Global Advanced Research Journal of Educational Research and Review (ISSN: 2315-5132) Vol. 11(8) pp. 325-336, November, 2023
Available online http://garj.org/garjerr
Copyright © 2023 Global Advanced Research Journals

Full Length Research Paper

Hate Speech Detection in Twitter: Natural Language Processing Exploration

Kelly Ochuko Egode[1]* Linda Oraegbunam[2] Adedamola Samuel Oyatunji[3] and Ojore Solomon Akwue[4]

[1]MSc Artificial Intelligence and Data Science, University of Hull, UK,
[2]MSc Applied Artificial Intelligence and Data Analytics, University of Bradford, UK,
[3]MSc Computer Science, Ulster University, UK.
[4]MSc Big Data and Business Intelligence, School of Computer Science, The Universidad International Isabel I de Castilla, Barcelona, Spain.

Accepted 15 November, 2023

The proliferation of social media platforms, particularly Twitter, has led to a significant rise in hate speech propagation, posing serious challenges to information dissemination and societal harmony. This paper proposes a novel approach leveraging state-of-the-art natural language processing (NLP) and deep learning techniques to automatically detect and prevent hate speech in real-time on Twitter. By employing machine learning algorithms and deep learning models such as Simple Recurrent Neural Network (SimpleRNN), Long Short-Term Memory Network (LSTM), and Gated Recurrent Unit (GRU), this study aims to surpass existing methods in hate speech classification. Utilising a dataset from Kaggle, the research conducts sentiment analysis and hate speech detection, addressing challenges such as data pre-processing and class imbalance. Various resampling techniques and model architectures are explored to optimise performance metrics including accuracy, precision, recall, F1-score, and area under the precision-recall curve (pr_auc score). The results indicated that while the Naïve Bayes algorithm achieved high precision, deep learning models, particularly best-performing LSTM architecture 2 - include accuracy: 0.950, precision: 0.633, recall: 0.674, F1-score: 0.653, pr_auc score: 0.622, and roc_auc score: 0.870, exhibited promising performance, albeit slightly below baseline expectations. Challenges such as limited training data and imbalanced datasets were identified as key factors impacting model performance. In conclusion, this research underscores the feasibility of leveraging NLP and deep learning for hate speech detection on social media platforms like Twitter. Future work entails exploring advanced models like BERT and ensemble methods to further enhance classification accuracy and mitigate the impact of data scarcity and imbalance.

Keywords: Machine Learning, Deep Learning, Recurrent Neural Network, Natural Language Processing, Hate Speech.

INTRODUCTION

In recent years, the proliferation of social media platform users has posed a threat to information dissemination. The social media platform, Twitter currently has over 217 million everyday active users that are currently monetized globally and about 500 million tweets sent per day (Omnicore, 2022). We have seen abuse of Twitter users

*Corresponding Author E-mail: kelly.e@omfeonix.com

through the use of nasty race, racist, or sexist statements, politics, religion, tribe, and ethnic genealogy to wreak mayhem around the world on multiple occasions. As a result, employing state-of-the-art natural language processing and deep learning technologies, we would be able to monitor and prohibit the use of Twitter as a source of hate speech propagation.

Furthermore, studies had shown that natural language processing in the detection of hate speech, utilises linguistic features such as parts-of-speech and bag of words techniques (Greevy and Smeaton, 2004). Also, the use of machine learning algorithms and statistical models to automate the hate-speech identification process online. This is in order to avoid the financial, material, and human costs associated with nasty communications.

As a result, combating hate speech online at its source by recognizing it automatically and in real-time using NLP deep learning sentiment analysis. This would prevent the tweet from being made public and avert havoc.

In this work, I will use a quantitative and sentiment analysis of tweets dataset provided by (Kaggle, 2020) regarding Twitter Sentiment Analysis by Analytics Vidya on Hate Speech, and employ machine learning and deep learning techniques to discover hate speech tweets.

Background

Gaydhani (2018) devised a machine learning model which can differentiate between two identified categories of hate language, namely offensive speech and hate speech. By using publicly available Twitter datasets, they trained their classifier models using n-gram and term frequency-inverse document frequency (TFIDF) as features and evaluated its metric scores. They performed comparative analysis of the results obtained using Logistic Regression, Naive Bayes and Support Vector Machines as classifier models. It was revealed that Logistic Regression performs better among the three models for n-gram and TFIDF features after tuning the hyperparameters to identify hate speech classes.

In another study, an ensemble Recurrent Neural Network (RNN) classifiers was developed using a variety of user-related features, such as a user's proclivity for racism or sexism, upon evaluation with 16k tweets, it shows the capacity to distinguish racism and sexism messages from normal text successfully (Pitsilis et al., 2018).

A killer natural language processing optimization ensemble deep learning approach (KNLPEDNN) was introduced by (Al-Makhadmeh and Tolba Amr, 2020). Their aim was for an effective learning process of detecting hate speech on social media websites; by classifying the text into neutral, offensive, and hate language. They were able to achieve minimum deviations mean square error of 0.019, cross entropy loss of 0.015, logarithmic loss of L-0.0238 and 98.71% accuracy. Further studies by (Zagidullina et al., 2021) had demonstrated BERT model capabilities on hate speech prevention by fine-tuning and testing it with twitter datasets. An improved precision ranging from 64% to 90% and acceptable recall levels in the low 60% were attained. However, in this project, I would be focusing on improving these accuracy metrics attained using other deep learning techniques like Simple Recurrent Neural Network (SimpleRNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

There are instances of high recall over precision in prediction metrics as shown by (Mandl et al., 2019; Zagidullina et al., 2021) reporting a weighted F1 score in the range of 70 percent to 80 percent, implying that precision is worse off than recall. It is vital we improve on the precision metrics with regards to the critical level of predicting the hate speech tweets real time.

Objective

The main goal of this research question is:

- To create a novel method for hate-speech classification that outperforms and improves the current state-of-the-art methods on the basis of performance and accuracy.

METHODOLOGY

The methodology framework Figure 1 was adopted, using the machine learning Naïve Bayes classifier and deep learning-based language models of recurrent neural network (RNN) state of the art types such as Simple Recurrent Neural Network (SimpleRNN); Long Short-Term Memory Network (LSTM) (Hochreiter S and Schmidhuber J 1997); and Gated Recurrent Unit (GRU) (Chung et al., 2014).

The twitter dataset has 31,962 tweets currently labelled. As a result, the sentiment of the tweets would be determined using a supervised machine learning method called Naive Bayes classifier (Pedregosa et al., 2011). The Naive Bayes algorithm was adopted due to its inductive learning methods and performance benefit. It requires very few training datasets to make accurate predictions.

To discover the best model, the SimpleRNN, LSTM, and GRU models would be compared. These are novel deep learning techniques used for sentiment classification. The impact of tweet emotions, word clouds, and hashtags on Hate Speech are then investigated.

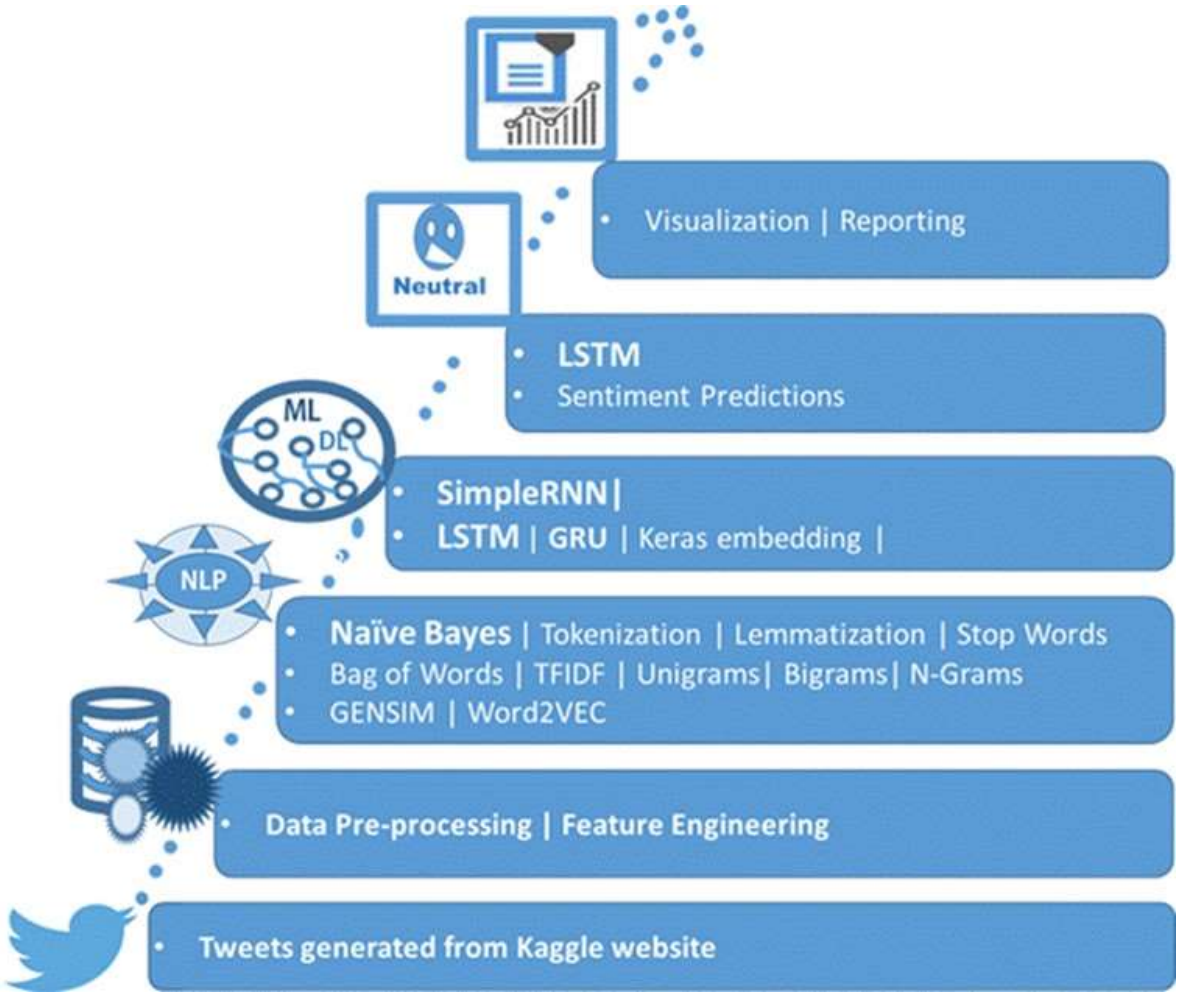

Figure 1. Framework for deployment of Sentiment Analysis of Hate Speech Tweets.

Data Collection

The dataset was collected from (Kaggle, 2020). Figure 2 shows the class imbalance data set is about 93% for non-hate and 7% for hateful comments and the same ratio for the test dataset. It should be noted that unnecessary extra information was observed to be present in the Twitter data acquired in this manner. Thus, increasing the difficulty of detecting hate speech on Twitter.

Data Preprocessing and Analysis

The need for clean tweet texts cannot be overemphasised in ensuring accurate and efficient information gathering from the texts. As a result, noise in the datasets was reduced by data preprocessing (Van Den Broeck et al., 2005; Oyebode et al., 2021). The following processes were undertaken:

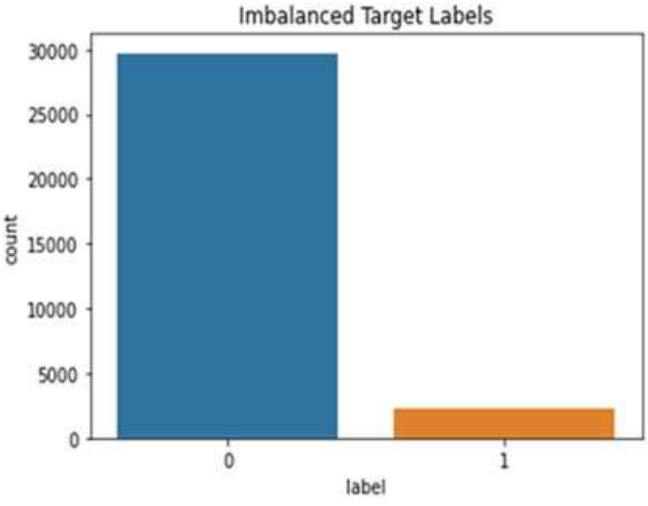

Figure 2. Class Imbalance Dataset.

label	tweet
0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
0	bihday your majesty
0	#model i love u take with u all the time in urð□□±!!! ð□□□ð□□□ð□□□ð□□□ð□□□ð□□¦ð□□¦ð□□¦
0	factsguide: society now #motivation

0	when father dysfunctional selfish drags kids into dysfunction #run
1	thanks #lyft credit cause they offer wheelchair vans #disapointed #getthanked
2	bihday your majesty
3	#model love take with time
4	factsguide society #motivation
5	huge fare talking before they leave chaos disputes when they there #allshowandnogo
6	camping tomorrow danny
7	next school year year exams think about that #school #exams #hate #imagine #actorslife #revolutionschool #girl
8	love land #allin #cavs #champions #cleveland #clevelandcavaliers
9	welcome here

Figure 3. Tweet Cleaning Pre-processing.

● **Removing the Twitter Handles, Punctuations, Numbers, Short words and Special Characters** - The Twitter dataset is cleansed of unwanted characters (Dey et al., 2020). I merged both the train and test datasets for ease of cleaning the Tweets that contain various unwelcome symbols, such as @, which make it more difficult to detect hate speech. Take this Tweet for example ***#model i hate u all the time in urÃ°ÂŸÂ±!!! Ã°ÂŸÂ˜Â****, and* ***@user #cookies***. Using the user defined functions, the "@user" was removed from the Tweet. Further analysis ensures the sub() method replaces a pattern with an empty string, or `` ``, after it is matched. Thus, Tweet's extraneous text, results to "# I hate you.". Thereafter, Twitter hashtags are deconstructed, leaving " i love you" as the final version of the Tweet. Then the texts are converted to lower case to ensure upper case are not seen as separate tokens. The remove_stopwords function was used to remove stop words, while make_bigrams and make_trigrams functions helped in deriving the bigrams and trigrams in the corpus. The map and lambda functions were equally used in the cleaning process Figure 3.

● **Tokenization** is the next step of converting my text from a list of sentences to a list of words. This ensures the input to my model would be the individual group of words instead of the entire sentences. This aids analysis and visualisation to get insight into the count of hateful words, identify classes of hateful terms, and then normalise them.

● **Normalisation** process commenced after the unnecessary characters are removed, the root words in the Tweet are derived using a lemmatization technique (Raza et al., 2019). Lemmatizing was chosen over stemming because it is an effective NLP strategy with a trade-off between speed and accuracy, reducing the number of tokens and sparsity of my datasets. The words are processed based on their root or origin. I used nouns, adjectives, verbs and adverbs in a lookup table to manage incoming text that are converted to their root word. Different versions of the same word are grouped together in lemmatization e.g "jump", "jumps", "jumped", "jumping" are normalised to jump. Therefore, the pre-processed tweets data would produce abstraction representations that are easier to apply machine learning Naïve Bayes techniques.

● Then, using word-based frequency, tweets are turned into vectors and used wordCloud to visualise the commons positive words like "Happy"," Love" and "Friend" and negative words – "Hate", "Black" and "Racist" as shown in Figure 4, had revealed hate speech been expressed by some persons.

● Understanding the impact of the hashtags in the text revealed the current trend amongst Twitter users at the period. We can see from Figure 5 that love, positive, healthy and smile are positive hashtags while trump, politics, allahsoil and libtard_libtard are negative hashtags predominantly used.

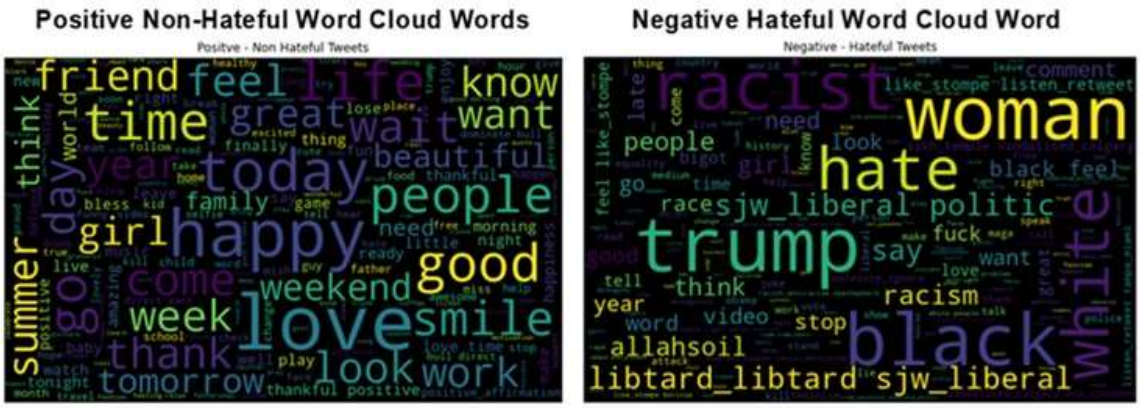

Figure 4. WordCloud Showing Positive and Negative Words most expressed.

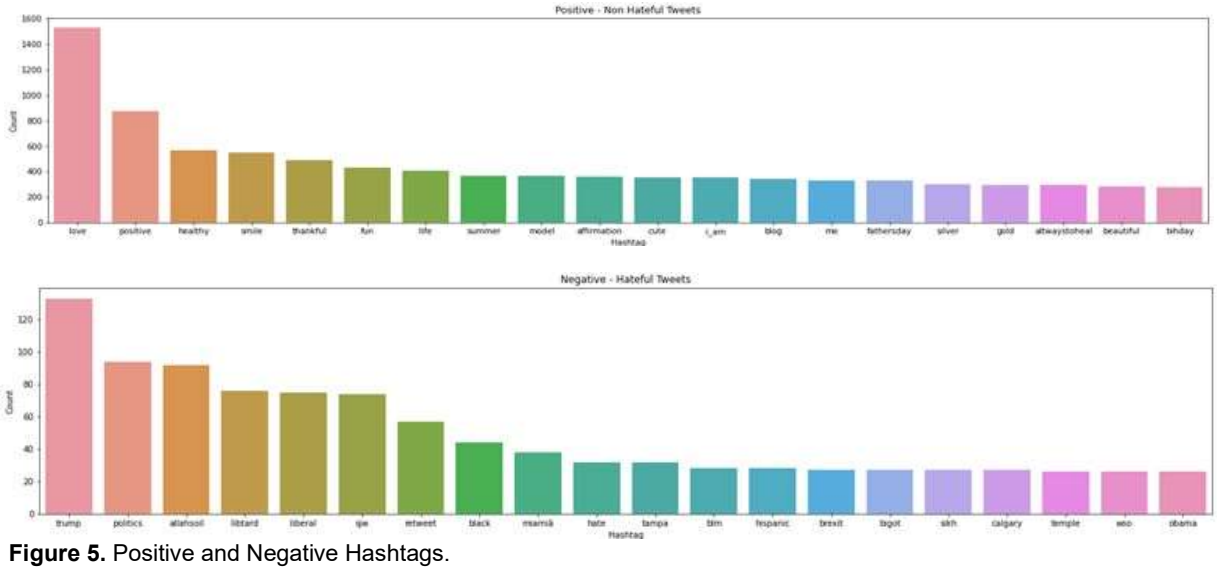

Figure 5. Positive and Negative Hashtags.

Sentiment Analysis Model

To enable the model to use the data for effective prediction, data collection, pre-processing, analysis, and feature engineering are required. As a result, Naive Bayes would require cleaning, but the other models would not.

Naïve Bayes Model

This is a statistical technique based on Bayes theorem and independent feature assumption. The tweets in words and sentences are translated into numbers (Rustam et al., 2021a). The Multinomial Naïve Bayes model is used; its inputs are the translated data to perform sentiment analysis on tweets. The hyper tuned model was used to establish a baseline for the framework. The probability inferences on the target class were obtained based on the mathematical expression shown in equations 1 - 2 below.

$$(X) = \frac{P(y) * P(y)}{P(X)} \quad (1)$$

$$y = argmax_y P(y) \prod_{i=1}^{n} \quad P(y) \quad (2)$$

SimpeRNN Model

The Simple Recurrent Neural Network (SimpleRNN) deep learning model deals with text data, time-series data, and other sequential data. It ensures the information cycles to the output layer using the previous hidden layer and the prior levels. It deals with the variable-length sequence as shown in Figure 6. This is achieved by a short-term memory; unlike an artificial neural network that considers only the immediate hidden layer.

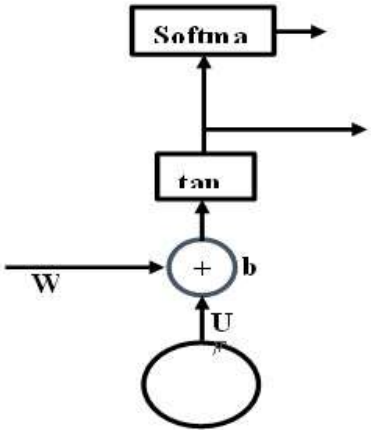

Figure 6. Simple RNN Model Architecture

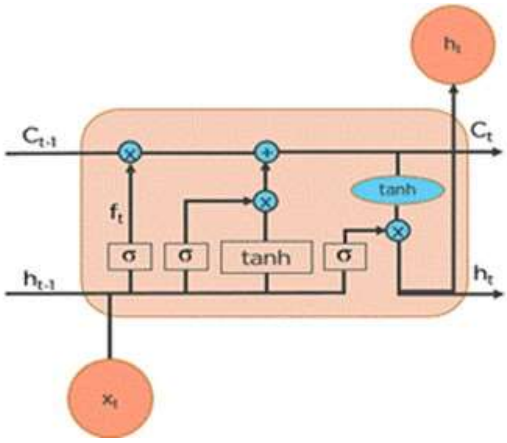

Figure 7, LSTM Model Architecture (Zhan et al., 2021)

The SimpleRNN manages the variable-length sequence X as shown in equation 3 by utilising a recurrent hidden state, the activation of which is dependent upon the previous activation at each subsequent iteration. The recurrent hidden states of the model are updated as given in equation 4, with the implementation done using equation 6. The sequence probability is broken down as shown in equation 7. It is finalised as a conditional probability distribution in equation 8.

$$X = (X_1, X_2, \ldots\ldots, X_T) \tag{3}$$

$$h_t = \{0, \quad t = 0\ \emptyset(h_{t-1}, x_t), \qquad otherwise \tag{4}$$

$$y = (y_1, y_2, \ldots\ldots, y_T) \tag{5}$$

$$h_t = g(Wx_t + Uh_{t-1}) \tag{6}$$

$$p(x_1, \ldots\ldots, x_T = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \ldots . p(x_T \mid x_1, \ldots x_{T-1}) \tag{7}$$

$$p(x_t \mid x_1, \ldots\ldots, x_{t-1}) = g(h_t) \tag{8}$$

LSTM Model

The Long Short-Term Memory (LSTM) is a recurrent neural network (RNN); deep learning model that was created to overcome the problem of gradient disappearance in RNNs owing to backpropagation (Chuluunsaikhan et al., 2020). The cell states of LSTM which are controlled by the three gates (input, forget and output) helps to resolve the problem of RNN Figure 7.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f \tag{9}$$

Forget gate, equation 9 that decides which information to delete that is not important from

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i$$
$$\tilde{C}_t = tanh\,(W_c\,.\,[h_{t-1}, x_t] + b_c) \tag{10}$$

Input gate, which decides which information to let through based on its significance in the current time step that is not important from previous time step as shown in equation 10

$$o_t = \sigma(W_o\,[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh(C_t) \tag{11}$$

The output gate, equation 11 determines how the information brought in from the input gate affects the output in the current time-step.

The LSTM model used the Keras word embedding to convert the tweets to vector sequence and pass the features through LSTM Keras Tensorflow for sentiment classification. Further hyper parameter tuning was carried out to get the best accuracy model (Rustam et al., 2021b).

GRU Model

The Gated Recurrent Unit is an RNN version that has a simplified neural architecture, it simply uses the hidden state to ensure that the model is trained quickly. As shown in Figure 8, GRU uses the update gate, equation 13 and reset gate, equation 12 to address the issue of gradient diminishing. These gates can be trained to retain information from the candidate hidden state, equation 14 and determine what information is allowed through to the output. As a result, the update gate ensures the transfer of important information from the new hidden state, equation 15 along the event chain to improve its forecasts.

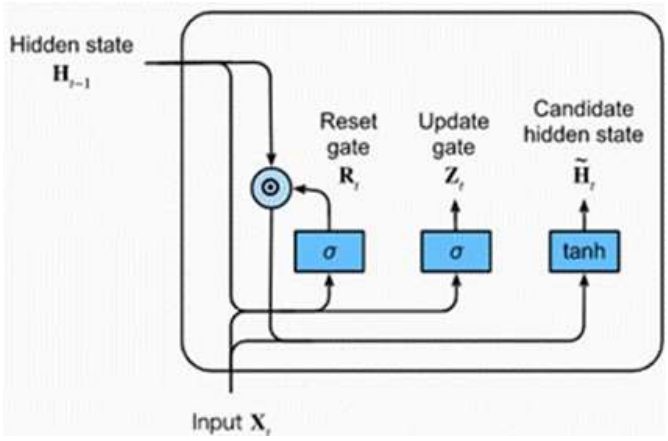

Figure 8. GRU Model Architecture (Zhang et al., 2021)

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \quad (12)$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \quad (13)$$

$$\tilde{H}_t = tanh\,(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h) \quad (14)$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \quad (15)$$

Experiments

The project experimental setup consists of collecting Kaggle hate speech data by downloading the csv files. Thereafter, they are read as a source dataset using python module pandas. Different user defined functions were developed to enable proper pre-processing of the tweets. Then, tokenization, lemmatization, WordCloud, N-grams and hashtags were generated to get a better understanding of hate speech sentiments.

This project applies Naïve Bayes Machine Learning, and Deep Learning recurrent neural network methods to hate speech twitter datasets.

For the machine learning Naïve Bayes model, using the SKlearn train_test_split method; I split my dataset into 80:20 ratio for training and testing. I then vectorised the datasets using Term Frequency Inverse Document Frequency TF-IDF. I thereafter carried out sentiment analysis on the training and testing data using Naïve Bayes algorithm with my alpha = 0.01. However, my dependent variable label is an imbalanced dataset. So, I employed the SKlearn imblearn library to carry out resampling methods – SMOTE (synthetic minority oversampling technique), NearMiss and SMOTETomek to get a balanced label dataset.

The Deep learning methods entails SimpleRNN, LSTM and GRU models using the Tensorflow Keras framework. There are four different architectures as shown in Table 1 below trained on the three neural network models. Thereafter, Keras tuner hyper parameter tuning was used to tune the three models and their best architecture retrained as shown in Figure 9. The imbalanced target dataset was addressed using the Keras class_weights.

The main aim of this study is to evaluate the classification performance on Hate speech twitter datasets using the best Naïve Bayes performance as a baseline. The following metrics were chosen - Accuracy; Confusion matrix; Precision: identify the correctness of classification; Recall: shows the number of positive cases correctly identified over total positive; F1-score: balances precision and recall which is reliable for imbalance class; Precision-Recall curve: calibrates the probability threshold; and Average Precision score: computes the average precision (AP) from prediction scores between 0 and 1 for imbalanced class.

RESULTS

Present your results, ideally supported with tables and / or graphs. Discuss them, how do they compare with baselines? Did you meet your objectives? If not, why not? Did you find anything interesting, unexpected? Anything worth investigating further?

The highlighted results in Tables 1, 3–7 represent the highest achieved accuracy as compared to baseline results.

The Naïve Bayes Algorithm best performance architecture was chosen as the baseline model for the project. Table 1 shows that architecture without resampling dataset has the highest precision: 0.995 and lowest recall: 0.440. The NEARMISS resampling with lowest accuracy: 0.845, precision: 0.303, F1-score: 0.459, pr_auc score: 0.800 and highest recall: 0.940, roc_auc: 0.889. However, the best architecture model with SMOTE resampling gave the following best metrics: F1-score: 0.699, precision: 0.636, recall: 0.777, pr_auc score: 0.807, roc_auc score: 0.872.

Table 3 shows that the SimpleRNN model architecture with Keras tuner compared to the other SimpleRNN architectures had the highest accuracy: 0.947, precision: 0.625, F1-score: 0.611, pr_auc score: 0.611. While architecture 1 has the lowest accuracy: 0.922, precision: 0.461, F1-score 0.542, pr_auc score: 0.509. However, SimpleRNN architecture 2 was chosen as the best model due to its high recall: 0.654, precision: 0.540, F1-score: 0.591.

The LSTM best Keras Tuner architecture Figure 9, had the poorest performance metrics accuracy: 0.939, precision: 0.553, F1-score: 0.613, pr_auc score: 0.534 compared to other LSTM architectures. While LSTM architecture 1 had the highest recall: 0.723. However, LSTM architecture 2 was chosen as the best model with accuracy: 0.950, precision: 0.633, recall: 0.674, F1-score:

Table 1. Naive Bayes Algorithm Model.

S/n	Model Name	Architecture	Accuracy	Precision	Recall	F1 Score	PR AUC	ROC AUC
1	Naïve Bayes	W/o Sampling	0.961	0.995	0.440	0.610	0.801	0.720
2	Naïve Bayes	W/o Sampling + Hyperparameter Tuning	0.962	0.968	0.480	0.642	0.815	0.739
3	Naïve Bayes	W/SMOTE Oversampling	0.953	0.636	0.777	0.699	0.807	0.872
4	Naïve Bayes	W/NEARMISS Under sampling	0.845	0.303	0.940	0.459	0.800	0.889
5	Naïve Bayes	W/SMOTETOMEK Sampling	0.952	0.622	0.790	0.696	0.810	0.877

Table 2. Architectures Configuration for Three Deep Learning Algorithms.

S/n	Model Names	Model Configuration / Parameters – (Model Architecture)
1	SimpleRNN, LSTM & GRU	Embedding Layer with 16-dimension, vocab size – 10000 & max length – 120; Model Layer(64), Dropout and Recurrent Dropout = 0.4, Dense(32) ReLu, Dropout and Recurrent Dropout = 0.4, Dense(10) ReLu, Dense(1) with Sigmoid, bias_initializer=output_bias
2	SimpleRNN, LSTM & GRU	Embedding Layer with 16-dimension, vocab size – 10000 & max length – 120; Model Layer(32), Dense(10) ReLu, Dense(1) with Sigmoid, bias_initializer=output_bias
3	SimpleRNN, LSTM & GRU	Embedding Layer with 16-dimension, vocab size – 10000 & max length – 120; Model Layer(32), Dense(3) kernel_initializer='he_uniform', Dense(10) ReLu, Dense(1) with Sigmoid, bias_initializer=output_bias
4	SimpleRNN, LSTM & GRU + Keras Tuner Random Search	Embedding Layer with 16-dimension, vocab size – 10000 & max length – 120; Model Layer(32), Dense(units = hp.Int("units", min_value=16, max_value = 512, step=16), activation= hp.Choice("activation", ['relu','tanh']), Dropout and Recurrent Dropout = 0.25, Dense(1) with Sigmoid, learning_rate = hp.Float("lr", min_value =1e-4, max_value=1e-2, sampling = "log")

0.653, pr_auc score: 0.622 and roc_auc score: 0.870 (Table 4).

In Table 5 the GRU architecture 2 had the highest recall: 0.725 and lowest accuracy: 0.929, precision: 0.497 when compared to the other GRU architectures. However, the GRU best Keras tuner architecture Figure 9, had the best performance metrics accuracy: 0.942, precision: 0.574, recall: 0.672, F1-score: 0.619, pr_auc score: 0.552.

The best deep learning model as shown in Table 6 is LSTM architecture 2. When compared to the baseline model, it is observed that its accuracy, precision and roc_auc score are 0.3% below baseline result. While the recall is 15.3% below, F1-score is 7% below, and pr_auc score 29.8% below. Further review of the weighted average Table 7 shows that the recall target of 95% was met while precision and F1-score were 1.1% below

Table 3. Simple Recurrent Neural Network Models Performance Metrics Achieved on Hate Speech Data.

S/n	Model Name	Architecture	Accuracy	Precision	Recall	F1 Score	PR_AUC	ROC AUC
1	SimpleRNN	1	0.922	0.461	0.658	0.542	0.509	0.862
2	SimpleRNN	2	0.937	0.540	0.654	0.591	0.543	0.862
3	SimpleRNN	3	0.937	0.542	0.629	0.583	0.566	0.888
4	SimpleRNN	Keras Tuner	0.947	0.625	0.598	0.611	0.567	0.839

Table 4. Long Short -Term Memory Models Performance Metrics Achieved on Hate Speech Data.

S/n	Model Name	Architecture	Accuracy	Precision	Recall	F1 Score	PR_AUC	ROC AUC
1	LSTM	1	0.942	0.568	0.723	0.637	0.586	0.861
2	LSTM	2	0.950	0.633	0.674	0.653	0.622	0.870
3	LSTM	3	0.947	0.610	0.692	0.649	0.561	0.852
4	LSTM	Keras Tuner	0.939	0.553	0.688	0.613	0.534	0.850

Table 5. Gated Recurrent Unit Models Performance Metrics Achieved on Hate Speech Data.

S/n	Model Name	Architecture	Accuracy	Precision	Recall	F1 Score	PR_AUC	ROC AUC
1	GRU	1	0.941	0.562	0.685	0.618	0.541	0.848
2	GRU	2	0.929	0.497	0.725	0.590	0.526	0.865
3	GRU	3	0.931	0.508	0.710	0.592	0.523	0.857
4	GRU	Keras Tuner	0.942	0.574	0.672	0.619	0.552	0.857

Table 6. Comparison of Best Deep Learning Models with Naive Bayes Baseline Performance Metrics Achieved on Hate Speech Data.

S/n	Model Name	Architecture	Accuracy	Precision	Recall	F1 Score	PR AUC	ROC AUC
1	Naive Bayes	SMOTE Oversampling	0.953	0.636	0.777	0.699	0.807	0.872
2	SimpleRNN	2	0.937	0.540	0.654	0.591	0.543	0.862
3	LSTM	2	0.950	0.633	0.674	0.653	0.622	0.870
4	GRU	Keras Tuner	0.942	0.574	0.672	0.619	0.552	0.857

baseline result of 96%. However, our weighted average performed better than the weighted average reported by (Mandl et al., 2019; Zagidullina et al., 2021) by 11% across precision, recall and F1-score.

The objective of the project of getting higher metrics was not met. This is due to lack of training dataset and the nature of the imbalance dataset. We observed that the Naïve Bayes algorithm could easily achieve very high metrics unlike RNN that require a large dataset. Thus, if we provide more data we would achieve high metrics using the LSTM architecture 2 model.

Table 7. Comparison of Best Deep Learning Models with Naive Bayes Baseline Weighted Average Performance Metrics Achieved on Hate Speech Data.

S/n	Model Name	Architecture	Precision	Recall	F1 Score
1	Niave Bayes	SMOTE Oversampling	0.960	0.950	0.960
2	SimpleRNN	2	0.940	0.940	0.940
3	LSTM	2	0.950	0.950	0.950
4	GRU	Keras Tuner	0.950	0.940	0.940

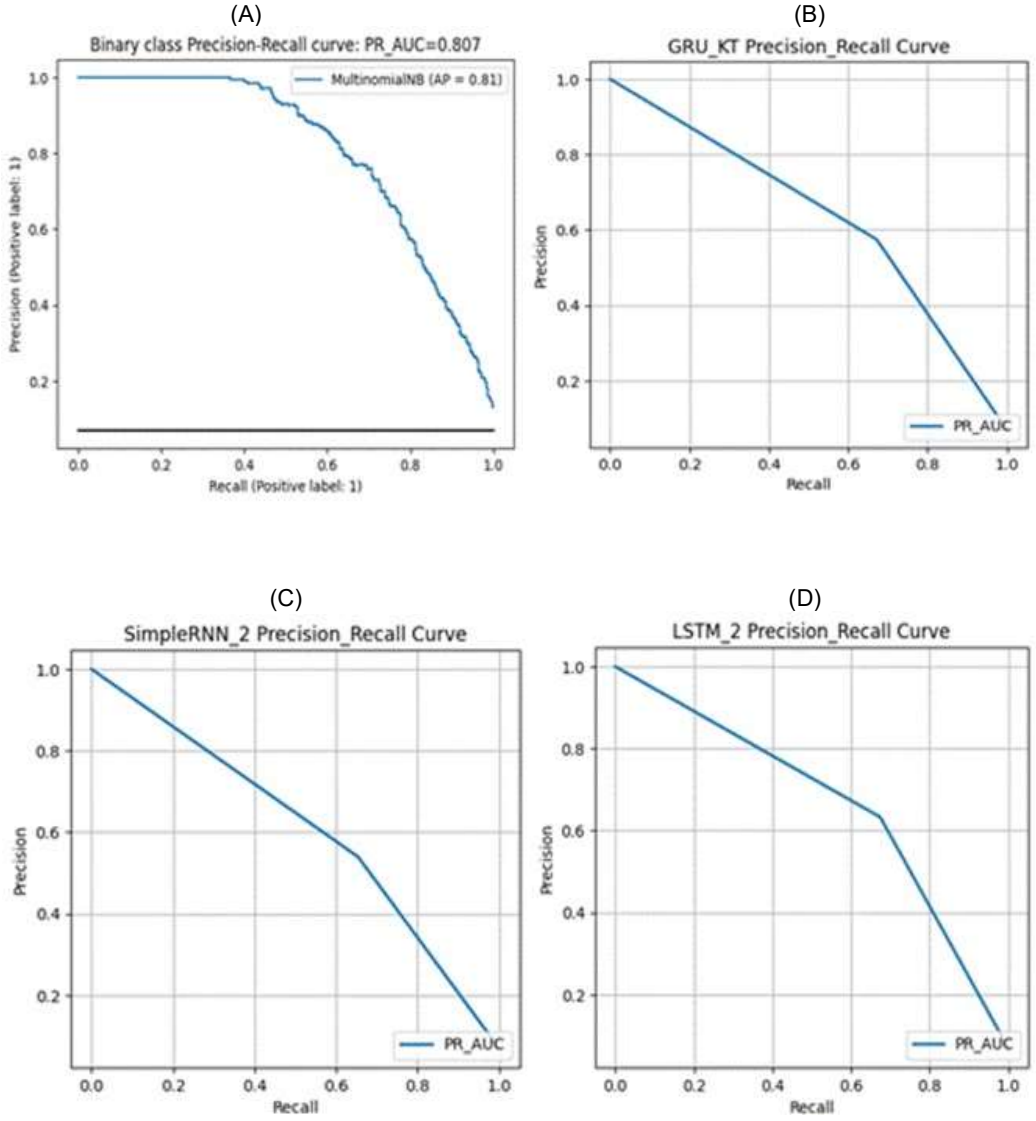

Figure 9. Precision Recall Curve of the Best Machine Learning & Deep Learning Models.

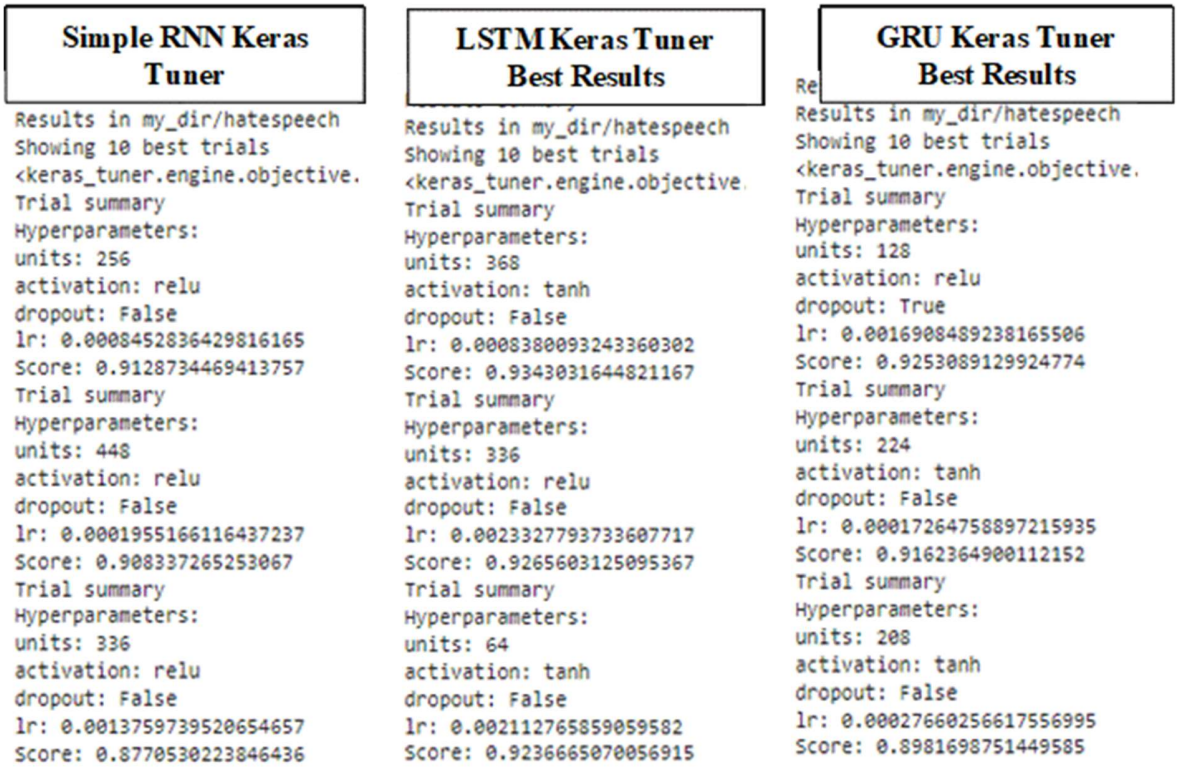

Figure 10. Keras Tuner Hyperparameter Tuning for the three Deep Learning Model's Best Results.

CONCLUSION

The need to limit using twitter platforms as a major source of curbing dissemination of hate speech cannot be over emphasized. Thus, our model has shown that it is possible to avert hate speech online. The main limitation of this project was limited train dataset and imbalanced nature.

Future work recommended is to use the BERT state of the art model and an ensemble model of SimpleRNN, LSTM and GRU to carry out sentiment analysis of this hate speech dataset.

REFERENCES

Al-Makhadmeh Z, Tolba Amr (2020). Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. *Computing. Archives for Informatics and Numerical Computation,* 102 (2): 501-522.

Chuluunsaikhan T, Ryu G, Yoo K, Rah H, Nasridinov A (2020). Incorporating deep learning and news topic modeling for forecasting pork prices: The case of south korea. *Agriculture (Basel),* 10 (11): 1-22.

Chung, J, Gulcehre C, Cho K, Bengio Y (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling.

Dey N, Mishra R, Fong SJ, Santosh KC, Tan S, Crespo RG (2020). COVID-19: Psychological and psychosocial impact, fear, and passion. *Digital Government: Research and Practice,* 2 (1): 1-4.

Gaydhani A, Doma V, Kendre S, Bhagwat L (2018). Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *arXiv Preprint arXiv:1809.08651,* .

Greevy E, Smeaton AF (2004). Classifying racist texts using a support vector machine. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.*

Jelodar H, Wang Y, Orji R, Huang H (2020). Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach.

Kaggle (2020). Twitter Sentiment Analysis - Analytics Vidya [Blog post] Practice Problem by Analytics Vidya Vinayak Dhage- March 2020. Available online: https://www.kaggle.com/datasets/dv1453/twitter-sentiment-analysis-analytics-vidya?select=train_E6oV3lV.csv [Accessed : 25/03/2022]

Mandl T, Modha S, Majumder P, Patel D, Dave M, Mandlia C, Patel A (2019). Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. *Proceedings of the 11th forum for information retrieval evaluation.*

Omnicore (2022) Twitter by the Numbers: Stats, Demographics and Fun Facts [Blog post] Salman Aslam-February 22, 2022. Available online: https://www.omnicoreagency.com/twitter-statistics/ [Accessed : 25/05/2022]

Oyebode O, Ndulue C, Adib A, Mulchandani D, Suruliraj B, Orji FA, Chambers CT, Meier S, Orji R (2021). Health, psychosocial, and social issues emanating from the COVID-19 pandemic based on social media comments: Text mining and thematic analysis approach. *J. MIR Med. Informatics.* 9 (4): e22734.

Pitsilis GK, Ramampiaro H, Langseth H (2018). Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence (Dordrecht, Netherlands),* 48 (12): 4730-4742.

Raza H, Faizan M, Hamza A, Mushtaq A, Akhtar N (2019). Scientific text sentiment analysis using machine learning techniques. *Int. J. Adv. Comp. Sci. Appl.* 10 (12):157-165.

Rustam F, Khalid M, Aslam W, Rupapara V, Mehmood A, Choi GS (2021a). A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis. *PloS One.* 16 (2): e0245909.

Rustam F, Khalid M, Aslam W, Rupapara V, Mehmood A, Choi GS (2021b). A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis. *PloS One,* 16 (2): e0245909.

Hochreiter S, Schmidhuber J (1997). Long short-term memory. Scikit-learn: Machine Learning in Python, Pedregosa *et al*., JMLR 12, pp. 2825-2830, 2011. Available Online :https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html [Accessed: 11/04/2022]

Staudemeyer RC, Morris ER (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. *arXiv Preprint arXiv:1909.09586,* .

Tom Davidson (2017). hate-speechand-offensive-language. https: //github.com/t-davidson/ hate-speech-and-offensive-language. Accessed: 2021-03-29.

Van Den Broeck J, Cunningham SA, Eeckels R, Herbst K (2005). Data cleaning: Detecting, diagnosing, and editing data abnormalities. *PLoS Med.* 2 (10): 966.

Zagidullina A, Patoulidis G, Bokstaller J (2021). Model bias in NLP -- application to hate speech classification using transfer learning techniques.

Zhang Aston, Lipton Zachary C, Li Mu, Smola Alexander J (2021). Dive Into Deep Learning. zhang2021dive. Online Source: https://d2l.ai/chapter_recurrent-modern/gru.html [Accessed : 25/05/2022]