Detection of Hate Speech Texts Using Machine Learning Algorithm

1st Mahamat Saleh Adoum Sanoussi
School of Information Engineering
Huzhou University
Zhejiang, China
email: ammsas@gmail.com

2nd Chen Xiaohua
School of Information Engineering
Huzhou University
Zhejiang, China
email: 97122046@sina.com

3rd George K. Agordzo
School of Mathematics and Big Data
Anhui University of Science and Technology
Anhui, China
email: gkk29667@gmail.com

4th Mahamed Lamine Guindo
College of Biosystems Engineering
Zhejiang University
Hangzhou, China

5th Abdullah MMA Al Omari
School of Information Engineering
Huzhou University
Zhejiang, China

6th Boukhari Mahamat Issa
Department of Electrical Engineering
Abeche Institute of Sciences and Technologies
Abeche, Chad

2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) | 978-1-6654-8303-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCWC54503.2022.9720792

***Abstract*—Identifying hate speech on social media has become increasingly crucial for society. It has been shown that cyberbullying significantly affects the social tranquillity of the Chadian population, mainly in places of conflict. This article aims to detect hate speech for texts written in "lingua franca", a mix of the local Chadian and French languages. The dataset consists of 14,000 comments extracted from the most visited Facebook pages and annotated in four categories (hate, offence, insult and neutral) were used for this study. The data were cleaned by Natural Language Processing techniques (NLP) and applied to three word embedding methods such as Word2Vec, Doc2Vec, and Fasttext. Finally, four Machine Learning methods, namely Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbours (KNN), were computed to classify the different categories. The result showed that FastText features representation as input to SVM classifier was the best with 95.4% accuracy for predicting the comment contained insult statement followed by hate statement 93.9%. The result demonstrated our model could be used to detect the hate speech made by Chadians on social media texts.**

***Index Terms*—hate speech, natural language processing, social media, text classification, word embedding**

I. Introduction

According to recent statistics, more than half of the worldś population will be using social media by the middle of this year [1]. This increase is mainly due to the accessibility of the internet in many countries. Chad is one of the countries, which also see the rise of users. Social media platforms generate enormous amounts of data due to tweets and comments made by users. Analyzing the behavior of individuals in social media can forecast what people post and what they comment on, share, and like [29].

Unfortunately, social networking is frequently exploited to disseminate hate and misinformation.

The term "hate speech" refers to any form of abusive writing or intimidating language that express bias against a particular group based on their race, religion, political affiliation, etc. Most of the social media platforms had elaborate policies and procedures to address the issue of identifying hate speech. Despite these regulations and protocols, it is challenging to restrict specific inappropriate comments that contain hateful text. For example, Facebook does not tolerate hate speech and sensitive content to avoid exclusion[2]. This challenge made hate speech and offensive languages attract researchers due to their spread in social media [1].

Chad is located in central Africa and has over 200 ethnic groups. Each group contributes to the diversity of the country's social structure in terms of culture and language [23]. A lingua franca is a common language used to communicate between groups of people who speak distinct dialects. French and Arabic are the most frequently used languages on social media. Besides the two languages mentioned above, there are numerous words and phrases written in dialects. For example, the appearance of these words "kirdi", "abid", "birti", "doum", "damboula", etc., all have a pejorative perception.

This work aims to capture and prevent the alarming spread of hate text on Facebook. News portal pages, political leaders, celebrities, and political activists leverage massive followings; their posts frequently generate substantial discussion and sharing. Significant differences in perspective, commonly used to divide people along political, religious, ethnic, and regional lines, often result in offensive comments or hatred. Unchecked hate tweets have the potential to do significant damage to our society and disproportionately harm marginalised people or groups [18].

The previous study has been carried out identification and classification of hate speech on categorised or labelled dataset

[1] https://www.statista.com/statistics/2784

[2] https://www.facebook.com/communitystandards

978-1-6654-8303-2/22/$31.00 ©2022 IEEE

(hateful, offensive, sexism, insult, racism) in single language [13] [32] [10] and code mixed data [30].

Various Machine Learning (ML) models have been investigated; unsupervised methods rely on lexicon-based approaches to deduce sentiment polarity from corpus data. In contrast, supervised approaches rely heavily on labelled data and Natural Language Processing (NLP) techniques to train a learning algorithm. There are both approaches supervised and unsupervised such as SVM,Naive Bayes, KNN, and Logistic Regression, widely used, which have demonstrated promising results [3] [5] [14]. This research makes two significant contributions:

- Provides a dataset of 14k annotated for hate speech in Chadian French-Arabic texts;
- Introduces a ML model with hyperparameter tuning optimisation to accurately classify every comment and detect hate speech texts.

In terms of performance, the model outperformed previous works. As features, our design used Natural Language Processing (NLP) techniques for data preprocessing and word embedding techniques to learn semantics between words. In evaluating the performance of several word vector representations techniques Word2Vec, Doc2Vec, and FastText are used in conjunction with various classification algorithms. We believe that future researchers working to advance the state-of-the-art in mixed-language hate speech detection can refer to the findings of this study.

The following are the main points of this article: section II discusses the related research on hate speech detection. section III introduces data collection and text corpus annotation of Chadian mixed-language posted on Facebook to detect hate speech. section IV describes the proposed methodology of our study and reports the experiment's performances. Finally, section V concludes the paper and makes recommendations for future work.

II. Related Work

Up to this point, the vast majority of research on cyberbullying has focused on monolingual dataset. The most popular data used to detect hate content from tweets is in English; Waseem et al. [32] studied approximately 17,000 tweets annotated in three labels: racism, sexism, and none. They analysed the impact of various extra-linguistic features in conjunction with character n-grams for hate speech detection. Their experiment on logistic regression classifier and 10-fold cross-validation to test the influence of multiple features on prediction performance showed a 74% accuracy score. They were Followed by Davidson et al. [13], which includes 25,000 tweets annotated in three categories hate, offensive, and neither. The technology which are used for features extraction was TF-IDF then compared several classification models. They found that logistic regression with L1 regularisation performed 91% precision using five-fold cross-validation.

R. Martins et al. [21] used a combination of emotional approaches through sentiment analysis lexicon-based and machine learning classification algorithms to predict hate speech contained in a text. They performed on datasets containing 975 preprocessed tweets; the result showed about 80.56% accuracy for hate speech identification.

Currently, S. Agarwal et al. [2] reviewed the popular annotated datasets then proposed an ensemble learning-based adaptive model to improve the cross-dataset for hate speech detection in the English language. They used A-Stacking, a hybrid classifier based on ensemble learning, to cluster the Tweets with Recurrent Neural Network (RNN). Dense-vector representations of COVID-19 related datasets indicate the models' performance with an F1 score of 89.9%.

Secondly, other studies used datasets other than English languages, Pereira-Kohatsu et al. [26] presented HaterNET as an intelligent system to monitor and visualise Twitter content for hate speech detection in Spanish. The corpus comprises 2M tweets filtered in absolute hate or relative. They combined features such as embeddings of words, emojis, and tokens, then used TF-IDF with classification models LSTM+MLP achieved 82.8% performance.

Fesseha et al. [15] studied low-Resource Language as Tigrinya for text classification. The experiment has been done on 30,000 social media texts manually annotated in a single label, and then comparing different word embedding techniques with classification algorithms has improved the accuracy result. The CBOW CNN with Word2Vec for feature extraction proved the best accuracy of 93.41%.

Vargas et al. [31]proposed a lexicon-based approach for hate/non-hate speech and offensive/non-offensive language detection for 7000 comments annotated in the Brazilian Portuguese language scraped from Instagram. They carried several experiments in which the promising result of learning methods was NB and MLP(Multilayer Perceptron) with Bag-of-words Multilingual Offensive Lexicon as input features-selection respectively gave 89% and 85% performance.

The detection of hate speech in Vietnamese social media texts has been made possible by Son T. Luu et al. [20]. They proposed a deep learning approach with a BERT-base-multilingual-case pre-trained model that achieved 86.88% accuracy for dataset over 30k comments annotated on three labels which are clean, offensive, and hate.

Sentiment lexicon approaches have been used in the Arabic language texts for hate speech detection [8] [7] [25] using Twitter and movie review dataset. One of the early studies on Arabic abusive language detection was by Albadi et al. [4], who compared various classification models with deep learning. The experiment found that a simple Recurrent Neural Network (RNN) architecture with Gated Recurrent Units (GRU) and pre-trained word embedding can detect religious hate speech with 84% accuracy in Arabic Twittersphere for 6k tweets. Aljarah et al. [6] approached the Random Forest classifier with TF-IDF emotions features that improved the previous performance with 88.6% accuracy. They experimented with tweets as dataset related to racism, journalism, sports orientation, terrorism, and Islam to detect violent expressions or phrases in Arabic social networks.

Recently, Safa et al. [7] built Arabic textual corpus crawled

from Twitter and labelled the data on the six-class distribution as clean, religious hate, gender hate, nationality hate, and ethnicity hate. The Machine and deep learning techniques developed numerous two-class, three-class, and six-class classification models combined with various feature extraction techniques (word embedding). The performance results of the different learned models and misclassification errors were 87% F-macro score for CNN with multilingual BERT features.

For their research, H. Aung et al. [9] analysed the sentiment in Myanmar text scraped from Facebook comments to determine the sentiment class positive, neutral, or negative. First, they studied the performance of word embedding techniques Word2Vec, TF-IDF and pre-trained Word2Vec, in which Word2Vec showed a better representation of word vectors. Secondly, after evaluating Logistic Regression against SVM and Random Forest, they found that it performed better in accuracy and F-measures than the other two ML methods with 80% performance.

Furthermore, these researches are some of the few that deal with hate speech detection in multilingual and multiple aspects, such as N. Ousidhoum et al [24], who analysed 13k tweets in English, French, and Arabic separated data. Their comparison study showed that the deep learning approach performed well than the classic BOW+Logistic Regression model. They reported that classification in single task single language (STSL) model consistently outperforms the multilingual multitask (MTML) model for directness tweets in EN 0.94, FR 0.8, and AR 0.84 F1 macro scores.

Sreelakshmi et al. [30] extracted 10k mixed Hindi-English data samples from Facebook to detect hate speech in texts.They used word embedding FastText method and trained with SVM-Radial Basis Function (RBF) classifiers; the result showed better feature representation 85% performance.

III. DATASETS

Dataset is significant in this study since it enables data processing and features extraction. Building a corpus of negative comments is one of the most critical tasks. We tackled this issue as sentiment lexicon baselines of mixed-language text corpora based on the Chadian context to compile labelled dataset focused on sensitive words obtained by crawling Facebook's pages. It contributes significantly to the difficulty to detect automatically misspelt texts or including paralinguistic signals [18]. We also note that the texts written in more than one mixed language constitute another difficulty to label the comment or tweet as hate or neither [13], especially when the target languages lack training data. The datasets gap still needs to be investigated, as shown in this article [11], that built a formal model to detect hate in French corpora.

A. Data collection

Aiming to detect hatred texts posted by Chadians on Facebook, we have collected comments from the Facebook public pages of newspapers, politicians, activists and groups. These Facebook pages were hand-picked based on their large followings, regularity of posts and comments, and the general buzz they generate.

An open-source Facepager API (Application Programming Interface) [17] was used to extract posts and comments from Facebook pages using web scraping techniques. Facebook's Graph API was developed to create a node for ten pages and link users in a database. Through https://findmyfbid.in/, we were able to find each page's unique identifier, which used to create the network's various nodes. The data was scraped from these nodes twice: in January 2021, 151,377 rows, and in April 2021, 131,918 rows of unfiltered data were scraped. Random posts from each page were gathered sequentially, and each post dragged the comments along with it. For the sake of directness, only posts that could potentially stimulate a heated debate were included in this massive data set. For this work, the data exported in CSV format has been separated into two files: train and test. The training file is designated to annotate the comment and fit the learning model, while the test file was used to score each comment's probability prediction. Table I demonstrates the data size collected for this study.

TABLE I
DATA COLLECTED

Datasets	Posts & Comments
Train file	7002
Test file	7001
Total	14003

B. Annotation

Annotation is labelling added to a text that helps readers understand and retain information, and this is the essential step in identifying the negative comments. Since the data collected was written in mixed languages, it was difficult to judge a particular statement without considering many facets of the texts.

We requested an additional team of four (4) persons to vote for the ambiguous comments. They were assigned a task to categorise each comment, including hate speech, offensive language, insult, and neutral. The majority agreement of the annotators determines the final decision to classify the comments.. The annotations applied to the training file resulted in a multi-label classification problem. Due to a lack of a hate lexicon for Chadian texts and the need for appropriate annotation, the search query includes phrases or terms such as political leader or party, southern vs northern, ethnic group, and religion (Muslims & Christians). The result obtained through a list of keywords well known as hate, offensive, and insult established to filter the data is shown in Table II.
Table III shows the samples extracted from the datasets and

TABLE II
ANNOTATED DATASETS DESCRIPTION

Data/Label	Hate Speech	Offensive	Insult	Neutral	Total
Posts	3	77	4	1627	1710
Comments	416	1165	322	3379	5292

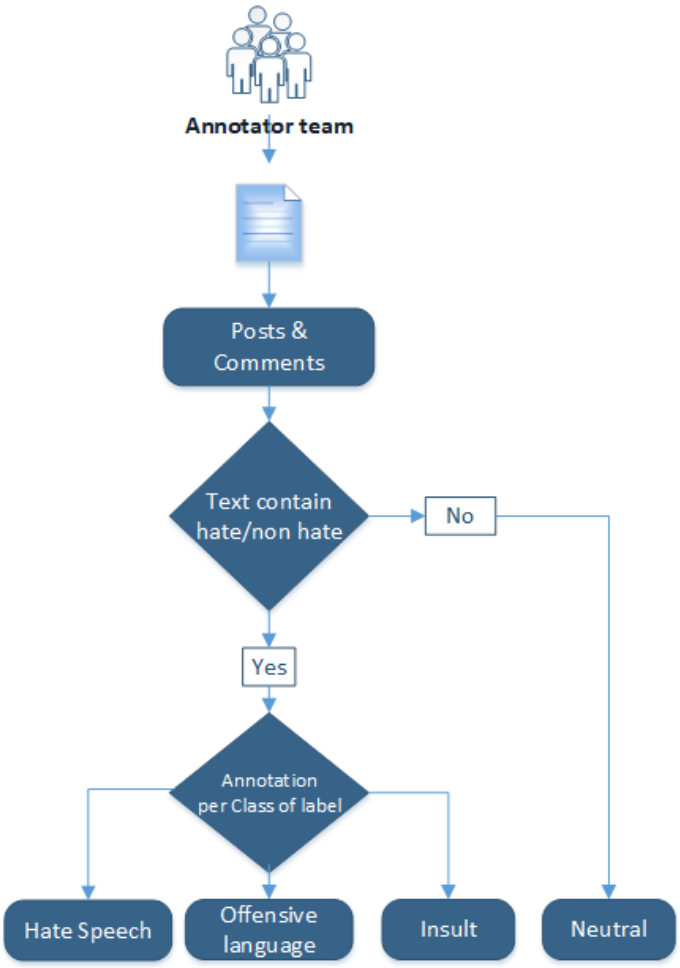

Fig. 1. Label decision

the categorisation that the annotators have done for mixed-language texts among one of the four categories.

The datasets labelled under four-class available to the community and researchers in our GitHub repository, open to the public[3]

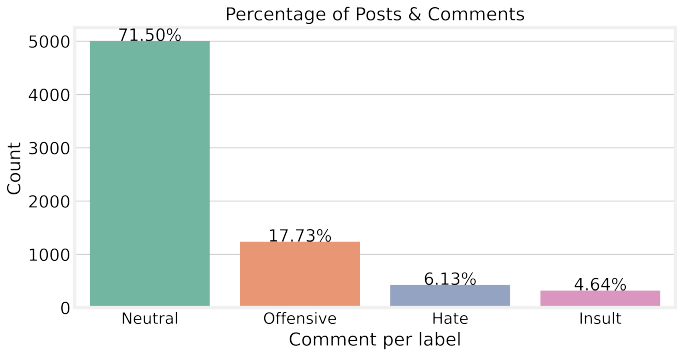

Fig. 2. Statistic of labelled data

[3]https://github.com/m-sas88/datasets_for_hate_speech

TABLE III
SAMPLE OF ANNOTATED DATASET

Class	Posts & Comments	Translate
Hate	(1) Inti ma kirdi sakite tu peux dire tout ce que tu veux parce que tu lèches les culs de français et sans la France t'es rien. Espèces de retardé mental que t'es. (2) Ça il faut demander aux dirigeants c'est pas mon problème dixit Tregoat. Wadam tiss da Inti da fi Europe da Reims U05 koula ti entraîner wa ? Abouk	(1) You are just a simple disbeliever; you can say whatever you want because you lick French asses, and without France, you are nothing. Mentally retarded in some ways. (2) It is necessary to question the leaders; Dixit Tregoat, this is not my problem. Can you son of a bitch coach a Reims U5 club? your father
Offensive	(1) Kan antetak ana abouy farak (2) Haramine sakit mais dirige opprimant le peuple.	(1) My father would be a bastard if I gave you (2) bandits who rule oppressing the people
Insult	(1) Le mot me manque !!! Sara da Sara birti da birti comme disait le Maréchal damboula hanakou ??????vous êtes les ennemi de développement du Tchad !!mps fi raskou bes yagot wilettt boy (2) Kirdi olo yoskou	(1) I miss the word !!! Sara still just Sara, a slave always remains slave as the Marshal Said your ass. You are the enemy of Chad's development; the MPS will dominate you, domestic boy. (2) You are a disbeliever black ass
Neutral	(1) Baddoulou rouaba achourou zibdé. Je m'en souviens encore. (2) En tout cas ana ma nahadji chey dans sa vous n'allez pas mange de piment dans ma bouche kadap hanakou	(1) Exchange yoghurt, then buy margarine; I still remember it. (2) In any case, I will not say anything; you are not going to eat chilli in my mouth; you lied.

IV. Proposed Methodology

In this section, we discussed the proposed methodology for hate texts classification. Hate speech detection is usually considered a supervised classification problem where the learning algorithms are trained on label dataset (texts) that comprise some form of abusive and hateful language. The success of these machine learning algorithms depends on their ability to understand complex models and non-linear relationships within data. However, researchers face difficulties identifying appropriate structures, architectures, and techniques for text classification [19].

Figure 3 describes the general process of the proposed method: To begin, text data are gathered and annotated. Secondly, text preprocessing is performed on the dataset. Thirdly, a model of feature representation is built. Fourth, using labelled data to build and train the machine learning classifier. After training our model, we conducted a new classification using unlabeled data to determine how well this classifier predicted new data (test file) within the predefined categories of hate speech

texts. Each stage consists of data cleaning text preprocessing, features extraction, and classification. We choose a set of optimal hyperparameter tunings for each learning classifier to improve the accuracy score. Finally, a comparative result is generated based on the accuracy score and optimal parameter combination to establish the most accurate supervised classification algorithm.

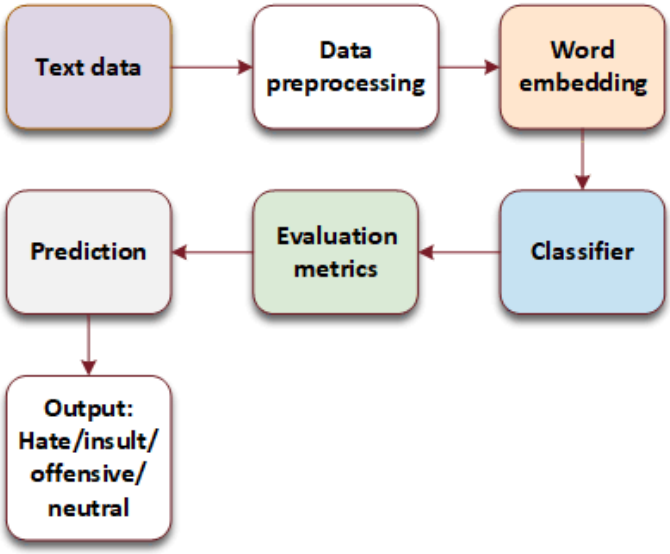

Fig. 3. Proposed methodology

A. Text preprocessing

Preprocessing is critical in any Natural Language Processing (NLP) task. The data obtained from Facebook contains much noise and many waste substances, which must be cleaned from each row of comments [27]. The following are included in this process data cleaning and NLP steps.

1) Data cleaning: This stage involves removing redundant data and spelling errors from our obtained corpus. As a result, the python package re (regular expression) removed special characters, user names, hashtags, Emoji, and URL links and converted the text to lowercase.

2) Natural Language Processing: To evaluate text data, we used natural language processing techniques such as tokenisation, stopwords, and lemmatisation to extract meaning from the text to emphasise the words that serve as the lemma for each word in the corpus. An NLP pipeline built using the python libraries NLTK and Spacy to perform the data.
Tokenisation consists of splitting the text into words (tokens) separated by whitespaces or punctuation characters, and stopwords are like pronouns, articles, and prepositions used frequently in Facebook's comments. Although stopwords are useless for text mining, we completely removed them from the corpus and did not apply any additional spelling corrections to the data. Figure 4 and Figure 5 give a data visualisation for the most frequent words present in hate and insult comments.

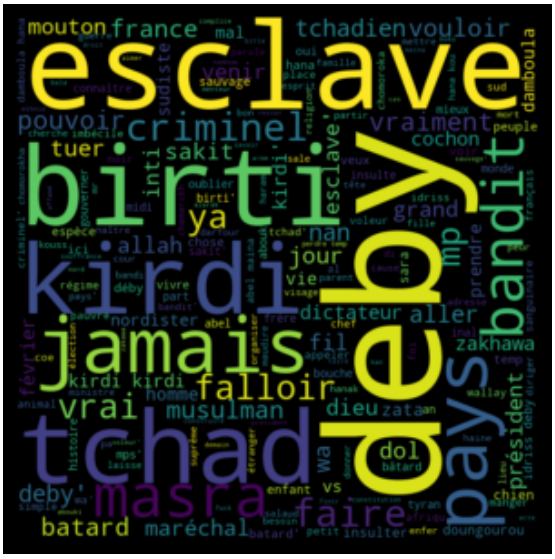

Fig. 4. WordCloud for most frequent words in Hate Comments

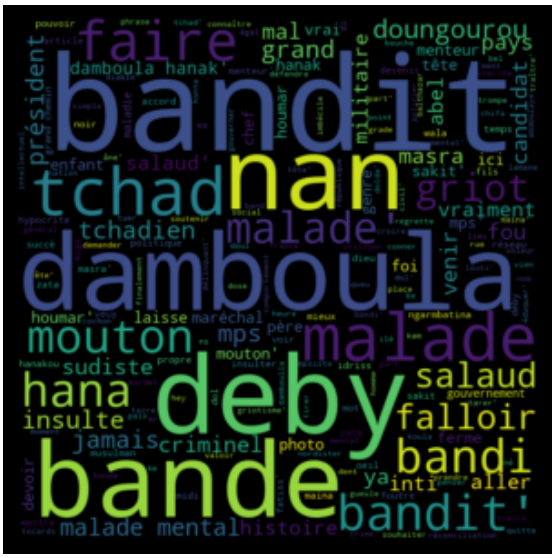

Fig. 5. WordCloud for most frequent words in Insult Comments

B. Feature extraction

The feature extraction approach identifies the most significant features for classification, followed by reducing redundant data and dimensionality. Feature extraction is required for NLP tasks since ML algorithms only accept numerical features of text data as input. We investigated the effect of comprehensive linguistic feature engineering on the detection of hate speech through tweet comments [28]. Three word embedding techniques have been studied such as Word2Vec, Doc2Vec and FastText [15] [9] [27] [16] to convert each token(feature) from the vocabulary to a vector of a real number. These feature extraction techniques were compared in our experiment to determine which approach produced a more accurate representation of the word.

1) Word2Vec: The Word Representation in Vector Space (Word2Vec) [22]takes words from a vocabulary as input and embeds them as vectors into a lower-dimensional space. This technique demonstrated its utility in NLP, it enables the identification of semantic connections between words by giving the context of a word and predicting the target word or surrounding it.

The library of Genism Python is used to import the Word2Vec module with two types of training methods CBOW and skip-gram. We used CBOW with the following parameters for model training and corpus building. The lowest number of words set is to two (2), the context window is the highest distance between the current and expected words assigned to five (5), the expected whole layer up to 300, and the epoch set to 30. The entire datasets train and test set were tokenised and used to build the model vocabulary, resulting in a total of 56163 and 69370 features in each dataset, respectively.

2) Doc2Vec: Doc2Vec or Paragraph Vector is an extension to Word2Vec aims at learning how to project a document into a latent D-dimensional space then convert sentences or paragraphs to vectors. The following parameters were fixed to the module, vector size set to sixty-four (64), window size supposed the maximum context location at which the words need to be predicted set to 5 and 40 for the number of passes epoch. To create the corpus for the Doc2Vec model, the train and test sets were used to feed the models, which accepted the entire document and converted it to vector representations of 300-size dimensions.

3) FastText: FastText is a library for learning word embedding and text classification created by Facebook's AI Research lab[4] . A numerical vector is associated with each character n-gram to train models on large corpora [12]. We used the pre-trained model then optimised the hyperparameters to find a better similarity of words. The vector size is set to the entire layer 300, and the window size is assigned to five (5). To develop FastText vocabulary, we followed the same procedure as done in Word2Vec and Do2Vec models.

C. Classification models

This subsection discusses the classifiers used for our experiments. To find the best model with a high prediction score, we combined feature vectors obtained from three (3) word embedding techniques with various machine learning classification algorithms, including Logistics Regression (LR), Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbours (KNN) [2]. These classifiers are supervised learning approaches in which the model predicts discrete output variables based on input variables. Assuming that our task is binary classification, the purpose is to classify the comment into four distinct categories.

1) Hyperparameter Tuning: The Hyperparameter technique determines the parameters necessary to improve the model's performance. This step is critical in machine learning because it enables the creation of the optimal model. The optimal hyperparameter is those cause the least amount of error in the validation set. GridSearchCV is the module that has been used to create a model for each combination of tune parameters.

2) Evaluation metrics: This section evaluates our model result based on two metrics: True prediction or False prediction for binary classification. The performance of each model is based on the counts of hate comments correctly and incorrectly predicted. The confusion matrix provides more insightful pictures of how the four (4) classification metrics are calculated: True Positive(TP), False Positive(FP), False Negative(FN), and True Negative(TN).

D. Experiments results

In this section, we discuss the approaches used and the results achieved. This experiment is carried out using Google Research Colaboratory, also known as Colab Notebook, which enables the writing and execution of Python code online. We imported the data science libraries Pandas, Numpy, Matplotlib, Scikit-learn, and NLP tools such as Gensim, NLTK, and Spacy.

We conducted several experiments to check the performance combination of word embedding methods as input to different classifiers to detect hate text. The dataset was split into 70% (per cent) to train the model and 30% (per cent) for the test. Repeated K-Fold cross-validation was adopted to estimate the performance of our dataset divided into five (5) folds and the number of repeats fixed to two(2). The model built based on four categories (hate, offensive, insult, and neutral) means each row of Comments belongs to one of the classes $if\begin{cases} True\ 1 \\ \\ False\ 0 \end{cases}$. Validation of the model was performed to categorise the probability prediction for each class label.

The results for each model are presented in distinct subheadings because our principal purpose is to compare the performance of classifiers and features strategy for prediction. The tables below summarise the performance metrics results achieved in our experiments for each model optimised with hyperparameter tuning.

From the results obtained in Table IV, Table V, Table VI,

TABLE IV
EXPERIMENT I PERFORMANCE RESULT(ACCURACY)

Feature extraction	Classifiers	hate	insult	neutral	offensive
CBOW	LR	92.8%	94.7%	63.4%	80.2%
	SVM	93.9%	95.3%	64.3%	82.3%
	RF	92.6%	94.0 %	64.7%	80.4%
	KNN	93.7%	95.1 %	61.9 %	82.1%

TABLE V
EXPERIMENT II PERFORMANCE RESULT(ACCURACY)

Feature extraction	Classifiers	hate	insult	neutral	offensive
Doc2Vec	LR	82.2%	82.0%	59.8%	59.7%
	SVM	93.9%	95.4 %	64.2%	82.3%
	RF	93.8%	95.3%	70.7 %	82.3%
	KNN	93.9%	95.4%	70.4%	82.3%

[4]https://fasttext.cc/

TABLE VI
EXPERIMENT III PERFORMANCE RESULT(ACCURACY)

Feature extraction	Classifiers	hate	insult	neutral	offensive
FastText	LR	93.9 %	95.4 %	70.4 %	82.3%
	SVM	93.9%	95.4%	71.5%	82.3%
	RF	92.5%	94.0 %	68.0%	81.0%
	KNN	93.9%	95.4%	70.2%	82.3%

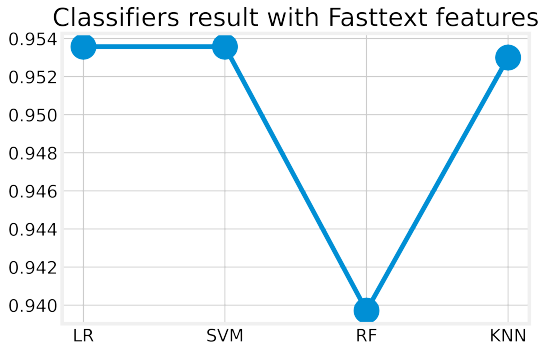

Fig. 6. Accuracy classifiers performance

SVM outperformed the other ML classifiers for hate text classification. We also denote Fasttext has surpassed Doc2vec and CBOW features in the detection of text containing hate, insult and offensive. To the best of our knowledge, no previous study has been conducted in the context of hate speech detection for Chadians using mixed language text in social media. The proposed classification approach and features extraction strategy was compared with the recent paper in Table VII

The best parameters for the different classifiers that achieved good performance is listed in Table VIII.

TABLE VII
PERFORMANCE COMPARISON

Languages	Dataset	Size	Methods	F1 score(%)
English, French, Arabic	Ousidhoum et al.2019 [24]	13000	STSL	89.0
Code-mixed Hindi-English	Sreelakshmi et al.2020 [30]	10000	Fasttext +SVM RBF	85.81
Chadian French-Arabic	Proposed approach	14000	Fasttext + SVM	95.4

TABLE VIII
OPTIMAL PARAMETER TUNE

Classifiers	Parameters
SVM	C: 1, Kernel: linear
LR	C: 1000, penalty: L2, solver: lbfgs
RF	algorithm : auto, leaf_size: 40, metric: minkowski, n_neighbors:30, p:2, weights: uniform
KNN	max_depth:80, max_features:3, max_leaf_nodes:500, n_estimators:1000

V. CONCLUSIONS

With the growth of social media users comes a rise in hateful content. This article proposes a mechanism for automatically detecting hate, offensive, and insult content. We acquired data from the Facebook platform due to a shortage of corpus for Chadian texts and the desire to add mixed language text for hate speech detection. Moreover, this study compared three feature engineering techniques and four ML algorithms to classify Facebook comments. The combination of Fasttext features with an SVM classifier for predicting the class labels produced the best results. The experiments showed the model's ability to detect Insult and Hate content in text. The f1-score of Insult and Hate class label showed the high score respectively 95.4% and 93.9%. We believe our results have made an essential step towards this task. We will investigate a deep annotation for our dataset for text considered political offensive, insult, sexist, racist, religious, and regional as part of future hate speech detection in mixed languages. We planned to build a robust feature based on a pre-trained model and deep learning approach.

REFERENCES

[1] Sindhu Abro, Sarang Shaikh, Zafar Ali, Sajid Khan, Ghulam Mujtaba, and Zahid Hussain Khand. Automatic hate speech detection using machine learning: A comparative study. *International Journal of Advanced Computer Science and Applications*, 11(8):484–491, 2020.

[2] Shivang Agarwal and C. Ravindranath Chowdary. Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. *Expert Systems with Applications*, 185:115632, dec 2021.

[3] Amer Al-Badarneh, Emad Al-Shawakfa, Basel Bani-Ismail, Khaleel Al-Rababah, and Safwan Shatnawi. The impact of indexing approaches on Arabic text classification. *Journal of Information Science*, 43(2):159–173, 2017.

[4] Nuha Albadi, Maram Kurdi, and Shivakant Mishra. Are they our brothers? analysis and detection of religious hate speech in the Arabic Twittersphere. *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, pages 69–76, 2018.

[5] Haifa K. Aldayel and Aqil M. Azmi. Arabic tweets sentiment analysis - A hybrid scheme. *Journal of Information Science*, 42(6):782–797, dec 2016.

[6] Ibrahim Aljarah, Maria Habib, Neveen Hijazi, Hossam Faris, Raneem Qaddoura, Bassam Hammo, Mohammad Abushariah, and Mohammad Alfawareh. Intelligent detection of hate speech in Arabic social network: A machine learning approach. *Journal of Information Science*, 47(4):483–501, 2021.

[7] Safa Alsafari, Samira Sadaoui, and Malek Mouhoub. Hate and offensive speech detection on Arabic social media. *Online Social Networks and Media*, 19(June):100096, 2020.

[8] Adel Assiri, Ahmed Emam, and Hmood Al-Dossari. Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis. *Journal of Information Science*, 44(2):184–202, 2018.

[9] Hay Mar Su Aung and Win Pa Pa. Analysis of Word Vector Representation Techniques with Machine-Learning Classifiers for Sentiment Analysis of Public Facebook Page's Comments in Myanmar Text. *2020 IEEE Conference on Computer Applications, ICCA 2020*, 2020.

[10] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. *26th International World Wide Web Conference 2017, WWW 2017 Companion*, abs/1706.00188(2):759–760, 2017.

[11] Delphine Battistelli, Cyril Bruneau, and Valentina Dragos. Building a formal model for hate detection in French corpora. *Procedia Computer Science*, 176:2358–2365, 2020.

[12] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
[13] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, pages 512–515, 2017.
[14] Fabio Del Vigna, Andrea Cimino, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. Hate me, hate me not: Hate speech detection on Facebook. *CEUR Workshop Proceedings*, 1816(May):86–95, 2017.
[15] Awet Fesseha, Shengwu Xiong, Eshete Derb Emiru, Moussa Diallo, and Abdelghani Dahou. Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. *Information (Switzerland)*, 12(2):1–17, feb 2021.
[16] Paula Fortuna, Juan Soler-Company, and Leo Wanner. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing & Management*, 58(3):102524, may 2021.
[17] T Jünger, J., & Keyling. GitHub - strohne/Facepager: Facepager was made for fetching public available data from YouTube, Twitter and other websites on the basis of APIs and webscraping., 2019.
[18] György Kovács, Pedro Alonso, and Rajkumar Saini. Challenges of Hate Speech Detection in Social Media. *SN Computer Science*, 2(2), apr 2021.
[19] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019.
[20] Son T. Luu, Kiet Van Nguyen, and Ngan Luu Thuy Nguyen. A Large-Scale Dataset for Hate Speech Detection on Vietnamese Social Media Texts. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12798 LNAI:415–426, 2021.
[21] Ricardo Martins, Marco Gomes, Jose Joao Almeida, Paulo Novais, and Pedro Henriques. Hate speech classification in social media using emotional analysis. *Proceedings - 2018 Brazilian Conference on Intelligent Systems, BRACIS 2018*, 1(April 2019):61–66, 2018.
[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pages 1–12, 2013.
[23] Donald George Morrison, Robert Cameron Mitchell, and John Naber Paden. Chad. In *Black Africa*, pages 411–418. Palgrave Macmillan UK, London, 1989.
[24] Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit Yan Yeung. Multilingual and multi-aspect hate speech analysis. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 4675–4684, 2020.
[25] Ahmed Oussous, Fatima Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. ASA: A framework for Arabic sentiment analysis. *Journal of Information Science*, 46(4):544–559, 2020.
[26] Juan Carlos Pereira-Kohatsu, Lara Quijano-Sánchez, Federico Liberatore, and Miguel Camacho-Collados. Detecting and monitoring hate speech in twitter. *Sensors (Switzerland)*, 19(21):1–37, 2019.
[27] Subbaraju Pericherla and E Ilavarasan. Performance analysis of Word Embeddings for Cyberbullying Detection. *IOP Conference Series: Materials Science and Engineering*, 1085(1):012008, feb 2021.
[28] Ekaterina Pronoza, Polina Panicheva, Olessia Koltsova, and Paolo Rosso. Detecting ethnicity-targeted hate speech in Russian social media texts. *Information Processing and Management*, 58(6):102674, nov 2021.
[29] Amandeep Singh, Malka N. Halgamuge, and Beulah Moses. *An Analysis of Demographic and Behavior Trends Using Social Media: Facebook, Twitter, and Instagram*. Elsevier Inc., 2019.
[30] K. Sreelakshmi, B. Premjith, and K. P. Soman. Detection of Hate Speech Text in Hindi-English Code-mixed Data. *Procedia Computer Science*, 171(2019):737–744, 2020.
[31] Francielle Alves Vargas, Fabiana Rodrigues de Góes, Isabelle Carvalho, Fabrício Benevenuto, and Thiago Alexandre Salgueiro Pardo. Contextual Lexicon-Based Approach for Hate Speech and Offensive Language Detection. *ArXiv*, abs/2104.12265, apr 2021.
[32] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June 2016. Association for Computational Linguistics.