University of Ibadan
Journal of Science and Logics in ICT Research

Detecting Hate Speech on Social Media Using Deep Learning Techniques

***Idris, David** didwiz83@gmail.com

Ogunseye, Elizabeth Oluyemisi elizabetholuyemisio@yahoo.com

Akinola, Solomon Olalekan akinola.olalekan@dlc.ui.edu.ng

Department of Computer Science, University of Ibadan, Ibadan, Nigeria

*** Corresponding Author**

Abstract

Hate speech is a recurring issue on social media platforms identified as an attack against a specific group of people based on certain common characteristics. As online data is created at a very fast rate by users, it has now become a daunting task to manually moderate the comments of users containing hate speech in a bid to reduce its negative effects on a platform. Previous works have been able to create models capable of detecting hate speech with good accuracy on hate speech detection on user comments and posts (known as tweets) on Twitter social media platform. Despite the good results obtained, this kind of models perform poorly when exposed to tweets that contained clever wordings, alternate spellings and rare words. Therefore there is a need to improve the model for the detection of hate speech in user comments on social media in order to address these problems. An ensemble model was developed from two baseline classifiers, NBSVM (Naive Bayes Support Vector Machine) and LSTM (Long Short Term Memory); combining the power of two well- known performing models from machine learning and deep learning using FastText embeddings from Facebook to improve hate speech detection even when clever wordings, alternate spellings and out of vocabulary words are used. This work was able to improve on the current state of the art hate speech detection by considering OOV (Out Of Vocabulary) words, clever and alternative spellings of words in developing a model that performed better than previous research works in detecting hate speech. The developed ensemble model proved to be able to detect hate speech even when clever wordings, alternate spellings and rare words were used in tweets. There was also an increase in the performance of the model's hate recall (77%) as compared to the existing popular work of Davidson *et. al*, (2017) hate recall (61%).

Keywords: *Hate speech detection, Social Media, Deep Learning, Support vector machine*

1. Introduction

Hate speech is usually outlined as any form of communication that disparages a person or a group of people on the premises of some characteristic such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic [1]. Examples are:

1. All black savages must die!
2. Those bastard Jews are taking over everything.

As user generated web content has increased over the years with the advent of new technologies and software that makes communications easier, hate speech has also grown tremendously. Online social media discussions are now more frequent and plays an important role of communication amongst members of a society. It is capable of bringing people together or against themselves depending on how it is used. Internet platforms struggles with moderation demands and news sites have been known to disable commenting on their articles, because they are being used to propel some agenda or offend readers.

Moderation practice cannot rely on humans anymore, because a single user alone can easily create a reasonable large amount of content in a short time frame that can take a toll on human moderators considering that a single social media platform can have

Idris, David, Ogunseye, Elizabeth Oluyemisi and Akinola, Solomon Olalekan. (2020). Detecting Hate Speech on Social Media Using Deep Learning Techniques, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 5 No. 1, pp. 22 - 38.
©U IJSLICTR Vol. 5, No. 1, June 2020

millions of users. Adding to that is the fact that moderation needs to be done with care. It is simply much more time consuming for the moderator compared to the perpetrators who only need to cut and paste their hate statement across different social media platforms on the internet. Anonymity also adds to the problem, as it seems to bring out the worst in people, since individuals can just type whatever they like while their "identities" are hidden [2].

Hate speech is a phenomenon that is constituting to the disunity of the good people of Nigeria at large. This is largely influenced by hate speech and tribalistic comments made on social media. In the bare minimum it causes depression, feeling of neglect, annoyance, unnecessary anger and breed hate in the minds of the victim, and also in bystanders who are witnesses to the act. In extreme cases hate speeches have led to hate crimes, where due to the increased emotional tension that has been riled up based on such derogatory remarks, the affected persons acted on some of these negative emotions, creating victims of cascading events that started with just one person's (in most cases) comment on a sensitive issue.

Previous research works in hate speech recognition face the problem of users being able to obfuscate tweets to beat the current state of the art hate speech detection by using new slang words or through inventive clever spellings of words that are not available in the popular pre-trained word embeddings such as Word2Vec or GloVe, but are highly common with hateful comments. Therefore there is a need to develop a model that improves on the current state of the art by detecting hate speech comments even when clever phrases are used.

The rest of this paper is organized as follows: In Section 2, relevant existing works are presented. Section 3 is devoted to the methodology adopted in this work while results and discussion are presented in Section 4. Conclusion is presented in Section 5.

2. Literature Review

Online social platforms are pervaded with hateful speech which are contents that expresses hatred for a person or group of people. Such content can frighten, intimidate, or silence platform users, and some of it can motivate other users to commit acts of violence known as a hate crime [3].

One of the major issues with defining hate speech is that it is subjective, with different organisations having different definitions for it. This issue has affected previous researchers [1, 4] on hate speech detection who did not come up with a standard definition for hate speech before annotating their data. This resulted in conflicting definitions of hate speech by the annotators which lead to poor quality of data, which in turn degraded the performance of their models.

This issue is also evident in the Nigerian hate speech bill proposed by one Senator Aliyu Sabi Abdullahi from Niger State which states that "*Any person who uses, publishes, presents, produces, plays, provides, distributes and/or directs the performance of any material, written and/or visual, which is threatening, abusive or insulting or involves the use of threatening, abusive or insulting words, commits an offence*" [5]. This bill clearly puts offensive and cyber bullying behaviour under the same umbrella as hate speech; thus, wrongfully classifying offensive statements as hate speech and meting out punishments meant for hate speech offenders to people convicted for making offensive statements.

Thus this study adopted the definition established by Davidson *et. al.*, [6] as any "*communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics*". This work improves on Davidson *et. al.*, [6] definition by combining it with the three-premise based hate speech definition by Ona *et. al.*, [7] which was used to define hate speech for this work. Three premises must be met for a speech to be classified as hate, they are:

1. A deliberate attack.
2. It must be directed towards a specific group of people.
3. It must be motivated by aspects of the group's identity.

Previous works have used a number of approaches in classifying hate speech; a re-occurring limitation observed from previous works was that only a binary definition and label was used for classification, i.e., "hate speech" and "not hate speech". Davidson *et. al*., [6] has shown in his work that this resulted in the "hate speech" label having other forms of abusive words in it other than hate, such as offensive words. He and his team were able to show that breaking up the hate speech definition label to a multi-class problem improves the classification accuracy. Davidson *et. al.* [6] were able to prove that multiclass definition of hate speech lead to improvement in hate speech detection and classification, and as such, our study also followed the same principle by using the three class labels; namely, hate speech, offensive language and neither as was used in his research work.

Gitari *et. al*., [4] used lexical based approach and developed a collection of hate verbs which consists of wordings that condone or indicate hate. This corpus of hate verbs proved useful for simple/naive systems where blacklisting of words was the main aim of the system. Sorzano *et.al.,* [8] worked on Dimensionality Reduction which helped to mitigate a model's issue with generalization and overfitting, as trying to fit a model to a dataset with high dimensions (containing lots of features) that makes a model prone to overfitting and slower training time.

Sharma [9] used Feature selection to filter out irrelevant or redundant features from a given dataset. It tries to find a subset of the original feature set where relevant features dominate. He also used Feature extraction to create a new, smaller set of features that captures most of the relevant information in the dataset. The impact of using annotators with varying backgrounds was revealed by Waseem *et. al*., [10], who achieved very different results using three different groups of annotators for hate speech labeling using the same machine learning techniques on the different set of annotated data.

Marzieh Mozafari *et. al.* [23] worked on transfer learning approach on pre-trained language model BERT to enhance hate speech detection on publicly available benchmark datasets. They were able to discover that when they used the pre-trained BERT model and then fine-tuned it on the downstream task by leveraging syntactical and contextual information, all BERT's transformers works better but was unable to 'debias' hate speech.

Apurva Parikh *et. al.* [24] worked on Identification of Hate Speech using Machine Learning and Deep Learning approaches for social media posts. The research was conducted to identify hate and offensive language in Twitter and Facebook posts. They used Machine Learning techniques such as Logistics Regression and Naïve Bayes classifier and a Deep Learning based approach that utilizes Convolutional Neural Networks. They used two approaches for their experiments in which in the first one, they represented the input features with CountVectorizer and Tfidf. The features were used to train logistic regression and Naïve-Bayes classifiers. Their second approach used three layers of ID Convolutional using hierarchical CNN. Their experiments did not go through data pre-processing. The accuracy of their methods did not go beyond 70% in all the sub-tasks used in their experiment.

Other researchers that used machine learning approaches like Gaydhani *et. al*. [11], Fauzi and Yuniarti [12], Burnap and Williams [13], Olteanu, Talamadupula and Varshney [14] and even Davidson *et. al*., [6] did not use an ensemble or multistep approach to boost the accuracy of their classifier. Although, Zhang *et. al*. [15] and Badjatiya *et. al*., [16] used an ensemble approach, their resulting classifiers were still unable to detect clever wordings, alternative spellings and out of vocabulary words (OVW).

Another limitation existing in this domain, was the lack of good comparative reports against other researchers' works. This might be due to the use of different datasets gathered by different researchers. This too lead to another problem during the course of this research work as noticed in all the hate speech dataset publicly available online, there was a large class imbalance between the hate speech class label to other class labels in the same dataset. This implies that most of the existing works were done with this class imbalance, and class imbalance has been proved capable of creating biases and making many machine learning model over fit on the dominant class labels.

3. Methodology

In this study, we aimed at detecting hate speech in tweets even when clever wording, alternative spellings and out of vocabulary words are used in such tweets. This study builds on the work of Davidson *et. al*. [6], where they proved that a multi-class label hate speech problem space improved accuracy and false positives. The approach was broken down to the following stages as shown in Figure 1 while Figure 2 shows the flow chart for the methodology:

1. Dataset Collection.
2. Human classification.
3. Train and evaluate baseline classifiers for the ensemble.
4. Ensemble baseline classifiers using soft voting.
5. Evaluate ensemble model

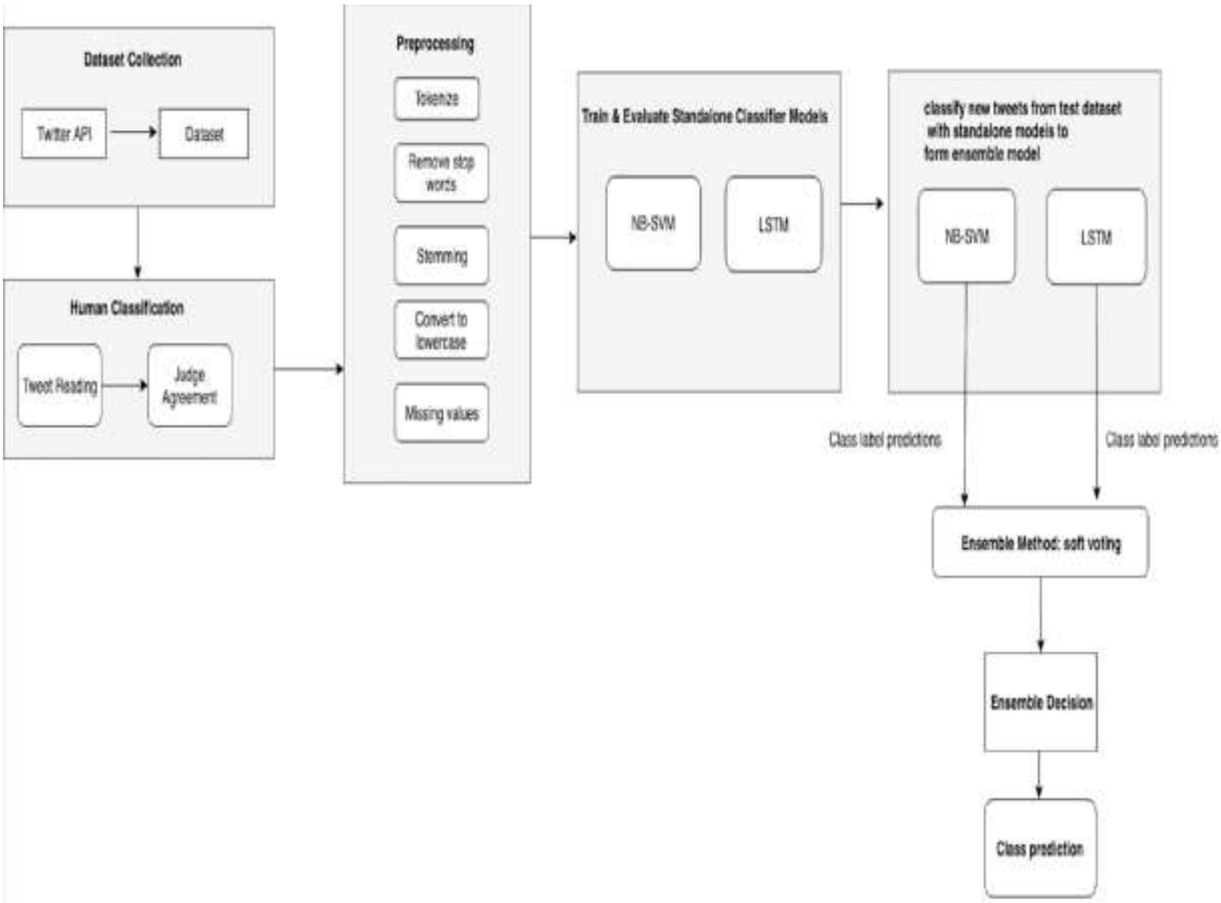


Figure 1: Proposed model methodology stages

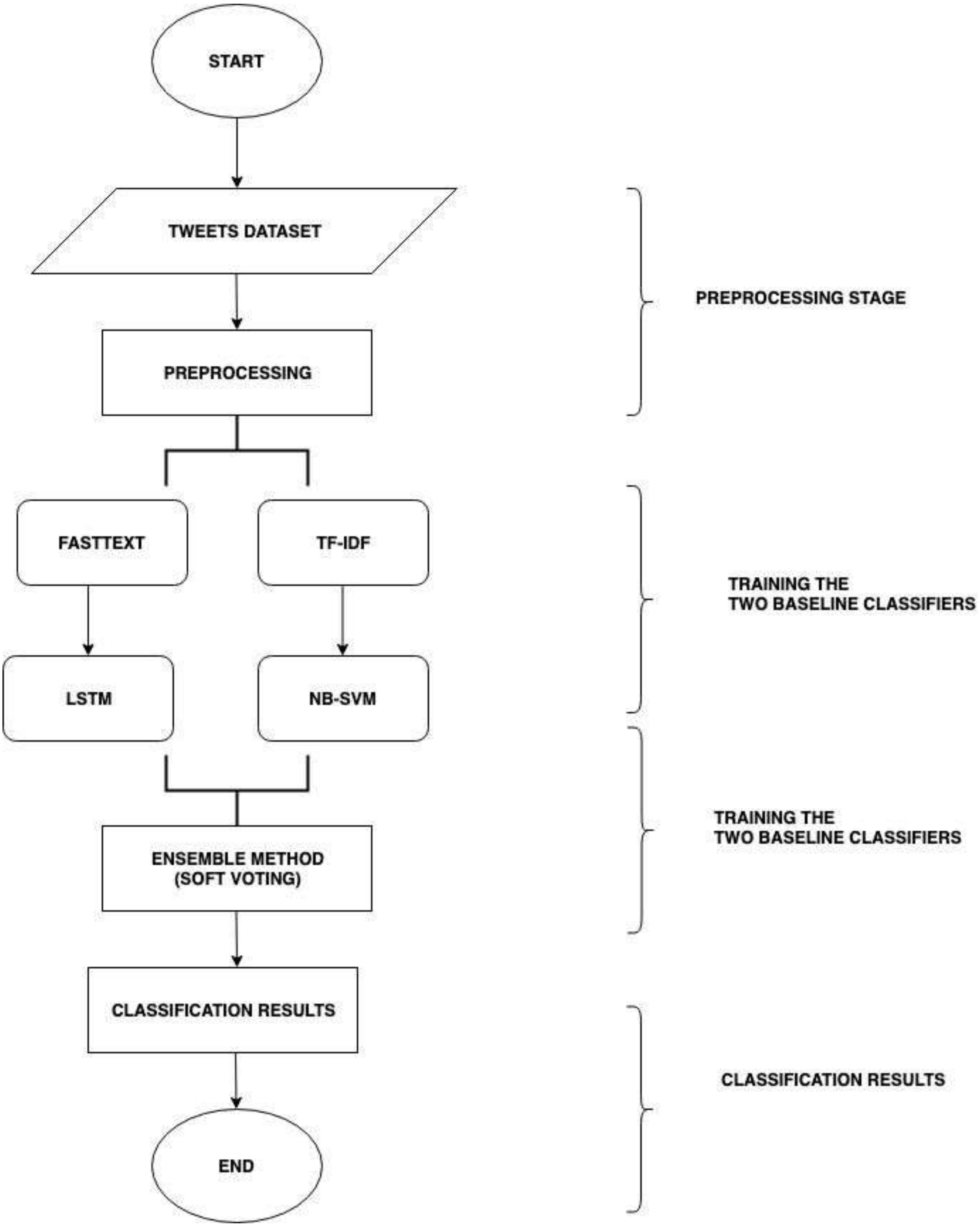


Figure 2: Proposed model methodology flow chart

3.1 Data Analysis

The collated dataset contained twenty-four thousand, seven hundred and eighty-three (24,783) tweets which contains 3 class labels namely “hate speech”, “offensive language” and “neither”. The dataset also has a count column that indicates the annotator vote of decision on that tweet when classifying. The dataset was split up for downstream tasks into 80% for training and 20% for testing performance.

3.2 Data Limitations

An issue with the data set that was pointed out in the literature was that there is a high level of class imbalance with the hate speech class label sample size. Figure 3 shows a bar chart indicating this class imbalance. Table 1 also goes into more details showing that the hate speech class label occupies only 5.7% of the entire dataset. The table also points out that there was a high level of disagreement on whether that tweet was a hate speech or not, compared to other class labels such as offensive language and neither, the annotators were more in unison.

3.3 Data Augmentation

This is a way of handling class imbalance; common techniques are: Up-Sampling which means the observations in the minority class are taken randomly and used to generate new similar samples to balance out the majority class. Down-Sampling means the observations in the majority class are removed randomly till the minority class are balanced out. A data manipulation python library Pandas was used to perform up sampling for the dataset to solve the problem of class imbalance before proceeding to the data preprocessing stage.

Figure 4 displays the result of the data augmentation performed on the dataset using up-sampling techniques on the dataset. Before performing the up sampling more specific hate tweets were obtained to boost up the hate speech sample distribution which is extremely low, since the model might end up under fitting for this class regardless of the data augmentation techniques used, because for text, data augmentation degrades the quality of the data, unlike images where data augmentation have been shown to result in quality data [17].

Table 1: class label data distribution and annotator disagreement count per class

Class	Data size (percentage)	Disagreement(agreement)
Hate Speech	1,430 samples (5.77%)	87.4% (12.6%)
Offensive Language	19,190 samples (77.43%)	22.1% (77.9%)
Neither	4,163 samples (16.79 %)	28.8% (71.2%)
Total	24,783 samples (100%)	

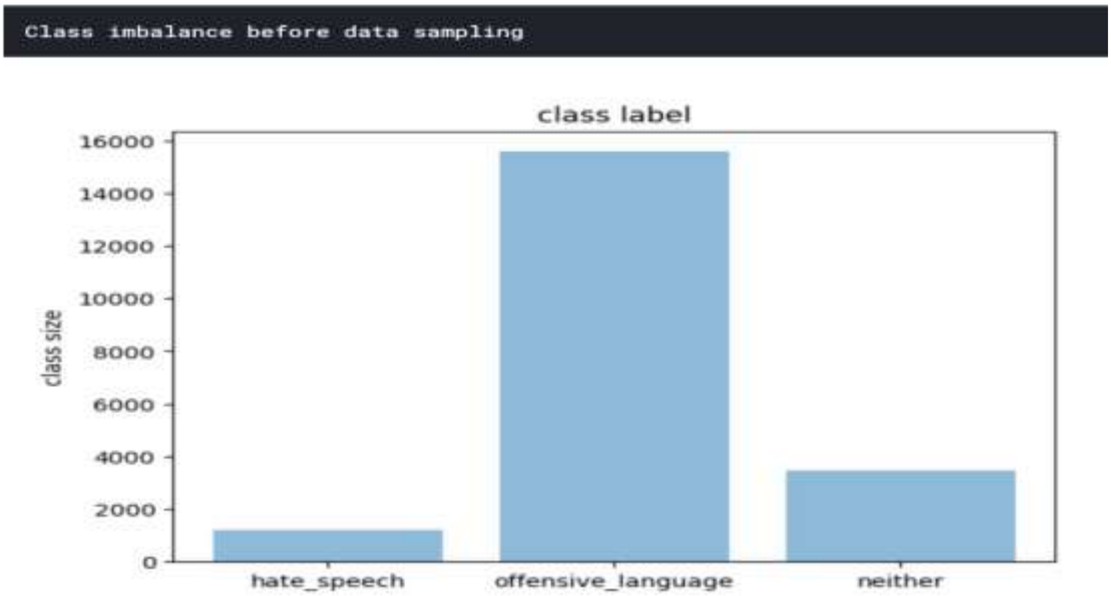
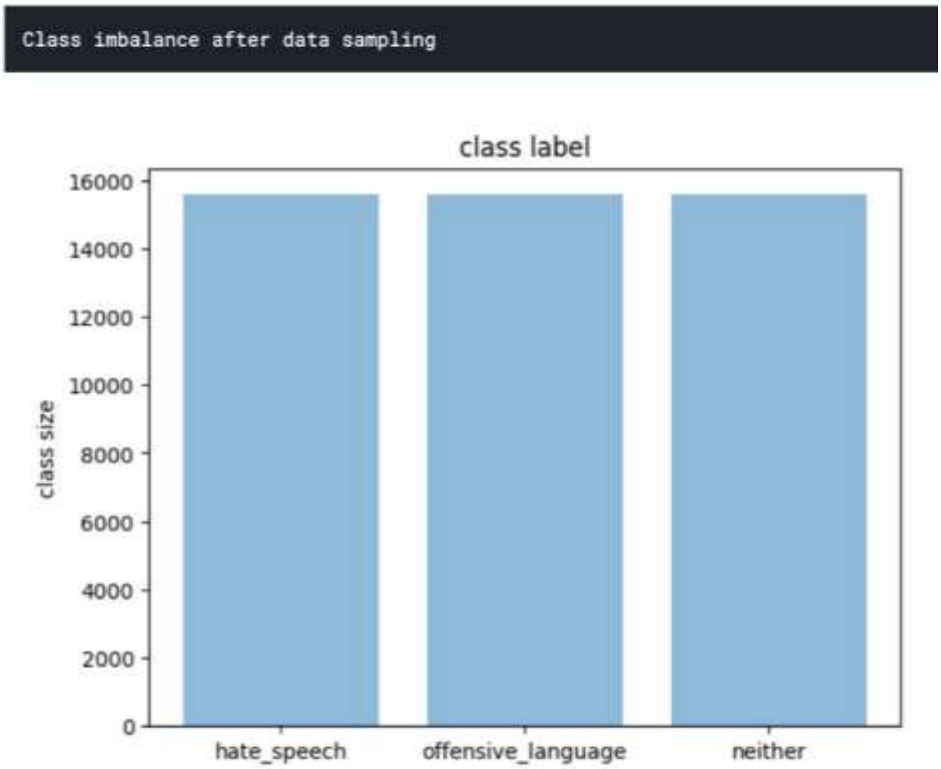


Figure 3 Bar chart showing class imbalance in the dataset.



Class imbalance after data sampling

Figure 4: Bar chart showing class balance after u p -sampling data augmentation techniques were applied.

3.4 Data Preprocessing

This is usually done to remove unnecessary “noise” from the dataset, to prevent the model from learning those unwanted features and thus resulting in a low accuracy, for example usernames were removed and replaced with @user which normalises the dataset in respect to usernames. Popular stop words in English such as “is”, “the”, “on”, “a” were removed using the stop words provided in the NLTK (Natural Language Toolkit) python library. Stemming techniques were applied to remove all morphological affixes and break words down to their stem and root words e.g. *moving* becomes *move*, *foolish* becomes *fool*. The Snowball Stemmer python library was used to accomplish this, all words were also converted to lower case as another normalization step.

The data was preprocessed before being trained on the two baseline classifiers, stop words were removed, stemmed, converted to lowercase, replace missing values, Table 2 shows some examples of the cleaned dataset.

Table 2: Table showing differences in processed tweet and raw tweet data.

Raw Tweet	Pre-processed Tweet
" bitch get up off me "	bitch get up off me
" got ya bitch tip toeing on my hardwood floors " &#128514; http://t.co/cOU2WQ5L4q	got ya bitch tip toe on my hardwood floor 128514
!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny	rt g anderson based she look like a tranny
"@DevilGrimz: @VigxRArts you're fucking gay, blacklisted hoe" Holding out for #TehGodClan anyway http://t.co/xUCcwoetmn	you re fuck gay blacklisted hoe hold out for tehgodclan anyway
"@Gizzy_Jones94: If she kiss u with her eyes open watch that bitch!"lmfao	jones94 if she kiss u with her eyes open watch that bitch lmfao

Table 3: One hot encoding for the class label values.

Class Label	Hate speech	Offensive language	neither	Previous values
hate speech	1	0	0	0
offensive language	0	1	0	1
neither	0	0	1	2

Finally all missing values were replaced with a default value “na” standing for “not available” using Pandas library fillna method which automatically looks for missing value in the given data frame and replaces the missing value in the field with the given value which is “na” is this case. One-hot-encoding was used to transform the dataset class label values into a binary problem of 1 and 0, as opposed to the 0 to 2 value set used in the original dataset, so the transformed class label values are as shown in Table 3.

3.5 Baseline Classifiers

A baseline classifier is a standalone classifier which will form part of our ensemble, for this research work the ensemble model consists of two baseline model classifiers which are NBSVM (Naive Bayes-Support Vector Machines) and LSTM (Long-Short-Term-Memory). These classifiers were chosen for this research work based on their success in previous works and because of how different they are, NBSVM is a machine learning algorithm while LSTM is a deep learning neural network. It was also ensured that their data representations were different because ensembles are proven to work best and guarantee a boost in accuracy when the baseline classifiers are far apart [18]. Random forests works so well because of this fact.

NBSVM also known as Naive Bayes - Support Vector Machine was first mentioned by Sida Wang and Chris Manning [19]. it was built using SVM (Support Vector Machines) over NB (Naive Bayes) log-count ratios as feature values and was proven to perform well on snippets and longer documents, for sentiment, topic and subjectivity classification, and is often better than previously published results using machine learning algorithms. Based on these observations, NBSVM was chosen to be one of the baseline classifiers for the ensemble model.

Data word representation using Term Frequency-Inverse Document Frequency (TF-IDF) were created for the NBSVM model. Although Wang and Manning [19] suggested using bag of words with n-grams, this research work used TF-IDF, which is a statistic reflecting the importance of a word in relation to its document to improve the classifier's accuracy [20].

The TF-IDF data representation matrix for the dataset was obtained by taking the number of times a word occurs in a tweet, and the inverse tweet frequency to get the co-occurrence of those words. The TF-IDF matrix was gotten using TFIDFVectorizer method from Scikit learn machine learning library to fit and transform the training set for downstream tasks like prediction. This matrix was then passed to the NBSVM model created with a popular class implementation on NBSVM which was obtained from Github a popular code resource and refactored for this research work.

3.6 NBSVM Model

The NBSVM model was trained on data representations gotten from TF-IDF (Term Frequency- Inverse Document Frequency) vectors. A linear kernel was used to separate the dataset classes properly on a linear plane.

The NBSVM model was trained on 70% of the training dataset while holding out 10% of the training dataset to evaluate the model's performance before testing it on the test dataset which is 20% of the original preprocessed dataset. Hyperparameters used to build the NBSVM model are as shown in Table 4.

Table 4: NBSVM Hyper Parameters.

Parameters	**Values**
Size of training data set	46k tweets (after sampling)
Size of test data set	10k tweets (after sampling)
Kernel	Linear
Regularization (c parameter)	2.0
Gamma	auto
Probability	True

3.7 LSTM Baseline Classifier

Long Short Term Memory (LSTM) discovered by Hochreiter and Schmidhuber [21] solves the vanishing or exploding gradient problem by adding a long term memory to RNN's short term memory, thus the name long short term memory.

3.7.1 LSTM Data Representation

Embeddings are data representations where real vector numbers are used to represent similar words. FastText is known to be able to represent OOV (Out Of Vocabulary) words by using character n-grams which enables it to detect OOV words that might have similar n-grams to a known words. This gives it an edge over Glove and Word2Vec that are not capable of doing this, instead researchers give unknown words a default value, this in turn might affect the model capabilities in handling a dataset with a lot of unknown words or words with alternative spellings.

This influenced this research work's decision to use this library to build the data embeddings since the aim of this work is to accurately detect hateful clever wordings, alternative spellings and unknown words in a tweet. A FastText pre-trained model was obtained from Facebook's FastText website which contained 2 million word vectors with 300 dimensions trained with subword information on Common Crawl (consisting of a total of 600 billion tokens). This pre-trained model was trained on the hate speech training dataset to extract the necessary features to build a data embedding which will be used for downstream tasks (such as creating the embedding layer for the LSTM model).

3.8 LSTM Model

The LSTM model was trained on the same split of the dataset that was used to build the NBSVM model, which was 80% training dataset and 20% test dataset. Hyperparameters are values set for the model before training the model, which is vital to performance, as they form the architecture of the model's deep learning network, Table 5 shows the hyper parameters used to build the LSTM classifier model.

3.9 The Ensemble Model

An ensemble uses a combination of two or more learners to improve the overall performance of a model. This study used an ensemble model to improve over the result of two baseline classifiers (NBSVM and LSTM). The results from the two baseline classifiers were saved to develop the ensemble model. A soft voting combination technique known as soft voting as shown in Figure 5 was used to combine results from the two baseline classifiers to get the average from the probabilities results of the two classifiers to obtain a more performant model.

Table 5: LSTM Hyper Parameters.

Parameters	**Values**
Size of training data set	46k tweets (after sampling)
Size of test data set	10k tweets (after sampling)
Batch Size	32
Loss function	Binary_cross entropy
Optimizer	Adam
Hidden LSTM Layers	100
Activation function	Sigmoid
Epoch	2
Dropout	0.2
Metric	Accuracy
Learning Rate	0.1

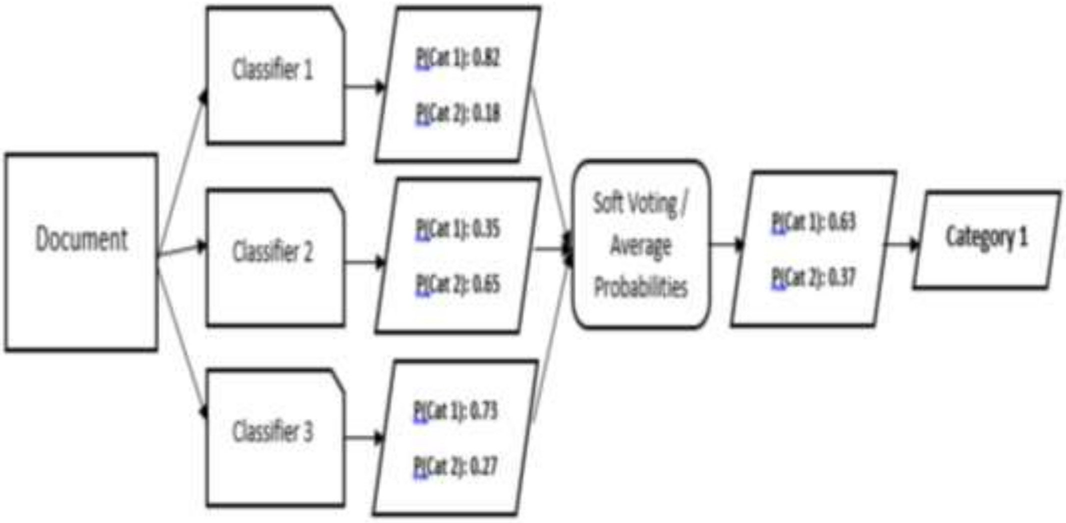


Figure 5: Soft Voting Ensemble method (Source: Fauzi & Yuniarti, 2018)

3.10 Implementation Environment

This study was carried out using a cloud based GPU kernel from Kaggle, Kaggle provides each kernel a free 13 gigabytes of ram NVidia Tesla p100 GPU for machine learning tasks, which helped to make the model training faster compared to when CPU is used. Kaggle kernel also provides easy setup and easy sharing of work. Its kernels are based on Jupyter which is a popular environment for doing machine learning work. Keras and Scikit learn Python libraries were used to build the ensemble baseline models.

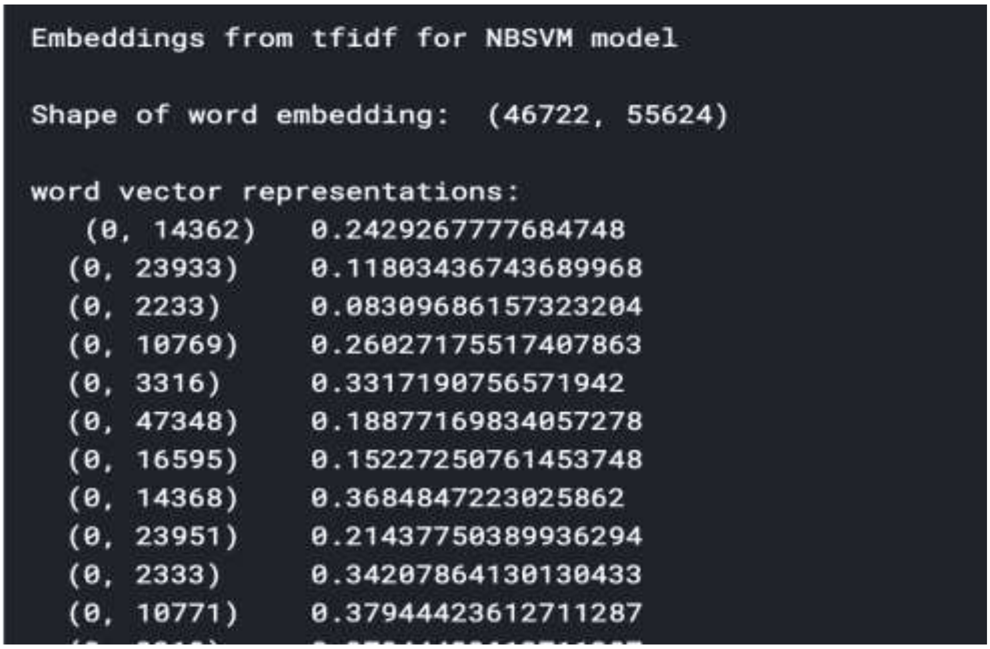
3.11 Performance Evaluation

In this section, the performance of the model was evaluated. The confusion matrix of the two baseline classifiers was used to evaluate the performance of the baseline model and finally the ensemble model. In this study, accuracy, precision and recall were used to evaluate the performance of the classification model. In order to determine the performance of the model with the aforementioned metrics, four significant parameters were considered which are the true positives, true negatives, false positives and false negatives.

4. Results

4.1 Result for NBSVM Baseline Data Representation

After preparing the data, it was vectorized to create the needed data representations for downstream tasks using TfidfVectorizer from Scikit Learn python library, a data embedding with the shape of [46722, 55624] was obtained where the first element in the shape is the number of rows in the embedding and the second element is the number of columns in the embeddings as shown in Figure 6.

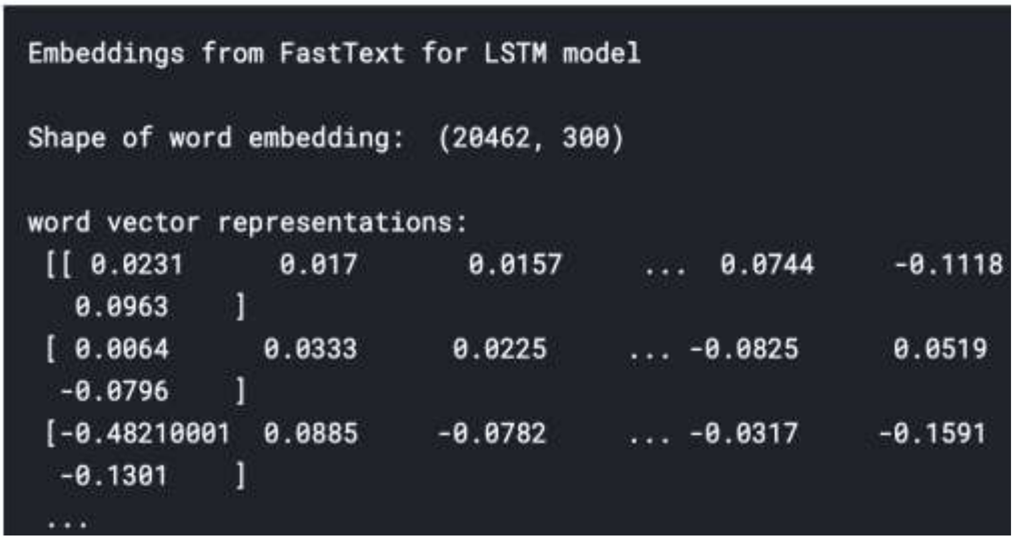


```
Embeddings from tfidf for NBSVM model

Shape of word embedding:  (46722, 55624)

word vector representations:
   (0, 14362)   0.2429267777684748
  (0, 23933)    0.11803436743689968
  (0, 2233)     0.08309686157323204
  (0, 10769)    0.26027175517407863
  (0, 3316)     0.3317190756571942
  (0, 47348)    0.18877169834057278
  (0, 16595)    0.15227250761453748
  (0, 14368)    0.3684847223025862
  (0, 23951)    0.21437750389936294
  (0, 2333)     0.34207864130130433
  (0, 10771)    0.37944423612711287
```

Figure 6: Embedding shape from TF-IDF for the NBSVM model

4.2 Result for LSTM Baseline Data Representation

Figure 7 shows the embedding shape. A max features parameter (which is the total number of rows) was set as 20479, and the maximum amount of words in a tweets as 100. Twitter allows 280 characters [22], an improvement over the previous limit of 140 characters by tweet. The vector size parameter was set to 300 in preparation for the embedding use in downstream tasks.



```
Embeddings from FastText for LSTM model

Shape of word embedding:  (20462, 300)

word vector representations:
 [[ 0.0231      0.017       0.0157     ...  0.0744     -0.1118
   0.0963    ]
 [ 0.0064      0.0333      0.0225     ... -0.0825      0.0519
  -0.0796    ]
 [-0.48210001  0.0885     -0.0782     ... -0.0317     -0.1591
  -0.1301    ]
 ...
```

Figure 7: Embedding shape from FastText for the LSTM model

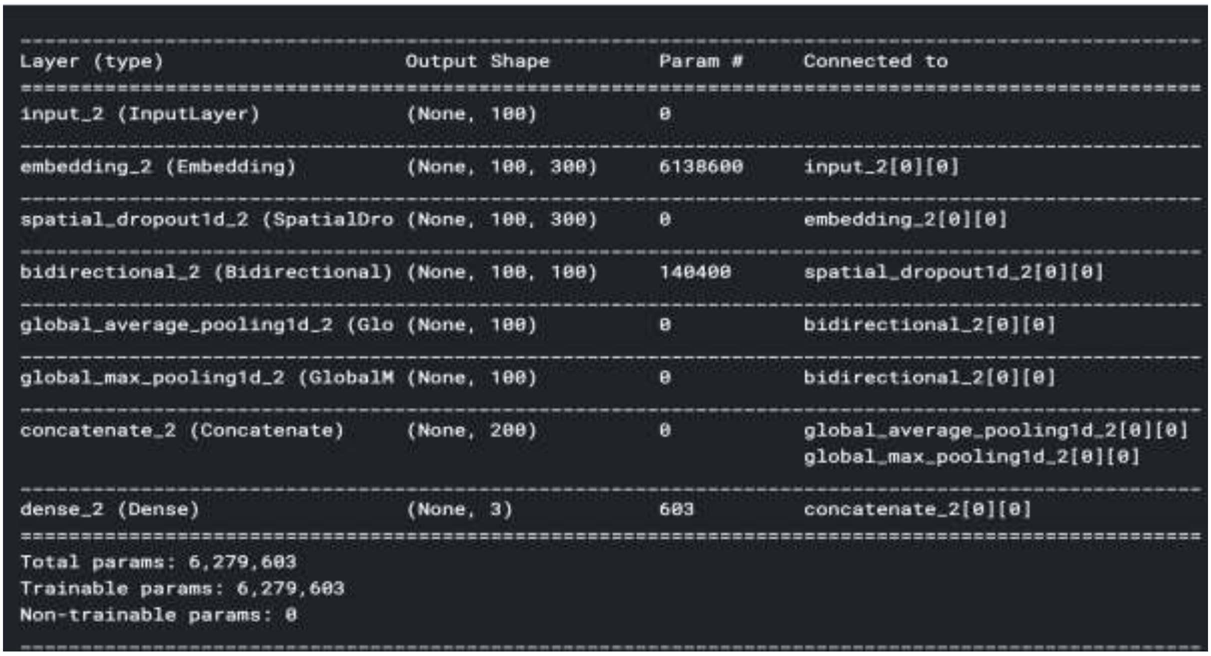
4.2.1 LSTM Model Summary

The LSTM model summary shows the result of the untrained model compiled with the set hyper parameters in Figure 8. There are about 6 million parameters in the embedding vector. The output of the embedding layer was passed to the LSTM layer which was subsequently passed to the dense layer.

4.3 Performance Evaluation Results

4.3.1 Baseline Classifiers

The NBSVM baseline classifier correlation map (confusion matrix) obtained after using the model to predict on the test data set is shown in Figure 9. The matrix shows the model to have an accuracy of 77%, with hate recall of approximately 70% and offensive language of 68% with a high neither recall of 94%.



```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            (None, 100)          0
__________________________________________________________________________________________________
embedding_2 (Embedding)         (None, 100, 300)     6138600     input_2[0][0]
__________________________________________________________________________________________________
spatial_dropout1d_2 (SpatialDro (None, 100, 300)     0           embedding_2[0][0]
__________________________________________________________________________________________________
bidirectional_2 (Bidirectional) (None, 100, 100)     140400      spatial_dropout1d_2[0][0]
__________________________________________________________________________________________________
global_average_pooling1d_2 (Glo (None, 100)          0           bidirectional_2[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_2 (GlobalM (None, 100)          0           bidirectional_2[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 200)          0           global_average_pooling1d_2[0][0]
                                                                 global_max_pooling1d_2[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 3)            603         concatenate_2[0][0]
==================================================================================================
Total params: 6,279,603
Trainable params: 6,279,603
Non-trainable params: 0
__________________________________________________________________________________________________
```

Figure 8: The LSTM model summary

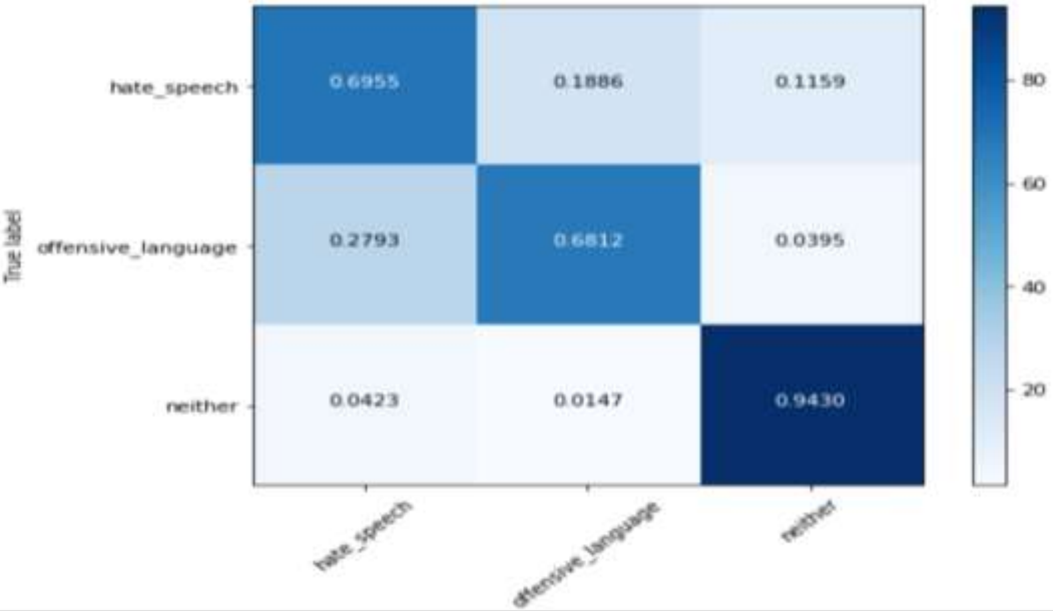


Figure 9: Correlation map from the NBSVM baseline classifier

The LSTM baseline outperformed the machine learning model (NBSVM baseline), which was expected as deep learning models have been proved to perform better than ordinary machine learning models in most cases. Figure 10 shows the LSTM baseline model to have an accuracy of 90%, with hate recall of approximately 85% and offensive language of 94% with a recall of 92%.

4.3.2 Ensemble Classifiers

After predictions with the baseline classifiers, ensemble combination techniques were applied using soft voting to average out the results from the two baseline classifiers. Figure 11 shows the confusion matrix.

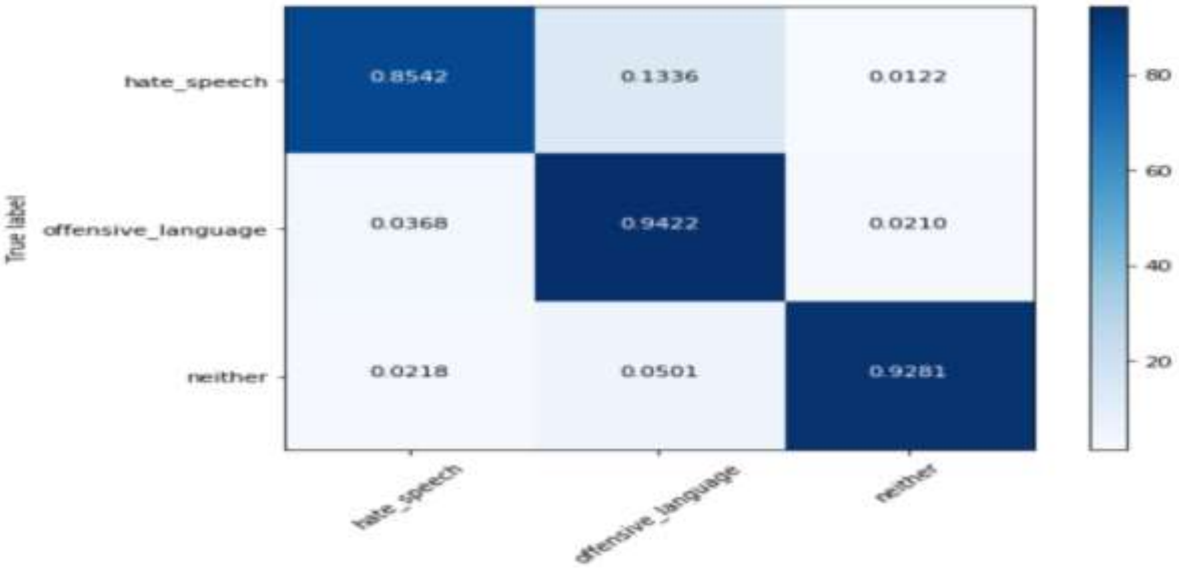


Figure 10: Correlation map from the LSTM baseline classifier

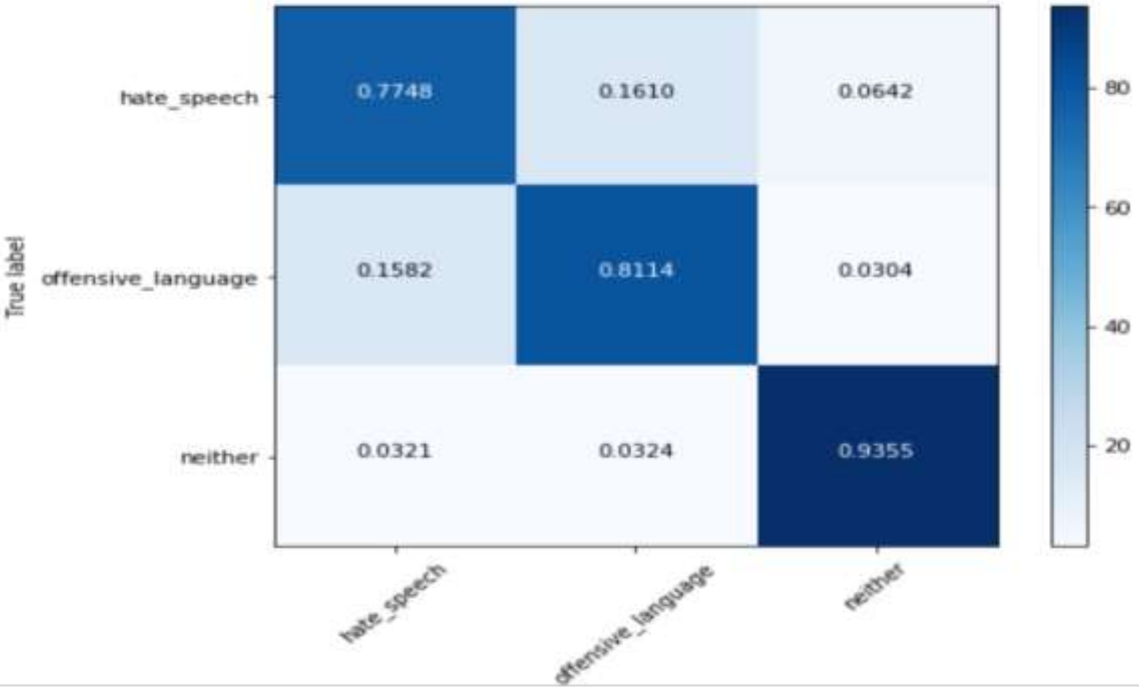


Figure 11: Correlation map from the ensemble model

An overall model accuracy of 84% and hate recall of 77% was achieved from our ensemble model which performed better than the model proposed by Davidson *et. al*, [6] whose model's hate recall was 61%. The ensemble model also performed better than some other research works that also used the same dataset as shown in the Table 6. The ensemble model's hate recall performance was better, which means the obtained ensemble model was able to correctly detect hate speech in statements with 77% accuracy.

4.4 Sample Tweets Classification

To test the ensemble model's ability in detecting Out Of Vocabulary words (OOV), some sample tweets known to contain clever wordings and alternative spellings were used to test the model to see its performance, Table 7 shows the model's predictions.

Table 6: Comparative report on the result of the ensemble model with previous works.

Researcher	**Classifier**	**Embedding**	**Hate F1 score**	**hate Recall**	**hate precision**	**overall model Accuracy**
David Thompson *et. al*.	SVM	TF-IDF n gram	0.51	0.61	0.44	0.89
Zhang *et. al.*	CNN + GRUs	word2Vec	0.94	n/a	n/a	71.85
Proposed model (soft voting)	**NBSVM + LSTM**	**Tf-idf n- gram + FastText**	**0.82**	**0.77**	**0.87**	**0.84**

Table 7: Sample prediction by the ensemble model to show OOV detection

Tweet	**Hate Speech(%)**	**Offensive Language(%)**	**Neither(%)**
All black savages must die.	0.80	0.13	0.49
God bless them all, to hell with all blacks	0.36	0.27	0.48
"Those bastards Jews are taking over everything."	0.75	0.14	0.48
'those f&ttgg&gt*t h*m*s'	0.5	0.19	0.04
Islam on lunatics should be left on some island for dead'	0.57	0.22	0.12

5. Conclusion

An ensemble model was developed using soft voting combination techniques on two baseline classifiers namely NBSVM (Naive Bayes Support Vector Machine) and LSTM (Long Short Term Memory). Facebook's FastText with character n-grams known for detection of rare and TF-IDF (Term Frequency Inverse Document Frequency) were used to create the data representations used in training the two baseline classifiers over a 50K augmented tweets dataset.

The NBSVM baseline performed fairly well with 77% accuracy while the LSTM baseline performed quite well with an accuracy of 90%. The ensemble model however dropped the performance of the LSTM and boosted the NBSVM to a cumulative 84% accuracy. On closer investigation, it was observed that the ensemble model performed better than the LSTM model in detecting OOV word, alternate spelling and clever wording of hate speech in tweets which satisfies the aim of this study. While improving on the hate recall accuracy, the ensemble model performed better than Davidson *et. al.* [6] and Zhang *et. al.* [15] overall model accuracy.

In Nigeria most of the time, people use local tone or communication like pidgin English on Social Media. Therefore, in the future we would like to extend this work on such local data set.

References

[1] 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

[2] Teemu Kinnunen (2017). Hate speech detection, https://futurice.com/blog/hate-speech-detection, accessed May, 2020

[3] Saleem, H. M., Dillon, K. P., Benesch, S., & Ruths, D. (2017). A Web of Hate: Tackling Hateful Speech in Online Social Spaces. *ArXiv:1709.10159 [Cs]*. Retrieved from http://arxiv.org/abs/1709.10159

[4] Gitari, N. D.; Zuping, Z.; Damien, H.; and Long, J. (2015). A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering 10:215–230.

[5] Ameh Godwin. (2018). Hate speech offenders to die by hanging – Nigeria Senate's new bill. Retrieved in June 2019 from https://dailypost.ng/2018/03/01/hate-speech-offenders-die-hanging

[6] 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

[7] Ona de Gibert, Naiara Perez, Aitor Garcıa-Pablos and Montse Cuadros (2018) Hate Speech Dataset from a White Supremacy Forum, https://www.researchgate.net/publication/327621135_Hate_Speech_Dataset_from_a_White_Supremacy_Forum Accessed May 2020

[8] Sorzano, C. O. S., Vargas, J., & Montano, A. P. (2014). A survey of dimensionality reduction techniques.*ArXiv:1403.2877 [Cs, q-Bio, Stat]*. Retrieved from http://arxiv.org/abs/1403.2877

[9] Sharma Pulkit. (2018). Comprehensive Guide to 12 Dimensionality Reduction Techniques. Retrieved 15 June 2019, from Analytics Vidhya website: https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-pyth on/

[10] Waseem Zeerak (2016). Are you a racist or am I seeing things? Annotator influence on hate speech detection on twitter," in *Proceedings of the first workshop on NLP and computational social science*.

[11] 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:1809.08651 [Cs]*. Retrieved from http://arxiv.org/abs/1809.08651

[12] Fauzi, M. Ali, Yuniarti Anny (2018). Ensemble Method for Indonesian Twitter Hate Speech Detection, *Indonesian Journal of Electrical Engineering and Computer Science* Vol. 11, No. 1, July 2018, pp. 294~299

[13] Burnap, P., and Williams, M. L. (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy & Internet 7(2):223– 242.

[14] Olteanu Alexandra, Talamadupula Kartik, Varshney Kush R (2017). The limits of abstract evaluation metrics: The case of hate speech detection, Proceedings of the 2017 ACM Web Science Conference, pp. 405-406

[15] Ziqi Zhang, David Robinson, and Jonathan Tepper (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network.

[16] Badjatiya Pinkesh, Shashank Gupta, Manish Gupta, Vasudeva Varma (2017). Deep learning for hate speech detection in tweets. Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759 -760.

[17] Munoz-Bulnes, J., Fernandez, C., Parra, I. Fern ˜ andez-Llorca, D. and Sotelo, M. A. (2017). "Deep Fully Convolutional Networks with Random Data Augmentation for Enhanced Generalization in Road Detection," *Workshop on Deep Learning for Autonomous Driving on IEEE 20th International Conference on Intelligent Transportation Systems*, pp. 366–371, 2017.

[18] Fauzi, M. A., & Yuniarti, A. (2018). Ensemble Method for Indonesian Twitter Hate Speech Detection. *Indonesian Journal of Electrical Engineering and Computer Science*, *11*(1), 294. https://doi.org/10.11591/ijeecs.v11.i1.pp294-299

[19] Wang, S., & Manning, C. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 90–94. Retrieved from https://www.aclweb.org/anthology/P12-2018

[20] Salton G & McGill M. J. (1983). Introduction to modern information retrieval.

[21] Hochreiter, S. and Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. *In Advances in Neural Information Processing Systems 9.* MIT Press, Cambridge MA. Presented at NIPS 96.

[22] Kastrenakes, J. (2018). "German court says Facebook's real name policy is illegal," Verge (12 February), at https://www.theverge.com/2018/2/12/17005746/facebook-real-name-policy-illegal-german-court-rules, accessed 20 June, 2020.

[23] Mozafari M., Farahbakhsh R., & Crespi N. (2019) "A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media" in International Conference on Complex Networks and Their Applications. Springer, 2019, pp. 928–940.

[24] Parikh, A., Desai, H., & Bisht, A.S. (2019): DA Master at HASOC 2019: Identification of Hate Speech using Machine Learning and Deep Learning approaches for social media post. In: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (December 2019) CEUR-WS.org/Vol-2517/T3-18