Journal of Information Systems and Informatics
Vol. 5, No. 4, December 2023 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v5i4.595 Published By DRPM-UBD

Detection of Hate Speech Code Mix Involving English and Other Nigerian Languages

Joseph N. Ndabula[1], Oyenike M. Olanrewaju[2], Faith O. Echobu[3]

[1,2,3]Faculty of Computing, Federal University Dutsinma, Katsina State, Nigeria
Email: ndabula@fudutsinma.edu.ng[1], oolanrewaju@fudutsinma.edu.ng[2], fadebiyi@fudutsinma.edu.ng[3]

Abstract

Hate speech is a recurrent event and has become a cause for global concern. The proliferation of hate speech has recently become prevalent, breeding room for violence and discrimination against specific individuals or groups. In Nigeria, message masking (use of language-mix) has become the new normal, especially in disseminating hateful and inciting comments. Hence, there is a need to curb the spread over social media. Therefore, this research focuses on detecting hate speech on social media with a code-mix of English, Pidgin and any of the three major Nigerian languages (Hausa, Igbo and Yoruba). The research used two machine learning algorithms: Support Vector Machine (SVM) and Random Forest (RF). Data were collected from tweets on the EndSARS protest and the 2023 Nigerian elections. The major features were extracted, and the text was converted into vectors using TF-IDF and Bag-of-words (BoW), which were used to train and test the model. The result showed that SVM performed better in classifying hate speech than RF on both TF-IDF and BoW features, averaging 93.43% for accuracy, 93.70% for precision, 93.43% for recall, and 93.57% for F1-score.

Keywords: Hate speech, Code-mix, Social Media, Support Vector Machine, Random Forest

1. INTRODUCTION

Hate speech is any verbal, written or behavioral communication that incites violence, denigrates a person or group, or uses cruel or prejudiced language against them based on sensitive information such as their nationality, religion, race, ethnicity, gender, health status, marital status etc. or any other protected traits [1]. It is widespread and is now seen as a threat to all individuals who abide by the law across the globe. This type of misconduct should be discouraged as it is dangerous and can hurt the targeted individual or group. Hate speech is not universally understood, and there is no general agreement on a single definition [2]. However, any speech encouraging criminal behaviour can be punished as a hate crime. It has been established that a more accurate explanation of hate speech can make annotators' tasks easier and, as a result, raise the annotators' agreement rate [3]. Although images and sounds can be used to spread hate speech, however, most hate speech posts on social media are text-based. [4]. Therefore, text classification

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

is the best approach to address this issue from a computer perspective. Machine Learning models have shown significant success in detecting hate speech. This research addresses the problem of message masking (code-mixing) with multiple languages. Existing research mainly focuses on detecting hate speech in a single language, which could be challenging and less effective if the text comprises multiple languages. Therefore, to curb this challenge, there is a need to train the models on datasets comprising several languages, especially a code mix of two or more languages. This study aims to detect hate speech in tweets with a code-mix of English and any of the three major Nigerian languages (Hausa, Igbo and Yoruba). It further contributes to developing reliable and effective models to combat hate speech and limit its spread while also upholding freedom of expression. Freedom of expression is a crucial and basic foundation for any democratic society [5].

Automatic hate speech detection is crucial to combating social media menaces, especially by applying Machine Learning techniques. [6] evaluated the performance of eight machine learning algorithms: Random Forest, Naïve Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbor, Adaptive Boosting, Multilayer perceptron and Logistic Regression, with three feature engineering techniques: Bigram, TF-IDF and word2vec using a dataset that is publicly accessible. The findings revealed that the Support Vector Machine algorithm performed the best when used with the Bigram function, with an overall accuracy of 79%. Similarly, [7] systematically examined the challenges of detecting hate speech and developed an experimental approach to identify hate speech and offensive comments. They applied inclusive and exclusive criteria to review 15 papers to determine which machine learning algorithm was mainly used for detecting hate speech and which was the most accurate. The findings demonstrated that the Support Vector Machine (SVM) emerged as the prevailing machine learning technique, while the Long Short-Term Memory (LSTM) model yielded the most favorable results. A study by [2] examines and identifies challenges associated with automated online approaches to detecting hate speech in texts. Difficulties identified include linguistic nuances, limitations of the data available for training and evaluating models, varying views of the parameters that define hate speech, and interpretability issues. They put forward a multi-view SVM methodology that provides performance close to the current state-of-the-art yet is more straightforward and facilitates the interpretation of decisions compared to neural techniques. The results showed that the multi-view SVM method outperforms the top-performing ensemble approach in accuracy and F1-score by 3.96% and 2.41%, respectively.

The research of [8] proposed an algorithm for detecting hate speech by utilizing machine learning and feature extraction approaches from text mining. The study collected hate speech data using mixed English-Odia code from Facebook and classified it into three categories. The model employed various machine learning

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

methods, including Random Forest, SVM and Naive Bayes, with feature extraction based on word2vec, word-unigrams, bigrams, trigrams, n-gram and Term Frequency-Inverse Document Frequency (TF-IDF). Two models were developed using the dataset: the binary and ternary models. Results from the research showed that SVM with word2vec features exhibited superior performance compared to the Naive Bayes (NB) and Random Forest (RF) models in relation to the binary and ternary categories. [9] trained Support Vector Classifier (SVC), Random Forest (RF), Logistic Regression (LR), Multinominal Naïve Bayes (MNB) and Decision Tree Classifier (DTC), along with n-grams feature sets extracted to learn the specific characters from a dataset of two Dravidian languages viz: Tamil and Malayalam. The model for the Malayalam language obtained an F1 score of 0.77, whereas that of the Tamil language model attained an F1 score of 0.87. [10] trained their model in English, Hindi and German. They created classifiers to classify posts into subtasks labelled Subtask-A, Subtask-B, and Subtask-C using classical machine-learning algorithms, including Multilayer Perceptron (MLP), Linear Classifier and SVM. The result shows that SVM outperforms other classifiers for English, while MLP best performs for subtasks A and B for German.

[11] conducted research employing a machine-learning approach to identify instances of hate speech within lengthy Indonesian documents, particularly on Facebook. Using the Facebook comments, they generated a novel dataset focused on hate speech in the Indonesian language. The outcomes of the experiment indicated that the implementation of TF-IDF, word unigram, charquad-gram, and lexicon features in conjunction with the Support Vector Machine (SVM) as the classifier yielded the most optimal performance, achieving an F1-score of 85%. Another study by [12] detected hate speech on Twitter based on Arabic content using Natural Language Processing and Machine learning methods. They trained the model with their dataset using Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM) and Naïve Bayes (NB). The best result was obtained using Random Forest with Term Frequency-Inverse Document Frequency (TF-IDF). An investigation conducted by [13] aims to inhibit the dissemination of hate speech via Facebook by considering textual content on public Italian pages. Two classifiers were created for the Italian language while leveraging morpho-syntactical features, lexicons based on word embedding and sentiment polarity. The classifiers include Support Vector Machines (SVM) and Long Short-Term Memory (LSTM). The result shows that the two classification approaches were effective, having an F1-score of 72%. Also, [14] developed a classifier to identify Islamophobic content on various social media platforms. In order to differentiate between non-Islamophobic content, strong Islamophobic content, and weak Islamophobic content, an automated software tool was created. Based on previous works, six different algorithms, Random Forest, Decision Trees, Support Vector Machines (SVM), Naïve Bayes, Logistic Regression and Deep Learning, were selected. The results showed that each of the six algorithms

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

performed commendably, with accuracy levels spanning from 61.23% to 72.17%. Support Vector Machine (SVM) and Deep Learning achieved the highest performance levels among these algorithms. Specifically, SVM showed an accuracy of 72.17%, better than deep learning of 1.03%.

Existing study shows that only a few researchers have worked on detecting hate speech in multilingual corpus or code-mix. Many researchers focus on single languages, which could be less effective in training the model to detect hate speech content reliably. Numerous scholars have conducted investigations on hate speech identification in diverse languages, encompassing English, Arabic, Italian, Indonesian, and Hindi. In a recent study by [15] efforts were directed towards detecting hate speech in English, Pidgin, and Hausa. However, according to our research, only a limited number of studies have been undertaken to detect hate speech in Nigerian languages. Also, a new dataset was created from tweets on the EndSARS protest and the 2023 Nigerian elections.

2. METHODS

The study is carried out using the steps shown in Figure 1. Each step is discussed in detail in this section.

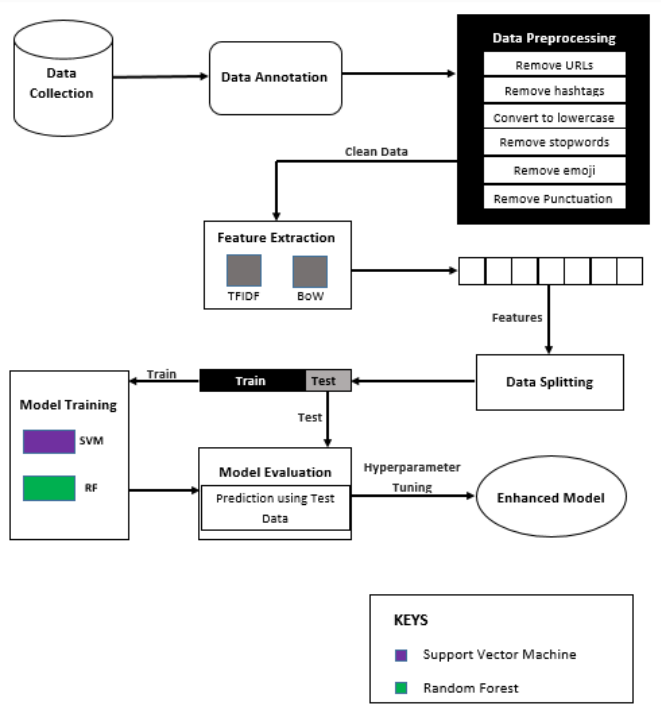

Figure 1. Research Framework

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.1 Data Collection/ Annotation

Data was collected from publicly available tweets during the ENDSARS protest in 2020 using the hashtag Endsars and the 2023 Nigerian general elections, which were used as the basis for training and testing the model. The focus was to retrieve data posted from 2020 till date relating to the ENDSARS protest, the 2023 elections and other related comments on the alleged police brutality. The data was crawled using the Twitter streaming Application Programming Interface (API) with snscrape library. After extracting the data, it was converted to a readable CSV (Comma Separated Value) format using the pandas library. An aggregate of 5,000 tweets were retrieved, of which over 2,000 tweets included pidgin, 153 tweets included Yoruba, 49 tweets included Igbo and 26 included Hausa. Table 1 shows some samples of the extracted tweets with code-mix of languages.

Table 1. Samples of extracted tweets with code-mix of English and Nigerian

Tweets	Language Combination
I hope you've not moved on and embraced apathy o. E go pain me. Hold onto it, an opportunity will present itself to make things right.	English and pidgin
Just how much is plateau state worth? Let just sell Nigeria and divide the money. What kind of oloriburukus are these ones	English and Yoruba
Na God dey protect us ooo. With the #EndSARS movement, we don automatically subscribe to vibes and insha Allah package.	English, Pidgin and Hausa
We have to end SARS now @EndSars Nna anyi ukwu Wilberforce	English and Igbo
SARS still dey operate. Who did we offend? #EndSARS	English and Pidgin
Come 2023 election make everybody buy gun keep cus itâ€™ll be bloody Once we dey voting center any political thug we see wey wan try nonsense A je kala kan ni o	English, Pidgin and Yoruba
Iwo to iwa e Koda, o tun wa anonymous message. E ni ta Fe sun to nfi epo pa Ara ni e. #EndSARS	Yoruba and English
And you were here Sir as a Governor for 8 good years, haba I Hope Uzodinma will do justice osiso on this road.	English and Igbo
Walahi ALL of YOU, their lives are on you. #EndSARS	English and Hausa
Biko, na eat I eat. I no collect money from politicians to attack peaceful #EndSARS protesters.	English, Pidgin and Igbo

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Tweets	Language Combination
They are looking for your handle to patronize you as a vendor but you don change am to #EndSARS, ebi to ma pa jan-won-jam-won eti e lo fi n sere yen.	English, Pidgin and Yoruba
Yes, people (esp northerners) abused us by calling us criminals, yahoo boys and kidnappers Daz y we were calling for #EndSARS, some started spreading it Dat we want to topple d govt suka maida zanga zangan abun adini, you know	English and Hausa

The entire dataset was compiled and manually labelled into three classes: hateful (Positive), not-hate (negative) and neutral. Care was taken to understand which statements to classify as hateful comments since hate speech is intricate and multi-faceted, making it challenging for humans and computational systems to comprehend. However, all tweets that incite hate, attack or disparage an individual or group are classified as hate. All factual and non-sentimental tweets which do not incite or attack are classified as not-hate, while tweets that are ambiguous, sarcastic or do not relate to the topic are classified as neutral. After the annotation, 751 tweets, which make up about 15.02% of the total tweets, were labelled as positive, 2644 tweets (52.88%) were labelled as negative, while 1605 tweets which make up 32.1% of the total tweets were labelled as neutral. All tweets labelled as neutral were removed, leaving a total of 3,395 tweets used for model training and validation. Figure 2 shows the distribution for each class of label.

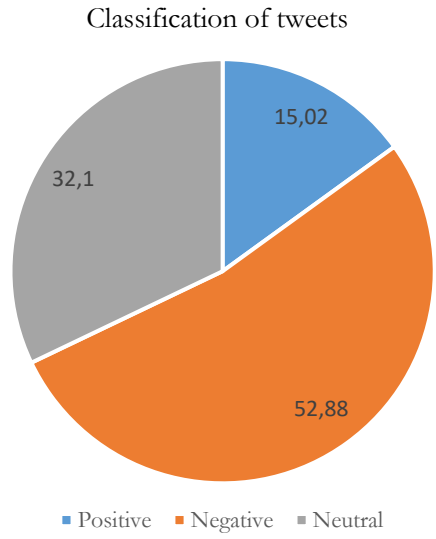

Figure 2. Class distribution of labelled tweets

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.2 Data Preprocessing

Before a dataset is ready for the analysis phase, it may contain missing values, inappropriate attribute data types, worthless attributes, and other issues that can affect the performance of the data during processing. Text preparation produces improved classification results, according to several studies. Therefore, various preprocessing approaches were used on the dataset to remove noisy, irrelevant, and non-information data. The tweets were also converted into lowercase during the preprocessing for uniformity and normalization. Pattern matching techniques were employed to remove URLs, usernames, punctuations, hashtags, white spaces, and stop-words from the collected tweets. In addition, the already processed tweets were tokenized and stemmed. Tokenization converts each individual tweet into distinct words or tokens. Subsequently, each word is converted to its root form using a Porter stemmer, for instance, the conversion of hateful to hate. Figure 3 shows the flowchart for the preprocessing.

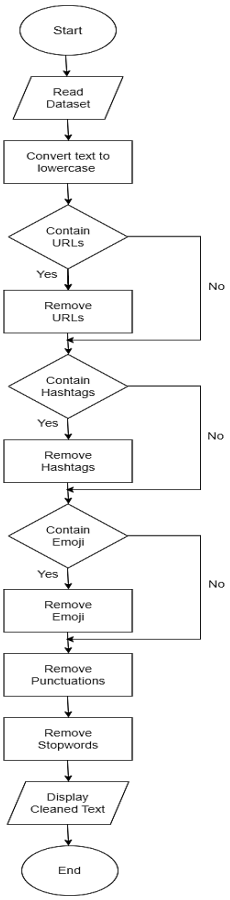

Figure 3. Flowchart for Data Preprocessing

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.3 Data Manipulation

The classification rules from the raw text are incomprehensible to machine learning systems. For these algorithms to comprehend classification rules, numerical features are required. Consequently, the key features were extracted from the raw text and represented numerically, reducing the larger dataset. In this research, two feature extraction methods were employed to transform the textual data into vectors: Bag-of-Words (BoW) and Term Frequency - Inverse Document Frequency (TF-IDF). This was done using the NLTK and scikit-learn library. The preprocessed data was split in accordance with the 80-20 principle (i.e., 80% of the data would be used to train the model while 20% would be used for the test). The classification model was trained on the classification rule using training data, while the model's performance was evaluated using test data.

2.4 Models

The "no free lunch theorem" posits that no single classifier outperforms on all datasets. As a result, different classifiers need to be used to determine which produces the best results. This study employed two traditional machine learning models, which will be discussed in detail below.

2.4.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) was introduced in 1992 by Boser, Guyon, and Vapnik. SVMs are algorithms used for supervised learning in regression and classification problems [16]. They fall under the category of linear classifiers that possess a generic nature. That is to say, Support Vector Machine (SVM) is a tool used for regression and classification prediction, which makes use of machine learning theory to increase prediction accuracy and reduce the likelihood of overfitting to the data. In a nutshell, SVMs are an excellent approach to making predictions while avoiding the issue of overfitting the model to the input data. The support vector machine came to prominence in the NIPS community and has since become a key and frequently used component of machine learning research around the world. When employing pixel maps as the input, Support Vector Machine (SVM) attains a level of accuracy comparable to sophisticated neural networks equipped with a wide range of features in a handwriting detection task [17]. Additionally, it is employed in various other applications, such as facial analysis, handwriting analysis, and others, particularly those based on pattern classification and regression. Originally designed for addressing classification problems, SVMs have recently been expanded to tackle regression problems as well [18]. Support vector machines (SVMs) are suitable for classifying linear and nonlinear data.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.4.2 Random Forest

The Random Forest algorithms are widely recognised techniques in supervised learning that combines decision trees to establish a forest, and it applies to both categorical and numerical data [19]. Throughout the procedure of creating a random forest, it swaps the bootstrap sample with the original data samples, and the number of observations in each sample remains the same as that of the original data set. Besides, one index is assigned for every data point, which helps generate bootstrap samples. A random forest is formed by a set of decision trees. In addition, the decision trees within the Random Forest may be classification or regression trees; hence, the random forest technique proves beneficial in addressing both classification and regression problems [20]. This popular approach is frequently favoured over decision tree learning because it offers multiple trained decision tree classifiers for the testing phase. In this method, sampling with substitutes helps minimize the tree's depth and maximize classification. Parallelization is another property of this technique, which brings about improved classification performance. The advantages of random forest include handling very large data, requiring very little pre-processing of data, and the data does not need normalization.

2.5 Hyperparameter Tuning

Hyperparameters are parameters whose values are used to control the learning process. Therefore, a set of optimal hyperparameter values will be selected for the learning algorithm and tuned to optimise the model to classify hate comments. For the purpose of this research, an exhaustive Grid search will be used.

2.6 Modelling Tools

Google COLAB was utilised to implement the model. Python programming language was used to write the code for preprocessing and Classification. The Scikit-Learn package was used to process the data, evaluate the results, and implement the classifiers. Graphs were plotted using the Matplotlib package, while reading of datasets and processing arrays were done using the Pandas and Numpy packages, respectively. Running python codes on local PCs (Personal Computers) is time consuming and requires high power, therefore, Google Colab was adopted to utilize cloud resources and minimize the work power of PCs.

2.7 Model Evaluation

The evaluation measures employed to appraise the model performance are accuracy, Precision, recall, and F1 measure.

a. **Accuracy:** indicates the ratio of correctly classified normal and hateful tweets to all correctly and incorrectly classified tweets. The formula gives accuracy:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

b. **Precision:** Precision is the ratio of accurately detected Hate tweets to the total number of Hate tweets. The formula gives the precision:

$$Precision = \frac{TP}{FP+TP} \quad (2)$$

c. **Recall:** Recall (also known as sensitivity) measures the classifier's ability to detect hateful tweets. The precision is calculated using the formula:

$$Recall = \frac{TP}{FN+TP} \quad (3)$$

d. **F1 Measure:** The F1 measure is a metric that indicates the balance between precision and recall measures. F1 measure is calculated as thus:

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

3. EXPERIMENTAL RESULTS

3.1 SVM

The Support Vector Machine (SVM) algorithm achieved an accuracy rate of 93.80%, a precision of 94.26%, a recall of 93.80%, and an F1-score of 94.02% when using the TF-IDF features. Conversely, when using the BoW features, the SVM algorithm attained an accuracy of 93.07%, a precision of 93.16%, a recall of 93.07%, and an F1-score of 93.12%. Table 2 provides an overview of the SVM performance on our Dataset, encompassing the TF-IDF and BoW features.

Table 2. Summary of SVM performance on TF-IDF and BoW features.

Feature Extraction	Accuracy	Precision	Recall	F1-score
TFIDF	93.80	94.26	93.80	94.02
BoW	93.07	93.16	93.07	93.12

3.2 Random Forest

The Random Forest (RF) algorithm with the TF-IDF features obtained an accuracy rate of 91.31%, precision rate of 94.37%, recall rate of 91.31% and F1-score of 92.61%, while with the BoW features it obtained an accuracy of 78.86%, precision of 92.58%, recall of 78.86% and F1-score of 84.60%. The summary of

RF's performance on our Dataset for both TF-IDF and BoW features is presented in Table 3.

Table 3. Summary of RF performance on TF-IDF and BoW features.

Feature Extraction	Accuracy	Precision	Recall	F1-score
TFIDF	91.31	94.37	91.31	92.61
BoW	78.86	92.58	78.86	84.60

3.3 Optimized Random Forest

In order to optimize the performance of the Random Forest algorithm, which exhibited the lowest performance when applied to Bag of Words features, hyperparameter tuning was conducted. Specifically, the hyperparameters were adjusted, including min_samples_leaf, n_estimators, max_depth, max_features and min_samples_split, using the grid search optimization technique. The hyperparameter value options were set as shown in Table 4.

Table 4. Hyperparameter value options for tuning RF

Hyperparameters	Value Options
max_depth	[None, 10, 20, 30]
n_estimators	[50, 100, 200]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
max_features	[‘auto’, ‘sqrt’]

After the tuning of the hyperparameters, it could be seen that there was an improvement in the accuracy and recall from 78.86% to 87.67%. Also, the F1-score improved from 84.60% to 89.87%. There was an 8.81% increment in the accuracy and recall value of the algorithms' performance and a 5.27% increase in the F1-score. Table 5 shows the comparison of the Random Forest classifier before and after the optimization.

Table 5. RF performance on BoW features before and after optimization

Evaluation Metrics	RF (Before optimization)	RF (After optimization)
Accuracy	78.86	87.67
Recall	78.86	87.67
F1-score	84.60	89.87

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3.4 Discussion and Comparison

3.4.1 SVM

It was observed from table 2 and figure 4 that SVM performed better with TF-IDF features than with BoW features across all metrices, having a difference of 0.73% in accuracy, 1.1% in precision, 0.73% in recall, and 0.9% in F1-score. It is therefore evident that the TF-IDF features extracted from the dataset were more suitable for training the SVM model on hate speech classification than BoW features. Figure 4 visualizes the summary of the SVM performance on TF-IDF and BoW features.

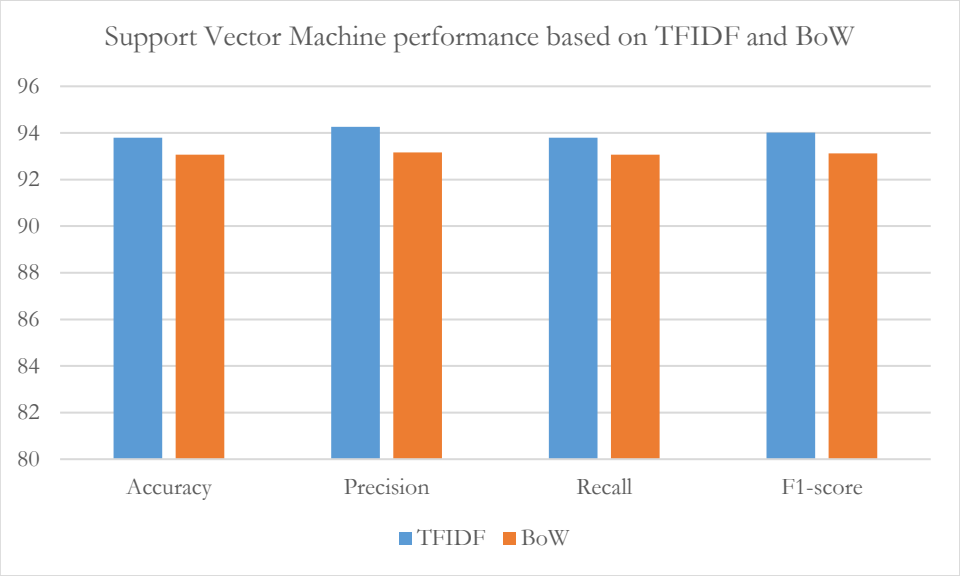

Figure 4. Support Vector Machine performance on TF-IDF and BoW features

3.4.2 Random Forest

It was observed from table 3 and figure 5 that the Random Forest classifier also performed better on TF-IDF features than with the BoW features across all metrices, having a difference of 12.45% in accuracy, 1.79% in precision, 12.45% in recall, and 8.01% in F1-score. Therefore, it is evident that the TF-IDF features extracted from the dataset were more suitable for training the Random Forest model on hate speech classification than the BoW features. Figure 5 visualizes the summary of the RF performance on TF-IDF and BoW features.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

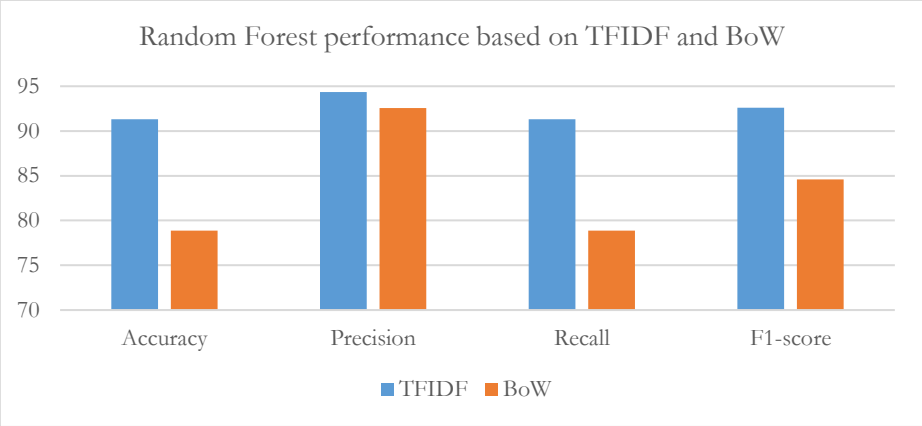

Figure 5. Random Forest performance on TF-IDF and BoW features

3.4.3 Comparison of Models

From the comparison in Table 6 and Figure 6, it was noticed that SVM performed better than RF on both BoW and TF-IDF features. It can also be observed that both SVM and RF classifiers performed better on TF-IDF features than they did with the BoW features.

Table 6. Comparison of SVM and RF classifiers on both TF-IDF and BoW features

Classifier/Features	Accuracy	Precision	Recall	F1-score
SVM-TFIDF	93.80	94.26	93.80	94.02
SVM-BoW	93.07	93.16	93.07	93.12
RF-TFIDF	91.31	94.37	91.31	92.61
RF-BoW	78.86	92.58	78.86	84.60

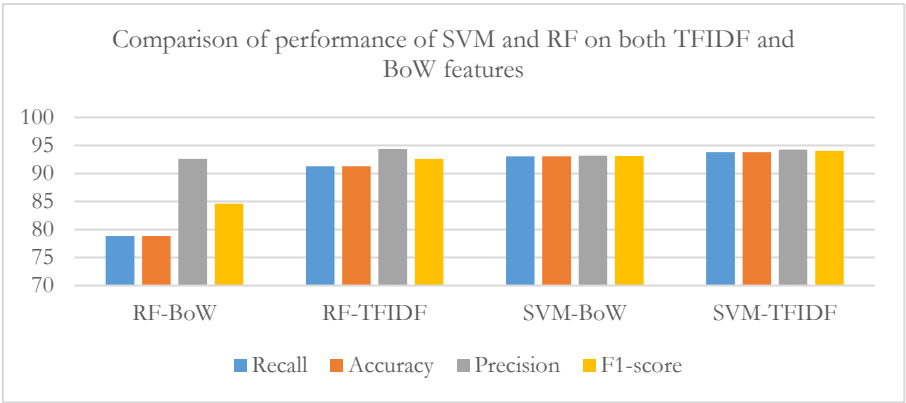

Figure 6. Summary of comparison of SVM and RF on TFIDF and BoW features

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3.4.4 Optimized Random Forest

After performing a hyperparameter tuning on RF classifier with BoW features using the Grid search technique, the model's performance was enhanced compared to the un-optimized version. It was observed that there was an 8.81% increment in the accuracy and recall value of the model's performance and a 5.27% increase in the F1-score. Figure 7 visualizes the performance of the RF classifier on BoW features before and after optimization.

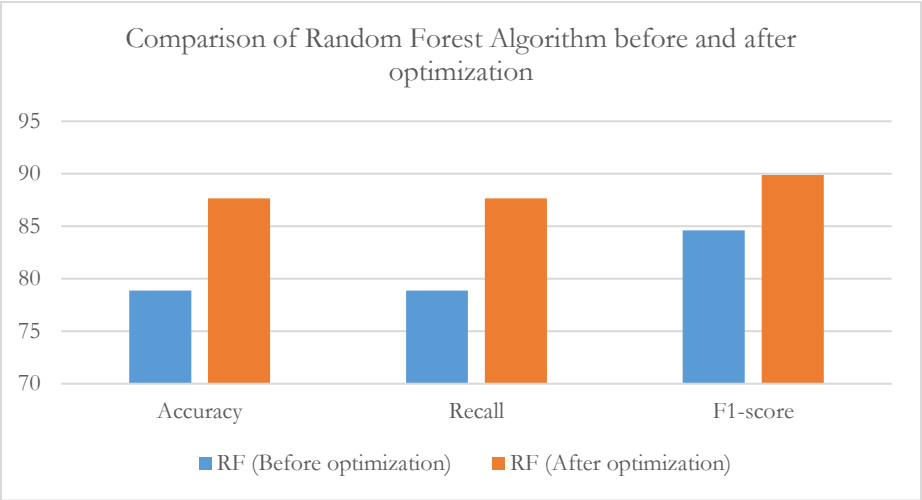

Figure 7. Visualization of RF performance on BoW features before and after optimization

4. CONCLUSION

The classification of hate speech has led researchers to develop numerous techniques to combat the problem. This study used automatic text classification techniques to detect hateful messages on social networks. The aim was to detect hate speech with a code-mix of English, Pidgin and any of the three major Nigerian languages (Hausa, Yoruba and Igbo). Dataset for the research was created from tweets during the EndSARS protest and the 2023 Nigerian general elections. The study also compared two machine learning algorithms (SVM and RF) and two feature extraction techniques (TF-IDF and BoW). The experimental result showed that the SVM algorithm outperforms the RF algorithm in hate speech classification on both BoW and TF-IDF features. More so, it was observed that both SVM and RF algorithms performed better on the TF-IDF features compared to the BoW features. The experiment obtained the best result using SVM on TF-IDF features, while the least performance was obtained from RF on BoW features. However, hyperparameter tuning was performed using the Grid

search optimization technique to enhance the performance of the RF algorithm on BoW features. After the optimization, the optimized RF improved 8.81% in accuracy and recall and a 5.27% improvement in F1-score over its initial performance on the BoW features. A major limitation of this research is that the model cannot identify the severity of hate. Therefore, future research can improve the model to predict the severity of hate in a code-mix text. Also, other languages could be explored, especially since Nigeria has so many languages. Finally, multimedia data can also be explored to detect hate speech content.

REFERENCES

[1] A. Guterres, "United nations strategy and plan of action on hate speech," United Nations, New York, NY, USA, 2019.

[2] S. MacAvaney, H. R. Yao, E. Yang, K. Russell, N. Goharian and O. Frieder, "``Hate speech detection: Challenges and solutions," PLoS ONE, vol. 14, no. 8, pp. 1-16, 2019.

[3] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky and W. Wojatzki, "Measuring the reliability of hate speech annotations: The case of the European refugee crisis," in Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication, Bochum, Germany, 2016.

[4] C. E. Ring, "Hate speech IN social media: An exploration of the problem and its proposed solutions," Colorado, 2013.

[5] E. C. o. H. Rights, "Annual Report 2017 of European Court of Human Rights, Council of Europe," ECHR, Strasbourg, France, 2017.

[6] S. Abro, S. Shaikh, Z. H. Khand, Z. Ali, S. Khan and M. Ghulam, "Automatic Hate Speech Detection using Machine Learning: A Comparative Study," International Journal of Advanced Computer Science and Applications, (IJACSA), vol. 11, no. 8, pp. 1-8, 2020.

[7] C. E. R. Salim and D. Suhartono, "A Systematic Literature Review of Different Machine Learning Methods on Hate Speech Detection," International Journal on Informatics Visualization, vol. 4, no. 4, pp. 1-6, 2020.

[8] S. K. Mohapatra, S. Prasad, D. K. Bebarta, T. K. Das, K. Srinivasan and Y.-C. Hu, "Automatic Hate Speech Detection in English-Odia Code Mixed Social Media Data Using Machine Learning Techniques," Applied Science, vol. 11, pp. 1-21, 2021.

[9] V. Pathak, M. Joshi, P. A. Joshi, M. Mundada and T. Joshi, "Using Machine Learning for Detection of Using Machine Learning for Detection of Social Media text," KBCNMUJAL, pp. 1-12, 2020.

[10] H. Nayel and H. L. Shashirekha, "DEEP at HASOC2019: A Machine Learning Framework for Hate Speech and Offensive Language Detection," in FIRE 2019, Kolkata, India., 2019.

[11] N. Aulia and I. Budi, "Hate Speech Detection on Indonesian Long Text Documents Using Machine Learning Approach," in International Conference on Computing and Artificial Intelligence (ICCAI), Bali, Indonesia, 2019.

[12] I. Aljarah, M. Habib, N. Hijazi, H. Faris, R. Qaddoura, B. Hammo, M. Abushariah and M. Alfawareh, "Intelligent detection of hate speech in Arabic social network: A machine learning approach," Journal of Information Science (JIS), vol. 47, no. 4, pp. 2-19, 2021.

[13] F. D. Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi and M. Tesconi, "Hate me, hate me not: Hate speech detection on Facebook," in In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy, 2017.

[14] B. Vidgen and T. Yasseri, "Detecting weak and strong Islamophobic hate speech on social media," Journal of Information Technology & Politics, pp. 1-14, 2019.

[15] S. M. Aliyu, G. M. Wajiga, M. Murtala, S. H. Muhammad, I. Abdulmumin and I. S. Ahmad, "HERDPhobia: A Dataset for Hate Speech against Fulani in Nigeria," arXiv preprint arXiv:2211.15262., pp. 1-3, 2022.

[16] M. Awad and R. Khanna, "Support Vector Machine for Classifiaction," in Efficient Learnhing Machines, Berkeley, CA., Apress, 2015, pp. 39-66.

[17] A. W. Moore, "Tutorials," 19 February 2020. [Online]. Available: http://www.cs.cmu.edu/~awm/tutorials.html. [Accessed 19 February 2020].

[18] V. Vapnik, S. Golowich and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in In M. Mozer, M. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems 9, Cambridge, MA, 1997.

[19] R. Sutton and A. Barto, Learning: An Introduction, 1998.

[20] N. Mohapatra, K. Shreya and A. Chinmay, "Optimization of the Random Forest Algorithm," in Advances in Data Science and Management. Lecture Notes on Data Engineering and Communications Technologies, vol. 37, Singapore, Springer, 2020, pp. 201-208.