Deep Learning Approaches for Hate Speech Detection

Ravi Tomar[1], Shaurya Pratap Singh[2], Siddhartha Verma[3], Amar Pal Yadav[4]

[1,2,3,4]NIET, Greater Noida, Uttar Pradesh, India

Abstract:
The main goals of this project are to develop reliable hate speech identification models that can recognise derogatory terminology, prejudiced attitudes, and damaging stereotypes on a variety of internet platforms. To help the models train and generalize efficiently, the study focuses on using big datasets with a variety of content types, including both hate speech and non-hate speech. The findings of this study suggest that machine learning has the potential to mitigate the negative consequences of hate speech by using automated filtering and flagging tools. The study also highlights the need for continued research and development to improve the accuracy, uniformity, and transparency of hate speech detection systems, and ultimately to foster a safer online environment to encourage all people.

Keywords: Hate Speech, Machine Learning, Offensive Language, Deep Learning neural networks etc.

1 Introduction

Hate speech using machine learning, Promoting inclusion and safety, in a digital world. In today's connected society, online communication has become a part of our lives. Social media platforms, forums, chat apps and other digital spaces provide opportunities for individuals to express themselves, interact with others and discuss issues but these open forums also create a concern, the proliferation of hate speech. Hate speech includes discriminatory or harmful content that targets individuals or groups based on factors such as race, religion, gender, and ethnicity.[27] This presents the challenge of creating a respectful digital environment. To address the growing issue of hate speech on the Internet, researchers and developers have turned to the power of machine learning and natural language processing (NLP) The field of detecting hate speech using machine learning is an upcoming project that intersects AI, NLP, and code. It aims to use various techniques to identify and categorize statements or sentences containing elements of hate speech. By doing so, the aim is to promote an online experience for all users[28]. This is basically training machine learning models to distinguish between hate speech and non-hate speech. The first step is to collect data sets that contain examples of hate speech and contain no hateful or negative speech. These datasets form the basis for training models to identify patterns, speech cues, and contexts indicative of hate speech[29].

The rapid pace of technological progress underscores the need for inclusive digital environments. Trying to prevent hate speech by using machine learning is an active step toward creating a diverse and respectful online environment. We want to make sure that online spaces are a place where people can connect with each other positively. Building systems that can automatically spot and eliminate hate speech[30]. Despite these different definitions, some recent studies claimed favorable results to detect automatic hate speech in the text [21-24]. According to some proposed ways, they used machine learning algorithms along with many other techniques to determine if something is hate speech or not. No matter how much effort we put in, comparing methods for hate speech classification remains a challenge. As far I know, nobody has looked into how different feature engineering techniques fare with machine learning algorithms.

2 Literature Survey

S.No.	Author	Technology Used	Description	Limitation
1	Mujtaba, G(2018)[6]	Feature Extraction	It is mapping from text data to real-valued vectors.	Bias in the training data, reliance on text-based information only.
2	MacAvaney (2019)[4]	Textual Feature Extraction	The study utilizes NLP, ML, DL, and lexicon-based methods for hate speech detection, addressing challenges and proposing solutions.	Limited exploration of multimodal data.
3	Sindhu Abro, Sarang Shaikh, Zafar Ali, Sajid Khan, Ghulam Mujtaba (2020)[26]	Supervised Learning	This study employs NLP, ML, and advanced feature extraction techniques for automated hate speech detection using supervised learning.	Lack of exploration into the robustness of the model across diverse datasets.
4	Gupta, K. K., Vijay, R., & Pahadiya, P. (2020)[33]	Word2vec	It is a technique used to learn vector representation of words, which can further be used to train machine learning models.	The inability to handle polysemy effectively, the requirement for large amounts of training data
5	Pahadiya, P., Vijay, D. R., kumar Gupta, K., Saxena, S., & Tandon, R (2020)[35]	Doc2vec	It is an unsupervised technique to learn document representations in fixed-length vectors. It is the same as word2vec, but the only difference is that it is unique among all documents.	The need for substantial training data, difficulty in handling variable-length documents
6	Gupta, K. K., Vijay, R., Pahadiya, P., & Saxena, S (2022)[29]	Machine Learning Classifiers	These are applied to numeric features vector to build the predictive model which can be used for prediction class labels.	Being a review paper without original research
7	Gupta, K. K., Rituvijay, Pahadiya, P., & Saxena, S (2022)[30]	Naïve Bayes	It's a probabilistic based classification algorithm, which uses the "Bayes theorem" to predict the class. It works on conditional independence among features.	The oversimplified independence assumption, which may not hold in real-world data

8	Gupta, K. K., Vijay, R., & Pahadiya, P (2022)[31]	Random Forest	It's a type of ensemble classifier consisting of many decision trees. It classifies an instance based on voting decision of each decision trees class predictions.	The lack of a comprehensive evaluation across various datasets and text domains
9	Pahadiya, P., Vijay, R., Gupta, K. K., Saxena, S., & Tandon, R (2022)[34]	Support Vector Machines	It's a supervised classification algorithm which constructs an optimal hyperplane by learning from training data which separates the categories while classifying new data.	The requirement for proper feature engineering, sensitivity to hyperparameter tuning, potential scalability issues with large datasets
10	Gupta, K. K., Vijay, R., Pahadiya, P., Saxena, S., & Gupta, M (2023)[27]	K Nearest Neighbor	It's a simple text classification algorithm, which categorize the new data using some similarity measure by comparing it with all available data.	The choice of the k parameter, potential scalability issues with large datasets
11	Pahadiya, P., Vijay, R., Gupta, K. K., Saxena, S., & Shahapurkar, T (2023)[28]	Decision Tree	It is a supervised algorithm. It generates the classification rules in the tree-shaped form, where each internal node denotes attribute conditions, each branch denotes conditions for outcome and leaf node represents the class label.	Dealing with evolving drug names, domain-specific terminology
12	Saxena, S., Vijay, R., Pahadiya, P., & Gupta, K. K (2023)[32]	Adaptive Boosting	It is one of the best-boosting algorithms, which strengthens the weak learning algorithms.	Sensitivity to noisy data and outliers, potential overfitting when weak classifiers are too complex

3 Methodology

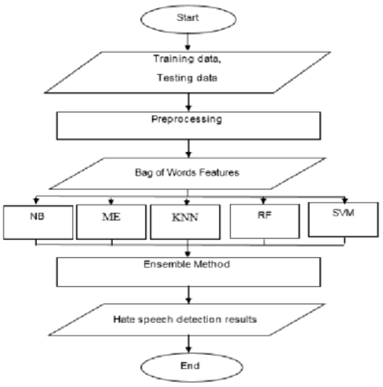

Fig.1: Flow Chart

The proposed system for dividing tweets into three categories—"hate speech, anger but not hate speech, and hate speech or anger speech"- is described in this section, data preprocessing, feature engineering, data segmentation, classification model development and evaluation The learning method consists of six main phases as shown in this figure. The following sections go into much more detail about each category.

Data Collection and Preprocessing: Data Sourcing: Collect diverse datasets containing examples of hate speech and non-hateful content from various online platforms and sources. Data Annotation: Annotate the collected data to ensure accuracy in labeling hate speech instances. Data Preprocessing: Clean, tokenize, and preprocess text data, including techniques such as stemming, lemmatization, and removing stop words. Balancing Data: Ensure a balanced dataset to prevent bias toward the majority class (non-hateful content).

Feature Engineering: Text Representation: Convert text data into numerical format using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embedding (e.g., Word2Vec, GloVe).

Feature Selection: Identify relevant features that contribute to hate speech detection and reduce dimensionality if needed.[31]

Model Development: Select Architecture: Choose appropriate machine learning models, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), or transformer-based models like BERT, depending on the nature of the data.

Training: Train the selected models on the preprocessed and feature-engineered data using suitable loss functions and optimization techniques.

Hyper Parameter Tuning: Optimize model hyper parameters through techniques like grid search or random search to improve performance.

Cross-validation: Implement cross-validation to assess the model's generalization and robustness.

Model Evaluation and Testing:

Evaluation Metrics: Assess model performance using metrics like accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC)[32].

Validation Set: Set aside a portion of the data for validation and fine-tuning model parameters.

Testing: Evaluate the model on an independent test set to measure its real-world performance.

Integration and Deployment:

API Development: Develop an API or integration module to enable seamless integration with digital platforms and services.

Continuous Monitoring: Implement monitoring mechanisms to track model performance in real-time and generate alerts for potential issues.

Feedback Loop: Establish a feedback loop to collect user feedback and continuously improve the model.

Customization for Cultural Contexts:

Data Augmentation: Augment datasets with content specific to different cultural contexts to ensure the model's effectiveness in diverse settings[33].

Custom Model Variants: Develop model variants customized for specific cultural or linguistic regions.

Education and Awareness:

Content Creation: Develop educational materials, such as guidelines, videos, or infographics, to raise awareness about the impact of hate speech and responsible online behavior.

User Engagement: Promote user engagement with these materials through online campaigns and community outreach.

Ethical Considerations:

Fairness: Ensure that the hate speech detection model is fair and unbiased, minimizing the risk of discrimination against specific groups.

Privacy: Prioritize user privacy by implementing data protection measures, especially if user-generated content is involved.

Transparency: Maintain transparency in the model's decision-making process by providing explanations for its classifications.

Continuous Improvement:

Monitoring Trends**:** Stay updated on emerging forms of hate speech and adapt the model accordingly[34].

Re-training: Periodically retrain the model with fresh data to maintain its effectiveness over time.

Reporting and Documentation:

Maintain detailed documentation of the entire project, including data sources, preprocessing steps, model architecture, hyper parameters, and evaluation results.

Regularly communicate project progress and findings to stakeholders and the broader community.. This comprehensive methodology forms the backbone of our project, ensuring a systematic approach to developing and deploying machine learning solutions for hate speech detection. Through meticulous planning and execution, we aim to contribute to a digital world where technology serves as a force for positive change, fostering respectful and inclusive online environments[35].

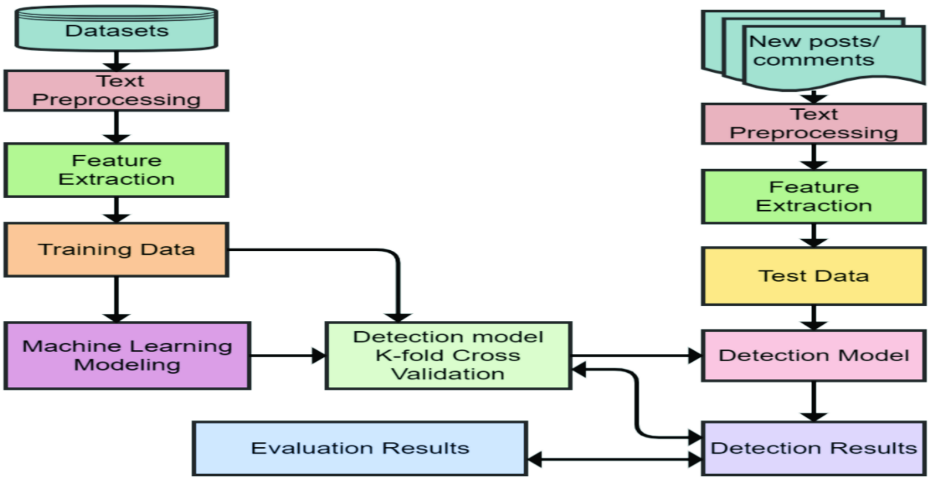

Fig.2: Data Preprocessing

4. Result and Discussion

Precision, recall, F-measure, and precision for all 24 analyzes are shown in Tables III through Table VI, respectively. The maximum values of the results are shown in bold. The performance of the different feature representation and classification algorithms used in the experimental scenarios is shown in all tables. There is absolute accuracy in all 24 studies (0.47), F-measure (0.46), recall (0.57), precision (57%). found in the TFIDF products used in MLP and KNN classifiers Bigram-based representation of attributes. Besides, the highest quality F-measure, recall (0.79), precision (0.77), and precision (79%). SVM was used to obtain (0.77) using TFIDF features. Bigram-based representation of attributes. When specifying the feature. TFIDF and bigram characterization gave good results e.g. Unlike Doc2vec and Word2vec.The dataset used for training and testing contains text data (tweets) labeled with two classes: hate speech (label 1) and non-hate speech (label 0).

After training and testing the machine learning model, the code provides several result:

Table . Precision of all analysis

Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.72	0.66	0.76	0.61	0.71	0.75	0.74	0.63
Word2vec	0.69	0.67	0.73	0.64	0.62	0.65	0.65	0.65
Doc2vec	0.70	0.69	0.69	0.69	0.61	0.66	0.66	0.75

Table . Recall of all analysis

Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.71	0.71	0.74	0.77	0.58	0.78	0.74	0.74
Word2vec	0.70	0.66	0.64	0.74	0.65	0.63	0.61	0.78
Doc2vec	0.74	0.64	0.68	0.75	0.61	0.66	0.65	0.78

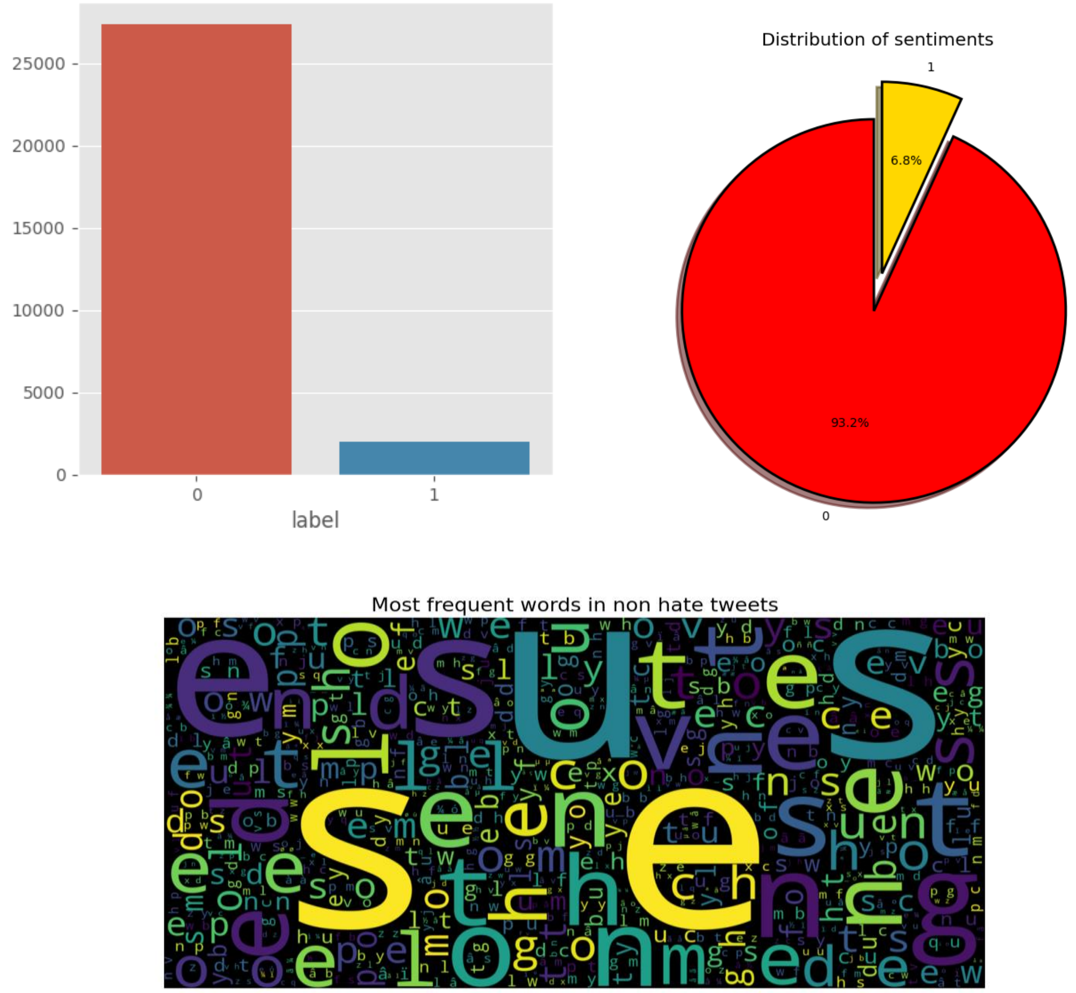

Fig.4: Data Label chart

5. Conclusion

One practical way to combat online hate speech and promote a safer and more welcoming digital environment is through machine learning-based hate speech detection It uses algorithmic models to identify and categorize content it is in anger or injury, hatred. While it holds promise for helping online communities, social media platforms, and content monitors take proactive measures to combat speech, there are issues with bias, context, and hostility of developing language. Continued learning, development and collaboration between technologists, ethicists and politicians is critical to improving these policies, reducing bias, and balancing freedom of expression and online information between protecting dangerous[22]. An algorithm was used to identify speech in these texts, study finds. Eight machine learning algorithms were compared alongside three techniques for feature engineering to classify hate speech texts. According to the study, TFIDF bigrams performed better than word2Vec and doc2Vec features engineering methods in terms of results. To sum up, SVM and RF surpassed LR, NB, KNN, DT, AdaBoost and MLP in regard to perform metrics. KNN showed the poorest results. What this research shows is that there is value in using different algorithms to detect hate speech in texts.

References

1. Hern, A., Facebook, YouTube, Twitter, and Microsoft sign the EU hate speech code. The Guardian, 2016. 31.
2. Rosa, J., and Y. Bonilla, Deprovincializing Trump, decolonizing diversity, and unsettling anthropology. American Ethnologist, 2017. 44(2): p. 201-208.
3. Travis, A., Anti-Muslim hate crime surges after Manchester and London Bridge attacks. The Guardian, 2017.
4. MacAvaney, S., et al., Hate speech detection: Challenges and solutions. PloS one, 2019. 14(8): p. e0221152.
5. Fortuna, P. and S. Nunes, A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 2018. 51(4): p. 85.
6. Mujtaba, G., et al., Prediction of cause of death from forensic autopsy reports using text classification techniques: A comparative study. Journal of forensic and legal medicine, 2018. 57: p. 41-50.
7. Cavnar, W.B. and J.M. Trenkle. N-gram-based text categorization. in Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval. 1994. Citeseer.
8. Ramos, J. Using tf-idf to determine word relevance in document queries. in Proceedings of the first instructional conference on machine learning. 2003. Piscataway, NJ.
9. Mikolov, T., et al. Distributed representations of words and phrases and their compositionality. in Advances in neural information processing systems. 2013.
10. Le, Q. and T. Mikolov. Distributed representations of sentences and documents. in International conference on machine learning. 2014.
11. Kotsiantis, S.B., I.D. Zaharakis, and P.E. Pintelas, Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 2006. 26(3): p. 159-190.
12. Lewis, D.D. Naive (Bayes) at forty: The independence assumption in information retrieval. in European conference on machine learning. 1998. Springer.
13. Xu, B., et al., An Improved Random Forest Classifier for Text Categorization. JCP, 2012. 7(12): p. 2913-2920.
14. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. in European conference on machine learning. 1998. Springer.

15. Zhang, M.-L. and Z.-H. Zhou, A k-nearest neighbor based algorithm for multi-label classification. GrC, 2005. 5: p. 718-721.
16. Abacha, A.B., et al., Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug–drug interaction extraction and classification. Journal of biomedical informatics, 2015. 58: p. 122- 132.
17. Ying, C., et al., Advance and prospects of AdaBoost algorithm. Acta Automatica Sinica, 2013. 39(6): p. 745-758.
18. Gardner, M.W. and S. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 1998. 32(14-15): p. 2627-2636.
19. Wenando, F.A., T.B. Adji, and I. Ardiyanto, Text classification to detect student level of understanding in prior knowledge activation process. Advanced Science Letters, 2017. 23(3): p. 2285-2287.
20. Burnap, P. and M.L. Williams, Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Science, 2016. 5(1): p. 11.
21. Gitari, N.D., et al., A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering, 2015. 10(4): p. 215-230.
22. Tulkens, S., et al., A dictionary-based approach to racism detection in dutch social media. arXiv preprint arXiv:1608.08738, 2016.
23. Greevy, E. and A.F. Smeaton. Classifying racist texts using a support vector machine. in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 2004. ACM.
24. Kwok, I. and Y. Wang. Locate the hate: Detecting tweets against blacks. in Twenty-seventh AAAI conference on artificial intelligence. 2013.
25. Sharma, S., S. Agrawal, and M. Shrivastava, Degree based classification of harmful speech using twitter data.
26. Sindhu Abro, Sarang Shaikh, Zafar Ali, Sajid Khan, Ghulam Mujtaba, Automate Hate Speech Detection using Machine Learning. International Journal of Advanced Computer Science and Applications, Vol. 11, No. 8, 2020.
27. Gupta, K. K., Vijay, R., Pahadiya, P., Saxena, S., & Gupta, M. (2023). Novel Feature Selection Using Machine Learning Algorithm for Breast Cancer Screening of Thermography Images. Wireless Personal Communications, 1-28.
28. Pahadiya, P., Vijay, R., Gupta, K. K., Saxena, S., & Shahapurkar, T. (2023). Digital Image Based Segmentation and Classification of Tongue Cancer Using CNN. Wireless Personal Communications, 1-19.
29. Gupta, K. K., Vijay, R., Pahadiya, P., & Saxena, S. (2022). Use of novel thermography features of extraction and different artificial neural network algorithms in breast cancer screening. Wireless Personal Communications, 1-30.
30. Gupta, K. K., Rituvijay, Pahadiya, P., & Saxena, S. (2022). Detection of cancer in breast thermograms using mathematical threshold based segmentation and morphology technique. International Journal of System Assurance Engineering and Management, 1-8.
31. Gupta, K. K., Vijay, R., & Pahadiya, P. (2022). Detection of abnormality in breast thermograms using Canny edge detection algorithm for thermography images. International Journal of Medical Engineering and Informatics, 14(1), 31-42.

32. Saxena, S., Vijay, R., Pahadiya, P., & Gupta, K. K. (2023). Classification of ECG arrhythmia using significant wavelet-based input features. International Journal of Medical Engineering and Informatics, 15(1), 23-32.
33. Gupta, K. K., Vijay, R., & Pahadiya, P. (2020). A review paper on feature selection techniques and artificial neural networks architectures used in thermography for early stage detection of breast cancer. Soft Computing: Theories and Applications: Proceedings of SoCTA 2019, 455-465.
34. Pahadiya, P., Vijay, R., Gupta, K. K., Saxena, S., & Tandon, R. (2022). Contactless non-invasive method to identify abnormal tongue area using K-mean and problem identification in COVID-19 scenario. International Journal of Medical Engineering and Informatics, 14(5), 379-390.
35. Pahadiya, P., Vijay, D. R., kumar Gupta, K., Saxena, S., & Tandon, R. (2020). A Novel method to get proper tongue image acquisition and thresholding for getting area of interest. International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN, 2278-3075.