# Enhancing Hate Speech Detection for Social Media Moderation: A Comparative Analysis of Machine Learning Algorithms

Chelsea Olivia Leo
*Bina Bangsa School*
*Jakarta, Indonesia*
chelseaolivialeo17@gmail.com

Bagus Jati Santoso
*Department of Informatics*
*Institut Teknologi Sepuluh Nopember*
Surabaya, Indonesia
bagus@if.its.ac.id

Baskoro Adi Pratomo
*Department of Informatics*
*Institut Teknologi Sepuluh Nopember*
Surabaya, Indonesia
baskoro@if.its.ac.id

2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA) | 979-8-3503-0922-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAMIMIA60881.2023.10427779

***Abstract*— In our ever-advancing technological age, a pressing concern emerges the challenge of preserving freedom of expression while combatting hate speech. Hate speech, in simple terms, involves hurtful messages directed at specific groups based on attributes like race, gender, and religion. Social media platforms have sought to curb hate speech through user reporting and automated content scrutiny, but their effectiveness is under scrutiny.**

**To improve hate speech detection, we explore several machine learning algorithms, such as Naive Bayes, Random Forest, Decision Trees, and Gradient Boosting. By tweaking these algorithms and comparing their performance across metrics like AUC, CA, F1 score, Precision, and Recall, we aim to identify the best approach. Furthermore, our study reviews the current state of hate speech detection, presents the methodology used, and discusses implications for social media moderation.**

**Our results show Random Forest as the top performer, but challenges remain. Future research may involve advanced classifiers, algorithm hybridization, and deep learning to enhance model performance in addressing the intricate landscape of online hate speech.**

***Keywords—hate speech, detection, social media, machine learning***

## I. Introduction

In this rapidly advancing era of technology, our society grapples with a growing concern: the freedom of expression in the context of hate speech. Hate speech, in common parlance, denotes the act of disseminating hurtful or derogatory messages targeted at specific groups, often categorized by factors such as race, gender, sexual orientation, religion, and physical abilities [1]. Over time, numerous social media platforms have endeavored to combat the proliferation of hate speech in the digital realm. Users are granted the ability to block or report accounts engaged in the dissemination of hateful content. Subsequently, these reported comments undergo thorough scrutiny to identify hate speech indicators, which are automatically removed if they meet the established criteria [2]. However, the efficacy of these measures in curbing hate speech remains questionable. Alarming reports from 2022 suggest that removing hateful comments has become increasingly challenging for social media giants like Twitter, Facebook, and Instagram, with the review processes slowing down [3]. It is evident that data science techniques, including Natural Language Processing (NLP) and specific Machine Learning (ML) algorithms, have struggled to distinguish hate speech from non-hateful content [4] effectively.

To enhance the quality of the hate speech detection process, it is crucial to explore more accurate machine learning algorithms. This can be achieved through a series of systematic tests and experiments, where the tuning parameters of different algorithms, such as Naive Bayes, Random Forest, Decision Trees, and Gradient Boosting, are modified. The results obtained from these diverse algorithms are meticulously compared, focusing on metrics such as F1 score, Precision, and Recall. The algorithm that yields the highest scores across these evaluation criteria should be considered as a viable replacement for the existing algorithm. By doing so, the detection of hate speech across the internet can be significantly expedited, becoming more efficient and accurate in the process.

The battle against hate speech is a complex and multifaceted challenge. It requires the development and implementation of innovative strategies and tools to address the ever-evolving nature of online hate speech. This comparative analysis of machine learning algorithms is a proactive step towards a more effective moderation system. The overarching goal is to facilitate a safer online environment by swiftly identifying and addressing hate speech while minimizing false positives and ensuring that legitimate expressions of free speech are not stifled.

In the following sections of this article, we will investigate the current landscape of hate speech detection, exploring the limitations of existing machine learning algorithms and the need for a more robust approach. We will also present the methodology used in our comparative analysis, detailing the various algorithms and their tuning parameters in Chapter III. The results of our experiments will be thoroughly analyzed, and implications for the field of social media moderation and content filtering will be discussed in Chapter IV. Ultimately, our conclusion will be presented in Chapter V.

## II. Related Works

Mullah and Zainon in [5] reviewed the advances in machine learning algorithms for hate speech detection in social media. They discussed various techniques, including supervised, unsupervised, and deep learning methods, and highlighted the challenges and future directions in this field. Further, Abro et al. [6 conducted a comparative study of automatic hate speech detection using machine learning.

They evaluated the performance of different classifiers, including Naive Bayes, Support Vector Machines, and Random Forest, on a dataset of hate speech tweets. In [7], Raufi and Xhaferri applied machine learning techniques for hate speech detection in mobile applications. They used a dataset of user comments and evaluated the performance of different classifiers, including Decision Trees, Random Forest, and Naive Bayes. Finally, a hate speech detection system for Facebook was developed by Del Vigna et al. in [8]. They used a dataset of user comments and evaluated the performance of different classifiers, including Support Vector Machines, Naive Bayes, and Logistic Regression.

There also exist some literature reviews and studies on hate speech, especially on Twitter. Alkomah and Ma [9] conducted a literature review of textual hate speech detection methods and datasets. They reviewed various studies that used machine learning algorithms for hate speech detection and highlighted the primary datasets, textual features, and machine learning models used in these studies. Malik et al. in [10] conducted a large-scale empirical comparison of deep and shallow hate-speech detection methods, mediated through the three most used datasets. They evaluated the performance of different deep learning models, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT), and compared them with traditional classifiers. They also analyzed the practical performance of these models, including detection accuracy, computational efficiency, capability in using pre-trained models, and domain generalization. A deep learning model for hate speech detection in tweets was proposed by Badjatiya et al [11]. They used a Convolutional Neural Network (CNN) to extract features from the tweets and a Long Short-Term Memory (LSTM) to classify them. Ayo et al. in [12] reviewed the state-of-the-art machine learning techniques for hate speech classification of Twitter data. They discussed various techniques, including supervised, unsupervised, and deep learning methods, and highlighted the challenges and future directions in this field.
Meanwhile, Pereira-Kohatsu et al. [13] developed a hate speech detection system for Twitter. They used a dataset of tweets and evaluated the performance of different classifiers, including Support Vector Machines (SVMs), Naive Bayes, and Random Forest.

Then, Watanabe et al. [14] proposed a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection on Twitter. They used a dataset of tweets and evaluated the performance of different classifiers, including Support Vector Machines (SVMs), Naive Bayes, and Random Forest. Meanwhile, Zhang et al. in [15] proposed a deep neural network model for detecting hate speech on Twitter. They used a Convolutional Gated Recurrent Unit (CGRU) to extract features from the tweets and a softmax layer to classify them. They evaluated the performance of their model on a dataset of hate speech tweets and compared it with traditional classifiers. Finally, Wassem in [16] investigated the influence of annotators on hate speech detection on Twitter. They used a dataset of tweets and evaluated the performance of different classifiers, including Support Vector Machines (SVMs), Naive Bayes, and Random Forest.

## III. Methodology

In this section, we detail the methodology employed in our study to conduct a comparative analysis of machine learning algorithms on linguistic distinctions within textual data related to hate speech. The data used in this study were sourced from the Kaggle datasets containing relevant text samples [17][18] and also using around 9,000 rows of data pulled from Twitter API by the Orange application. To facilitate the execution of various operations, we leveraged the capabilities of the Orange data mining and machine learning toolkit. Orange provided us with a user-friendly environment for data pre-processing, feature extraction, and algorithm implementation, ensuring a systematic and efficient approach to our research. This section outlines the key steps involved in our methodology, including data collection and annotation, tokenization, data pre-processing, feature extraction, and the application of machine learning algorithms for classification.

### *A. Data Collection and Annotation*

The success of our study heavily relies on the quality and relevance of the data collected. To ensure a comprehensive and representative dataset, we utilized various sources, including official reports, NGO publications, news articles, and online forums that discuss topics related to hate speech. It's important to note that the data obtained were ethically sourced, respecting privacy and confidentiality concerns. Furthermore, each data point was meticulously annotated by domain experts to classify the text into categories relevant to hate speech. This step aimed to provide a solid foundation for the subsequent analysis, ensuring the accuracy and specificity of the dataset.

### *B. Tokenization and Data Pre-Processing*

Prior to analysis, the collected textual data underwent tokenization and thorough pre-processing. Tokenization involves breaking down the text into smaller, meaningful units such as words or phrases, allowing for a more granular analysis. Subsequently, the data underwent several pre-processing steps, including removing punctuation, stop words, and irrelevant symbols. Furthermore, the text was converted to lowercase to ensure consistency in the analysis, while also standardizing the data for uniformity across all entries. This phase aimed to refine the dataset for better compatibility with the machine learning algorithms, enabling more effective analysis of linguistic patterns.

### *C. Feature Extraction*

Feature extraction is a crucial step in the process of analyzing text data. The main objective was to transform the pre-processed text into a format suitable for machine learning algorithms. In this study, we employed various methods for feature extraction, including techniques like Bag-of-Words (BoW), TF-IDF (Term Frequency-Inverse Document Frequency), and word embeddings such as Word2Vec or GloVe. These methods were chosen for their ability to capture the essential linguistic nuances and patterns present in the textual data related to hate speech. The extracted features served as inputs for the subsequent classification phase.

*D. Classification*

The heart of our study lies in the comparative analysis of several machine learning algorithms for text classification in the context of hate speech.

*1) Naive Bayes*

Naive Bayes stands as a straightforward yet efficient machine learning algorithm utilized in classification tasks. Its foundation rests upon Bayes' theorem, a mathematical principle for calculating the likelihood of a particular event occurring based on the probabilities of related events. In the context of Naive Bayes, a "naive" assumption is made: it assumes that the features (attributes) used to describe the data are mutually independent, simplifying the computational process.

The algorithm's operation commences with the calculation of prior probabilities for each class, derived from the training data. Subsequently, when presented with a new data point, it computes the likelihood of observing the features within the context of a particular class. These probabilities are then multiplied to derive the "posterior probability" for each class. The class exhibiting the highest posterior probability is assigned as the predicted class for the new data point. Naive Bayes excels when handling high-dimensional data and can perform satisfactorily even with limited training data, making it a favored choice for tasks like text classification and spam detection.

*2) Random Forest*

Random Forest stands out as a versatile and robust ensemble learning algorithm extensively applied in both classification and regression tasks. Its operation revolves around the creation of multiple decision trees during the training phase, with their outputs aggregated to make predictions. Each decision tree is constructed using a random subset of the training data and a random subset of the available features, effectively mitigating overfitting and enhancing the algorithm's ability to generalize.

In the training process, bootstrap aggregating is employed for the construction of each tree within the forest. This technique entails drawing random samples with replacements from the original training data, leading to diverse training sets for each tree. When it comes to making predictions for new data, each tree in the forest independently generates an outcome. The final prediction is then determined by either majority voting (in the case of classification) or averaging (for regression) of these individual tree predictions. Random Forest excels in handling noisy data, capturing intricate data relationships, and avoiding overfitting.

*3) Decision Trees*

Decision Trees represent fundamental machine learning algorithms applicable to both classification and regression tasks. Their functionality revolves around dividing the input space into distinct regions, with each region corresponding to a specific class (in classification) or a predicted value (in regression). The creation of a decision tree involves a recursive process of splitting the data based on various feature values, with the objective of optimizing class separation or minimizing variance in the context of regression.

Commencing from the root node, the algorithm identifies the feature that most effectively divides the data into separate subsets. This process iterates for each subset, generating branches that lead to subsequent nodes. The criteria for splitting can differ and are influenced by metrics such as Gini impurity (for classification) or mean squared error (for regression). Decision Trees offer interpretability and have the capacity to capture intricate data relationships. However, they are susceptible to overfitting, especially when they become excessively deep. To address these limitations and enhance the predictive capabilities of decision trees, techniques like pruning and ensemble methods like Random Forest and Gradient Boosting have been developed.

*4) Gradient Boosting*

Gradient Boosting represents a potent machine learning algorithm aimed at enhancing predictive accuracy by progressively amalgamating the predictions of weak learners, often in the form of decision trees. In contrast to Random Forest, which constructs multiple trees independently, Gradient Boosting builds trees sequentially with a primary emphasis on rectifying the errors committed by preceding trees.

The algorithm commences by creating an initial model, which is typically uncomplicated. Subsequent models are then developed with the objective of diminishing the residual errors from the previous models. In each iteration, a new decision tree is tailored to address the negative gradient of the loss function in relation to the predictions generated by the current ensemble.

## IV. Results and Evaluation

In the study, an exhaustive evaluation of model efficacy was undertaken using the Orange data mining software as the analytical instrument. This facilitated an in-depth examination of various algorithms, yielding considerable insights into their operational effectiveness.

The assessment focused on the deployment of machine learning algorithms to differentiate between two distinct classes: those correlated with human speech (positive) and those that were not (negative). The intricacies of this binary classification were elucidated through the adoption of metrics such as the F1 score, Precision, and Recall.

These metrics provided a detailed evaluation of the models. The F1 score offered an integrative measure of the model's predictive capabilities, whereas Precision and Recall delivered a targeted understanding of their accuracy and their proficiency in identifying relevant instances. Through this methodical assessment, a deep appreciation of each algorithm's strengths and weaknesses was acquired, significantly contributing to the development of detection systems aimed at addressing hate speech.

Preliminary experiments were predominantly focused on the Naïve Bayes algorithm, with results scrupulously recorded in Table I. For the sake of precision, all values were rounded

to three decimal points. Among the four algorithms monitored, Randow Forest showed the optimal results

Subsequent experiments embraced the Random Forest model, varying the number of trees to 5, 7, 10, and 12. The findings of these tests, summarized in Table II and rounded to three decimal places for precision, indicated that the suboptimal performance within the four algorithms tested was achieved by Random Forest with 12 trees, achieving a notably high F1 score of 0.834, along with a precision of 0.834 and a recall of 0.835.

TABLE I. MODEL PERFORMANCE METRICS FOR NAÏVE BAYES

| AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|
| 0..91 | 0.783 | 0.783 | 0.827 | 0.783 |

TABLE II. MODEL PERFORMANCE METRICS FOR RANDOM FOREST

| Number of Trees | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|
| 5 | 0.885 | 0.812 | 0.811 | 0.811 | 0.812 |
| 7 | 0.897 | 0.824 | 0.823 | 0.823 | 0.824 |
| 10 | 0.903 | 0.834 | 0.833 | 0.834 | 0.834 |
| 12 | 0.906 | 0.835 | 0.834 | 0.834 | 0.835 |

The investigation into gradient boosting models uncovered a discernible inconsistency in F1 Scores across various samples, indicative of the model's complexity and challenge in achieving predictive stability. This inconsistency is particularly notable when the model parameters are adjusted, such as varying the number of trees at 80, 100, and 120, along with learning rates of 0.05, 0.1, and 0.15.

TABLE III. MODEL PERFORMANCE METRICS FOR GRADIENT BOOSTING

| Number of trees | Learning Rate | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 80 | 0.05 | 0.812 | 0.749 | 0.729 | 0.782 | 0.749 |
| | 0.1 | 0.852 | 0.776 | 0.763 | 0.800 | 0.776 |
| | 0.15 | 0.869 | 0.791 | 0.781 | 0.801 | 0.791 |
| 100 | 0.05 | 0.825 | 0.758 | 0.740 | 0.788 | 0.58 |
| | 0.1 | 0.862 | 0.785 | 0.774 | 0.806 | 0.785 |
| | 0.15 | 0.878 | 0.799 | 0.791 | 0.814 | 0.799 |
| 120 | 0.05 | 0.836 | 0.763 | 0.747 | 0.792 | 0.763 |
| | 0.1 | 0.87 | 0.792 | 0.782 | 0.811 | 0.792 |
| | 0.15 | 0.884 | 0.807 | 0.800 | 0.819 | 0.807 |

In the context of this research, it is pertinent to acknowledge the stratification in algorithmic efficacy. As delineated in Table IV, the Tree algorithm exhibited moderate performance, lagging in all three evaluative metrics – F1 score, precision, and recall. These results highlight its comparative limitations in this specific evaluative framework.

TABLE IV. MODEL PERFORMANCE METRICS FOR TREE ALGORITHM

| Min Number of Instances | Maximal Tree Depth | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 2 | 75 | 0.84 | 0.806 | 0.802 | 0.809 | 0.806 |
| | 100 | 0.845 | 0.807 | 0.804 | 0.807 | 0.807 |

## V. CONCLUSION

This study offers a comprehensive analysis of various machine learning algorithms' effectiveness in the detection of hate speech. Our findings demonstrate a clear variation in performance, with certain algorithms, such as Random Forest with specific parameter settings, outperforming others in key metrics like F1 score, Precision, and Recall. These insights underscore the importance of meticulous parameter tuning and algorithm selection in enhancing hate speech detection systems. While no single model consistently excelled across all metrics, the Random Forest algorithm, with an optimized number of trees, emerged as a notably robust candidate, suggesting a promising direction for future research and application.

Furthermore, the research delineates the complexity of distinguishing hate speech using automated systems, highlighting the challenges inherent in such tasks. It reinforces the need for ongoing development and refinement of machine learning models to address the dynamically changing landscape of online communication. The stratification of algorithmic performance indicates that while progress has been made, there is still considerable room for improvement.

## REFERENCES


[1] United Nations, "What is Hate Speech?". https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech (accessed Oct. 10, 2023).

[2] IDI–Yad Vashem. Recommendations for Reducing Online Hate Speech. https://www.idi.org.il/media/13570/recommendations-for-reducing-online-hate-speech.pdf (accessed Oct. 10, 2023).

[3] CBS News. Twitter and other social media sites slipped on removing hate speech in 2022, EU review says. https://www.cbsnews.com/news/twitter-other-social-media-slip-on-removing-hate-speech-european-union-review/ (accessed Oct. 10, 2023).

[4] Pereira-Kohatsu JC, Quijano-Sánchez L, Liberatore F, Camacho-Collados M. Detecting and Monitoring Hate Speech in Twitter. Sensors. 2019; 19(21):4654.

[5] Mullah, N. S., & Zainon, W. M. N. W. (2021). Advances in machine learning algorithms for hate speech detection in social media: a review. IEEE Access, 9, 88364-88376.

[6] Abro, S., Shaikh, S., Khand, Z. H., Zafar, A., Khan, S., & Mujtaba, G. (2020). Automatic hate speech detection using machine learning: A comparative study. International Journal of Advanced Computer Science and Applications, 11(8).

[7] Raufi, B., & Xhaferri, I. (2018, September). Application of machine learning techniques for hate speech detection in mobile applications. In 2018 International Conference on Information Technologies (InfoTech) (pp. 1-4). IEEE.

[8] Del Vigna12, F., Cimino23, A., Dell'Orletta, F., Petrocchi, M., & Tesconi, M. (2017, January). Hate me, hate me not: Hate speech

detection on facebook. In Proceedings of the first Italian conference on cybersecurity (ITASEC17) (pp. 86-95).

[9] Alkomah, F., & Ma, X. (2022). A literature review of textual hate speech detection methods and datasets. Information, 13(6), 273.

[10] Malik, J. S., Pang, G., & Hengel, A. V. D. (2022). Deep learning for hate speech detection: a comparative study. arXiv preprint arXiv:2202.09517.

[11] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017, April). Deep learning for hate speech detection in tweets. In Proceedings of the 26th international conference on World Wide Web companion (pp. 759-760).

[12] Ayo, F. E., Folorunso, O., Ibharalu, F. T., & Osinuga, I. A. (2020). Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. Computer Science Review, 38, 100311.

[13] Pereira-Kohatsu, J. C., Quijano-Sánchez, L., Liberatore, F., & Camacho-Collados, M. (2019). Detecting and monitoring hate speech in Twitter. Sensors, 19(21), 4654.

[14] 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.

[15] Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on twitter using a convolution-gru based deep neural network. In The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15 (pp. 745-760). Springer International Publishing.

[16] Waseem, Z. (2016, November). 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).

[17] Andrii Samoshyn. Hate Speech and Offensive Language Dataset. Kaggle. https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset (acccesed Jul, 11, 2023)

[18] Ali Toosi. “Twitter Sentiment Analysis”. Kaggle. https://www.kaggle.com/datasets/arkhoshghalb/twitter-sentiment-analysis-hatred-speec (accessed Jul 11, 2023)