Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 09, No. 01, March 2023.

A DEEP DECISION FORESTS MODEL FOR HATE SPEECH DETECTION

Kennedy Malanga Ndenga

(Received: 2-Nov.-2022, Revised: 4-Feb.-2023, Accepted: 14-Feb.-2023)

ABSTRACT

Detecting and controlling propagation of hate speech over social-media platforms is a challenge. This problem is exacerbated by extremely fast flow, readily available audience and relative permanence of information on social media. The objective of this research is to propose a model that could be used to detect political hate speech that is propagated through social-media platforms in Kenya. Using Twitter textual data and Keras TensorFlow Decision Forests (TF-DF), three models were developed; i.e., Gradient Boosted Trees with Universal Sentence Embedding (USE), Gradient Boosted Trees and Random Forest, respectively. The Gradient Boosted Trees with USE model exhibited a superior performance with an accuracy of 98.86%, a recall of 0.9587, a precision of 0.9831 and an AUC of 0.9984. Therefore, this model can be utilized for detecting hate speech on social media platforms.

KEYWORDS

Hate speech detection, TensorFlow decision forests, Gradient boosted trees, Universal sentence embedding, National cohesion and integration commission.

1. INTRODUCTION

Social media has permeated our lives such that a majority of unstructured communications happen through these platforms. Social media provides a ready audience anytime. As a result, a large fraction of discourses that were initially verbally articulated have shifted to social-media platforms. Control of communication over social media is difficult to achieve or enforce, considering that the operators of social-media platforms are foreign companies operating from different legislative environments compared to those of their users. This problem is exacerbated by the extremely fast flow and dynamism of information on social media. Therefore, sensitive and potentially harmful content can propagate through social-media platforms quickly without being detected. Hate speech is an example of such communication that can be difficult to deal with particularly when it is propagated through social-media platforms. Actually, Mr. António Guterres –the United Nations (UN) Secretary-General– describes Social media as a global megaphone for hate [1].

According to the UN's Strategy and Plan of Action on Hate Speech, hate speech is any kind of communication in speech, writing or behaviour that attacks or uses a pejorative or discriminatory language with reference to a person or a group on the basis of who they are. In other words, this happens based on their religion, ethnicity, nationality, race, colour, descent, gender or any other identity factor [1]. Hate speech has been identified as a trigger of violence and suffering in several parts of the world, including the Tigray region of Ethiopia, Guinea, Sri Lanka, …etc. [2]. In Kenya, political hate speech was blamed for the 2007/2008 post-election violence [3]. During the COVID-19 pandemic, social-media platforms contributed immensely to Sino-phobic hate sentiments, where an Asian community was blamed for the pandemic [34]. Khan et al. add that social-media platforms accelerated propaganda related to the Shaheen Bagh protests in New Delhi against the National Register of Citizens, Citizenship Amendment Act and National Population Register [34].

Propagation of hate speech over social-media platform is relatively a new phenomenon, considering that social-media platforms are recent disruptive technologies. Hate speech can be propagated on social media through: text messages, pictures, videos, emoji or emoticons. Sometimes, hate speech could be obfuscated in online content that seems ordinary. Various approaches, including intentional misspelling by swapping characters, elongating words using many repeated letters or putting spaces between letters, have been used to obfuscate hate speech in social-media discourses [35]. Hiding of hate-text messages in images and hate sarcasm using images or videos are other techniques of hiding hate speech over social-media platforms [6].

K. Malanga Ndenga is with the Department of Pure & Applied Sciences, Kirinyaga Univ., Kutus, Kenya. Email: kmalanga@kyu.ac.ke

Poletto et al. [4] argue that explicitly defining hate speech is challenging because of widespread vagueness in the use of related terms, such as abusive, toxic, dangerous, offensive or aggressive language, that often overlap and are prone to strongly subjective interpretations. However, they go ahead and conclusively define hate speech as a content defined by its action; i.e., generally spreading hatred or inciting violence or threatening people's freedom, dignity and safety by any means and by its target – which must be a protected group or an individual targeted for belonging to such a group and not for his/her individual characteristics. Figure 1 depicts this definition.

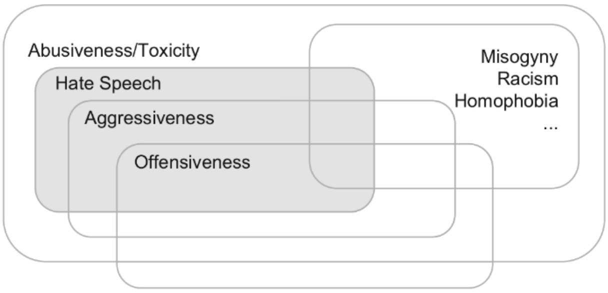

Figure 1. Relationships between hate speech and related concepts [4].

It is challenging to deal with hate speech that is propagated over social-media platforms as compared to that which is propagated over traditional media, like newspapers, magazines, TV, radio or billboards. Unlike in traditional media, online hate speech can be produced and distributed easily, at low cost and anonymously while having the potential to reach a global and diverse audience in real time [1]. The relative permanence of online content is also problematic when hateful discourse can resurface and regain popularity over time [1]. The UN concludes that efforts of understanding and monitoring the dynamics of hate speech across diverse online communities and platforms often stall given the sheer scale and diversity of the phenomenon, current technological limitations of automated monitoring systems and the opacity of online companies [5]. Therefore, online hate speech detection and control represent a phenomenon that deserves research attention.

The purpose of this research is to propose a model that could be used to detect political hate speech that is propagated through social-media platforms in Kenya. Such a model could be used as a tool of decision support for monitoring future social-media discourses. The study described in this work treats only the textual form of hate speech. The remaining parts of the article are organized as follows: Section two is a discussion of previous work that is related to this study, where various natural-language processing (NLP) techniques for hate speech detection are assessed; Section three describes the approach proposed by this research; Section four outlines the method of research experiments; Section five presents results from the experiments; Section six is a discussion of the results, while section seven concludes the study. This study has made three main contributions to knowledge. The first contribution is methodological, where a new method of hate speech detection has been described by the study. Proposing a new model for hate speech detection is the second contribution. Lastly, the study has introduced the Kenyan culture in the area of hate speech detection.

2. RELATED WORK

Hate speech detection has attracted the attention of researchers in the recent few years. Gomez et al. [6] studied hate speech detection in multi-modal publications formed by text and image data from Twitter. They found out that multi-modal models cannot outperform text models in detecting hate speech. Based on this finding, this study developed a hate-speech detection model using the text component of tweets from Kenyan political discourses.

Mullah et al. [7] reviewed machine-learning algorithms and techniques for hate speech detection in social media. They found out that the majority of existing research work on hate speech detection used classical machine-learning techniques as compared to ensemble and deep-learning techniques. They also identified some open challenges in hate speech detection which include: cultural variations, pandemic or natural disasters, data sparsity, imbalanced datasets and dataset availability concerns. They emphasized the need to take the campaign of hate speech prevention to other non-western parts of the world, since culture and tradition play a significant role in hate speech detection efforts [7]. Mullah et al.'s [7] emphasis is agreeable, since contextual meanings of words in discourses vary from one cultural region to another. Thus, models trained on text datasets in one culture may have some degree of bias if used to test data from a different culture. Therefore, by experimenting with data from the Kenyan culture, models in this field are enriched.

Ombui et al. [3] studied the identification of hate speech in code-switched text messages by exploring the performance of different features across various machine-learning algorithms. They established that character level Term Frequency-Inverse Document Frequency performed best given a code-switched dataset of 25k annotated tweets using support vector machine algorithm as compared to six other conventional and two deep-learning algorithms.

Khan et al. [34] presented HcovBi-Caps –a Convolutional, BiGRU and Capsule network-based deep-learning model– to classify hate speech and evaluated it over two Twitter-based benchmark datasets; i.e., DS1 which was balanced and DS2 which was unbalanced. They found out that HCovBi-Caps showed comparatively better performance over the unbalanced dataset with precision, recall and f-score values of 0.90, 0.80 and 0.84, respectively. In another study, Khan et al. [36] introduced BiCHAT, a BiLSTM deep-learning model for hate speech detection. The model was trained and evaluated over three benchmark datasets; i.e., HD1, HD2 and HD3 extracted from Twitter. They found out that their BiCHAT model outperformed three state-of-the-art models used in studies with an improvement of 8%, 7% and 8% in terms of precision, recall and f-score, respectively. They acknowledge that the performance of their models in the two studies were based on evaluations done on non-diverse datasets in a non-multilingual set-up. Like Gomez et al. [6], Koutlis et al. [37] presented MemeTector, a multi-modal model for classifying images as memes or regular images. They proposed that their model could be utilized in online social environments to detect hate speech and disinformation. Aggarwal et al. [38] too proposed an approach to solve the problem of identifying hate memes. However, the two studies did not compare how their multi-modal models performed relative to text-only models in detecting hate speech [37]-[38].

Poletto et al. [4] systematically analyzed resources made available for hate speech detection models from the perspective of: their development methodology, topical focus, language coverage, among other factors. Their survey found out that datasets are available in several languages for hate speech detection, but focus on different topics. Like Mullah et al. [7], Poletto et al. [4] added that there is a challenge of developing hate speech detection architectures which are stable and well performing across different languages and abusive domains. They also noted that biases in the design and annotation of the training dataset, as well as topic biases in the training datasets as compared to the test and volatile nature of topics are factors to keenly consider when developing resources for hate speech detection.

Badjatiya et al. [8] investigated the application of deep neural network architectures for the task of hate speech detection and found out that embeddings learned from deep neural-network models when combined with gradient-boosted decision trees led to the best accuracy values. Dorris et al. [9] introduced the HateDefender –a hate speech and offensive language defence system– which consists of a detection model based on deep Long Short-term Memory (LSTM) neural networks that according to them can effectively detect hate speech with an average accuracy of 90.82%. Like Badjatiya et al. [8] and Dorris et al. [9], deep-learning techniques have also been used to develop models for detecting hate speech in this research. However, the approach in this study differs from those of the discussed studies, since it utilizes different embeddings used as discussed in Section 3.

Other works use the more recent and well-known type of DNN; namely, transformers, in particular the BERT model proposed by Google [22]. These techniques of language modeling are presented as a general model used for a variety of NLP tasks; for example, translation, question and response tasks. However, when used in transfer learning based on a pre-trained model, the model is fine-tuned using a special dataset more suitable for the task. For instance, Mozafari et al. [23] proposed a novel transfer learning approach based on BERT – an existing pre-trained language model – in order to overcome the problem of lack of sufficient amount of labelled hate speech data. Velankar et al. [24] presented baseline classification results using deep-learning models based on CNN, LSTM and Transformers. Using a multi-lingual BERT version fine-tuned with an annotated Marathi language dataset, Velankar et al. [24] showed that transformers outperformed other methods. Most recent works tackle hate speech detection using neural networks to train models of detection. This means that they usually need huge datasets so that their models can converge therefore making them not suitable with a relatively small dataset. It is believed that models based on decision forests as applied in this study could overcome this limitation.

2.1 Techniques for Hate Speech Detection

Hate speech is a classification problem. Classical, ensemble and deep-learning classification techniques have previously been used to achieve hate speech detection with varying degrees of success [7]. Classical machine-learning algorithms require more structured data and are more dependent on human intervention to learn, whereby human experts determine the hierarchy of features through data labelling for the machine to understand differences between data inputs [39]. Examples of such algorithms include support vector machines (SVMs), Naive Bayes (NB), Logistic Regress (LR), Decision Trees (DTs) and K-Nearest Neighbour (KNN) [7]. Unlike classical machine learning, deep-learning algorithms automate much of the feature-extraction process, thus eliminating some of the manual human intervention required [39]. Deep-learning algorithms differ from machine-learning algorithms, since they require large datasets to learn reasonably, while machine learning requires less to learn [7]. On the other hand, ensemble learning schemes combine multiple base machine learning algorithms of any type – e.g. decision tree, neural network, linear regression model, …etc. – to make a decision, typically in supervised machine-learning tasks [40]. Classical machine-learning techniques are unable to effectively analyze some text datasets that are very large and not linearly separable; however, deep-learning algorithms have capabilities of effectively analyzing such datasets [7]. Since this study is dealing with unstructured text dataset, it was found prudent to apply deep-learning techniques to analyze the data.

Variants of deep learning that have been used for hate speech detection are mostly Deep Neural Network techniques, which include Convolution Neural Network, Recurrent Neural Networks, i.e., Long Short Term Memory (LSTM) and Gated Recurrent Units (GRUs) [7]. Deep neural networks have multiple hidden layers [41]. Long Short Term Memory (LSTM) networks [10] and their variants have been used by Gomez et al. [6], Dorris et al. [9], Ong [11] and Badjatiya et al. [8] for the classification of hate speech texts from social media. Long Short Term Memory networks are a special kind of Recurrent Neural Networks (RNNs) capable of learning long-term dependencies in a sequence (sentence) [12]. An RNN is a network with loops allowing information to persist and can be thought of as multiple copies of the same network, each passing a message to a successor [12]. Gomez et al. [6] used LSTM, because they believed that these algorithms provide a strong representation of tweet text data. Their LSTM models gave an f-score of 0.703, an area under the ROC of 0.732 and a mean accuracy of 68.4 for tweet text input. Ong [11] found out that Bidirectional LSTM (BiLSTM) convolutional neural network model was optimal with the highest f1-score.

Badjatiya et al. [8] performed experiments on a benchmark dataset of 16K annotated tweets and showed that three deep learning architectures i.e., FastText, Convolutional Neural Networks (CNNs) and Long Short Term Memory Networks (LSTMs) deep-learning methods, outperformed char/word n-gram methods by approximately 18 f1 points. On the other hand, Ombui et al. [3] explored both conventional and deep-learning classifiers and found out that Support Vector Machine algorithm yielded the best classification accuracy when classifying code-switched text messages. Unlike the previously discussed techniques, Keras and TensorFlow Decision Forests (TF-DF) were used to develop three models for this study; i.e., the Gradient Boosted Trees with Universal Sentence Embedding model, the Gradient Boosted Trees model and the Random Forest model.

3. PROPOSED APPROACH

For model development, Keras TensorFlow Decision Forests (TF-DF) were used to train neural network models for classifying tweeter text data. Keras is a deep-learning framework that makes it easier to run new experiments [13]. It ships deep learning-powered features in real products [14]. Keras is a high-level neural network library that runs on top of TensorFlow [41]. TensorFlow Decision Forests are a collection of state-of-the-art algorithms for the training, serving and interpretation of Decision Forest models which support classification, regression and ranking [15]. A decision forest is an ensemble of randomly trained decision trees whose prediction is the aggregation of predictions of its decision trees [16], [17]. A decision tree is a hierarchical structure of connected nodes, such that during training, all training data {v} is sent into the tree followed by optimizing parameters of split nodes, so as to optimize a chosen energy function, as shown in Figure 2.

During testing, a split (internal) node applies a test to the input data v and sends it to the appropriate child iterating the process until a leaf (terminal) node is reached (beige path), as depicted in Figure 3 [16].

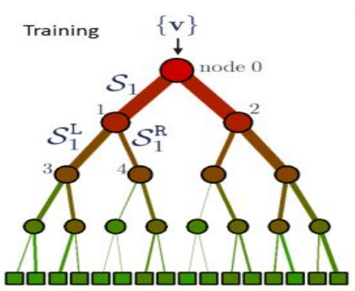

Figure 2. Training of a decision tree [16].

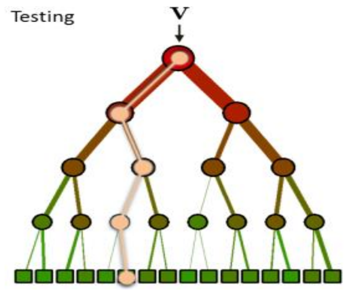

Figure 3. Testing of input data v in a decision tree [16].

Criminisi et al. [16] argue that decision forests compare favourably with respect to other techniques and have led to one of the biggest success stories of computer vision in recent years. They attribute the recent revival of decision forests to the discovery that ensembles of slightly different trees tend to produce much higher accuracy on previously unseen data, a phenomenon known as generalization. According to Rokach [18], decision trees have a high predictive performance for a relatively small computational effort, are able to handle a variety of input data including textual data and scale well to big data as compared to other methods. For experiments in this work, the Gradient-boosted trees and Random Forest algorithms were used.

Boosting is a method for converting a weak learning algorithm into one that achieves arbitrarily high accuracy by sequentially applying weak learners to repeatedly re-weighted versions of the training data [19]. In gradient boosting, the learning procedure consecutively fits new models to provide a more accurate estimate of the response variable with an objective of constructing new base-learners that maximally correlate with the negative gradient of the loss function associated with the whole ensemble [33]. Random forests are a combination of tree predictors, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [20]. Krauss et al. [19] explain that boosting works by sequentially applying weak learners to repeatedly reweighted versions of the training data whereby after each boosting iteration, misclassified examples have their weights increased and correctly classified examples have their weights decreased. They conclude that each successive classifier focuses on examples that have been hard to classify in the previous steps [19]. Mathematically, an ensemble of trees can be written as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{1}$$

where K is the number of trees, f_k is a function in the functional space F and F is the set of all possible classification and regression trees (CARTs) [21]. The objective function to be optimized is given by:

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k) \tag{2}$$

where $\omega(f_k)$ is the complexity of the tree f_k [21].

GloVe embedding –an unsupervised learning algorithm for obtaining vector representations for words– has been extensively used in hate speech detection (e.g. by Gomez et al. [6], Dorris et al. [9], Ong [11] and Badjatiya et al. [8]). Pre-trained embeddings are useful when available training data is limited for data-hungry deep-learning methods [25]. The Universal Sentence Encoder (USE) embedding [26] was used to encode text into high-dimensional vectors for text classification. USE is a model for producing sentence embeddings that demonstrate superior transfer to a number of other NLP tasks as compared to pre-trained word embeddings, such as those produced by Word2vec or GloVe [25]. Word embeddings give a way to use an efficient, dense representation of vectors in which similar words have similar encodings [26]. The USE is trained and optimized for greater-than-word length text, such as sentences, phrases or short paragraphs the input of which is a variable-length English text and the output is a 512 dimensional vector [26]. This encoder differs from word-level embedding models, since it is trained on a number of natural-language prediction tasks that require modeling the meaning of word sequences rather than just individual words [26]. The USE was partially trained with custom text-classification tasks in mind, as shown by Figure 4.

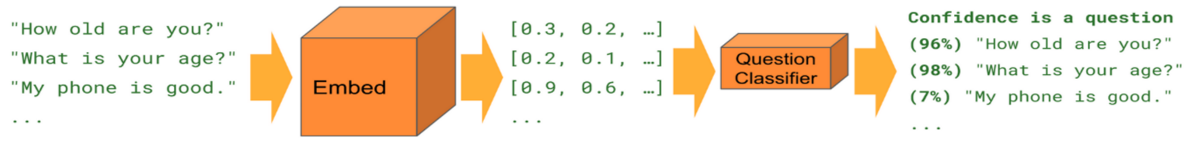

Figure 4. Classification task of the Universal Sentence Encoder (USE) [26].

4. Experimental Method

Data whose scope was of political conversations in Kenya was collected from Twitter using Twitter API V2 [27]. For training the models, data was collected from 1st January 2017 to 31st May 2022 whereas for testing on real data, data was collected from 1st August 2022 to 10th August 2022. This period was significant, since it corresponded to the week of the 2022 general elections in Kenya which were held on 9th August 2022. It was believed that during this period, political hate speech could hit a peak. To ensure that relevant data was collected from Twitter, a query was developed whose search criteria included words that had been identified by the National Cohesion and Integration Commission (NCIC) as words that had been used to propagate political hate against specific groups of people in Kenya [28]. This ensured that tweets that were mined were potentially of political hate speech. While tweet data with many attributes including text, images and videos were downloaded, focus was mainly on the text attribute since most hate speech messages on social media are constructed through texts [7] and that text models tend to outperform multi-modal models in detecting hate speech [6]. A total of 46957 records were used for training the model, while 17873 records were used to test it.

Although the context of tweets that were mined focused on hate speech words as defined by the NCIC, it is good to note that not all tweets with these words are necessarily of hate nature. To annotate hate tweets from non-hate tweets, sentiment analysis was performed on the text for each tweet to determine its speech score using the NLTK's Valence Aware Dictionary and Sentiment Reasoner (VADER) [29]. The current version of VADER has the capability of properly handling sentences with punctuation, word-shape, sentiment-laden slang words as modifiers, sentiment-laden emoticons, sentiment-laden initialism and acronyms and utf-8 encoded emoji [30]. Therefore, pre-processing the text data through tokenization or lemmatization was not done before performing sentiment analysis on it with VADER. For each statement, VADER gives probability scores for positive, neutral and negative sentiments all of which must sum to 1. It also gives a compound (average) score for all the scores. A compound score was used to determine sentiments for each tweet text. VADER recommends positive sentiments for compound scores greater than 0.05, neutral for compound scores greater than -0.05, but less than 0.05 and negative sentiments for compound scores of less than -0.05. For the case of this study, all neutral and positive sentiments were considered as not constituting hate speech, while all negative sentiments were considered as potentially constituting hate speech. Thus all tweets with a sentiment compound score of greater than -0.05 were labelled as 0; i.e., "Not_Hate", while those less than or equal to -0.05 were labelled as 1; i.e., "Hate". Figure 5 shows a schema of the method applied in this study.

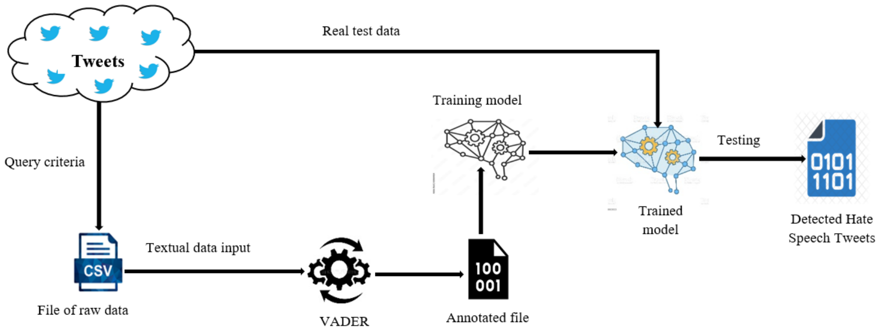

Figure 5. Schema for the method applied in this study.

USE embeddings [26] and Keras TensorFlow Decision Forests (TF-DF) [15] were used to encode text into high-dimensional vectors and train models for text classification, respectively. Default Keras hyper-parameters were applied. Three neural-network models were developed using Keras. In the first model, raw text was first encoded via pre-trained embeddings and then passed to a gradient boosted tree model for classification. In the second and third models, raw text was directly passed to the gradient boosted trees model and RandomForestModel, respectively. The models were compiled with Google Colab [31] TPU by passing Accuracy, Recall, Precision and AUC metrics. TF-DF automatically detects the best loss for the task; i.e., either classification or regression.

5. RESULTS

Table 1 shows a summary of the results from the experiments.

Table 1. Performance of the three models measured against Accuracy, Recall, Precision and AUC.

Model	Algorithm	Accuracy	Recall	Precision	AUC
Model_1	GradientBoostedTrees with USE embeddings	0.9886	0.9587	0.9831	0.9984
Model_2	GradientBoostedTrees	0.9272	0.6302	1.0000	0.9240
Model_3	RandomForestModel	0.9272	0.6302	1.0000	0.8085

The following graphs show accuracy and loss against number of trees for the models.

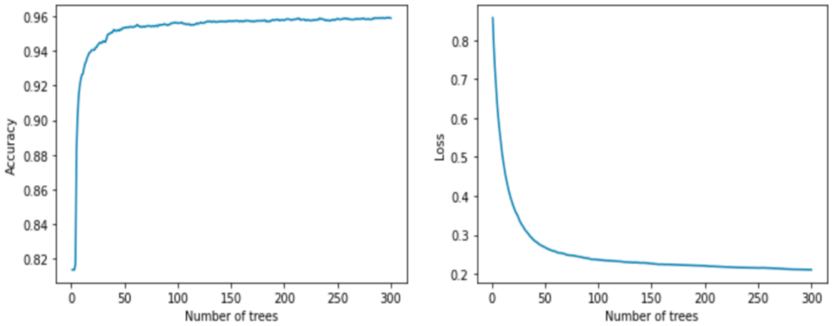

Figure 6. Graph showing accuracy (left) against number of trees and loss against number for model 1.

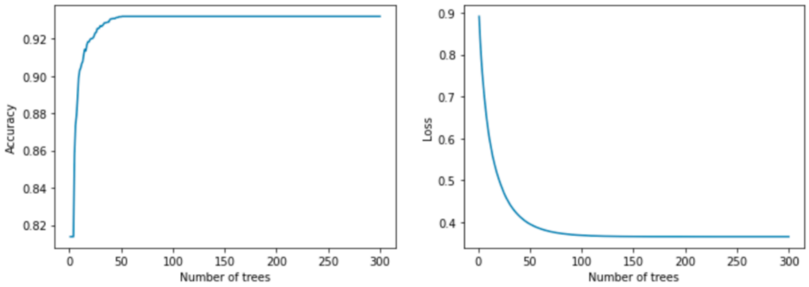

Figure 7. Graph showing accuracy against number of trees (left) and loss against number for model 2.

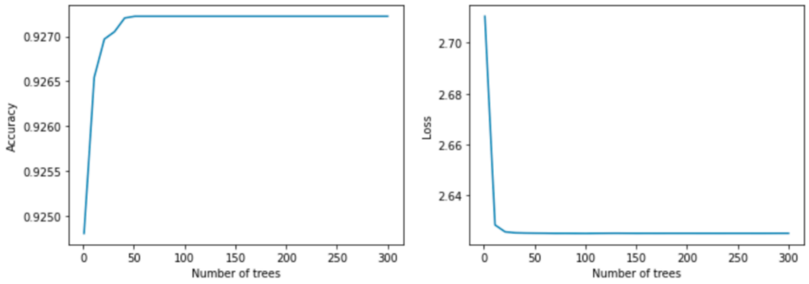

Figure 8. Graph showing accuracy against number of trees (left) and loss against number for model 3.

6. DISCUSSION

From Table 1, it can be observed that model_1; i.e., the GradientBoostedTrees with USE embeddings model has generally performed well according to most metrics. The training logs of accuracy against number of trees and the loss against number of trees also indicate that this model has a superior performance. Training logs show the quality of the model according to the number of trees in the model [32]. When compared with previous studies on detection of hate speech on social-media platforms, the approach of GradientBoostedTrees with USE embeddings still exhibits a superior performance as demonstrated by Table 2. While experiments for these studies were done in different contexts with different datasets, comparing them may give us an idea on the performance of individual models. An accurate comparison would require replication of the experiments while keeping the dataset constant, but varying the modelling algorithms.

Table 2. Performance comparison of hate speech detection models.

Author	Best performing technique	Metrics for measuring model performance

Gomez et al. [6]	LSTM + Glove embeddings	F-Score = 0.703, AUC = 0.732
		Accuracy = 68.4%

Ong (2019) [11]	BiLSTM-CNN	F1-Score = 0.75

Badjatiya et al. [8]	LSTM, LSTM + Random Embedding + Gradient, Boosted Decision Trees	Precision = 0.930
		Recall = 0.930
		F1 = 0.930

Dorris et al. [9]	LSTM	Accuracy 90.82%

This research	GradientBoostedTrees with USE Embeddings	Accuracy = 98.86%
		Recall = 0.9587
		Precision = 0.9831
		AUC = 0.9984

The graphs in Figures 6–8 also show that the GradientBoostedTrees with USE embeddings model suffers less loss in comparison to the GradientBoostedTrees model and the RandomForestModel. The purpose of this research was to propose a model for detecting political hate speech propagated through social-media platforms. Considering the performance of the GradientBoostedTrees with USE embeddings model, this model is proposed for detecting hate speech on social-media platforms.

However, we should remember that this model works only on textual data from social-media discourses. We are aware of other methods through which hate speech is transmitted over social-media platforms. As discussed earlier, these methods include obfuscation of hate messages by distorting spelling of words, hiding hate messages in images and using sarcasm through text or audio-visual methods. There is a need to continue searching for models that can help the society detect hate speech propagated through these methods. Although Gomez et al. [6] found out that text models outperform multi-modal models, we should not stop research on multi-modal methods. If we do so, multi-modal hate speech will become a key conduit of hate speech propagation on social media. Finally, there is a need for the research community to invest in replication of already proposed models, so as to test their viability and applicability.

6. CONCLUSION

Although propagation of hate-speech over social-media platform is relatively a recent phenomenon, several researchers have proposed various approaches for tackling this problem with varying success. The purpose of this research was to propose a model for detecting political hate speech in Kenya propagated through social-media platforms. Experimenting with data from Twitter, it was found out that the GradientBoostedTrees with USE embeddings model exhibited a superior performance as compared to other models previously proposed in literature. However, actualizing this model at production level may pose some challenge, since it may be difficult to apply it in real-time detection of hate speech, considering that it requires massive computational resources for training, compiling and testing the model. For this research, Colab's TPU was used to train and test the model. Currently, such a resource may not be available around the clock for industrial deployment of such a model. Secondly, the model depends heavily on hate-speech words as defined by the NCIC. These words have their original meanings which are not necessarily of hate-speech nature. Similarly, as language evolves, the meaning of the NCIC's hate speech words is bound to evolve too. Therefore, it is not guaranteed that those words will always imply hate speech. This challenge can easily generate false positive results.

FUTURE WORK

A replication of experiments in this research with an objective of achieving results that can be generalized is necessary.

ACKNOWLEDGEMENTS

The author would like to thank Twitter, Inc. for allowing the study to access data from their databases for research purpose, Google LLC for availing their TPU computing resources for this research and VADER developers for making their tool open source.

REFERENCES

[1] U. Nations, "Understanding Hate Speech," [Online], Available: https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech, 20 Oct. 2022.

[2] U. Nations, "Hate Speech," [Online], Available: https://www.un.org/en/hate-speech/impact-and-prevention/why-tackle-hate-speech, 20 Oct. 2022.

[3] E. Ombui, L. Muchemi and P. Wagacha, "Hate Speech Detection in Code-switched Text Messages," Proc. of the 3rd IEEE Int. Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1–6, Ankara, Turkey, 2019.

[4] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco and V. Patti, "Resources and Benchmark Corpora for Hate Speech Detection: A Systematic Review," Language Resources and Evaluation, vol. 55, no. 2, pp. 477–523, 2021.

[5] U. Nations, "Hate Speech," [Online], Available: https://www.un.org/en/hate-speech/impact-and-prevention/challenges-of-tracking-hate, 20 Oct. 2022.

[6] R. Gomez, J. Gibert, L. Gomez and D. Karatzas, "Exploring Hate Speech Detection in Multimodal Publications," Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV), pp. 1459-1467, 2020.

[7] N. S. Mullah and W. M. N. W. Zainon, "Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review," IEEE Access, vol. 9, DOI: 10.1109/ACCESS.2021.3089515, 2021.

[8] P. Badjatiya, S. Gupta, M. Gupta and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," Proc. of the 26th Int. Conf. on World Wide Web Companion, pp. 759–760, DOI:10.1145/3041021.3054223, 2017.

[9] W. Dorris, R. Hu, N. Vishwamitra, F. Luo and M. Costello, "Towards Automatic Detection and Explanation of Hate Speech and Offensive Language," Proc. of the 6th Int. Workshop on Security and Privacy Analytics, pp. 23–29, DOI: 10.1145/3375708.3380312, 2020.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] R. Ong, "Offensive Language Analysis Using Deep Learning Architecture," arXiv: 1903.05280, DOI: 10.48550/arXiv.1903.05280, 2019.

[12] Github, "Understanding LSTM Networks," [Online], Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 21 Oct. 2022.

[13] F. Chollet, "Keras," [Online], Available: https://keras.io/, 21 Oct. 2022.

[14] F. Chollet, "Introduction to Keras for Engineers," [Online], Available: https://keras.io/gettingstarted/ intro to keras for engineers/, 21 Oct. 2022.

[15] Google, "Tensorflow Decision Forests," [Online], Available: https://www.tensorflow.org/decision forests, 20 Oct. 2022.

[16] A. Criminisi, J. Shotton and E. Konukoglu, "Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-supervised Learning [Internet]," Microsoft Research, MSR-TR-2011-114, 2011.

[17] G. Developers, "Decision Forests," [Online], Available: https://developers.google.com/machine-learning/decision-forests/intro-to-decision-forests-real, 20 Oct. 2022.

[18] L. Rokach, "Decision Forest: Twenty Years of Research," Information Fusion, vol. 27, pp. 111–125, 2016.

[19] C. Krauss, X. A. Do and N. Huck, "Deep Neural Networks, Gradient-boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500," Europ. J. of Operat. Research, vol. 259, no. 2, pp. 689–702, 2017.

[20] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[21] Xgboost Developers, "Introduction to Boosted Trees," [Online], Available: https://xgboost.readthedocs.io/en/stable/tutorials/model.html, 20 Oct. 2022.

[22] Google, "Open Sourcing Bert: State-of-the-art Pre-training for Natural Language Processing," [Online], Available: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html, 20 Oct. 2022.

[23] M. Mozafari, R. Farahbakhsh and N. Crespi, "A BERT-based Transfer Learning Approach for Hate Speech Detection in Online Social Media," Proc. of the Int. Conf. on Complex Networks and Their Applications, Computational Intelligence Book Series, vol. 881, pp. 928–940, Springer, 2019.

[24] A. Velankar, H. Patil, A. Gore, S. Salunke and R. Joshi, "L3Cube-mahahate: A Tweet-based Marathi Hate Speech Detection Dataset and BERT Models," Proc. of the 3rd Workshop on Threat, Aggression and

Cyberbullying (TRAC 2022), pp. 1-9, Gyeongju, Republic of Korea, 2022.

[25] D. Cer, Y. Yang, S.-Y. Kong et al., "Universal Sentence Encoder," arXiv: 1803.11175, 2018.

[26] T. Hub, "Universal-sentence-encoder," [Online], Available: https://tfhub.dev/google/universal-sentence-encoder/4, 20 Oct. 2022.

[27] Twitter, "Tweet Downloader," [Online], Available: https://developer.twitter.com/apitools/downloader, 20 Oct. 2022.

[28] N. Cohesion and I. Commission, "Hatelex: A Lexicon of Hate Speech Terms in Kenya," Nairobi, Tech. Rep., 2022.

[29] C. Hutto and E. Gilbert, "Vader: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Proc. of the Int. AAAI Conf. on Web and Social Media, vol. 8, no. 1, pp. 216–225, 2014.

[30] GitHub, "Vadersentiment," [Online], Available: https://github.com/cjhutto/vaderSentiment, Oct. 2022.

[31] Google, "Welcome to Colaboratory," [Online], Available: https://colab.research.google.com, Oct. 2022.

[32] Google, "Build, Train and Evaluate Models with Tensorflow Decision Forests," [Online], Available: https://www.tensorflow.org/decision forests/tutorials/beginner colab, 20 Oct. 2022.

[33] A. Natekin and A. Knoll, "Gradient Boosting Machines: A Tutorial," Frontiers in Neurorobotics, vol. 7, p. 21, 2013.

[34] S. Khan, A. Kamal, M. Fazil et al., "HCovBi-Caps: Hate Speech Detection Using Convolutional and Bi-directional Gated Recurrent Unit with Capsule Network," IEEE Access, vol. 10, pp. 7881–7894, 2022.

[35] B. Vidgen, T. Thrush, Z. Waseem and D. Kiela, "Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection," Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. on Natural Language Processing, vol. 1: Long Papers, pp. 1667–1682, DOI: 10.18653/v1/2021.acl-long.132, 2021.

[36] S. Khan, M. Fazil, V. K. Sejwal, M. A. Alshara, R. M. Alotaibi and Kamal, "BICHAT: BiLSTM with Deep CNN and Hierarchical Attention for Hate Speech Detection," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 7, pp. 4335 – 4344, 2022.

[37] C. Koutlis, M. Schinas and S. Papadopoulos, "MemeTector: Enforcing Deep Focus for Meme Detection," arXiv: 2205.13268, DOI: 10.48550/arXiv.2205.13268, 2022.

[38] A. Aggarwal, V. Sharma, A. Trivedi et al., "Two-way Feature Extraction Using Sequential and Multimodal Approach for Hateful Meme Classification," Complexity, vol. 2021, pp. 1–7, 2021.

[39] IBM, "AI *vs.* Machine Learning *vs.* Deep Learning *vs.* Neural Networks: What’s the Difference?" [Online], Available: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks, 30 Jan. 2023.

[40] O. Sagi and L. Rokach, "Ensemble Learning: A Survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. e1249, 2018.

[41] V. Zocca, G. Spacagna, D. Slater and P. Roelants, Python Deep Learning, 2nd Edition, ISBN: 978-1789348460, Packt Publishing Ltd, 2017.

ملخص البحث:

يشكّل الكشْف عن خِطاب الكراهية والسّيطرة على انتشاره عبر منّصات التّواصل الاجتماعي تحدّياً كبيراً. وتتفاقم هذه المشكلة بفعل التّدفّق فائق السرعة، وتوافر الجمهور، وديمومة المعلومات على منصّات التّواصل الاجتماعي. ويهدف هذا البحث الى اقتراح نموذج يمكن استخدامه للكشْف عن خِطاب الكراهية السّياسي الذي ينتشر على منصّات التّواصل الاجتماعي في كينيا.

وقد تمّ تطوير ثلاثة نماذج لهذا الغرض باستخدام نُصوصٍ مأخوذة من تويتر. وبمقارنة نتائج النّماذج الثلاثة المطوّرة، اتّضح أنّ نموذج (GBT) مع التّضمين العامّ للجُمل (USE) قد حقّق أفضل النّتائج بدقّةٍ بلغت (99.86%)، واستعادةٍ بلغت (0.9587)، وضبطٍ مقداره (0.9831)، بينما كانت المساحة المحصورة تحت المنحنى لهذا النّموذج (0.9984). وعليه، فإنّ هذا النّموذج يمكن استخدامه للكشْف عن خِطاب الكراهية الذي ينتشر على منصّات التّواصل الاجتماعي.