Measuring Hate Speech in Social Media Posts

Kareem Ehab Kassab 900182771
Computer Science and Engineering
The American University in Cairo
kareemikassab@aucegypt.edu

Mahmoud Elshinawy 900183926
Computer Science and Engineering
The American University in Cairo
mahmoudelshenawy@aucegypt.edu

I. Introduction

With the growing of social media platforms and the Internet, all people are posting, commenting on each others' posts, and engaging in different ways. Keeping this in mind, some people post offensive sentences against some races, religions, health status, colour, disability, sexual orientation, women, Arab, ethnicity, disability and so many others just to name a few. These kinds of posts are called hate speech posts, and the victims of hate speech may in some cases commit suicide as a result of these posts. Hate speech is any type of offensive, rude words that someone uses when talking to someone or posting addressing certain category of people. For hate crime, it is not just some words representing racism against certain category of people, but it is more of causing destruction or setting fire as an indication of intimidation or murder [Kpekoll 2020].

Putting hate crime dangers discussed above into consideration, we shall detect hate speech posts and remove them before they go viral. Given the fact that it is impossible to monitor all the posts issued at a certain point in time depending on human employees to classify which posts are hate speech and which are not, machine learning comes into the picture to solve this issue. In this paper, we will compare different machine learning models including Naive Bayes, SVM, Logistic Regression, Fast Text, and other methods to decide on the best approach to solve this problem. Additionally, we are going to investigate number of different datasets to get the most suitable dataset to work with to detect hate speech in social media posts.

II. Literature Review

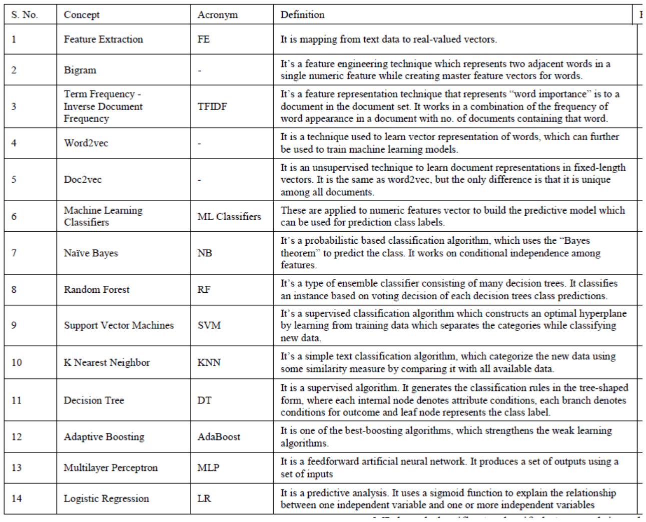
Hate Speech is an important text classification problem in machine learning that is being addressed in multiple paradigms including online communities, social media, and video games just to name a few. This has gained importance because with the growing of such online communities, grew the need to contain, and followingly, auto-detect hate speech to limit its usage in these social networks. Different methodologies and algorithms, datasets, and metrics have been used overtime to increase performance and each of them had performance strengths and weaknesses and will be discussed in this literature review.

The used methods in the literature can be classified into classical machine learning approaches, ensemble approach, and deep learning.

A. Classical Machine Learning Approaches

Examples include support vector machines (SVM), Naive Bayes (NB), Logistic Regress (LR), Decision Trees (DT), K-Nearest neighbor (KNN), etc. The commonly used ML algorithms for hate speech detection (Mullah and Zainon 2021). In Sindhu and Zahid's study, they implemented the previoudly mentioned algorithms and more illustrated in Figure 1 (Shaikh et.al 2020).

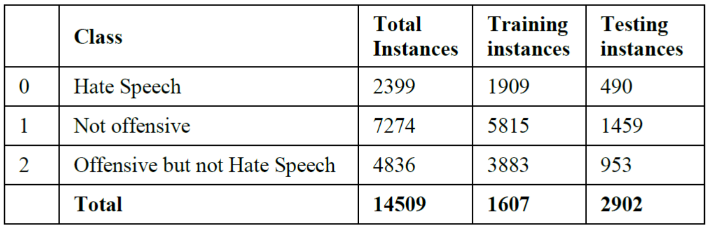
Their methodology included using a dataset of 14509 tweets from twitter classified into 3 classes: 16% of tweets belong to class hate speech. In addition, 50% of tweets belong to not offensive class and the remaining 33% tweets are offensive but not hate speech class. Their preprocessng included putting tweets in lowercase removing URLs usernames, hasthags, punctuation, and then tokenization from tweets– converting each tweet into a token or words. Finally, using porter stemmer to return words to their root forms. In their feature engineering, they used n-gram with TFIDF, Word2vec, and Doc2vec.



S. No.	Concept	Acronym	Definition
1	Feature Extraction	FE	It is mapping from text data to real-valued vectors.
2	Bigram	-	It's a feature engineering technique which represents two adjacent words in a single numeric feature while creating master feature vectors for words.
3	Term Frequency - Inverse Document Frequency	TFIDF	It's a feature representation technique that represents "word importance" is to a document in the document set. It works in a combination of the frequency of word appearance in a document with no. of documents containing that word.
4	Word2vec	-	It is a technique used to learn vector representation of words, which can further be used to train machine learning models.
5	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.
6	Machine Learning Classifiers	ML Classifiers	These are applied to numeric features vector to build the predictive model which can be used for prediction class labels.
7	Naïve Bayes	NB	It's a probabilistic based classification algorithm, which uses the "Bayes theorem" to predict the class. It works on conditional independence among features.
8	Random Forest	RF	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.
9	Support Vector Machines	SVM	It's a supervised classification algorithm which constructs an optimal hyperplane by learning from training data which separates the categories while classifying new data.
10	K Nearest Neighbor	KNN	It's a simple text classification algorithm, which categorize the new data using some similarity measure by comparing it with all available data.
11	Decision Tree	DT	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.
12	Adaptive Boosting	AdaBoost	It is one of the best-boosting algorithms, which strengthens the weak learning algorithms.
13	Multilayer Perceptron	MLP	It is a feedforward artificial neural network. It produces a set of outputs using a set of inputs
14	Logistic Regression	LR	It is a predictive analysis. It uses a sigmoid function to explain the relationship between one independent variable and one or more independent variables

Fig. 1. Machine Learning Approach.

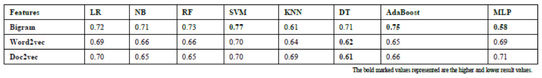
They divided the dataset to 80% training and 20% testing data. The following table shows the distribution of data after division of the sets.



	Class	Total Instances	Training instances	Testing instances
0	Hate Speech	2399	1909	490
1	Not offensive	7274	5815	1459
2	Offensive but not Hate Speech	4836	3883	953
	Total	**14509**	**1607**	**2902**

Fig. 2. Data Distribution After Set Division.

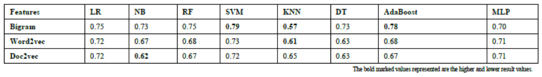
Later they tried the different algorithms mentioned in table 1 as there is no optimum algorithm for all datasets. They reported the results to be as follows: "In all 24 analyses, the lowest precision (0.58), recall (0.57), accuracy (57%) and F-measure (0.47) found in MLP and KNN classifier using TFIDF features representation with bigram features. Moreover, the highest recall (0.79), precision (0.77), accuracy (79%) and F-measure (0.77) were obtained by SVM using TFIDF features representation with bigram features. In feature representation, bigram features with TFIDF obtained the best performance as compared to Word2vec and Doc2vec"(Shaikh et.al 2020). Tables 3 of Table 6 show the precision, recall, F-measure and accuracy of all 24 analyses.



Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.72	0.71	0.73	**0.77**	0.61	0.71	**0.75**	**0.58**
Word2vec	0.69	0.66	0.66	0.70	0.64	**0.62**	0.65	0.69
Doc2vec	0.70	0.65	0.65	0.70	0.69	**0.61**	0.66	0.71

The bold marked values represented are the higher and lower result values.

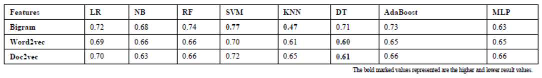
Fig. 3. Precision of All 24 Analysis.



Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.75	0.73	0.75	**0.79**	**0.57**	0.73	**0.78**	0.70
Word2vec	0.72	0.67	0.68	0.73	**0.61**	0.63	0.68	0.71
Doc2vec	0.72	**0.62**	0.67	0.72	0.65	0.63	0.67	0.71

The bold marked values represented are the higher and lower result values.

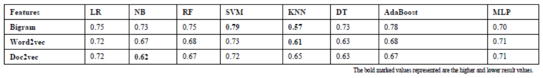
Fig. 4. Recall of All 24 Analysis.



Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.72	0.68	0.74	**0.77**	**0.47**	0.71	0.73	0.63
Word2vec	0.69	0.66	0.66	0.70	0.61	**0.60**	0.65	0.65
Doc2vec	0.70	0.63	0.66	0.72	0.65	**0.61**	0.66	0.66

The bold marked values represented are the higher and lower result values.

Fig. 5. F-Measure of All 24 Analysis.



Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.75	0.73	0.75	**0.79**	**0.57**	0.73	0.78	0.70
Word2vec	0.72	0.67	0.68	0.73	**0.61**	0.63	0.68	0.71
Doc2vec	0.72	**0.62**	0.67	0.72	0.65	0.63	0.67	0.71

The bold marked values represented are the higher and lower result values.

Fig. 6. Accuracy of All 24 Analysis.

If we were to summarize the classifier discussion in this study, we would notice that the lowest performance was obtained amongst the NB, DT, MLP and KNN classifiers, while SVM classifier had the best performance between all other classifiers. Figure 7 shows the confusion matrix for the SVM in this study (Shaikh et.al 2020). Class 0 represents hate speech, class 1 represents not offensive speech, and class 2 is offensive but not hate speech.

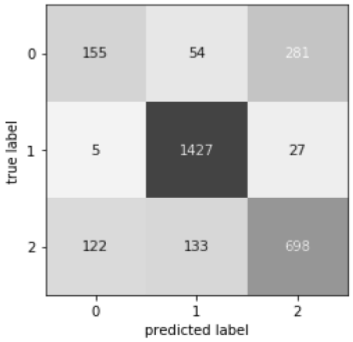
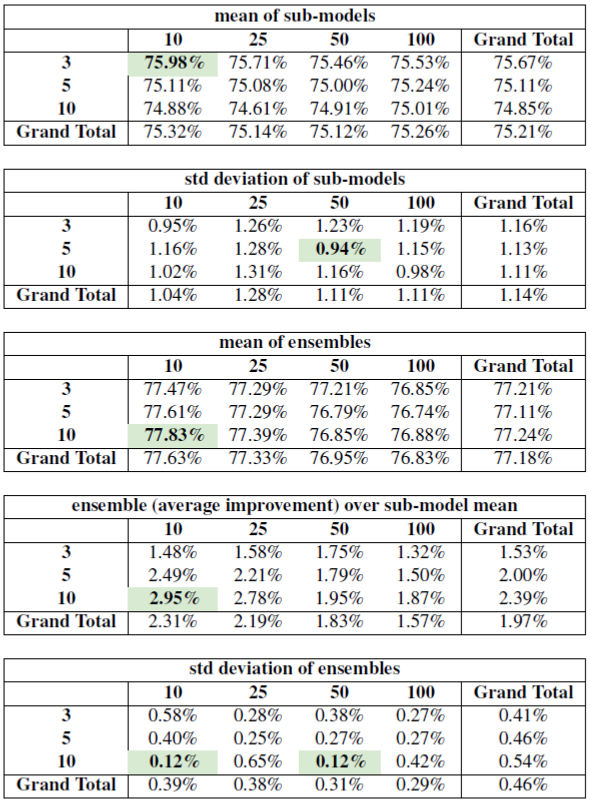


Fig. 7. Confusion Matrix for SVM.

B. *Ensemble Approach*

Ensemble techniques can be used for sentiment classification of tweets (Melton et.al 2020). In Joshua's study, they used ensemble methods with neural networks using variable weight initializations. They used 2 datasets: the abusive speech one (Waseem and Hovy, 2016) and the SemEval 2013 sentiment analysis (Nakov, 2013). They divided the first one 85% training and 15% testing. Using CNN classifier they yielded the following results



mean of sub-models					
	10	25	50	100	Grand Total
3	**75.98%**	75.71%	75.46%	75.53%	75.67%
5	75.11%	75.08%	75.00%	75.24%	75.11%
10	74.88%	74.61%	74.91%	75.01%	74.85%
Grand Total	75.32%	75.14%	75.12%	75.26%	75.21%

std deviation of sub-models					
	10	25	50	100	Grand Total
3	0.95%	1.26%	1.23%	1.19%	1.16%
5	1.16%	1.28%	**0.94%**	1.15%	1.13%
10	1.02%	1.31%	1.16%	0.98%	1.11%
Grand Total	1.04%	1.28%	1.11%	1.11%	1.14%

mean of ensembles					
	10	25	50	100	Grand Total
3	77.47%	77.29%	77.21%	76.85%	77.21%
5	77.61%	77.29%	76.79%	76.74%	77.11%
10	**77.83%**	77.39%	76.85%	76.88%	77.24%
Grand Total	77.63%	77.33%	76.95%	76.83%	77.18%

ensemble (average improvement) over sub-model mean					
	10	25	50	100	Grand Total
3	1.48%	1.58%	1.75%	1.32%	1.53%
5	2.49%	2.21%	1.79%	1.50%	2.00%
10	**2.95%**	2.78%	1.95%	1.87%	2.39%
Grand Total	2.31%	2.19%	1.83%	1.57%	1.97%

std deviation of ensembles					
	10	25	50	100	Grand Total
3	0.58%	0.28%	0.38%	0.27%	0.41%
5	0.40%	0.25%	0.27%	0.27%	0.46%
10	**0.12%**	0.65%	**0.12%**	0.42%	0.54%
Grand Total	0.39%	0.38%	0.31%	0.29%	0.46%

Fig. 8. CNN Classifier.

C. *Deep Learning*

With much larger datasets and non-linear data being introduced, deep learning has been introduced to be able to provide higher accuracy than the classical machine learning algorithms (Nanlir, 2021). The deep learning approaches focused on 2 variations of 2 main approaches: the CNN (convolutional neural network) and RNN (recurring neural network) (Nanlir, 2021). Using different variants of neural networks including Long Short Term Memory (LSTM), and Gated recurrent units (GRUs) to solve Task6 of SemEval-2019; in this research, the BILSTM-CNN gave the best results. In another research that tried DNN (deep neural network), In this research, the following variants of DNN were used; FastText, CNNs and LSTMs; it also showed an improved results than its preceding one (Nanlir, 2021). However, the deep learning approaches needed more data to learn and describe as it can be seen in the curve:

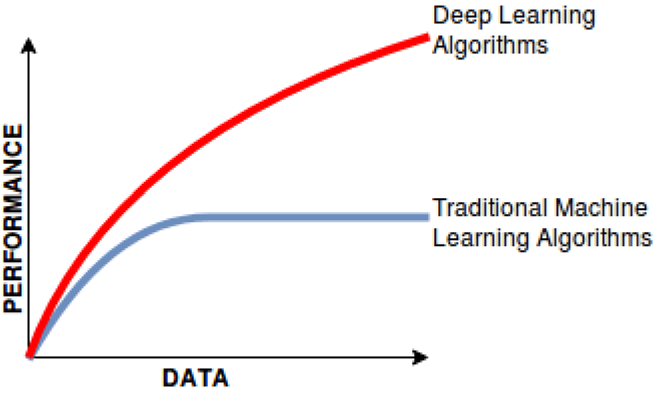
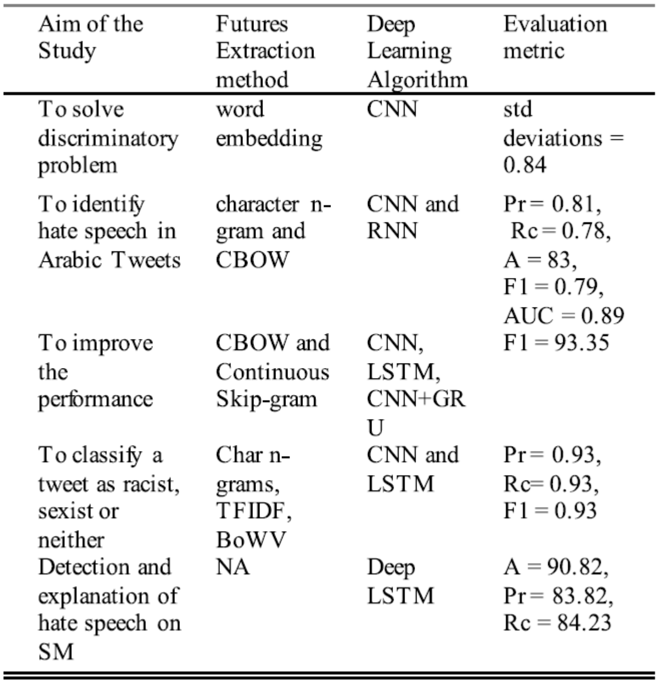


Fig. 9. Deep Learning vs Traditional Machine Learning

Nanlir and Wan Summarized their efforts in the following table:



Aim of the Study	Futures Extraction method	Deep Learning Algorithm	Evaluation metric
To solve discriminatory problem	word embedding	CNN	std deviations = 0.84

To identify hate speech in Arabic Tweets	character n-gram and CBOW	CNN and RNN	Pr = 0.81,
			Rc = 0.78,
			A = 83,
			F1 = 0.79,
			AUC = 0.89

To improve the performance	CBOW and Continuous Skip-gram	CNN, LSTM, CNN+GRU	F1 = 93.35

To classify a tweet as racist, sexist or neither	Char n-grams,
	TFIDF,
	BoWV
		CNN and LSTM	Pr = 0.93,
			Rc= 0.93,
			F1 = 0.93

Detection and explanation of hate speech on SM	NA	Deep LSTM	A = 90.82,
			Pr = 83.82,
			Rc = 84.23

Fig. 10. Nanlir and Wan Summarization

III. Dataset Choice

In this section, we will compare some datasets to choose the one with the best quality in terms of size and features

A. *hate_speech18 Dataset*

The data is extracted from a white racism forum called StormFront and other subforums indicated by subforum_id feature. The labels has been put manually by analyzing the sentences provided in the dataset. The hate_speech18 dataset can be found here.

This dataset contains the following column fields:

- text: the actual message.
- user_id.
- subforum_id.
- num_contexts.
- label: hate, noHate, or relation.

The problem with this dataset is that it is about 100 instance which is so small, so we will not be using this dataset for our project.

B. *Hate Speech and Offensive Language Dataset*

The data is obtained from Kaggle and its most recent update was two years ago. It contains 24874 instances which is not large enough. This dataset can be found here. The data is collected from Twitter tweets, and then CrowdFlower users are asked to mark the tweet as one of three categories: hateSpeech, offensiveLanguage ,or neither, with a minimum three users to participate in each tweet, and applying the majority vote. The following are the column fields in the dataset:

- count: number of CrowdFlower users who voted for the tweet.
- hate_speech: how many people of those users mark the tweet as a hate speech.
- offensive_language: how many people of those users mark the tweet as offensive language.
- neither:how many people of those users mark the tweet as neither.
- class: deciding the tweet class hateSpeech, offensiveLanguage ,or neither.
- tweet: tweet actual text.

This dataset has its limitations. For instance, the size of the data is not enough, the data is not the most recent, and it is only derived from Twitter tweets, so I would prefer if we have more data taken from different platforms

C. *Measuring Hate Speech Kaggle Dataset*

The data is obtained from Kaggle and it is the most recent as it updated on January 21st, 2022. It contains 135557 instances which is large enough. This dataset can be found here.

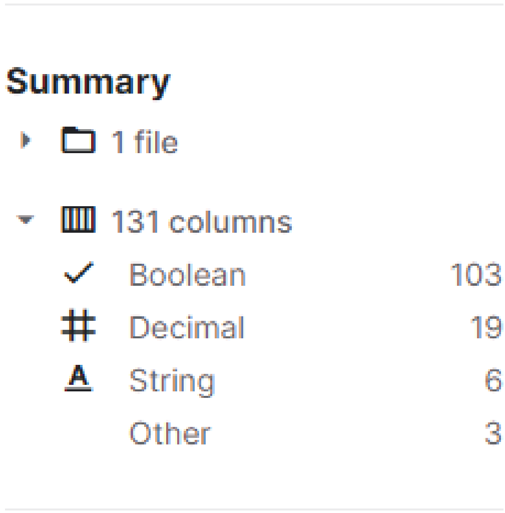
The following are the most important column fields in the dataset:

- hate_speech_score - continuous hate speech measure, where higher = more hateful and lower = less
- hateful
- text - lightly processed text of a social media post
- comment_id - unique ID for each comment
- annotator_id - unique ID for each annotator
- sentiment - ordinal label that is combined into the continuous score
- respect - ordinal label that is combined into the continuous score
- insult - ordinal label that is combined into the continuous score
- humiliate - ordinal label that is combined into the continuous score
- status - ordinal label that is combined into the continuous score
- dehumanize - ordinal label that is combined into the continuous score
- violence - ordinal label that is combined into the continuous score
- genocide - ordinal label that is combined into the continuous score
- attack_defend - ordinal label that is combined into the continuous score
- hatespeech - ordinal label that is combined into the continuous score
- annotator_severity - annotator's estimated survey interpretation bias

Unlike the previous datasets, this dataset contains enough number of training instances, but it has some problems. For instance, there are 132 unneeded columns as shown by the following figure such as target_age_seniors, other target target features, annotator_ideology just to name a few. This problem can be easily overcome by discarding all these features from the data before using it. Given the fact that this dataset is the most recent and it has large enough size, we will be using it for training our model.

IV. Discussion and Results

Addressing the previous works in the literature



Summary

▸ 🗀 1 file

▾ ▥ 131 columns

✓	Boolean	103
#	Decimal	19
A	String	6
	Other	3

Fig. 11. Measure Hate Speech Kaggle Summary.

review. We intend to explore several implementations of neural networks including RNNs, DNNs, and CNNs. We are aiming to explore their different accuracies and maybe compare them to the SVM implementation in the literature review or our own. This decision was influenced mostly by observing the difference in accuracy, precision, and recall in the datasets above. Since we know that there is no optimum algorithm for all datasets we will need to validate our work on our own datasets. Each of the addressed studies were using a set or two of their own. Most probably, we will start by implementing the classical machine learning classifiers and compare their results to each other, and then to the neural networks. Our main aim is to try to maximize the accuracy of identifying hate speech from text; this is useful in making online communities more healthy, friendly, and suitable for everyone in the online social space.

References

[1] Calvin Erico Rudy Salim and Derwin Suhartono. 2020. A systematic literature review of different machine learning methods on hate speech detection. JOIV : International Journal on Informatics Visualization 4, 4 (2020), 213. DOI: http://dx.doi.org/10.30630/joiv.4.4.476

[2] Chris Kennedy. 2019. Measuring hate speech. (October 2019). Retrieved February 13, 2022 from https://hatespeech.berkeley.edu/

[3] "Desktop Operating System Market Share Worldwide." StatCounter Global Stats, gs.statcounter.com/os-market-share/desktop/worldwide. Magnetics Japan, p. 301, 1982].

[4] Gorman, Mel (15 February 2004). Understanding the Linux Virtual Memory Manager

[5] Joshua Melton, Arunkumar Bagavathi, and Siddharth Krishnan. 2020. Del-hate: A deep learning tunable ensemble for hate speech detection. 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) (2020). DOI:http://dx.doi.org/10.1109/icmla51294.2020.00165

[6] Kpekoll. 2020. Hate speech and hate crime. (October 2020). Retrieved February 13, 2022 from https://www.ala.org/advocacy/intfreedom/hate

[7] Love, Robert (2010). Linux kernel development. Addison-Wesley. p. 4.

[8] Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. (2013). Semeval-2013 task 2: Sentiment analysis in twitter. In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), pages 312–320.

[9] Nanlir Sallau Mullah and Wan Mohd Zainon. 2021. Advances in machine learning algorithms for hate speech detection in Social Media: A Review. IEEE Access 9 (June 2021), 88364–88376. DOI:http://dx.doi.org/10.1109/access.2021.3089515

[10] Safa Alsafari, Samira Sadaoui, and Malek Mouhoub. 2020. Deep Learning Ensembles for hate speech detection. 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI) (2020). DOI:http://dx.doi.org/10.1109/ictai50040.2020.00087

[11] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. Retrieved February 13, 2022 from https://journals.plos.org/plosone/article?id=10.1371

[12] Sindhu Abro, Sarang Shaikh, Zahid Hussain, Zafar Ali, Sajid Khan, and Ghulam Mujtaba. 2020. Automatic hate speech detection using machine learning: A comparative study. International Journal of Advanced Computer Science and Applications 11, 8 (2020). DOI: http://dx.doi.org/10.14569/ijacsa.2020.0110861

[13] Sudhir Kumar Mohapatra, Srinivas Prasad, Dwiti Krishna Bebarta, Tapan Kumar Das, Kathiravan Srinivasan, and Yuh-Chung Hu. 2021. Automatic hate speech detection in English-Odia code mixed social media data using Machine Learning Techniques. Applied Sciences 11, 18 (2021), 8575. DOI:http://dx.doi.org/10.3390/app11188575

[14] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. Retrieved February 13, 2022 from https://journals.plos.org/plosone/article?id=10.1371

[15] Waseem, Z. (2016). Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. Proceedings of the 1st Workshop on Natural Language Processing and Computational Social Science, pages 138–142.