International Journal of Advanced Statistics and Computer Science – Volume 1 Issue 10, May 2021

Available at www.ijascs.com

RESEARCH ARTICLE OPEN ACCESS

Hate Speech Detection using Machine Learning Technique - Logistic Regression

Akshay Gunjal[1], Shaun Dsilva[2], Davon Carvalho[3], Dr. Vaishali Jadhav[4]

[1](Information Technology, St. Francis Institute Of Technology, Mumbai
Email: gunjal.akshay06@ student.sfit.ac.in),
[2](Information Technology, St. Francis Institute Of Technology, Mumbai
Email: shaundsilva610@ student.sfit.ac.in),
[3](Information Technology, St. Francis Institute Of Technology, Mumbai
Email: dave6286@ student.sfit.ac.in),
[4](Information Technology, St. Francis Institute Of Technology, Mumbai
Email: vaishalijadhav@sfit.ac.in)

----------##########----------

Abstract:

Escalation in internet forums has increased the amount of hate speech in massive scale. Abusive and threatful content is spreading quickly because of the increase in number of social media platforms. Also, there are meagre filtration techniques to detect and remove hate speech. In our proposed system we have used twitter dataset for detecting hate speech. Various classification models like Radom Forest, Naive Bayes, Support Vector Machine (SVM), Logistic Regression are used. Different features like Term Frequency-Inverse Document Frequency (TFIDF), Count Vectorizer and Word2Vec are considered.

Keywords — **Hate Speech detection, TF-IDF, twitter, tweets, SVM, Text-Analytics, Logistic Regression.**

----------##########----------

I. INTRODUCTION

Hate speech is defined as ”Hate Speech are words, activities which is restricted in light of the fact that it prompts acts that trigger insurgency and savagery perspectives toward others or gatherings“. [4]

With 59 percent of the global population digitally connected to the internet, majority of them are registered to various social media platforms. With growing population the user generated content i.e sharing and exchanging of data is also rising, with this the amount of hate speech is also steadily increasing. Because of absence of strategy and steady implementation, both Facebook and Twitter are utilized for assault on individuals dependent on qualities like race, nationality, sex and sexual direction. Despite the fact that there is no proper meaning of Hate Speech there is an agreement that it targets distraught gathering of people in a way that is conceivably hurtful to them. [1]

Oksanen et al. [2] researched the degree of openness to and exploitation by online disdain material among youthful web-based media clients and announced that 67% of youth, between 15 to 18 years of age, had been presented to digital disdain on Facebook and YouTube where 21% of them had become the survivors of such material.

Section 153A in The Indian Penal Code – Advancing ill will between various gatherings on grounds of religion, race, spot of birth, home, language, and so on, and doing acts biased to upkeep of congruity. [12]

Section 295(A) of the Indian Penal Code (IPC) enacted in 1927 – Whoever, with conscious and noxious expectation of offending the strict sensations of any class of 273 (residents of India), 274 (by words, either spoken or composed, or by signs or by noticeable portrayals or something else), affronts or endeavours to

affront the religion or the strict convictions of that class, will be rebuffed with detainment of one or the other depiction for a term which may stretch out to 4(three years), or with fine, or with both. [13]

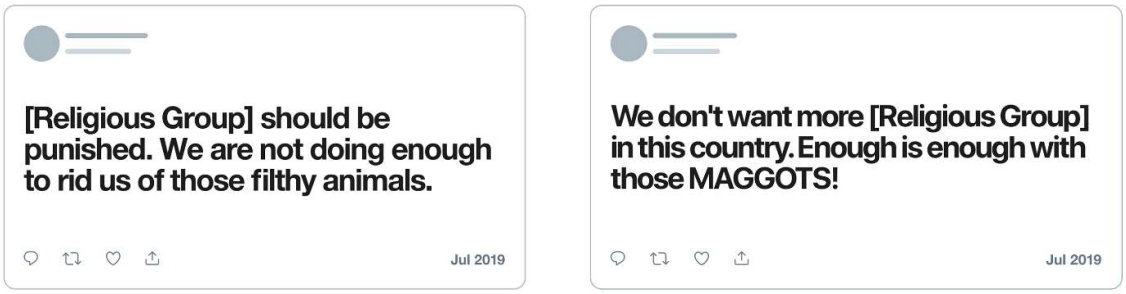

Fig. 1 Example of religious hate speech on twitter

This paper puts forward an attempt to detect hate speech based on Twitter dataset using various machine learning algorithms and comparing the results of those algorithms. The algorithms which we used were Random Forest, Naive Bayes, Support Vector Machine (SVM), Logistic Regression and features like Term Frequency-Inverse Document Frequency (TFIDF), Count Vectorizer, and Word2Vec. To demonstrate the working of this project we have deployed the model on a web page using StreamLit and an Android Chatting Application which will detect hate speech.

II. Related Work

This research explains the classification of hate speech based on Twitter dataset using SVM and Random Forest. It uses various hateful and offensive tweets and passes them through hate speech detection algorithm to classify them. In this research uni-grams are collected from the tweets and are passed through the classification algorithm. It achieved the accuracy of 87.4% for classifying offensive tweets and 78.4% in case of hate tweets. [7]

Hate Speech isn't just spreading through words yet additionally through talks and remarks for pictures, status or recordings those are being transferred to informal organization. There are lot of models for text processing but very few for image processing. This research focuses on hateful sentiments in form of texts in images which are present on various social media platforms. It was developed using Latent Semantic Analysis (LSA) having F1-score of 88%. It is a lightweight approach. [9]

In this examination a regulated learning model is work to arrange digital disdain towards ladies. It utilizes Twitter dataset. Turkish tweets dependent on decision of attire for ladies have been gathered and are passed to five AI calculations which incorporate Support Vector Machine (SVM), Naive Bayes, Random Forest, and Random Tree with exactness of 74%, 75%, 75%, 77% individually. [10]

In this exploration the creator gathers tweets of different arbitrary records whose tweets contain disdain discourse through twitter API. An Artificial Neural Network technique is utilized alongside Backpropagation calculation to enhance it. The consequence of test had normal exactness of 80.664%, review of 90.07%, precision of 89.47%. [11]

From the above works a few algorithms were chosen, since the model is based on text classification, Latent Semantic Analysis (LSA) was discarded. Out the other chosen algorithms Logistic Regression was selected as the final algorithm since it produced best results.

III. RESEARCH METHOD

A. Data Mining

Information mining is an interaction utilized by organizations to transform crude information into helpful data. Information mining is considered as accessible innovation, for example, AI, data set frameworks, insights, and representation. [4]

B. Data Cleaning

The dataset used is a Twitter Dataset. It has 24,802 tweets and the data is classified into 3 classes i.e. hate, offensive and non-hate. In the original dataset, the class column identifies each tweet as 0 for hate speech, 1 for offensive language and 2 for neither. The labels were changed to 2 for offensive language, 0 for non-hate and 1 for hate speech. All the texts were converted to lower case, text in square brackets was removed, special characters were taken off, and words containing numbers were eliminated.

The most frequent words for each classification were identified and the normalized frequency was calculated. The corpus of those words having higher normalized frequency was created and the stop words were unattached.

C. Lemmatization

Lemmatization is a standardization strategy in NLP that is utilized to get ready content, words and records for additional handling. Lemmatization attempts to make a word reference base type of the word (lemma) by utilizing profound morphological investigation, a word reference, and the setting of the word in its decrease cycle. In this way, the importance of a word is less inclined to change when lemmatize is utilized. [8]

D. Feature Engineering

A combination of 3 features was utilized in this study. This includes Count Vectorizer, Term Frequency-Inverse Document Frequency (TF-IDF), and Doc2Vec. The tally vectors were fundamentally term recurrence (TF). These highlights were changed over into vectors utilizing check Vectorizer and TFIDF Vectorizer libraries accessible in Scikit-learn AI library. The TF-IDF vectors as a component were utilized to figure the recurrence of explicit term in each tweet to that of whole dataset. TF-IDF scores for various word blends were determined.

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum t' \in d f t', d}$$

$$\mathrm{idf}\,(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Count Vectorizer is used to convert a text document to vector. It also enables pre-processing before converting to vectors. It removes the punctuation marks and converts all the words to lower case. It outputs a unique vector for every unique word and shows the frequency of a particular word.

One disadvantage of using Count Vectorization or TF-DF Vectorization is that we can run into the Curse of Dimensionality, which occurs when the number of features explodes. This problem happens because these methods create Sparse Vectors, that are the length of the total vocabulary of the text corpus. This corpus has a vocabulary of 20,277. Therefore, that is the number of columns in the Sparse Matrix, creating a ton of extra space of 99% 0s could possibly hurt the model.

Doc2Vec feature engineering is an extension of Word2Vec. It means to figure out how to extend a report into an inactive d-dimensional space. In particular, the Distributed Bag of Words (DBOW) model disregards the setting words in the information, however, rather powers the model to anticipate words haphazardly tested from the section in the yield.

E. Classifiers

In this section baseline Random Forest, Logistic Regression, Naive Bayes, and Support Vector Machine (SVM) models will be used.

- **Random Forest Classifier**: Random Forest, as the name proposes comprises of enormous number of choice trees. Every individual tree in irregular timberland gives out a class expectation and the class with most votes turns into the last forecast.
- **Logistic Regression**: Logistic Regression is another common model used for classification tasks. Additionally, this model tends to work better with larger datasets.
- **Naive Bayes**: Naive Bayes is another common baseline classification algorithm that uses Bayes Theorem. Every pair of features being classified is independent of each other. Each tuple classifies the condition as fit("Yes") or unfit("No").

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

- **Support Vector Machine (SVM)**: An SVM is a type of classifier that modifies the loss function for optimization to not only consider overall accuracy metrics of the resulting predictions, but also to maximize the decision boundary between the data points. This helps tune the classifier as a good balance between under fitting and over fitting.

F. Model

The vectors made from TF-IDF Vectorizer were passed to the Random Forest, Logistic Regression and it was seen that Logistic Regression performed better compared to Random Forest. The F1-score increased. Now further the vectorizer was passed to Naive Bayes and detected that the F1-score decreased. This model performed worse than both the Random Forest and Logistic Regression models. Support Vector Machine model produced the highest F1-score.

Now the features were changed to Doc2Vec, Count Vectorizer and Doc2Vec Distributed Bag of Word model (DBOW) was generated. Passing these vectors to SVM, Doc2Vec method produced an overfit model.

Using Count Vectorization on the Logistic Regression baseline actually produced the highest F1 and Recall out of all the other models. It was able to achieve balance. Although the Linear SVM with Doc2Vec had a higher Recall, its F1 was much lower. Additionally, this model was underfit, meaning that it was not too specific to the training data.

G. Dealing with class imbalance

The Logistic Regression model dealt with class imbalance. To improve this model two class imbalance methods were used.

- **Oversampling with SMOTE**: This method oversamples the minority class," Hate Speech". Rather than simply oversampling the minority class with replacement (which adds duplicate cases

to the dataset), the algorithm generates new sample data by creating 'synthetic' examples that are combinations of the closest minority class cases. After synthetically resampling the data, there is no longer needed to lean on penalized class weights to improve the model tuning.

- **Under-Sampling with Tomek Links:** This method under-samples the majority class," Not Hate Speech." Tomek joins are sets of extremely close cases, yet of inverse classes. Eliminating the cases of the larger part class of each pair builds the space between the two classes, working with the arrangement cycle.

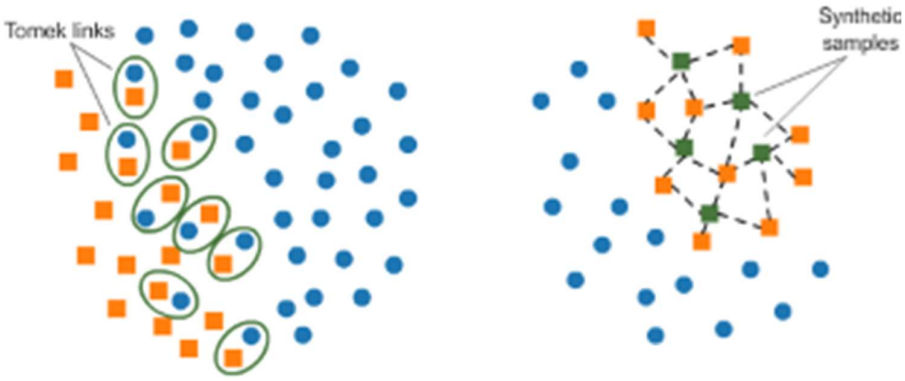

Fig. 2 Working of SMOTE and Tomek

IV. RESULTS

Although using Tomek Links performed better than using SMOTE, the resulting F1-score still is not as good as the initial Logistic Regression model's F1-score. But ideally, the need was for a balance between high Recall and high F1-score. The Logistic Regression model with Count Vectorizer had a slightly lower Recall, but a much higher F1-score. Ultimately, Logistic Regression was used as the final model.

F1-Score: 0.3958
Weighted F1-Score: 0.9121

TABLE 1: PERFORMANCE OF VARIOUS ALGORITHMS

Sr. No	Classifier	Precision	Recall	Weighted-F1
1	Random Forest-TFIDF	0.412844	0.161290	0.927249
2	Log Reg-TFIDF	0.293900	0.569892	0.913449
3	Naive Bayes-TFIDF	0.411765	0.125448	0.925487
4	SVM-TFIDF	0.360947	0.437276	0.928112

It's a pretty close call between the Logistic Regression and SVM baselines. An observation was made that the precision of Naive Bayes and Random Forest is better compared to other models whereas recall of Logistic Regression and SVM is greater. But overall, the Linear SVM model performed the best across both uniform and weighted F1 with values 0.360947 for precision, 0.437276 for recall and 0.928112 for weighted-f1.

TABLE 2: PERFORMANCE OF VARIOUS ALGORITHMS

Sr. No	Classifier	Precision	Recall	Weighted-F1
1	SVM-Doc2Vec	0.205148	0.634660	0.873748
2	SVM-Count Vectorizer	0.271218	0.536496	0.910436
3	Log Reg-Count Vectorizer	0.289831	0.624088	0.912113
4	Log Reg Oversampled	0.232558	0.474453	0.902366
5	Log Reg Under sampled	0.570175	0.237226	0.937658

The evaluation metrics between all of these iterations were very similar. For instance, the baseline Linear SVM with Doc2Vec had the highest Recall, but a mediocre F1-score. Doc2Vec was able to predict more of the positive class ”Hate Speech” when it comes to Recall, but the F1-scores for each label are much lower than the TF-IDF vectorization method. Whereas the Logistic Regression model with Count Vectorizer had a slightly lower Recall, but a much higher F1-score. Ultimately, Logistic Regression model was the final model with precision of 0.289831, recall of 0.624088 and weightedf1 of 0.912113.

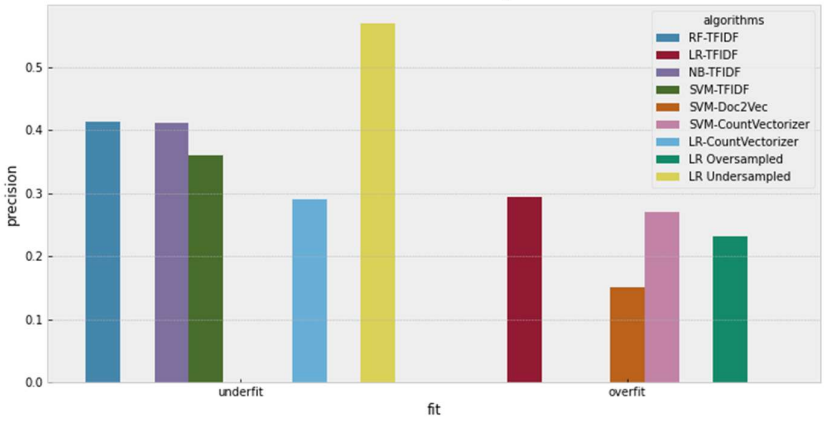

Fig. 3 Precision

Fig. 3 shows the examination of accuracy upsides of the multitude of calculations referenced in the above tables.

$$\text{precision} = \frac{TP}{TP + FP}$$

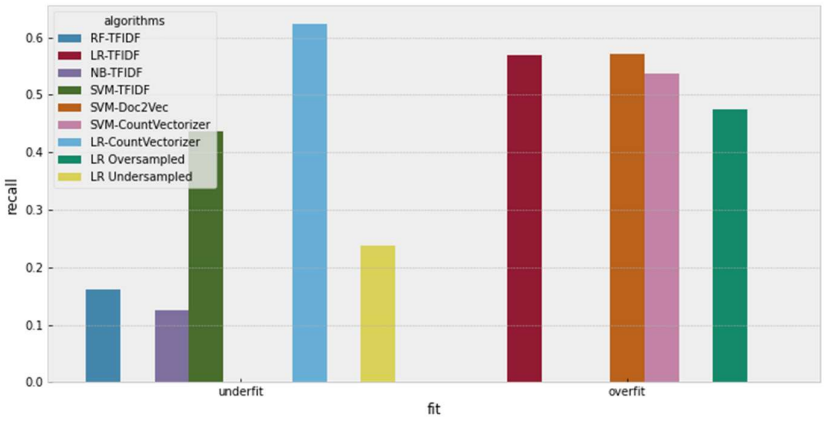

Fig. 4 Recall

The comparison of recall values of individual algorithms is depicted in the graph giving a conclusion that the performance of Logistic Regression is superior in relation to others.

$$\text{recall} = \frac{TP}{TP + FN}$$

F1-score is used as the main metric. The F1-score finds the harmonic mean between Precision and Recall, and it's useful for data with high class imbalance. The F1-score of SVM using TF-IDF i.e 0.395462 and Logistic regression using Count Vectorizer i.e., 0.395833 is almost similar. But Logistic Regression has slightly higher value than SVM.

$$F1 = 2.\frac{\text{precision. recall}}{\text{precision} + \text{recall}}$$

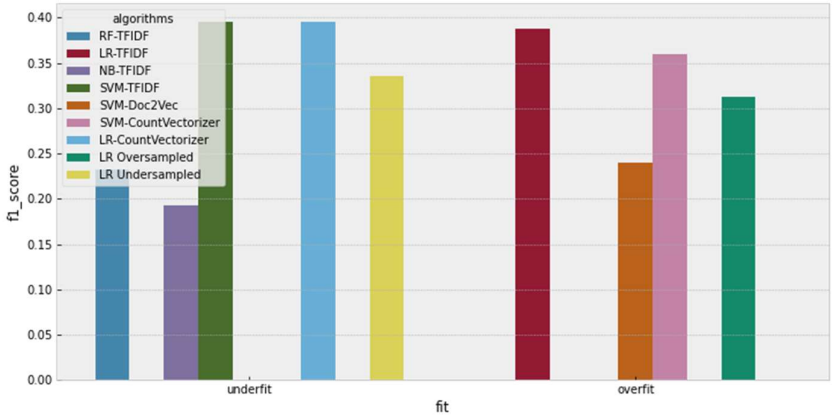

Fig. 5 F1-Scores

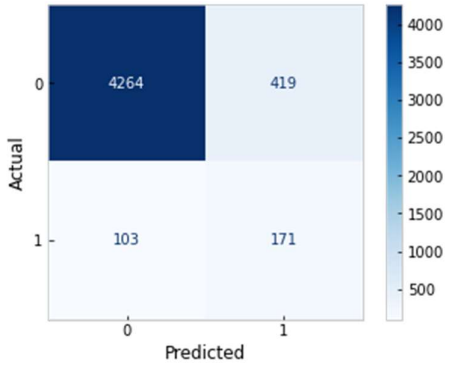

Fig. 6 Confusion Matrix

According to this confusion matrix, the True Negative rate is high i.e., 4264, but the True Positive rate is much lower i.e. 171. Also, False Positive is 419 and False Negative is 103. Further normalizing the confusion matrix to get better understanding of the relations.

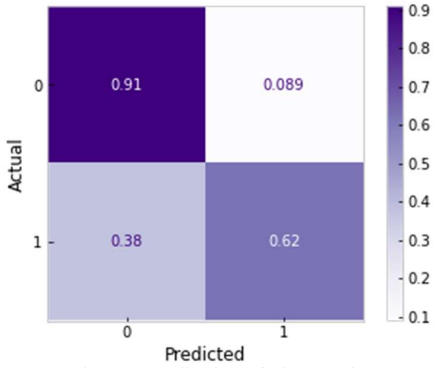

Fig. 7 Normalized Confusion Matrix

The final model has a true negative rate of 91% and true positive rate of 62%. Also, we can see that only 8.9% of predictions are False Positives. Which mean that they were classified as" Hate Speech" when it's not.

V. CONCLUSION

Logistic Regression had an F1 of .3958 and Recall of .624. Although this project had extensive pre-processing and modelling iterations, there is still room for improvement. It is important to understand why the model performed poorly and how that relates to the problem. The F1-score was brought down by the" Hate Speech" label predictions. The model was able to predict 91% of the" Not Hate Speech" labels correctly but could only predict 62% other label.

Two major issues that we faced in this project were:

1. Imbalanced dataset
2. Difficulty of the model to understand hate speech

The issue of class imbalance is manageable with pre-processing techniques and oversampling/under sampling techniques. However, identifying hate speech is an overall problem that many major tech companies like Twitter, Facebook and Instagram are still struggling with.

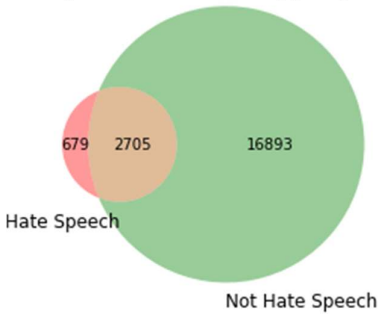

Fig. 8 Comparison of unique words

Here, there are 679 words unique to the" Hate Speech" label, 16893 as "Non Hate Speech" and 2705 as "Offensive". Some of these words are meaningless, but some are especially hateful terms. Ultimately, automating hate speech detection is an extremely difficult task because of the nuances in English language.

This machine learning model is deployed on web-based application created using StreamLit and an Android chat application where the hate / offensive words are censored.

VI. FUTURE SCOPE

This project was able to get that process started, but there is much more work to be done to keep this content off public-facing forums.

To further develop this project, here are some immediate next steps that anyone could execute:

- Collect more potential" Hate Speech" data.
- Improve final model with different pre-processing techniques, such as removing offensive language as stop words.
- Evaluate model with new tweets or other online forum data to see if it can generalize well.
- Using LSA for detecting hate speech in images and other media as well.

REFERENCE

[1] Jacobs, James B. "Hate Crime: Criminal Law and Identity Politics: Author's summary." Theoretical Criminology 6, no. 4 (2002): 481-484J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[2] Oksanen, Atte, James Hawdon, Emma Holkeri, Matti Nasi, and Pekka ¨ Ras¨ anen. "Exposure to online hate among young social media users." ¨ In Soul of society: a focus on the lives of children & youth. Emerald Group Publishing Limited, 2014M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.

[3] Ombui, E., Muchemi, L., & Wagacha, P. (2019). Hate Speech Detection in Code-switched Text Messages. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). doi:10.1109/ismsit.2019.8932845 (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[4] Niam, Ilham Maulana Ahmad, Budhi Irawan, Casi Setianingsih, and Bagas Prakoso Putra. "Hate Speech Detection Using Latent Semantic Analysis (LSA) Method Based on Image." In 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), pp. 166-171. IEEE, 2018. *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[5] Chakraborty, Puja, and Md Hanif Seddiqui. "Threat and Abusive Language Detection on Social Media in Bengali Language." In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1-6. IEEE, 2019. A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[6] Raufi, Bujar, and Ildi Xhaferri. "Application of machine learning techniques for hate speech detection in mobile applications." In 2018 International Conference on Information Technologies (InfoTech), pp. 1-4. IEEE, 2018.

[7] 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. doi:10.1109/access.2018.2806394.

[8] Albadi, Nuha, Maram Kurdi, and Shivakant Mishra. "Are they our brothers? Analysis and detection of religious hate speech in the Arabic Twittersphere." In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 69-76. IEEE, 2018.

[9] Wickramaarachchi, Wiraj Udara, and R. K. A. R. Kariapper. "An approach to get overall emotion from comment text towards a certain image uploaded to social network using Latent Semantic Analysis." In 2017 2nd International Conference on Image, Vision and Computing (ICIVC), pp. 788-792. IEEE, 2017.

[10] S¸ ahi, Havvanur, Yasemin Kılıc¸, and Rahime Belen Saglam. "Automated ˇ Detection of Hate Speech towards Woman on Twitter." In 2018 3rd International Conference on Computer Science and Engineering (UBMK), pp. 533-536. IEEE, 2018.

[11] Setyadi, Nabiila Adani, Muhammad Nasrun, and Casi Setianingsih. "Text analysis for hate speech detection using backpropagation neural network." In 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), pp. 159-165. IEEE, 2018.

[12] Indiankanoon.org. 2021. Section 153A in The Indian Penal Code. [online] Available at: ¡https://indiankanoon.org/doc/345634/¿ [Accessed 28 April 2021].

[13] Indiankanoon.org. 2021. Section 295A in The Indian Penal Code. [online] Available at: ¡https://indiankanoon.org/doc/1803184/¿ [Accessed 28 April 2021].

[14] En.wikipedia.org. 2021. Hate speech - Wikipedia. [online] Available at: ¡https://en.wikipedia.org/wiki/HateSpeech¿ [Accessed 29 April 2021].

[15] Schmidt, Anna, and Michael Wiegand. "A survey on hate speech detection using natural language processing." In Proceedings of the fifth international workshop on natural language processing for social media, pp. 1-10. 2017.

[16] Alorainy, Wafa, Pete Burnap, Han Liu, Amir Javed, and Matthew L. Williams. "Suspended accounts: A source of tweets with disgust and anger emotions for augmenting hate speech data sample." In 2018 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 2, pp. 581-586. IEEE, 2018.

[17] Rohmawati, Umu Amanah Nur, Sari Widya Sihwi, and Denis Eka Cahyani. "SEMAR: An Interface for Indonesian Hate Speech Detection Using Machine Learning." In 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 646-651. IEEE, 2018

[18] Ruwandika, N. D. T., and A. R. Weerasinghe. "Identification of hate speech in social media." In 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 273-278. IEEE, 2018.

