2023 International Conference for Advancement in Technology (ICONAT)
Goa, India. Jan 24-26, 2023

Hate Speech Detection Network Using LSTM

Chirag Lala
AIT CSE
Chandigarh University
India
chirag42001@gmail.com

Pulkit Dwivedi
AIT-CSE
Chandigarh University
India
pdwivedi1990@gmail.com

2023 International Conference for Advancement in Technology (ICONAT) | 978-1-6654-7517-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICONAT57137.2023.10080786

***Abstract*—Social media is an extremely popular form of communication today. People offer their opinions and insights on a variety of topics, including politics, video games, and their personal lives. These platforms are occasionally used by some people to propagate false information about another person or group of people. The term ”hate statement” refers to this kind of offensive material. One of the most well-known social media platforms is Twitter. But many people also use Twitter to disseminate offensive material. It is very hard to manually weed out abusive comments from the hundreds of millions of tweets that are generated every day on Twitter. Therefore, these offensive tweets ought to be automatically filtered out. In this study, we are developing an LSTM model for categorising tweets as either containing hate content or not. The dataset used is a publicly available dataset on Kaggle. This model has an accuracy of 98.04% on the training dataset and 97.19% on the validation dataset. For the test dataset, a precision of 0.98 is obtained for non-hate tweets and 0.93 for hate statement tweets.**

***Index Terms*—LSTM, RNN, CNN, Natural Language Processing, Deep Learning**

I. Introduction

Twitter is one of the most popular social media platforms in the world today. According to the company's ”second quarter 2022 results” which came on July 22, 2022, the number of monetizable daily active users(mDAU) [1] was 237.8 million.

As per bolggingwizard.com [2] at least 500 million tweets are sent every day on Twitter. All of this generates lots of data. On Twitter, people share their opinions and mindsets about almost everything, ranging from politics to product reviews. People share their views, and this influences other people and has an effect on their mindset. So, Twitter generates a large volume of sentiment rich data. It provides businesses with an opportunity to grow by giving them a large audience to market and advertise their product.

Such a huge volume of data could not be analysed directly by a single person, and we need the computing power of machines to analyse such a large volume. The analysis of this data could tell companies what customers are thinking about them and their products, which would ultimately help them analyse and improve their products.

From textual data if we remove the factual information what is left are opinions, emotions, and perspectives, and all of this forms the sentiment contained in that text. And we can use this sentiment rich data in order to make intelligent decisions.

People from varying backgrounds (cities, countries,religions) go through varying experiences in their daily life and like to share their thoughts on Twitter. This provides big opportunities for researchers and companies to understand and analyse what different people are thinking about a certain topic or product. Or what is the stuff that is being talked about more, in other words, what is the current hot topic.

If a certain trend regarding certain types of goods or services is picking up, proper analysis of the situation provides an opportunity for a new business of that product or service. All of the things described above are tasks of NLP [5]. In this paper, we will try to predict slightly different things using NLP for Twitter sentiment Analysis.

As almost everyone is allowed to create an account on Twitter without any kind of background or authenticity check (which is completely understandable as the number of people using the platform is so high and growing everyday, it is almost impossible to put a check on), and post tweets and react on other tweets on the platform, people many times start using the website as a platform to promote negative things and influence other peoples minds in a negative way.

People try to spread hate speeches about a group or a particular individual, which many times leads to highly undesirable outcomes in the physical world. These things might affect the mind of an individual to a very great extent, which could ultimately spoil his life and that of thousands of others also. Sometimes individuals might get effected by some very radical or extremist groups which ultimately leads to brain washing of the individual.

When these tweets come to notice of a significant amount of people then complaints are filed against them and sometimes complete accounts are taken down (or just tweets removed). But by that time, the damage is already done. So there should be a way for the platform to detect and censor out these types of tweets and that is what we are doing in this paper.

A considerable amount of work has been done in the field of ”Twitter Sentiment Analysis”. Hate speech detection is a different task when compared with product review , but it still is a classification task, and we can use similar algorithms. The kinds of biases that model might develop will be different and interesting to see.

Many other researchers have also done work for hate speech detection on Twitter.The goal of this paper is to present an LSTM model for hate speech detection on Twitter data and compare the results with already done work by other

researchers to understand which models are the best for the work.

II. Related Work

Watanabe et al. [3] used unigrams and other features that are automatically collected from dataset, which are used to train the algorithm.They conducted binary classification to detect if the tweet entry is offensive, and ternary classification to classify tweet as hateful, offensive or clean. For binary they reached an accuracy of 87.4% and for ternary 78.4%.

Fauzi et al. [4] worked on Indonesian Language dataset for classifying hate speech.They used ensembel method for hate speech classification.They used five separate algorithms and two ensembel methods of hard and soft voting. The algorithms used were Random Forest,K - Nearest Neighbours, Naive Bayes, Support Vector Machines and Maximum Entropy.They got best result when they used soft voting, the result had F1 score of 79.8% and 84.7% for unbalanced and balanced datasets.

Zhang et al. [6] argues that hate speech detection is a tough task because their analysis on various datasets, showed that hate speech does not have unique discriminating features. They proposed a Deep Neural Network model for the purpose of extraction of features.And the model is very effective when capturing hate speech semantics.Their model performed better that the best methods by five percent and eight percent in macro average F1 and hateful content detection respectively.

Pitsilis et al. [7] used an ensemble model in their work. They used various RNN classifiers in their ensemble model.They also added features to their model that worked on user related information.All this data is also fed to classifiers along with usual words vectors that are made from tweets.The classification of their model had higher quality than best state-of-the-art methods.

Kshirsagar et al. [8] presented neural network approach for hate speech classification. Their model worked on normal classification into hate speech and non hate speech and also worked on classifying statement into racist or sexist class specifically.They used already trained word embeddings in their model. They have used method of max/mean pooling on simple transformations of these word embeddings.Their model performs better than the best modern models.They used lesser parameters and less feature processing as compared to other methods.

For the purpose of learning semantic word embeddings, Badjatiya et al. [9] undertook extensive tests on deep learning architectures. Their research demonstrates a performance difference of 18 F1 points between word n-gram and deep learning models.

Typically, many models train supervised learning algorithms using carefully (manually) designed features. Robinson et al. [10] compare manual feature engineering to automatic feature engineering. They used Twitter data for feature selection analysis, and their results go against the grain of the popular wisdom regarding the significance of manual feature engineering. They have demonstrated that the precisely planned features are substantially reduced by over 90% via automatic feature selection. When automatic feature selection is used instead of manual feature selection, the models that result perform better.

The significance of various pre-processing methods in sentiment categorization and natural Language processing has been highlighted by Naseem et al. [11]. Two alternative word level feature extraction models were applied. Both conventional and deep learning classifiers were employed by them to verify performance increases.

According to Mutanga et al. [12], most machine learning algorithms perform poorly due to ineffective sequence transduction.The authors created a model based on transformers. The attention-based recurrent neural networks and various transformer baselines were contrasted with the DistilBERT transformer technique. Their research demonstrated that the parallelizable DistilBERT algorithm outperformed alternative baseline methods

With the aid of convolutional operations, Roy et al. [13] proposed DCNN (Deep Convolutional Neural Network) model that captures Twitter semantics and achieves precision, recall, and F1 scores of 0.97, 0.88, and 0.92, respectively, using tweet text and the GIoVe embedding vector.

Using a stacked weighted ensemble model, Kokatnoor et al. [14] have combined five separate classifiers using random forest, naive bayes, soft voting, linear regression and hard voting. The results revealed enhanced performance when compared to solo classifiers, with an accuracy of 95.54 percent in binary categorization of tweets into hate speech.

Aluru et al. [15], have worked on multilingual hate speech identification from sixteen data sources in nine languages. The scientists found that basic models, like LASER embedding with logistic regression, performed well in low resource settings, but BERT-based models outperformed them in high resource settings.

An LSTM classifier was used by Syam et al. [16] to do binary categorization of tweets into hate and non-hate. They developed a system that can find tweets based on hashtag searches.

The impact of user traits on Twitter hate speech recognition was examined by Unsvåg et al. [17]. Three separate datasets were used in experiments on a hate speech classifier that revealed that adding certain user characteristics to textual features somewhat improved classification performance.

A probabilistic clustering technique for categorising hate speech was created by Ayo et al. [18]. To gather tweets with hate speech keywords, they employed a meta data extractor.Using the tf-idf model, feature representation was carried out. The classification of tweets was done automatically using a rule-based clustering approach. The categorization of hate speech also made use of fuzzy logic.

Isnain et al. [19] combined a continuous bag-of-word (CBOW) architecture with the word2vec feature extraction approach and a bidirectional long short term memory technique. Accuracy, precision, recall, and F measure were employed for

testing purposes. They were successful in obtaining a F score of 96.29

LSTM-based categorization has been suggested by Bisht et al. [20] to distinguish between hate speech and offensive language. They explain how to employ LSTM and Bi-LSTM in conjunction with word embeddings.

Oriola et al. [21] performed their work on South African tweets dataset.Both n-gram, syntactic based features were extracted and analysed using various algorithms. Their result showed that optimised SVM with character n-gram and gradient boost with word n-gram performed best when just classifying into hate speech, but they showed poor result when detecting other classes. For detection of other other classes multi tier models performed best.

Omar et al. [22] in his work has first emphasised upon the fact that most of the hate speech detection research has been done for English language, and very less work has been done on Arabic Language.So, they first collected the data from various different online platforms. And to test the quality/effectiveness of their dataset they proposed twelve ML models and two DL models,among which RNN showed better performance than other models.

Hochreiter et al. [23] propsed LSTM for the first time and described why LSTM helps to store information over longer time than normal RNNs.

III. PROPOSED METHODOLOGY

A. Data Preprocessing

1) Removing Emojis: Tweets are unstructured data and contain a lot of unnecessary information. One of those things are emojis. Emojis are used a lot by people on various social media platforms, to express their feelings and Twitter is no exception to this. Although emojis can convey very important information about feelings of individuals and what they want to say, but in our work we are not considering any contribution of emojis. So, before starting any kind of modelling we are removing emojis from all the tweet entries.

2) Puntutation Removal: Tweets are in English language and have punctuation marks, although sometimes these punctuation marks can also give some information regarding feeling expressed in the tweets, mostly they will not convey useful information regarding the emotion in the tweet. And hence, in this work we are removing all punctuation marks before moving forward.

3) Stop-word Removal: Stop words are words that are the most commonly occurring words in any language. These words do not add any useful information to text. We remove these words to remove low-level information in the text. So that our model can focus on the important information.Removal of stopwords actually reduces the size of our dataset which results in reduce of the training time. Hence in this work we are removing all the stop words from all the tweet entries.

4) Oversampling: Oversampling and under-sampling are methods to balance imbalanced datasets. Imbalanced datasets are those datasets which have a very skewed ratio of the majority and minority classes. This is known as bias in the training dataset and this bias might lead to poor performance of model. Sometimes this imbalance might lead to the model completely ignoring the minority class in training dataset.

To remove this imbalance we can either collect more data and this time do it in a balanced way.Otherwise, there are two ways called as under-sampling and oversampling for removing this imbalance.

In under-sampling we reduce the occurrences of the majority group in our dataset. In oversampling we duplicate the occurrences of the minority group in our dataset.

While analysis of the dataset we find that our dataset has 29720 non-hate tweets and 2242 hate tweets,this is skewed towards non-hate speech tweets.Initially ratio of tweets that did not contain hate content to those that did was approximately 14 to 1. Oversampling is employed on this situation to maintain balance. By replicating the entries of hate tweets, the number of entries for hate tweets is increased to three times the original(and rose to 6726). Following this procedure, the ratio of tweets that are not hate statements to those that are, decreases to approximately 5:1.

B. Train-Test-Validation Split

The train dataset is split into train and test sets and test set gets 20% percent of the entries. Then the new train set is further divided into two parts, one of them is train dataset and the other is the validation dataset.20% of the entries goes to the validation set.Even after division between train, test and validation sets none of the sets were skewed towards any class of data.

C. Tonkenzation

Tokenization means breaking down the text or data we have into smaller parts.Tokenization of a text can be done into sentences or words. The final sentences or words are known as tokens.In this work tokenization has been done in words.

Tokenization is important because before any actual model building can be done or machine learning or deep learning models can be applied the text need to be divided into units such as words or numbers so that this can be fed to the models.

The maximum vocabulary has been set to 50 thousand words,having very long vocabulary might lead to more training time for the model.

D. Model Building

1) Layer 1- Embedding layer: The model's very first layer is the embedding layer. The work of embedding layer is to take the words in the vocabulary as input (which have already been encoded in form of integers), make word vectors out of them. These are the word vectors that are to be fed to the further layers for training.These word vectors are learned overtime when model is trained.As, training of model continues the word vectors for similar words become similar (or close) and word vectors for dissimilar words become far apart. The embedding dimensions have been taken as 16 for this work. This means that words are vectorized in to a vector of dimension 16.

2) Layer 2-Dropout Layer: A Dropout layer is used to solve the problem of over-fitting. When we use multi-layer neural networks, there is always a tendency for overfitting to occur. Overfitting means when our model performs very very well on training dataset and does not perform good on unseen data. We may either increase the number of training samples, lower the number of features, or do regularisation to address the issue of overfitting. In the embedding layer already regularization has been implemented. Dropout layer is used to decrease the number of features.Dropout layer does this by completely deactivating some of the neurons and this deactivation happens randomly.

3) Layer 3-LSTM Layer: This layer is the LSTM (Long Short Term Memory) layer. Although LSTM is a form of RNN (recurrent neural network), it was created to address RNN's flaws.RNN gives good predictions when predicting based on recent information, but if a word has long term dependency then RNN fails to predict it.LSTM is good at prediction when any word has a long term dependency(it can retain information for a long period of time). Figure 1 represents architecture of a general LSTM layer.For figure and equations of LSTM reference has been taken from a git hub page [25]. Hochreiter et al. [23] was the first to propose LSTM. An LSTM layer uses three different types of gates: forget gates, input gates, and output gates.

Forget Gate : The output of this gate decides that which information from the previous cell state has to be discarded now. This is decided by a sigmoid function, which outputs between 0 and 1, 0 means completely discard it and 1 means completely keep it.Sigmoid decides this by looking at the current input word and previous hidden layer.

$$frgt = \sigma(W_{frgt}[h_{t-1}, x_t] + b_{frgt}) \tag{1}$$

$$\sigma(g) = \frac{1}{1 + e^{-g}} \tag{2}$$

Forget gate output is represented by $frgt$, hidden state from the previous word is represented by h_{t-1}, the current input is represented by x_t, and bias is represented by b_frgt. The values with which h_{t-1} and x_t will be multiplied are different, and W_frgt it is only a representation of those values.

Input Gate : The output of this gate decides that what information from the current input has to be used to update the previous cell state to new cell state. The gate takes input of previous hidden state and current word and uses a sigmoid function. The output of this gate is then multiplied with output of tanh function.The tanh function also takes the same inputs.

$$inpg = \sigma(W_{inpg}[h_{t-1}, x_t] + b_{inpg}) \tag{3}$$

Above is equation for output of input gate

$$C_{0t} = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{4}$$

These two are multiplied and then added with multiplication of previous cell state and forget gate output.

$$C_t = inpg * C_{0t} + frgt * C_{t-1} \tag{5}$$

C_{0t} is the output of tanh function,$inpg$ is the output of input gate,h_{t-1} represents hidden state from previous word and x_t represents current input and b_{inpg} and b_c represents bias.C_t represents the new cell state.C_{t-1} represent the old cell state.W_c and W_{inpg} are representative of the values with which previous hidden state and current input will be multiplied in both equations.

Output Gate : Output of this gate contributes in determination of next hidden state. This also takes the same input as the other two gates and this also uses the sigmoid function. Output of this gate is multiplied with tanh of new cell state for determination of hidden state.

$$otpg = \sigma(W_{otpg}[h_{t-1}, x_t] + b_{otpg}) \tag{6}$$

$$h_t = otpg * tanh(C_t) \tag{7}$$

$otpg$ represents output of output gate, h_{t-1} represents hidden state from previous word and x_t represents current input and b_{otpg} represents bias.W_{otpg} is representative of the values with which h_{t-1} and x_t will be multiplied,h_t represents new hidden state and C_t represents new cell state.

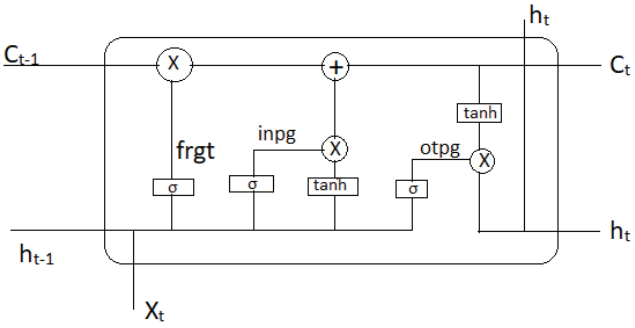

Fig. 1. Architecture of LSTM layer

4) Layer 4: Flatten Layer: This layer flattens out multi-dimensional matrix into a one dimensional array.Layer 4 is simply used for flattening out the output of layer 3. Layer 3 in this model gives a 50x16 array, and layer 4 after flattening it gives an array of 800 elements.

5) Layer 5:Dense Layer: A dense layer has multiple neurons and the same input is fed to all the neurons. Output from all the neurons in the previous layer is fed to each of the neurons in the Dense layer. In this work layer 5, which is a dense layer is fed a output of flatten layer having 800 elements and, it gives an output of 512 elements because it has 512 neurons. A ReLU (Rectified Linear Unit) activation function is used here, equation for which is given below.

$$ReLU(x) = max(0, x) \tag{8}$$

6) Layer 6:Dropout Layer: Layer 6 is again a dropout layer. Takes input from the dense layer having 512 elements, and gives output having 512 elements but with some neurons deactivated.

7) Layer 7:Dense layer: This is the final output layer this has only one neuron and a sigmoid activation function and gives output between 0 and 1.

E. Layer Summary

Layer (type)	Output Shape
embedding (Embedding)	(None, 50, 16)
dropout (Dropout)	(None, 50, 16)
lstm (LSTM)	(None, 50, 16)
flatten (Flatten)	(None, 800)
dense (Dense)	(None, 512)
dropout_1 (Dropout)	(None, 512)
dense_1 (Dense)	(None, 1)

Fig. 2. A Summary Of Layers And Outputs

Figure 2 gives summary of all the layers that are being used in this model.It also mentions the output shape of each layer used in this model.

IV. EXPERIMENTAL RESULTS

A. Dataset Used

The dataset that has been used in this work has been taken from Kaggle.com [24], it is publicly available dataset on Kaggle. This same dataset has been used in the work by Roy et al. [13]. There is no description given on the website, so we don't know the original source of data and who has collected it.

The dataset used in this work is an English language dataset.There are different train and test files in the dataset. Train file has a total of 31962 tweet entries and test file has 17198 tweet entries. The tweet entries in the train file have been assigned the labels 0 and 1, with 0 designating non-hate statement and 1 representing hate statement, respectively. The tweets in the test file, however, have not been categorized into hate statement and non-hate statement categories.

Out of 31962 Twitter entries in the train file, 29720 tweet entries have been categorized as non-hate statements and 2242 tweet entries as hate statements.

B. Model Performance and Comparison with State-Of-The-Art Models

The greatest accuracy for the model is 98.04% on the training dataset and 97.19% on the validation set. For non-hate statement tweets and hate statement tweets, the model's precision on the test dataset is 0.98 and 0.93, respectively. For non-hate statement and hate statement tweets, the model shows recall of 0.98 and 0.91, respectively. Precision, recall, F1 score, and accuracy are the assessment criteria that are being employed. The model's performance on each of these assessment measures is shown in Figure 3.

	precision	recall	f1-score
0	0.98	0.98	0.98
1	0.93	0.91	0.92
accuracy			0.97
macro avg	0.95	0.95	0.95
weighted avg	0.97	0.97	0.97

Fig. 3. Performance Of Model

TABLE I
COMPARISON WITH STATE-OF-THE-ART MODELS

Accuracy Metric	Proposed LSTM	Roy et al. CLSTM [13]	Roy et al. DCNN [13]	Kamble et al. [26]
Precision	0.93	0.75	0.97	0.83
Recall	0.91	0.43	0.88	0.79
F1-score	0.92	0.55	0.92	0.81

Figure 4 shows a distplot of how output is distributed between 0 and 1 for the test dataset. Figure 5 shows distplot

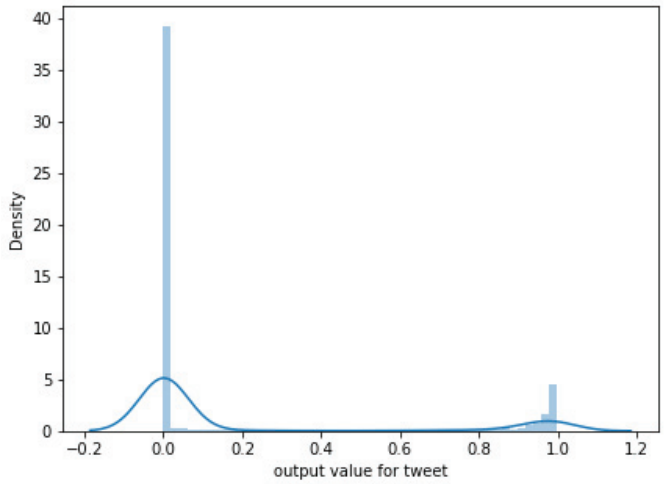

Fig. 4. Distribution Of Output On Test Data Using Distplot

of how the output is distributed between 0 and 1 for the test file. Table I shows a comparison between the proposed model and existing models.

V. CONCLUSION

In this study, an LSTM model is used to categorise tweets as either hate statements or non-hate statements. On training and validation data, the model's best accuracy is 98.04 % and 97.19 %. For the test dataset, it has a recall of 0.98 and 0.91

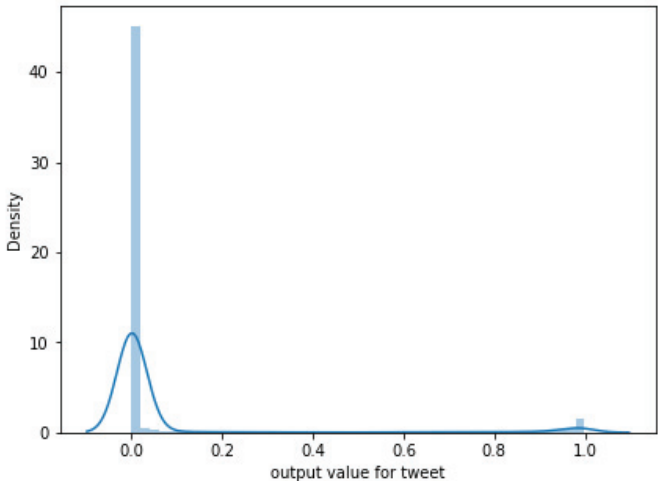

Fig. 5. Distribution Of Output On Test File Using Distplot

on non-hate and hate statements and precision of 0.98 and 0.93 on non-hate and hate statements.

The current work may be greatly improved. First and foremost, the same dataset should be used to compare effective CNN and LSTM models against transformer-based models. Then, attempts should be made to make similar models but with the feature of taking user or account information and history into consideration before predicting the category. Models can be made to categorise hate statements further into different classes.

REFERENCES

[1] Twitter Second Quarter 2022 Results https://s22.q4cdn.com/826641620/files/doc_financials/2022/q2/Final_Q2'22_Earnings_Release.pdf

[2] https://bloggingwizard.com/Twitter-statistics/

[3] H. Watanabe, M. Bouazizi and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," in IEEE Access, vol. 6, pp. 13825-13835, 2018

[4] Fauzi, Muhammad & Yuniarti, Anny. (2018). Ensemble Method for Indonesian Twitter Hate Speech Detection. Indonesian Journal of Electrical Engineering and Computer Science.

[5] P. Dwivedi and A. Upadhyaya, "A Novel Deep Learning Model for Accurate Prediction of Image Captions in Fashion Industry," 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2022, pp. 207-212, doi: 10.1109/Confluence52989.2022.9734171.

[6] Ziqi Zhang, Lie Luo , "Hate speech detection: A solved problem? The challenging case of long tail on Twitter," Semantic Web, vol. 10, no. 5, pp. 925–945, Sep. 2019.

[7] Pitsilis, Georgios & Ramampiaro, Heri & Langseth, Helge. (2018). Effective hate-speech detection in Twitter data using recurrent neural networks. Applied Intelligence. 48. in press. 10.1007/s10489-018-1242-y.

[8] Kshirsagar, Rohan & Cukuvac, Tyrus & McKeown, Kathy & McGregor, Susan. (2018). Predictive Embeddings for Hate Speech Detection on Twitter. 26-32. 10.18653/v1/W18-5104.

[9] Pinkesh Badjatiya,Shashank Gupta,Manish Gupta,Vasudeva Varma, "Deep Learning For Hate Speech Detection in Tweets" in 26th International World Wide Web Conference, perth, Australia, 2017

[10] Robinson, David, Ziqi Zhang and Jonathan A. Tepper. Hate Speech Detection on Twitter: Feature Engineering v.s. Feature Selection ESWC (2018).

[11] Naseem, Usman Razzak, Imran & Eklund, Peter. (2021). A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on Twitter. Multimedia Tools and Applications. 80. 1-28. 10.1007/s11042-020-10082-6.

[12] Raymond T Mutanga, Nalindren Naicker, Oludayo O Olugbara, Hate Speech Detection in Twitter using Transformer Methods in (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 11, No. 9, 2020

[13] Roy, P., Tripathy, A., Das, T. Gao, X. (2020). A framework for hate speech detection using deep convolutional neural network. IEEE Access, 8:204951–204962. doi: 10.1109/ACCESS.2020.3037073

[14] Kokatnoor, Sujatha & Dr. K, Balachandran. (2020). Twitter Hate Speech Detection using Stacked Weighted Ensemble (SWE) Model. 87-92. 10.1109/ICRCICN50933.2020.9296199.

[15] Sai Saketh Aluru, Binny Mathew1, Punyajoy Saha,Animesh Mukherjee Deep Learning Models for Multilingual Hate Speech Detection European Conference, ECML PKDD 2020, Ghent, Belgium

[16] S. S. Syam, B. Irawan and C. Setianingsih, "Hate Speech Detection on Twitter Using Long Short-Term Memory (LSTM) Method," 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2019, pp. 305-310, doi: 10.1109/ICITISEE48480.2019.9003992

[17] Elise Fehn Unsvåg and Björn Gambäck. 2018. The Effects of User Features on Twitter Hate Speech Detection. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pages 75–85, Brussels, Belgium. Association for Computational Linguistics.

[18] Femi Emmanuel Ayo, Olusegun Folorunso, Friday Thomas Ibharalu, Idowu Ademola Osinuga, Adebayo Abayomi-Alli, A probabilistic clustering model for hate speech classification in Twitter,Expert Systems with Applications,Volume 173, 2021,114762,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2021.114762.

[19] Auliya Rahman Isnain, Agus Sihabuddin, Yohanes Suyanto Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection IJCCS (Indonesian Journal of Computing and Cybernetics Systems) Vol.14, No.2, April 2020, pp. 169 178 ISSN (print): 1978-1520, ISSN (online): 2460-7258 DOI: 10.22146/ijccs.51743

[20] Akanksha Bisht, Annapurna Singh, H. S. Bhadauria, Jitendra Virmani, Kriti Detection of Hate Speech and Offensive Language in Twitter Data Using LSTM Model Recent Trends in Image and Signal Processing in Computer Vision, 2020, Volume 1124, ISBN : 978-981-15-2739-5

[21] Oluwafemi Oriola and Eduan Kotze, "Evaluating machine learning techniques for detecting offensive and hate speech in South African tweets," IEEE Access, vol. 8, pp. 21496–21509, 2020.

[22] Ahmed Omar, Tarek M. Mahmoud, Tarek Abd-El-Hafeez Comparative Performance of Machine Learning and Deep Learning Algorithms for Arabic Hate Speech Detection in OSNs Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020), 2020, Volume 1153, ISBN : 978-3-030-44288-0

[23] Sepp Hochreiter, Jurgen Schmidhuber, "Long short- ¨ term memory," Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276, 1997

[24] https://www.kaggle.com/code/ronikdedhia/Twitter-hate-speech-detection/data?select=train.csv

[25] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[26] Satyajit Kamble and Aditya Joshi, "Hate speech detection from code-mixed hindi english tweets using deep learning models," 2018, arXiv:1811.05145. [Online]. Available: http://arxiv.org/abs/1811.05145