Identification of Hate Speech on Social Media

Abhijeet Verma[a], Anshuman Singh[b], Sudhakar Tripathi[c], Sanjay Agarwal[d], Puneet Joshi[e]

[a,b,c] *Department of Information Technology, Rajkiya Engineering College, Ambedkar Nagar*

[d,e] *Department of Electrical Engineering, Rajkiya Engineering College, Ambedkar Nagar*

[a] abhijeetverma064@gmail.com,[b] anshumansingh9904@gmail.com,[c] stripathi@recabn.ac.in,[d] sanjay@recabn.ac.in,[e] drpuneetj@recabn.ac.in

Abstract

With the advent of microblogging services like Twitter, Facebook, and Tumblr, people's communication has become more indirect and reliable. People from different lifestyles and cultures interact with each other. They express their thoughts on many topics every day. This led to interpersonal conflict. As a result, the use of hate is increasing. Malicious language has become a serious problem. Manually searching for such content on these websites can be difficult. In recent years, detecting malicious language in content that are available online has become increasingly important in applications such as controversial event detection and analyzing the sentiment. Classifying the text of content that is generated online can be a daunting task because of the complexity of human spoken language and online users' microblogging, which has resulted in many informal and error-prone creations. This article presents a system to classify tweets into 2 groups (non-hate and hate), and also shows the implementation of the hate speech detection problem for text data written in English. There are several proposed studies on similar questions, all using classical machine learning approaches that rely heavily on feature engineering. Moving from one domain to another means redesigning the design of the function. This preliminary study proposes another method based on a deep learning approach that does not require feature engineering and can be applied to various contexts. Using a dataset derived from Twitter posts with more than 27,000 tweets, the proposed method provided the better results, with F1 score of 90% or higher.

1. Introduction

Hate speech spawning and spreading—the main cause of hate crime—is easy in the cyber virtual world that can’t be reached by the traditional legal system because there is a problem with anonymity ."Any statement that disparages a person or a group on the basis of specified attributes (referred to as hate types or hate classes) such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics" is how hate speech is defined". According to a Pew Research Center research survey of 4,248 US adults conducted in 2017, 41% of online users have experienced online hate and foul language. These types of occurrences occur when unsuitable photographs, offensive words, or messages are shared. Identifying and eliminating such content has been a challenge that social media academics have been working on for years. The automation of this activity has rapidly increased in latest years, in tandem influential and interested in the identification of hate speech online on social media. Although hate speech is one of the aforementioned classifications, we are focusing on filtering out tweets that contain hostile content based on racism from tweets in this study. Because of the vagueness of the work, it is fairly difficult. In natural language, a single word can have numerous meanings in a variety of situations. For instance, the word "fan". Occasionally referred to as an electrical appliance," this term refers to someone who is a celebrity's fan. Similarly, some words share the same situation. Variation in spelling is a different situation in which a user, whether knowingly or accidentally, alters a character or multiple characters in a word. If a simple term is used, the system does not recognize it as hate speech. The basic approach is employed. In terms of previous work, the majority of it focuses on manual feature extraction or representation learning, which is followed by machine learning classifiers. Deep learning models, on the other hand, have shown significant improvements in task accuracy when it comes to text. As a result, we want to integrate deep learning models with different M.L (machine learning) models to classify tweets. Collaborative learning is very popular these days, and it produces better results than previous methods. In this proposed strategy that we are using , we used a machine learning solution to overcome the natural language resistance. We're using Kaggle's dataset of more than 27,000 publicly available tweets. The main solution of this paper that we are providing are as follow (a) find training sets with annotations on Twitter and comments that are basic facts (b) developing several classification models and (c) evaluating the performance of models on each platform separately and By comparing the results of different approaches, as well as those reported in, our strategy improves accuracy by 5% .

2. Related work

Many different scholars have worked on finding hate speech in the English language. These studies used machine learning algorithms and data sources such as Twitter, Facebook, YouTube comment sections, and other online forums. The primary aim of the study was to focus on hate speech directed at people who are vulnerable in society. Most of them were minorities, teenagers, black people, etc. Hate speech is escalating day by day as social media is emerging. Identifying hate speech on social media such as Discord, Youtube, Reddit , Twitter and Facebook is challenging for the researchers. Many different approaches to classifying hate content that are used by researchers are SVM, Decision Tree, Naive Bayes, Logistic Regression, LSTM, and

CNN. Researcher Hema Krishnan worked on a hate speech detection system on Twitter using the Naive Bayes Classifier, on the basis of sentimental data . Naufal Riza implemented Naive Bayes to classify Indonesian language's hate speech. Kelvin Kiema and George Okeyo used the Naive Bayes classifier to conduct Twitter's Hate Speech. They produced better performance by producing an accuracy of 67.47%, a recall of 62%, and a precision of 58%. Along with Naive Bayes, Some researchers used another classifier such as Neural Network. Sajna Sharma uses the random forest classifier to solve word prediction. And she predicted dangerous words by using the same classifier and obtained a result of 76.42% accuracy. She got an accuracy of 72.4% by using the Naive Bayes algorithm and 72% by using SVM. Researcher Davidson also uses different classifiers like Porter-Steemer and logistic regression, Naive Bayes', random forest, and decision tree. He has developed a model of linear SVM and logistic regression to produce a high level of accuracy after evaluation by 5-fold cross-validation. Serven Malmasi uses the same dataset that was used by Davidson. E. Chandra Sekaran, A. Srinivasan, E. Gilbert, and M. Samory use a new concept that is a bag of community" to identify abusive content from the online community. They also use different algorithms such as Naive Bayes linear SVC and Logistic regression for the classification. Above, all the algorithms based on naive Bayes performed best. Burnap and William mainly focus on racism and identify hate speech on Twitter. Nohata et al., and n-gram function to predict hate speech detection. Wasen and Hovy used logistic regression and n-gram function to classify racism and sexism to classify tweets. Davidson and other researchers found that racism and homosexuality are malicious and offensive. They used many classifiers, such as Naive Bayes, decision tree, random forests, logistic regression, and support vector machines, to classify into different categories. Badjatiya et al. used deep learning models like convolutional neural networks and long short-term memory to detect hate speech content. They used several embeddings such as random, GloVe, and fastText embedding and found the best combination is LSTM, Random Embedding, and Gradient Boosted Decision Tree to classify disliked tweets.

In our research, we used different approaches that were previously used by researchers. We categorized tweets into hate and non-hate. To promote objectivity, we adopted their method of annotating the dataset, which included annotators from various backgrounds. We compared the performances of six classifiers: RF (Random Forest), NB (Naive Bayes), CNN, LR (Logistic regression) , SVM, and CNN+LSTM.

3. Methodology

We suggested an automated text classification strategy to address the shortcomings of the crude hate speech classification. In our research study, we constructed a text classifier machine learning model that may be utilized to detect hate content in a text corpus. To choose the features of the text corpus and classify them into hate and non-hate categories, we are utilizing feature extraction, which is a word embedding technique. We use many sorts of machine learning models to determine the optimal model for our data set for this goal. The diagram below shows each phase of the approach in detail.

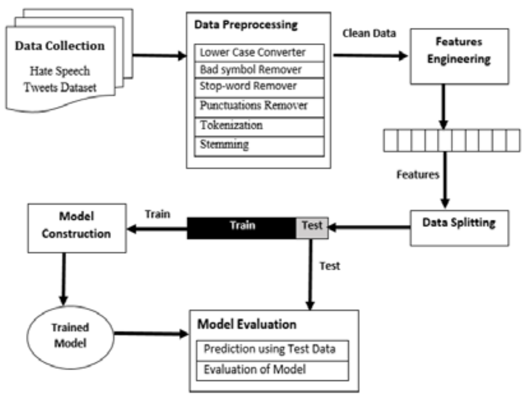


Fig. 1. System Overview
Source: :
https://thesai.org/Publications/ViewPaper?Volume=11&Issue=8&Code=IJACSA&SerialNo=61

3.1 Data Set Description: There are approximately 27,000 rows of tweets in the dataset, which contains three columns, namely: id, label, and tweets. Hate is classified as a '1', while non-hate is classified as a '0.' In the data collection, there were 27,030 tweets.

Class Label	**Number of tweets**	**Maximum Length of Tweet**	**Number of tokens with stop-words**
Hate	**14,741**	**70 words**	**6,382**
Non-Hate	**12,289**	**70 words**	**5,261**

Table 1 : Description of Dataset

3.1.1 Data Collection and Data Set:

For this study, we assembled a dataset of publicly released hate speech tweets. There are two types of tweets in this dataset: non-hate speech and hate speech. The training dataset has 27,030 tweets. Nearly half of the tweets are hate speech, with the rest being non-hate. We're training our model with an online dataset because it has sufficient samples and can be used with whatever model we're working on.

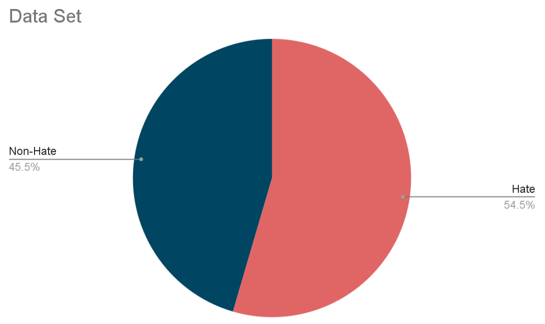


Fig. 2. Class-wise Data Distribution

3.2 Data Preprocessing:

Data preprocessing increases classification results, so it's critical to perform it if we want the best results. As a result, we applied different types of preprocessing approaches to remove noise and non-informative characteristics from Twitter's tweets in our dataset. Our tweet dataset was cleansed of the following elements:

1. URLs are being eliminated.
2. Special characters are being deleted.
3. Corrections to the spelling.
4. Getting rid of the stop words.

Tokenization and stemming were also done on preprocessed tweets. Tokenization breaks down each tweet into tokens or words, which the porter stemmer then breaks down into root forms, such as furious to insult.

3.3 Data Splitting:

After doing preprocessing, we partitioned the preprocessed data in an 80-20 ratio. Using the training data, our model that we used to classify tweets is trained to learn classification rules. The test data is also used to test the classification model. We're dividing the data into two parts so that we can train our model on 80% of it and then test it on the other 20% to see how accurate it is in producing the result.

3.4 Classifications:

We used supervised learning approaches to find hate speech in the English language. We evaluate the performance of all the methods that we are using, i.e., six methods by using the collected dataset: Logistic Regression, Linear SVM, RF Classifier, NB Classifier, CNN, and CNN+LSTM.

3.4 Model Description:

In this study, we use a variety of ML models, such as CNN, NB, SVM, RF, and LR. We measured the accuracy and F1 Score of each model we employed to find one that was the best classifier for our dataset. For classification, we are mainly using CNN+LSTM as our main model.

3.4.1 CNN Model :

To categorize hateful words, we suggest using CNN-based models. The most significant constituent are the convolutional layers,Which extracts features by sliding the filter across the input.Convolution is followed by max pooling to capture the output's most important properties.
To illustrate this idea, consider the example of a comment: "In this pic you are looking ugly and disgusting , you deserve to die.". Each of the words such as ‘looking’, ‘ugly’, ‘disgusting’, ‘deserve’, ‘die’ alone are not always indicative of hate. But combinations such as "looking ugly," " ugly and disgusting," and "deserve to die" can be more indicative features. And for explicit images, we used image processing to check what’s hateful in the image.
Convolutional neural networks (CNNs) has three types of layer: Pooling, Convolution and fully connected layers. Pooling and Convolution layers extract information, while a fully connected layer, integrate those features into final result, such as classification.

Convolutional Layer: This is the Primary layer of the convolutional network. Which extracts features by sliding the filter across the input.Its objective is to find a certain collection of features in the pictures supplied as input.

Pooling Layer: The sizes of the feature maps is lowered using pooling layers. As a result, it lessens the quantity of network computation and the number of parameters that must be learned.

Fully connected layer :The output of the final Pooling Layer serves as the input for the so-called Fully Connected Layer at the end of a CNN.Classification is carried out by fully connected layers using the features that were extracted by the preceding layers.

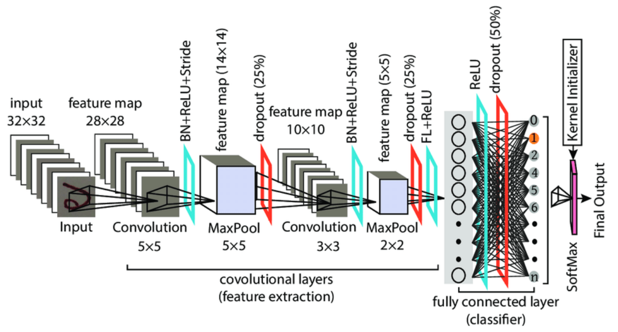


Fig. 2. The Architecture of CNN

Source: https://www.researchgate.net/figure/Proposed-adopted-Convolutional-Neural-Network-CNN-model_fig2_332407214

3.4.6. Long Short Term Memory

To detect online hate, we need a significant volume of labeled data that has been divided into hate and non-hate categories. Several academics developed various techniques for detecting online hate speech in social community forums, all of which were successful. Preprocessing the dataset and using several machine learning models is the typical technique.

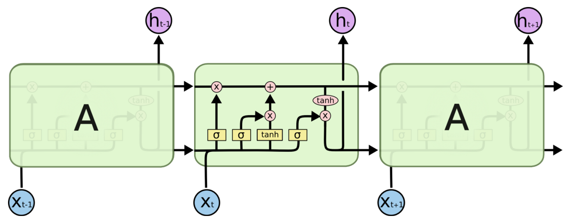


Fig. 7. Architecture Diagram of the LSTM

Source: https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/

LSTM has three gates: input gate, output gate and forget gate.The task of input gate is adding information to the state of the cell and forget gate determines what information must be remembered and what can be forgotten and output gate chooses what output to produce based on the current internal cell state.

4. Proposed Approach

This research provides the most significant contribution by offering feature reduction strategies in conjunction with a combination of deep learning models that include two neural network layers, CNN and LSTM. The suggested technique outperforms traditional deep learning models in terms of prediction accuracy. The CNN-LSTM architecture is used to process the data. The model's first layer, the embedding layer, accepts tweets and converts each word into a 100-pixel vector. Because a tweet can only be 90 characters long, this layer will generate a 90*100 matrix. The output matrix will contain the weights obtained from the matrix multiplication, resulting in a vector for each word. These embedding vectors are used by the CNN layer to retrieve context information. The CNN layer's output is passed into an LSTM layers, which is then passed on to a fully linked dense layer, This produces a single output as the final result. This model is trained and tested on batches of size 128, as shown in Fig. 8.

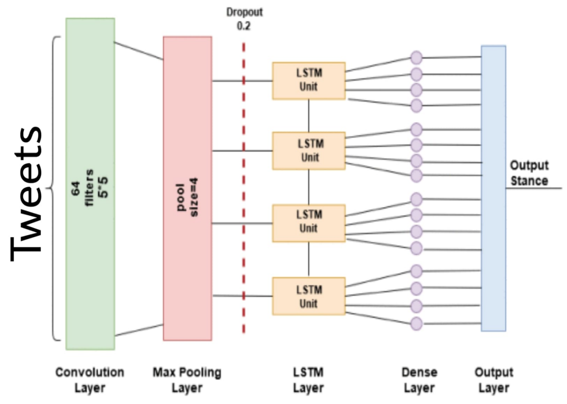


Fig. 8. Proposed model architecture diagram.

Source: https://ieeexplore.ieee.org/document/9178321?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound

4.1 Input and Convolution Layer

The first step is to use the Keras tokenizer to tokenize the text of the tweets. The embedding layer of keras converts tokens into word vectors by the use of word2vec word embeddings. The output from the word embedding layer, that is, a vector of words, is sent to the convolution layer. The convolutional layers are used to extract a set of semantic or structural properties derived from a set of input arrays. Each word vector is received by CNN neurons n Filters of varying sizes can be used to generate a variety of features. There are several filters f, each with a different kernel size. (c*e) is the outcome of applying the function c to each word embedding e. Because the kernel size in our work is 5, a filter of size 64 will produce 5-word combinations. Tables 2 and 3 illustrate the input and output shapes of CNN+LSTM with various settings.

CNN-LSTM

Dropout(0.2)
Conv(5x5,@64)
Dense(3 neuron)
Max Pooling (4x4)
Activation=’sigmoid’
LSTM (150 neurons)

Table 2: Layer Structure of the Proposed Model

Layer(type)	Output Shape	Param #
embedding_15(Embedding)	(None,90,100)	2709100
conv1d_15(Conv1D)	(None, 86, 64)	32064
max_polling1d_11 (MaxPolling1D)	(None, 21, 64)	0
dropout_14 (Droupot)	(None, 21, 64)	0
lstm_11 (LSTM)	(None, 150)	99000
dense_15 (Dense)	(None,3)	153

Table 3 : Model Parameter Structure

4.2ActivationFunction, Max-Pooling, and Dropout

The relu activation function affects every CNN neuron's output. In the first phase of the ReLu activation function, the negative values are converted to zero, which indicates network non-linearity. As there is no negative value in the CNN layer. As a result, the input shape and the output shape of the CNN layer are the same. After the completion of the ReLu function, the value of each neuron is sent into a 1-D max-pooling layer. In the convolution procedure known as maximum pooling, the kernel pulls the highest value possible from the convolution area. It will also assist in lowering the size of the input function to the next level, so overfitting can be avoided. Since the Twitter dataset values are at least 0.2, the output dropout level is equivalent to the given input.

4.3 LSTM

LSTM is the next layer, with 100 units. For our data, We must create a sequence structure that looks like a long chain. while keeping track of prior inputs. Because it has three gates: input gate, output gate and forget gate, the LSTM is the ideal option for this task. These gates determine which information is valuable for categorization and what information can be forgotten based on the dropout value. The cell memory block, which is essential for prediction, saves previous input. Each and Every unit in the dense layer is connected to all 100 units in the LSTM layer's output.

4.4 Dense

The dense layer, which is the final layer of the proposed model, is highly connected with a single output. The batch size is 128 and the iteration number is 10.

5. Classifier Evaluation

We must ensure the integrity of our outcomes. As a result, we conducted a quantitative analysis of the frequency of data collection, data release mode, and data types. The obtained data is meticulously annotated before being entered into the constructed model for prediction. So we used an evaluation measure to examine the data and outcomes to see how well the model worked, how skewed the results were, and how generalizable our findings were. We employed the evaluation metric we devised throughout the trial.

The idea of positive and negative affects all of the numbers we assess for accuracy, precision, recall, and F- scores. Positive speech is defined as hate speech, whereas negative speech is defined as speech that is not hateful. Figure 8 shows the definitions of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

1) **Recall** :Percentage of actual consent is expected to be positive.

Recall = TP/(TP+FN)

2) **Precision:** It tests how many outcomes actually are positive outcomes for all well-predicted outcomes.

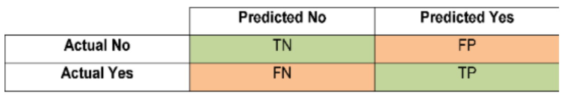
Precision = TP/(TP+FP)

3) **Accuracy**: It is calculated by dividing the total number of relevant predictions by all the predictions.

Accuracy= (TP + TN) /(TP + TN + FP + FN)

4) **F1 Score**:It is the precision and recall harmonic mean. It accounts for both false positives and false negatives.

F1 Score=2*(Precision*Recall)/(Recall+ Precision)



	Predicted No	Predicted Yes
Actual No	TN	FP
Actual Yes	FN	TP

Fig. 8. Confusion Matrix

6. Result :

This section will be evaluated by comparing six classifier models that have been used by us in the research. These models used different feature sets

to regard the evaluation matrix, i.e., precision accuracy,F1 score and recall.
Table 3 is showing the outcomes of the experiments we conducted using various methodologies. In our research, we performed multiple classifiers and selected the top three classifiers among all classifiers.Random Forest, CNN, and CNN+LSTM are the selected classifiers.
The approaches we are using as our main models produce good outcomes. "CNN+ Long Short Term Memory" is the best of all approaches on our given dataset.The performance of various machine learning methods, including Random Forest, Naive Bayes, Logistic Regression and Random Forest was compared to that of the proposed CNN+LSTM model.The same data set and split training and testing techniques were applied.
In Table 4, the hyper-parameters of our main model, CNN+LSTM, are detailed.

Parameter Description	**Values**
Maximum length of tweet	90
Size of filters	5
Number of filters	64
Pooling size	4
Activation function	Relu, Sigmoid
Number of convolution layers	1
Learning rate	0.01
Batch size	128
Loss function	binary_crossentropy
Optimizer	Adam
Epoch	10

Table 4: Hyperparameters used in the proposed approach

The Recall, Precision and f1-score values for the CNN+LSTM model are shown in Table 5.

	Precision	**Recall**	**F1-Score**	**Support**
Hate	0.92	0.92	0.92	2932
Non-Hate	0.90	0.91	0.91	2475

Table 5: Accuracy values for CNN+LSTM Classifier

Table 6 compares the accuracy, precision, F1 score, and recall of many different machine learning models to our CNN+LSTM model in great detail.

Model	**Accuracy**	**Precision**	**F1_Score**	**Recall**
SVM	.61	.60	.71	.87
CNN+LSTM	.92	.94	.92	.90
Logistic Regression	.62	.61	.70	.83
Naive Bayes	.57	.56	.71	.96
Random Forest	.77	.81	.78	.76
CNN	.91	.92	.92	.91

Table 6 : The results of several classification algorithms and approaches for hate speech detection tasks are shown.

Figure 9 depicts a graphical representation of the evaluation matrix of several classifiers. Each classifier is represented by a different colour.

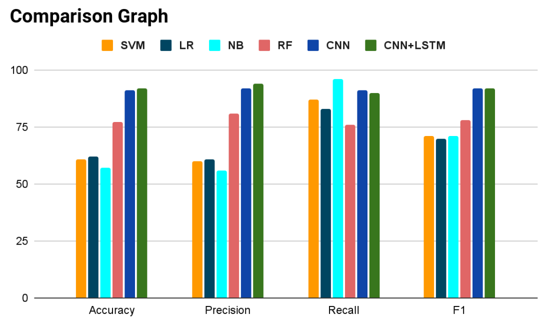


Fig. 9. Comparison graph of all classifiers

However, the CNN+LSTM confusion matrix in fig. 10 shows that the model successfully predicted 2248 out of 2475 (non-hate class) and 2694 out of 2887 (hate class) predictions.

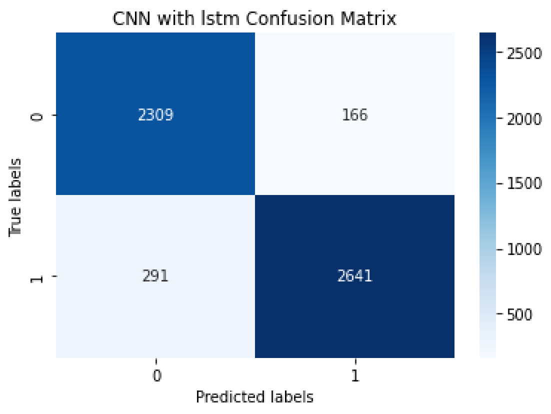


Fig. 10. Confusion Matrix of CNN+LSTM

7. Conclusion :

Hate speech has become a major concern, and a computerized hate speech detection system is one of the solutions that will assist in the solution of this major problem.Machine learning classifiers have been proposed as a method for detecting hate speech and offensive language among different online platforms that are prone to face hate . The project's purpose was to develop an automated approach for eliminating social hatred on social media and in online communities. In our model evaluation, our CNN+LSTM model outperformed other existing approaches. When we tested the model using test data, we obtained 92.1% accuracy. To address this issue, more examples of offensive language that do not contain hostile phrases can be found here. To improve the outcomes even more, increase precision of the hateful class and recall for the offensive class. The model's lack of consideration for the negative word in a phrase is one of its drawbacks. This region can be improved by incorporating linguistic components. Furthermore, the performance of Naive Bayes was the worst. The result of our research is important because it will be used by future researchers to evaluate and compare their findings. This study is also scientifically noteworthy since it provides experimental data for automated text categorization in the form of several scientific measurements.
There are two significant limitations to our research. First, our model is lacking in terms of predicting accuracy in real time. Finally, it just divides hate speech into two groups and is unable to identify the message's intensity. As a consequence, the suggested machine learning model, which is also useful for predicting how strong hateful messages will be. It will be developed in the future.
Hate speech identification is a difficult task for academics. The text in the social media post is followed by photographs, which are subsequently followed by text. Even in text, code-mixed languages are used.
To make the algorithm more robust, future work on hate speech identification could include multilingual examples in several languages as well as multi-modal types of social media posts.

8. Reference :

1. Davidson T, Warmsley D, Macy M, Weber I. Automated hate speech detection and the problem of offensive language. In: Pro- ceedings of the 11th International AAAI Conference on Web and Social Media, ICWSM '17, pp. 512–515; 2017.
2. Djuric N, Zhou J, Morris R, Grbovic M, Radosavljevic V, Bha- midipati N. Hate speech detection with comment embeddings. In: Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, p. 29–30. Association for Computing Machinery, New York.
3. Wang B, Ding Y, Liu S, Zhou X. Ynu_wb at HASOC 2019: Ordered neurons LSTM with attention for identifying hate speech and offensive language. In: Mehta, P, Rosso, P, Majumder P, Mitra M (eds.) Working Notes of FIRE 2019 - Forum for Infor- mation Retrieval Evaluation, Kolkata, India, December 12-15, 2019, CEUR Workshop Proceedings, vol. 2517, pp. 191–198. CEUR-WS.org; 2019.
4. Yuan S, Wu X, Xiang Y. A two phase deep learning model for identifying discrimination from tweets. In: Pitoura E, Maabout S, Koutrika G, Marian A, Tanca L, Manolescu I, Stefanidis K (eds.) Proc. EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15–16, 2016, pp. 696–697. OpenPro- ceedings.org; 2016. https ://doi.org/10.5441/002/edbt.2016.92.
5. S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural Comput. 9, pp. 1735-1780, 1997.
6. Zhang Z, Robinson D, Tepper J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In: European Semantic Web Conference. Springer; 2018. p. 745–760.
7. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. in a European conference on machine learning. 1998. Springer.
8. Wenando, F.A., T.B. Adji, and I. Ardiyanto, Text classification to determine the level of understanding of students in the process of activating prior knowledge. Advanced Science Letters, 2017.
9. Seliya, N., T.M. Khoshgoftaar, and J. Van Hulse.Examining the relationship between the performance indicators of the classifier. 200921st IEEE international conference on tools with artificial intelligence. 2009. IEEE.

10. Chaudhari, U.V. and M. Picheny, Matching criteria for vocabulary-independent search. IEEE Transactions on Audio, Speech, and Language Processing, 2012.
11. Del Vigna et al. have classified the Hate Speech of Facebook comments into fine-

grained classes. They have used two different approaches with SVM and LSTM to identify the Hate comments. experiments were conducted on HatebaseTwitter, Stanfront and TRAC datasets.

12. LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vi- sion. Circuits and Systems (ISCAS), Proceed- ings of 2010 IEEE International Symposium on, 253-256.
13. Chand N, Mishra P, Krishna CR, Pilli ES, Govil MC. A comparative analysis of SVM and its stacking with other classification algorithms for intrusion detection. In: 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring). IEEE.
14. Z. Mossie, "Social media dark side content detection using transfer learning emphasis on hate and conflict," in Companion Proceedings of the Web Conference 2020.
15. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
16. Severyn, A.; Moschitti, A. Twitter Sentiment Analysis with Deep Convolutional Neural Hema Krishnan,M. Sudheep Elayidom,T. Santhanakrishnan,Emotion Detection of Tweets using Naïve Bayes Classifier,International Journal of Engineering Technology Science and Research,Volume 4, Issue 11,November 2017Networks.In Proceedings of the 38th InternationalACMSIGIR Conference on Research and Development in Information Retrieval—SIGIR '15, Association for Computing Machinery (ACM), Santiago, Chile, 9–13 August 2015; pp. 959–962.
17. Thomas Davidson, Dana Warmsley, Michael Macy,and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language.In Proceedings of ICWSM.
18. Schmidt, A., & Wiegand, M. (2017). A Survey on Hate Speech Detection using Natural Language Processing. Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, 1–10. https://doi.org/10.18653/v1/W17-1101
19. Mandl T, Modha S, Patel D, Dave M, Mandlia C, Patel A. Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages). In: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation; 2019.
20. Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. ArXiv:1703.04009 [Cs]. Retrieved from http://arxiv.org/abs/1703.04009
21. Zhang Z, Robinson D, Tepper J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In: European Semantic Web Conference. Springer; 2018. p. 745–760.
22. Hema Krishnan,M. Sudheep Elayidom,T. Santhanakrishnan,Emotion Detection of Tweets using Naïve Bayes Classifier,International Journal of Engineering Technology Science and Research,Volume 4, Issue 11,November 2017.