Detecting Hate Speech and Offensive Language: Evaluating Multiple Machine Learning Approaches on a Common Dataset

Ksh. Krishna B. Singha
Trinity Institute of Professional Studies,
New Delhi, India

Arti Bajaj
Trinity Institute of Professional Studies,
New Delhi, India

ABSTRACT

Social media platforms are readily exploited nowadays to propagate hate and offensive speeches that may be directed towards an individual, an organization, a particular society or societies, a country, and so on. These messages can sometimes lead to very horrific consequences, which this civilization witnessed in the recent past. To avoid such a scenario, it is very crucial to control the spread of such content in a timely manner. The task of identifying and filtering out these contents is very essential prior to being made available on the social media platforms. Machine learning algorithms have proved to be one of the popular techniques used for this task. This work presents the performance of three machine learning algorithms—Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), and Naïve Bayes (NB)—and an ensemble learning method, Random Forest (RF), on a dataset of X(formerly Twitter) hate speech text data. We see the detection of hate speech in a tweet as a classification problem—hate and non-hate class. The dataset has been resampled to balance the data in the two classes after cleaning the text using various natural language processing techniques. Suitable feature engineering techniques are used to extract and select important features for the classification purpose. For each of the learning techniques, we evaluated the performance on the feature set. The SVM technique gave the highest F1 score of 98%, whereas the NB technique performed the lowest F1 score of 92%.

General Terms

Classification task, Machine Learning Algorithms.

Keywords

Hate Speech, Offensive Speech, Hate Speech Classification, Social Media, Support Vector Machine, Machine Learning, Random Forest, Ensemble Learning, Stochastic Gradient Descent, Naïve Bayes.

1. INTRODUCTION

People use social media for exchanging information, and the facility provided by today's technology is easily exploited in the name of freedom of expression by its users. Hate speech content on media platforms such as X(formerly Twitter) is a menace these days, sometimes leading to violence and riots or social disorders.

Though there are no specific definitions of hate speech, according to Poletto F, et al. Hate Speech (HS) [1], lying at the intersection of multiple tensions as expression of conflicts between different groups within and across societies, is a phenomenon that can easily proliferate on social media. Davidson, T. et al. [2] defines hate speech as a language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group. Hate speech content may, however, vary from one context to another. Another aspect is that hate speech may not include all instances of offensive language, as they are based on race, ethnicity, gender, and sexual orientation, specific community, religion, beliefs, ideology, and likewise.

Detecting hate content in speech is a challenge, as content considered as hate speech may not be one in some other context. So, a solution to solving this problem for a certain set of people and community may not be suitable for some and hence cannot be generalized to the solution at all.

This paper presents the performance of four machine learning algorithms on a hate speech dataset, which is available here. The dataset has three labels or classes to detect a tweet as hate_speech, offensive_language and neither, but the former two classes have been combined as one and label it as hate speech and consider the problem as a binary classification problem. Also, resampling is done to balance the data in the two classes. The machine learning algorithms, viz. Random Forest, Support Vector Machine, Stochastic Gradient Descent and Naïve Bayes are applied separately on the common dataset to train on the same set of features and develop a model for each algorithm for detecting hate speech. The tfidfvectorizer and one-hot encoding are used for feature engineering purposes.

The precision, recall, and F1 score for each model are analyzed. The prediction results are promising, and the SVM outperforms the other three in terms of F1 score, attaining a whopping 98%. The work is concluded after a proper analysis and examination of the result that fine-grained labelling is crucial, and the text contraction in the text pre-processing stage could play a vital role in feature engineering. The paper is organized as follows: the first section introduces the topic and its significance; the second section presents the related work available in the literature. The third section is about the motivation behind this research effort, followed by the fourth section which describes the methodology adopted in the study. The fifth section covers the proposed work which is comprised of the sub sections on Data Preprocessing, experimental set up with analysis. These sections describe the stages for cleaning the text data, resampling the imbalanced data in the dataset, brief idea about the chosen algorithms performance, followed by experimental results and analysis, and finally the conclusion and future scope for research.

2. RELATED WORKS

In recent years, the research on detecting hate speech and/or offensive language content on social media platforms has been becoming a trend, the reason being its obvious requirement to filter out such content before being made available to its audience and readers. Different researchers in the area put forward their works and are available in literature in abundance. The popular approaches to detect and classify these contents are machine learning based algorithms. Most of these works vary from one another in terms of feature engineering and classification algorithms and certain other factors, such as choice of hyperparameters, etc.

Aljero, M. K. A. et al. [3] proposed an approach for detecting hate speech in English tweets employing an ensemble of three classifiers—support vector machine (SVM), logistic regression (LR), and XGBoost classifier (XGB), which are trained using word2vec and universal encoding features. It claimed to improve the performance of the widely used single classifiers as well as the standard stacking and classifier ensemble using majority voting outperforming all state-of-the-art systems.

A classifier model employing multiple deep models is implemented by Mnassri, K., Rajapaksha et. al. [4], integrating transformer-based language models BERT and neural networks. The model is evaluated with several ensemble techniques such as soft voting, maximum value, hard voting and stacking using three publicly available X(formerly Twitter) datasets (Davidson, HatEval2019, OLID). Their stacking ensemble models performed better, giving an F1 score of 97% on the Davidson dataset.

Hegde, A. et al. in [5] proposed models using an ensemble classifier comprising random forest, multilayer perceptron, and gradient boosting to classify hate speech expressed in Indo-Aryan languages. Divided into subtasks 1A and 1B for English, Hindi and Marathi languages and Subtask 2 for code-mixed text in English-Hindi language pair, they trained using TF-IDF of different features like word uni-gram, character n-grams, Hashtag vectors (HastagVec) followed by using the pre-trained embeddings: word2Vec and Emo2Vec; and claimed to achieve obtained 43rd, 23rd, 18th, 10th, and 15th rank for English, Hindi and Marathi Subtask 1A and Subtask 1B respectively.

In [6] Agarwal, S. et al. propose an ensemble learning-based adaptive model for automatic hate speech detection, improving the cross-dataset generalization and working towards overcoming the strong user bias present in the available annotated datasets. Tested using various experimental setups on issues like COVID-19 and US presidential elections, the loss in performance observed under cross-dataset evaluation is claimed to be the least among similar models.

Another ensemble learning-based classifier by Mutanga, R.T., et. al [7] presents a voting ensemble ML that harnesses the strengths of LR, DT, and SVM for automatic detection of hate speech in tweets and was evaluated against ten widely used ML methods on two standard tweet datasets, achieving an improved average F1 score of 94.2%.

Zimmerman, S., et al. [8] presented an ensemble method, adapted for usage with neural networks, which utilizes a publicly available embedding model and is tested against a hate speech corpus from X(formerly Twitter). The robustness of the results was claimed to be confirmed by testing against a popular sentiment dataset.

Yanling Zhou, et al. [9] presented a fusion model combining classifiers such as embeddings from Language Models (ELMo), Bidirectional Encoder Representation from Transformers (BERT) and Convolutional Neural Network (CNN), and apply these methods to the data sets of the SemEval 2019 Task improving the overall classification performance. The results claimed that accuracy and F1-score of the classification are significantly improved.

Extensive experiments were performed by Badjatiya, P., et al. [10] to learn semantic word embeddings with multiple deep learning architectures with a benchmark dataset of 16K annotated tweets; and the deep learning methods said to outperform state-of-the-art char/word n-gram methods by ~18 F1 points.

Another work by Ika Alfina, et al. [11] prepared a Indonesian language dataset covering hate speech against religion, race, ethnicity, and gender; compared the performance of several features and machine learning algorithms for hate speech detection on the dataset. Word n-grams with n=1 and n=2, character n-grams with n=3 and n=4, and negative sentiment were used for features and classified using Naïve Bayes, Support Vector Machine, Bayesian Logistic Regression, and Random Forest Decision Tree. Their work achieved an F-measure of 93.5% when using word n-gram feature with Random Forest Decision Tree algorithm and said to outperform character n-gram.

Asogwa, D. C., et al. [12] presented a machine learning model for hate speech classification using Support Vector Machine (SVM) and Naïve Bayes(NB). They claimed to achieve near state-of-the-art performance while being simpler and producing more easily interpretable decisions than other methods. The empirical evaluation of this technique has resulted an accuracy of approximately 99% and 50% for SVM and NB respectively over the test set.

In addition to the above works available in the literature, there are many more also that are overlapped in one way or another considered here. In [13], Burnap P., presented a supervised ML text classifier using a combination of probabilistic, rule-based, and spatial-based classifiers with a voted ensemble meta-classifier to train human-annotated X(formerly Twitter) data that distinguishes between hateful and/or antagonistic responses with a focus on race, ethnicity, or religion, and more general responses. Mohapatra, S. K., et al. [14] presents an ML-based hate speech detection model by utilizing text mining feature extraction techniques where they collected hate speech of English-Odia code-mixed data from a Facebook public page and manually organized them into three classes. Another deep learning-based hate speech classifier model was proposed by Gambäck, B., & Sikdar, U. K. [15]; four CNN models were trained on resp. character 4-grams, tested by 10-fold cross-validation, and the model based on word2vec embeddings is claimed to perform best, with higher precision than recall and a 78.3% F1 score. An offensive and hate speech detection for the Arabic language, based on two-class, three-class, and six-class Arabic-Twiter datasets are developed by Alsafari, S., et al. [16] using single and ensemble CNN and BiLSTM classifiers which is trained with non-contextual (Fasttext-SkipGram) and contextual (Multilingual Bert and AraBert) word-embedding models. A multi-label text classification by Ibrohim, M. O., & Budi, I. [17] is available in the literature for abusive language and hate speech detection of hate speech in Indonesian X(formerly Twitter). They used various machine learning approaches as the data transformation method and term frequency, orthography, and lexicon features as feature extraction techniques for the model. Their experiment results show that the RFDT classifier using LP as the transformation method gives the best accuracy with fast computational time. In [18], Gomez, R., Gibert, J., et al. propose different models that jointly analyze textual and visual information for hate speech detection, comparing them with unimodal detection. Even they found out that even though images are useful for the hate speech detection task, current multimodal models cannot outperform models analyzing only text.

3. MOTIVATION

It is important and interesting to note that a user can exploit the freedom of using their social media accounts for expressing anything that comes to their mind. It may, however, sometimes lead to very unpleasant and/or horrific scenarios because of the content in the expression. The outcome of such social media content can affect one directly and indirectly, and therefore the task of identifying and detecting such content should be done using the best feasible method. As there are many techniques available, we are motivated to do our part in combating these hazardous practices by experimenting on a dataset with some of the average and best performers of machine learning techniques available in the literature. Through this study we are able to analyze the experimental results and share the findings for further course of study in that direction.

4. METHODOLOGY

The problem of hate speech detection on social media is considered a classification problem. Three baseline algorithms, i.e., SVM, SGD, and NB, and one ensemble learning algorithm, ensemble learning (RF), have been employed to construct a text classifier for detecting hate content in a text corpus. The selection of algorithms is based on the performance of the algorithms on this particular type of task. RF is chosen for its accuracy, training speed in a parallel method, capacity to select the best features automatically, and error balancing capability in unbalanced datasets [19]. The feature selection is done using term frequency and inverse document frequency. The dataset considered for the present work is readily available online, and many have experimented on it [2]. For the experimental purpose, the Python machine learning library, viz. the Scikit-learn has been employed for the four machine learning algorithms, numpy for mathematical functions and array computations, pandas for data analysis and manipulation, and nltk for text pre-processing and cleaning.

5. PROPOSED WORK

This section explains the detailed steps of the proposed work, In the first stage, the unnecessary columns in the dataset is identified for removal; the remaining columns are then cleaned for further processing resampled to balance the dataset. For each of the four chosen algorithms, the pre processed and cleaned dataset is trained and tested. The following sub sections explain the details of the work for the four algorithms in detail.

5.1 Data Pre-Processing

Before using the data for feature extraction and presentation to the algorithm for training, it is required to make it acceptable by the ML algorithms. This step for preparation of data to make it eligible for processing by the ML algorithms is known as data preprocessing. The chosen dataset has many symbols and other unnecessary tokens as part of a tweet, which do not contribute to identifying tweets as hate or non-hate classes. This is done after converting all the words to lowercase letters. It involves the removal of noisy words. These words include-

- hash symbols,
- numbers,
- retweet symbols,
- X(formerly Twitter) mentions,
- punctuation marks,
- extra spaces,
- emotional symbols,
- URLs,
- connecting words and
- prepositions, etc.

The Porter algorithm has been employed for the removal of stop words and regular expressions for the removal of the noisy words. The steps followed for pre-processing and cleaning the data is shown in the figure below:

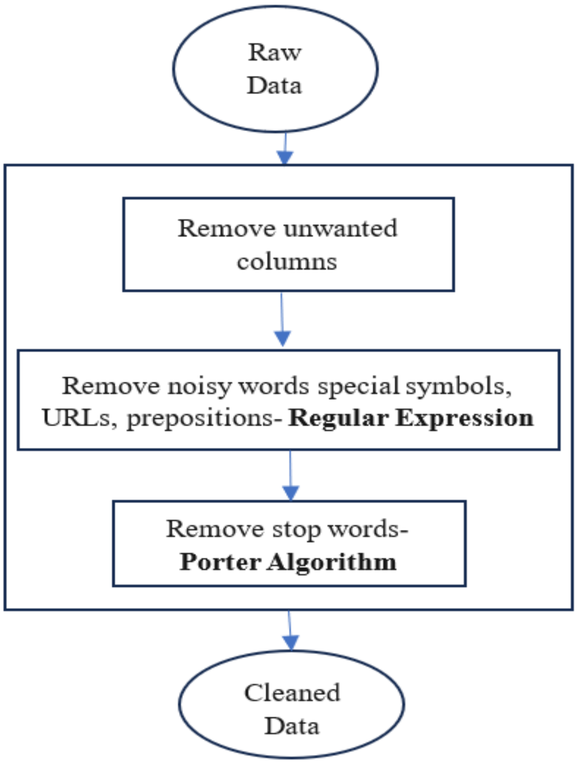
Raw Data
Remove unwanted columns
Remove noisy words special symbols, URLs, prepositions- **Regular Expression**
Remove stop words- **Porter Algorithm**
Cleaned Data

Figure 1: Data Pre-Processing and Cleaning

5.2 Experimental Set up and Analysis

The setup for the purpose of this study has been done in four similar phases for each of the experiments conducted using the four machine learning algorithms. The dataset [2] under consideration has seven columns: count, hate_speech, offensive_language, neither, class, and tweet. Here, the hate and offensive speech are considered as same for the classification task. So, the original dataset has been modified accordingly; tweets that are marked as hate_speech and offensive_language are considered as hate only, and thereby the tweets come under two classes: either hate or non_hate. Also, for experimental purpose, we are interested in only three columns of the dataset, which are id, class, and tweet column, The remaining columns are non-significant for the training and therefore dropped from the dataset. The column id represents the unique id of each tweet; the class column marks whether the tweet is a hate or a non-hate speech. Here a 0 value in the class column classifies the corresponding tweet in the hate class and a 1 in the non-hate class.

The tweet column has unwanted characters, symbols, and English stopwords which are of no purpose for the study and are removed by using the stopwords function from nltk library and porterstemmer algorithm (M.F. Porter, 1980) for stemming the suffix morphemes and by employing regular expression library. The result of this exercise is the cleaned data without any stopwords and symbols (Figure 1).

The dataset has 24783 rows, out of which 20620 are of non-hate and the remaining 4163 are of hate class. So, this imbalance in the category of tweets is resampled to make a balance in the two classes using the resample function from the sklearn library. After resampling, both hate and non-hate classes have been balanced to 20620 each.

The text data vectorization is done by using CountVectorizer from the Python scikit-learn library. CountVectorizer converts the text data into a matrix of token counts. TfidfTransformer transforms the count matrix into a TF-IDF representation. We used the TfidfVectorizer function for feature extraction. The number of features vocabulary is limited to the top 5000 words by frequency. In the case of SVM and NB models, LabelEncoder is used to convert the text labels into numerical form. This step is essential because the SVM and NB classifiers require numerical input. To assemble multiple steps into a single object, we use a pipeline for SGD and RF, which is a sequential processing chain. It ensures that the steps are executed in order, with the output of one step becoming the input to the next. Here we set the chi-squared scoring function to select the top 1200 (value of k) most relevant features. Feature selection can help improve model performance by reducing dimensionality and focusing on the most important features. Once this feature selection part is done, the four respective models are trained using the training set. The models are trained on 80% of the data while it is tested on the remaining 20%. The model learns the relationships between features and labels in the training subset of the whole data. The testing set is used to evaluate the performance of the trained model. The model's accuracy and other metrics are measured on this unseen data to assess how well it generalizes to new, unseen examples.

To evaluate the performance of the trained machine learning models on the test data, we employ accuracy_score, classification_report, and confusion_matrix from the sklearn metrics library. The accuracy of predictions of these models and its classification performance is calculated using these functions. A performance chart of the four algorithms is shown as below in the table:

Table 1: Classification Report and Accuracy Score

Model using Algorithm	Class	Precision	Recall	F1 Score	Accuracy Score
Naive Bayes (NB)	Non-hate (class-0)	0.91	0.95	0.93	0.9243
	Hate (class-1)	0.95	0.90	0.92	
Stochastic Gradient Descent (SGD)	Non-hate (class-0)	0.99	0.95	0.97	0.9671
	Hate (class-1)	0.95	0.99	0.97	
Support Vector Machine (SVM)	Non-hate (class-0)	0.98	0.95	0.97	0.9679
	Hate (class-1)	0.95	0.98	0.97	
Random Forest (RF)	Non-hate (class-0)	0.96	0.97	0.97	0.9419
	Hate (class-1)	0.84	0.80	0.82	

As could be seen from the above table, there is a competition between SGD and SVM models. The NB, SGD, and SVM exhibit a similar precision success rate in identifying class-1, which is 95%, where the RF model stays far behind them with a rate of only 84%. Considering the recall rate for class-1, the SGD proves to be the best performer with 99% recall rate while SVM comes close with 98% recall rate; here again the RF model could not be that reliable as compared with the other models, with its score of only 80%; the NB model score being in between these two extremes, as 90% recall rate. The best F1 score stands at 97% for class-1 prediction recorded by both SGD and SVM models. As for predicting class 1, both SGD and SVM are proved to be the best performers out of the chosen four models.

For prediction of class-0, we can see a change in pattern, that RF is performing at par with SGD and SVM in terms of their F1 score, which is 97% whereas that of NB is 93%. In terms of accuracy score, the top performers are also SGD and SVM.

From the above experiments and findings, the following can be illustrated for the four models:

The Naïve Bayes (NB) model has good balance between precision and recall for both the classes; and a good high overall accuracy; however, its weaknesses being the slight lower recall rate for hate (class-1), as compared to the other three models indicating some hate instances might be missed.

In the case of the Stochastic Gradient Descent (SGD) model, it can be said that there is an excellent balance between precision and recall for both classes and a very high overall accuracy. As the recall rate (0.99) indicates, the model is especially good at detecting hate (class-1). It has a slightly lower recall rate for the non-hate class (0.95) as compared to the RF model.

The strength of the Support Vector Machine (SVM) model here is the high precision and recall for both classes, resulting in high F1 scores and an overall accuracy. It exhibits a similar performance metric as that of SGD. The weakness of the model is not that significant, though; the recall rate for the non-hate class (0.95) could be slightly improved.

The Random Forest (RF) model performs well in identifying the non-hate class with high precision and recall. However, the RF model struggles more to correctly identify hate instances.

Of these, both **SGD** and **SVM** models come out to be the b**est performers,** showing the highest overall performance with balanced precision, recall, and F1 scores for both classes. They also have the highest accuracy scores (~0.967), making them the best choices for this classification task. **Naive Bayes** also performs well, with slightly lower recall for the hate class but still maintains high precision and overall accuracy. **Random Forest** performs well for the non-hate class but significantly underperforms for the hate class, making it less reliable for balanced classification tasks.

SGD and SVM are the suggested models if maximizing hate speech detection while preserving high overall accuracy is the aim. Naive Bayes provides an excellent trade-off between simplicity and performance if computing efficiency or a more straightforward implementation are top priorities.

6. CONCLUSION

The task of detecting hate speech (class-1) accurately is crucial due to its potential social impact. Therefore, using the F1 score would be more appropriate because it accounts for both false positives and false negatives, providing a balanced view of model performance in an imbalanced classification scenario. Given the importance of accurately identifying hate speech and the class imbalance, the F1 score is the more suitable metric for your evaluation. Based on the F1 score for the hate class, **SGD** and **SVM** are the top performers, with **SGD** slightly easier to implement and often faster to train.

7. FUTURE SCOPE

As the internet and social media platforms continue to grow, the necessity for effective hate speech detection systems becomes increasingly critical to provide safe and courteous online environments. Machine learning algorithms in hate speech detection are broad and changing. The major areas where we can expect significant advancements and opportunities are:

Multilingual models are employed with advanced algorithms for the development of sophisticated models for improved accuracy and generalization across multiple languages and dialects, addressing the global nature of online platforms. These capabilities can be multiplied by making the model context-aware, which will enable the system to understand the nuances and subtleties of language, including sarcasm, slang, and cultural references, in addition to integrating text with other modalities, such as images, videos, and audio, to detect hate speech that relies on context beyond just text.

Another scope for research in this direction is for detection and mitigation of hate speech promptly by developing scalable solutions that can process vast amounts of data in real time. Ethical considerations and bias mitigation are another area where research and development should be focused on reducing bias in hate speech detection algorithms to ensure fair treatment of all user groups. Creating transparent models with explainable AI (XAI) techniques to allow users to understand why certain content was flagged as hate speech.

The future of hate speech detection in machine learning holds promise for creating safer online spaces through advanced technology, ethical considerations, and collaborative efforts across multiple domains.

8. REFERENCES

[1] Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., & Patti, V. (2021). Resources and benchmark corpora for hate speech detection.: a systematic review. Language Resources and Evaluation, 55, 477-523.

[2] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017), May). Automated hate speech detection and the problem of offensive language. In Proceedings of the international AAAI conference on web and social media (Vol. 11, No. 1, pp. 512-515).

[3] Aljero, M. K. A., & Dimililer, N. (2021). A novel stacked ensemble for hate speech recognition. Applied Sciences, 11(24), 11684.

[4] Mnassri, K., Rajapaksha, P., Farahbakhsh, R., & Crespi, N. (2022, December). BERT-based Ensemble Approaches for Hate Speech Detection. In GLOBECOM 2022-2022 IEEE Global Communications Conference (pp. 4649-4654). IEEE.

[5] Hegde, A., Anusha, M. D., & Shashirekha, H. L. (2021). Ensemble based machine learning models for hate speech and offensive content identification. In Forum for Information Retrieval Evaluation (Working Notes)(FIRE), CEUR-WS. org.

[6] Agarwal, S., & Chowdary, C. R. (2021). Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. Expert Systems with Applications, 185, 115632.

[7] Mutanga, R. T., Naicker, N., & Olugbara, O. O. (2022). Detecting Hate Speech on Twitter Network using Ensemble Machine Learning. International Journal of Advanced Computer Science and Applications, 13(3).

[8] Zimmerman, S., Kruschwitz, U., & Fox, C. (2018, May). Improving hate speech detection with deep learning ensembles. In Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).

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

[10] 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).

[11] Alfina, I., Mulia, R., Fanany, M. I., & Ekanata, Y. (2017, October). Hate speech detection in the Indonesian language: A dataset and preliminary study. In 2017 international conference on advanced computer science and information systems (ICACSIS) (pp. 233-238). IEEE.

[12] Asogwa, D. C., Chukwuneke, C. I., Ngene, C. C., & Anigbogu, G. N. (2022). Hate speech classification using SVM and naive BAYES. arXiv preprint arXiv:2204.07057.

[13] Burnap, P., & Williams, M. L. (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy & internet, 7(2), 223-242.

[14] Mohapatra, S. K., Prasad, S., Bebarta, D. K., Das, T. K., Srinivasan, K., & Hu, Y. C. (2021). Automatic hate speech detection in english-odia code mixed social media data

using machine learning techniques. Applied Sciences, 11(18), 8575.

[15] Gambäck, B., & Sikdar, U. K. (2017, August). Using convolutional neural networks to classify hate-speech. In Proceedings of the first workshop on abusive language online (pp. 85-90).

[16] Alsafari, S., Sadaoui, S., & Mouhoub, M. (2020, November). Deep learning ensembles for hate speech detection. In 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 526-531). IEEE.

[17] Ibrohim, M. O., & Budi, I. (2019, August). Multi-label hate speech and abusive language detection in Indonesian Twitter. In Proceedings of the third workshop on abusive language online (pp. 46-57).

[18] Gomez, R., Gibert, J., Gomez, L., & Karatzas, D. (2020). Exploring hate speech detection in multimodal publications. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 1470-1478).

[19] 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.

[20] Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012, September). Detecting offensive language in social media to protect adolescent online safety. In 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing (pp. 71-80). IEEE.

[21] Kumar Sharma, H., Kshitiz, K., & Shailendra. (2018). NLP and machine learning techniques for detecting insulting comments on social networking platforms. In Proceedings on 2018 international conference on advances in computing and communication engineering, ICACCE 2018, IEEE (pp.265–272). Online Resources: https://rdcu.be/duIzm (retrieved on 15th August 2024)