IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 12, 2019, accepted January 5, 2020, date of publication January 20, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968173

Evaluating Machine Learning Techniques for Detecting Offensive and Hate Speech in South African Tweets

OLUWAFEMI ORIOLA AND EDUAN KOTZÉ
Department of Computer Science and Informatics, University of the Free State, Bloemfontein 9301, South Africa

Corresponding author: Oluwafemi Oriola (OriolaO@ufs.ac.za)

This work was supported by the University of the Free State, Postdoctoral Research under Grant 2019.

ABSTRACT In recent times, South Africa has been witnessing insurgence of offensive and hate speech along racial and ethnic dispositions on Twitter. Popular among the South African languages used is English. Although, machine learning has been successfully used to detect offensive and hate speech in several English contexts, the distinctiveness of South African tweets and the similarities among offensive, hate and free speeches require domain-specific English corpus and techniques to detect the offensive and hate speech. Thus, we developed an English corpus from South African tweets and evaluated different machine learning techniques to detect offensive and hate speech. Character n-gram, word n-gram, negative sentiment, syntactic-based features and their hybrid were extracted and analyzed using hyper-parameter optimization, ensemble and multi-tier meta-learning models of support vector machine, logistic regression, random forest, gradient boosting algorithms. The results showed that optimized support vector machine with character n-gram performed best in detection of hate speech with true positive rate of 0.894, while optimized gradient boosting with word n-gram performed best in detection of hate speech with true positive rate of 0.867. However, their performances in detection of other threatening classes were poor. Multi-tier meta-learning models achieved the most consistent and balanced classification performance with true positive rates of 0.858 and 0.887 for hate speech and offensive speech, respectively as well as true positive rate of 0.646 for free speech and overall accuracy of 0.671. The error analysis showed that multi-tier meta-learning model could reduce the misclassification error rate of the optimized models by 34.26%.

INDEX TERMS Machine learning, South Africa, Twitter, hate speech, offensive speech.

I. INTRODUCTION

Social networks are among the most impactful innovations in the 21st century. A popular social networking platform is Twitter, which allows subscribers to propagate information in the cyberspace using alphanumeric, special characters, hyperlinks, images, emoticons, and other icons. Over the years, it has experienced several changes in functionalities [1]. For example, the maximum size of character per tweet has recently been increased from 140 to 280, thereby encouraging more flexibility in interaction.

As a result of the rights to freedom of expressions in many climes, the propagation of offensive and hate speech via Twitter has risen regardless of the term of service prohibiting such speech. According to United Nations strategy and plan of action on hate speech,[1] hate speech has no international legal definition, but it is hinged on incitement, which is an explicit and deliberate act aimed at discrimination, hostility and violence. Similarly, offensive speech has been defined as the text, which uses abusive slurs or derogatory terms [2], which in many contexts have been confused with hate speech.

Machine learning has been used to classify and detect Twitter offensive and hate speech in contexts such as racial, sexist, misogyny, religious, refugee and immigrants. These have involved binary [3], multiclass classifications [4], [5] or both [6]. In the implementation of machine learning, supervised learning techniques such as classical single algorithms like Logistic Regression, Support Vector Machine [7] and Decision Tree algorithms [4] have been commonly used to predict the class of tweets. Other more complex tree algorithms such as Random Forest [4] and Gradient Boosting [8]

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

[1] https://www.un.org/en/genocideprevention/documents/

have also been used. In some instances, some of these algorithms have been optimized using parameter or hyper-parameter optimization [2], [6] or combined by majority voting [9], weighted voting [9] and meta-learning [10] to improve their individual performances.

Apart from the common use of machine learning, majority of the detection tasks have focused on English tweets because of the availability of English corpora and the widespread use of the language. However, the available English corpora do not cover every possible context of offensive and hate speech [11]. Therefore, offensive and hate speech detection have suffered setbacks in unpopular contexts like South Africa domain.

Twitter is the third most subscribed social networks in South Africa; its users account for twenty percent of active social media users in South Africa [12]. Recent media reports [13] showed that racially divisive comments were greatly propagated via social media before 2019 elections. It said hateful post increased by one hundred and seventy percent in the period. A recent cross-domain hate speech study [14] also supported the insurgence of offensive and hate tweets despite the legal solutions. Thus, there is need for additional solution to tackle the menace.

In order to be able to detect offensive and hate speech in South Africa without contradictions, we rely in this study on acceptable definitions of hate speech and related terms in South Africa. South Africa is ruled by a constitution [2], where the rights of its citizens, including the right to freedom of expressions, are enshrined. Therefore, the following excerpts from the constitution and other acts of constitution were used to distinguish among hate speech, offensive speech and free speech in South African context.

> "Freedom of expressions do not extend to propaganda for war; incitement of imminent violence; or advocacy of hatred that is based on race, ethnicity, gender or religion, and that constitutes incitement to cause harm."[2]
>
> "Any person who intentionally publishes, propagates or advocates anything or communicates to one or more persons in a manner that could reasonably be construed to demonstrate a clear intention to be harmful or to incite harm; or promote or propagate hatred, based on one or more of the following grounds: (a) age; (b) albinism; (c) birth; (d) colour; (e) culture; (f) disability; (g) ethnic or social origin; (h) gender or gender identity; (i) HIV status; (j) language; (k) nationality, migrant or refugee status; (l) race; (m) religion; (n) sex, which includes intersex; or (o) sexual orientation, is guilty of an offence of hate speech."[3]
>
> "It is not unfair discrimination to take measures designed to protect or advance persons or categories of persons disadvantaged by unfair discrimination or the members of such groups or categories of person."[4]

Based on these, we define hate speech as any unfairly discriminatory expression that demonstrates a clear intention to be harmful or to incite harm; promote or propagate hatred against a person or group of persons. In addition, offensive speech is defined is any fair or unfair expression that is not hate speech but discriminatory against a person or group of persons, while free speech is any expression that justifies the freedom of expressions' right. It is neither hate speech nor offensive speech.

II. DISTINCTIVENESS OF SOUTH AFRICAN TWEETS

South Africa is a multilingual society with more than seven formal languages such as Afrikaans, IsiZulu, IsiXosha, Sesotho, Setswana, Venda, English and others. English is the language of business and government [15]. Thus, majority of the citizens use it as second language. It is as a result of this that most communications in public space like Twitter are in English. However, some of the tweets might contain any other native languages for expressivity and convenience purposes, where they might be used as noun, pronoun, adjective, adverb, verb, conjunction, preposition or interjection. Examples are given in the following tweets:

- Did #*Orania* participate on #Elections2019?
- *Kana Orania* still exists. So many years in democracy 🙆🏾‍♂️
- *Ima* bad man wit good intentions 🤗😊😏😏!!
- Book me to play at weddings, I do funerals and divorce parties as well, hit up my manager *swart* for details.

The analysis of the South African words in the tweets is presented as follows:

- Orania is a noun (name of a minority Afrikaner community in South Africa [16]).
- Kana is an adverb (it is IsiZulu word for 'when')
- Ima is a verb (it is IsiZulu word for 'stop')
- Wit is an abbreviation for 'with' in this context (but it is an Afrikaans word for 'white').
- Swart is a noun (it is Afrikaans word for 'black')

Also, non-standard English words, specific to South Africa might be used. Examples include those presented as follows:

- *Needa* ask *yo momma* how you should treat a man with polish..fuck wrong *wit em*
- *Hai ho* it's travelling time - Eurovision week *tho* 🎉

The analysis of the non-standard English words is presented as follows:

- Needa is a modified English term (it is used for 'need' in IsiZulu)
- Yo is an abbreviation (it is used for 'you')
- Momma is a noun (it is a name of a person)
- Wit em is an abbreviation (used as 'with them')
- Tho is an increment

[2] ttps://www.justice.gov.za/legislation/constitution/SAConstitution-web-eng-02.pdf

[3] https://www.justice.gov.za/legislation/hcbill/B9-2018-HateCrimesBill.pdf

[4] https://www.justice.gov.za/legislation/acts/2000-004.pdf

- Hai ho is a noun phrase (it is an IsiZulu phrase for 'alive to')

A careful consideration of the examples above shows that that apart from alphanumeric lexicons, there are other lexicons such as punctuations, emoticons, emojis and special symbols. Therefore, identifying the meanings of expressions using standard English dictionary would be challenging and might be more challenging for machine learning.

III. RELATED WORKS

Based on the survey of previous works on hate speech detection, none has focused on offensive and hate speech detection for South African tweets. However, many works have employed different features and machine learning algorithms for detection of ternary English tweets. Wazeem and Hovy [4] annotated 16,914 tweets related to an Australian TV show 'My Kitchen Rules' consisting 3,383 sexist, 1,972 racist and 11,559 belonging to neither of them. Features such as character n-grams combining unigram, bigram, trigram and four-gram added to gender, location and length of tweets excluding spaces were evaluated. With Logistic Regression classification over 10-fold validation, a combination of n-gram and gender had the best performance best F1-score of 0.7393 follow by char n-gram with 0.7389. However, the identified genders were unreliable because they depended on the usernames and other indicators in the profiles. Only 52.34 percent of the tweets could be identified. In [17], the dataset used in [4] was consolidated with additional class belonging to both racist and sexist. Character n-gram had the best performance with F1-score of 91.24 using logistic regression algorithm. Watanabe *et al.* [6] combined three different datasets such as crowdflower,[5] crowdflower[6] and multiclass tweets [4]. The combined dataset of 23,010 tweets consisted clean, offensive and hateful tweets in the same proportion in both training and test sets. Some kinds of sentiment, semantic, unigram and pattern-based features were extracted and optimized to obtain the maximum accuracy. Machine learning algorithms such as Random Forest, Support Vector Machines and J48graft were used to perform both binary and ternary classifications. The result showed that J48 graft ternary classification with cross validation recorded the best accuracy and F1-score of 0.784 and 0.784, respectively for combination of all the features, while it recorded 0.877 and 0.878 for binary classification. A machine learning method was proposed to detect hate speech and offensive language on twitter using n-gram features weighted with TFIDF values in [2]. By using hyperparameter optimization model, logistic regression with the n-gram range from 1 to 3 had the best accuracy of 95.6 percent. The outcome showed that 4.8 percent of the offensive tweets were misclassified as hateful.

Zhang [18] solved the problem of detecting 'long-tail' hate speech in twitter datasets that lack unique and discriminative features using deep neural networks (DNN).

DNN such as combination of convolutional neural networks (CNN) and gated recurrent unit (GRU) (CNN-GRU) and combination of CNN and skipped CNN were evaluated using pretrained word embeddings with n-gram features. The datasets used in [19], [20] as well as different models of dataset used in [17] were used for the analyses. The results showed that both methods recorded least and highest micro F1-score of 0.83 and 0.94 as against the previous 0.78 and 0.91. However, the DNN relies on large pretrained embeddings or corpus, which are not available in the context of South African English. Davidson *et al.* [19] developed a 25k corpus of tweets for automatic classification of tweets into hate speech, offensive language, and neither, in which only 5% were hate speech.. The features used were part of speech features derived from NLTK[21], sentiment lexicon feature, sentence quality score, n-gram features as well as syntactic features such as numbers of hashtags, user mentions, retweets, URLs, characters, words and syllables in each tweet. Logistic Regression with L1 regularization was first used to reduce the dimensionality of the data followed by Naïve Bayes, Decision Trees, Random Forests and Linear Kernel Support Vector Machines. Logistic Regression performed better than other models with overall precision, recall and F1 score of 0.91, 0.90, and 0.90, respectively. Malmasi and Zampieri [10] focused on discrimination of hate speech from profanity using Twitter dataset. The dataset consisted 14,509 tweets, with 2,399 hate speech, 4,836 offensive and 7,274 neither. Character n-gram, word n-gram, skip-gram, Brown cluster, Brown cluster with skip-gram and their combination were analyzed with single and ensemble classifiers. Char n-gram with Support Vector Machine had the best performance of 0.78 accuracy for single classifier. The Stacking-based metaclassifier, with Support Vector Machine had the best performance of 79.8 accuracy and 0.45 F1-sore for the minority hate speech class compared to other ensemble classifiers such as plurality voting, mean probability voting, etc. MacAvaney *et al.* [5] proffered solution to the detection of hate speech using multi-view stacked Support Vector Machine, in which different features of n-gram were extracted. Applying the method on Stormfront [22], TRAC [23], Hatebase Twitter [19] and HatEval,[7] it had the best performance, with accuracy of 0.6121 and macro F1 of 0.5368 for TRAC and second best performance, with accuracy of 0.8033 and macro F1 of 0.8031for Storm compared to other novel techniques.

The work of [10] motivated this work; however, the following contributions make our work different:

1) development of a realistic Twitter corpus to address the challenges of detecting offensive and hate speech in South African tweets,

[5] https://www.crowdflower.com/data-for-everyone/
[6] https://www.data.world/crowdflower/hate-speech-identification/
[7] https://competitions.codelab.org/competitions/199935

2) formulation of distinctive features to effectively distinguish South African offensive and hate speech from free speech, and
3) evaluation of machine learning algorithms based on hyper-parameter optimization, ensemble and multi-meta-learning models to effectively detect offensive and hate speech in highly imbalanced corpus of tweets.

IV. METHODOLOGY

The following steps were carried out to achieve the goal of this paper, which is to effectively detect offensive and hate speech in South African tweets. The diagram in Fig. 1 presents the framework used in the study.

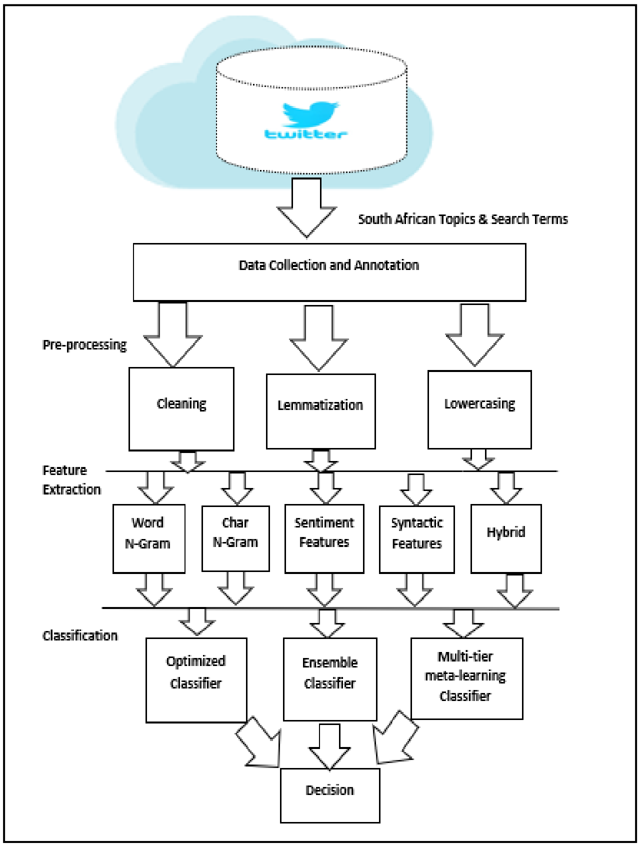

FIGURE 1. Framework for hate speech detection in South African tweets.

A. DATA COLLECTION AND ANNOTATION

Total of 21,350 tweets of South African discourses on Twitter between the period of May 5, 2019 and May 13, 2019 were collected using Twitter Archiver,[8] a publicly available plugin for Google Sheets, which is based on Twitter Search API. The collection targeted tweets related to 2019 South African national elections, popular South Africa individuals and trending issues such as land reclamation, Orania and white communities. Non-English tweets were removed except code-mixed English tweets, with Afrikaans, IsiZulu and Sesotho words.

Also, repeated tweets as well as tweets with empty word characters were also removed.

Six South Africans citizens, who were familiar with national issues in South Africa and English literates were recruited and trained to annotate tweet corpus as either hate speech 'HT', offensive speech 'OFF' or free speech 'FS'. Apart from English, all the annotators were literate in one other South African language. Two were literate in Afrikaans; two were literate in IsiZulu and two were literate in Sesotho. 15,702 tweets that are remaining after cleaning were divided into three samples. The first sample containing 7,100 English, code-mixed English and Afrikaans tweets was annotated by two Afrikaans annotators; the second sample containing 4,500 English, code-mixed English and IsiZulu tweets was annotated by two IsiZulu annotators, while the last sample containing 4,102 English, code-mixed English and Sesotho tweets was annotated by two Sesotho annotators.

The rules that were used for the annotation are as follows:

1) A tweet is a hate speech if:
- it is targeted against a person or group of persons.
- it uses derogatory or racial slur words repetitively within the tweet
- it makes use of disparaging terms with the intent to harm or incite harm.
- it refers to and supports other hateful facts, hate tweets and organization.
- it makes use of idiomatic, metaphorical, collocation or any other indirect means of expressions that are harmful or may incite harm
- it expresses violent communications

2) A tweet is an offensive speech if:
- it is targeted against a person, group of persons or organization.
- it is not a hate speech.
- it abuses a target using profane, derogatory or slur words.

3) A tweet is a free speech if:
- It is targeted or not targeted against a person, group of persons or organization.
- it is neither hate speech nor offensive speech
- it is expressed by government or licenced agency of government with the intent for mass advocacy and enlightenment.

Some examples of tweets and their annotations are presented as follows:

- im in the studio man... WHAT 😐✋ man quit playin wit me man.... nah fr DONT play *liek* that..... a-are u serious? how u kno... put that on everything.....damn,,,, ima call u back. IMA CALL U BACK!!!!(FS)
- Racists really want preferential treatment and admissions to our government schools. They can honestly *fuck* right off! Not on our taxes, must *fuck* off to *Oran*ia. SA will be desegregated, like it or not *mofos* (HT)

[8] https://digitalinspiration.com/product/twitter-archiver

TABLE 1. Quantity and Kappa score of annotated samples.

Sample	Tweet Type	Number of Tweets Annotated	Kappa Scores(K)	Percentage of Tweets with Full Agreement
Sample 1	English, English with Afrikaans	7,100	0.837	91.92
Sample 2	English, English with IsiZulu	4,500	0.579	93.80
Sample 3	English, English with Sesotho	4,102	0.633	98.68

- If you don't understand that what #trevornoah just said is dangerous and can justify a war then your short sighted as fuck, just because he mentioned Julius Malema in a bad way you forget that he literally just mentioned Genocide in South Africa. wake up black child (OFF)
- Ready to get on this P and pineapple juice. Yess ready to get a lil drunk wit my man (FS)
- If it turns out that Mihlali's lying, I will not regret having believed her. I will always take the word of *umuntu omnyama* over whites, until presented with the facts, any day. I will die on this hill. (HT)
- Can't believe people believe whites 💀 over *umuntu omnyama* (HT)
- Why did the DA white voter base vote ff+. They didnt want Miamane leading them and yena he dreams of equal South Africa. Vuka *muntu omnyama.* (FS)
- There's a Xhosa chick somewhere asking for *Imali yokuvota* (OFF)
- When mealie pap and flour goes through the fermentation process in order to produce Mageu, its a must for it to be stored in a hot area. It must feel the heat but at end *umuntu omnyama* will have a delicious cup of mageu. I hope you understand coach (FS).

After applying Cohen Kappa statistics [24] on the three annotated samples, inter-annotator reliability agreement scores (Ê) for Afrikaans, IsiZulu and Sesotho annotators were 0.837, 0.579 and 0.633, respectively. Table 1 summarizes the quantity and inter-annotator agreement scores for the different samples. Since none of the inter-rater agreement score was less than moderate, the annotations were relied upon. To improve the reliability of the corpus, the tweets from all samples that both annotators agreed to (full agreement) were merged into a corpus. The total number of tweets in the corpus was 14,896.

The corpus statistics presented in Table 2 showed the total and average distribution of words and characters in the dataset.

TABLE 2. Corpus statistics.

Parameters	Values
Total number of unique words	28,912
Number of unique English words	27,466
Number of unique non-English South African words	1,446
Average number of words per tweet	23.76
Average length of tweet	142.75

B. TOKENIZATION AND DATA PREPROCESSING

The count indicators of syntactic and sentiment information were first extracted as stated in (IV-C) before tokenizing and preprocessing the tweets. The procedure used for cleaning and formatting of tweets include the following steps:

- tokenization from TweetTokenizer [21] in NLTK[9] because it handles emoticons, HTML tags, URLs, retweets, user mentions and Unicode characters correctly.
- stemming from wordnet lemmatizer [21] in NLTK.[10]
- removal of username
- removal of punctuations
- removal special characters and symbols including emoticons and emojis
- removal of hash symbols in hashtags
- removal of English stop words. Stop words of other languages were not removed because they may be meaningful cues in English and offensive contexts. Example 'bane' is a stop word in IsiZulu and Sesotho, but it is a negative word in English.
- change of all texts to lower case.

C. FEATURE EXTRACTION

All the features used in [3] with modifications were used for our analysis.

1) WORD N-GRAM FEATURES

Word n-gram features are counts of sets of sequential N words per tweet, where N is the number of words in the tweets, which may range from 1 to N. In this study, we evaluated unigram (n = 1) and bigram (n = 2) word features because of their good performances in previous works [2], [3], [10]. In order to optimize their performance, the n-grams were weighted by term frequency-inverse document frequency (TF-IDF) [2], which offset the number of word in a document by the frequency of the word (term) in a corpus. The TF-IDF for a given term t in a document (tweet) d is given as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) * \text{IDF}(t) \qquad (1)$$

when $\text{IDF}(t) = \log [\, n / (\text{DF}(t) + 1]$

[9] https://www.nltk.org/
[10] https://www.nltk.org/

n = total number of documents in the document set
DF(t) = document frequency of t
The weighted n-gram is given as:

$$W(t,d) = NGram(t,d) \times TF\text{-}IDF(t,d) \tag{2}$$

The word n-gram feature space resulted in total of 28,912 unigram and 120,706 bigram word tokens.

2) CHARACTER N-GRAM FEATURES

Character n-gram (or simply char n-gram) features are counts of sets of sequential N alphanumeric characters per tweet, where N is the number of alphanumeric characters in the tweets, which may range from 1 to N. In this study, we evaluated trigram (n = 3) and four-gram (n = 4) character features because of their good performances in [10]. The n-grams were also weighted by TF-IDF as presented above. The character n-gram feature space resulted in total of 10,466 trigram and 59,573 four-gram character tokens.

3) SYNTACTIC-BASED FEATURES

The following syntactic information were extracted because they are linguistic features that determine the underlying grammatical structure of sentences and documents in English. We employed their count indicators (frequency of occurrence) rather than their binary indicators for more explicit contextualization.

- capital letters such as A, B, ..., Z
- small letters such as a, b, ..., z
- uppercase words such as COME, RIGHT
- lowercase words such as dog, land
- length of tweets including spaces
- alphanumeric words such as Red, black, White
- exclamation marks such as !
- question marks such as ?
- full stops such as.
- quotes such as “”, ‘’
- special characters such as @, #, $, %, ^, &, ∗, _,etc
- hash tags marked with characters such as #Orania

4) NEGATIVE SENTIMENT-BASED FEATURES

Hate speech are often marked by negative meanings [25]. Negative terms such as negative polarity scores and lexicons, emoticons and emojis have been used for sentiment analysis [26]. In the contexts of hate speech detection, count indicators of the negative sentiment features except negative polarity score, which relies on mathematical formulae were joined to the count of English, Afrikaans and IsiZulu slur words in Hatebase [27] to improve the features.

- negations
- negative words based on Opinion Lexicon[28]
- negative emoticons based on Urban Dictionary[11]
- negative emojis based on Twitter based on Urban Dictionary[12]
- Hatebase slur words

TABLE 3. Dataset distribution.

Class	Training	Test
HS	283	85
OFF	943	347
FS	9,946	3,292
Total	11,172	3,724

D. CLASSIFICATION

Every category of features and their hybrid based on vertical stacking in Python 3.6 [29] were analyzed. We evaluated different hyper-parameter configurations of machine learning classifiers such as Logistic Regression (LogReg), Support Vector Machine (SVM), Random Forest (RF) and Gradient Boosting (GB) for optimal performance. In addition, we combined the four classifiers using different ensemble models. We also applied multi-tier meta-learning model similar to multi-tier meta-learning model proposed by [30], [31] and evaluated it with LogReg, SVM, RF and GB meta-learners. All classifier trainings were performed using ten-fold cross-validation followed by testing. The dataset was split into 75/25 training dataset (n = 11,172) and testing dataset (n = 3,724) as presented in Table 3.

As shown in Table 3, the class distribution of the experimental corpus was highly imbalanced, where the majority of the tweets were free speech. In machine learning classification, class imbalance can lead to decrease performance and accuracy [32]. Therefore, we reduced the imbalance by applying synthetic minority oversampling technique(SMOTE)[33] because of its performance in previous work [9].

The SMOTE algorithm is presented as follows:

```
SMOTE (T, N, k)
Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k
Output: (N/100) ∗ T synthetic minority class samples
Step 1. (∗ If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd. ∗)
Step 2. if N < 100
Step 3.     then Randomize the T minority class samples
Step 4.     T = (N/100) ∗ T
Step 5.     N = 100
Step 6. endif
Step7. N = (int)(N/100) (∗ The amount of SMOTE is assumed to be in integral multiples of 100. ∗)
Step 8. k = Number of nearest neighbors
Step 9. numattrs = Number of attributes
Step10. Sample [ ][ ]: array for original minority class
            samples
Step11. newindex: keeps a count of number of synthetic
            samples generated, initialized to 0
Step12. Synthetic [ ][ ]: array for synthetic samples
            (∗ Compute k nearest neighbors for each
            minority class sample only. ∗)
Step13. for i ← 1 to T
```

[11] https://www.urbandictionary.com/define.php?term =emoticon
[12] https://www.urbandictionary.com/tags.php?tag =emoji

```
Step14.    Compute k nearest n eighbors for i, and save
             the indices in the nnarray
Step15.    Populate (N, i, nnarray)
Step16. endfor Populate (N, i, nnarray) (∗ Function to
           generate the synthetic samples. ∗)
Step17. while N != 0
Step18.    Choose a random number between 1 and k,
             call it nn. This step chooses one of the k
             nearest neighbors of i.
Step19.    for attr ← 1 to numattrs
Step20.        Compute: dif = Sample[nnarray[nn]]
                   [attr] − Sample[i][attr]
Step21.        Compute: gap = random number
                   between 0 and 1
Step22.        Synthetic[newindex][attr]=
                   Sample[i][attr]+gap ∗ dif
Step23.    endfor
Step24.     newindex++
Step25.      N = N − 1
Step26. endwhile
Step27. return (∗ End of Populate. ∗)
```

The algorithms for the implementation of the hyper-parameter configuration, ensemble classifier and multi-tier meta-learning models are presented as follows:

1) HYPER-PARAMETER OPTIMIZATION

Exhaustive grid search was performed over all possible hyperparameter configurations to choose the optimal setting for the classifiers.

Given classifiers, S= (1, …, m) with parameters, R = (1, …, r) and values, $V = (v_{11}, \ldots, v_{mp})$, the V corresponding to the highest prediction score is selected.

```
Hyper-parameter ( )
Input: Values $V = (v_{11}, \ldots, v_{mp})$
Output: Max score of V
Do for S=1 to m
    Do for I = 1 to h
        Do for R=1 to r
            $Score_{sum} \leftarrow 0$
            Do for CV_list = 1 to 10 // Cross-validation
                Model←S (train, R, V)//for train and test set
                Score ←S (test, Model)
                $Score_{sum} \leftarrow Score_{sum+Score}$
            $Score_{val} \leftarrow Score_{sum} /10$
Max ($Score_{sum}$)
```

The hyper-parameters combination settings are presented in Table 4.

2) ENSEMBLE CLASSIFIER MODEL

We evaluated majority and weighted voting ensembles for their performances in [9]. We also evaluated meta-learning model for its performance in [5], [10]. In majority voting ensemble, the voting classifier counted the number of unique class labels for each test instance and assigned to a test instance a class label that was voted by majority of the classifiers, while in weighted voting, the class labels were weighted by the average of brute-force probability scores assigned to the classifiers to classify the test instance. The meta-learning ensemble model combined different sets of predictions from word n-gram, character n-gram, syntactic and negative sentiment features and applied generalized stacked ensemble [34] to predict the final label.

TABLE 4. Hyper-parameter combination settings.

Algorithm	Parameters	Values
LogReg	C	log(-4), log(4), log(20)
	Penalty	L1, L2
	Solver	Liblinear
SVM	Kernel	Linear
	C	0.001, 0.01, 0.1, 1, 10, 100
GB	Learning Rate	0.001, 0.01, 0.1
	N-estimators	100, 200, 300
	Min_samples_split	100, 200, 300
	Max_depth	4, 5, 6, 7
	subsample	0.5, 0.6,1
RF	N-estimators	10, 30, 50, 80, 100
	Max_depth	10, 30, 50, 80, 100
	Min_samples_leaf	1, 2, 4
	Min_samples_split	2, 5, 10

Given S base classifiers (1, …, m), let $P = (P_{11}, \ldots P_{mn})$ be the set of predictions for different features, F= (F1, …, F_n) of instances I. The predictions of the classifiers are combined using weighted, majority voting or meta-learning ensemble E for labelling decision L= (1, 2, 3) using the algorithms presented as follows:

```
Majority Voting ( )
Input: Labels L predicted by S
Output: Labels $L_p$ predicted by E
Do for I = 1 to h
    Do for S=1 to m
        Do for F= 1 to n
            Do for L= 1 to 3
                Max_Count ← M
                If Count > M/2
                    L← label corresponding to Max_count
                EndIf
End
Weighted Voting ( )
Use brute force to find the optimal weight of the base learners W= {$w_1$, $w_2$, $w_3$)
$W_T = w_1 + w_2 + w_3$
Do for I = 1 to h
    Do for L= 1 to 3
        $W_{sumIL} \leftarrow 0$
        Do for Q =1, …, q //weak classifiers
            $W_{sumIL} \leftarrow W_{sumIL+Wq*}P_{qL}$
        $W_{avg} \leftarrow W_{sumIL/}W_T$
        L ← label corresponding to highest $W_{avg}$
End
Meta-learning ( )
Input: Labels L predicted by S
Output: Labels $L_p$ predicted by E
//Initialize predictions for each feature in horizontal axis
Do for D= 1 to d
    //initialize predictions for each feature in vertical axis
    Do for M = 1 to m
        If D =! M Then
            Construct a new dataset T(d,m)
            LT ← Train(T(d,m))
            $L_P$ ← Test(LT, $L_{test)}$
        Endif
End
```

Fig. 2 indicates the classical meta-learning model.

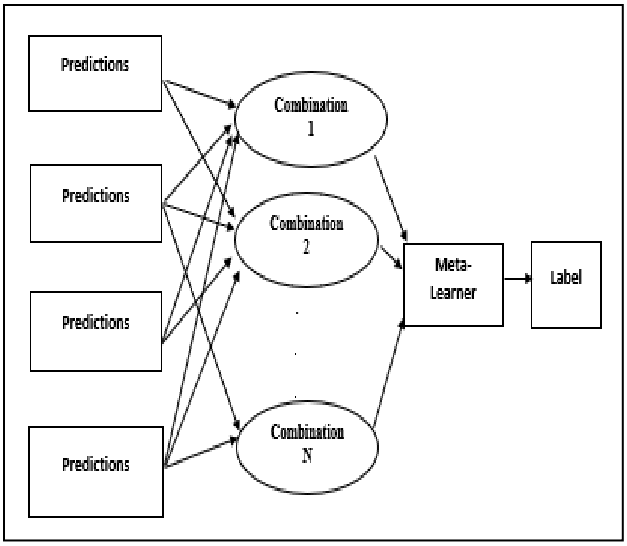

FIGURE 2. Meta-learning model with multiple features.

3) MULTI-TIER META-LEARNING MODEL

The multi-tier stacked generalization meta-learning model applied two-phases of stacked ensemble. The first phase involved stacked ensemble of all classifier predictions. The second phase involved stacked ensemble of meta-learning predictions. The stacked ensemble of meta-learning predictions made use of heuristics over different meta-levels and sets of meta-features to select the best prediction labels. The Fig. 3 indicates the multi-tier meta-learning model.

```
Multi-tier meta-learning ( )
Input: The predictions $P^o$ corresponding to $L_p$ by the
        meta-learners.
Output: The labels L predicted by the multi-tier
        meta-learning classifiers G.
Do for R= 2 to 4 // meta-levels
    Do for U = 2 to 4//meta-features
        B[X] ← Instant (R, U)
```

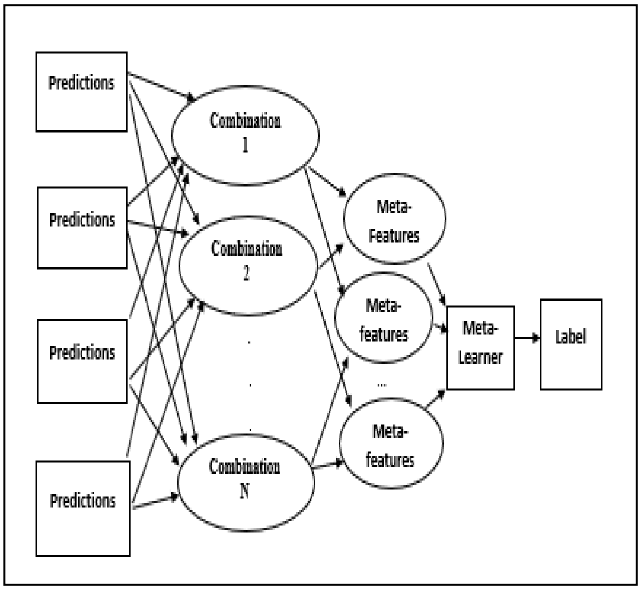

FIGURE 3. Multi-tier meta-learning model with multiple meta-features.

```
            Z ← Meta-learning (B[X], $P^o$)
        If $Z_{ru}$ >$Z_{r+1,u+1}$
        L ← $Z_{ru}$(Label corresponding to best prediction for G)
        Endif
End
```

E. PERFORMANCE METRICS

The performance of the different techniques used for the classification problem were evaluated using true positive rate (TPR), accuracy (Acc) and macro-averaging for precision (P), recall (R) and F1-score (F1) since we were dealing with multiclass problem. The macro-averaging obtained performance measures from each of the k one-vs-all matrices separately and calculated their average. In deciding best algorithm during classification, TPR or also known as sensitivity was chosen over accuracy since it indicates the fraction of correctly predicted tweets per class. The formula for TPR is:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{3}$$

where TP = number of instances of class with label L that is predicted correctly, TN = number of instances of class with label L that is predicted incorrectly.

V. EXPERIMENTAL RESULTS

Sci-kit learn [35], mlxtend [34] and Python 3.6 [29] were used to implement the models. We present the TPR of the hyper-parameter optimized model for LogReg, SVM, RF and GB algorithms for each of the features and their combinations in Table 5, while Table 6 indicates their precision, recall, and F1. We also present the TPR of the ensemble models for each of the features and their combinations in Table 7, while Table 8 indicates their precision, recall, and F1. Furthermore, we present the TPR of the multi-tier

TABLE 5. TPR and accuracy for optimized LogReg, SVM, RF, GB models on the test set.

Feature	Class	LogReg	SVM	RF	GB
Word n-gram	HT	0.294	0.317	0.035	0.529
	OFF	0.723	0.726	0.406	0.867
	FS	0.935	0.930	0.966	0.804
	Acc	0.900	0.897	0.893	0.803
Char n-gram	HT	0.152	0.894	0.094	0.658
	OFF	0.688	0.069	0.631	0.763
	FS	0.959	0.701	0.972	0.864
	Acc	0.915	0.646	0.921	0.850
Syntactic Features	HT	0.235	0.152	0.176	0.317
	OFF	0.495	0.340	0.331	0.452
	FS	0.527	0.722	0.553	0.737
	Acc	0.517	0.674	0.686	0.538
Negative Sentiment Features	HT	0.200	0.211	0.223	0.258
	OFF	0.276	0.236	0.161	0.276
	FS	0.620	0.619	0.684	0.552
	Acc	0.578	0.574	0.625	0.519
Hybrid	HT	0.341	0.247	0.094	0.494
	OFF	0.743	0.648	0.682	0.757
	FS	0.943	0.434	0.975	0.904
	Acc	0.911	0.449	0.928	0.881

TABLE 6. Precision, Recall and F1 for optimized LogReg, SVM, RF, GB models on the test set.

Feature	Metric	LogReg	SVM	RF	GB
Word n-gram	P	0.61	0.61	0.55	0.52
	R	0.65	0.66	0.47	0.73
	F1	0.62	0.62	0.50	0.56
Char n-gram	P	0.62	0.65	0.66	0.59
	R	0.60	0.55	0.57	0.76
	F1	0.61	0.35	0.59	0.62
Syntactic Features	P	0.37	0.37	0.38	0.39
	R	0.42	0.41	0.42	0.44
	F1	0.32	0.36	0.37	0.34
Negative Sentiment Features	P	0.35	0.35	0.35	0.35
	R	0.37	0.36	0.36	0.36
	F1	0.32	0.31	0.32	0.30
Hybrid	P	0.63	0.38	0.69	0.60
	R	0.68	0.44	0.58	0.72
	F1	0.68	0.30	0.61	0.63

TABLE 7. TPR and accuracy for ensemble models on the test set.

Feature	Class	Weighted Voting	Majority Voting	Meta-learning
Word n-gram	HT	0.294	0.258	0.129
	OFF	0.708	0.737	0.556
	FS	0.942	0.665	0.960
	Acc	0.905	0.915	0.903
Char n-gram	HT	0.200	0.211	0.105
	OFF	0.662	0.685	0.622
	FS	0.942	0.962	0.970
	Acc	0.899	0.919	0.918
Syntactic Features	HT	0.235	0.235	0.164
	OFF	0.389	0.397	0.317
	FS	0.662	0.662	0.737
	Acc	0.627	0.627	0.679
Negative Sentiment Features	HT	0.000	0.211	0.235
	OFF	0.000	0.198	0.175
	FS	1.000	0.665	0.658
	Acc	0.883	0.611	0.603
Hybrid	HT	0.011	0.000	0.058
	OFF	0.605	0.567	0.507
	FS	0.986	0.987	0.982
	Acc	0.929	0.925	0.917

TABLE 8. Precision, Recall and F1 for ensemble models on the test set.

Feature	Metric	Weighted Voting	Majority Voting	Meta-learning
Word n-gram	P	0.62	0.63	0.60
	R	0.65	0.65	0.55
	F1	0.63	0.64	0.57
Char n-gram	P	0.59	0.65	0.72
	R	0.60	0.62	0.62
	F1	0.60	0.64	0.67
Syntactic Features	P	0.38	0.38	0.37
	R	0.43	0.43	0.40
	F1	0.36	0.36	0.36
Negative Sentiment Features	P	0.29	0.36	0.36
	R	0.33	0.36	0.36
	F1	0.31	0.32	0.32
Hybrid	P	0.75	0.58	0.66
	R	0.53	0.52	0.52
	F1	0.56	0.54	0.56

meta-learning model for LogReg, SVM, RF and GB algorithms in Table 9 while Table 10 indicates their precision, recall, and macro-F1 scores. Figure 4 depicts how the TPR scores vary with different techniques, while Figure 5 depicts how the precision, recall and F1 vary with different techniques. The confusion matrices for the best models for detection of both offensive and hate speech (multi-objective) are presented in Table 11, Table 12, and Table 13.

The results in Table 5 showed that both word n-gram and character n-gram features had the best performances among the features. The optimized model of GB with word n-gram technique recorded the best TPR of 0.867 for detection of offensive speech, while the SVM with char n-gram technique had the best TPR of 0.894 for detection of hate speech. The results in Table 6 for precision, recall and F1 also showed that word and character n-gram had the best performance.

TABLE 9. TPR and accuracy results for multi-tier meta-learning models on the test set.

Meta-feature	Class	LogReg	SVM	RF	GB
Word +character +syntactic +negative_ sentiment	HT	0.211	0.858	0.858	0.858
	OFF	0.916	0.887	0.887	0.887
	FS	0.764	0.646	0.646	0.646
	Acc	0.765	0.671	0.674	0.674

TABLE 10. Precision, Recall and F1 results for multi-tier meta-learning models on the test set.

Meta-Feature	Class	LogReg	SVM	RF	GB
Word +character +syntactic +negative_ sentiment	P	0.47	0.50	0.50	0.50
	R	0.63	0.80	0.80	0.80
	F1	0.49	0.50	0.50	0.50

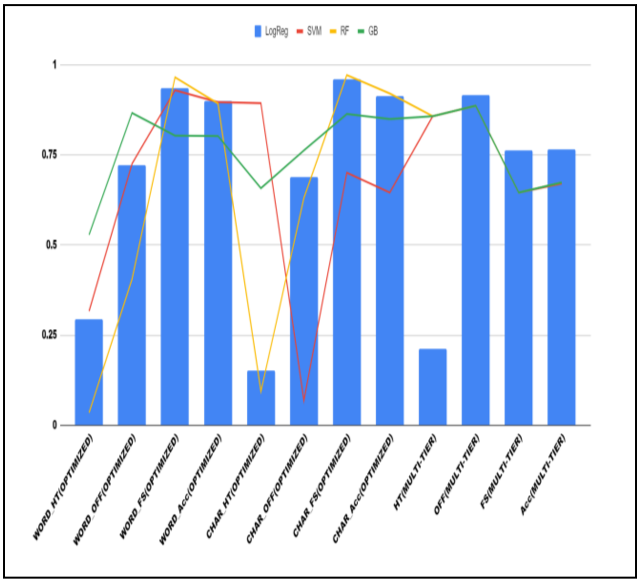

FIGURE 4. TPR for the best techniques.

The result in Table 7 and Table 8 showed that the weighted, majority voting and meta-learning ensemble models, most especially word n-gram and character n-gram only performed well in the detection of offensive speech, with TPR of 0.708 and 0.737, respectively but poorly in the detection of hate speech. Word n-gram and character n-gram also had the best TPR, macro precision and F1 scores. The results in Table 5, Table 6, Table 7 and Table 8 showed that the hybrid features trailed the n-gram features in performances, while negative sentiment despite the notion that hate speech have negative sentiment [25] and syntactic-based features performed worst in the detection of hate speech.

The results in Table 9 showed that multi-tier meta-learning with three meta-features comprising word n-gram, character n-gram and syntactic meta-features recorded the best TPR of 0.858 and 0.887 for hate speech and offensive speech, respectively. Specifically, SVM, RF and GB had the best TPR for hate speech, while LogReg had the best TPR for offensive speech. There was negligible difference of 0.01 between the F1 scores of LogReg and the trio of SVM, RF and GB in Table 10.

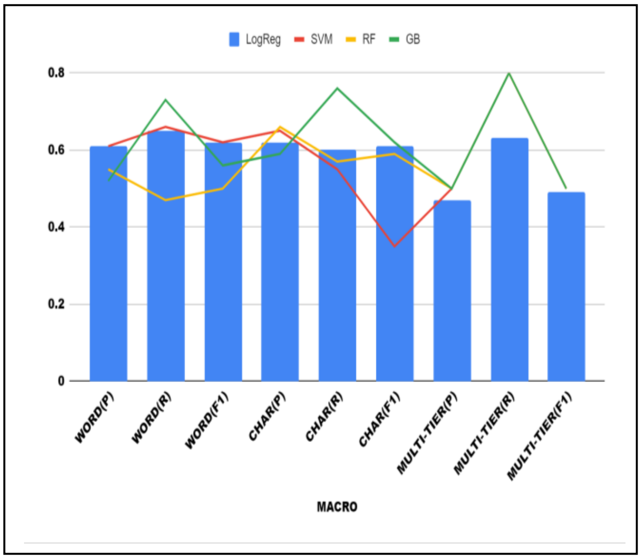

FIGURE 5. Precision, Recall and F-measure for the best techniques.

TABLE 11. Confusion matrix for hyper-parameter optimization model of word n-gram.

Class	FS	OFF	HT
FS	2647	415	230
OFF	29	301	17
HT	36	4	45

TABLE 12. Confusion matrix for hyper-parameter optimization model of character n-gram.

Class	FS	OFF	HT
FS	2845	143	304
OFF	66	265	19
HT	25	4	56

TABLE 13. Confusion matrix for the multi-tier meta-learning model.

Class	FS	OFF	HT
FS	2129	428	735
OFF	17	308	22
HT	8	4	73

The analyses of the confusion matrices for the best techniques in Table 11, Table 12 and Table 13 showed that there were more balances in the classification of each class of tweets for multi-tier meta-learning classifier than optimized GB for word n-gram and SVM for character n-gram features, which recorded the highest TPR for offensive and hate speech, respectively. By appying Cochran's Q test [36] at significance (α) equal to 0.05 on the best classification techniques, we obtained p-value of 7.04×10^{-96} meaning the classifiers did not perform equally.

Performing multiple post hoc pair-wise tests using McNemar [37] to determine which pairs have different population proportions, p-values equal to 1.46×10^{-75} was obtained for pair of hyper-parameterized GB with word n-gram and hyper-parametrized SVM with character n-gram, 1.27×10^{-89} for hyper-parameterized GB with word n-gram and multi-tier meta-learning model, and $0.00212(2.12 \times 10^{-3})$ for hyper-parameterized SVM with character n-gram and multi-tier meta-learning. The p-values which were lower than 0.05 showed that the classifiers performed differently.

The analysis of Fig. 4 showed that the peak TPR for optimized GB for word and SVM for char n-grams were in the same range (0.85-0.9) as TPR for SVM, RF and GB multi-tier meta-learning model with TPR of 0.858 and 0.887 for hate speech and offensive speech, respectively. The analysis of Fig. 5 showed that the multi-tier meta-learning model recorded the highest recall. Altogether, the precision, recall and F1 for SVM, RF and GB multi-tier meta-learning models were not less than 0.5, 0.8 and 0.5.

The results showed that optimized GB models of word n-gram and SVM model of character n-gram as well as SVM, RF and GB multi-tier meta-learning model had the best performances. However, the classifiers performed differently on the test set. Therefore, they should be applied to complement one another. The results further showed that despite the effectiveness of word and character n-gram features in detection of hate speech and offensive speech, the good performance of one, led to the poor performance of the other. Also, when the n-gram features were combined with syntactic and sentiment-based features using vertical stacking or ensemble classifiers, the performances were poor. However, the combination of word n-gram, character n-gram, syntactic-based features, and negative sentiment-based features using multi-tier meta-learning techniques recorded consistently good performances in detection of hate speech and offensive speech. Both gradient boosting and support vector machine were the best in terms of performance in all models.

VI. ERROR ANALYSIS

A total 143 misclassified samples from 300 test set sample were manually analyzed to understand the challenge of detecting offensive and hate speech using classification approach. The challenges are presented as follows:

A. ABSENCE OF NON-DISCRIMINATIVE FEATURE

Majority of the misclassified tweets (88.11%) were in this category, where the tweets were misclassified as a result of:

1) Frequency of certain features in other classes. Examples include:

- *This girl dead on said Xhosa men and butch lesbians are synonyms*
- *# AfterVotingIExpect Xhosa women to stop cheating. Phela Xhosa women don't just cheat. They cheat mercilessly, they show no mercy. The kinda cheating that when you find out, you have no choice but to join the church choir.*

The above tweets were annotated as offensive speech but were classified as hate speech by all classifiers because of lack of any feature to discriminate it from hate speech.

- *Y'all better stop playing wit my man Polamalu*

The above tweet was annotated as free speech but was classified as offensive speech by all classifiers except multi-tier meta-learning model because of the presence of negative words found in offensive speech.

2) Meta-feature limitation, in which the inadequacy of meta-features employed in multi-tier meta-learning was responsible for misclassification. Examples include:

- *BBCNEWS 9:17pm SOUTH AFRICA ELECTION. Early results show governing ANC ahead O1 its rivals SOUTH AFRICA ELECTION. ANC is on track to...*
- *Such exciting times in wealthy South Africa with an economy that's growing at such a phenomenal rate that we're funding tertiary education, great healthcare and we can obviously afford an election re-run. Not to mention the holiday which the economy can easily handle.*

The tweets above were annotated as free speech, but multi-tier meta-learning model classified them as hate speech.

B. NON-IDENTIFICATION OF CONTEXTS

The second largest reason for misclassification (10.48%) was as a result of inability of classifier to identify the contexts of the tweets, which help to determine the class of tweets. Examples include:

- *I sometimes enjoy @user and his wit. But he also is a stubborn young man who makes a fool of himself. This is not a good look for him. Stick to safe interviews if you don't want folks to ask real questions.*

It was annotated as free speech but was classified as offensive speech by all classifiers because of the presence of profane words found in offensive speech. The contextual information(advice) could not be captured.

- *"We have been taught to hate ourselves, we have been taught that black skin is inferior" @user*

It was annotated as free speech but was classified as offensive speech by all classifiers except meta-learning because of the presence of profane words found in offensive speech. The contextual information (self-targeted) could not be captured.

C. IMPLICITNESS ISSUE

Implicitness involves background and external information outside of the tweets and represents 1.38 percent of the misclassified errors. Examples include:

- *We are not a patriotic country. That's why you offer people land for saying 1 Sotho word. That's why Jan van Riebeck and and friends just took everything.*

By Implicitness, the annotators labelled the tweets as hate speech but hyperparameter optimization model of word feature misclassified it as free speech. However, the multi-tier meta-learning and char n-gram model correctly classified it as hate speech.

- *Remember the battle of Isandlwane, blood river mAfrika okhokho bekhusela umhlaba, vote @user, # SAElections2019*

By Implicitness, the annotators labelled the tweets as hate speech. The hyperparameter optimization model of both word and character n-gram features misclassified it as free speech. However, the multi-tier meta-learning model classified it correctly as hate speech.

The error analyses showed that multi-tier meta-learning reduced the misclassification errors due to implicitness, discriminative features and contexts by 34.26%. However, multi-tier meta-learning would not correctly classify 15.38% of tweets, which n-gram features correctly classified.

VII. FEATURE ANALYSIS

Information Gain (IG)[38] was applied to measure the impact that different word features of tweets have on the detection of offensive and hate speech. A high IG score indicates that the feature has a greater impact on the detection. Table 14 and Table 15 present the top twenty highest ranking features for offensive and hate speech according to the ranking of IG scores, respectively.

TABLE 14. Ranking of features based on IG scores for Offensive Speech.

SN	Feature	IG score
1	white	0.0653
2	fuck	0.0265
3	wit	0.0186
4	man	0.0183
5	indie	0.0138
6	Ramaphosa	0.0105
7	bitch	0.0081
8	Africa	0.0075
9	South	0.0074
10	election	0.0059
11	nigga	0.0046
12	Malema	0.0046
13	ANC	0.0045
14	don't	0.0035
15	Dick	0.0033
16	president	0.0032
17	Cyril	0.0031
18	shit	0.0031
19	music	0.0027
20	shower	0.0027

The analysis of Table 14 showed that the most important features of offensive speech were 'white', with IG score of 0.0653 followed by 'fuck' with IG score of 0.0265. They were both English words. In fact, all the twenty most important features of the offensive speech were English words, some of which are slur words such as 'fuck', 'bitch', 'nigga', 'dick' and 'shit'. Despite the closeness of the IG scores of the features showing that all were important, the word 'white' however had much greater impact on the determination of offensive speech than other features based on the IG score of 0.0653. On the overall, the most informative features of offensive speech in South African tweets were English terms and slur words, which IG scores ranged between 0.002 and 0.06.

TABLE 15. Ranking of features based on IG scores for Hate Speech.

SN	Feature	IG score
1	monna	0.0097
2	Xhosa	0.0073
3	indie	0.0054
4	ke	0.0027
5	ho	0.0025
6	ramaphosa	0.0025
7	Africa	0.0021
8	South	0.0021
9	wa	0.0020
10	election	0.0017
11	wit	0.0016
12	guy	0.0014
13	ANC	0.0013
14	dribble	0.0013
15	man	0.0013
16	ba	0.0012
17	music	0.0012
18	hun	0.0011
19	killed	0.0010
20	girl	0.0010

The analysis of Table 15 showed that the most important features of hate speech were 'monna', 'Xhosa' and 'indie', with IG scores of 0.0097, 0.0073 and 0.0055, respectively. In fact, six of the top twelve most important features were IsiZulu words, such as 'monna', 'ke', 'ho', 'wa' and Afrikaans words, such as 'indie' and 'wit', many of which are predicates. In fact, none of the important non-English South African words was slur word or derogatory term.

Therefore, the most informative features of hate speech in South African tweets were non-English South African words, which were predicates. However, none of the features of the tweets was too significant to determine hate speech based on their low IG scores which ranged between 0.001 and 0.009 unlike the important features of the offensive speech.

VIII. CONCLUSION

In this work, we have collected an English corpus of South African tweets for offensive and hate speech detection. The corpus was annotated by multilingual annotators because the tweets consisted different cues from South African languages. Four distinctive feature sets and their combinations were extracted from the tweets after tokenization and preprocessing. Three categories of improved machine learning models such as hyper-parameter optimization, ensemble and multi-tier meta-learning were applied on different machine learning algorithms such as Logistic Regression, Support Vector Machine, Random Forest and Gradient

Boosting to classify the tweets as either hate speech, offensive speech or free speech.

The outcomes of the experiment showed that Support Vector Machine, Random Forest and Gradient Boosting multi-tier meta-learning model were the most consistent and balanced in terms of detection of offensive and hate speech with true positive rate of 0.887 and 0.858 and overall accuracy of 0.671. However, the optimized Gradient Boosting with word n-gram recorded the best true positive rate of 0.867 for offensive speech with overall accuracy of 0.803, while it recorded low true positive rate of 0.529 for hate speech. Optimized Support Vector Machine with character n-gram recorded the best true positive rate of 0.894 for hate speech with overall accuracy 0.646, while it recorded very low true positive rate of 0.069 for offensive speech. The tests of the null hypotheses whether there was no difference among the classification performances showed that there were differences in their performances. Therefore, multi-tier meta-learning and optimized models with word n-gram and character n-gram would complement one another. The error analysis showed that misclassification errors due to implicitness, discriminative features and contexts in hyper-parameter optimized model with n-gram features was reduced by 34.26% when complemented by multi-tier meta-learning classifier. The feature analysis showed that English slur words were the most informative features for determining offensive speech, while non-English South African predicates were the most informative features for determining hate speech.

In future, we will focus on improving the detection of hate speech and offensive speech by hybridizing optimized Support Vector Machine and Gradient Boosting classifiers and multi-tier meta-learning classifiers. Novel word dense embeddings will also be developed and evaluated with the proposed algorithms and deep neural network algorithms (for example LSTM) to detect offensive and hate South African tweets.

REFERENCES

[1] S. Dredge. (2014). Twitter Changes: 20 Hits and Misses From the Social Network's History. The Guardian. Accessed: Sep. 10, 2019. [Online]. Available: https://www.theguardian.com/technology/2014/oct/22/twitter-changes-hits-misses-history

[2] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, "Detecting hate speech and offensive language on Twitter using machine learning: An N-gram and TFIDF based approach," in *Proc. IEEE Int. Advance Comput. Conf.*, Sep. 2018, pp. 1–5.

[3] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval)," in *Proc. 13th Int. Workshop Semantic Eval. (SemEval)*, 2019, pp. 75–86.

[4] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in *Proc. NAACL Student Res. Workshop*, 2016, pp. 1–6.

[5] 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, Aug. 2019, Art. no. e0221152.

[6] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018.

[7] E. Greevy and A. F. Smeaton, "Classifying racist texts using a support vector machine," in *Proc. 27th Annu. Int. Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2004, pp. 468–469.

[8] P. Saha, B. Mathew, P. Goyal, and A. Mukherjee, "HateMonitors: Language agnostic abuse detection in social media," Sep. 2019, pp. 1–8, *arXiv:1909.12642v1*. [Online]. Available: https://arxiv.org/abs/1909.12642v1

[9] M. A. Fauzi and A. Yuniarti, "Ensemble method for Indonesian Twitter hate speech detection," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 11, no. 1, p. 294, Jan. 2019.

[10] S. Malmasi and M. Zampieri, "Challenges in discriminating profanity from hate speech," *J. Exp. Theor. Artif. Intell.*, vol. 30, no. 2, pp. 187–202, Mar. 2018.

[11] N. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D.-Y. Yeung, "Multilingual and multi-aspect hate speech analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Language Process. (EMNLP-IJCNLP)*, 2019, pp. 1–10.

[12] J. Clement. (2019). *South Africa: Digital Population as of January 2019*. Accessed: Nov. 17, 2019. Statista. [Online]. Available: https://www.statista.com/statistics/685134/south-africa-digital-population/

[13] PeaceTech Lab. (2019). *Monitoring and Analysis of Hateful Language in South Africa Report #6*. Accessed: Nov. 15, 2019. [Online]. Available: https://www.peacetechlab.org/south-africa-report-6

[14] T. De Smedt, S. Jaki, E. Kotzé, L. Saoud, M. Gwózdz, and G. De Pauwand Walter Daelemans, "Multilingual cross-domain perspectives on online hate speech," CLiPS, Comput. Linguistics Psycholinguistics, Univ. Antwerp, Antwerp, Belgium, Tech. Rep. CTRS-008 and 8, Sep. 2018, pp. 1–24. [Online]. Available: https://www.uantwerpen.be/clips

[15] F. Duncan. WORLDVIEW: Few South Africans Speak English, so Why is it the Language of Business and Politics. BizNews. Accessed: Jul. 4, 2019. [Online]. Available: https://www.biznews.com/premium/2018/06/28/english-language-business-politics

[16] E. Kotzé and B. Senekal, "Employing sentiment analysis for gauging perceptions of minorities in multicultural societies: An analysis of Twitter feeds on the afrikaner community of Orania in South Africa," *J. Transdiscipl. Res. South. Africa*, vol. 14, no. 1, pp. 1–11, Nov. 2018.

[17] Z. Waseem, "Are you a racist or am i seeing things? Annotator influence on hate speech detection on Twitter," in *Proc. 1st Workshop NLP Comput. Social Sci.*, 2016, pp. 138–142.

[18] Z. Zhang and L. Luo, "Hate speech detection: A solved problem? The challenging case of long tail on Twitter," *Semantic Web*, vol. 10, no. 5, pp. 925–945, Sep. 2019.

[19] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. 11th Int. AAAI Conf. Web Social Media (ICWSM)*, no. 11, 2017, pp. 512–515.

[20] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on Twitter," in *Proc. 1st Workshop Abusive Lang. Online*, 2017, pp. 41–45.

[21] E. L. S. Bird, *Analyzing Texts With Natural Language Toolkit: Natural Language Processing With Python*, 1st ed. Newton, MA, USA: O'Reilly Media, 2009.

[22] O. De Gibert, N. Perez, A. García-Pablos, and M. Cuadros, "Hate speech dataset from a white supremacy forum," in *Proc. 2nd Workshop Abusive Lang. Online (ALW2)*, 2018, pp. 11–20.

[23] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proc. 1st Workshop Trolling, Aggression Cyberbullying (TRAC)*, 2018, pp. 1–11.

[24] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.

[25] N. Frydas, "Monitoring and detecting online hate speech," Mandola, Brussels, Belgium, Tech. Rep. D1.5, 2017. [Online]. Available: http://www.mandola-project.eu/

[26] J. C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, "Detecting and monitoring hate speech in Twitter," *Sensors*, vol. 19, no. 21, p. 4654, Oct. 2019.

[27] HateBase. 2019. *HateBase: The World's Largest Structured Repository of Regionalized, Multilingual Hate Speech*. Accessed: Apr. 20, 2019. [Online]. Available: https://hatebase.org/

[28] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, 2015th ed. 2015. [Online]. Available: https://www.cambridge.org/za/academic/subjects/computer-science/knowledge-management-databases-and-data-mining/

[29] PythonTM. (2017). *Python 3.6.4*. Accessed: Apr. 25, 2019. [Online]. Available: https://www.python.org/downloads/release/python-364/

[30] R. Pari, M. Sandhya, and S. Sankar, "A multi-tier stacked ensemble algorithm for improving classification accuracy," *Comput. Sci. Eng.*, vol. 1, no. 1, p. 1, 2018.
[31] R. Pari, M. Sandhya, and S. Sankar, "A multi-tier stacked ensemble algorithm to reduce the regret of incremental learning for streaming data," *IEEE Access*, vol. 6, pp. 48726–48739, 2018.
[32] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[33] N. V. Chawla, K. W. Bowyer, and L. O. Hall, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[34] S. Raschka, Mlxtend 0.9.0. 2017.
[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[36] S. Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack," *J. Open Source Softw.*, vol. 3, pp. 24–25, Apr. 2018.
[37] Q. Mcnemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.
[38] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

OLUWAFEMI ORIOLA received the B.Sc. degree (Hons.) in computer science from Adekunle Ajasin University, Akungba Akoko, Nigeria, in 2006, and the M.Sc. and Ph.D. degrees from the University of Ibadan, Nigeria, in 2010 and 2015, respectively. He is currently a Postdoctoral Fellow in the Department of Computer Science and Informatics, University of the Free State, Bloemfontein, South Africa, with research focus on application of machine learning for detection of abusive South African languages in micro-blogging social media. His research interest includes corpus and cyber-security threat intelligence.

EDUAN KOTZÉ received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees in computer information systems from the University of the Free State, Bloemfontein, South Africa, in 1998, 2003, and 2008, respectively. He is currently the Head of the Natural Language Processing and Data Warehousing Research Group, Department of Computer Science and Informatics, University of the Free State. His research interests include algorithms and natural language processing (NLP) techniques to process large sets of unstructured data, business intelligence, and data warehousing adoption.