Cyberbullying Detection in Social Networks: Artificial Intelligence Approach

Nureni Ayofe Azeez[1,*], Sunday O. Idiakose[1], Chinazo Juliet Onyema[2] and Charles Van Der Vyver[3]

[1]*Department of Computer Sciences, University of Lagos, Nigeria*
[2]*Department of Computer Science, Federal University of Technology, Owerri*
[3]*School of Computer Science and Information Systems, North-West University, Vanderbijlpark Campus, South Africa*
E-mail: nurayhn1@gmail.com; idiakosesunday@gmail.com; chinazo.onyema@futo.edu.ng; Charles.VanDerVyver@nwu.ac.za
[*]*Corresponding Author*

Received 11 February 2021; Accepted 16 May 2021;
Publication 18 June 2021

Abstract

Over the past decade, digital communication has reached a massive scale globally. Unfortunately, cyberbullying has become prevalent, with perpetrators hiding behind the mask of relative internet anonymity. In this work, efforts were made to review prominent classification algorithms and also to propose an ensemble model for identifying cases of cyberbullying, using Twitter datasets. The algorithms used for evaluation are Naive Bayes, K-Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest, Linear Support Vector Classifier, Adaptive Boosting, Stochastic Gradient Descent and Bagging classifiers. Through experimentations, comparisons were made with the classifiers against four metrics: accuracy, precision, recall and F1 score. The results reveal the performances of all the algorithms used with their corresponding metrics. The ensemble model generated better results while Linear Support Vector Classifier (SVC) was the least effective

Journal of Cyber Security and Mobility, Vol. 10_4, 745–774.
doi: 10.13052/jcsm2245-1439.1046

of all. Random Forest classifier has shown to be the best performing classifier with medians of 0.77, 0.73 and 0.94 across the datasets. The ensemble model has shown to improve the results of its constituent classifiers with medians of 0.77, 0.66 and 0.94, as against the 0.59, 0.42 and 0.86 of Linear Support Vector Classifier.

Keywords: Cyberbullying, machine learning, detection, algorithms, twitter, cybercrime, social media.

1 Introduction

Social media networks have been hugely popular in the past one and a half decade. Platforms such as Facebook, Telegram, Snapchat, Instagram, Twitter, etc., have facilitated efficient, real-time communication among millions of people around the world, cutting across political and geographical boundaries (Whittaker and Kowalski, 2015). Untoward antisocial human behaviour has steadily found its way into this utopia (Lauw et al., 2010). These platforms, with a certain level of anonymity they provide, have seen aggressive, violent, sexist and other discriminatory and harmful comments, posts and exchanges directed towards other members of these platforms. This phenomenon is known as cyberbullying. Cyberbullying has become an important part of policy making for social media platforms as they recognize the harm it can cause through its adoption. Cyberbullying has been rampant in recent times and cyberbullies are now hacking and threatening users anonymously and perpetrating various nefarious activities without hiding. Sampasa-Kanyinga et al. (2014) discovered some effects of cyberbullying to include increase in the risk of suicide, depression, mental breakdown, substance and drug abuse, reduced productivity, poor performance in schools or places of work, low self-esteem, harm on oneself and many more.

Twitter is a popular social media network with over 500 million users that allows users to type, read and send 140-character messages commonly known as tweets. The platform generates on a daily basis over 500 million tweets. Twitter also contains unique features such as followership, usernames, profile pictures, locations, biography, hashtags, retweets, etc. It is estimated that about 80% of twitter users tweet through the use of their mobile phones. The media platform is already becoming a playground for cyberbullies (Algaradi et al., 2016). This work aims to utilize behavioural markers present in tweet content to identify and detect cyberbullying in a global social media networks by using a machine learning approach.

2 Related Works

Gutierrez-Esparza et al. (2019) presented a research on classifying incidents of cyber-aggression on common social media platforms most precisely for Spanish language users in a country like Mexico. The classifiers used are Random Forest classifier algorithm, OneR and Variable Importance Measures (VIMs). The choice of these classifiers was to aid the classification of aggressive and belligerent posts into three categories: violence and aggression based on sexual orientation, violence against opposite sex (women) and racial discrimination. The Random Forest approach was executed in two phases: the comment extraction and feature selection. Feature selection in random Forest was done with VIM before classification was implemented. OneR classifier evaluated the rate of occurrence of each term in the data set and also computed the average frequency of the occurrence of all the terms in the data set. The results of the experiment conducted with OneR classifier showed improvements on the classification of the three cyber aggression cases with above 90% accuracy when compared to Random Forest. The metrics used for evaluation are accuracy, confusion matrix, sensitivity (TRP), specificity (SPC) and Kappa Statistic.

Zahra (2016) focused on the research of cyberbullying mitigation and detection on Instagram. This study focuses on tackling and addressing ways to mitigate depression, suicide and anxiety resulting from the occurrence of cyberbullying. The author used Naïve Bayes Classifier. The author also investigated the design and effectiveness in using technological procedures and mechanisms to curb cyberbullying with the aid of tertiary prevention on the Instagram social media platform.

Nandhini et al. (2016) used an information retrieval algorithm to detect and classify cyberbullying activities. They proposed a framework for observing and grouping cyberbullying activities such as provocation, blazing, bigotry and psychological oppression. The algorithm utilized in this examination was Naïve Bayes classifier for arranging the cyberbullying movement as well as Lavenshtein calculation for cyberbullying identification. The mean exactness gotten from fornmspring.me showed a 93.79% precision and a 94.59% precision when myspace.com was considered.

The work of Amrita (2017) was directed at displaying collaborative detection of Cyberbullying demeanour in the data of twitter. This exploration is significant in recognizing cyberbullying conduct precisely by breaking down tweets progressively. This exploration presents another strategy known as circulated community-oriented methodology for distinguishing cyberbullying.

It includes a system of hubs for discovery which remains solitary and fit for characterizing tweets it gains. The hubs cooperate in circumstances where they need help with characterizing a tweet. The examination assesses various examples of coordinated effort and measures the exhibition of each example to detail.

Gomez-Adorno et al. (2018) detected aggressive tweets in Spanish language on twitter using machine learning techniques. The authors grouped tweets of 75% non-forceful and 25% forceful circulation. After training, the dissemination was 64.58% non-forceful tweets and 35.42% forceful tweets. This experimentation demonstrates that the measure of forceful tweets was a large portion of the non-forceful tweets.

Daniel (2017) used Support Vector Machine Model and N-grams in the automated recognition of cyberbullying and cyber harassment. They used these classifiers which had the option to arrange remarks on YouTube with a general precision of 81.8%. It later expanded to 83.9% when the misclassified remarks were added to the preparation set. The calculation is a multi-stringing one and can be run on numerous servers while the framework effectively determined the exactness by characterizing three remarks for every second. The algorithm had the negative precision of the class inside 10% of the positive class because of the balanced train set.

Van Hee et al. (2018) focused on the automatic detection of cyberbullying in social media text by modelling posts put in writing by bullies, victims and spectators of bullying on the internet. The framework was assessed on a manual which explained cyberbullying assortment of works for English and Dutch. It demonstrated that their methodology is relevant to various dialects provided that information to these dialects available and usable. A lot of paired grouping tests were performed to analyse the plausibility of programmed cyberbullying recognition via web-based networking media. Two classifiers were prepared on the English and Dutch assortment of works. Their analyses demonstrated that the methodology used was a reliable approach for the location of cyberbullying signals via web-based networking media. Afterward, the element and hyper parameter improvement of their models showed a score of 64.32% and a 58.72% for F1 for both English and Dutch languages respectively.

Haidar et al. (2017) focused their research on the mitigation and discovery of cyberbullying by building up a multilingual framework for cyberbullying identification. They conducted their research on cyberbullying in Arabic language with the use of AI. The framework utilized datasets from Twitter

Table 1 Summary of reviewed articles

Author	Year	Approach	Strengths	Weaknesses
Gutierrez-Esparza, et al.	2019	Classification of Cyber-Aggression Cases with the application of machine learning approach.	Improvements on the classification of the three (3) cyber – aggression scenarios with above 90% degree of accuracy compared to Random Forest.	Need for development and implementation of more tools that operate in Spanish Language, difficulty to get resources for algorithm training like data sets, corpora and others. Also, translation of the regional languages isn't much available on the internet making it difficult for researches to be done and inconvenient.
Van Hee et al.	2018	Automatic and real time detection of cyberbullying in social network text by modelling posts written by bullies, victims & spectators of bullying online	Promising technique for the uncovering cyberbullying signals on social network.	Difficulty in identifying and detecting victims
Gomez-Adorno et al.	2018	Logistic regression algorithm (detecting aggressive tweets in Spanish)	Application of SMOTE to solve imbalanced data which produced better output in the corpus.	Inability to solve imbalanced data problem with deep analysis of SMOTE process. No application of linguistic patterns
Daniel	2017	Support Vector Machine Model & N-grams in the automated identification of cyberbullying & cyber-harassment on YouTube.	Ability to classify comments on YouTube with a high degree of accuracy	Presence of non-cyberbullying comments classified as cyberbullying.
Haidar, Charmoun, and Serhrouchni	2017	Multilingual system for cyberbullying detection (Arabic language)	Showed that cyberbullying in Arabic language can be detected, It's a possibility.	Result isn't as perfect as that of detection in English language

(*Continued*)

Table 1 Continued

Author	Year	Approach	Strengths	Weaknesses
Amrita	2017	Collaborative detection of cyberbullying attitude and behavior in twitter data	Output displayed showed an improvement in recall and precision of the method adopted.	Inability in examining the possibility of choosing a node depending on previous tweets suggestions given.
Nandhini and Sheeba	2016	Naïve Bayes classifier for cyberbullying classification and Lavenshtein distance algorithm for cyberbullying detection	Produced high accuracy of over 90% for cyberbullying detection and classification.	Cannot be used for other social network platforms, only applicable to formspring.me and myspace.com
Zahra	2016	Cyberbullying detection and mitigation on Instagram.	Proposition of an executable method that evaluates the cyberbullying mitigation tool. It also provided a means to compare various solutions.	Unavailability of more datasets online.

and Facebook. With Naïve Bayes classifier, the result showed that at any rate, 33% of cyberbullying was perceivable in Arabic language.

In this work, nine classification algorithms were evaluated to determine their efficiency and reliability in identifying cases of cyberbullying by using Tweet content from Twitter datasets. The algorithms used for evaluation are Naive Bayes, K-Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest, Linear Support Vector, Adaptive Boosting, Stochastic Gradient Descent and Bagging classifiers.

Existing literature on this topic reports very imbalanced class distribution, sampling from potentially sensitive topics which tend to elicit untoward reactions from even the best of persons.

To the best of the authors' knowledge, there is no research output where these set of algorithms have been utilized for this purpose.

3 Methodology

3.1 Data Collection

The dataset utilized in this study was collected from publicly available data repositories on twitter. Data collection was limited to related tweets to keep the context of the data as similar as possible. Three datasets were collected, totalling 350,000 tweets. The data repository presents only data

that is publicly accessible via the Twitter Application Programming Interface (API) which is its primary data source that respects Twitter's policies on fair use and privacy protections (Eichstaedt et al., 2015).

The datasets include the following:

1. Tweets related to the Charleston church shooting incident.
2. Tweets relating to and collected during the Charlottesville riots
3. Tweets from suspected Russian bots during the 2016 United States of America Presidential Elections.

The datasets capture information about the tweet and the user of the tweet. This information includes the following user Id – the unique identifier for the publisher of the tweet, user name – the publishers unique name visible to all those viewing the users' profiles or tweets, screen name – a moniker chosen by the user; this moniker is not necessarily unique, user statuses count – the number of tweets made by the user as at when the data was collated, user favourites count – number of favourite tweets by the user, followers count – the number of twitter users subscribing to notifications on the content published by the user, user location – optional user provided location, user description – a short text snippet usually used by the user to describe himself or herself to other twitter users, often a form of biography, user time zone, full text content of the tweet, is retweeted – whether the tweet is the content retweeted from another twitter user, retweeted status text – the text content of the retweeted tweet, quoted status text – tweet that allows users to quote tweets and optionally add a tweet voicing their opinions on the initial tweet; the quoted tweet status is the original text of the quoted tweet, hashtags – the hashtags used in the tweet body, verified – whether the user publishing the content is verified or not, retweet count – how many people retweeted the tweet in question, mentions – how many twitter users were 'mentioned' by their usernames in the tweet – these users will get notified of the tweet.

3.2 Machine Learning Algorithms

Various features extracted from the tweets were utilized to develop and implement a model for uncovering cyberbullying behavior. Nine machine learning classification algorithms were utilized. They are discussed as follows:

1. **Naïve Bayes Classifier:** The purpose of using a Naïve Bayes Classifier is to predict the likelihood that an event will occur with the assistance of evidence that is present in the data. A multinomial Naïve Bayes algorithm classifier was used because it is suitable and more efficient

Table 2 Links to datasets

Dataset	Links
One	https://www.kaggle.com/limvaljean/charlottesville-and-twitter?select=aug15_sample.csv https://www.kaggle.com/limvaljean/charlottesville-and-twitter?select=aug16_sample.csv
Two	https://www.kaggle.com/limvaljean/charlottesville-and-twitter?select=aug17_sample.csv https://www.kaggle.com/limvaljean/charlottesville-and-twitter?select=aug18_sample.csv
Three	https://www.kaggle.com/vikasg/russian-troll-tweets?select=tweets.csv https://www.kaggle.com/vikasg/russian-troll-tweets?select=users.csv

for features that describe discrete frequency counts which is similar to the features of the data present in the dataset obtained.
Given a class variable or hypothesis (y) and a dependent feature or evidence $(x1, \ldots, xn)$
Therefore,

$$P(y|x1, x2, x3 \ldots .xn) = \frac{P(y)P(x1, x, x3, \ldots\ldots xn|y)}{P(x1, x, x3, \ldots\ldots xn)} \quad (1)$$

where:

$P(y)$ *are labels*

$P(x)$ *are comments*

$P(y|x1, x2, x3 \ldots .xn)$ is how probable was the hypothesis (labels) given the observed evidence (Zhang, H. 2004).
$P(x1, x2, x3 \ldots .xn|y)$ is how probable is the evidence, given that the hypothesis is true.
$P(y)$ is how probable was the hypothesis before observing the evidence.
$P(x1, x2, x3 \ldots\ldots xn)$ is how probable is the new evidence under all possible hypothesis.

2. **K-Nearest Neighbor (KNN) Algorithm:** This is a supervised machine learning classifier that memorizes observations from within a labelled test set to determine and predict arrangement and sorting labels for new and unlabelled observations. It forecasts based on how comparable training observations are to new incoming observations. K denotes the number of nearest neighbors. A case is categorized by a majority vote

of its neighbors, with the case being given to the class (Cyberbullying or non-cyberbullying) most common amongst its K nearest neighbors measured by a distance function. For instance, if k = 1, then the case is only assigned to the class of its nearest neighbor. Consider a rendition of the algorithm's pseudocode below (Zhang, Z. 2016):

Consider k as the desired number of nearest neighbors and

$S = p1, \ldots, pn$. *is the set of training samples*

$p1 = (xi,\ ci)$. *where xi is the d-dimensional feature vector of the point pi and ci is the class that pi belongs to.*

For each $p' = (x',\ c')$

Compute the distance $d(x',\ xi)$ *between* p' *and all* pi *belonging to* S

Sort all points pi according to the key $d(x',\ xi)$

Select the first k points from the sorted list, those are the k *closest training samples to* p'

Assign a class to p' *based on majority vote:*

$c' = argmaxy \sum_{k=0}^{n} (xi,\ ci)$

belonging to $S, I(y = ci)$

end

3. **Logistic Regression Classifier:** This is one of the most popular and most used machine learning classifiers. They are mostly used for binary classifications which produces a binary outcome between 0 and 1. The algorithm measures the relationship between one or more independent variable which are the features obtained from the dataset and the dependent variable which are the labels that is to be predicted by calculating probabilities using the logistic function also known as the sigmoid function. The values obtained are then converted into binary values to deduce a prediction. The sigmoid function is an S- shaped curve that can acquire any real valued number and place it into a value between 0 and 1 but never exactly those limits.

Suppose there is a training set $\{(x^{(1)}, y^{(1)}),\ (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$, composed of **m** samples, where $(x^{(1)}, y^{(1)})$ is the first sample, $x^{(m)}$ is the input variable of the **m**-th sample with $y^{(m)}$ as its output variable, where y^m is bound between 0 and 1.

The hypothesis function for logistic regression is generally given as follows:

$$h(x) = \frac{1}{1 + e^{Tx}} \tag{2}$$

An optimized cost function can be given as:

$$Cost(h(x), y) = -y\log(h(x)) - (1-y)\log(1-h(x)) \qquad (3)$$

(Bisaso et al. 2018).

4. **Decision Tree Classifier:** This is a supervised machine learning classifier that is used for classification as well as regression. It creates a model that determines the outcome of a target variable by the learning of basic decision rules determined and identified from the data features. The algorithm formulates a set of rules which are used for making predictions. The classifier takes two arrays as input, one is the array X (features for the training samples) and secondly, an array Y of integer values (0 & 1, labels for the training sample). The decision tree classifier is a flowchart like tree structure where a node represents a feature, the branches on the tree represent a decision rule and the leaf node represents the output.
   The entropy of a given information source x, H(x) is defined as follows:

$$\mathrm{H(x)} = \sum_{xeX} \mathrm{p(x)}\log p(x) \qquad (4)$$

   Where p(x) is the probability of occurrence of x (Quinlan, 1986).
5. **Random Forest Classifier:** This classifier can be used for classification and regression problems. It is a supervised classification algorithm that creates a forest with a number of trees as the name implies. In a random forest classifier, the greater the number of trees the greater the accuracy. It models a number of decision trees to create a forest without using the same attribute selection measures as in decision tree classifier. In random forest unlike decision tree classifier that used information gain or gini index to get the root node, it obtains the root node randomly while the splitting of the attribute nodes also occurs randomly (Khaled et al. 2014).
6. **Linear Support Vector Classifier:** This is a classifier that fits to the data as "best fit" hyperplane which splits or breaks the data into categories. After obtaining the hyperplane, the classifier is fed with some feature attributes to view what the predicted class is. This classifier is under the Support Vector Machine (SVM) algorithm. It makes use of a linear kernel (Aziz et al., 2019).
7. **AdaBoost Classifier:** Adaptive Boosting is an ensemble classifier (consists of various classifier algorithms) that has its result as the combined

output of the other classifier algorithms. The classifier combines weak classifier algorithms (in this case, decision trees) to produce strong classifier algorithms with the selection of training set at each iteration level and the designation of the correct amount of weight in the final voting. Any machine learning classifier algorithm can be the base classifier if it can accept the designation of weights of the training set. The algorithm is given as follows (Chengsheng et al., 2017):

Given: $(x_1, y_1), \ldots (x_m, y_m)$ *where* $x_i \in X, y_i \in \{-1, +1\}$
Initialize all weights of your samples to 1 divided by number of training sample
$D_1 = \frac{1}{m}$ *for* $i = 1, \ldots, m$
For $t = 1, \ldots, T$:

- *Train weak learner using distribution* D_t
- *Get weak hypothesis* $h_t : \chi \to \{-1, +1\}$
- *Select* h_t *with low weighted error*: $\varepsilon_t = Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$
- *Choose* $\propto = \frac{1}{2} ln(\frac{1-\varepsilon_t}{\varepsilon_t})$
- *Update for* $i = 1, \ldots, m$: $D_{t+1}(i) = \frac{D_t(i) e^{(-\propto_t h_t y_i(x_i))}}{Z_t}$

Where Z_t *normalization factor, (chosen so that* D_{t+1} *will be a distribution)*
With the final hypothesis given as:

$$H(x) \equiv sign \left(\sum_{t=1}^{T} \propto_t h_t(x) \right) \tag{5}$$

8. **Stochastic Gradient Descent Classifier:** With this classifier, minute samples are picked at random instead of the entire data set for each iteration. There is a term known as batch which consists of the total number of samples from a data set used for the calculation of each iteration's gradient. Stochastic Gradient Descent is used to handle a large dataset unlike Gradient Descent that cannot handle large or huge dataset because it will be very computationally expensive to execute. It solves this problem by using a batch size of one (sample) to execute each iteration. The sample is shuffled randomly and chosen for executing the iteration (Feng et al 2020).
$Y = mx + b$; the straight-line equation where m is the slope and b is its intercept
$m = m - \partial m; b = b - \partial b$; these are parameters with small change
$Cost = \frac{1}{N} \sum_{i=1}^{N} (Yi' - Yi)^2$; this is the cost function for N samples

9. **Bagging Classifier**: It is also an ensemble meta-estimator classifier where base classifiers (in this case, decision trees) are fitted on each random subsets of the original dataset and then calculate their individual predictions, either by averaging or voting to produce a final prediction. Introduction of randomization into its construction happens in this classifier. Each base classifier is trained in parallel accompanied by a training set that is obtained by randomly drawing with replacement, N data from the original training dataset where N is the original training set size. The training set for each of the base classifier is independent of each other. Bagging reduces overfitting by averaging or voting (Azeez et al., 2019).
   Bagging classifier algorithm includes the classifier generation and the classification:
   i. Classifier generation:
      a. *Let N be the size of the training set.*
      b. *For each of t iterations:*
         i. *Sample N instances with replacement from the original training set*
         ii. *Apply the learning algorithm to the sample*
         iii. *Store the resulting classifier*
   ii. Classification:
      a. *For each of the t classifiers:*
         i. *Predict class of instance using classifier*
      b. *Return most predicted class.* (Dey, 2018)
10. **Ensemble Classifier:** the proposed ensemble model is a technique combining multiple machine learning classifiers and models, attempting to produce better results that the constituent models. The constituent classifiers are individually trained on the dataset, predictions made. These predictions are then combined to make a final prediction. Different methods exist by which to make this final prediction including stacking, voting, bagging and boosting. In this paper, voting is used to make the final prediction. Voting is implemented here using predicted class labels for majority rule. The constituent estimators used in the ensemble are Multinomial NB, Linear SVC and Logistic Regression (Azeez et al., 2021).

11. **Justification for the choice of Multinomial NB, Linear SVC and Logistic Regression for Ensemble.**

The algorithms in the ensemble were chosen to provide a novel blend of algorithms, as the Random Forest, Ada Boost and Bagging are already meta estimators. While a mix of KNN and Logistic Regression may have provided better results, the paper showed similarly improved results could be obtained even using so called "weak learners".

Linear SVC (Support Vector Classifier) offers the following benefits:

i. It gives room for the application of different regularization in the formulation
ii. Linear classifier is relatively faster than non-linear classifier
iii. There is an approximation to a bound on the test error rate of SVC
iv. It performs relatively better whenever there is a clear margin of separation between classes.
v. SVM is relatively memory efficient

Also, Multinomial NB was also considered because of the following important rare features:

i. It is very easy to implement as the user only need to calculate probability.
ii. This algorithm can be used on both continuous and discrete data.
iii. It is very simple and can be used for predicting real-time applications.
iv. It is very scalable and can easily handle large datasets.

Finally, the authors decided to make use of Logistic Regression because of these advantages:

i. Logistic regression is easier to implement, interpret, and very efficient to train.
ii. It is very fast at classifying unknown records. Logistic regression is less inclined to over-fitting but it can over fit in high dimensional datasets.

3.3 Dataset Annotation

Eight thousand tweets were selected at random from the cleaned dataset, manually codifying them as either cyberbullying or non-cyberbullying. They were labelled with human coding with the assistance of three individuals familiar with twitter terminology, slang and parlance. These individuals include a graduate of Civil Engineering, a graduate of Veterinary Medicine and a Masters' degree student in Computer Science. They were introduced

to the prevailing definitions of cyberbullying in literature, with further orientation on the context in which the tweets were posted.

The tweets were classified as:

- Cyberbullying: if the tweet content manifests the indications and features of cyberbullying
- Non-Cyberbullying: if the tweet content does not manifest the indications of cyberbullying.

The tweets were considered cyberbullying if two of the codifiers labelled them as such. Non-English tweets, tweets containing only media or emojis were removed from the dataset. Using a select group of intellectuals familiar with the tweet context and twitter parlance yielded better coded results, though it was time consuming (Azeez et al., 2019).

3.4 Proposed Ensemble Model

The proposed Ensemble model combines three estimators viz Linear SVC, Multinomial NB and Logistic Regression. Its architecture is described in Figure 1.

The data is first collected and collated, manual data annotation is done. Data is cleaned, text is normalized, removing links and stop words. Count vectorization is done on the dataset, the constituent classifiers are used to build their respective models by splitting the data into test and train sets. The ensemble model is built by running a majority vote on the results of the individual constituent models.

3.5 Metrics Used for Evaluation

The relevant metrics and measures were computed. These metrics include False Positive, True Positive, True Negative, False Negative, F-score, Accuracy, Precision, and Recall and the boxplot characteristics (Inter-Quartile range, Median, Lower Quartile, Upper Quartile,).

- True Positives (TP) – These are the accurately predicted positive values. It implies that the value of actual class is 'Yes' and the value of predicted class is also 'Yes'. In this case, this is when a tweet is correctly identified as cyberbullying.
- True Negatives (TN) – These are the accurately predicted negative values. It implies that the value of actual class is NO and the value of

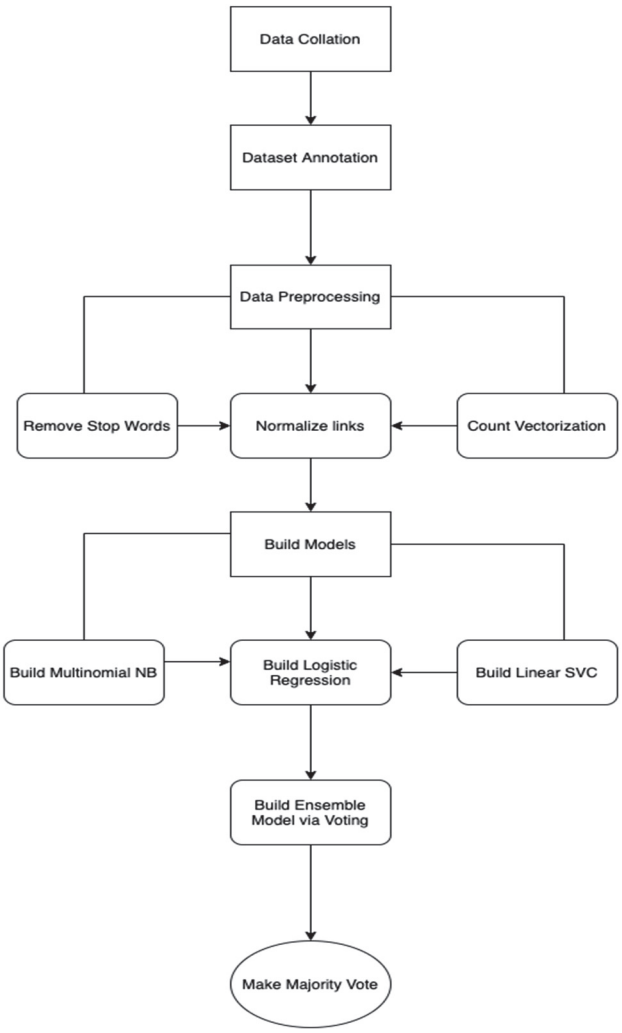

Figure 1 Architecture of proposed ensemble model.

predicted class is also NO. In this case, this is when a tweet is correctly identified as not being cyberbullying.

- False Positives (FP) – This occurs when the actual class is NO and predicted class is YES. It occurs when a tweet is incorrectly labelled as containing cyberbullying.

- False Negatives (FN) – This occurs when the actual class is YES but predicted class is NO. This happens when a tweet is incorrectly labelled as not containing cyberbullying.
- Accuracy: This is the most prominent performance measure. This is determined by finding the ratio of correctly predicted observation to the total observations. It is the ratio of correctly labeled tweets to the whole pool of tweets (Azeez et al., 2021).

$$\frac{(\mathrm{TP}) + (\mathrm{TN})}{(\mathrm{TP}) + (\mathrm{FP}) + (\mathrm{TN}) + (\mathrm{FN})}$$

  High accuracy does not necessarily correspond to the model's suitability. Accuracy is a great measure mostly for symmetric datasets where values of false positive and false negatives are almost same.
- Precision: This is the ratio of accurately predicted positive observations to the total predicted positive observations. In this case, precision responds to the question of all tweets labelled as cyberbullying such as: how many are actually instances of cyber bullying behavior?

$$\frac{(TN)}{(TN) + (FN)}$$

- Recall (Sensitivity): This is the ratio of accurately determined and predicted positive observations to the total observations in the entire class.

$$\frac{(TP)}{(TP) + (FN)}$$

- F1-Score: This is the weighted average of the precision and recall. It takes cognizance of both false positives and false negatives. F1 is usually more useful than accuracy, and this holds true in this work as an uneven class distribution is present. It is a very useful when seeking a balance between Precision and Recall.

$$\frac{2 \times \mathit{precision} \times \mathit{recall}}{\mathit{precision} + \mathit{recall}}$$

- Median (Middle Quartile): It denotes the mid-point of the data and is shown by the line that divides the box into two parts. It represents the most likely score.
- Inter-Quartile Range: The range of scores from lower to upper quartile is knowns as the inter-quartile range. Half of all scores will usually fall in this range.

- Upper Quartile: The upper quarter of all scores. Seventy-five percent of the scores fall below the upper quartile.
- Lower Quartile: The lower quarter of all scores. Twenty-five percent of scores fall below the lower quartile.
- Whiskers: The lower and upper whiskers denote scores outside the middle 50%. Whiskers often (but not always) stretch over a wider range of scores than the middle quartile groups.

4 Results Obtained

All classifiers were run against the three datasets to determine their performances with the considered metrics. Tables 3–5 show the results obtained. Figures 2–7 show the linear graphs visualizing the scores against the classifiers and boxplots visualizing the spread of the results.

Table 3 shows the results of the analysis of metrics for the classifiers using the first dataset. Evaluation metrics include accuracy, precision, F1-score and recall. The classifiers are k-neighbors classifier, Ada boost classifier, Bagging classifier, Decision tree classifier, Logistic regression, random forest classifier, multinomial Naïve Bayes, Stochastic Descent Classifier, the proposed Ensemble model and Linear Support Vector Classifier.

From the results shown in Table 3, Logistic regression is most accurate, closely followed by random forest; k neighbors is most precise, closely followed by the Ada boost and bagging classifiers; linear SVC is the least sensitive, exhibiting the lowest recall score.

Table 3 Results obtained by classifiers [Dataset 1 (Charleston)]

Classifier	Accuracy	Precision	Recall	F1 Score
KNeighborsClassifier	0.793677	0.719211	0.761899	0.73994
AdaBoostClassifier	0.788686	0.713281	0.762177	0.736919
BaggingClassifier	0.785358	0.705843	0.761847	0.732776
DecisionTreeClassifier	0.733777	0.703346	0.731516	0.717154
LogisticRegression	0.813644	0.662016	0.813644	0.73004
RandomForestClassifier	0.810316	0.708952	0.755415	0.731446
MultinomialNB	0.763727	0.685979	0.752979	0.717919
SGDClassifier	0.798669	0.690102	0.770739	0.728195
LinearSVC	0.570715	0.681112	0.557655	0.613232
Ensemble (Voting Class.)	0.808652	0.699381	0.764716	0.730591

Table 4 Results obtained by classifiers [Dataset 2 (Charlottesville)]

Classifier	Accuracy	Precision	Recall	F1 Score
BaggingClassifier	0.727669	0.667622	0.691798	0.679495
DecisionTreeClassifier	0.727669	0.682083	0.700238	0.691041
AdaBoostClassifier	0.710240	0.679380	0.686426	0.682885
RandomForestClassifier	0.749455	0.712815	0.630309	0.669028
KNeighborsClassifier	0.690632	0.607203	0.674019	0.638869
LogisticRegression	0.742919	0.554756	0.742919	0.635196
SGDClassifier	0.732026	0.565357	0.728583	0.636674
LinearSVC	0.381264	0.601311	0.360247	0.450562
MultinomialNB	0.300653	0.5290806	0.256486	0.345488
Ensemble (Voting Class.)	0.679739	0.620927	0.642607	0.631581

Table 5 Results obtained by classifiers [Dataset 3 (Russian Trolls)]

Classifier	Accuracy	Precision	Recall	F1 Score
BaggingClassifier	0.954264	0.913515	0.954264	0.933445
SGDClassifier	0.955814	0.913580	0.955814	0.934220
DecisionTreeClassifier	0.939535	0.917544	0.939253	0.928272
LogisticRegression	0.952713	0.914160	0.952713	0.933038
KNeighborsClassifier	0.955814	0.913580	0.955814	0.934220
AdaBoostClassifier	0.950388	0.913350	0.950388	0.931501
RandomForestClassifier	0.955039	0.913548	0.95504	0.933833
LinearSVC	0.840310	0.926350	0.83385	0.877670
MultinomialNB	0.565891	0.926626	0.550819	0.690927
Ensemble (Voting Class.)	0.953488	0.925012	0.944154	0.934485

The Ensemble model yielded improved results as compared to the constituent classifiers. With better precision and F1 score than all constituent classifiers. Logistic Regression yielded marginally better accuracy than the Ensemble model. Compared to the other classifiers and models, the ensemble model had the third best accuracy, it did not yield as good a precision, it however had a solid F1 score.

Figure 2 shows the combined line graph of the results obtained by evaluating the metrics for all the classifiers. The line graphs show that the classifiers performed best when measuring accuracy and perform least when measuring precision (selectivity). Linear SVC and Stochastic DG perform

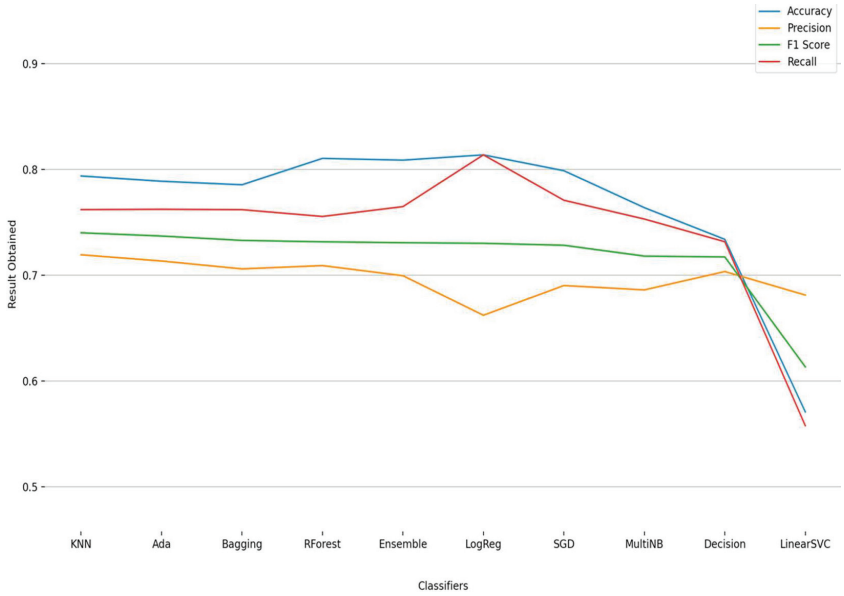

Figure 2 Line graph of Results Obtained v Classifiers (Dataset 1).

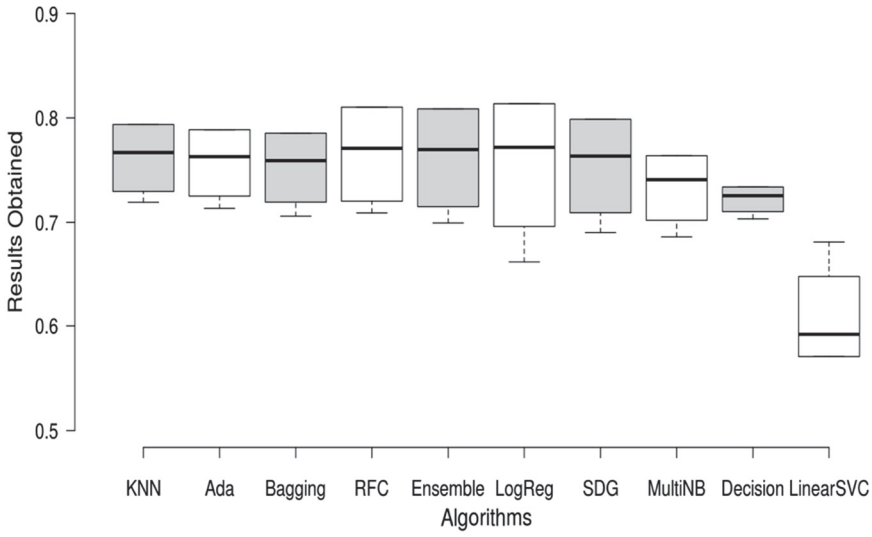

Figure 3 Analysis of results obtained against classifiers using Boxplots (Dataset 1).

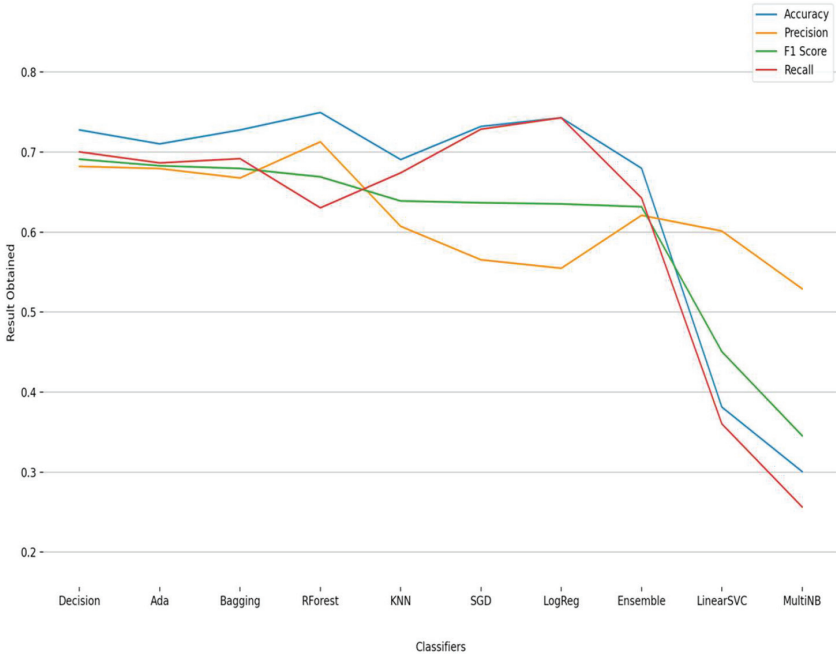

Figure 4 Line graph of Results Obtained v Classifiers (Dataset 2).

generally worse when compared with the rest classifiers except for precision where the performance is on par with the other metrics.

Figure 3 shows that Decision Trees, Multinomial NB and SDG Classifiers score similarly for all metrics, with Linear SVC scoring very poorly, its upper whisker (highest observed metric) barely grazing the Lower whiskers of other metrics. This implies that the highest scores for LinearSVC barely matches with the lowest scores from other algorithms. LinearSVC is best avoided for this dataset. KNN, Bagging, Logistic Regression, Ensemble & Random Forest classifiers have the joint highest medians. These classifiers will have the most likely scores regardless of choice of metric. Logistic Regression & Random Forest also record the highest scores outright, but they is significant variance in the metric scores, having inter quartile ranges of 0.11 and 0.12 respectively. Bagging and KNN have the highest upper whiskers while retaining lower inter quartile ranges at 0.8 and 0.6 respectively. They also have a similar median to the highest scoring upper whisker scores (Logistic Regression, Ensemble & Random Forest). The box plot shows Bagging and KNN will likely provide the most consistently high metric scores since they

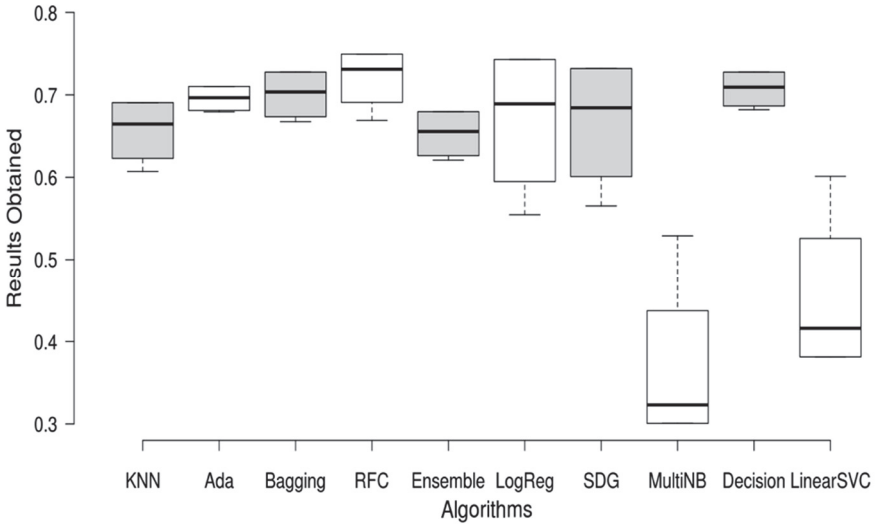

Figure 5 Analysis of results obtained against classifiers using Boxplots (Dataset 2).

have the combined highest upper whiskers together with the lowest inter quartile ranges and score very close to the highest median scores ($\sim$0.1 difference).

Table 4 shows the results of the analysis of metrics for the classifiers using the second dataset. Evaluation metrics include accuracy, precision, F1-score and recall. The classifiers are k-neighbors classifier, Ada boost classifier, Bagging classifier, Decision tree classifier, Logistic regression, random forest classifier, multinomial Naïve Bayes, Stochastic Descent Classifier, the Ensemble model and Linear Support Vector Classifier.

From the results in Table 4, Random Forest classifier is the most accurate, closely followed by Bagging classifier and Logistic Regression. Precision is generally lower than Accuracy, with Bagging classifier being the most precise which is 0.5 points worse off than its accuracy. While logistic regression has accuracy of 0.7429, its performance in precision was poor (0.5548). The F1 scores were generally similar, with Bagging classifier having the highest (0.688).

The Ensemble model has an average performance on the accuracy, precision and f1 score metrics. It again yielded better performance when compared to its constituent classifiers except for accuracy where it was slightly outperformed by Logistic Regression.

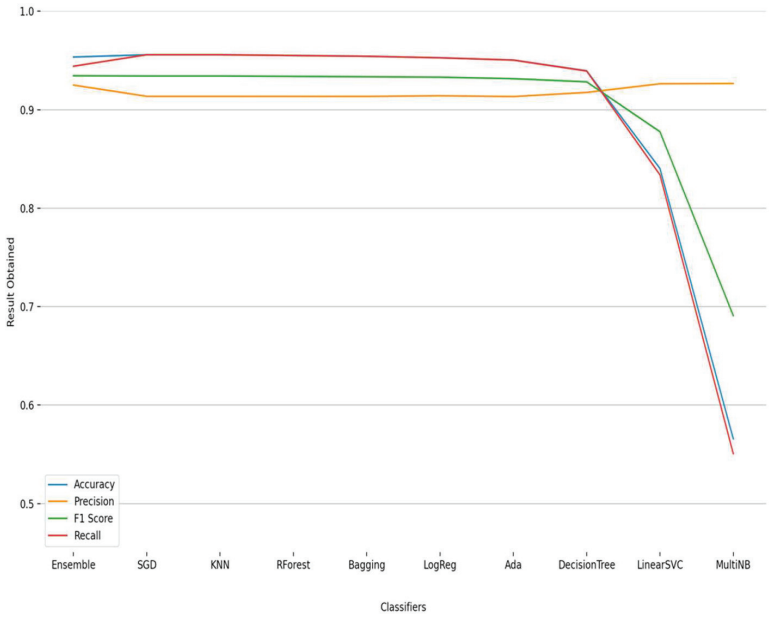

Figure 6 Line graph of Results Obtained v Classifiers (Dataset 3).

Figure 4 shows the combined line graph of the results obtained by evaluating the metrics for all the classifiers. The line graphs show that the classifiers performed best on the accuracy, the performance exhibited by a classifier for precision and F1 score were generally similar.

Multinomial NB exhibited a slightly different behavior, scoring better at precision (0.5235) than accuracy (0.2941). The scores were generally poor, with its accuracy score being significantly poorer than those of the other classifiers.

Figure 5 shows that the performance of the algorithms varies widely. Multinomial NB shows a poor performance, though with a modest score of 0.52 as the best expectation. Random Forest and Bagging classifiers see the highest median scores. They also have the highest Upper Whiskers, with Bagging classifier having a slightly higher 1st quartile score. Logistic Regression has a very high Upper whisker and 3rd quartile score but a noticeable inter quartile range of 0.15 means there is a significant likelihood of scoring lower. The box plot shows that Bagging and Random Forest classifiers provided the most consistently high scores for any of the metrics, with a likely score of 0.72.

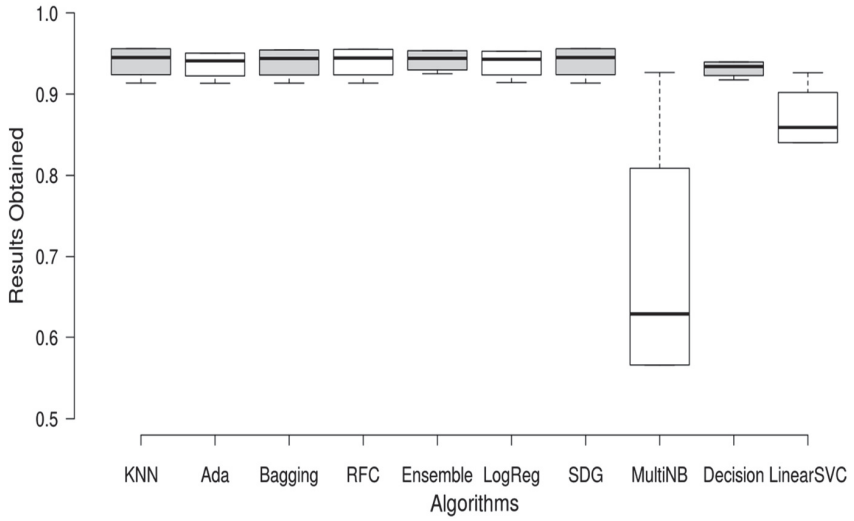

Figure 7 Analysis of results obtained against classifiers using Boxplots (Dataset 3).

Table 5 shows the results of the analysis of metrics for the classifiers using the third dataset. Evaluation metrics include accuracy, precision, F1-score and recall. The classifiers are k-neighbors classifier, Ada boost classifier, Bagging classifier, Decision tree classifier, Logistic regression, random forest classifier, multinomial Naïve Bayes, Stochastic Descent Classifier and Linear Support Vector Classifier. From the results in Table 5, all the classifiers score excellently well except Multinomial NB. Multinomial NB scored averagely in the accuracy, recall and F1 score metrics, while performing excellently well in the precision metric.

The Ensemble model yielded better results than its constituent classifiers. The results improved marginally upon those of Logistic Regression even for the accuracy metric. As a whole, the ensemble model compared favourably with other classifiers for accuracy, performed third best for precision, and had the highest F1 Score.

Figure 6 shows the combined line graph of the results obtained by evaluating the metrics for all the classifiers. The line graphs show that the classifiers generally scored very high for the metrics. The accuracy score was generally highest, closely followed by the F1 score. Multinomial NB, Linear SVC and SDG exhibited a slightly different behavior, scoring better at precision than accuracy and F1 score. Multinomial NB fares significantly worse off than the rest for F1 score and accuracy.

Figure 7 shows that all algorithms except Linear SVC score similarly high for median which means any of them (with the exception of Linear SVC) is a good choice if we are not particular about the metric we used. Bagging and Ensemble classifiers have the best combination of high upper whisker with high lower whisker while maintaining an inter quartile range of 0.2 The box plot shows that any classifier bar Linear SVC is an extremely good choice. Linear SVC has a worst performance, with a median (most likely) score of 0.61. Bagging classifier has the best combination of high upper whisker with high lower whisker while maintaining an inter quartile range of 0.2.

5 Conclusion

In this paper, machine learning approaches have been utilized to detect cyberbullying in social media networks (Twitter) while the efficiency of the algorithms has been tested and empirically confirmed. Random Forest Classifier has shown to be the best performing classifier with medians of 0.77, 0.73 and 0.94 across the datasets (datasets 1–3). The proposed Ensemble model performed well, generating results that outperformed the constituent classifiers. One of the shortcomings of this approach is that the available dataset on Twitter divulges information about users while fields such as age and gender of posters are unavailable. This study is also limited to English language tweets and data. In the future, effort will be made to transcode tweet content in other languages and also handle various challenges noticed in the process of experimentation. More so, effort shall be made in the future to use deep learning technique for better results.

References

[1] Al-garadi, M. A., Varathan, K. D., and Ravana, S. D. (2016). Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. Computers in Human Behavior, 11.

[2] Amarashinghe, T., Aponso, A., and Krishnarajah, N. (2018). Critical Analysis of Machine Learning Based Approaches for Fraud Detection in Financial Transactions. China: Association for Computing Machinery.

[3] Amrita, M. (2017). Collaborative Detection of Cyberbullying Behavior In Twitter Data. Indiana: Department of Computer Science.

[4] Astor, M. (2017, August 13). A Guide to the Charlottesville Aftermath – The New York Times. Retrieved from The New York Times: https://www.nytimes.com/2017/08/13/us/charlottesville-virginia-overview.html

[5] Aziz S., M. U. Khan, Z. Ahmad Choudhry, A. Aymin and A. Usman, "ECG-based Biometric Authentication using Empirical Mode Decomposition and Support Vector Machines," 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2019, pp. 0906–0912, doi: 10.1109/IEMCON.2019.8936174.

[6] Badawy, Adam, E. F., and Kristina, L. (2018, August). Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign. 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 258–265.

[7] Bail, C., Guay, B., Maloney, E., Combs, A., Hillygus, S., Merhout, F., ... Volfovsky, A. (2020). Assessing the Russian Internet Research Agency's impact on the political attitudes and behaviors of American Twitter users in late 2017. In A. Underdal (Ed.), Proceedings of the National Academy of Sciences Jan 2020.

[8] Bastos, M., and Farkas, J. (2019, August 6). Social Media + Society, 5(3).

[9] Bisaso, K. R., Karungi, S. A., Kiragga, A., Mukonzo, J. K., and Castelnuovo, B. (2018). A comparative study of logistic regression-based machine learning techniques for prediction of early virological suppression in antiretroviral initiating HIV patients. BMC medical informatics and decision making, 18(1), 77. https://doi.org/10.1186/s12911-018-0659-x

[10] Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., and Vakali, A. (2017, June 25-28). MeanBirds: Detecting Aggression and Bullying on Twitter. pp. 1–10.

[11] Chengsheng, Tu and Huacheng, Liu and Bing, Xu. (2017). AdaBoost typical Algorithm and its application research. MATEC Web of Conferences. 139. 00222. 10.1051/matecconf/201713900222.

[12] Daniel, D. (2017). Machine Learning for The Automated Identification of Cyberbullying and Cyberharassment. 1–146.

[13] Dey, D. (2018). ML | Bagging Classifier.

[14] Eichstaedt, J., Schwartz, H., Kern, M., Park, G., Labarthe, D., Merchant, R., ... Seligman, M. (2015, February). Psychological Language on

Twitter Predicts County-Level Heart Disease Mortality. Psychological Science, 26(2), 159–169.

[15] Feng Bao, Thomas Maier. Stochastic gradient descent algorithm for stochastic optimization in solving analytic continuation problems. Foundations of Data Science, 2020, doi: 10.3934/fods.2020001

[16] Gomez-Adorno, H., Bel-Enguix, G., Sierra, G., Sanchez, O., and Quezada, D. (2018). A Machine Learning Approach for Detecting Aggressive tweets in Spanish. Mexico City.

[17] Gutierrez-Esparza, G. O., Vallejo-Allende, M., and Hernandez-Torruco, J. (2019, May 2). Classification of Cyber-Aggression Cases Applying Machine Learning. pp. 1–17.

[18] Haidar, B., Chamoun, M., and Serhrouchni, A. (2017). A Multilingual System for Cyberbullying Detection: Arabic Content Detection using Machine Learning. Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 6, 1–10.

[19] Hani, J., Nashaat, M., Ahmed, M., and Mohammed, A. (2019). Social Media Cyberbullying Detection using Machine Learning. (IJACSA) International Journal of Advanced Computer Science and Applications, 2–6.

[20] Khaled Fawagreh, Mohamed Medhat Gaber and Eyad Elyan (2014) Random forests: from early developments to recent advancements, Systems Science & Control Engineering, 2:1, 602–609, DOI: 10.1080/21642583.2014.956265

[21] Lauw, H. W., Shafer, J. C., Agrawal, R., and Ntoulas, A. (2010). Homophily in the digital world: a LiveJournal case study. Internet Computing, IEEE, 14(2), 15e23.

[22] Klein, A. (2019). From Twitter to Charlottesville: Analyzing the Fighting Words Between the Alt-Right and Antifa. International Journal of Communication, 13, 297–318.

[23] Nandakumar, V., Kovoor, B. C., and Sreeja, M. U. (2018). Cyberbullying revelation in twitter data using naïve Bayes classifier algorithm. International Journal of Advanced Research in Computer Science, 1–4.

[24] Nandhini, B. S., and Sheeba, J. I. (2016). Cyberbullying Detection and Classification Using Information Retrieval Algorithm. India.

[25] Sampasa-Kanyinga, H., Roumeliotis, P., and Xu, H. (2014). Associations between cyberbullying and school bullying victimization and suicidal ideation, plans and attempts among Canadian schoolchildren. PLoS One, 9(7).

[26] Quinlan, J.R. Induction of decision trees. Mach Learn 1, 81–106 (1986).

[27] Rushing, W. (2018, April 3). After Charlottesville. Contexts, 17(1), 16–27.
[28] Whittaker, E., and Kowalski, R. M. (2015). Cyberbullying via social media. Journal of School Violence, 14(1), 11e29.
[29] Zhang, H (2004) The Optimality of Naïve Bayes.
[30] Zhang, Z (2016) Introduction to Machine Learning: K-Nearest Neighbors. Ann Transl Med. 2016, 4(11): 218 doi: 10.21037/atm.2016.03.37
[31] Azeez NA, Ayemobola TJ, Misra S, Maskeliūnas R, Damaševičius R(2019). "Network Intrusion Detection with a Hashing Based Apriori Algorithm Using Hadoop MapReduce". Computers. 2019; 8(4):86.
[32] Azeez, N.A., Salaudeen, B.B., Misra, S.; Damasevicius, R; Maskeliunas, R (2019) "Identifying Phishing Attacks in Communication Networks using URL Consistency Features", International Journal of Electronic Security and Digital Forensics (InderScience). https://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijesdf
[33] Azeez, N.A.; Odufuwa, O.E.; Misra, S.; Oluranti, J.;Damaševičius, R.(2021) Windows PE Malware Detection Using Ensemble Learning. Informatics 2021, 8, 10. https://doi.org/10.3390/informatics8010010

Biographies

Nureni Ayofe Azeez obtained his B.Tech. (Hons.) from the Federal University of Technology, Akure, Nigeria in 2005, MSc from the University of Ibadan, Oyo State, Nigeria in 2008, and Ph.D. from University of the Western Cape, South Africa in 2013, all in Computer Science. His areas of research include Security & Privacy, Access Control, Grid and Cloud Computing, Sensor Networks, E-Health and ICT4D. He is a recipient of The Young Scientist Award at the 22nd International CODATA Conference that was held in Cape Town, South Africa in October 2010. He is currently a Senior Lecturer in the Department of Computer Sciences, University of Lagos, Nigeria.

Sunday O. Idiakose graduated with a B.Sc. (Hons) in Computer Science with Second Class Upper Division from the University of Benin, Edo State, Nigeria in 2015. He observed his mandatory National Youth Service Corps (NYSC) programme in Imeko, Ogun State between 2016 and 2017. He has recently defended his M.Sc. programme thesis in Computer Science at the University of Lagos, Lagos, Nigeria. His area of interests include cloud computing, cyber security and distributed systems.

Chinazo Juliet Onyema earned a (B.Tech) (honour) degree in Mathematics and Computer Science from the Federal University of Technology Owerri (FUTO), Nigeria and Masters (MSc) degree in Computer Science from the University of Lagos, Nigeria. She is currently an assistant lecturer in the Department of Computer Science, Federal University of Technology Owerri (FUTO), Nigeria. Her research interests include Cyber Security and Internet of Things (IoT). She is a member of Nigeria Computer Society (NCS).

Charles Van Der Vyver obtained his B.Sc. from the Potchefstroom University for Christian Higher Education, Vanderbijlpark, South Africa in 2003, B.Sc. Hons in 2004, M.Sc. in 2007 and Ph.D. in 2011, all from the North-West University, Vanderbijlpark, South Africa, all in Computer Science. His areas of research include Security & Privacy, Water Poverty and Water Management. He is a recipient of a best paper award in 2015 in Kuala Lumpur, Malaysia. He delivered the keynote address during a conference in London, United Kingdom in 2019. He is the recipient of several Faculty and institutional research awards. He is currently a Senior Lecturer in the School of Computer Science and Information Systems, Faculty of Natural- and Agricultural Sciences, North-West University, Vanderbijlpark, South Africa.