph Figure 2 and ROC curves Figure 3 demonstrate the effectiveness of proposed approach. The metrics table highlights the comparative performance of all algorithms, showing minor percentage variations across the different evaluation metrics.

Table 3: Summary of Results

Algorithm Name	Metric	Dataset 1	Dataset 2	Dataset 3
Gaussian Naive Bayes	Accuracy	0.779347	0.764000	0.842306
	Precision	0.591543	0.565705	0.838239
	Recall	0.567836	0.512337	0.764846
	F1	0.574523	0.478338	0.788595
	Confusion Matrix	$\begin{bmatrix} 106 & 205 \\ 342 & 1826 \end{bmatrix}$	$\begin{bmatrix} 376 & 107 \\ 11 & 6 \end{bmatrix}$	$\begin{bmatrix} 1413 & 259 \\ 72 & 355 \end{bmatrix}$
Decision Tree	Accuracy	0.805567	0.642000	0.838971
	Precision	0.593427	0.497475	0.825439
	Recall	0.566610	0.497805	0.781196
	F1	0.574400	0.496169	0.797597
	Confusion Matrix	$\begin{bmatrix} 85 & 177 \\ 305 & 1912 \end{bmatrix}$	$\begin{bmatrix} 295 & 101 \\ 78 & 26 \end{bmatrix}$	$\begin{bmatrix} 1355 & 236 \\ 102 & 406 \end{bmatrix}$
Multi-layer Perceptron	Accuracy	0.832594	0.744000	0.849452
	Precision	0.645764	0.374245	0.846926
	Recall	0.517553	0.496000	0.781552
	F1	0.495883	0.426606	0.803741
	Confusion Matrix	$\begin{bmatrix} 19 & 23 \\ 392 & 2045 \end{bmatrix}$	$\begin{bmatrix} 372 & 125 \\ 3 & 0 \end{bmatrix}$	$\begin{bmatrix} 1398 & 244 \\ 72 & 385 \end{bmatrix}$
AdaBoost	Accuracy	0.828963	0.736000	0.856122
	Precision	0.770076	0.368737	0.844921
	Recall	0.522968	0.498645	0.808191
	F1	0.499793	0.423963	0.822730
	Confusion Matrix	$\begin{bmatrix} 22 & 9 \\ 415 & 2033 \end{bmatrix}$	$\begin{bmatrix} 368 & 131 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1354 & 206 \\ 96 & 443 \end{bmatrix}$
Support Vector Machine	Accuracy	0.837434	0.738000	0.859457
	Precision	0.918652	0.369000	0.866639
	Recall	0.502469	0.500000	0.786453
	F1	0.460638	0.424626	0.812550

	Confusion Matrix	$\begin{bmatrix} 2 & 0 \\ 403 & 2074 \end{bmatrix}$	$\begin{bmatrix} 369 & 131 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1427 & 243 \\ 52 & 377 \end{bmatrix}$
Proposed Algorithm	Accuracy	0.775313	0.712000	0.824678
	Precision	0.956631	0.621391	0.807885
	Recall	0.812798	0.637440	0.756530
	F1	0.859379	0.626293	0.769216
	Confusion Matrix	$\begin{bmatrix} 220 & 165 \\ 392 & 1702 \end{bmatrix}$	$\begin{bmatrix} 299 & 77 \\ 67 & 57 \end{bmatrix}$	$\begin{bmatrix} 1382 & 114 \\ 254 & 349 \end{bmatrix}$

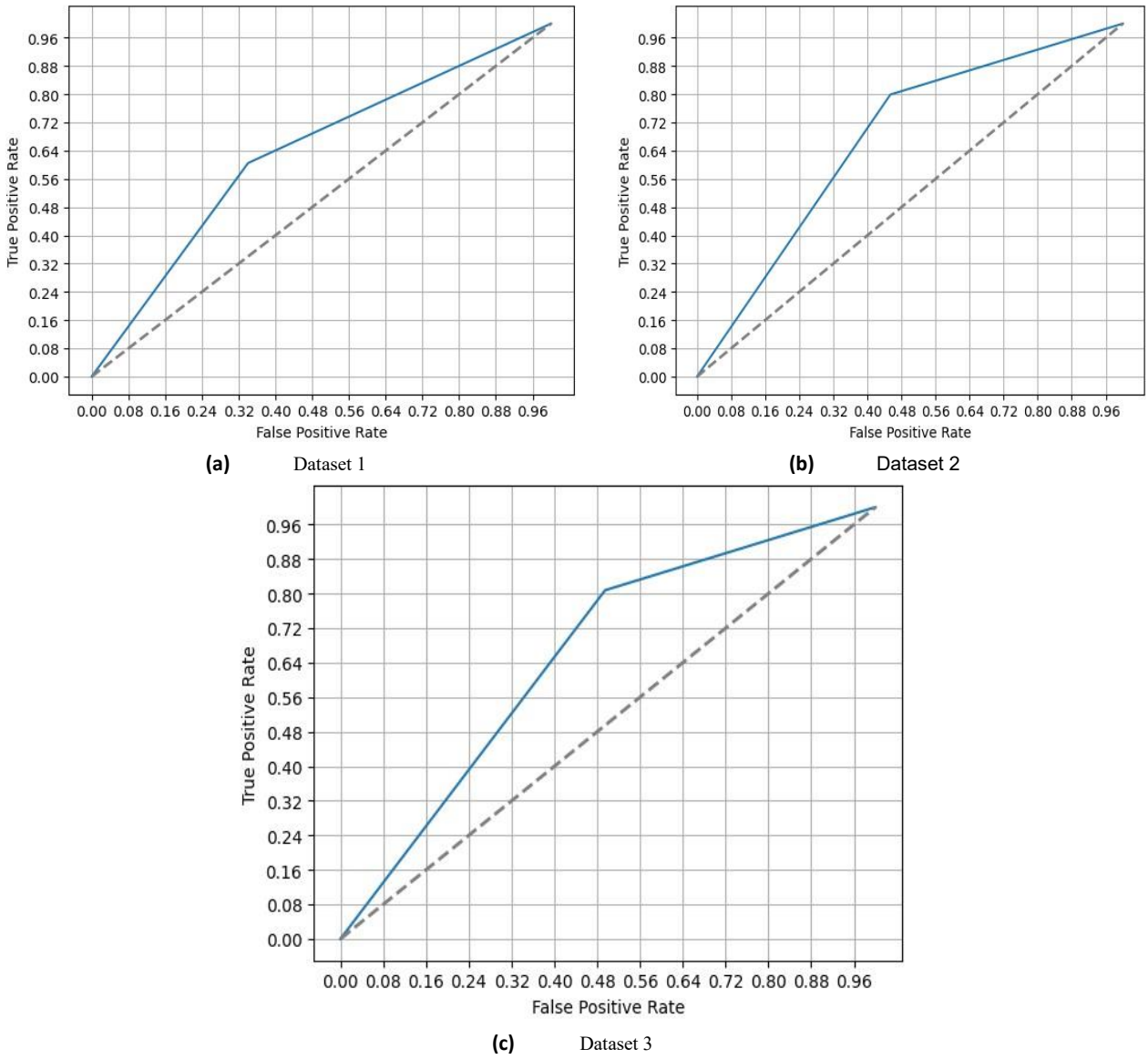

Fig. 3: ROC Curves

Accuracy

In terms of Accuracy, the large and complex algorithms demonstrated an overall dominance on all datasets. The SVM and MLP models demonstrated the best overall accuracy scores followed closely by AdaBoost. These models are complex in their structure, thus able to demonstrate higher abilities in capturing

various intricacies. Simpler models like GNB, Decision trees and our proposed algorithm yield accuracy scores with a deviation of around (2-3%) across the datasets. Notably, the Decision tree algorithm struggles to capture the intricacies of nuanced topics like politics, reflected by the scores in Dataset 1.

Precision, Recall, and F1 Score

Our proposed algorithm achieved strong performance in Datasets 1 and 2, scoring the highest across Precision, Recall, and F1 metrics. Although it shows a slight decline in performance on Dataset 3, it remains competitive with the top performers, and its scores do not fall far from those of other leading models. The ROC curves further highlight the robustness of the proposed approach, showing a well-defined shape indicative of reliable classification.

More complex models, such as SVM and MLP, tend to perform better on Dataset 3 because of their less nuanced topics, which make it easier for these models to classify accurately (e.g., sex- and race-based hate speech). Meanwhile, GNB and our proposed algorithm demonstrate consistent performance across all datasets, with our model showing less bias in classifications than GNB. This consistency reflects the balance achieved by the simpler structure of the proposed algorithm and threshold optimization, making it adaptable to different data complexities without overfitting.

Confusion Matrix

The confusion matrices reveal biases that are present in more complex models. This also highlights the skewed predictions stemming from bias. The confusion matrix of SVM on Dataset 1 shows a clear tendency toward positive classifications, with zero true negatives detected. This indicates the potential for overfitting and failure to generalize effectively. Similar patterns are observed in complex models like AdaBoost and Multi-Layer Perceptron (MLP), where the model often leans heavily toward either positive or negative classifications depending on the dataset, resulting in significant misclassification.

In contrast, simpler models such as GNB and Decision Tree exhibit a more balanced classification pattern, suggesting that their structure may help mitigate bias. Our Proposed Algorithm demonstrates the most balanced confusion matrix overall, showing a lower tendency to lean toward any specific classification. This balanced performance underscores the advantages of simplicity and feature independence, as overly complex models risk introducing bias and compromising generalization. This analysis suggests that simpler models, combined with threshold optimization as seen in our proposed approach, can achieve robust, unbiased classification.

Threshold Optimization in Proposed Algorithm

The Accuracy versus Threshold diagrams provide insights into the characteristics of each dataset. In the first dataset, the optimal threshold lies on the negative side, suggesting that a wide range of words is utilized, with many having low bias. This pattern aligns with the political context, where nuanced and diverse vocabulary tends to be employed, reflecting subtler expressions and less overt bias.

In contrast, the other two datasets displayed a positive threshold peak, indicating a higher frequency of biased word usage in biased texts. In these cases, context plays a smaller role, as specific, frequently repeated words (often associated with sexist or racist language) dominate, making the bias more explicit.

Analysis of Performance Variation Across Datasets

The proposed algorithm's accuracy varies from 71.2% (Dataset 2) to 82.5% (Dataset 3), reflecting the inherent difficulty of each bias detection task. Dataset 2 (political partisanship) yields the lowest accuracy because political bias is expressed through nuanced vocabulary, contextual framing, and rhetorical devices rather than through explicitly biased terms. A word-level approach inherently captures less of this context-dependent bias, explaining the performance gap. In contrast, Dataset 3 (racial and gender-based hate speech) contains more lexically distinctive biased terms, enabling higher word-level discriminability.

Despite lower raw accuracy compared to SVM (83.7%) and MLP (83.3%) on Dataset 1, the proposed algorithm achieves substantially higher F1 scores (0.859 vs. 0.461 and 0.496 respectively). This disparity arises because complex models achieve high accuracy by predominantly predicting the majority class, as evidenced by their confusion matrices showing near-zero true negatives in SVM and near-zero minority-class detections in MLP and AdaBoost. The proposed algorithm's balanced confusion matrices indicate that it detects both biased and non-biased content reliably, making it more suitable for practical deployment where false negatives (missed bias) are costly.

Figure 2 shows the path traversed by the classifier to determine the optimal threshold.

Classification Balance Analysis

To quantify classification balance, Table 4 reports the True Positive Rate (TPR) and True Negative Rate (TNR) for each algorithm across all three datasets. A well-calibrated classifier should maintain meaningful values for both rates rather than maximizing one at the expense of the other.

Table 4: Classification Balance: True Positive Rate (TPR) and True Negative Rate (TNR) across datasets

	Dataset 1		Dataset 2		Dataset 3	
Algorithm	TPR	TNR	TPR	TNR	TPR	TNR
GNB	0.8422	0.3408	0.3529	0.7784	0.8313	0.8450
Dec. Tree	0.8624	0.3244	0.2500	0.7449	0.7992	0.8516
MLP	0.8391	0.4523	0.0000	0.7484	0.8424	0.8514
AdaBoost	0.8304	0.7096	0.0000	0.7374	0.8218	0.8679
SVM	0.8373	1.0000	0.0000	0.7380	0.8787	0.8544
Proposed	0.8127	0.5714	0.4596	0.7952	0.5787	0.9237

On Dataset 1, complex models exhibit pronounced imbalance. SVM achieves a perfect TNR of 1.000 but at the cost of minority-class detection, while MLP and AdaBoost similarly favor TNR over TPR. GNB and Decision Tree achieve high TPR (0.842 and 0.862) but substantially lower TNR (0.341 and 0.324). The proposed method maintains a more balanced profile (TPR 0.813, TNR 0.571), avoiding such extreme skew.

On Dataset 2, SVM, MLP, and AdaBoost exhibit complete classification collapse with TPR of 0.000, assigning every sample to the majority class. GNB and Decision Tree retain partial minority-class sensitivity but remain heavily skewed. The proposed method achieves the highest TPR (0.460) and TNR (0.795), demonstrating the most balanced detection on this challenging dataset.

On Dataset 3, performance differences narrow due to the more lexically distinctive vocabulary. The proposed method achieves the highest TNR (0.924) alongside a competitive TPR (0.579), yielding the most balanced profile. SVM attains the highest TPR (0.879) but at a lower TNR (0.854), and other models exhibit comparable trade-offs.

Across all three datasets, the proposed algorithm is the only classifier that consistently avoids degenerate cases where one rate approaches 1.0 while the other collapses toward 0. This robustness to class-distribution shifts and domain variation confirms that the Bayesian scoring mechanism, combined with adaptive threshold optimization, preserves minority-class detectability without sacrificing majority-class performance, reinforcing its practical advantage for scenarios where balanced detection is essential.

Conclusion and Future Work

This paper presented a Bayesian probabilistic framework for bias detection in tweets, combining WordNet-based vocabulary filtering, z-score normalization of bias ratios, and adaptive threshold optimization. The proposed algorithm was evaluated against five established classifiers across three diverse datasets. Key findings are:

- The algorithm achieves accuracy ranging from 71.2% to 82.5% and the highest F1 score of 0.859 on Dataset 1, demonstrating competitive performance with substantially simpler computation
- Unlike complex models (SVM, MLP, AdaBoost) that exhibit skewed confusion matrices indicating majority-class bias, the proposed approach maintains balanced classification across all datasets
- The threshold optimization mechanism adapts to dataset characteristics: negative optimal thresholds for nuanced political content and positive thresholds for explicit hate speech, providing interpretable insights into dataset bias profiles
- The word-level Bayesian scoring enables transparent attribution of bias, allowing practitioners to identify which words drive classification decisions

The primary limitation of this work is the reliance on individual word-level features, which cannot capture context-dependent, multi-word, or implicit bias. Additionally, the WordNet vocabulary filter excludes slang, neologisms, and domain-specific terms that may carry bias.

Future work will address these limitations through the following directions:

1. Context-aware bias detection: Integrating transformer-based contextual embeddings (e.g., BERT, RoBERTa) as complementary features to capture phrase-level and sentence-level bias patterns while retaining the interpretability of word-level scores
2. Multi-lingual and cross-cultural extension: Adapting the framework to non-English languages by substituting WordNet with multilingual lexical resources such as BabelNet or Open Multilingual Wordnet, enabling cross-cultural bias analysis
3. Dynamic vocabulary augmentation: Incorporating slang lexicons and evolving language resources to detect bias expressed through informal or emergent vocabulary not present in standard dictionaries
4. Real-time deployment: Developing a streaming pipeline for real-time bias scoring of social media feeds, leveraging the algorithm's stateless and computationally lightweight design
5. Fairness-aware evaluation: Conducting subgrouplevel analysis across demographic

categories (race, gender, political affiliation) using metrics such as equalized odds and demographic parity to formally assess classification fairness

Acknowledgment

The authors gratefully acknowledge the support and facilities provided by the Department of Computer Science at the University of Visvesvaraya College of Engineering. We also extend our thanks to the publishing and editorial team for their efforts in refining this work and facilitating its distribution to a wider audience.

Funding Information

This work received no external funding.

Author's Contributions

Prasanth G Rao: Conceptualized the primary framework and served as the principal ideator for the research.

Harsha Chigurupati: Contributed to related work research, solution conceptualization, and manuscript drafting and review.

Krish Hashia: Contributed to related work research, algorithm design, implementation, and results aggregation and verification.

Thriveni J, P Deepa Shenoy and Venugopal K R: Provided guidance in formulating the problem definition, refining the algorithms, reviewing the experiment results, structuring the overall research experimentation and manuscript drafting.

Ethics

Ethical approval This study did not involve human participants or animals.

Data Sources

All data used in this work were obtained legally from open-source resources.

Conflict of Interest

The authors declare that there are no conflicts of interest related to this work.

Data Availability

The data used in this study are publicly available online. Additional details can be provided by the authors upon reasonable request.

Materials Availability

No new materials were created or used in this study.

Code Availability

The code supporting this work is available from the authors upon reasonable request.

References

Alsaad, A., Taamneh, A., & Al-Jedaiah, M. N. (2018). Does social media increase racist behavior? An examination of confirmation bias theory. *Technology in Society*, *55*, 41–46. https://doi.org/10.1016/j.techsoc.2018.06.002

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the 2010 International Conference on Language Resources and Evaluation (LREC*, 2200–2204.

Dai, W., Tao, J., Yan, X., Feng, Z., & Chen, J. (2023). Addressing Unintended Bias in Toxicity Detection: An LSTM and Attention-Based Approach. *2023 5th International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 375–379. https://doi.org/10.1109/icaica58456.2023.10405429

Dalvi, M. (2024). Twitter sentiments analysis (nlp). *Kaggle*.

Datta, P., Whitmore, M., & Nwankpa, J. K. (2021). A Perfect Storm: Social Media News, Psychological Biases, and AI. *Digital Threats: Research and Practice*, *2*(2), 1–21. https://doi.org/10.1145/3428157

Founta, A., 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. *Proceedings of the International AAAI Conference on Web and Social Media*, *12*(1), 491–500. https://doi.org/10.1609/icwsm.v12i1.14991

Ganguly, S., Kulshrestha, J., An, J., & Kwak, H. (2020). Empirical Evaluation of Three Common Assumptions in Building Political Media Bias Datasets. *Proceedings of the International AAAI Conference on Web and Social Media*, *14*, 939–943. https://doi.org/10.1609/icwsm.v14i1.7362

Ghosh, S., Baker, D., Jurgens, D., & Prabhakaran, V. (2021). Detecting Cross-Geographic Biases in Toxicity Modeling on Social Media. *Proceedings of the Seventh Workshop on Noisy User-Generated Text (W-NUT 2021)*, 313–328. https://doi.org/10.18653/v1/2021.wnut-1.35

Guo, L., Feng, Z., Chi, Y., Wang, M., & Liu, Y. (2022). *Coronavirus statistics causes emotional bias: A social media text mining perspective*. https://doi.org/10.48550/arXiv.2211.08644

Hamborg, F., Donnay, K., & Gipp, B. (2019). Automated identification of media bias in news articles: an interdisciplinary literature review. *International Journal on Digital Libraries*, *20*(4), 391–415. https://doi.org/10.1007/s00799-018-0261-y

Hargittai, E. (2020). Potential Biases in Big Data: Omitted Voices on Social Media. *Social Science Computer Review*, *38*(1), 10–24. https://doi.org/10.1177/0894439318788322

Hollingshead, W., Quan-Haase, A., & Blank, G. (2022). Representativeness and Bias in Social Media Research: Quantitative and Qualitative Approaches to Sampling. *The SAGE Handbook of Social Media Research Methods*, 79–90. https://doi.org/10.4135/9781529782943.n8

Iacus, S. M., Porro, G., Salini, S., & Siletti, E. (2020). Controlling for Selection Bias in Social Media Indicators through Official Statistics: a Proposal. *Journal of Official Statistics*, *36*(2), 315–338. https://doi.org/10.2478/jos-2020-0017

Kaiser, J., Vaccari, C., & Chadwick, A. (2022). Partisan Blocking: Biased Responses to Shared Misinformation Contribute to Network Polarization on Social Media. *Journal of Communication*, *72*(2), 214–240. https://doi.org/10.1093/joc/jqac002

Koçak, B., Ponsiglione, A., Stanzione, A., Bluethgen, C., Santinha, J., Ugga, L., Huisman, M., Klontzas, M. E., Cannella, R., & Cuocolo, R. (2024). Bias in artificial intelligence for medical imaging: fundamentals, detection, avoidance, mitigation, challenges, ethics, and prospects. *Diagnostic and Interventional Radiology*, *31*(2), 75–88. https://doi.org/10.4274/dir.2024.242854

Kulshrestha, J., Eslami, M., Messias, J., Zafar, M. B., Ghosh, S., Gummadi, K. P., & Karahalios, K. (2017). Quantifying Search Bias: Investigating Sources of Bias for Political Searches in Social Media. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 417–432. https://doi.org/10.1145/2998181.2998321

Lin, L., Wang, L., Guo, J., & Wong, K.-F. (2025). Investigating bias in llm-based bias detection: Disparities between llms and human perception. *Proceedings of the 31st International Conference on Computational Linguistics*, 10634–10649.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., & Patel, A. (2019). Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. *Proceedings of the 11th Forum for Information Retrieval Evaluation*, 14–17. https://doi.org/10.1145/3368567.3368584

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2022). A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, *54*(6), 1–35. https://doi.org/10.1145/3457607

Morstatter, F., & Liu, H. (2017). Discovering, assessing, and mitigating data bias in social media. *Online Social Networks and Media*, *1*, 1–13. https://doi.org/10.1016/j.osnem.2017.01.001

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), e0237861. https://doi.org/10.1371/journal.pone.0237861

Papakyriakopoulos, O., Hegelich, S., Serrano, J. C. M., & Marco, F. (2020). Bias in word embeddings. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 446–457. https://doi.org/10.1145/3351095.3372843

Rao, P. G., Chigurupati, H., Hashia, K., Thriveni, J., Deepa Shenoy, P., & Venugopal, K. R. (2026). Detecting Bias in Social Media Using SentiWordNet. *Data Management, Analytics and Innovation*, *1369*, 197–209. https://doi.org/10.1007/978-981-96-5860-2_12

Rodrigo-Ginés, F.-J., Carrillo-de-Albornoz, J., & Plaza, L. (2024). A systematic review on media bias detection: What is media bias, how it is expressed, and how to detect it. *Expert Systems with Applications*, *237*, 121641. https://doi.org/10.1016/j.eswa.2023.121641

Sap, M., Card, D., Gabriel, S., Choi, Y., & Smith, N. A. (2019). The Risk of Racial Bias in Hate Speech Detection. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1668–1678. https://doi.org/10.18653/v1/p19-1163

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36. https://doi.org/10.1145/3137597.3137600

Singh, G., Ghosh, S., & Ekbal, A. (2023). Promoting Gender Equality through Gender-biased Language Analysis in Social Media. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 6210–6218. https://doi.org/10.24963/ijcai.2023/689

Spinde, T., Plank, M., Krieger, J.-D., Ruas, T., Gipp, B., & Aizawa, A. (2021). Neural Media Bias Detection Using Distant Supervision With BABE - Bias Annotations By Experts. *Findings of the Association for Computational Linguistics: EMNLP 2021*, 1166–1177. https://doi.org/10.18653/v1/2021.findings-emnlp.101

Statista. (2024a). Millennials' news consumption in the united states. *Statista*.

Statista. (2024b). Reasons for social media usage worldwide. *Statista*.

Wei, X., Kumar, N., & Zhang, H. (2025). Addressing bias in generative AI: Challenges and research opportunities in information management. In *Information & Management* (Vol. 62, Issue 2, p. 104103). https://doi.org/10.1016/j.im.2025.104103

Wewelwala, S. H., & Sumanathilaka, T. G. D. K. (2024). Hybrid Approaches to Emotion Recognition: A Comprehensive Survey of Audio-Textual Methods and Their Application. *2024 4th International Conference on Advanced Research in Computing (ICARC)*, 167–172. https://doi.org/10.1109/icarc61713.2024.10499740